# End-of-study internship

## Group Key Management and IoT Security

Ramy Chemak

*INSA Centre Val de Loire*
*Heudiasyc UMR CNRS 7253*

August 31st 2021

# Overview

- ***Stage 1:*** Election-based Key Management Protocol for IoT
    1. Literature Review
    2. Election-based protocol
    3. Future works
- ***Stage 2:*** IoT Security engineering
- ***Conclusion & Feedback***

# Stage 1:
# Election-based Key Management Protocol for IoT

# Stage 1

# Literature Review

# Group Key Management: Sum up

- Network subdivided into several groups
- Each network node belong to a group
- The Key Manager (KM) manages different cryptographic keys
- Security considerations:
  - ✓ Forward secrecy
  - ✓ Backward secrecy
  - ✓ Collusion attack recovery

# Group Key Management: Problematic

The KM is responsible for:

- rekeying the group when needed
- generating keys for joining nodes
- revoking keys for leaving nodes

The KM is responsible for:

- rekeying the group when needed
- generating keys for joining nodes
- revoking keys for leaving nodes

**Single point of failure**

The Key Manager is responsible for the network's security infrastructure. Hence, its breakdown or compromise can jeopardize the overall network's security.

# Multi Group Key Management Protocol

- Subdivided into 3 layers
- Considerate multi services and heterogeneous networks

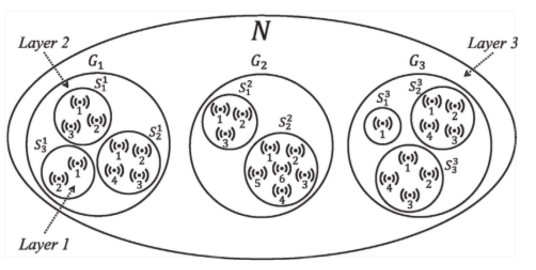
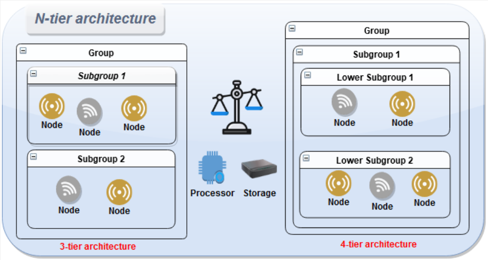
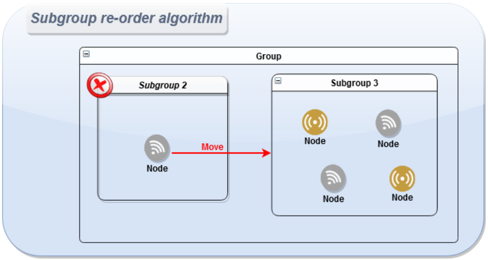



**Figure:** Source [2]: Example of a network partitioning

- Analysis of an n-tier architecture



- Analysis of an n-tier architecture
- Re-order algorithm upon leave



- Analysis of an n-tier architecture
- Re-order algorithm upon leave
- Sub-grouping sequences

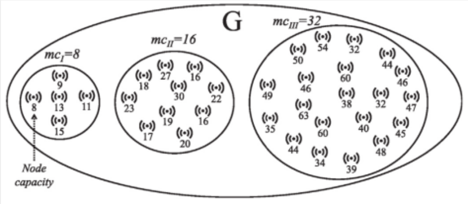
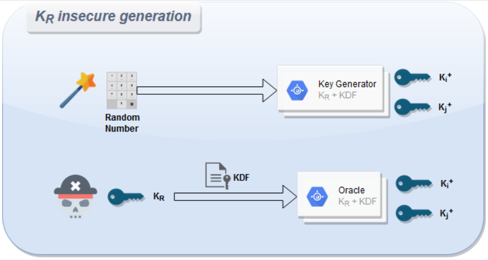


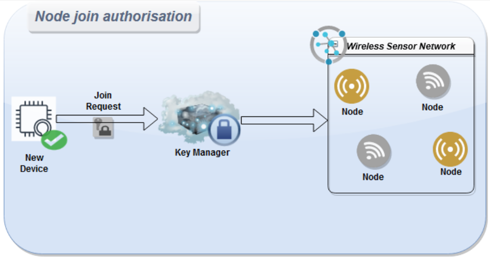

**Figure:** Source [2]: Example of a group partitioned using powers of 2 sequence

- Analysis of an n-tier architecture
- Re-order algorithm upon leave
- Sub-grouping sequences
- Refresh key generation



# MGKMP: Working tracks and proposals

- Analysis of an n-tier architecture
- Re-order algorithm upon leave
- Sub-grouping sequences
- Refresh key generation
- Node's join authorization & pre-secure channel

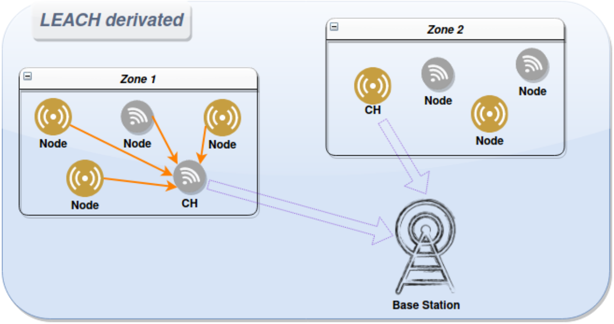


# MGKMP: Working tracks and proposals

- Analysis of an n-tier architecture
- Re-order algorithm upon leave
- Sub-grouping sequences
- Refresh key generation
- Node's join authorization & pre-secure channel
- Key Manager's single point of failure

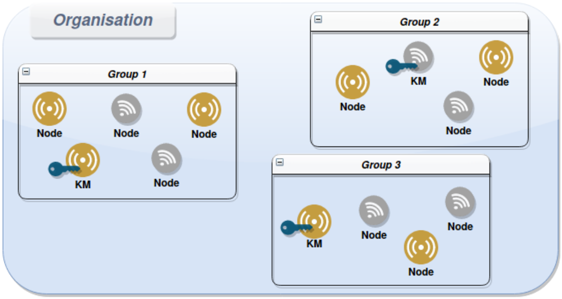
# Cluster Head schemes

- First considered for network routing purposes
- Network nodes are grouped in several groups or clusters
- Each group has its own Cluster Head, which is no more than one of its nodes



## Proposed solution

By applying Cluster Head schemes to Group Key Management (GKM), we are able to achieve a decentralized GKM architecture which solves the single point of failure's issue.



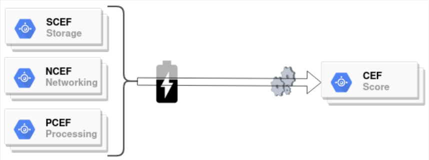
# Election-based protocol

# Workflow overview

- New decentralized solution for GKM
  - ✓ Capacity Evaluation Function
  - ✓ Election process
  - ✓ Failure recovery
- Simulations
- International conference paper

# Technical Eligibility Criteria

- Ensure the KM's reliability
- Considerate the nodes capacities:
  - ✓ Storage
  - ✓ Networking
  - ✓ Processing
- Also considerate energy



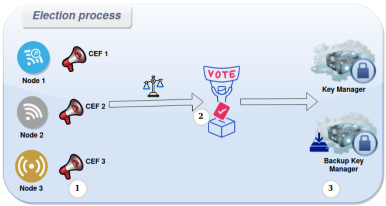
## Capacity Evaluation Function score

$$c_k = e_k \, . \, \left(w_s s_k + w_n n_k + w_p p_k\right)$$

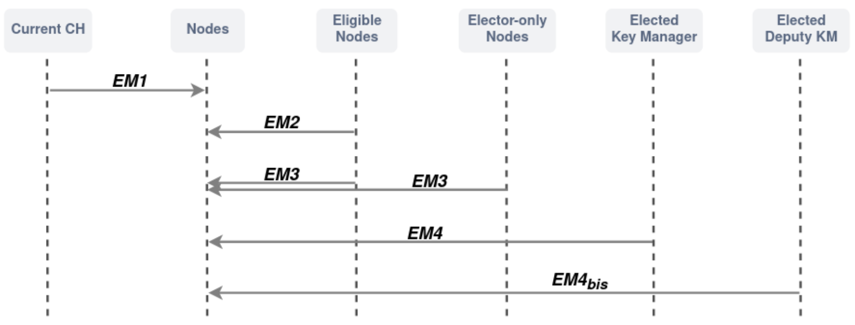
$$\text{where} \begin{cases} c_k : \textit{capacity score of a node } u_k \\ e_k : \textit{energy attribute of a node } u_k \\ s_k : \textit{storage capacity of a node } u_k \\ p_k : \textit{processing capacity of a node } u_k \\ n_k : \textit{networking capacity of a node } u_k \\ w_i : \textit{capacities weights} \end{cases}$$

# Election process

- All nodes are voters, but not all are eligibles candidates
- Eligible nodes broadcast their CEF score
- Nodes vote for best two candidates
- Elected Key Manager & Deputy KM claim their roles

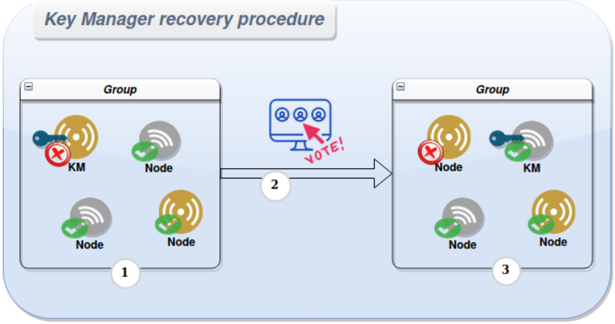


# Election process

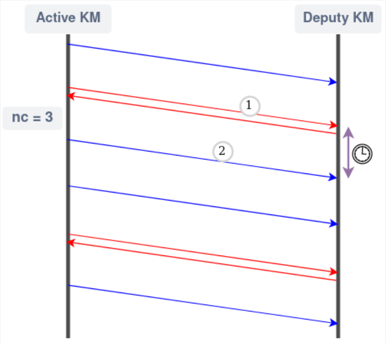


# Failure recovery

- Security enforcement
- Maintain Integrity & Availability

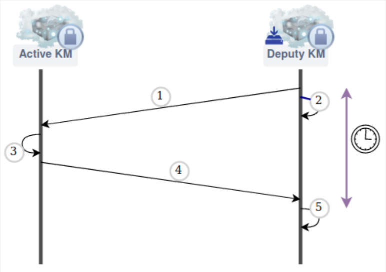


- Security enforcement
- Maintain Integrity & Availability
- Two check-over routines
    1. Simple check-over
    2. Double check-over
- Performance-security compromise



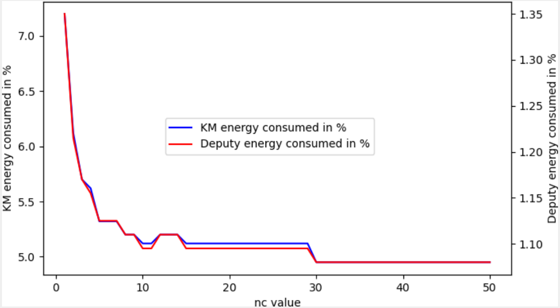
# Failure recovery

- Double check-over
- Challenge-response procedure
- Requires fast & correct answer
- Ensures both integrity & availability



# Simulation

We are able to quickly reach a satisfactory performance-security compromise.



# International publication

- Conference paper in the proceedings of the ***International Conference on Communications Software (SoftCOM 2021)***
- Acceptance notification received on July 23rd 2021

SoftCOM 2021

23-25 September 2021 // Hvar, Croatia

Technically co-sponsored by:

IEEE ComSoc IEEE Communications Society

IEEE Advancing Technology for Humanity

29th International Conference on Software, Telecommunications and Computer Networks

# Future works

# Future works

- More advanced simulations

# Future works

- More advanced simulations
- Real-environment experiments

# Future works

- More advanced simulations
- Real-environment experiments
- Revision of the current protocol

# Stage 2: IoT Security engineering

- Threats landscape IoT
    - ✓ IoT Malwares

- Threats landscape IoT
  - ✓ IoT Malwares
- Common IoT vulnerabilities
  - ✓ Default passwords
  - ✓ Irregular updates
  - ✓ IoT fleet management
  - ✓ . . . etc

# Conclusion & Feedback

Thanks for your attention !

# References


Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan.
**Energy-Efficient Communication Protocol for Wireless Microsensor Networks.**
page 10, 2000.

Mohamed Ali Kandi, Hicham Lakhlef, Abdelmadjid Bouabdallah, and Yacine Challal.
**A versatile Key Management protocol for secure Group and Device-to-Device Communication in the Internet of Things.**
*Journal of Network and Computer Applications*, 150:102480, January 2020.

Marco Tiloca and Gianluca Dini.
**GREP: A group rekeying protocol based on member join history.**
In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 326–333, Messina, Italy, June 2016. IEEE.

# Technical Eligibility Criteria

## Storage Capacity Evaluation Function

$$s_k = pm \;.\; \frac{sc_k}{ks}$$

$$\text{where} \begin{cases} s_k : \textit{storage capacity of a node } u_k \\ sc_k : \textit{storage capability of a node } u_k \\ pm : \textit{usable percentage of memory by protocol} \\ ks : \textit{size of a key} \end{cases}$$

# Technical Eligibility Criteria

## Processing Capacity Evaluation Function

$$p_k = pp \, . \, \frac{cc_k}{cs \, . \, (p + m_j)}$$

$$\text{where} \begin{cases} p_k : \textit{processing capacity of a node } u_k \\ cc_k : \textit{computation capability of a nod } eu_k \\ pp : \textit{usable percentage of processor by protocol} \\ cs : \textit{overhead of crypto system} \\ p : \textit{number of subgroups of the group } G \\ m_j : \textit{number of nodes in subgroup } S_j \end{cases}$$

# Technical Eligibility Criteria

## Networking Capacity Evaluation Function

$$n_k = bw_k \, . \, \frac{rr_k}{(p + m_j) \; . \; \max(ms)}$$

$$\text{where} \begin{cases} n_k : \textit{networking capacity of a node } u_k \\ bw_k : \textit{bandwidth of } u_k \textit{ usable by the protocol} \\ rr_k : \textit{radio range of } u_k \\ ms : \textit{size of a message} \\ p : \textit{number of subgroups of the group } G \\ m_j : \textit{number of nodes in subgroup } S_j \end{cases}$$

# Technical Eligibility Criteria

## Energy correlation

$$e_k = \frac{re_k}{ed_k \,.\, pu_k}$$

$$\text{where} \begin{cases} e_k : \textit{energy attribute of a node } u_k \\ re_k : \textit{residual energy of a node } u_k \\ ed_k : \textit{energy drainage of } u_k \\ pu_k : \textit{percentage of processor in use for } u_k \end{cases}$$

# Protocol revision

### Energy correlation (plain)

$$e_k = \frac{re_k}{ed_k \, . \, pu_k}$$

OR

### Energy correlation (configurable)

$$e_k = \frac{re_k^{\alpha}}{ed_k^{\beta} \, . \, pu_k^{\gamma}}$$