Threat Modeling in Cyber-Physical Systems

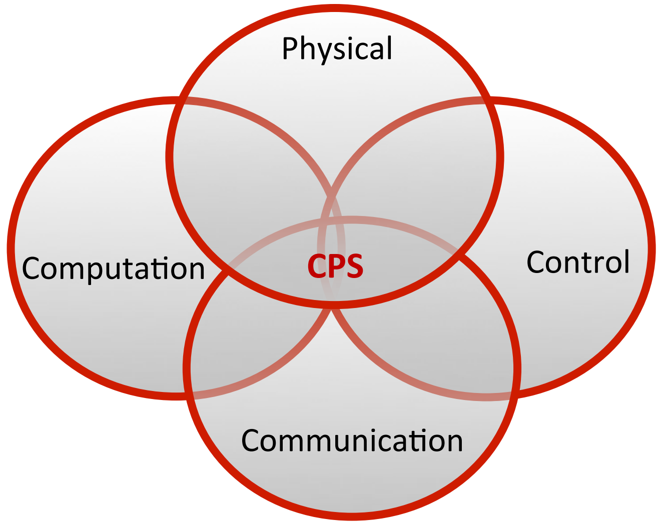
May 16, 2017

By
Emeka Eyisi Ph.D.
Mark Moulin Ph.D.
Devu Manikantan Shila Ph.D.

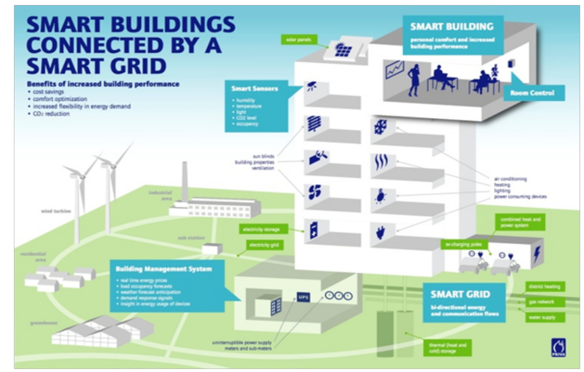
United Technologies Research Center

Cyber-Physical Systems (CPS)

Smart Home



Smart Building



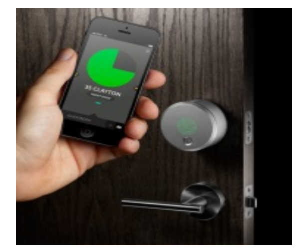
NG-Aerospace

Attacks on CPS

Smart Bulb Hacking

Vehicle Hacking

Attacker



Smart Lock Hacking

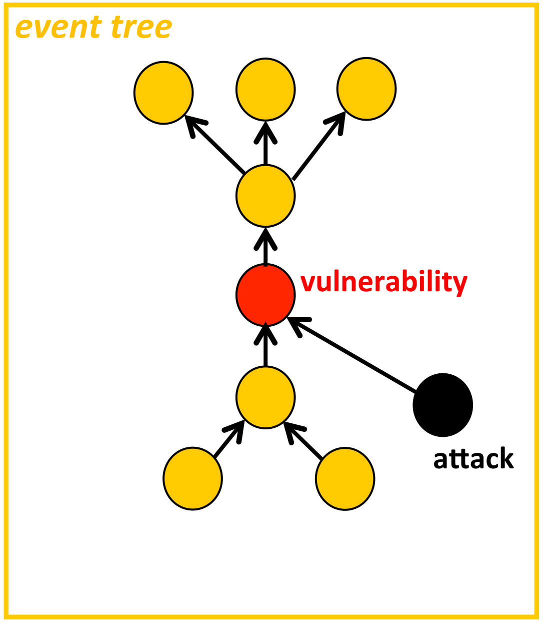
“RANSOMWARE”

CPS Attacks (Common Methods)

Attack Name	Impact	Source
Rogue Node	Breach of system integrity	Physical space
Communication Jamming	Loss of network availability	Physical space
Denial of Service	Increase network load; Loss of network availability	Physical space; Rogue node
Black Hole	Breach of network integrity. Loss of network availability	Compromised network
Gray Hole	Breach of network integrity. Loss of network availability	Compromised network
Network Isolation	Breach of network integrity. Loss of network availability	Compromise network nodes; Black hole attack
Packet Sniffing	Breach of confidentiality of communication	Access to a network; Rogue node
Fuzzing	Disclose network messages	Access to a network
Password Cracking	Breach of authenticity	Brute-force attack
Firmware Modification	Breach of firmware integrity	Modify firmware of devices on same network
Code Injection	Breach of confidentiality/integrity	Firmware modification
False Data Injection (Communication based)	Breach of data integrity	Network Authentication
False Data Injection (Database-based)	Breach of data integrity	Database access control
False Data Injection (Sensor based)	Breach of data integrity	Compromised system
Pointer Attack	Manipulating a pointer	Compromised system
Malware Infection	Breach of system integrity and properties	Compromised system
Command Injection	Breach of integrity	Fuzzing; Packet sniffing; Rogue node
Relay Attack	Breach of authenticity	Physical space; Transmitted signal capture
Replay Attack	Breach of authenticity and integrity	Access to communication

Problem Statement and Motivation

- Most of the exploitations found today can be prevented by fixing errors in design, implementation and installation
- Security analysis are typically exercised after design stage - forcing relaxation of trust assumptions (use weak trust models)
- Attacks graphs (trees) provide an useful way of modeling the vulnerabilities of a system and potential exploits during the design stage
- Manual construction of graphs very tedious and error-prone

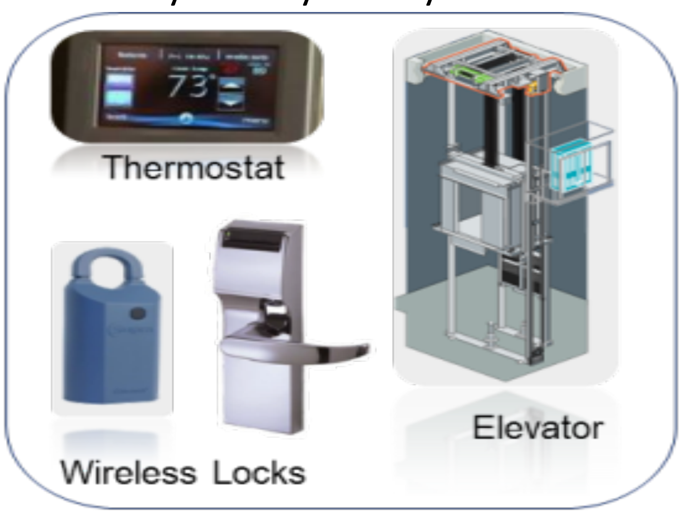


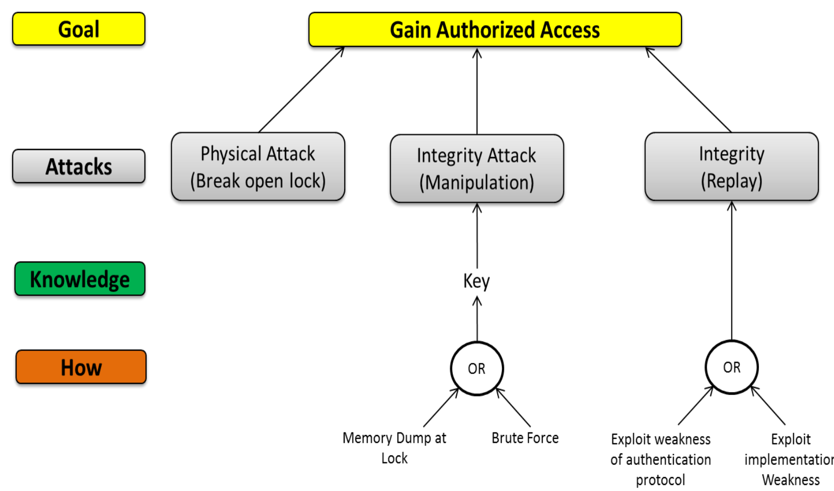
Automatically analyze the security posture of heterogeneous and complex cyber physical system designs against a holistic set of threat models (known and emerging)

ATTACK GRAPHS

- Attack Graph (AG) is a collection of scenarios showing how a malicious agent can compromise or violate the security property of the system model in variety of situations to reach the specific goal:
  - What are the ways that an attacker can reach a specific goal?
  - What is the highly probable path for an attacker?
  - What countermeasures shall a defender deploy?
  - What is the minimal set of components that needs to be protected so that attacker cannot achieve the goal?

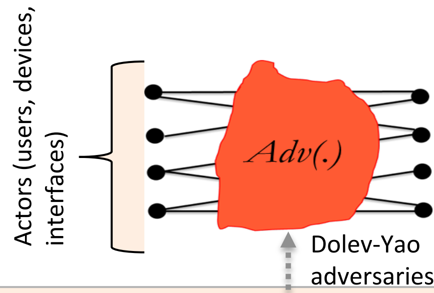
Cyber Physical Systems





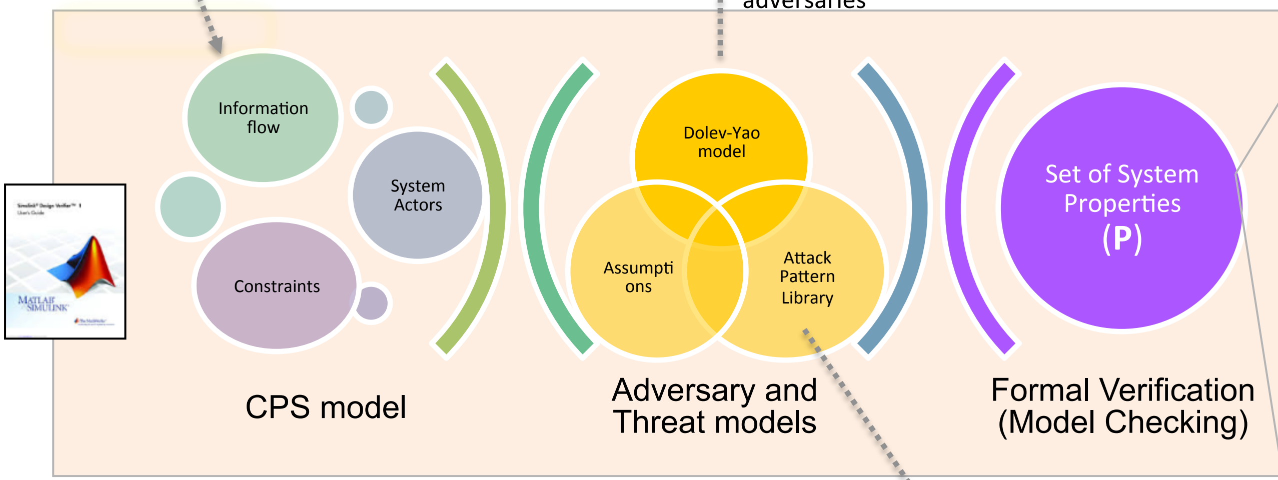
Formal Verification-Based Attack Tree Generation

Cyber Physical Systems

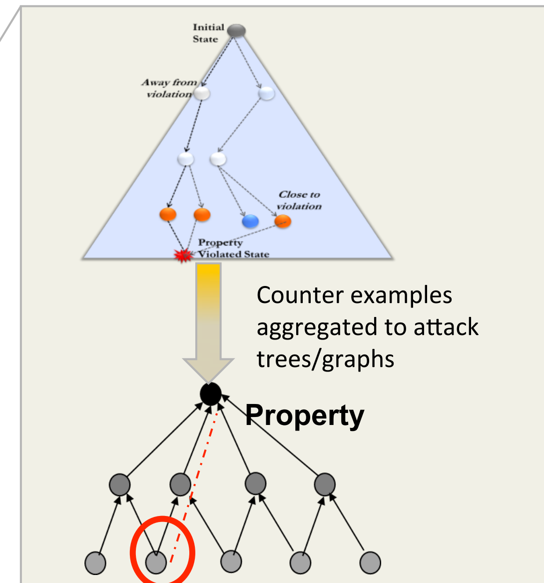


Three steps to produce attack graphs

1. Identify system vulnerabilities or critical points (based on adversary and threat models) – Sub-goals of an attacker
2. Operational system impact: Violation of properties (P)
3. Aggregation of counterexamples to attack graph



Attack Pattern Library

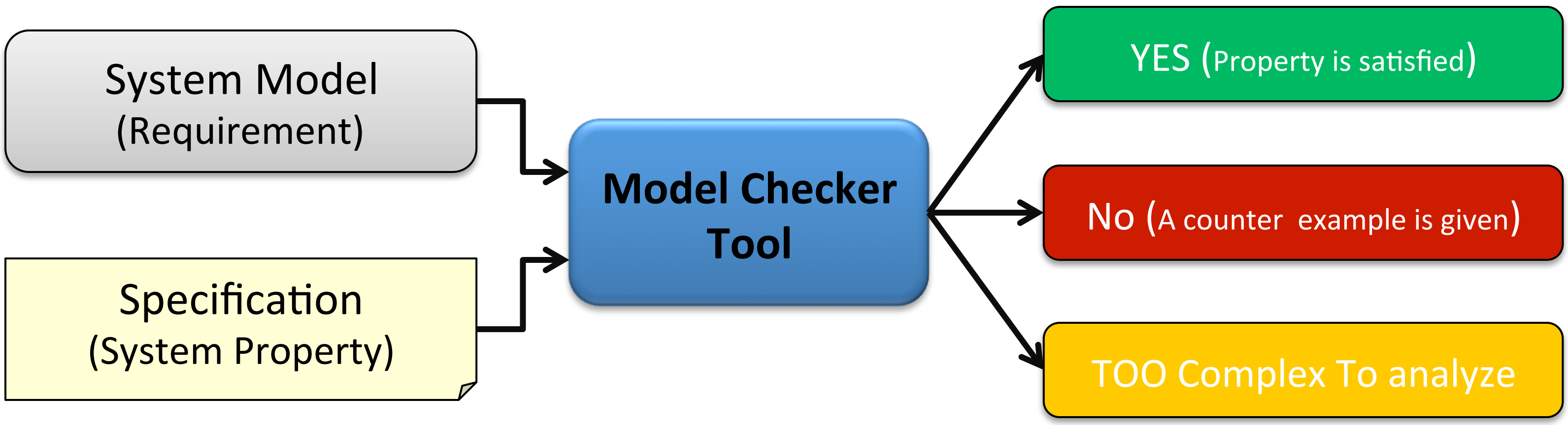


- What are the critical components or elements that needs to be secured?
- What are the minimum set of defenses?
- What is the effectiveness of a given countermeasure

Formal Verification (Model Checking)

Model Checking

- Automatic, model-based, property-verification approach
- Mathematically analyze system properties and models
- Exhaustively check that no test case exists that can lead to a violation of specification
  - ➢ If any exists, an example of such test case is returned



Specification

Temporal Logic

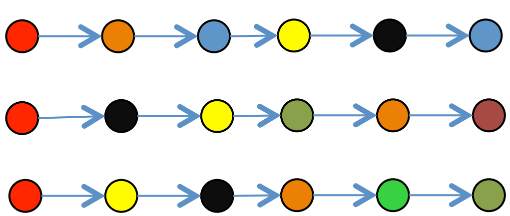
- Express properties of event ordering in time without explicitly introducing time
- Examples LTL, CTL, CTL*, MTL, HyperLTL etc.
- Differ in
  - Syntax
  - Semantics/Meaning
  - Properties that can be expressed
  - Complexity – efficiency of evaluating a property
  - Underlying model of time.

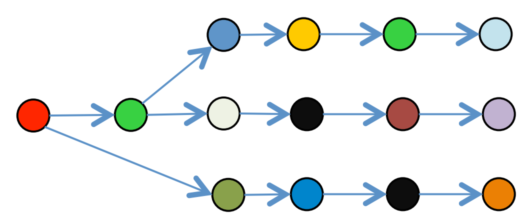
Linear Time Logics

- Each moment in time has a unique possible successor
- Example Linear-time Temporal Logic

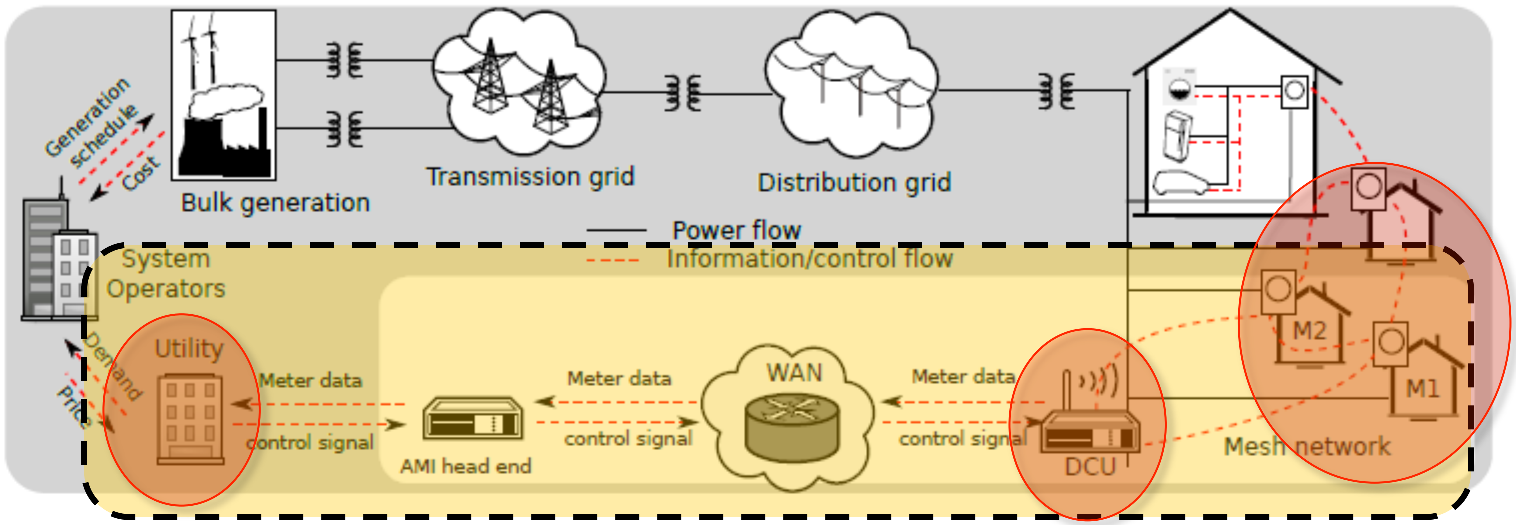
Branch Time Logic

- Model of time is a tree-like structure and each moment in time can several possible successors
- Example Computation Tree Logic (CTL)





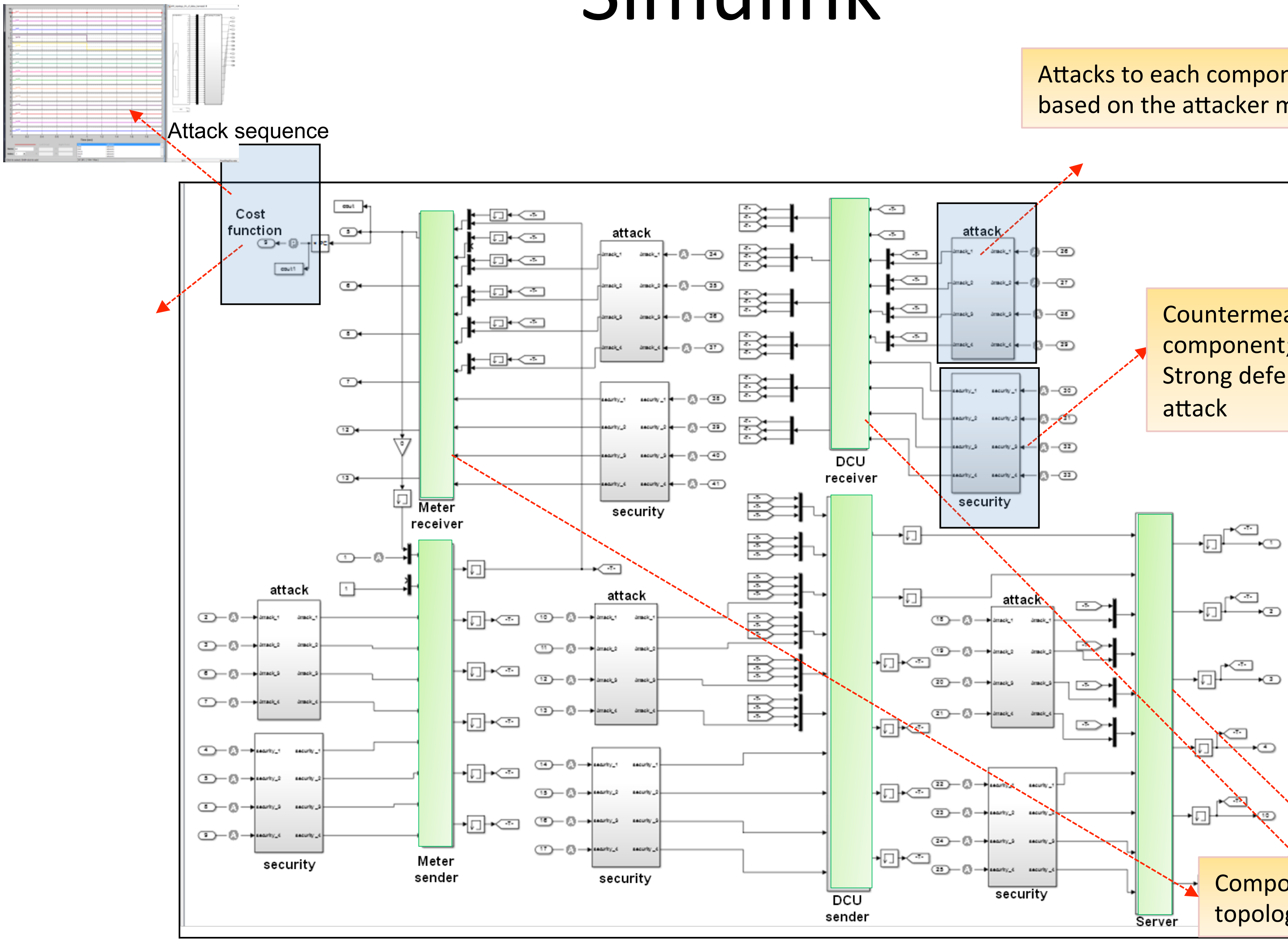
Smart Grid AMI Architecture



Smart grid topology (exchanging meter data, control signal with AMI)

- Security properties investigated:
  - Blackout (unavailability or corruption of meter data)
- Attacker model considered:
  - Physical access, local access, remote access
  - Attacker affects vulnerabilities at each component and supply voltage level
- Effects of countermeasures at each component
- Information flow between components (meter data, control signal)

Smart Grid AMI Model Checking with Simulink

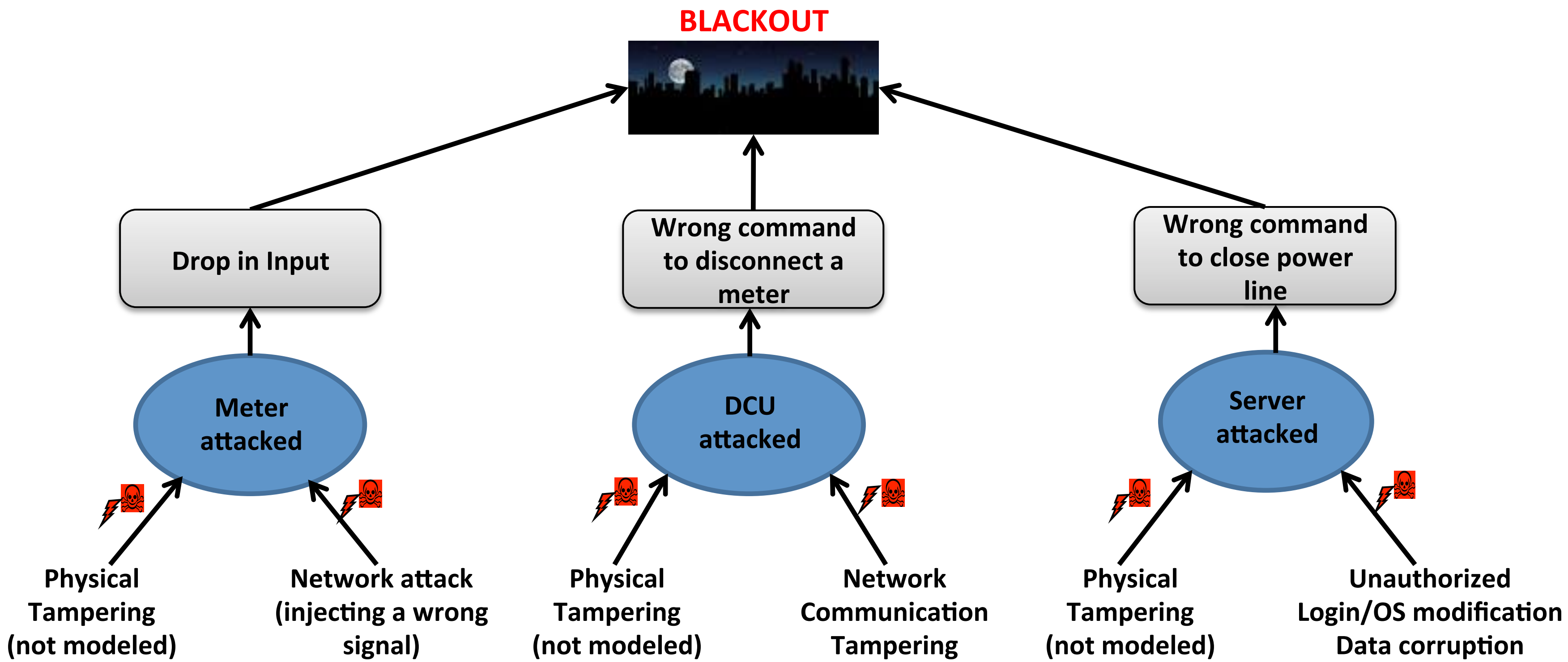


Attacks to each component based on the attacker model

Countermeasures for each component;
Strong defense nullifies the attack

Components of the topology

Smart Grid AMI Modeling and Properties



System property

- **Non-existence** of **<u>Blackout</u>**

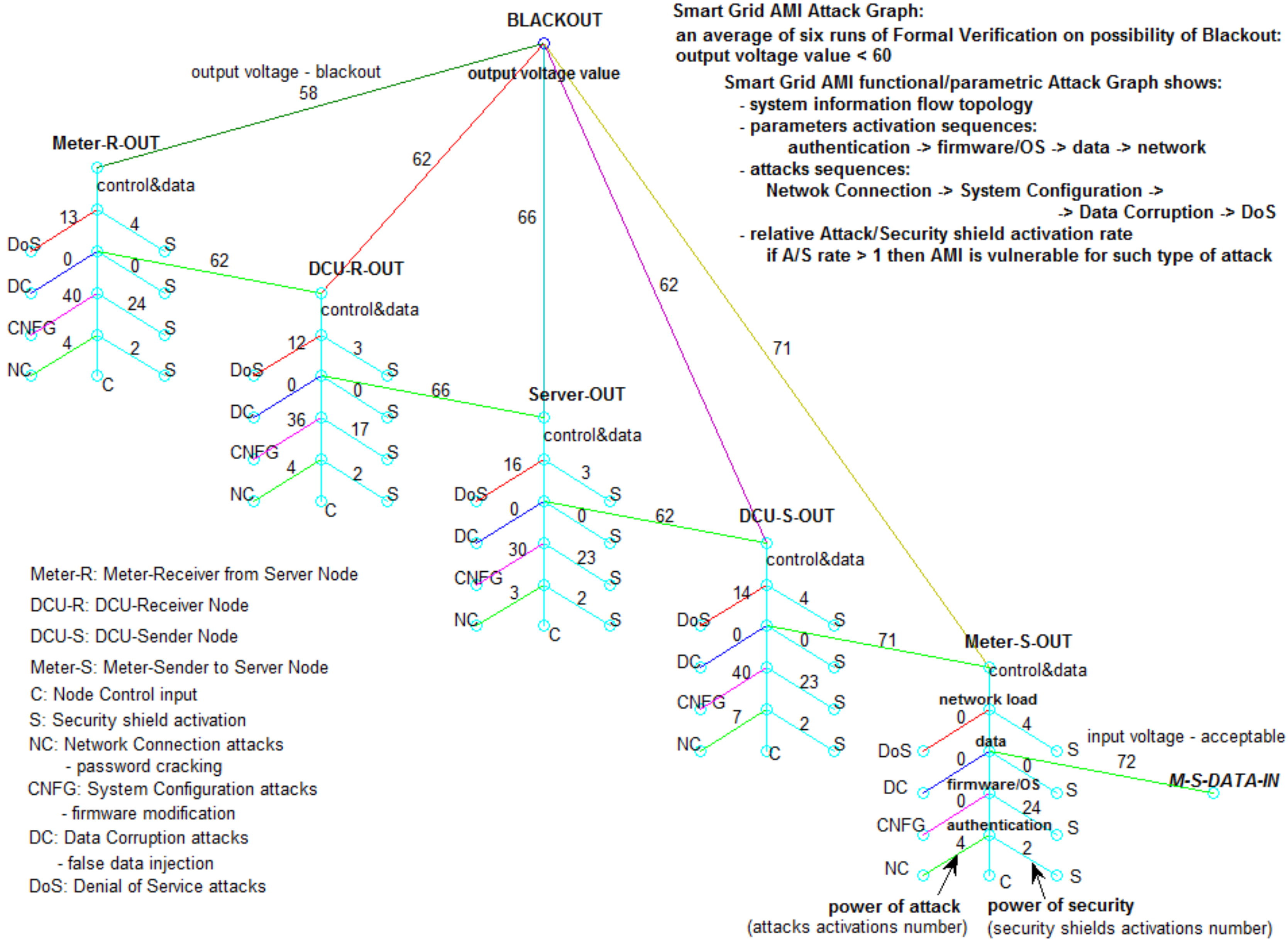
Modeling methodology

- Protocol information flow is modeled in Simulink as a modular system.
- Data (messages) encryption algorithms are modeled as arithmetical functions of scalable complexity.

Validation

- System is tested according to AG flow and FV counterexamples scenarios

Smart Grid AMI Attack Graph



Smart Grid AMI Attack Graph:

an average of six runs of Formal Verification on possibility of Blackout: output voltage value < 60

Smart Grid AMI functional/parametric Attack Graph shows:

- system information flow topology
- parameters activation sequences:
  authentication -> firmware/OS -> data -> network
- attacks sequences:
  Netwok Connection -> System Configuration -> -> Data Corruption -> DoS
- relative Attack/Security shield activation rate
  if A/S rate > 1 then AMI is vulnerable for such type of attack

Meter-R: Meter-Receiver from Server Node

DCU-R: DCU-Receiver Node

DCU-S: DCU-Sender Node

Meter-S: Meter-Sender to Server Node

C: Node Control input

S: Security shield activation

NC: Network Connection attacks
- password cracking

CNFG: System Configuration attacks
- firmware modification

DC: Data Corruption attacks
- false data injection

DoS: Denial of Service attacks

Conclusion and Future Work

- Secure-In-Design is important and vital in ensuring long term solutions for CPS
- Attack Graphs provide promising methodology for capturing vulnerabilities and exploiting paths and mechanisms
- Exploring the Integration of Formal Verification and Machine Learning in the synthesis of attack graphs