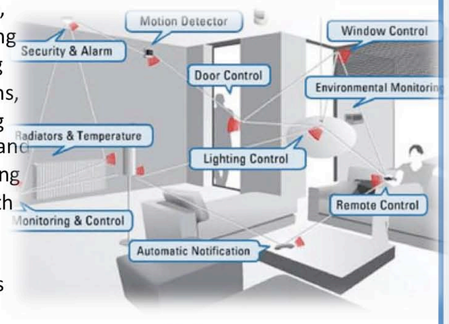
High security at a low cost?

Ender Yüksel, Hanne Riis Nielson, Flemming Nielson

DTU Informatics, Language Based Technology, {ey,riis,nielson}@imm.dtu.dk

Motivation 1

In the future tiny devices with microcontrollers and sensors will be in charge of numerous activities in our lives. Tracking our energy consumption and CO_2 emission, controlling our living conditions, enforcing security and monitoring our health will be some examples of their functions. They will form wireless sensor networks to communicate with one another, moreover their power consumption will be very low. It is not hard to predict that our modern society will depend on the correct operation of these devices, and the security of the network they are operating.



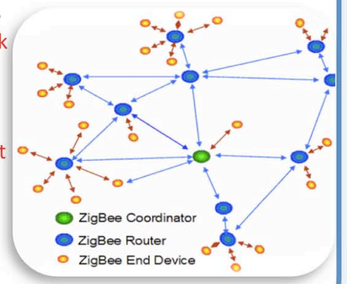
Running real-world example: *ZigBee* 3

ZigBee is a standardization for wireless sensor networks that is promoted by a large consortium of industry players. ZigBee is a low-rate standard in terms of:

cost, power consumption, range, and *bandwidth.*

The mere exception is intended to be in security. However, it is a challenging task to provide secure networking in such a low-rate environment.

ZigBee employs AES encryption. Network Key (NK) is the only mandatory key in network, shared by all the devices. Trust Center (TC), unique in each network, creates and distributes the NKs.



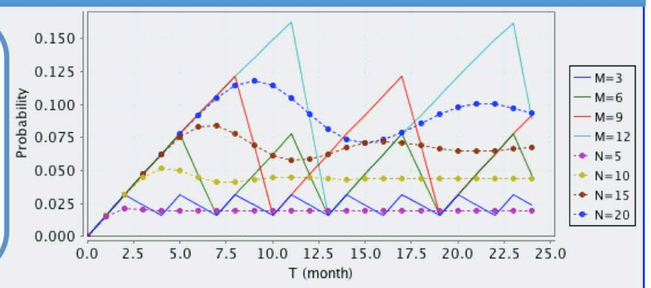
Key update strategies and proper determination of related security parameters still remain as gaps in the ZigBee standard.

Selected Results 5

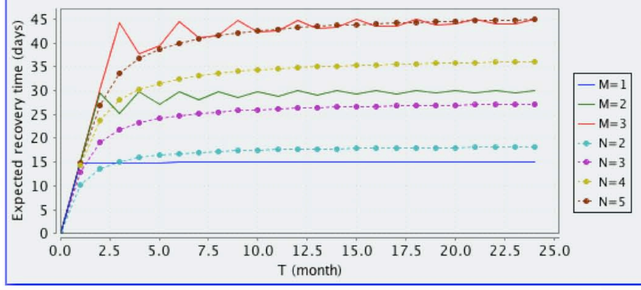
Probability of key compromise at a specific time

```
P=? [ F[30*T,30*T] Comp ]
```

The results show us that the Leave-based key update (Lb) provides a fairly stable risk for especially low threshold values. Besides we learn that in terms of the maximum key compromise probability, the two key update strategies are comparable for (M=9,N=20), (M=6,N=15), and (roughly) (M=3,N=5).



The dotted lines represent the Lb key update for different threshold values (i.e. update takes place after in total of N devices left the network). Solid lines represent the Tb key update for different period values (i.e. update takes place every M months). The experiments shown here are for the Home Automation Application Profile of ZigBee standard. The parameter values are: R_join=1/7, R_leave=1/365, P_comp=1/100, Max=20. The time unit is taken as 1 day.



Comparison of Mean Time to Recovery (MTTR)

```
R{"Recovery"}=? [ C<=30*T ]
--------------------------
R{"Compromise"}=? [ C<=30*T ]
```

Although the results for Tb are no surprise, the ones for Lb are not easy to guess without the help of this graph. In terms of recovery, there is a fight between M=1-N=2 (Tb wins!), M=2-N=3 (Lb wins!), and M=3-N=5 (Draw!). We see that for N=4 comparison does not make sense.

Problem 2

To achieve secure communication, cryptographic protocols are employed. Limited resources of devices restrict us to use symmetric encryption where all devices in the network share a common key. When a new device joins the network it will register with the coordinator and store the cryptographic key in its memory. But what happens when a device leaves the network? There is a risk that it still contains the key, and in the worst case this means that the key is in the hands of a dishonest person.

The Key Update Paradox

Naturally, we want to ensure that the risk of using a compromised key in a network is as small as possible

update the key often!

Still, this operation is computationally expensive and we would not like to perform it too often.

update the key rarely!

How can we balance these two viewpoints?

Modelling 4

The system is modelled as a Continuous Time Markov Chain (CTMC) where the frequencies of the various actions are exponentially distributed. The model is analysed using the PRISM model checker. The properties are expressed in the Continuous Stochastic Logic (CSL) and internally PRISM uses a number of stochastic algorithms to compute the analysis results.

Instead of [exponential distribution / rate for update action] we have implemented [phase-type distribution / instant transition] for modelling [periodical / leave-based] key updates more accurately.

The Network

Network keeps track of joining and leaving devices.

New devices may join	Devices may leave network
- *We may want new sensors/services*	- *We may remove some sensors/services*
- *We may rejoin an out of order device*	- *Batteries can be drained*

The Trust Center

Trust Center implements the key update strategies.

Time-based (Tb)	*Leave-based* (Lb)
- *The keys are updated periodically*	- *The keys are updated when a number of devices leave*

```
module NETWORK
 Size: [0..Max] init Max;
 [join]   Size<Max  -> R_join*(Max-Size):       (Size'=Size+1);
 [leave]  Size>0    -> R_leave*(1-P_comp)*Size: (Size'=Size-1);
 [leaveC] Size>0    -> R_leave*P_comp*Size:     (Size'=Size-1);
 [leaveR] Size>0    -> R_leave*Size:            (Size'=Size-1);
endmodule

module TRUSTCENTER
 Comp: bool init false;
 C_leave: [0..N] init 0;
 [join]   true -> true;
 [leave]  C_leave<N-1 -> (C_leave'=C_leave+1);
 [leaveC] C_leave<N-1 -> (C_leave'=C_leave+1) & (Comp'=true);
 [leaveR] C_leave=N-1 -> (C_leave'=0) & (Comp'=false);
endmodule

rewards "Recovery"              Comp:  1; endrewards
rewards "Compromise"  [leaveC] !Comp:  1; endrewards
```

A selection of the PRISM code for the leave-based model.

Example Advice 6

1. Choose application scenario:
- *Home Automation.*
  - *maximum 20 devices.*
  - *avg. join: 1 dev. per week*
  - *avg. leave: 1 dev. per year*
  - *key compromise risk: 1%.*

2. Define requirements:
- *R1 prob. that the key is compromised must be less than 10% at any time*
- *R2 prob. that the key recovery takes more than 3 months, must be lower than 99 %.*
- *R3 number of key updates should be less than 0.07 per day.*

3. Determine the parameters satisfying the requirements:

Req.	Tb (max) threshold	Lb (max) threshold
R1	6 months	15 leaves
R2	6 months	10 leaves
R3	5 months	9 leaves

RESULT: We can choose either
- *the time-based key update with threshold of 5 to 6 months, or*
- *the leave-based key update with threshold of 9 to 10 devices.*

Conclusion 7

We presented how stochastic model checking can be used to determine optimal security configurations for desired application profile, environmental settings and security requirements. Using this method, one can strike an acceptable balance between cost and security, and derive results to be used in real life.

Acknowledgements Bo Friis Nielsen (DTU), Dave Parker (Oxford University), Marta Kwiatkowska (Oxford University) Matthias Fruth (Oxford University), Robert Cragie (ZigBee Alliance)

Related Publications

- E. Yüksel, H.R. Nielson, F. Nielson. ***A Secure Key Establishment Protocol for ZigBee Wireless Sensor Networks***. In Proc. of the 24th International Symposium on Computer and Information Sciences (ISCIS 2009), pages 350-355, IEEE, 2009.
- E. Yüksel, H.R. Nielson, F. Nielson. ***ZigBee-2007 Security Essentials***. In Proc. of the 13rd Nordic Workshop on Secure IT-systems (NordSec 2008), pages 65-82, 2008.

and MT-LAB A VKR CENTRE OF EXCELLENCE

Ender Yüksel ey@imm.dtu.dk | ***Hanne Riis Nielson*** riis@imm.dtu.dk | ***Flemming Nielson*** nielson@imm.dtu.dk