Hacking

Erik Poll

Digital Security group

Institute for Computing & Information Science (ICIS)

Faculty of Science

Radboud University

Computer systems keep getting hacked...

1,912 views | Aug 16, 2019, 01:56am

European Central Bank Breach: ECB Confirms Hack And Shuts Down Website

Texas government organisations hit by ransomware attack

Hacked EV chargers could cause blackouts — study

Blake Sobczak, E&E News reporter

Published: Monday, August 19, 2019

UN: North Korean Hackers Raised $2B to Fund Weapons Program

ETHICAL HACKERS SABOTAGE F-15 FIGHTER JET, EXPOSE SERIOUS VULNERABILITIES

BY JASON MURDOCK ON 8/15/19 AT 8:19 AM EDT

How come this keeps happening?

- **Why can we not make computer systems without security flaws?**
- **Why are these flaws so dangerous?**

Exploring this will lead us to special properties of computers and central research questions in computing science

How to hack a computer system

Basically, two ways to do this :

1. **Attack the user**
   - **eg. phishing email to get username & password**
   - **aka social engineering**

1. **Attack the software**
   - **find flaw & exploit it**
   - **‘real’ hacking**

Hacking

Hacking = using something in a way it was not intended to be used,

getting it to behave in an unintended way

More hacking

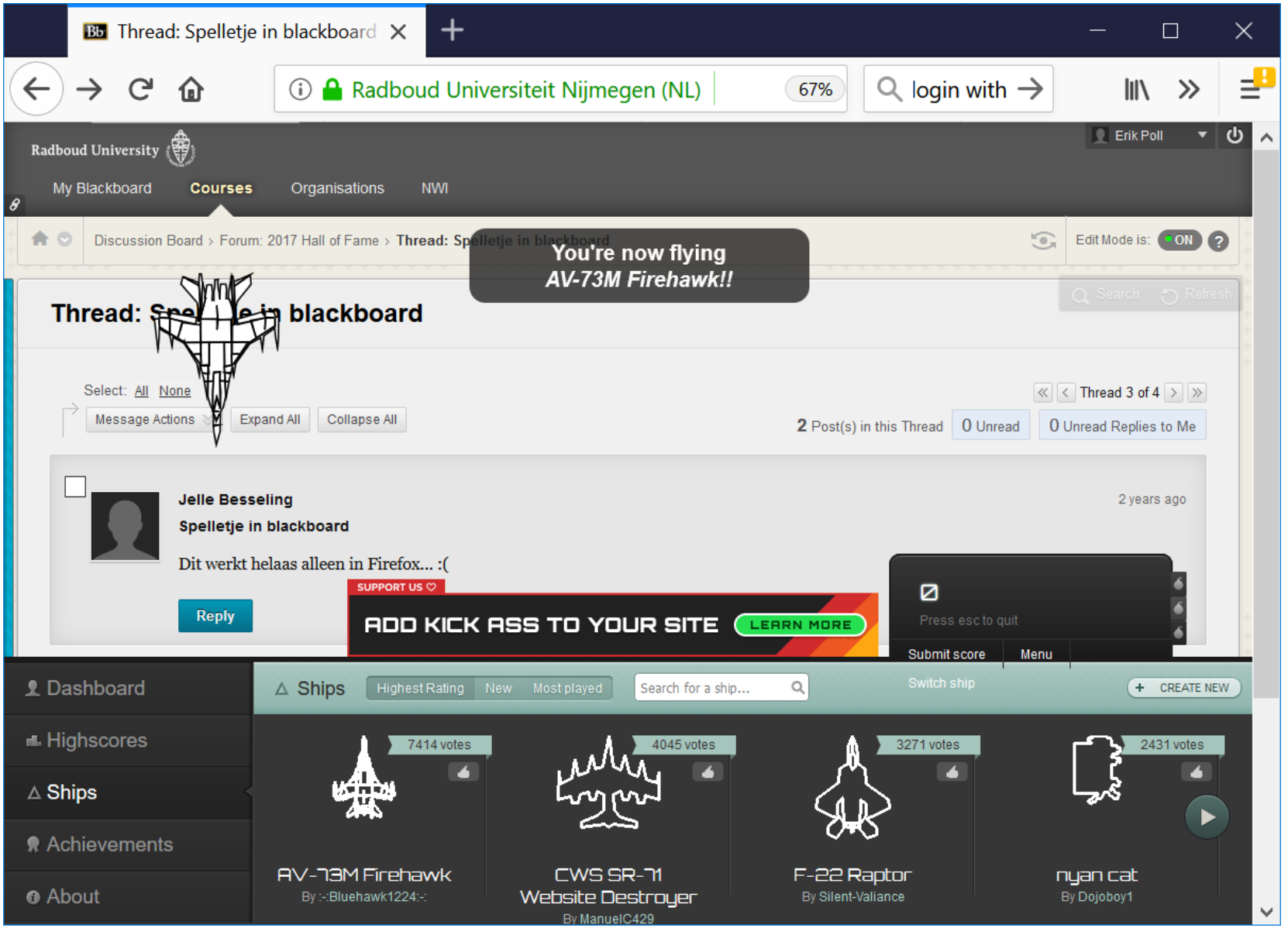
[Simone Giertz, shitty robot]

https://www.youtube.com/watch?v=D3sTjj1eeAA

Using charge pole to cook waffles

[Matthias Dalheimer, CCC'2018, https://evsim.gonium.net]

Hacking: game inside RU website



Hacking computers vs hacking mechanical devices

- Hacking the RU website you can re-program it to do ***anything***
  - e.g. play computer game, change your grades, mine bitcoin, use it to hack or DoS other computers, …
- Hacking a power drill, washing machine, hard disk you can only make it do variations of the same theme
  - unless there is a computer in it …

Special property 1: Software & programmability

Software can do *anything*

- Software gives computers the power & flexibility to do anything
  - Eg. smartphone can be used to text, surf the web, listen to music, watch videos, play games, online shopping, internet banking, …
  - Here we also use the power & flexibility of digital information
    - Digital information used to represent text, images, sound, video, digital money in your online bank account, digital items in online shopping basket, …

Downside: if an attacker gets in, this power can be used against you

Special property 2: Digital vs Analog

- Mechanical systems are analog systems
  - the speed of hard disk can be anything, giving an infinite number of possibilities for the speed

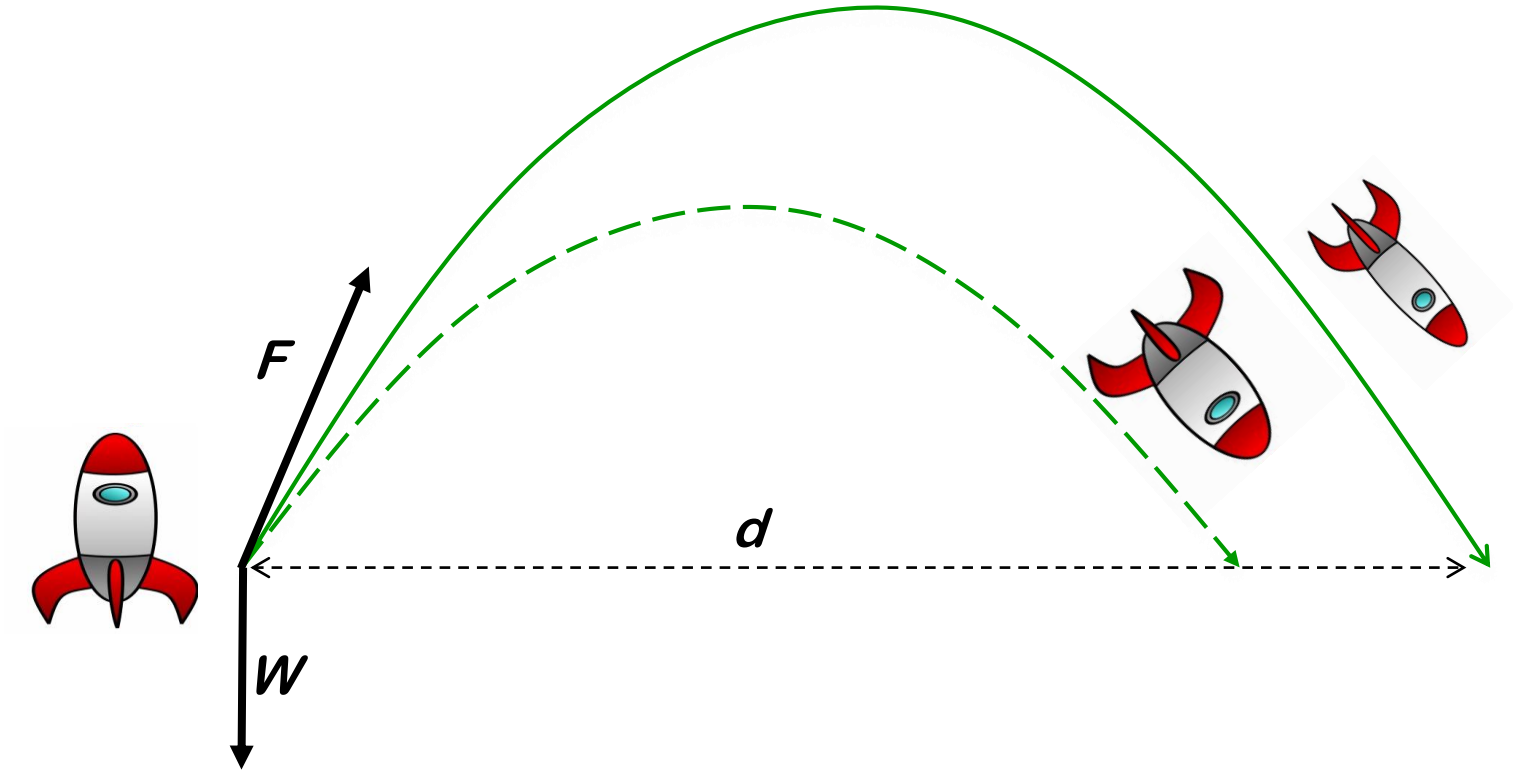
- Computers are digital or discrete systems
  - a bit is 0 or 1, byte can have 256 values, etc.
  - finite (but very large) number of possibilities
    - a computer with 35 byte memory has more states than there are sub-atomic particles in the universe

Paradox: understanding the finite behaviour of a computer is much harder than understanding the infinite behaviour of spinning hard disk

Understanding analog systems: rocket science

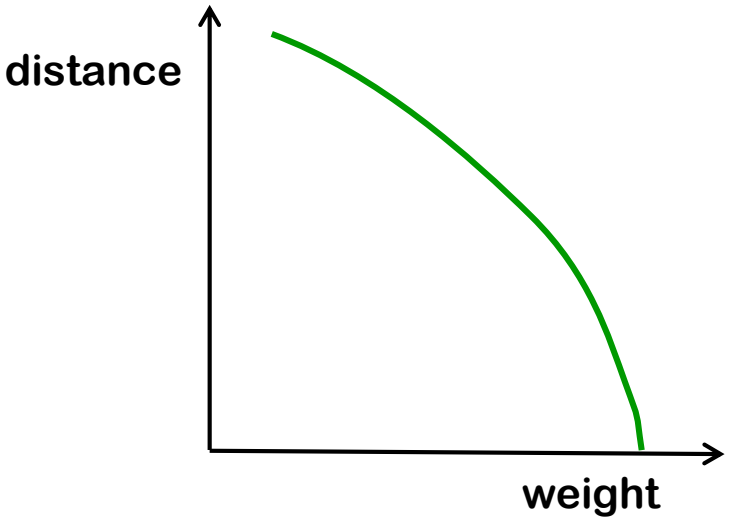
Mathematical model of flight path,
allows us to define for example $\mathbf{f} \in \mathbb{R} \longrightarrow \mathbb{R}$

where $\mathbf{f}(\mathit{Weight})$ is distance d travelled

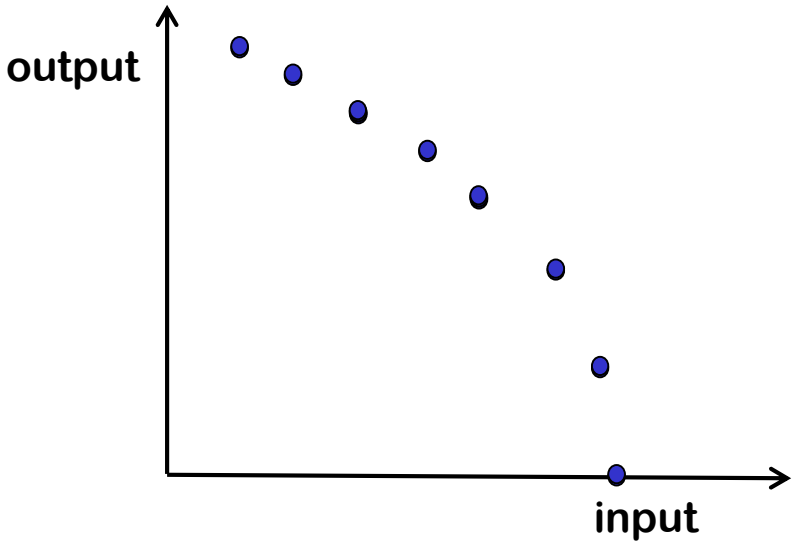


Analog vs DIGITAL systems

Analog system can be described with 'smooth' mathematical functions



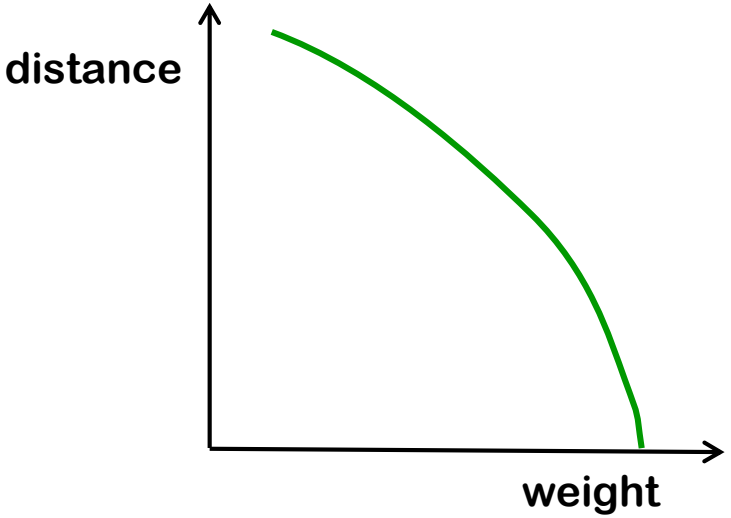
Digital systems are very different:



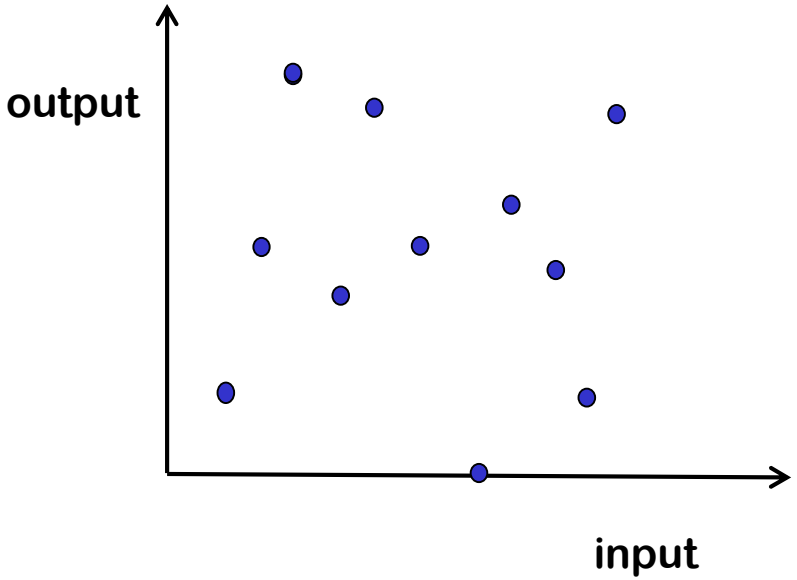
Small changes in input typically give small changes in output

Analog vs DIGITAL systems

Analog system can be described with 'smooth' mathematical functions



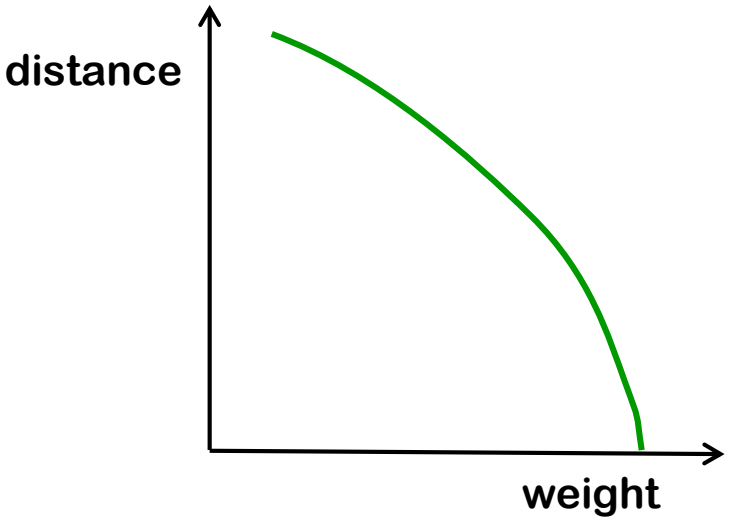
Digital systems are very different:



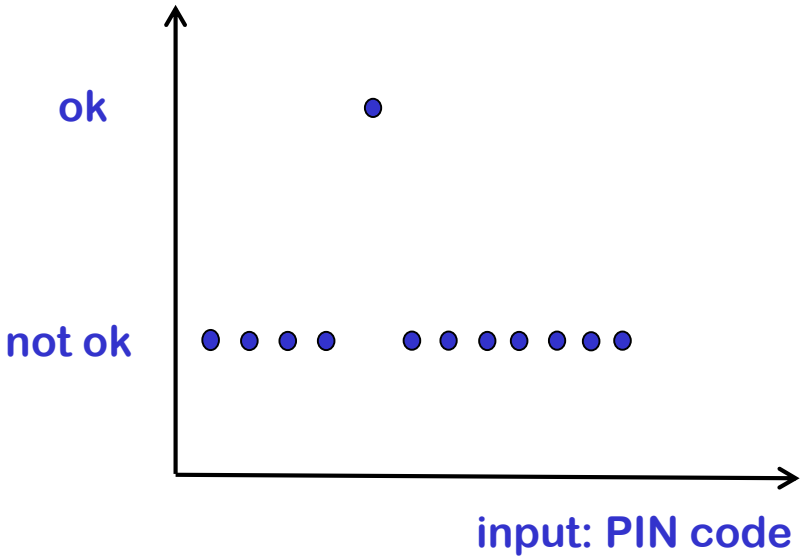
Small changes in input typically give small changes in output

Analog vs DIGITAL systems

Analog system can be described with 'smooth' mathematical functions



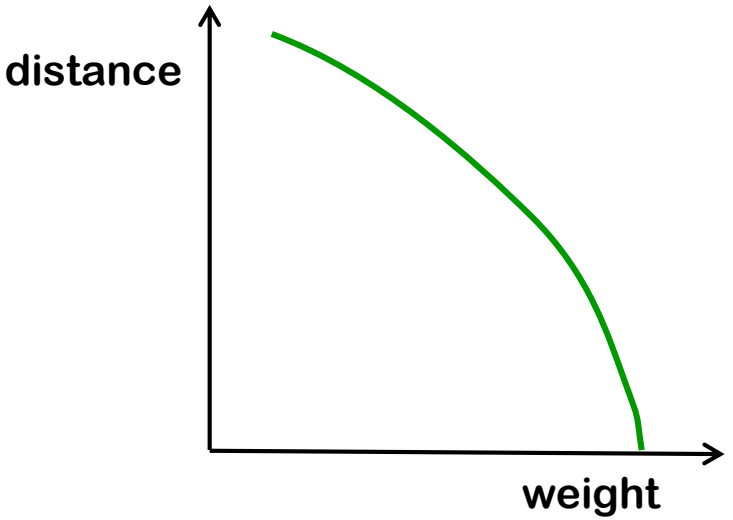
Digital systems are very different:



Small changes in input typically give small changes in output

Analog vs DIGITAL systems

Analog system can be described with 'smooth' mathematical functions



Small changes in input typically give small changes in output.

Digital systems are very different:

- Tiny change in input can completely change the behaviour
- Also, digital systems have memory, so they behave different over time

This makes understanding computer systems *much* more difficult.

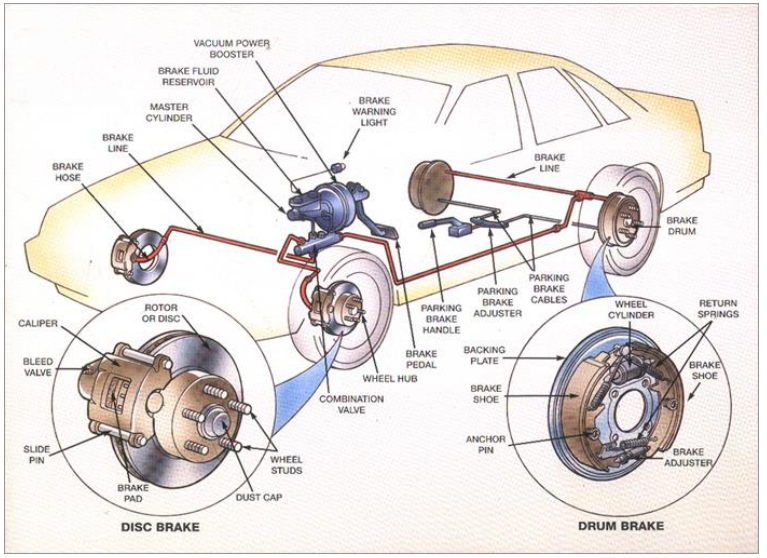
First launch of Ariane V rocket

Costly software bug:
converting from 64-bit floating point number to 16-bit signed integer

[https://www.youtube.com/watch?v=kYUrqdUyEpI]

Analog vs DIGITAL systems

- If an analog, mechanical braking system can stop the car at 40 km/h, it can also stop the car at 30 km/h



- If a digital, computerised braking system can stop a car at 40 km/h, it might fail to stop at car at 32.767 km/h

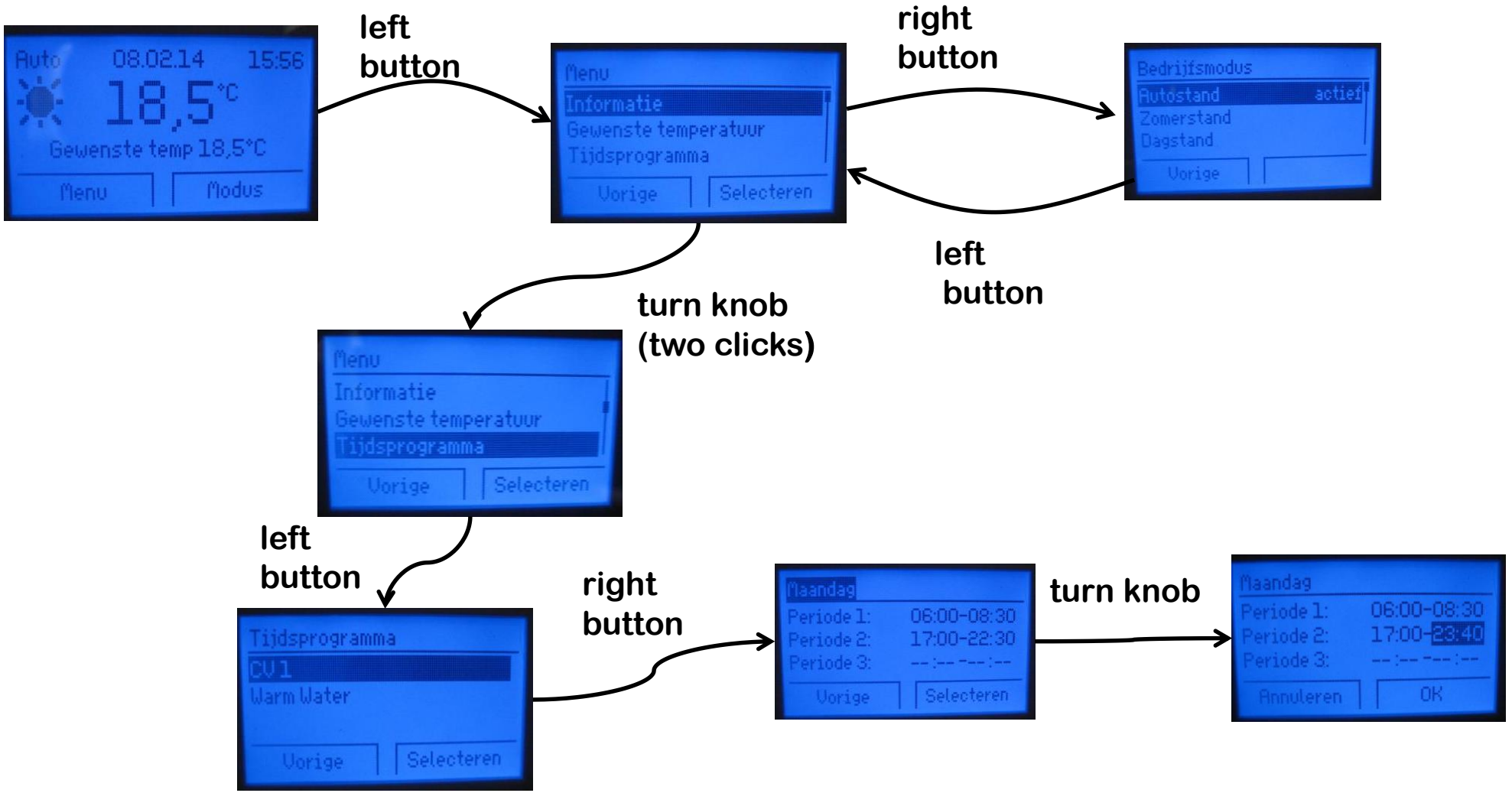
Example computer

How to understand – or *model* - this software?

Two ways to understand how it works

1. **read the manual**
2. **mess around to discover how it work**

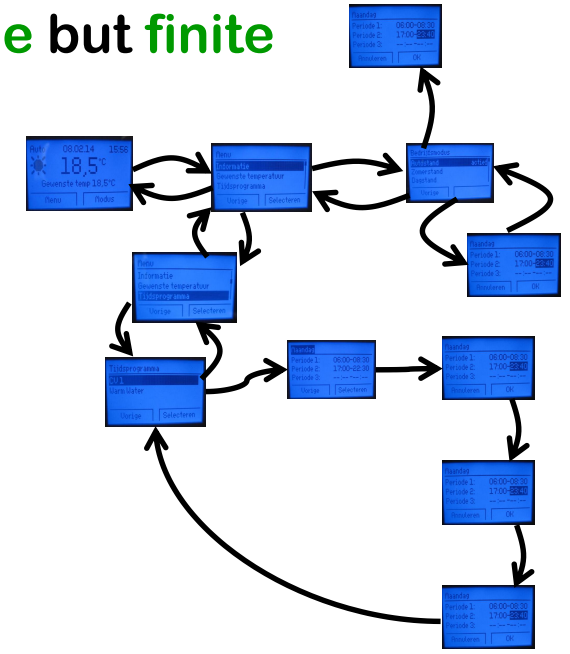
Understanding – or *modelling* – this software



This type of model is called a finite automaton

Understanding – or *modelling* – this software

Automaton describes the state space, which is huge but finite



- Exploring *enough* of the state space
  to operate the heating system is doable
- Exploring the *whole* state space is infeasible.
  There could be a bug somewhere, and
  an attacker might find & abuse it…

Central research questions in computing science:

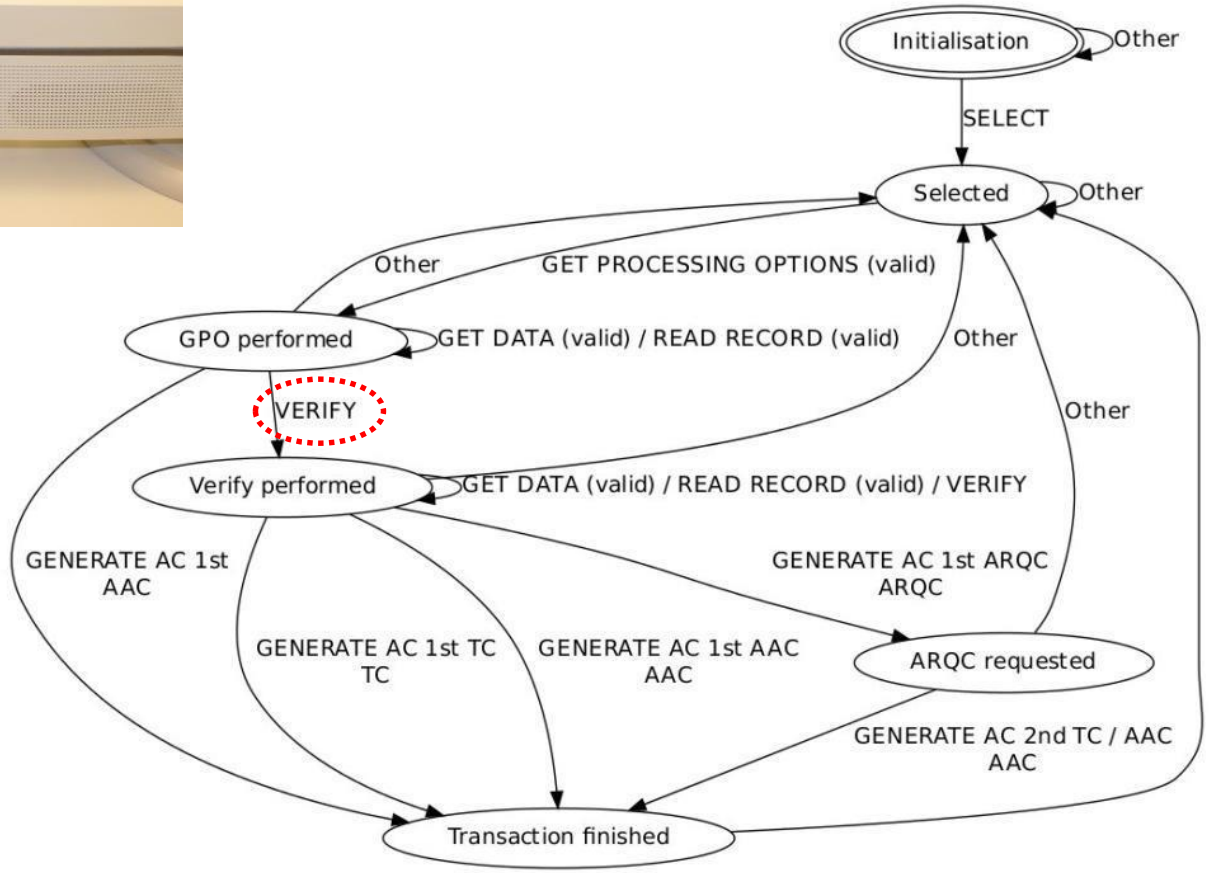
- How can we analyse such a system to know there are no bugs?
- Or: how can we construct it in such a way that there are no bugs?

Another example of a computer

Bank card has small computer inside, that you can talk to

- via **contact interface** 
- via **contactless interface**

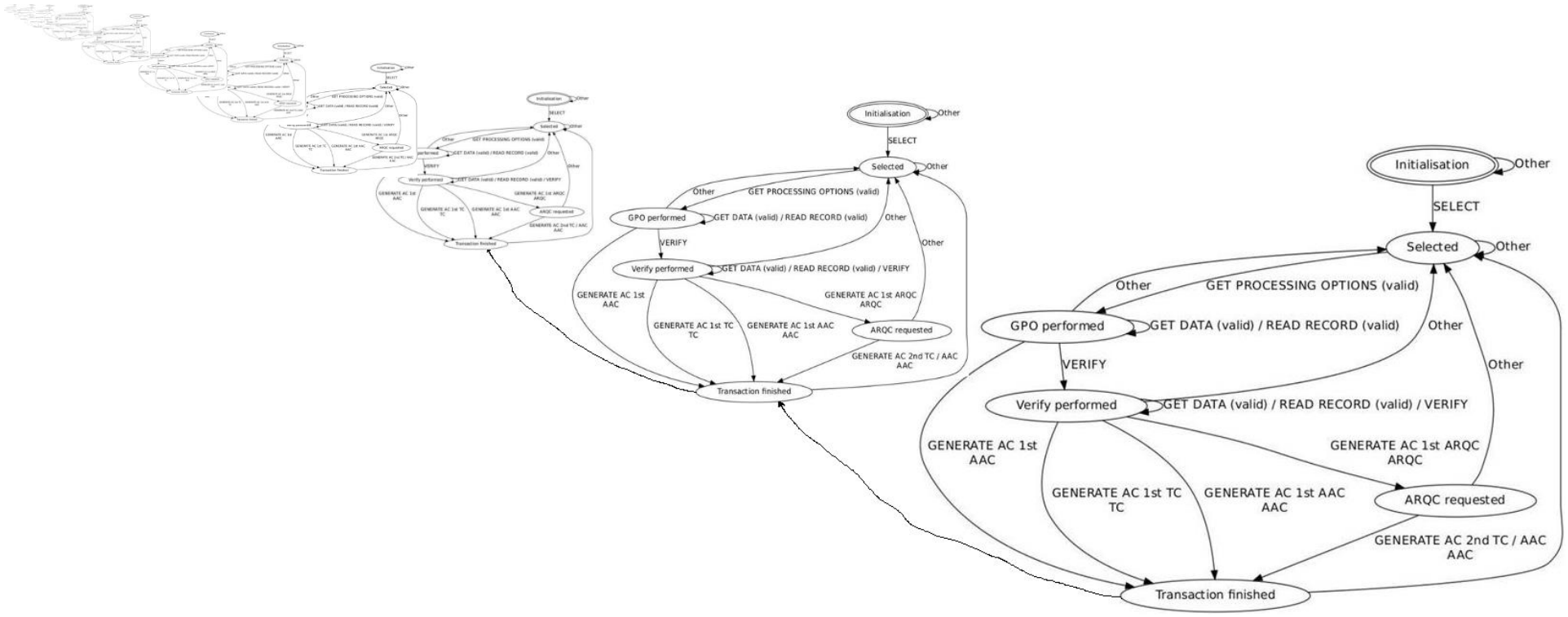
Exploring the behaviour of



Exploring the behaviour of

This is a simplification, or an abstraction

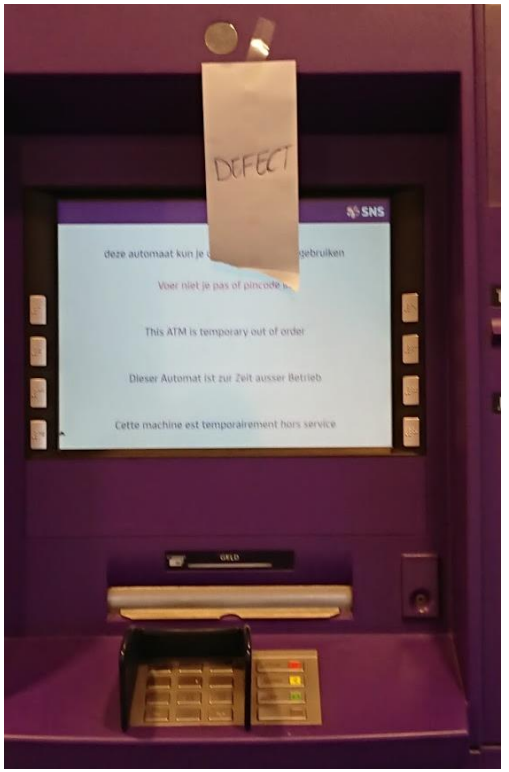
The real state space is 2.45 x 10^{55} times as big



Problems found in bank cards & terminals

We found no expoitable mistakes, but students did find some Denial of Service (DoS) possibilities

- in contactless payment terminals, causing them to crash
- in the first contactless bankcards of two Dutch banks, which enabled access to PIN functionality via contactless interface



Conclusions

- Computing Science is not rocket science:
  it's way more complicated than that
  - Software is *the* most complex artefact engineered by humankind

- What makes computing science special (but tricky)
  1. Software gives amazing power & flexibility.
     But for security this is our Achilles' heel:
     an attacker can exploit this to re-program a hacked device
  2. We are dealing with digital, discrete systems, where
     a tiny change can completely change the behaviour

Some links with first year courses

- **Imperative programming:** How do you program a computer?
- **Processors**: How does a computer execute software to do anything?
- **Hacking in C**: How can attackers abuse this to make a computer do different things than intended?
- **Security**: How can to prevent people getting free electricity at charge pole?
- **Languages & Automata:** How can we describe/model the behaviour of dynamic systems and their input languages?
- **Combinatorics**, **Mathematical Structures**, and **Matrix Calculation:** mathematics to reason logically about discrete structures
- **Information Modelling & Databases:** How can all sorts of information be digitally represented in a computer system?

Thanks for your attention!