IoT Security and Privacy

CPS and IoT Security

Alessandro Brighente

Master Degree in Cybersecurity

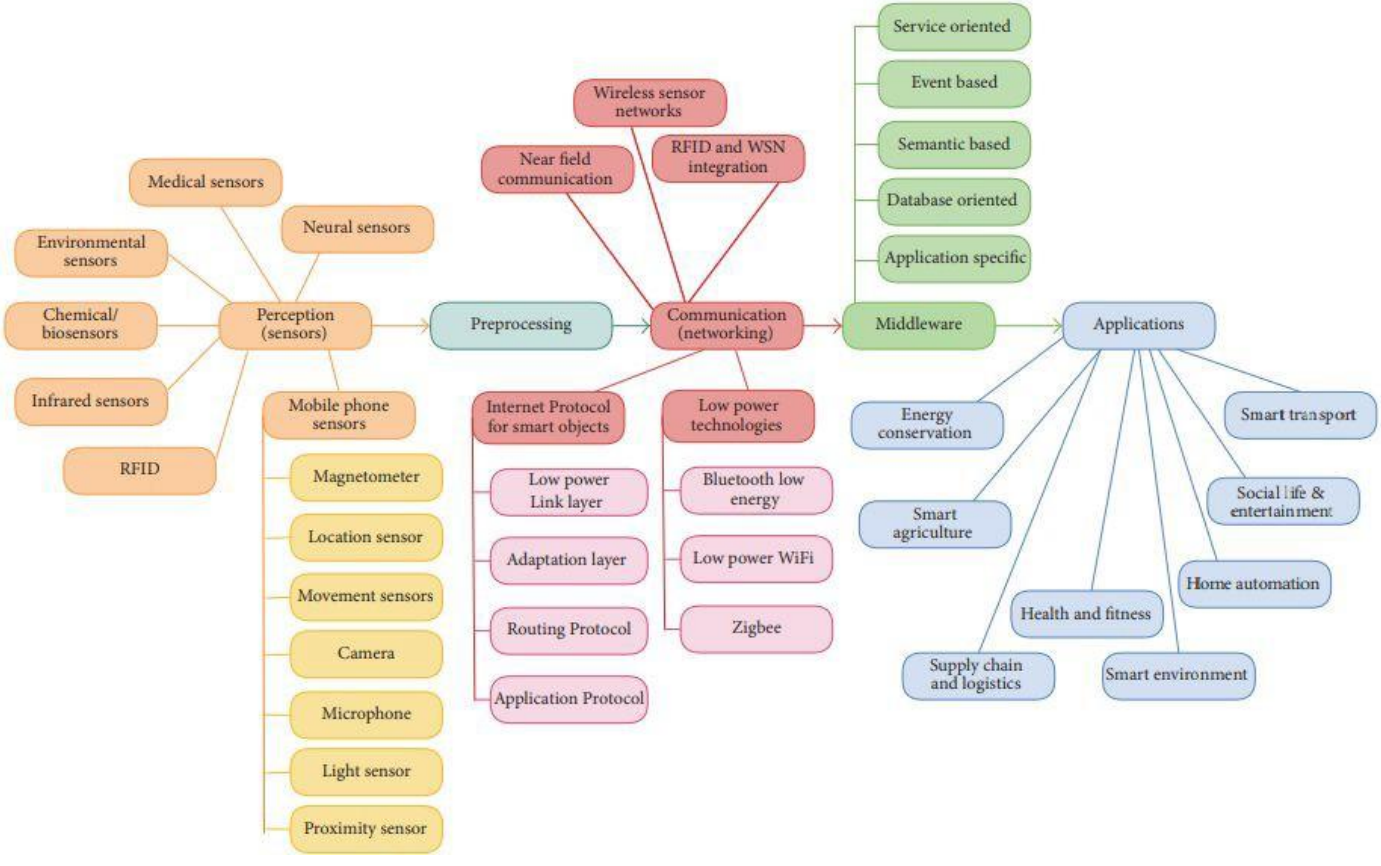
UNIVERSITÀ DEGLI STUDI DI PADOVA

SPRITZ SECURITY & PRIVACY RESEARCH GROUP

The Internet of Things

- An Internet of Things (IoT) describes a group of physical devices equipped with sensing, processing, and communication capabilities able to exchange information with each other over the Internet or other communication networks
- It is a result of development in different fields, including embedded devices, sensor networks, automation, and control systems
- We have already seen examples of internet of things

IoT Taxonomy



IoT Architecture

- An IoT system consist of three main layers: i) devices, ii) edge gateway, iii) cloud
- Devices are the *things*, i.e., those devices equipped with sensors and actuators that collect data and report it to the gateway
- Gateway is a data aggregation system to pre-process data, securing connectivity to cloud, the event hub, and sometimes fog computing
- Cloud contains the applications built using microservices, storage, event queuing, and messaging systems

Network Architecture

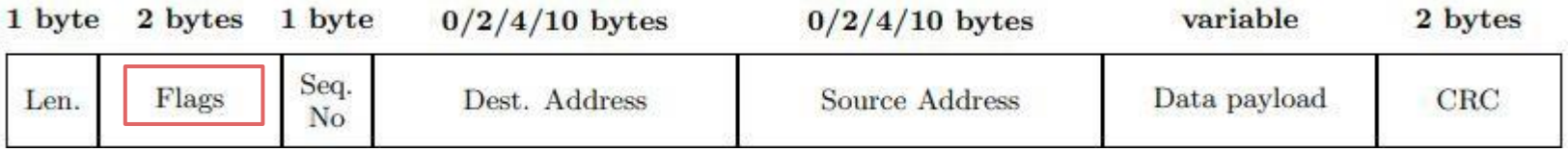
SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- We expect IoT networks to comprise a huge number of devices
- We use IETF IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN)
- Usually operates on top of IEEE 802.15.4 defined for low rate - PAN
- For industrial networks, we have IPv6 over TSCH model of IEEE 802.15.4e (6TiSCH)
- Data transport is provided by IETF Constrained Application Protocol (CoAP), ZeroMQ, and MQTT

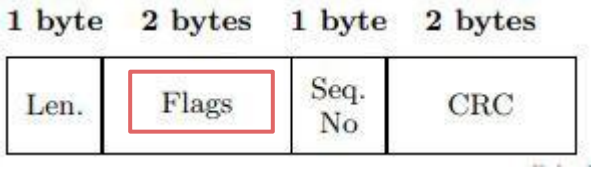
- Low Rate PAN standard specifying lower protocol layers (physical and MAC)
- Addressing uses a 64 bit node ID and 16 bit net ID
- Basic channel access mode is carrier-sense multiple access with collision avoidance (CSMA/CA)
- Check whether channel is occupied, if not send a RTS packet and wait for CTS
- If CTS received, send packet
- Uses data packets and ack packets

Data Packet format



Also indicates whether security is enabled

ACK Packet format



IEEE 802.15.4 Security

- A link layer security protocol needs to provide four basic security services: access control, message integrity, message confidentiality, and replay protection
- In 802.15.4 security is handled at the media access control layer
- The application specifies the security stack and sets the appropriate control parameters
- Security is not enabled by default

- An application has a choice of *security suites* that controls the type of security protections provided for the transmitted data
- It defines eight different security suites
  - No security
  - Encryption only (AES-CTR)
  - Authentication only (AES-CBC-MAC)
  - Encryption and authentication (AES-CCM)
- Each category that supports authentication comes into three variants depending on the size of the MAC (4, 8, or 16 bytes)

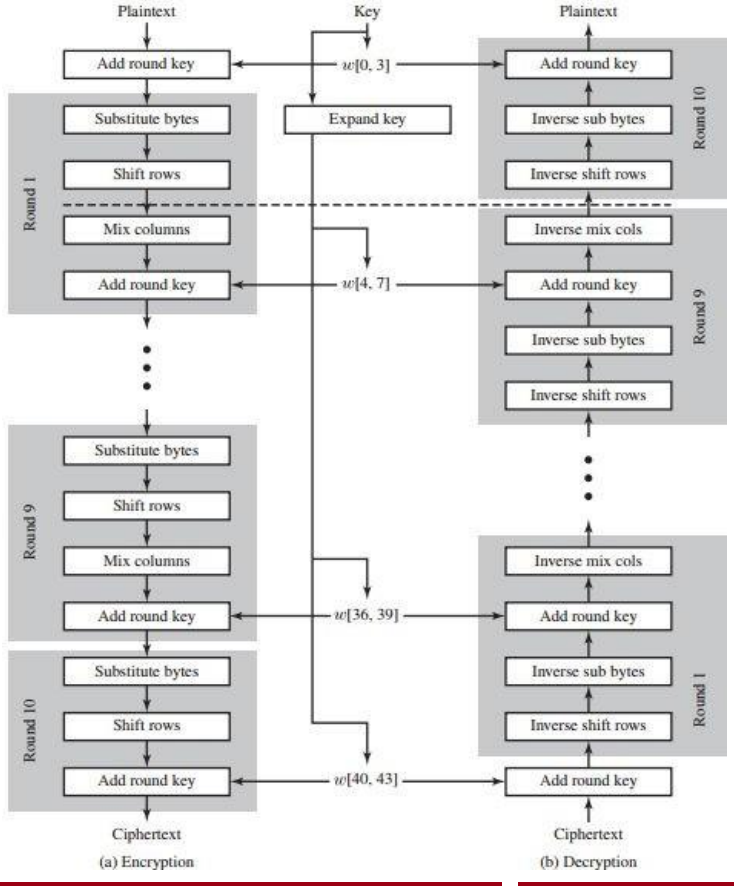
Advanced Encryption Standard

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- The Advanced Encryption Standard (AES) is one of the most famous block ciphers
- It is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001
- For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits

Advanced Encryption Standard



Advanced Encryption Standard

SPRITZ
Security & Privacy
Research Group

Università
degli Studi
di Padova

- The input of the encryption and decryption block is a single 128-bit block
- This block is copied into the **state** array, which is modified at each stage of encryption and decryption
- After the final stage, **state** is copied to an output matrix
- The 128 bit key is depicted as a squared matrix
- The key is then expanded into an array of key schedule words

Advanced Encryption Standard

- The key is expanded into an array of key schedule words, each 4 bytes long and the total key schedule is 44 words for the 128-bit key
- A **key schedule** process is a strategy to derive multiple keys from a single longer key
- **Key Expansion:** The process of generating subkeys from the original key is often referred to as key expansion. It involves applying various operations to the original key to produce a set of round keys, which are used in multiple rounds of encryption or decryption

- **Round Keys:** The subkeys generated by the key schedule are often called round keys because they are used in each round of the encryption or decryption process. These round keys are derived from the original key using the key schedule algorithm.
- **Round Constants:** In some key schedule algorithms, additional round constants are used to enhance the security of the key expansion process. These constants are added to the intermediate key values to introduce additional entropy and prevent attacks such as key recovery.

Advanced Encryption Standard

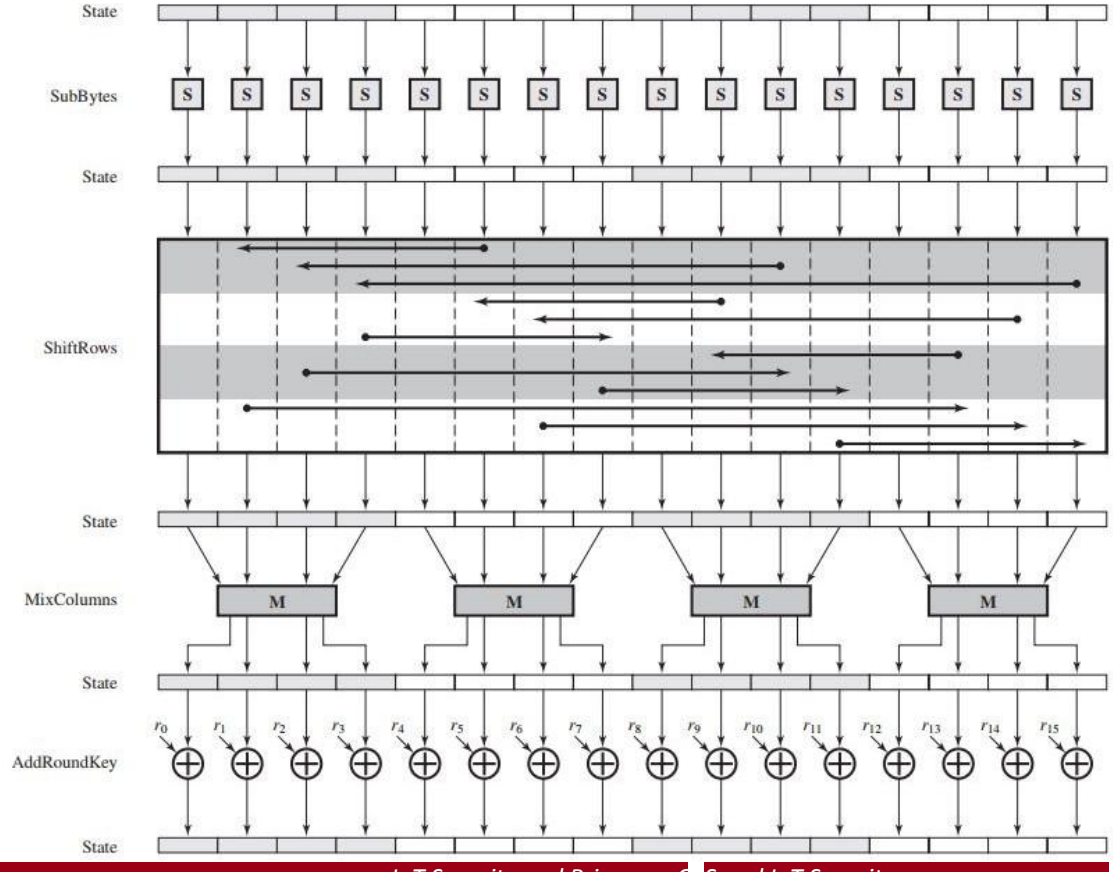
- For different stages are used: one permutation and three substitutions
- **Substitute bytes:** uses an S-Box to perform a byte-by-byte substitution of the block
- **Shift rows:** simple permutation performed row by row
- **Mix columns:** substitution that alters each byte in a column as a function of all of the bytes in the column
- **Add round key:** bitwise XOR of the current block with a portion of the expanded key

Advanced Encryption Standard

- An S-box is essentially a lookup table that maps input bit patterns to output bit patterns according to a fixed substitution rule. Each possible input value corresponds to a unique output value.

Input	Output
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	0111
0111	1001

Advanced Encryption Standard



Advanced Encryption Standard

- For both enc and dec, the cipher begins with an Add Round Key stage, followed by nine rounds that each includes all four stages
- This is followed by a tenth round of three stages
- Only the Add Round Key makes use of the key, hence the cipher begins and ends with an Add Round Key stage
- Any other stage applied at the beginning or end is reversible without knowledge of the key

Advanced Encryption Standard

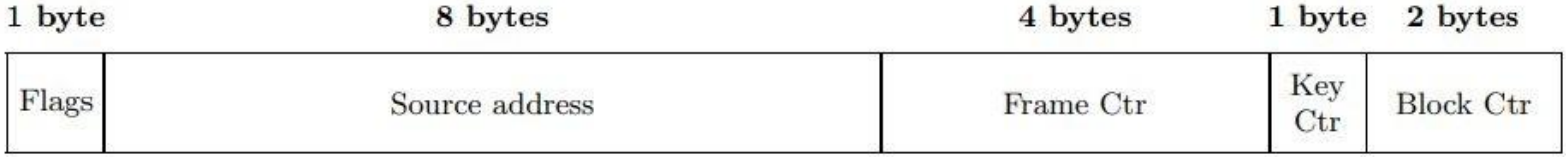
SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

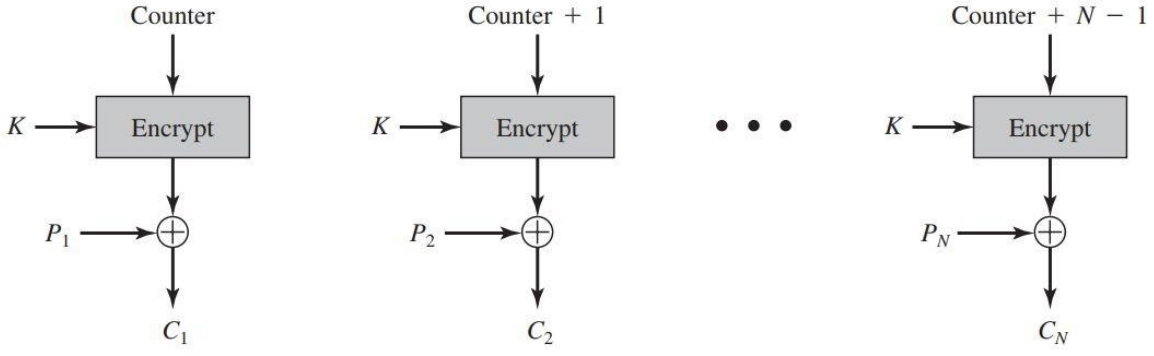
- The Add Round Key itself is not formidable
- The other three stages together scramble the bits, but by themselves would provide no security because they do not use a key
- Overall, the scheme is a sequence of XOR and scrambling operations
- Thus it is both highly efficient and secure
- Each stage is easily reversible: for the substitute byte, shift row, and mix columns stages an inverse function is used in the decryption algorithm
- For the add round key, the inverse is achieved by XORing the same round key to the block

AES-CTR

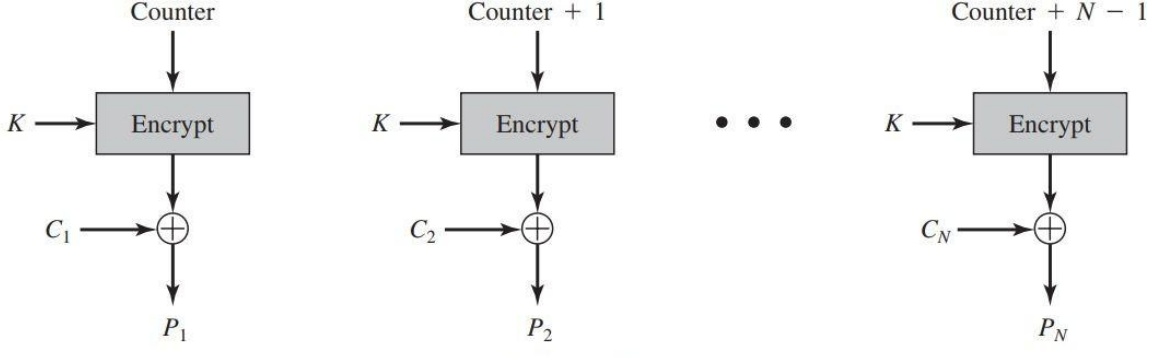
- confidentiality protection using AES block cypher with counter mode
- The sender breaks the cleartext packet into 16 byte blocks $p_1, \ldots, p_n$ and computes $c_i = p_i \oplus E_k(x_i)$ where each block uses its own counter x
- The receiver recovers the plaintext as $p_i = c_i \oplus E_k(x_i)$
- The counter, known as nonce or IV, is composed of a static flags filed, sender's address, and three separate counters



AES-CTR



(a) Encryption



(b) Decryption

AES-CTR

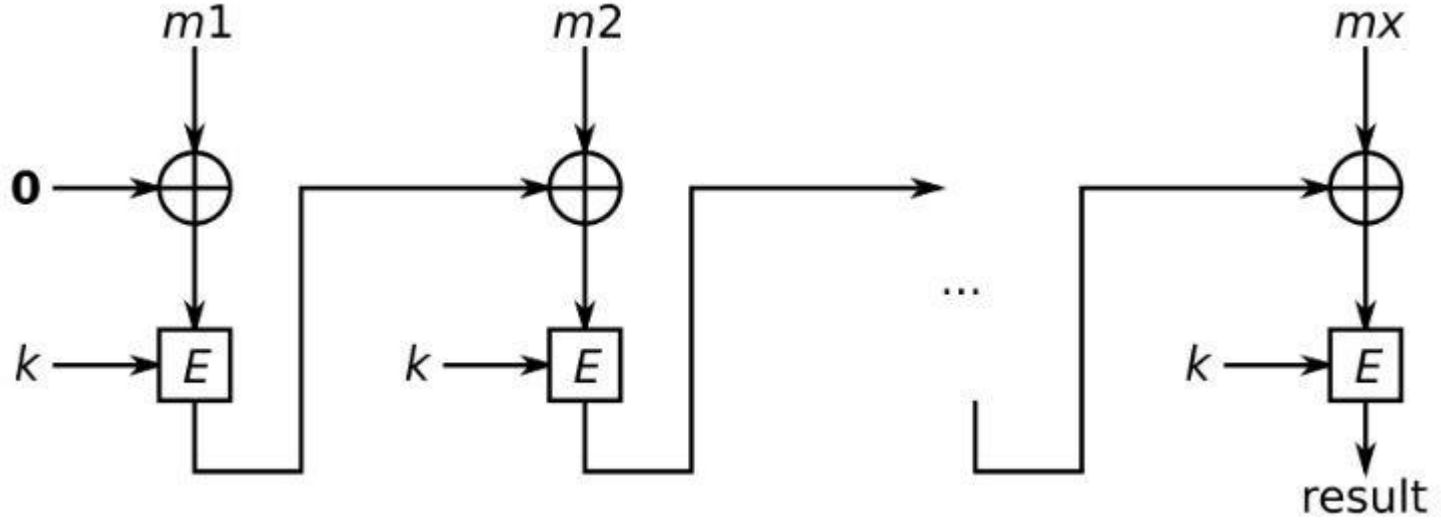
- The frame counter is maintained by the hardware radio and the sender increments it after encrypting each packet
- The key counter is under application control
- Requirement: nonce must never repeat within the lifetime of any single key and frame and key counter should prevent nonce reuse
- The 2 bytes block counter ensures that each block will use a different nonce value

Advantages

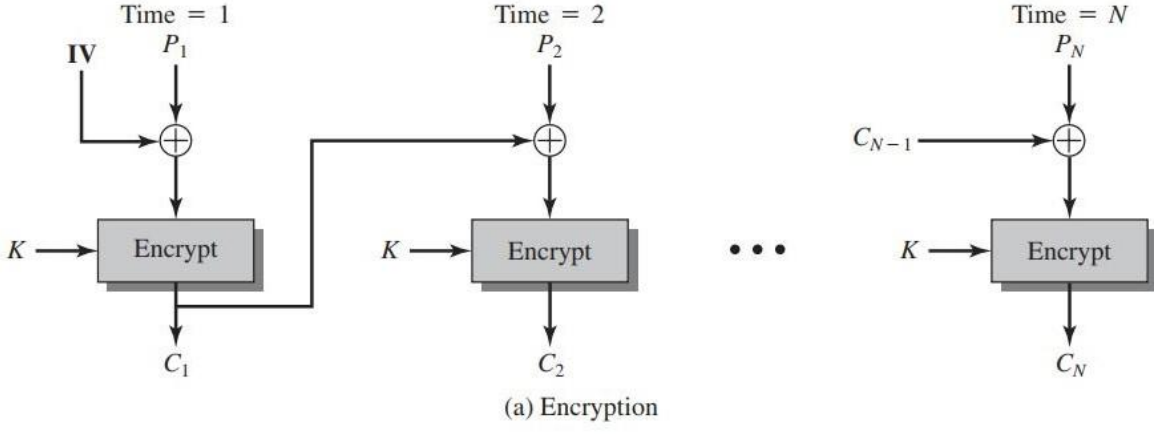
- Hardware efficiency: it uses no chaining, so can be done in parallel over blocks
- Software efficiency: for the same reason, processors supporting parallelism can be effectively utilized
- Random access: possible to decrypt just a single block

AES-CBC-MAC

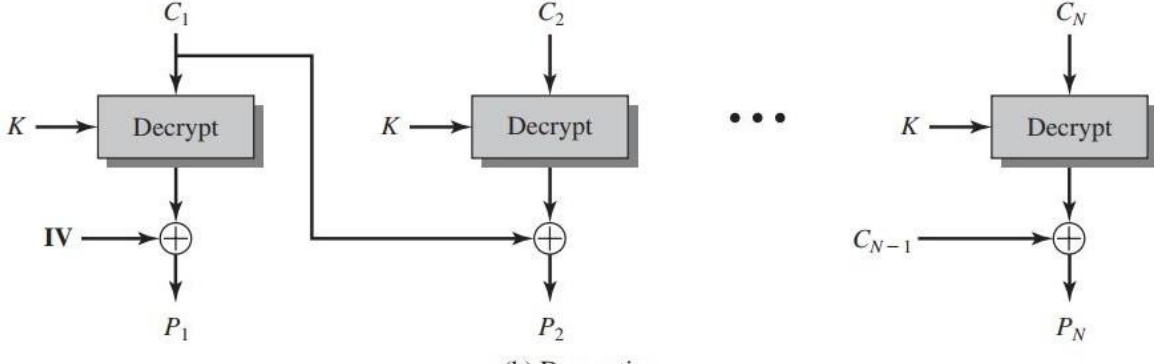
- Provide integrity protection via CBC-MAC algorithm
- It can only be computed by parts having symmetric key
- MAC protects the packet headers and data payload



AES-CBC



(a) Encryption



(b) Decryption

AES-CBC

- To produce the first block of ciphertext, an IV is XORed with the first block of plaintext
- On description, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext

- The IV should be protected as the key
- An adversary fooling the receiver into using a different IV may trigger selective bit flips

- For decryption, each cipher block is passed through the decryption algorithm
- The result is XORed with the preceding ciphertext block to produce the plaintext block
- Given $C_j = \mathrm{E}(K, [C_{j-1} \oplus P_j])$

  We have

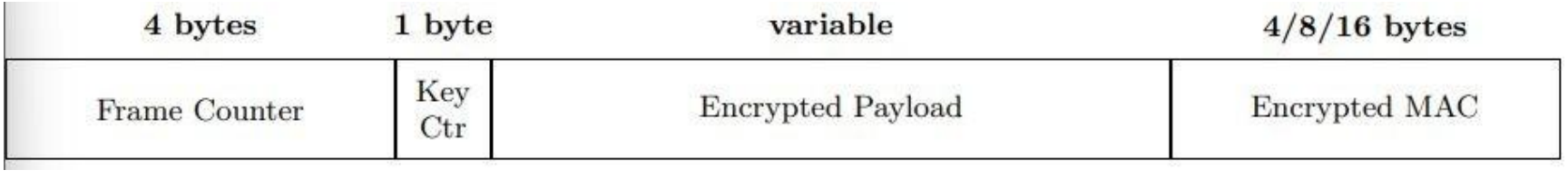
$$\mathrm{D}(K, C_j) = \mathrm{D}(K, \mathrm{E}(K, [C_{j-i} \oplus P_j]))$$

$$\mathrm{D}(K, C_j) = C_{j-1} \oplus P_j$$

$$C_{j-1} \oplus \mathrm{D}(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

AES-CCM

- Provides both encryption and authentication
- It first applies integrity protection over header and data payload using CBC-MAC
- Encrypts data payload and MAC using AES-CTR mode



Keying models

- Govern what key a node uses to communicate with another node
- Network shared keying: single network-wide shared key. Key management becomes trivial and memory requirements are minimal
- However, vulnerable to insider attacks and single key compromise
- A single compromised node can undermine the security guarantees of the entire network
- If we expect nodes to be occasionally compromised or captured, not a good approach

Keying models

- Pairwise keying: limit the scope of every key
- Each pair of nodes shares a different key
- Thus, if a node is compromised, only the security of communication with its pair is undermined
- Comes with an increased overhead
- Each node must store a key for every other node it communicates to
- Select the proper key when communicating with a node
- IoT nodes have limited resources

Keying models

- Group keying: compromise between pairwise and network keying
- A single key is shared among a set of nodes and is used on all links between any two nodes in that group
- Groups can be created based on locations, network topology, or similarity of function
- Partial resistance to node compromise and partial improved management of resources

Secure Group Pairing

- IoT devices need a pairing mechanism which establishes shared cryptographic keys between devices
- Traditional pairing methods employ a centralized approach where a user pairs each device with a trusted IoT gateway through external helper device (e.g., type a pwd on the smartphone)
- However the central gateway is a single point of failure, sometimes preventing devices from communicating → move towards decentralized networking (e.g., OpenThread)

Context-Based Pairing

- In order to limit as much as possible the human intervention, there is an increasing interest in context-based pairing
- Co-located sensors establish shared keys based on the entropy extracted when they observe common events
- Although nice, it is limited to homogeneous devices which need to have the same sensing modalities
- Furthermore, if based on events, it might take some hours or days to have pairing

Zigbee

- Zigbee is a higher layer protocol based on IEEE 802.15.4 to create PAN networks
- It is usually leveraged for home automation, medical device data collection, and small scale projects
- It has a range of 10-100 m in line of sight
- Longer distances are achieved via multi-hop in a mesh network of intermediate devices

Zigbee Devices

Three type of devices in Zigbee:

- Zigbee Coordinator (ZC): root of the network tree and bridge with other networks. Only one ZC, since it is the originator of the network. Trusted node containing e.g., keys
- Zigbee Router (ZR): act as intermediate device to pass data to other devices. They are usually mains powered to always be available
- Zigbee End Device (ZED): minimal functionalities to talk to the parent node. Battery powered and wake up only when has something to say

Zigbee Security

- Zigbee security builds on top of IEEE 802.15.4 security
- Keys and modes we've seen for 802.15.4 are basic for Zigbee
- A momentary exception exists for the addition of a previously unpaired and unconfigured device
- We need to assume trust in the initial installation of keys
- Within the protocol stack, we need access policies to cope with the lack of cryptographic separation between different layers

Security Architecture

- Zigbee uses 128-bit keys to implement its security mechanism
- A key can be associated to a network or to a link, acquired via pre-installation, agreement, or transport
- There should be an initial master key obtained via a secure medium
- Establishment of link keys is based on a master key
- Trust center: special device in the network which other services trust for the distribution of secure keys
- Ideally, all devices will have the trust center address and initial master key

- The security architecture is distributed to different layers

  MAC layer

Layer	Capabilities
MAC	• Single hop reliable communications • Security level specified by upper layers
Network	• Outgoing frames use the appropriate link key according to routing
Application	• Key established and transport services to both ZDO and applications

Device Authentication

- After joining the network, an end-device needs to exchange security information with the trust center
- Needs to obtain the current network key from the trust center and establish a new end-to-end trust center link key
- It consists of four steps

Device Authentication

- Establish the Trust Center Link Key (TCLK): each device has a pre-installed TCLK typically obtained from the device installation code
- This key is provided to the TC through out-of-band means
- Establish the transport key: the TC and node can derive a transport key from the TCLK
- Distribute the network key: the TC can send to the new node the network key encrypted via transport key
- Establish new link key: as soon as the join procedure is completed, the TC updates the TCLK of the joining device

LoRAWAN

- Personal Area Networks sometimes are not sufficient for IoT purposes
- Long Range (LoRa) is a proprietary radio communication technique
- LoRa Wide Area Network (LoRaWAN) defines the communication protocols and system architecture to create a larger network than PAN
- Also in this case, we consider battery powered resource constrained devices
- It is a cloud based Medium Access Control (MAC) layer protocol
- Manages communications between LPWAN gateways and end-node devices

LoRAWAN Security

- The LoRa alliance designed security measures for LoRaWAN accounting for low power consumption, low complexity, low cost, and high scalability
- As part of the network join procedure, a LoRaWAN end-device establishes a mutual authentication with the LoRaWAN network
- MAC and application messaging are origin authenticated, integrity and replay protected, and encrypted
- End-to-end encryption for application payloads

LoRAWAN Security

- LoRaWAN uses AES, and each device has a unique 128 bit AES key and a globally unique identifier (EUI-64-based DevEUI)
- Allocation of EUI-64 identifiers require the assignor to have an Organizationally Unique Identifier from IEEE registration authority
- LoRaWAN networks are identified by a 24-bit globally unique identifier assigned by the LoRa Alliance

Mutual Authentication

- The Over-the-air activation (or join procedure) test whether both devices know the AppKey
- The proof is obtained by computing an AES-CMAC(AppKey) on the device's join request and by the backend receiver
- CMAC is a One-Key MAC that fixes security deficiencies of CBC-MAC, i.e., the fact that the latter is secure only for fixed-length messages
- Nevertheless, a variation of CBC-MAC

Mutual Authentication

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- Two keys are derived by LoRaWAN authentication:
  - Providing integrity protection and encryption of the LoRaWAN MAC commands (NwkSKey)
  - E2E encryption of application payloads (AppSKey)
- NwkSKey is distributed to the LoRaWAN network to prove and verify packet integrity and authenticity
- AppSKey is distributed to the application server to encrypt/decrypt the application payload

IoT Network Formation

- Network Formation refers to the process that nodes in wireless network use to join a network
- Again focusing on IEEE 802.15.4, we focus on the *e* version and the time slotted channel hopping
- 6TiSCH refers to the use of IPv6 addressing over the time slotted channel hopping medium access control methodology

Time Slotted Channel Hopping

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- TSCH divides time into small and fixed duration time slots that repeats over time
- A single timeslot is long enough to transmit a packet (e.g., 10 ms for max 127 bytes long) and receive its ACK (if needed)
- Several timeslots collectively form a slotframe
- A node might be in a receiving, transmitting or idle state in a timeslot
- Nodes use a scheduling function to decide their communication timeslot, resulting in *deterministic* channel access

Time Slotted Channel Hopping

- Sometimes, a single slot may be shared by multiple users
- However in this case, we use TSCH-CSMA/CA
- To deal with interference and multi-path fading, IEEE 802.15.4e leverages the availability of multiple channels for communications
- The mechanism of changing the physical communication channel after each timeslot is called channel hopping
- Nodes need to schedule one transmission channel along with the timeslot

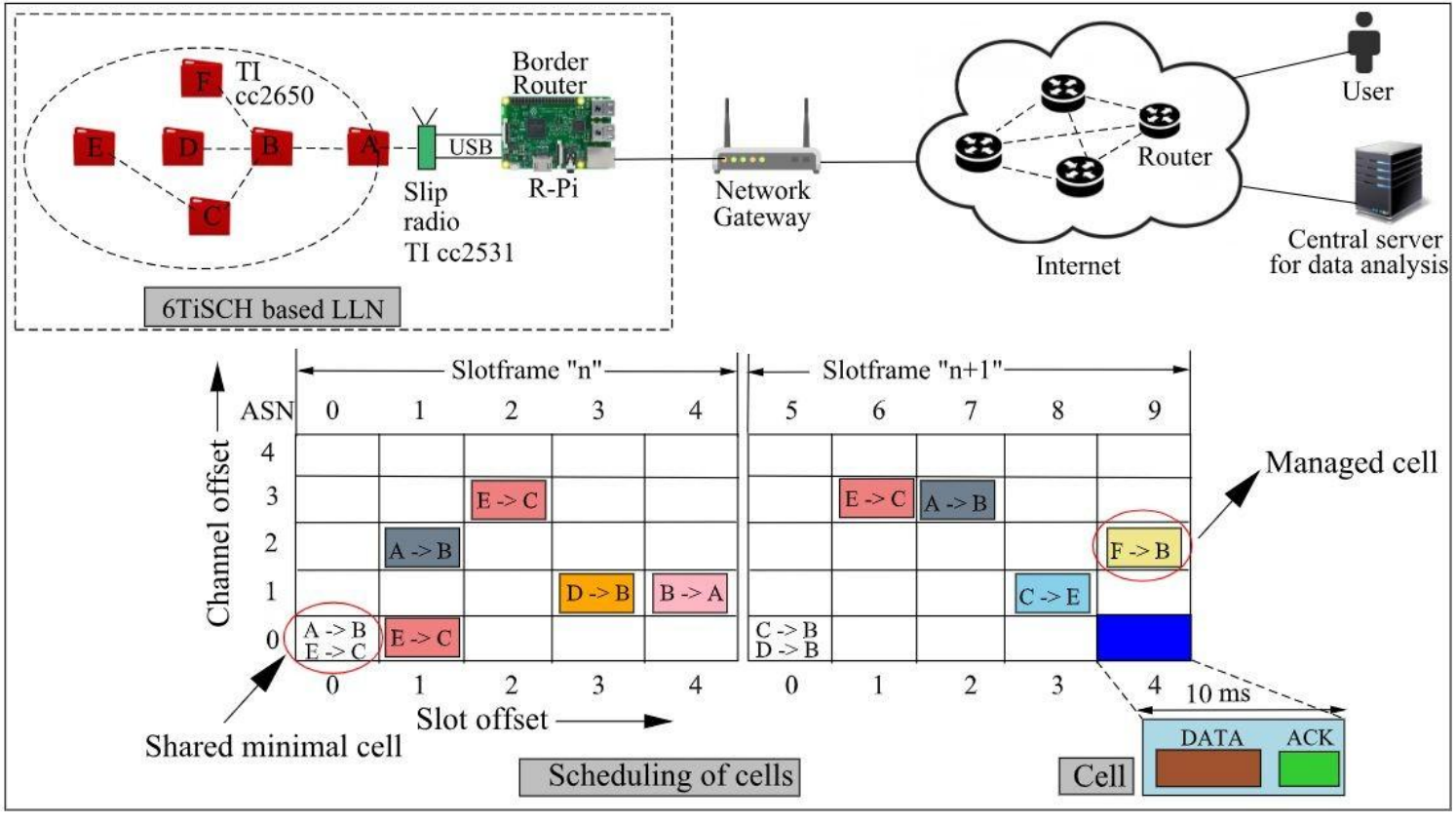
Time Slotted Channel Hopping

- The nodes deterministically compute their physical channel using

$$f = F[(ASN + \text{channel offset}) \; mod \; N_{channels}],$$

- Where F denotes a lookup table for channels, ASN is the absolute slot number (total number of timeslots elapsed from starting of the network), channel offset is a fixed integer assigned by the scheduling algorithm, and $N_{channels}$ denotes the number of channels used in the network

Time Slotted Channel Hopping



6TiSCH Network Formation

- We consider the formation of a 6TiSCH network
- There exists a root node, called Joint Registrar/Coordinator (JRC) which periodically broadcasts Enhanced Beacon (EB) frames
- EBs contain basic network information such as the JRC ID, duration of a timeslot, number of time slots in a slot frame, channel hopping sequence, location of the shared cell
- Pledges are new nodes willing to join a 6TiSCH network

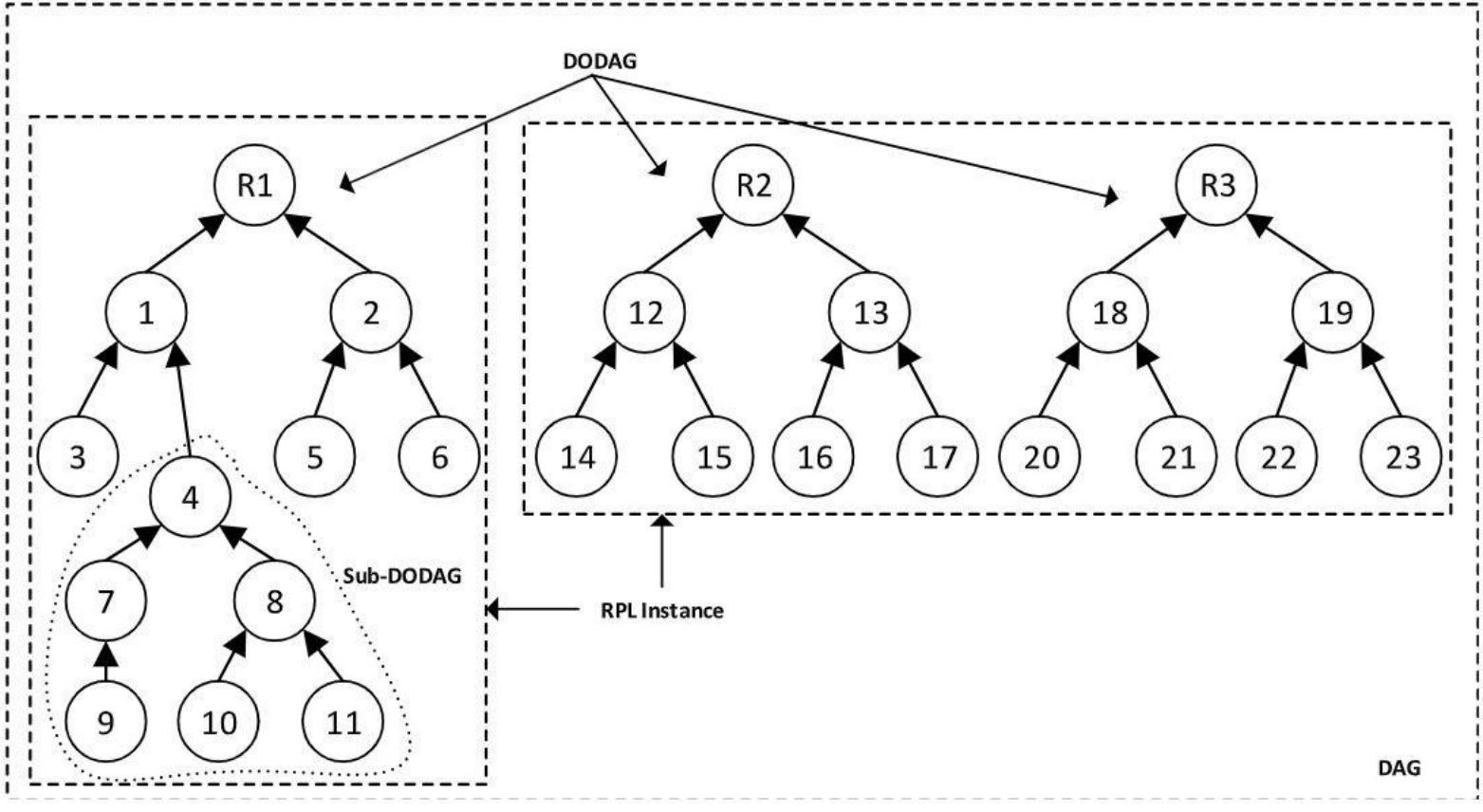
IPv6 Routing Protocol

- IPv6 Routing Protocol (RPL) was developed by the IETF Routing Over Low Power and Lossy Networks (ROLL) working group
- It aims at providing routing services in low power lossy networks
- Distance vector routing protocol that organizes network devices into Directed Acyclic Graphs (DAGs)
- Distance vector = measure the distance in terms of number of hops

IPv6 Routing Protocol

- A DAG represents a network where all nodes are connected in a way such that there are no round-trip paths and traffic is routed to reach one or more *root* nodes
- In DAG we can find one or more Destination Oriented DAG (DODAG), where the topology has one root node only
- To enable several applications to work simultaneously, but independently, inside the network several RPL instances can co-exist inside a DAG

IPv6 Routing Protocol



IPv6 Routing Protocol

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- Nodes in RPL use *Objective Functions* to define essential configurations such as routing metrics, optimization objectives, calculate the rank, select parents in the DODAG
- Nodes are also assigned a *node rank*, i.e., an unsigned integer value that defines the node's relative position to the DODAG's root node
- The value depends in the objective function and is strictly increasing going down the DODAG
- Rank is used by nodes to decide the set of parents to choose

RPL Control Messages

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- There are five types of control messages in RPL
- *DODAG Information Object (DIO):* advertised by each node when it wants to join a DODAG, create one, or maintain one
- It contains information that is used to identify RPLs instances (RPLInstanceID), DODAG ID, DODAG version number, RPL mode of operation, rank of the sending node, DODAG configuration
- Usually, DIO messages are multicast by nodes when they receive DIO messages from other nodes (except for the root node)

RPL Control Messages

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- There are five types of control messages in RPL
- *DODAG Information Solicitation (DIS):* used by nodes when they want to join a DODAG and they did not receive any DIO message for a period of time (neighbor probing)

- The response depends on how it was sent (if unicast, or multicast)
- The use of DIS messages can however introduce a vulnerability that can be exploited by an attacker…

RPL Control Messages

- There are five types of control messages in RPL
- *Destination Advertisement Object (DAO):* while DIS messages are used to create and maintain upward routes (towards the root), DAO are used to find downward routes
- Contains path information for reachable nodes by its sender, used to create routing tables at receiving nodes

RPL Control Messages

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- There are five types of control messages in RPL
- *Consistency Check (CC):* used by RPL to synchronize security counters/timestamps between each pair of nodes and provide a challenge response mechanism as a basis for control messages replay attack protection
- Always sent as secure messages and hence only available in RPL secure mode

Formation and Maintenance

- As previously discussed, DODAG formation is done via DIO messages and always starts from the root node
- The root node should set most of the DIO base fields: RPL instance ID; DODAG ID; DODAG version; RPL mode of operation
- Upon receiving DIO messages, each node should i) calculate its own rank, ii) decide on which DODAG to join, iii) select at least one preferred parent from a set, iv) multicast is own DIO message (changing the rank)

Formation and Maintenance

- To maintain the topology, DIO messages should be exchanged on a regular basis and nodes can discard them if they do not result in any change to the current configuration

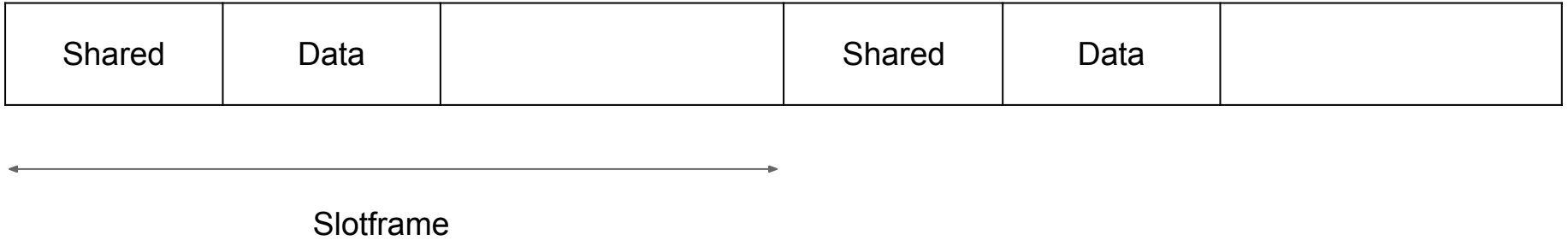
- RPL uses the *trickle* algorithm to control when to send DIO messages
- Each node maintains a DIO counter (with a threshold) and a trickle timer
- DIO messages are sent when the trickle timer reaches its time or upon reception of a DIO message that causes a configuration change

Formation and Maintenance

- Whenever a DIO message is received and discarded, the DIO message counter is increased
- If the counter reaches a threshold vale, both the counter and the trickle timer are reset and the trickle time is doubled
- DIO counter and trickle time will return back to their original setting when a change is made due to a received DIO message
- This procedure allows for a fewer broadcast of DIO messages when the network is stabilized

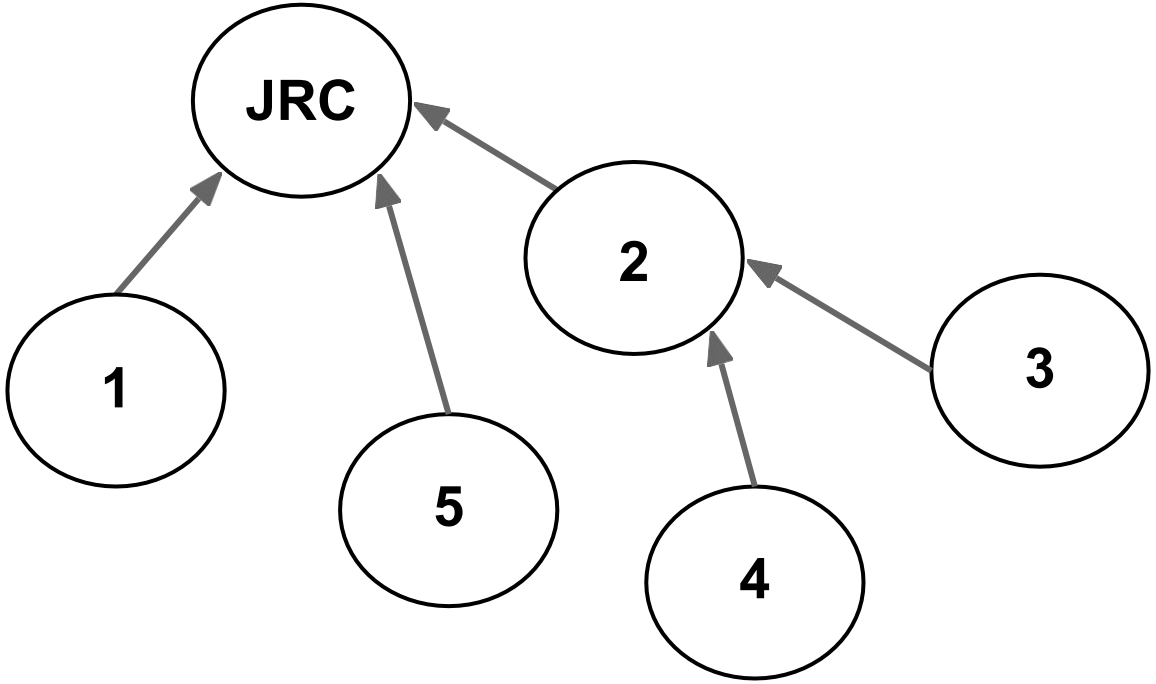
IoT Network Formation

- When pledges want to join the network, they start scanning until they receive a valid EB
- When it receives an EB from an already joined node, it becomes a TSCH synchronized node
- The channel is slotted, and is divided into control slots and shared slots

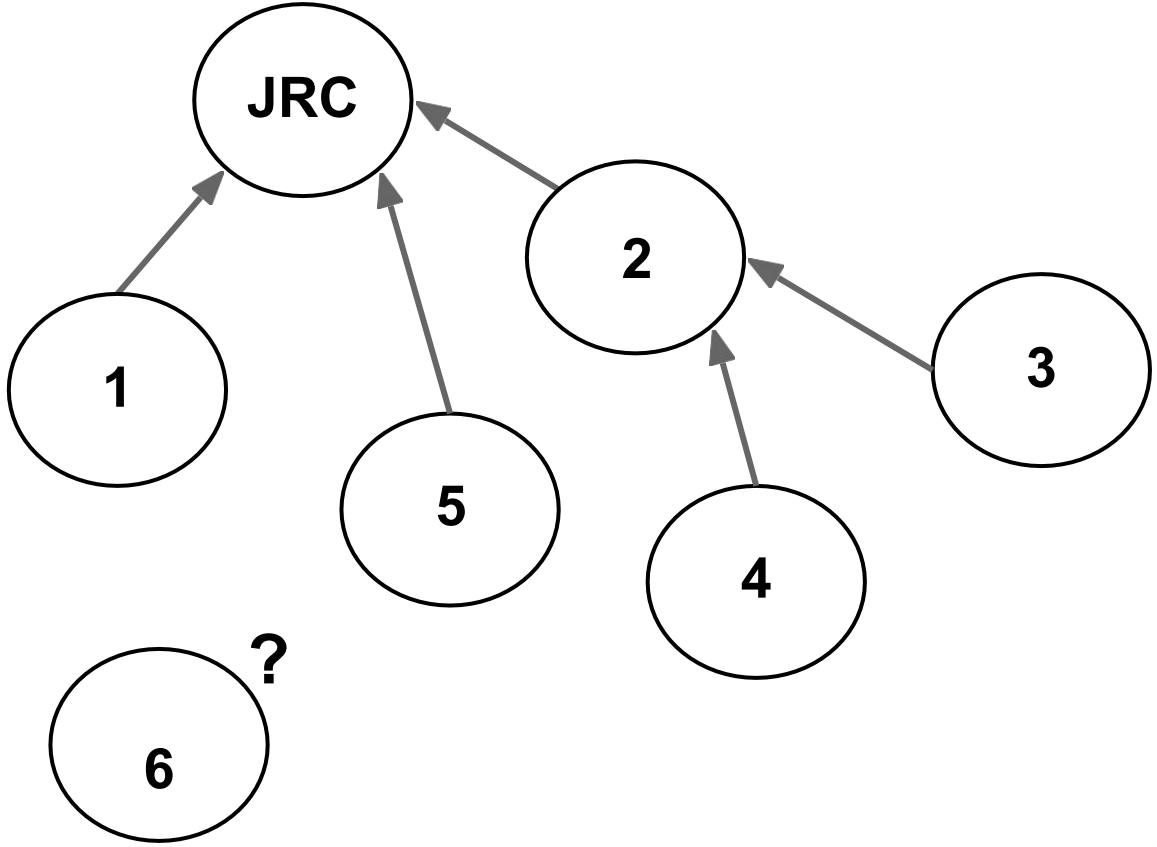


IoT Network Formation

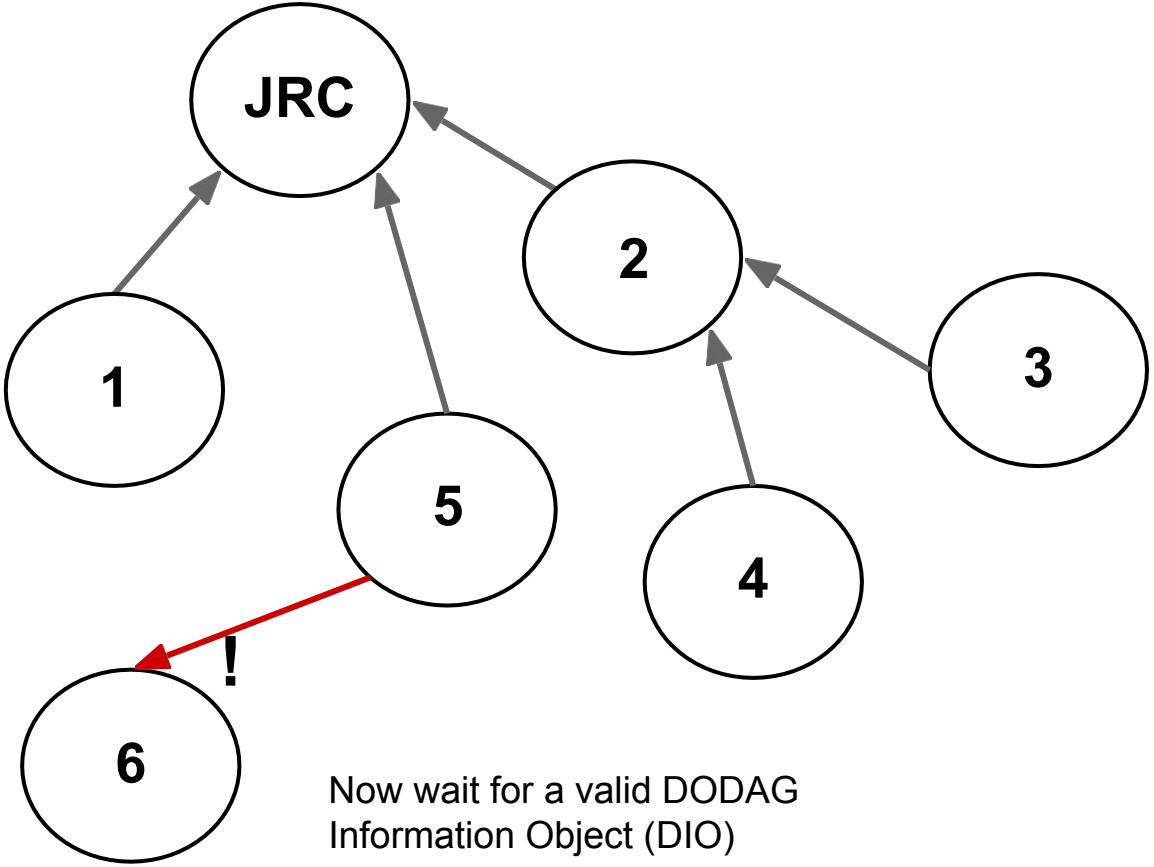
- The network is organized as a Destination Oriented Directed Acyclic Graph (DODAG)



When to Join?

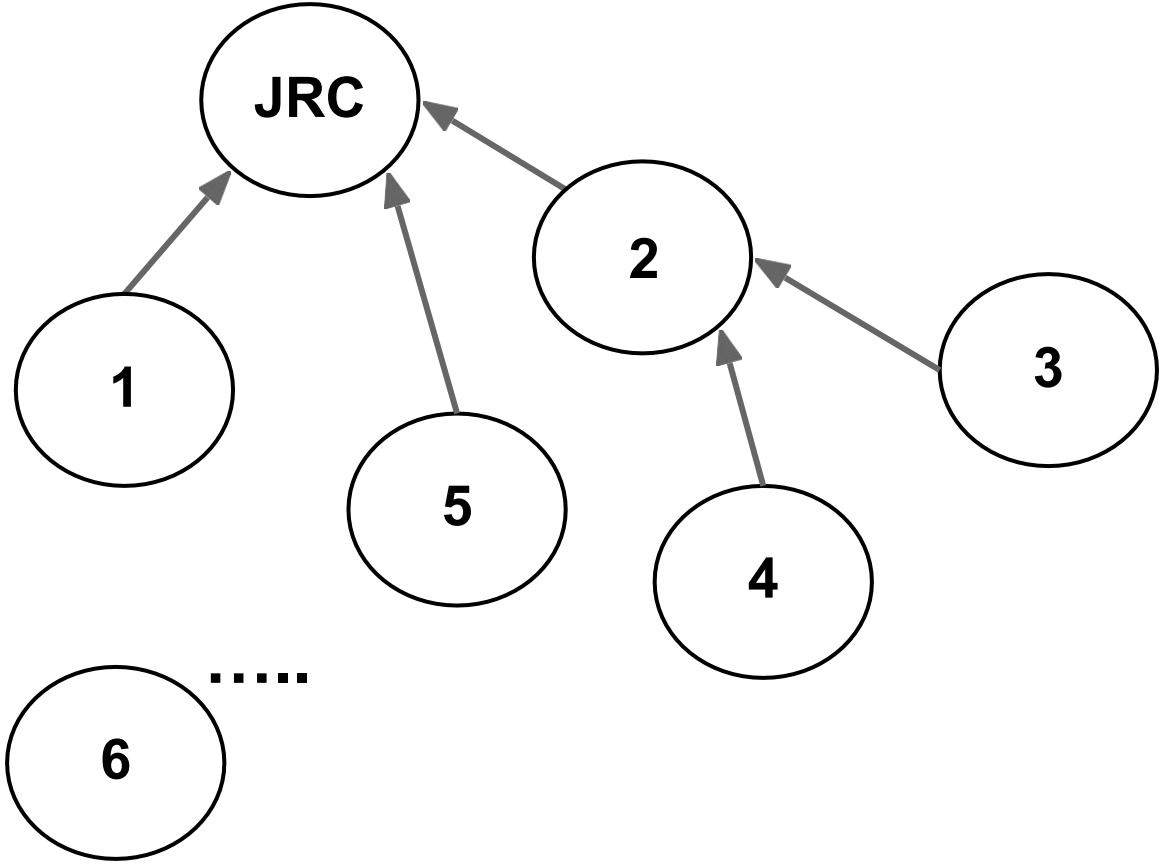


When to Join?

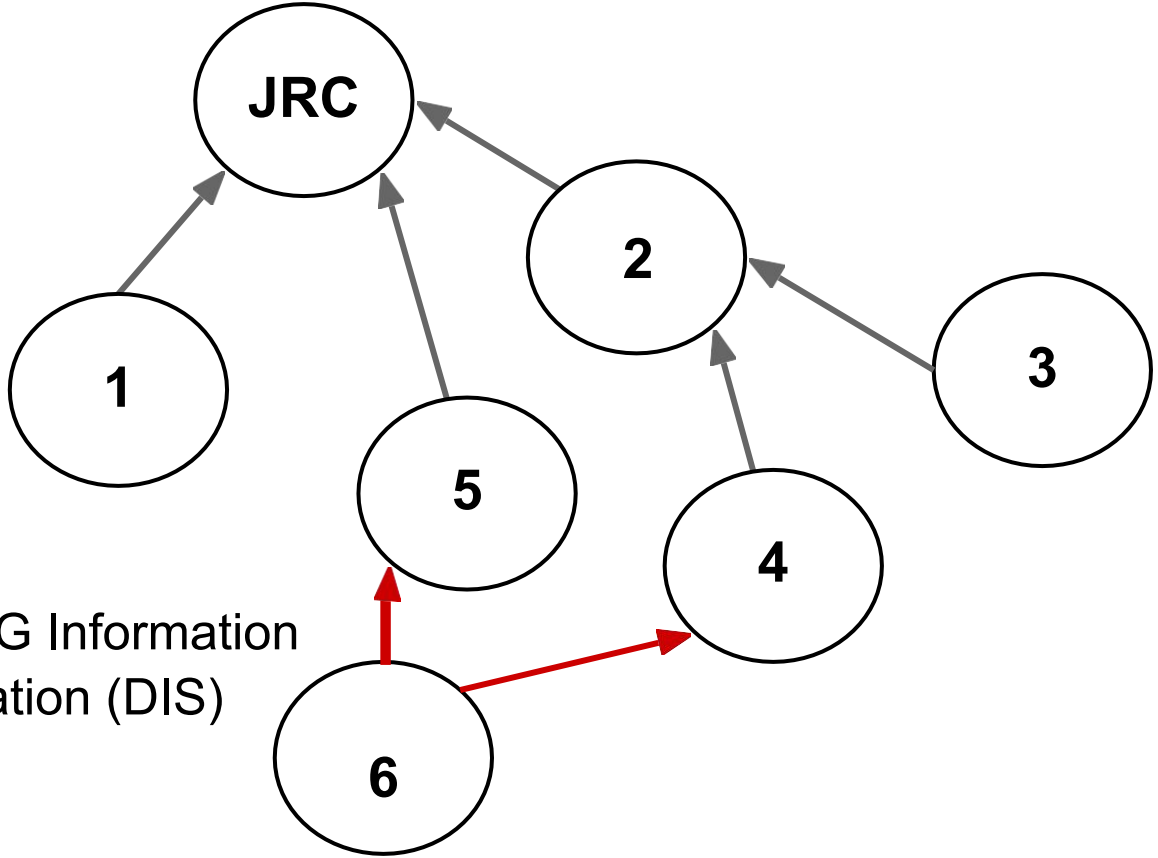


Now wait for a valid DODAG Information Object (DIO)

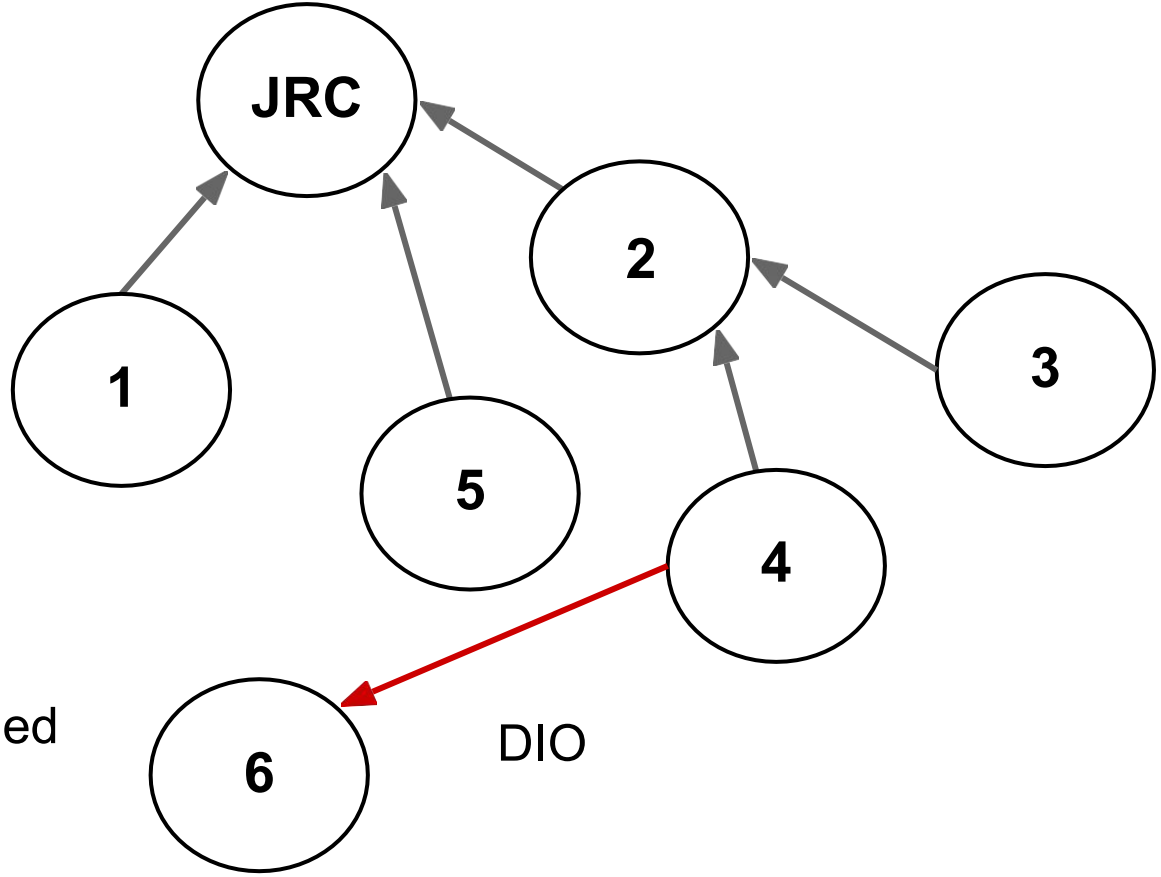
When to Join?



When to Join?



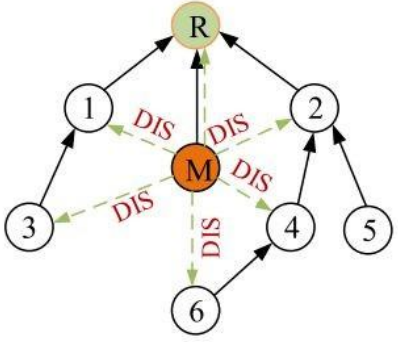
When to Join?

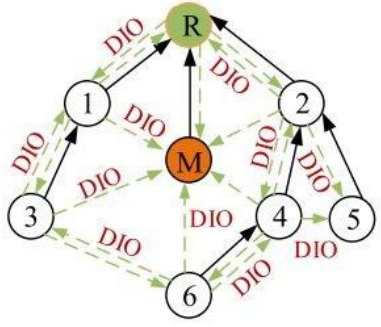


DIS Attack

- DIS packets can be sent by arbitrary nodes to solicit the sending of DIO packets
- DIS attack: increase the number of transmissions in DIO packets in the network
- Goal: increase energy consumption and congest the shared slot

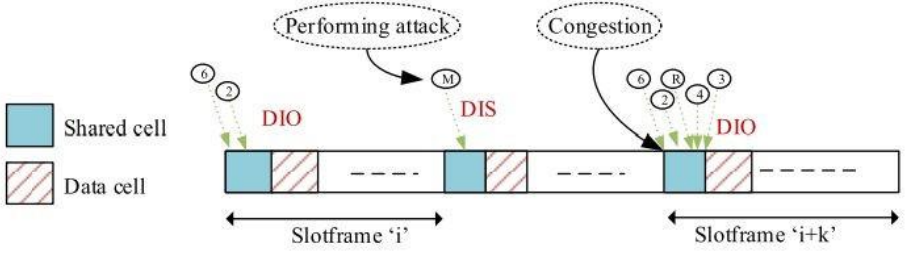
DIS Attack





(a) A malicious node transmits its DIS packet.

(b) Legitimate joined nodes transmit their DIO packet in response.



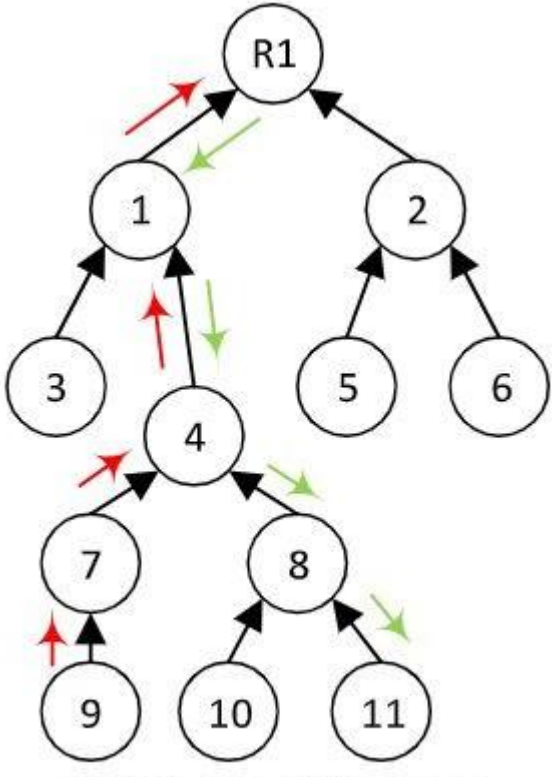
(c) Effect of DIS attack on shared cell's congestion.

RPL Modes of Operation

- RPL supports three types if communications in 6LoWPAN: multipoint-to-point (resembling upward traffic), point to multipoint (resembling downward), and point to point (resembling traffic between two non-root nodes)
- P2P has to be done using one of three ways: non-storing mode, storing mode, or one-hop P2P

Non-Storing Mode

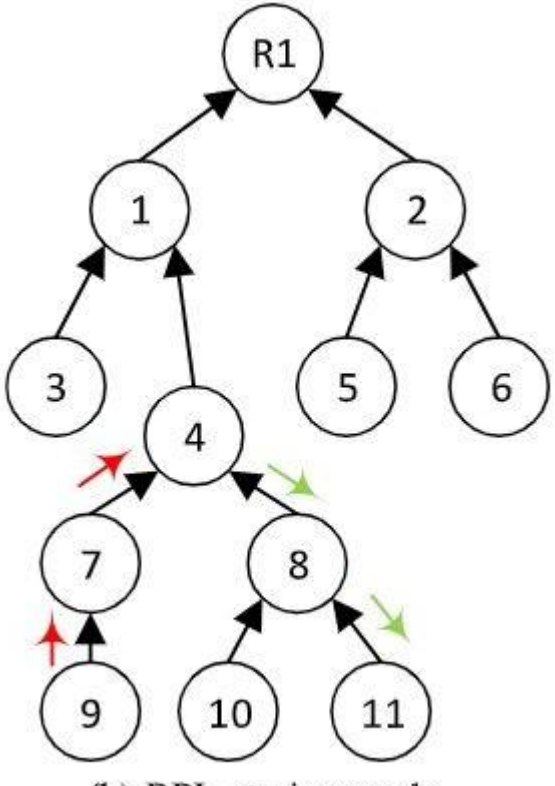
- In *non-storing* mode, traffic goes all the way up to the root node which will be responsible for sending it towards its destination using source routing
- Nodes send DAO messages towards their DODAG root node, responsible of routing tables (based on received DAO information)



(a) RPL non-storing mode

Storing Mode

- In *storing* mode, each node keeps a downward routing table for its sub-DODAG and use it to forward P2P traffic (up to a common ancestor)
- Each node sends DAO messages to its DAO parent
- Receiving node should use DAO to create and maintain downard routing tables



(b) RPL storing mode

RPL Security Features

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- RPL has several *optional* security features
- Designed to use link-layer security mechanism when they are available to secure message transmission
- Besides link layer, RPL has three security modes: i) unsecured, ii) preinstalled, iii) authenticated

- *Unsecured:* default mode which depends on link layer security
- *Preinstalled:* preinstalled symmetric keys are manually configured on the nodes which uses these keys to handle and generate the secure version of RPL control messages

RPL Security Features

- *Authenticated Mode:* nodes that want to join as leaves need to use pre-installed keys to join, but nodes with routing capabilities must use preinstalled keys to obtain another key from authentication authority

- How to do that is left to implementation
- RPL has also optional *replay protection*, called Consistency Check (CC). These checks use a non-repetitive value and the stored status information to check if the received CCM nonce value has been used before from the original node

RPL Security Features

- In the second and third mode, to support confidentiality, integrity and authenticity, nodes used AES/CCM with 128-bit keys to generate 32-bit and 64-bit MACs
- Also, RPL uses RSA with SHA-256 for digital signatures of the messages to provide confidentiality and authenticity with optional 2048 and 3072-bit signatures

Rank Attack

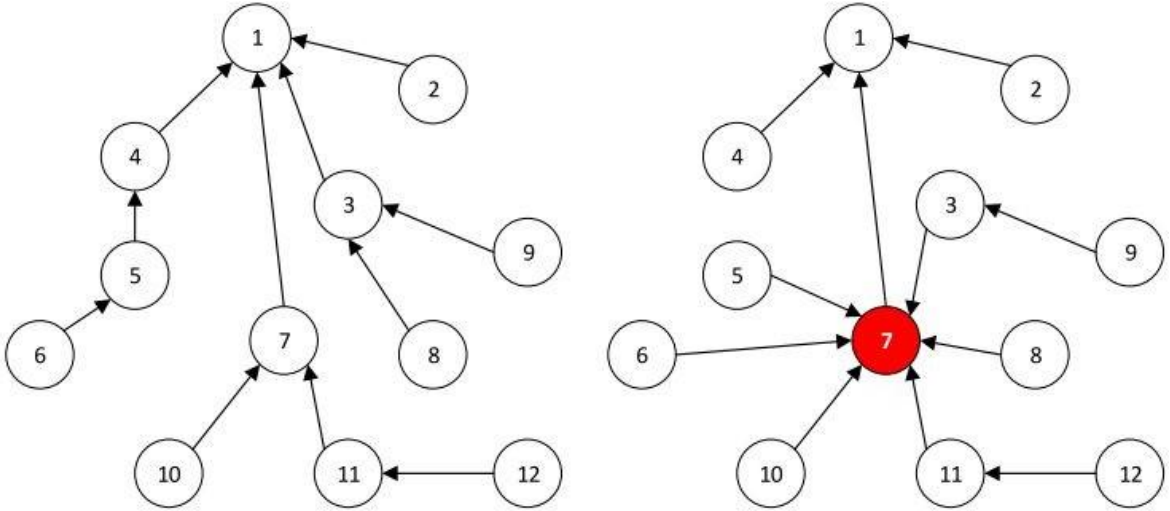
- Each node chooses its parent based on two values: the rank and objective function
- The rank should increase going downward in the DODAG, and the role of the preferred parent selection is to select the one with the best rank
- The objective of the attacker is to manipulate these values to affect the network topology

Rank Attack

- Manipulation can be performed in two ways
- First, the attacker changes its rank by a specific value based on its neighbors rank value
- Second, the adversary manipulates its rank through the use of a different objective function to deceive legitimate nodes into giving the malicious node a better rank

Types of Rank Attack

- Decreased rank attack: malicious nodes advertise lower rank to other nodes resulting in many of them selecting the adversary as preferred



(a) Normal topology (b) Decreased rank attack

Types of Rank Attack

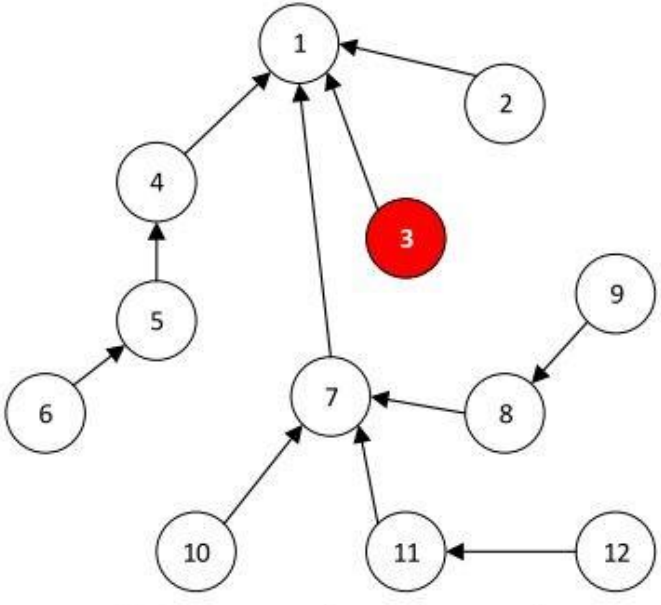
- Increased rank attack: the attacker is near the routing node and advertises higher rank

  and worse routing metrics

- The idea is to cause topology disruptions

  and delays, as nodes will need to select

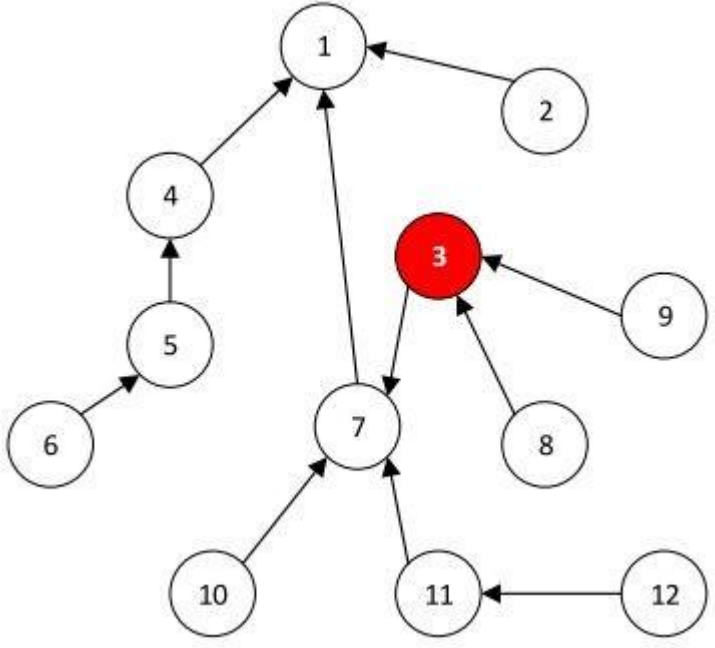
  further nodes as parents



(c) Increased rank attack

Types of Rank Attack

- Worst parent attack: the attacker advertises its true rank but selects the worst parent for itself
- Deceive nodes into connecting to the attacker and cause delays due to the worst path they unwillingly select



Neighboring Attack

- In this attack, the attacker node will forward any received DIO message it gets to its neighboring nodes (no modification)
- This creates the illusion that the original sender is in the range of the neighboring nodes
- Worst case scenario: the original sender has a good rank and adversary's neighbors choose it as preferred parent although being out of range

Neighboring Attack

- Alone, the neighboring attack only causes a slight increase in the end-to-end delay
- However, suitably combined with other attacks gets more dangerous
- An adversary could launch a DIS attack to get DIO messages with better metrics, then selecting one of these messages to perform a neighbor attack, increasing the effect of such an attack

RPL Storing Mode

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- RPL can work in a Point-to-Point fashion, i.e., create traffic between two nodes that are not root nodes of the DODAG
- In storing mode, each node keeps a downward routing table for its sub-DODAG and use it to forwards P2P traffic
- In practice, traffic goes upward up to a common ancestor of sender and destination that routes the packet to the destination node

RPL Storing Mode Attack

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- Routing table overload: the adversary sends many bogus routes (via DAO)

  until the node saturates
- Route table falsification: a malicious node advertises fake routes to

  other nodes that might exists but not be part of the attacker's

  sub-DODAG causing packet losses or longer delays
- All these attacks also cause resource exhaustion due to the

  increased overhead and repetitive repair attempts

Attacks Inherited from WSN

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- Wireless Sensor Networks (WSNs) are the networks from which IoT was born
- Therefore, IoT inherited part of the routing attacks that existed in WSNs
- Although the working principle is the same, attacks needed to adapt to the new IoT paradigm

Blackhole and Selective Forward

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- In a blackhole attack, a malicious node(s) will drop all packets it receives instead of forwarding them (DoS)
- To be less detectable, an attacker may decide to selectively drop packets (i.e., only forward RPL control messages) → selective forward or greyhole attacks
- Selective-forward attacks cannot be detected nor mitigated by the self-healing mechanisms of RPL because they pass control messages

Sinkhole attacks

- Malicious node(s) try to be sink for as much nodes as possible by advertising a fabricated link with better metrics
- Sinkhole by themselves are not very powerful, they need to be combined with other attacks
- These attacks can be performed by advertising DIOs with better metrics or having several adversaries directing all passing traffic toward another adversary

Wormhole attacks

- To create this attack, two adversaries need to cooperate to create a tunnel between them and transmit traffic through it instead of the regular path
- Three ways to create a wormhole:
  - Packet encapsulation: malicious nodes use a legitimate path between them and encapsulate packets to hide the hop count
  - Relay: deceive nodes to be neighbors
  - Out-of-band link: create links that are not part of the network

Clone ID and Sybil attacks

SPRITZ
SECURITY & PRIVACY
RESEARCH GROUP

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- In Clone ID attack, a malicious node(s) takes the identity of another legitimate node
- In Sybil attacks, each malicious node takes several identities from legitimate nodes
- With sybil attacks an attacker can submit forged information to manipulate the system, disturb the routing topology and reputation-based systems
- Can be mitigated by adding location information and DHTs

Mitigation Techniques

- To detect some of these attacks (or their declinations) there have been many proposals in terms of Intrusion Detection Systems (IDSs)
- Signature-based IDSs: use a database of signature patterns of the attacks
- Anomaly-based IDSs: create a normal behavior profile and compare the current observations with the normal behavior
- Specification-based: create a normal profile based on protocol specification
- Hybrid IDSs: combine two of the aforementioned methods

Placement of IDSs

- Centralized IDS: the IDS resides either on the root node or on a dedicated host and uses the traffic passing by to detect attacks
- In many cases, it is required that the central node of the IDS send periodic request for updates to unmonitored areas
- Advantage: most of the heavy works occurs inside a powerful node, usually capable of performing firewall functionalities as well
- On the other hand, challenging to monitor the network during the attack

Placement of IDSs

- Distributed IDS: each node will have a full IDS implementation, making it responsible for detecting attacks around it
- Usually nodes collaborate to increase the efficiency of the detection
- However, this approach consumes a lot of resources throughout the network
- It is usually required to optimize the IDS periodically to minimize its effects

Placement of IDSs

- Hybrid IDS placement: to get the best of both worlds
- Central nodes with more resources are responsible for computationally intensive tasks (analyzing data, decision making)
- Normal nodes are responsible for lightweight duties (e.g., monitoring neighbor nodes, send info about traffic passing through them, responding to requests from central nodes)
- Requires optimization