draft-tiloca-6tisch-robust-scheduling-01

Authors: Marco Tiloca

Simon Duquennoy

Gianluca Dini

Recap

- An external adversary can easily and efficiently:
  - Derive the communication pattern of a victim node
  - Selectively jam the exact cells of the victim's schedule
  - The attack is effective, stealthy, targeted and low-power

- Preventive solution against selective jamming
  - Efficient pseudo-random shuffling of cells, at each slotframe
  - Agnostic of the specific scheduling algorithm
  - No communication overhead (only local computation)

- Resulting new schedule
  - Collision-free and consistent
  - Unpredictable to the adversary

Updates from -00 (1/3)

- Attack importance
  - Selective jamming of the exact victim’s cells
  - High effectiveness with minimal exposure (i.e., low risk of detection)
  - High energy efficiency, i.e. can be carried out on battery
  - More convenient than a wide-band constant jamming

- Adversary model
  - External, i.e. not controlling any node in the network
  - Can target one or many nodes in the network
  - Will target specific nodes and their traffic, i.e. not the network as a whole

Updates from -00 (2/3)

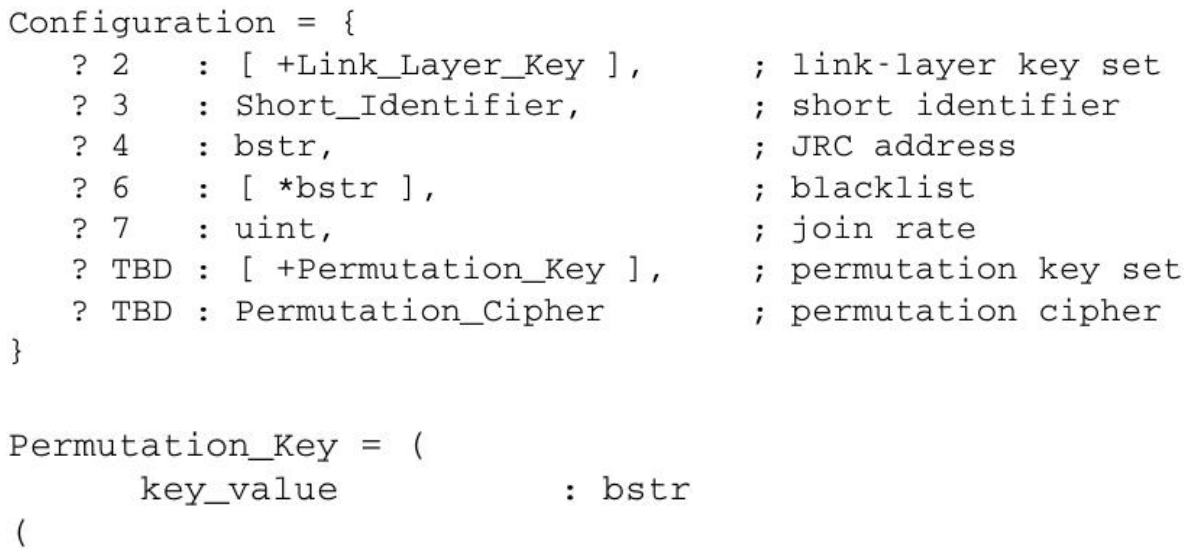
- Solution limitations
  - Intended to operate on slotframes used only for data transmission
  - NOT intended to operate on slotframes used (also) for joining traffic
- Keep the joining process feasible and deterministic
  - We can’t shuffle slotframes with a “minimal cell” or other randez-vouz cells
  - Cells for joining are practically in separate slotframes, e.g. Slotframe 0
- The adversary can still:
  - Jam the “minimal cell” or other randez-vouz cells
  - Jeopardize the joining process altogether

Updates from -00 (3/3)

- Provisioning of the permutation keys
  - MAY happen within CoJP in the Minimal Security Framework
  - Aligned with the latest format of the CoJP Join Response message

- New parameters
  - Permutation Key Set (1 or 2 keys)
  - Permutation Cipher

- Error handling is described



```
Configuration = {
    ? 2    : [ +Link_Layer_Key ],       ; link-layer key set
    ? 3    : Short_Identifier,          ; short identifier
    ? 4    : bstr,                      ; JRC address
    ? 6    : [ *bstr ],                 ; blacklist
    ? 7    : uint,                      ; join rate
    ? TBD  : [ +Permutation_Key ],      ; permutation key set
    ? TBD  : Permutation_Cipher         ; permutation cipher
}

Permutation_Key = (
        key_value             : bstr
(
```

Summary and next steps

- Addressed comments and actions from IETF 103
  - Attack importance and adversary model
  - Limitations of the solution
  - Key provisioning in the Join Response of CoJP

- Next steps
  - Need for document reviews – Anyone interested?

Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-6tisch-robust-scheduling