The Monitoring Core (M-Core): Toward Fully Securing Heterogeneous Wireless Sensor Networks

CAP Group @ Georgia Institute of Technology – *Georgia State University

Marco Valero*, Sang Shin Jung, A. Selcuk Uluagac, Yingshu Li*, and Raheem Beyah

MAIN IDEA

The M-Core is a modular, extensible and lightweight security layer that gathers relevant data for the development of defense mechanisms. Similar to Metasploit, which significantly reduces the time to manufacture an exploit, the M-Core is being developed to reduce the design and development time for new detection and defense mechanisms for WSNs.. The M-Core allows for the monitoring of both internal and external threats simultaneously facilitating the execution of new or existing detection and defense techniques against different attacks in parallel.

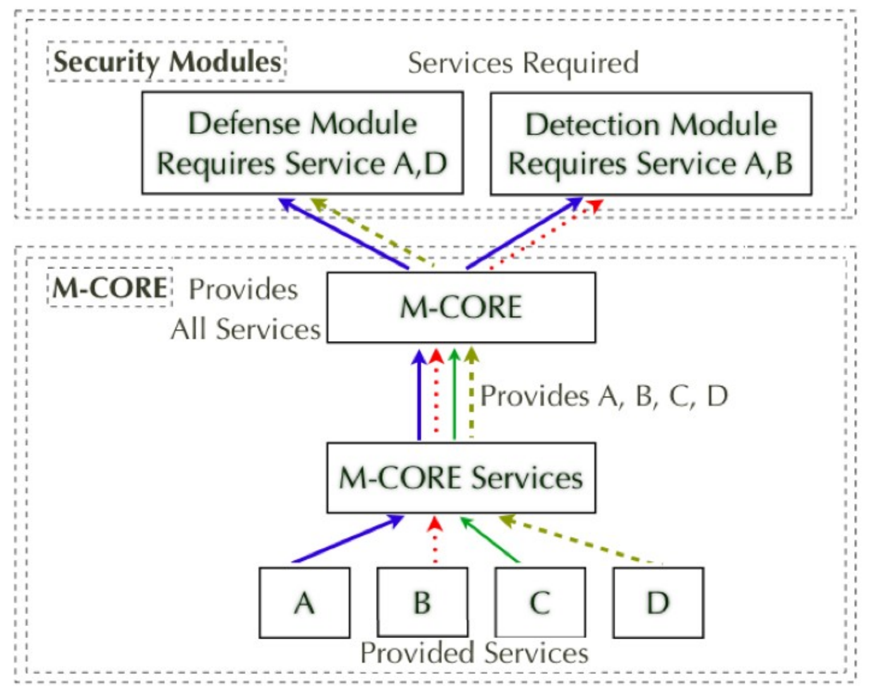
THE MONITORING CORE (M-CORE)

M-Core: To Support Distributed Security Systems

- Wireless Sensor Networks (WSNs) are deployed for monitoring in different domains (e.g., health care, military, critical infrastructure) and should be resilient to attacks.
- The problem with the traditional approach to defending sensor networks is that the solution for the *Jamming* attack does not defend against other attacks (e.g., *Sybil* and *Selective Forwarding*).
- M-Core addresses the challenges with the traditional approach to securing sensor networks and presents a comprehensive framework that can defend against all known and forthcoming attacks.
- M-Core has a built-in modular and flexible software architecture that provides an easy means to add, remove, and replace sub-modules. It is a lightweight monitoring and control layer invisible to upper layers.

M-CORE ARCHITECTURE

The *M-Core services* module advertises all the services provided by the sub-modules to the *M-Core* module, and the *M-Core* module allows the security modules to access those services.



To implement a defense mechanism against Sybil Attacks, our *sybil* module uses the *rssivalue* interface (service) provided by the RSSI sub-module of the M-Core.

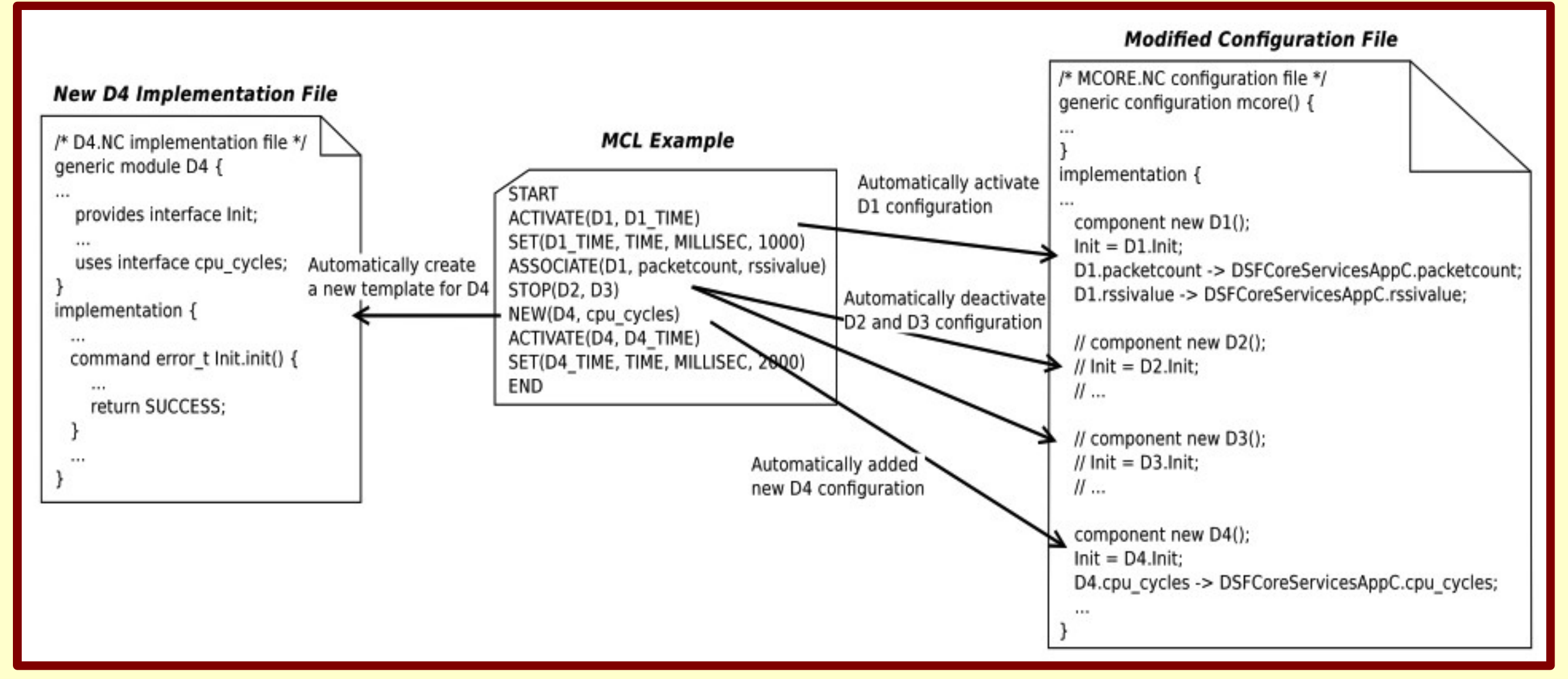
M-CORE SERVICES

Component	Interface	Commands/Events	Action
channelScan	channelinfo	getConsecutiveSuccess	Returns the number of consecutive successfully sent packets
		getpps	Returns the number of received packets per second
		setThreshold	Sets the threshold for acceptable consecutive sent packets rate
packetCount	packetcount	getPacketCount	Returns the total number of received packets
		lostPacket	Returns the number of packets lost by each node
RSSI	rssivalue	getRssiTable	Returns neighbors RSSI table
		initRssiTable	Initializes the neighbors RSSI table
Sensing	sensingstat	getAvgSenseValue	Returns the average sensed value aggregated at the M-Core
neighborsComm	neighbors	request	Triggers a neighbor discovery message
currentNeighbors	neighborsinfo	getNeighbors	Returns the number of current neighbors
		initNeighbors	Initializes current neighbors table
packetInformation	packetsinfo	getTable	Return the packets information table
		initTable	Initializes packet information table

THE M-CORE CONTROL LANGUAGE (MCL)

- Facilitates the use of the M-Core.
- Simplifies the development of new defense mechanisms.
- Utilizes the M-Core sub-modules to activate, deactivate or create new defenses.
- Generates all programming components needed for the underlying sensor software architecture (e.g., configuration files, module files and wiring).

MCL EXAMPLE


Automatically create a new template for D4
Automatically activate D1 configuration
Automatically deactivate D2 and D3 configuration
Automatically added new D4 configuration

New D4 Implementation File

```
/* D4.NC implementation file */
generic module D4 {
...
   provides interface Init;
   ...
   uses interface cpu_cycles;
}
implementation {
  ...
  command error_t Init.init() {
     ...
     return SUCCESS;
  }
  ...
}
```

MCL Example

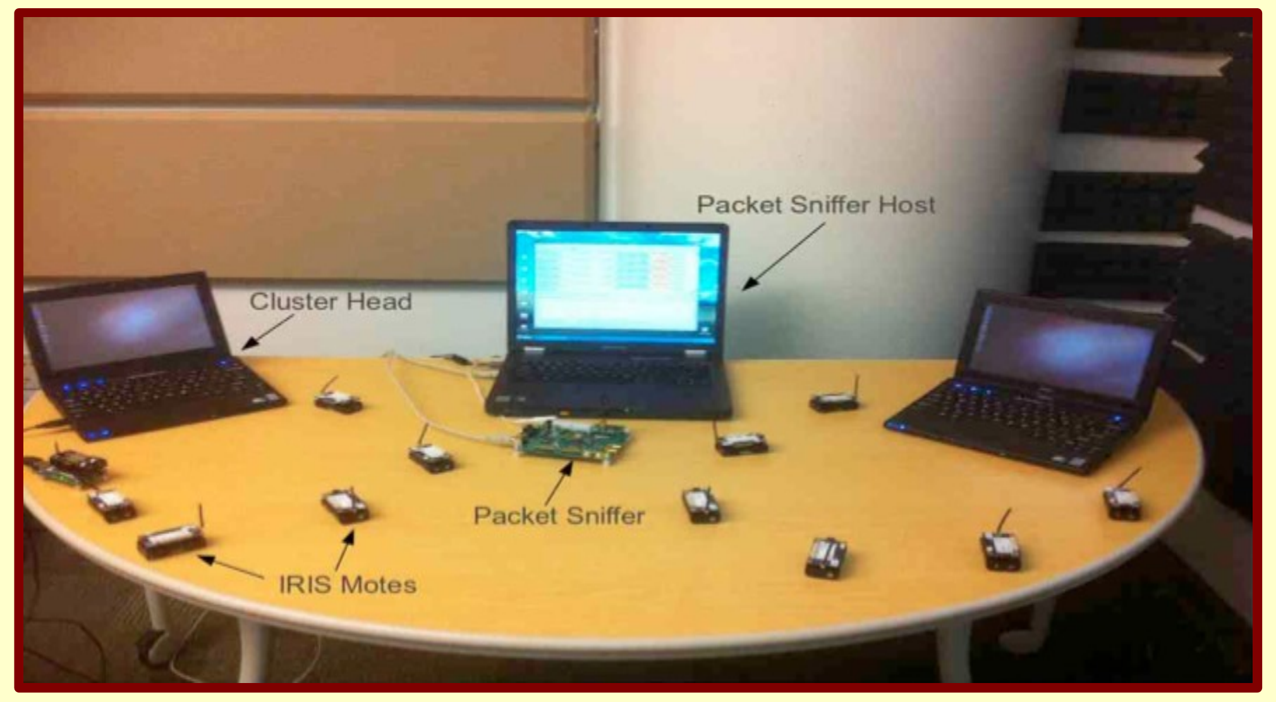
```
START
ACTIVATE(D1, D1_TIME)
SET(D1_TIME, TIME, MILLISEC, 1000)
ASSOCIATE(D1, packetcount, rssivalue)
STOP(D2, D3)
NEW(D4, cpu_cycles)
ACTIVATE(D4, D4_TIME)
SET(D4_TIME, TIME, MILLISEC, 2000)
END
```

Modified Configuration File

```
/* MCORE.NC configuration file */
generic configuration mcore() {
...
}
implementation {
...
  component new D1();
  Init = D1.Init;
  D1.packetcount -> DSFCoreServicesAppC.packetcount;
  D1.rssivalue -> DSFCoreServicesAppC.rssivalue;

  // component new D2();
  // Init = D2.Init;
  // ...

  // component new D3();
  // Init = D3.Init;
  // ...

  component new D4();
  Init = D4.Init;
  D4.cpu_cycles -> DSFCoreServicesAppC.cpu_cycles;
  ...
}
```

Using M-Core and MCL, we implemented detection and defense mechanisms against *Jamming, Selective Forwarding, Sybil, and Internal* attacks representing different layers of the communication stack simultaneously on sensors.

TESTING M-CORE



M-CORE INFO

Memory Footprint

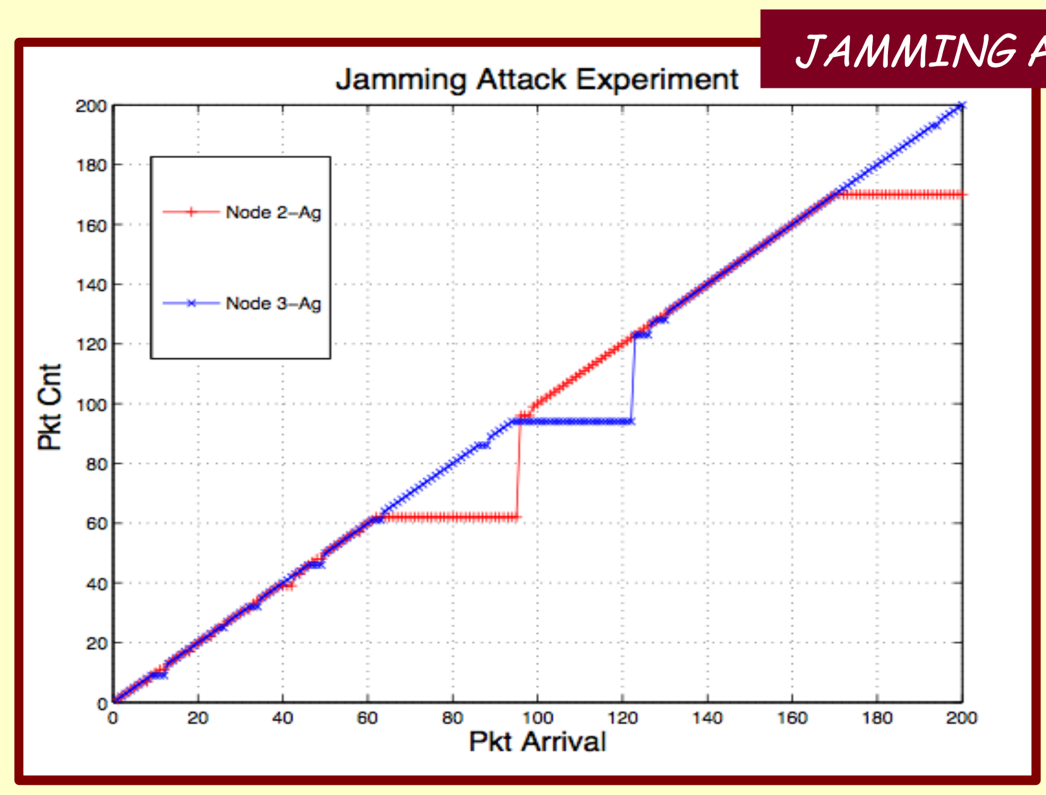
TYPE	ROM	RAM
Plain	18172	1641
Plain w/Security	20838	1743
M-Core-Plain	19328	2236
M-Core(Security)	23024	2346
M-Core(Security+Sensing)	30954	2507

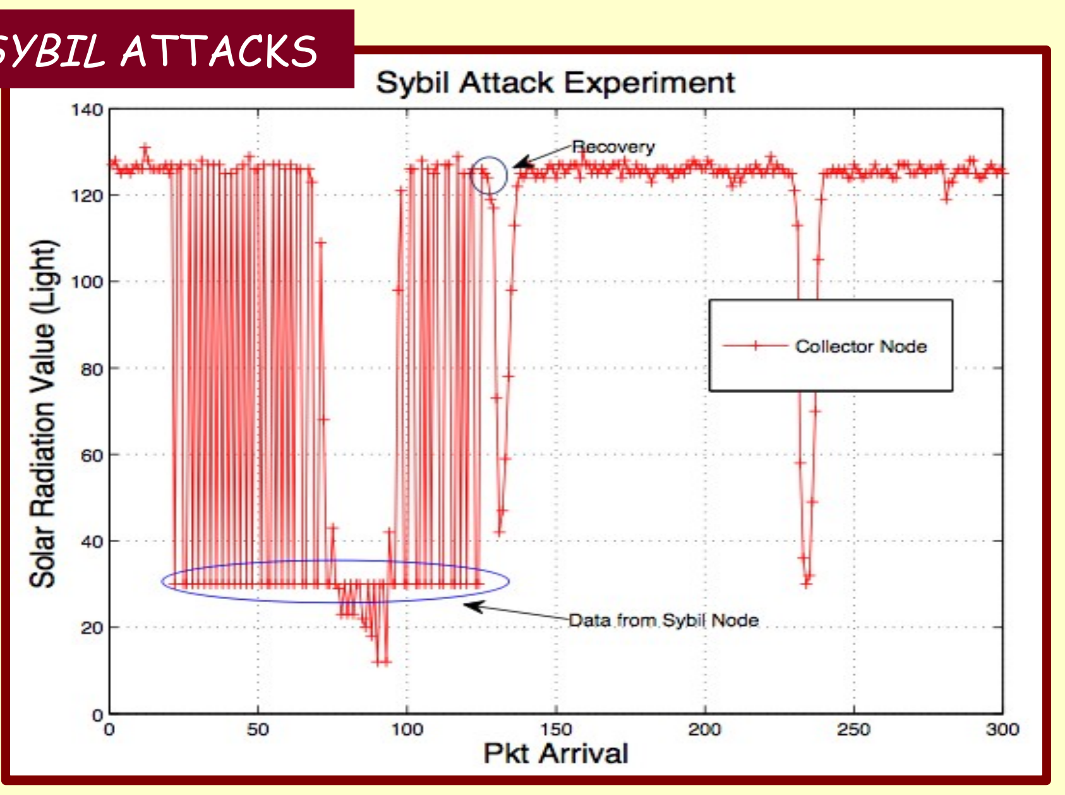
CPU Ticks

	M-Core	Plain	Diff
TX	352	239	113
RX	1600	1600	0
Sensing	550	546	4

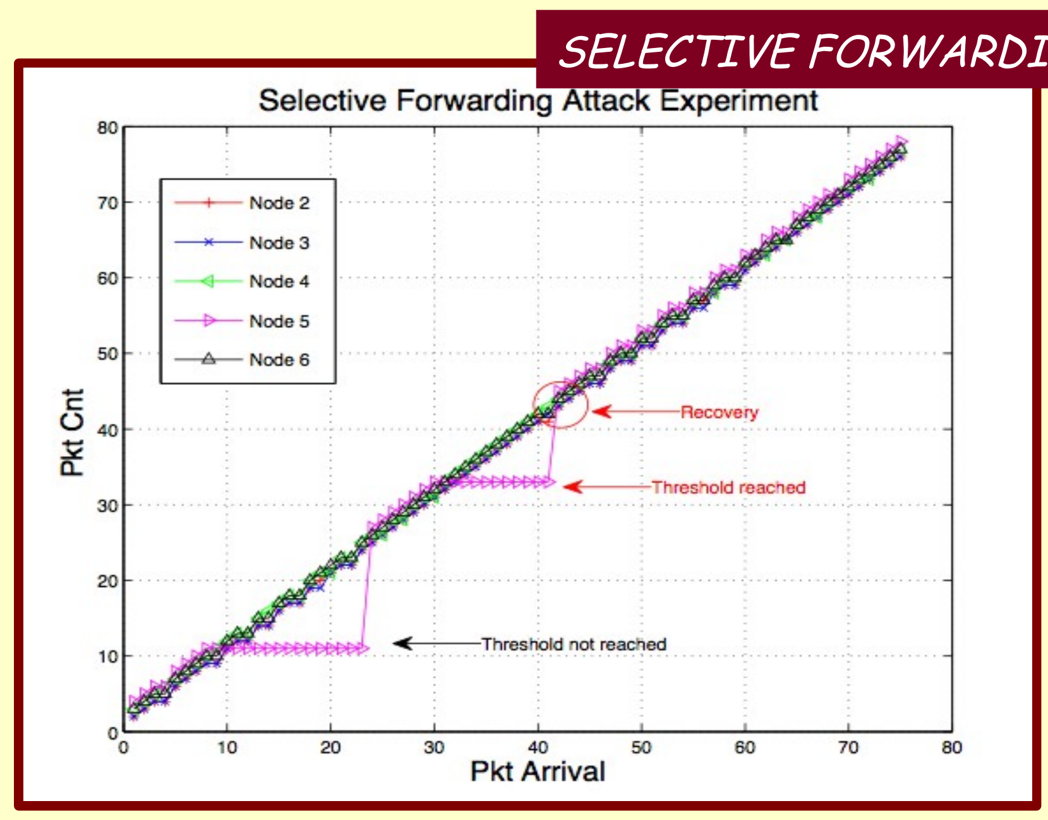
DEFENSE MECHANISMS USING M-CORE

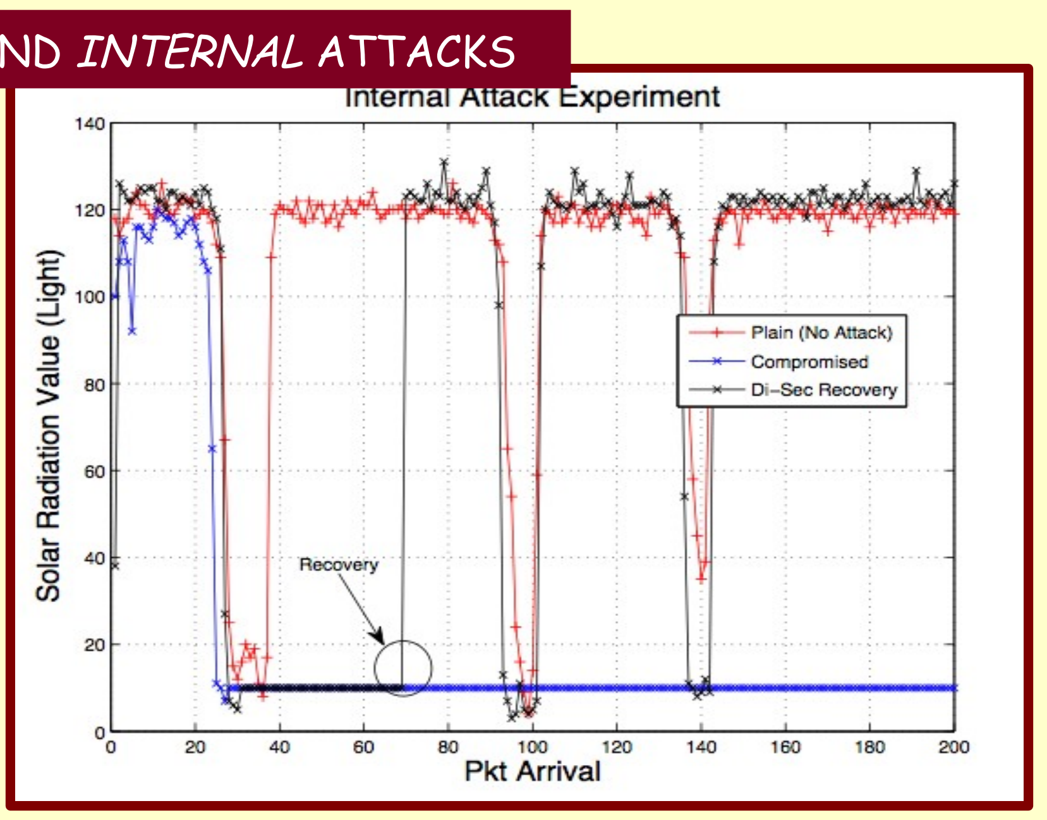
JAMMING AND SYBIL ATTACKS





SELECTIVE FORWARDING AND INTERNAL ATTACKS





REFERENCES

- Marco Valero, Sang Shin Jung, Arif Selcuk Uluagac, Yingshu Li, and Raheem Beyah. "Di-Sec: A Distributed Security Framework for Heterogeneous Wireless Sensor Networks." To appear in the Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), March 2012.
- A.Selcuk Uluagac, C. Lee, R. Beyah, J. Copeland,."Designing secure protocols for wireless sensor networks," Proceedings of the 3rd International Conference on Wireless Algorithms Systems and Applications (WASA), Oct., 2008
- A. Pathan, H.-W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," in Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, vol. 2, Feb. 2006.
- M. Healy, T. Newe, and E. Lewis, "Security for wireless sensor networks: A review," in Sensors Applications Symposium, 2009. SAS 2009. IEEE, Feb. 2009.