Assessing trade-offs between energy consumption and security in sensor networks: simulations or testbeds?

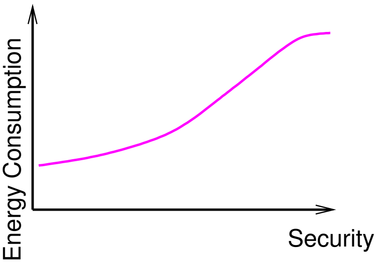
F. Maraninchi[1], Grenoble INP/Ensimag, VERIMAG
mainly summarizing discussions with:

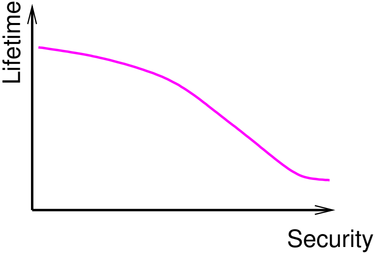
Laurent Mounier, Karine Altisen, Stéphane Devismes,
Pascal Lafourcade, Raphaël Jamet, Ozgün Pinarer, ...

November 6, 2014

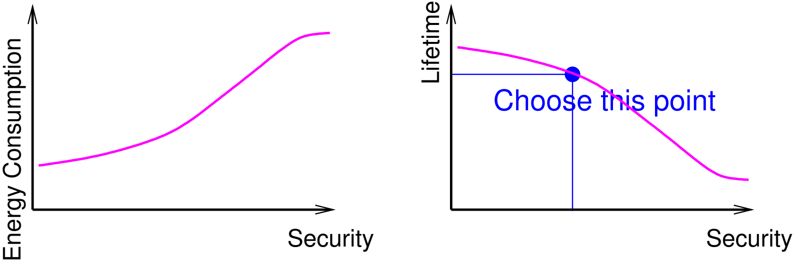
[1] http://orcid.org/0000-0003-0783-9178

Trade-off Between Security and Energy Consumption in Sensor Networks Applications...

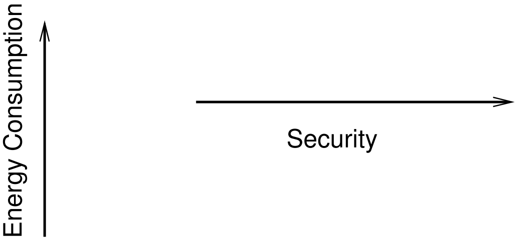




Trade-off Between Security and Energy Consumption in Sensor Networks Applications...



Two Sub-Questions



1. How to evaluate energy consumption of a complete HW/SW solution?
2. How to define and implement measurable security?

In such a way that we trust simulations/experiments.

We worked on points 1 and 2 in the ANR ARESA and ARESA2 projects, with Orange Labs; and also on point 1 in the ANR HELP project, with STMicroelectronics.

This Talk

- Brief presentation of one particular security protocol for WSNs, for which security has been evaluated on abstract models of WSNs

  SR3: Secure Resilient Reputation-based Routing. Karine Altisen, Stéphane Devismes, Raphaël Jamet, Pascal Lafourcade - IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS 2013)

- More realistic assumptions and new questions on the attacker models
- Some (hopefully) more general comments/conclusions/questions

The SR^3 Protocol

The SR^3 Protocol: "Expected" Attackers

Dolev-Yao principles for security = the attacker is the strongest we can imagine, the only thing it cannot do is decrypt a message without the key.

The SR^3 Protocol: “Expected” Attackers

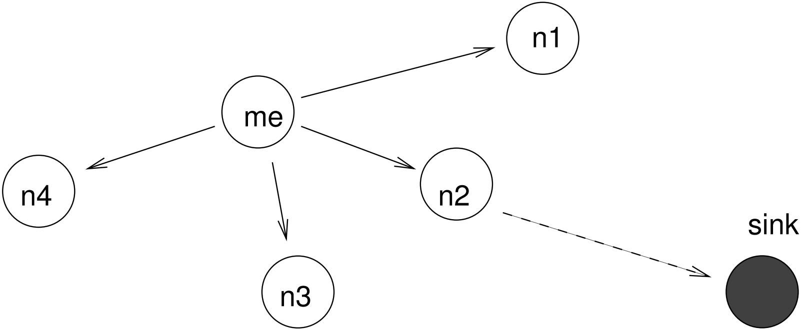
Dolev-Yao principles for security = the attacker is the strongest we can imagine, the only thing it cannot do is decrypt a message without the key.

Such a powerful attacker of a WSN will always win.

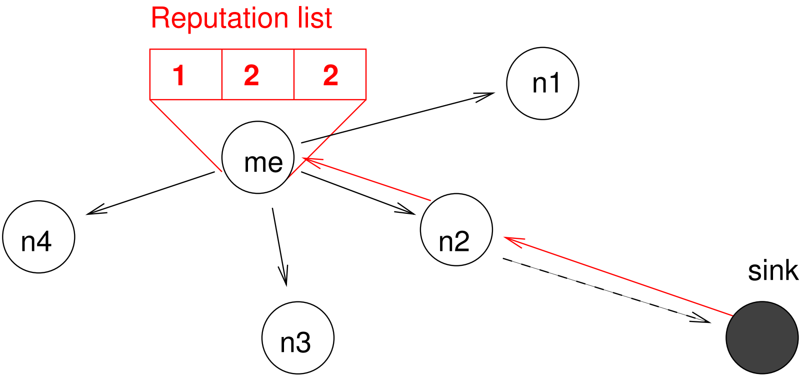
Weaker attackers considered:
“normal” nodes (w.r.t. HW capabilities) whose SW has been compromised
Example: a node can start behaving as a blackhole at some point in time.

Expected properties of SR^3: if there are not too many such nodes, the protocol will maintain a good delivery rate, at a reasonable cost.

The SR^3 Protocol: Main Ideas

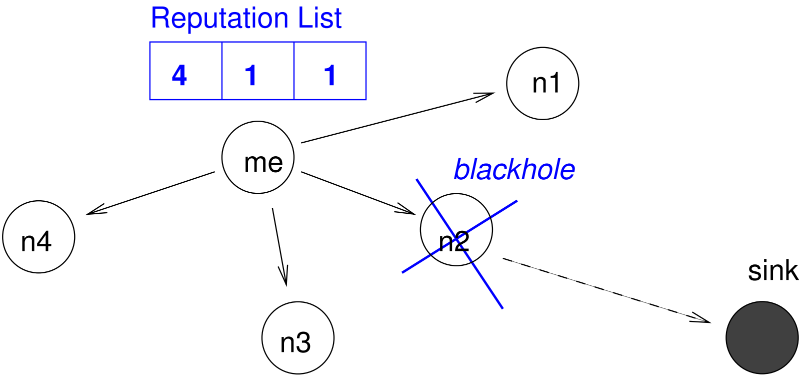


The SR^3 Protocol: Main Ideas



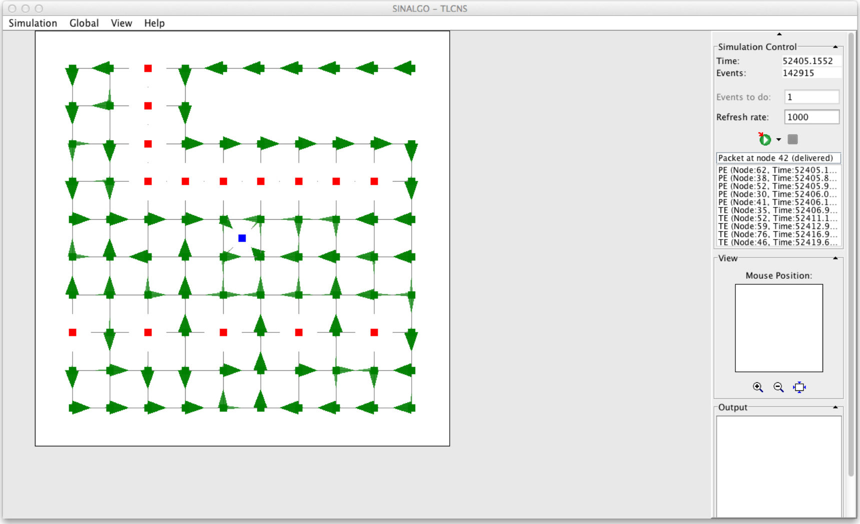
Choice of a neighbor:
(i) part of the time: random; otherwise random weighted by reputation list.

The SR^3 Protocol: Main Ideas



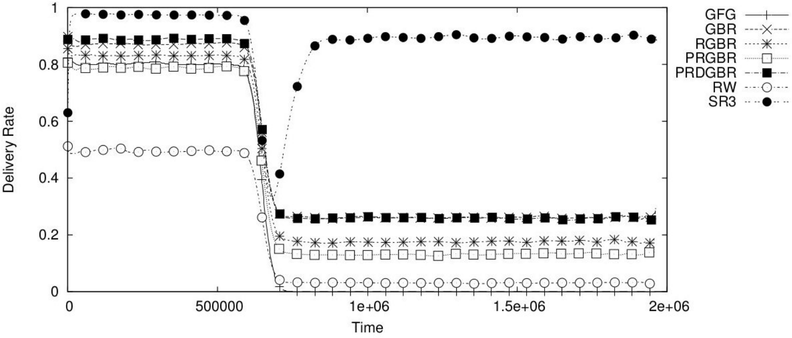
The SR^3 Protocol: Simulations with Sinalgo

(A distributed algorithm simulator, ETH Zürich)



The SR^3 Protocol: Simulations with Sinalgo

(A distributed algorithm simulator, ETH Zürich)



A node that has acquired a good reputation turns into a blackhole: the delivery rate drops, and then SR3 recovers.

Size of the reputation list: if increased, better routes, longer time to recover.

The SR^3 Protocol: Energy Consumption

For the moment, the Sinalgo models are based on 2 important hypotheses:

- The radio channel and the MAC layer provide the routing level with an ideal platform (no collisions)
- Energy consumption is directly related to messages (number of messages sent, length of the routes)

The SR^3 Protocol: Energy Consumption

For the moment, the Sinalgo models are based on 2 important hypotheses:

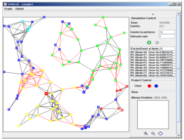
- The radio channel and the MAC layer provide the routing level with an ideal platform (no collisions)
- Energy consumption is directly related to messages (number of messages sent, length of the routes)

Next step: how would SR^3 behave with more realistic assumptions:

- Realistic radio channel and MAC protocol
- Energy consumption related to the behavior of the HW elements (CPU, radio, ...)

Experiments with SR^3

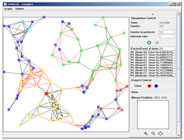
Sinalgo



WSNET
(ideal MAC)

Experiments with SR^3

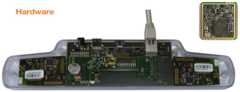
Sinalgo

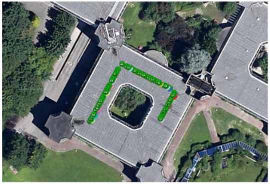


WSNET
(ideal MAC)

More understandable

SensOrLabs
testbed

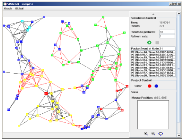




More realistic...

Experiments with SR^3

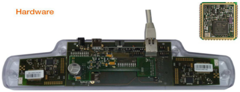
Sinalgo



WSNET (802.11)

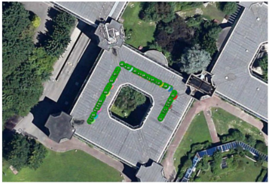
An Event-driven Simulator for Large Scale Wireless Networks

SensOrLabs testbed



WSNET (ideal MAC)

An Event-driven Simulator for Large Scale Wireless Networks



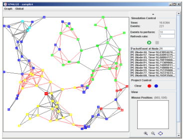
+ "realistic" medium

More understandable

More realistic...

Experiments with SR^3

Sinalgo



WSNET
(ideal MAC)

WSNET
(802.11)

+ realistic radio
(Q. Lampin)

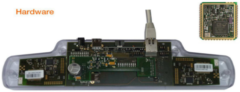
ΔT

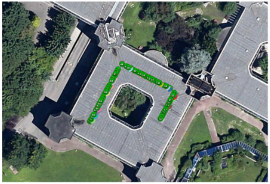
TX

consumption/
state

+ "realistic" medium

SensOrLabs
testbed





More understandable

More realistic...

What do We Expect?

1. Degraded performances due to the realistic radio channel + MAC
   longer routes, ...

What do We Expect?

1. Degraded performances due to the realistic radio channel + MAC
   longer routes, ...
2. New behaviors due to interference effects hidden by the "ideal communication" hypothesis
   spatial and temporal correlations for the success rate of transmissions, implying weird behaviors of the reputation lists, ...

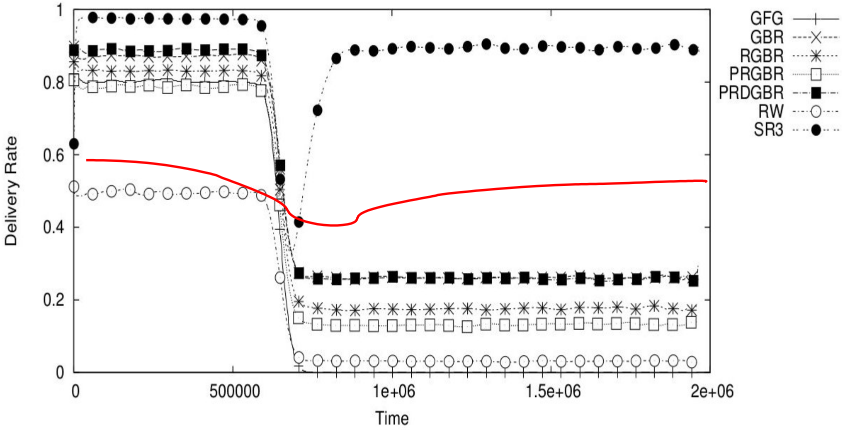
We try and build a situation for point 2
(topology, moment when a node becomes a blackhole, ...).

Unexpected Observation

With a realistic radio channel, it's hard to build an efficient attack!

Unexpected Observation

With a realistic radio channel, it's hard to build an efficient attack!



After Some Thought...

Very important impact of the attacker's position:
— Where to put the attacker to make it as strong as possible?

An idea:
We would like the honest nodes to suffer from the realistic channel,
but NOT the attackers!

General Comments

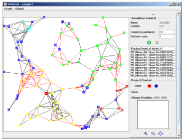
The More Details, The Better?

The more details you can put in a reasonably fast simulation, the better?

No, because:
— More details means less understandable results
— More details in the execution environment (HW+channel+attackers) make it more particular, hence the SW is less robust/secure.

Comments on the Experimental Settings

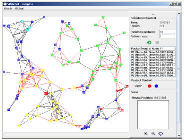
Sinalgo



WSNET
(ideal MAC)

Comments on the Experimental Settings

Sinalgo

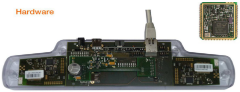


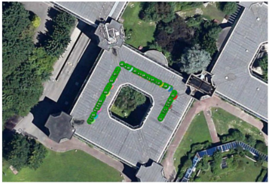
WSNET
(ideal MAC)

WSNet
An Event-driven Simulator for Large Scale Wireless Networks

More understandable

SensOrLabs
testbed

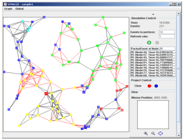




More realistic...

Comments on the Experimental Settings

Sinalgo

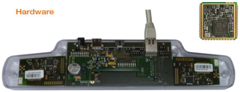


WSNET (802.11)

WSNet

An Event-driven Simulator for Large Scale Wireless Networks

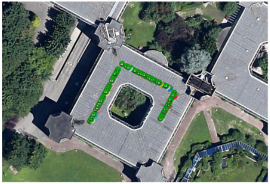
SensOrLabs testbed



WSNET (ideal MAC)

WSNet

An Event-driven Simulator for Large Scale Wireless Networks



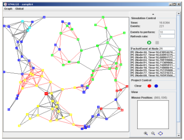
+ "realistic" medium

More understandable

More realistic...

Comments on the Experimental Settings

Sinalgo



WSNET
(ideal MAC)

An Event-driven Simulator for Large Scale Wireless Networks

WSNET
(802.11)

An Event-driven Simulator for Large Scale Wireless Networks

+ realistic radio
(Q. Lampin)

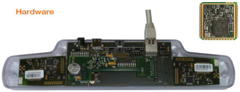
ΔT

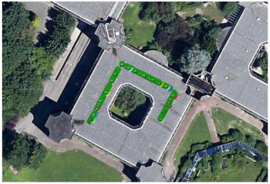
TX

consumption/
state

+ "realistic" medium

SensOrLabs
testbed





More understandable

More realistic...

The Ideal Modeling/Simulation Framework For Embedded+Distributed Systems?

(Given) Software
(drivers,
low–level,
MAC,
routing,
application)

The Ideal Modeling/Simulation Framework For Embedded+Distributed Systems?

(Given) Software
(drivers,
low–level,
MAC,
routing,
application)

(Has to be modeled) Adversary

HW of a node
radio channel

attackers

The Ideal Modeling/Simulation Framework For Embedded+Distributed Systems?

(Given)
Software
(drivers,
low–level,
MAC,
routing,
application)

Very particular ok

(Has to be modeled)
Adversary

HW of a node
radio channel

attackers

The Ideal Modeling/Simulation Framework For Embedded+Distributed Systems?

(Given) Software
(drivers, low–level, MAC, routing, application)

Very particular ok

(Has to be modeled) Adversary

HW of a node
radio channel

attackers

Unfriendliness should be exagerated

Exagerating Unfriendliness of the Execution Environment

(1) Exagerating Unfriendliness of the HW platform:

— The very goal of “transaction-level modeling” (TLM) for systems-on-a-chip.
— Feasible even though the REAL object code runs on rather abstract HW models (bit-accurate models)

Example with two timers:

Embedded SW:

```
start timer 1 delay 10
start timer 2 delay 20
wait expiration 1
wait expiration 2
```

Exagerating Unfriendliness of the Execution Environment

(1) Exagerating Unfriendliness of the HW platform:

— The very goal of "transaction-level modeling" (TLM) for systems-on-a-chip.
— Feasible even though the REAL object code runs on rather abstract HW models (bit-accurate models)

Example with two timers:

Embedded SW:

```
start timer 1 delay 10
start timer 2 delay 20
wait expiration 1
wait expiration 2
```

HW models of increasing unfriendliness:
1) exact timing: 10, 20
2) loose timing: [8,12], [18,22]
3) no timing: one day, one day

If even loose timing is considered too friendly, then untimed models are the only choice to guarantee SW robustness.

Exagerating Unfriendliness of the Execution Environment

(2) Exagerating unfriendliness of the attackers in a WSN?

We could imagine a mixed simulation mode in which:
— The honest nodes play by the rules, i.e. they suffer from the imperfections of the radio channel+MAC
— The attackers don't play by the rule, i.e., for them the transmission is perfect.

Exagerating Unfriendliness of the Execution Environment

(2) Exagerating unfriendliness of the attackers in a WSN?

We could imagine a mixed simulation mode in which:
— The honest nodes play by the rules, i.e. they suffer from the imperfections of the radio channel+MAC
— The attackers don't play by the rule, i.e., for them the transmission is perfect.

For the 2 timers: if even loose timing is considered too friendly, then untimed models are the only choice.

Exagerating Unfriendliness of the Execution Environment

(2) Exagerating unfriendliness of the attackers in a WSN?

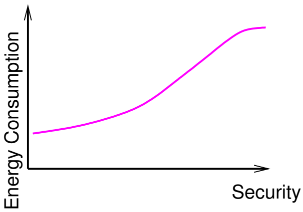
We could imagine a mixed simulation mode in which:
— The honest nodes play by the rules, i.e. they suffer from the imperfections of the radio channel+MAC
— The attackers don't play by the rule, i.e., for them the transmission is perfect.

For the 2 timers: if even loose timing is considered too friendly, then untimed models are the only choice.

For the attackers: if any position with the realistic radio channel is too particular/friendly, the only choice is to build an *ideal* position .

Conclusion (1)

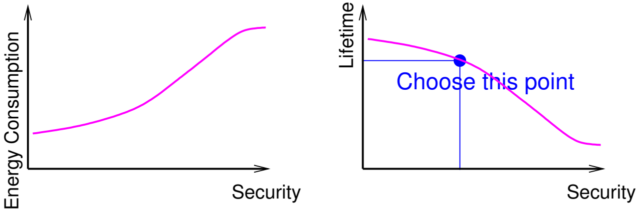
We are quite far from trusting views like:



We need frameworks in which we can think in terms of
our solution/the execution environment, i.e., the adversary

Conclusion (1)

We are quite far from trusting views like:



We need frameworks in which we can think in terms of
our solution/the execution environment, i.e., the adversary

Conclusion (2)

A very useful tool would be a mix of:

- testbeds and
- high level models with perturbations
  = exagerated view of the discrepancies between ideal and real behaviors

Questions?