Data Consistency Introduction

Levente Buttyán, László Csik

Laboratory of Cryptography and System Security (CrySyS)

Budapest University of Technology and Economics

{levente.buttyan,laszlo.csik}@crysys.hu

Overview

- Introduction
- Goals
- Adversary types
- Initialization
- Model maintenance
- Consistency – Inconsistency
- Heuristics
- Example
- Conclusion

Introduction

- To meet performance goals VANETS will highly rely on node-to-node communication
  - Emergency signals
  - Road condition information
  - E-commerce applications
  - Route planning

  → Network security is important in these cases

- This kind of communication can be tampered easily

- The traditional approach ensures data integrity/authentication

  → Rises privacy problems, requires security overhead

Observation: We should rather deal with transmitting fraud data than data modification

Introduction II

- If the message has high importance, it must be authenticated
- If it is not the case, it might be sufficient to somehow try to filter fraud messages

  →This is the goal of Data Consistency enforcing primitives

- If there is no applied cryptography
  - The security in a VANET relies upon the potentially more challenging problem of ***detecting*** and ***correcting*** malicious data
  - These data can be generated by the car or by the user
  - We should defend against dishonest users
    - In large scale VANET there is no guarantee that a previously honest node will not be corrupted
  - And also against faulty sensors
- If a sensor is tampered this kind of attack cannot be prevented, neither detected, by cryptographic mechanisms

Adversaries in VANETS

Attack is successful if target node or nodes accept incorrect data as valid

Classification of attacks

- Attack nature
  - Adversary lies about themselves or about other node(s)
- Attack target
  - Local vs Remote attacks
- Attack scope
  - Effected area is limited or extended
- Attack impact
  - Undetected, Detected, Detected and Corrected

Data Consistency - Initialization

- Each communicating node maintains a ***model** of the VANET* containing all the knowledge that the node has of the VANET

- A model contains different ***rules***, that are derived from the physical world
  - Two nodes can never occupy the same location
  - Node rarely travels faster than 200 km/h
  - Other external constraints

- The node seeds the model with data measured by itself
  - It is assumed that data used to seed the model (collected by the node) is trusted

Model updating

- The node with an initialized model can then test the validity of data received from other nodes against this model of the VANET:
  - If all the data agrees with the model (perhaps with high probability), the node accepts the validity of the data

- → The problem occurs when the data is inconsistent with the model

- To deal with inconsistency the model must define ***heuristics*** that are used to resolve the conflicts
  - These heuristics are basically based on the assumption that a node is not malicious with high probability
  - If Sybil attacks are not feasible, the above defined statement holds

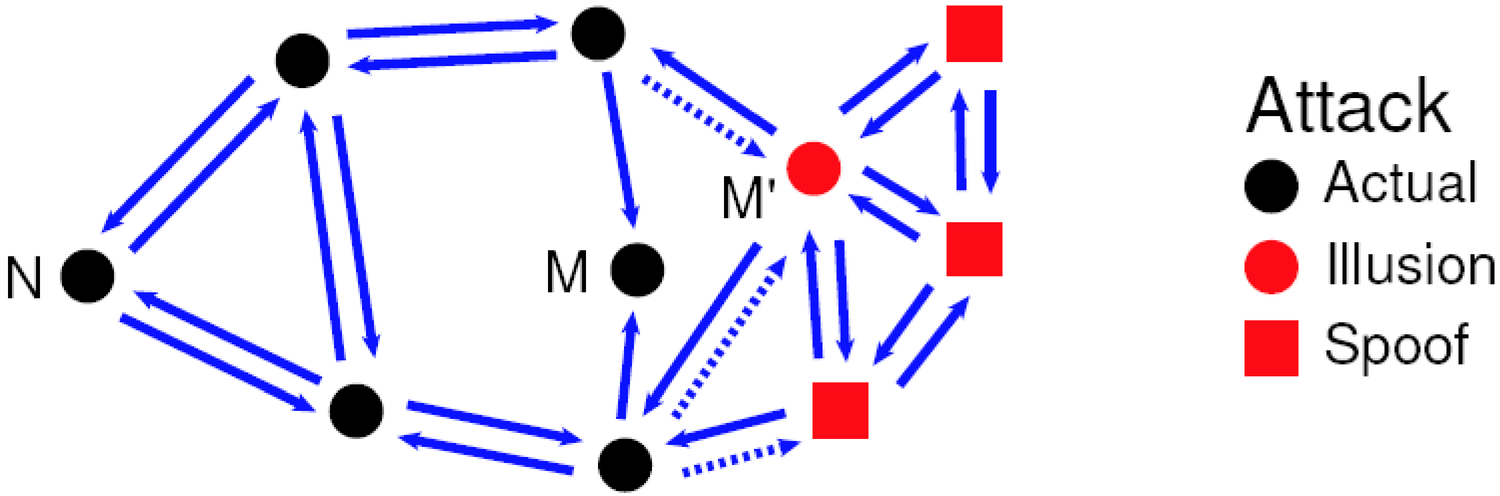
- Sybil attack is when a malicious node can create additional virtual nodes, with their own virtual observations

Heuristics

- Heuristics can be application specific
- It defines techniques to resolve inconsistency
- It also contains some kind of ordering precedency

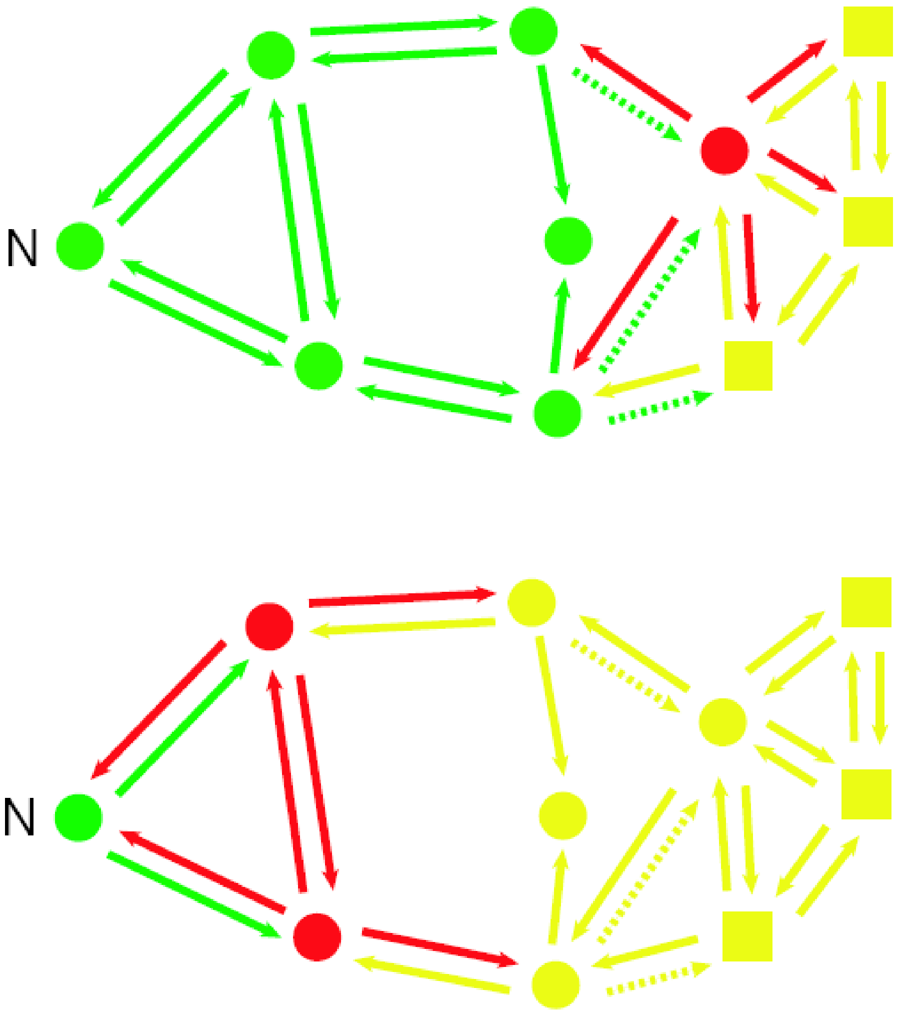
- The heuristic defines a list of possible explanations on the inconsistent model
- It decides using the ordering function
- Usually the node accepts the most probable explanation (Occams Razor)
- It sets all additional information the explanation requires

The real model



- A single malicious node *M* creates spoofs to support a false location *M’*.
- Blue arrows: Observations
- Dashed arrows: Missing observations.

Possible explanations



Explanation

🟢 Actual

🔴 Malicious

🟡 Spoof

Example – Explanation

- In the previous solution the first possible explanation requires less malicious node

  → it can be accepted, and it is the correct

- This solution was originally for distributed sensors, but it is also applicable to topology
  - Originally to correct fraud data
  - Median / Average
  - RANSAC Paradigms

Conclusions

- Error corrections makes the system fault tolerant
- Increases robustness
- The solution can correct errors that cannot be detected via simple cryptography
- It can eventually correct the received data, not just simply receive it
- Although this requires a good working model and good heuristics

  →Both of them are hard to be measured, defined

  We should check whether there is a good model of VANETS and define heuristics