

# Cryptographic Engineering

## L01: Introduction

Prof. Nele Mentens

February 5, 2024

# Course overview

Lectures first 6 weeks:

- Feb 5: Course overview + introduction to the course topics
- Feb 12: Side-channel analysis attacks + countermeasures: Part I
- Feb 19: Side-channel analysis attacks + countermeasures: Part II
- Feb 26: Fault analysis attacks + countermeasures
- March 4: Efficient and secure cryptographic implementations: Part I
- March 11: Efficient and secure cryptographic implementations: Part II

# Course overview

- March 18: workshop to discuss Assignment 1: research paper analysis
  - presence is mandatory – at least one group member shows up to discuss the progress
- March 25 + April 1: no lecture
- April 8: presentation session for Assignment 1 (**longer session**)
- April 15 + 22: guided practical sessions to prepare for Assignment 2: hands-on project on physical attacks (**come prepared, instructions will be published in Brightspace!!!**)
  - presence is mandatory to get access to the ChipWhisperer boards that are used for the project
- April 29 + May 6: workshops to discuss Assignment 2
  - presence is mandatory – at least one group member shows up to discuss the progress
- May 13: presentation + demonstration session for Assignment 2 (**longer session**)

# Assignment 1: Research paper analysis

- **Deadlines:**
  - **March 14:** groups and papers approved by the lecturer (communicate by email – the earlier the better!)
  - **April 8:** presentation session (**longer session**)
  - **April 10:** paper submission - upload in Brightspace (one upload per group, mention group members)
- **Groups:** You can work alone or in groups of 2-3 people (teaming up is recommended)
- **Task:** Make a critical analysis and comparison of at least 2 research papers on a selected topic
  - Assignment description with suggested topics and papers will be published in Brightspace (you can also suggest other topics and/or additional papers)
- **Grade Assignment 1:** 50% report grade + 50% presentation grade
  - The lecturer and assistants will ask questions after the presentation, mainly on the studied topic, but possibly also on other related topics that are covered in the lectures

# Additional guidelines on Assignment 1

Guidelines on the report:

- The report you write should be around 4 pages
- It should start with an introduction to the topic and it should end with a conclusion
- It is not enough to summarize what is written in the papers that you analyze. You should compare those papers based on relevant metrics that you derive from the papers, and you should give a critical reflection on the content of the papers

The assessment will be based on the quality of:

- Introduction and conclusion
- Exposition of the research papers analysed
- Exposition and justification of the considered metrics
- Structure, grammar & spelling
- Answers to questions (assessed in presentation only)
- Use of allotted time (assessed in presentation only)
- Attractiveness of the slides (assessed in presentation only)

# Assignment 2: Practical project on physical attacks

- **Deadlines:**
  - **April 18:** groups and projects approved by the lecturer (communicate by email – the earlier the better!)
  - **May 13:** presentation day (**including a demo**) (**longer session**)
  - **May 15:** report submission - upload in Brightspace (one upload per group, mention group members)
- **Groups:** You can work alone or in groups of 2-3 people (teaming up is recommended)
- **Task:** Hands-on project with a ChipWhisperer board
  - 2 guided practical sessions will introduce you to working with the board
  - Assignment description will be published in Brightspace
  - Options:
    - Improve one of the implementations covered in the practical sessions + evaluate
    - Evaluate implementations that were not covered in the practical sessions (no need to implement from scratch, a lot of implementations are published online)
    - Apply an analysis approach not covered in the practical sessions to implementations covered in the practical sessions
    - ...
  - You are expected to hand in a Jupyter Notebook that allows us to recreate your work
- **Grade Assignment 2:** 50% project work grade + 20% report grade + 30% presentation grade

# Additional guidelines on Assignment 2

The assessment will be based on:

- 50% project work
  - difficulty of the project
  - practical approach
  - achieved results
- 30% presentation
  - introduction and motivation of the topic
  - structure of the presentation
  - practical demonstration
- 20% report
  - description of the project work and results
  - critical analysis of the results
  - writing quality

There is no upper or lower bound for the number of pages in the report, but we suggest to have around 4 pages.

# Course grade

**Course grade** = 30% research paper analysis grade + 70% project grade



# Literature and course material

## Lectures:

- ❖ Slides will be published in Brightspace
- ❖ No textbook; this is a research-oriented course
  - Ask the lecturer for literature recommendations about specific topics that you are interested in

## Practical sessions + project:

- ❖ ChipWhisperer-Lite ARM containing the STM32F3 processor
  - Datasheet: [https://www.mouser.com/datasheet/2/894/NAE-CW1173\\_datasheet-1289272.pdf](https://www.mouser.com/datasheet/2/894/NAE-CW1173_datasheet-1289272.pdf)

```
each: function(e, t, n) {  
  var r, i = 0;  
  o = e.length;  
  a = M(e);  
  if (n) {  
    if (a) {  
      for (; o > i; i++)  
        if (r = t.apply(e[i], n), r !== !1) break  
    } else  
      for (i in e)  
        if (r = t.apply(e[i], n), r !== !1) break  
  } else if (a) {  
    for (; o > i; i++)
```

# Introduction to the course topics

```
  return e  
},  
trim: b && !b.call("\uffeff\u00a0")  
  return null == e ? "" : b.call  
} : function(e) {  
  return null == e ? "" : (e + "").replace(C, "")  
},  
makeArray: function(e, t) {  
  var n = t || [];  
  return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n  
},  
isArray: function(e, t, n) {  
  var r;  
  if (t) {  
    if (n) return m.call(t, e, n);  
    for (n = t.length, n = n > 0 && n < Math.max(8, r + n) : n : 8; r > n; r++)  
      if (n in t && t[n] === e) return n  
  }  
}
```

# Outline

- Secure embedded systems and IoT networks
- Attacks on (embedded) systems
  - Lecture 2+3+4
- Efficient and secure cryptographic implementations
  - Lecture 5+6

```
each: function(e, t, n) {  
  var r, i = 0;  
  o = e.length;  
  a = M(e);  
  if (n) {  
    if (a) {  
      for (; o > i; i++)  
        if (r = t.apply(e[i], n), r === !1) break  
    } else  
      for (i in e)  
        if (r = t.apply(e[i], n), r === !1) break  
  } else if (a) {  
    for (; o > i; i++)
```

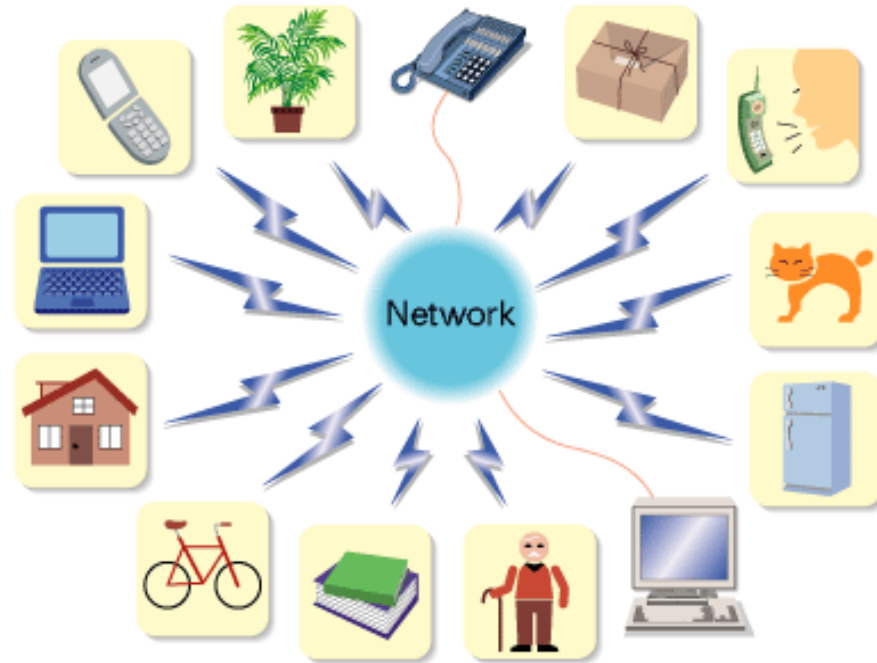
# Secure embedded systems and IoT networks

```
  return e  
},  
trim: b && !b.c  
return null  
} : function(e)  
  return null == e ? "" : (e + "").replace(C, "")  
},  
makeArray: function(e, t) {  
  var n = t || [];  
  return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e)), n  
},  
isArray: function(e, t, n) {  
  var r;  
  if (t) {  
    if (n) return m.call(t, e, n);  
    for (n = t.length, n = n > 0 && Math.max(0, r + n) : n : 0; r > n; r++)  
      if (n in t && t[n] === e) return n  
  }  
}
```

# Embedded systems

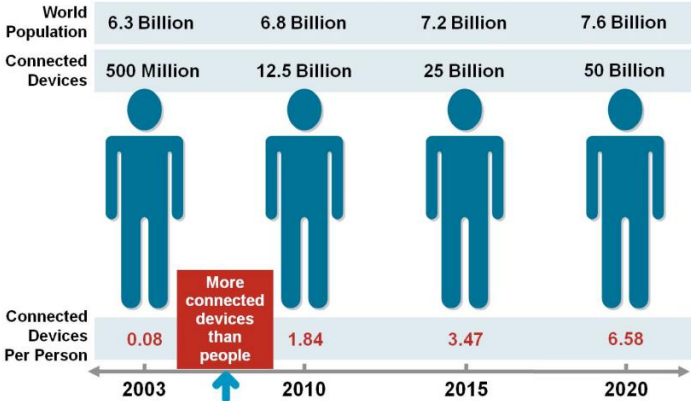
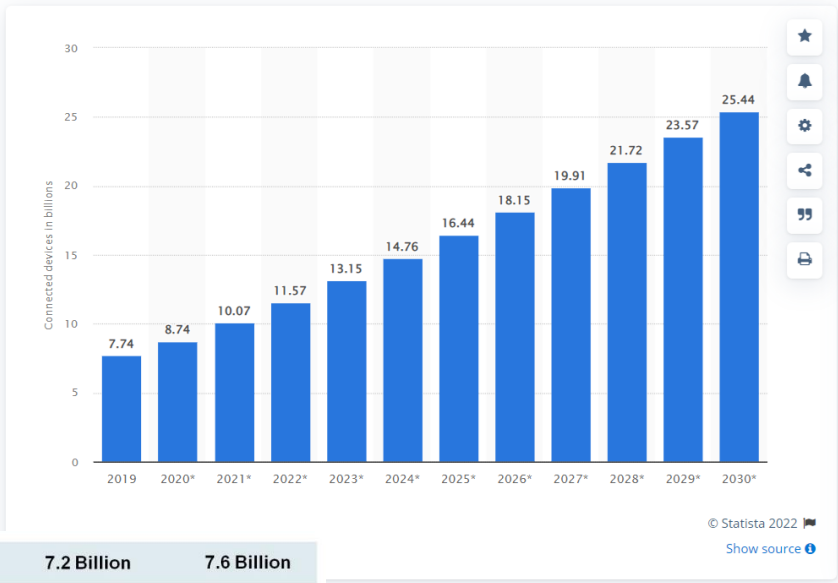
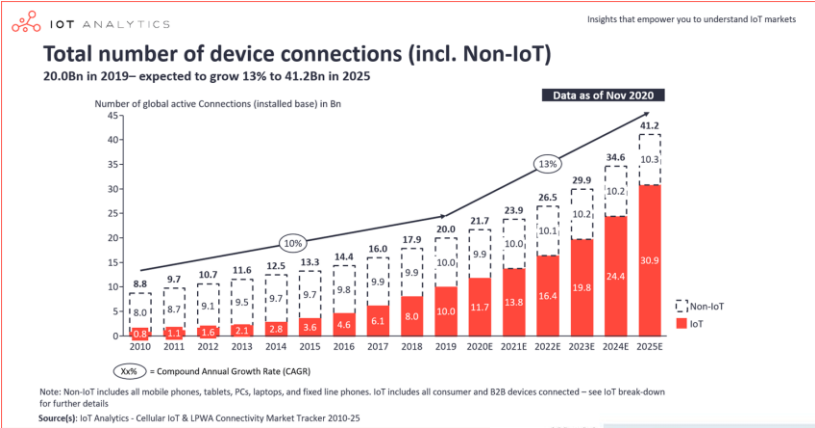
- An embedded system is a computer system that has a dedicated function within a larger mechanical or electrical system.
- It is embedded as part of a complete device, often including electrical or electronic hardware and mechanical parts.
- It is often constrained with respect to energy consumption, operating speed, cost,...
- Traditional laptops, desktops, servers,... are NOT embedded systems

# Internet of Things (IoT)



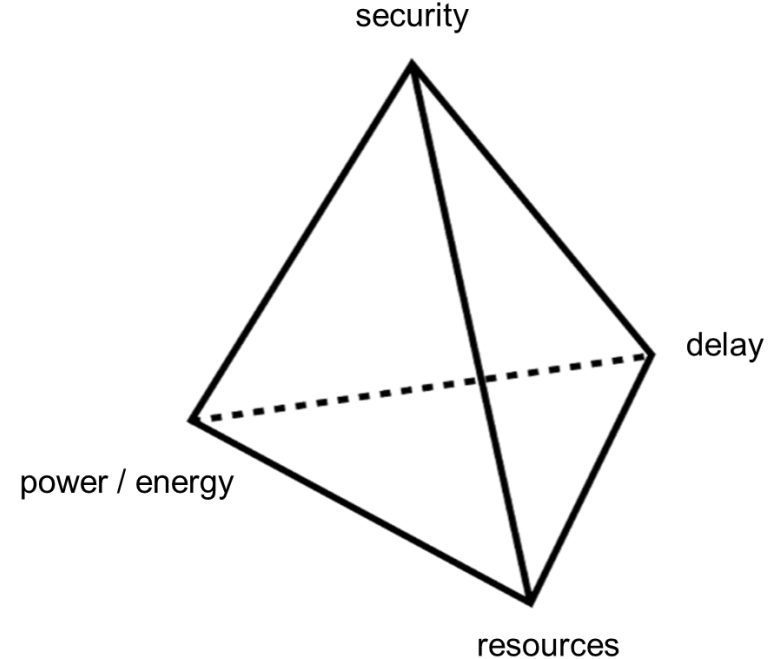


# Connected devices



# Secure systems

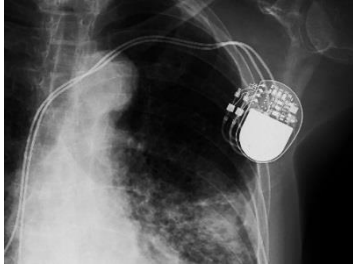
- Data security is not the core functionality of a system
- Data security needs to be added with a minimal overhead in delay, resources (cost), power/energy consumption
- One aspect of (embedded) security is to equip electronic systems with appropriate implementations of cryptographic algorithms and protocols → this course





# Design goals

- Different applications have different requirements



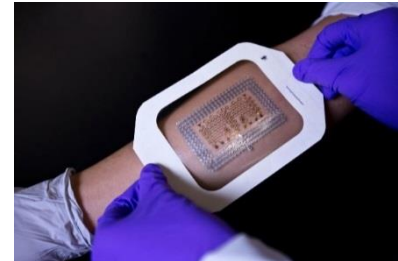
low energy – pacemaker



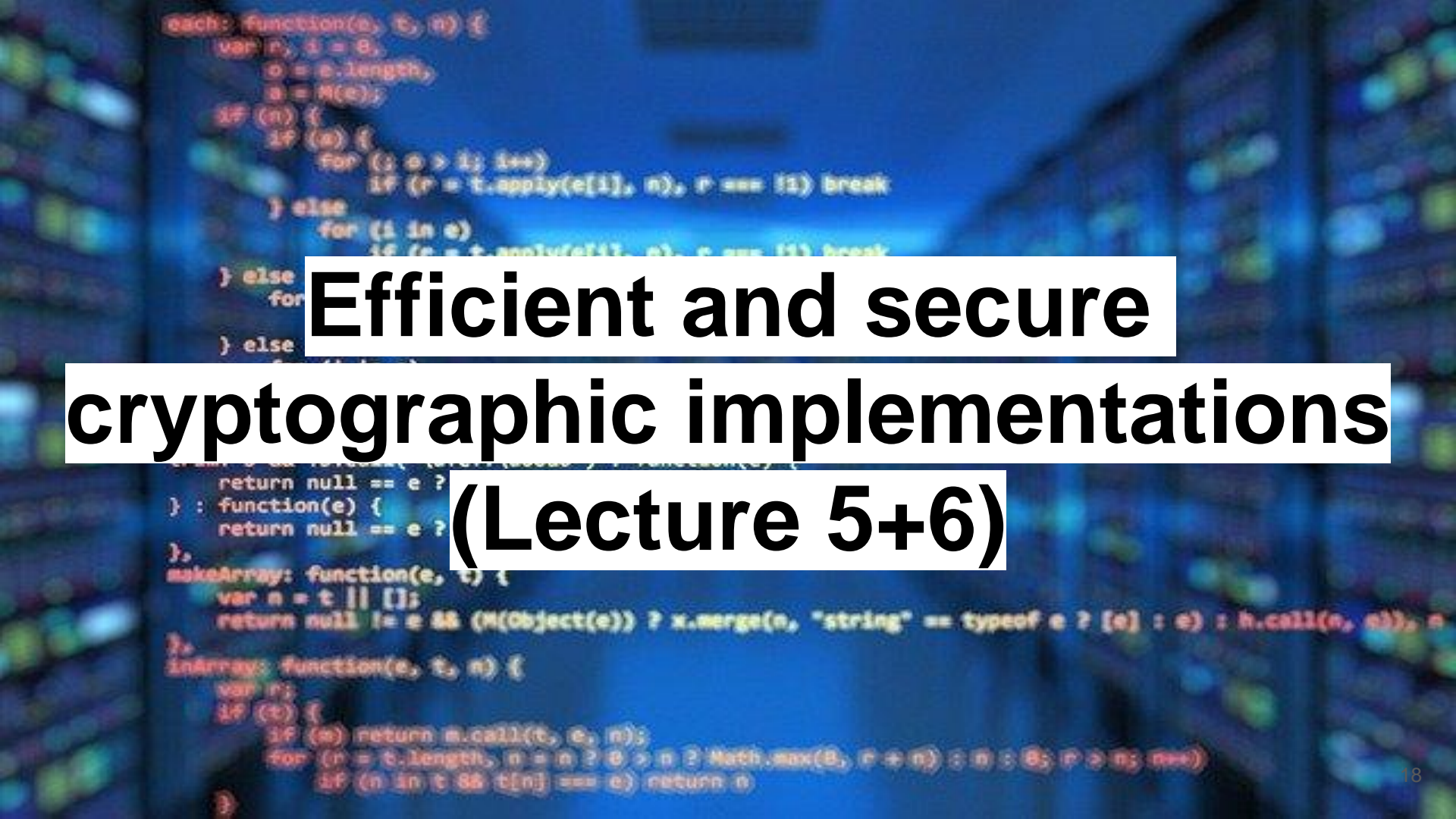
low power – RFID access control



high performance – video conferencing



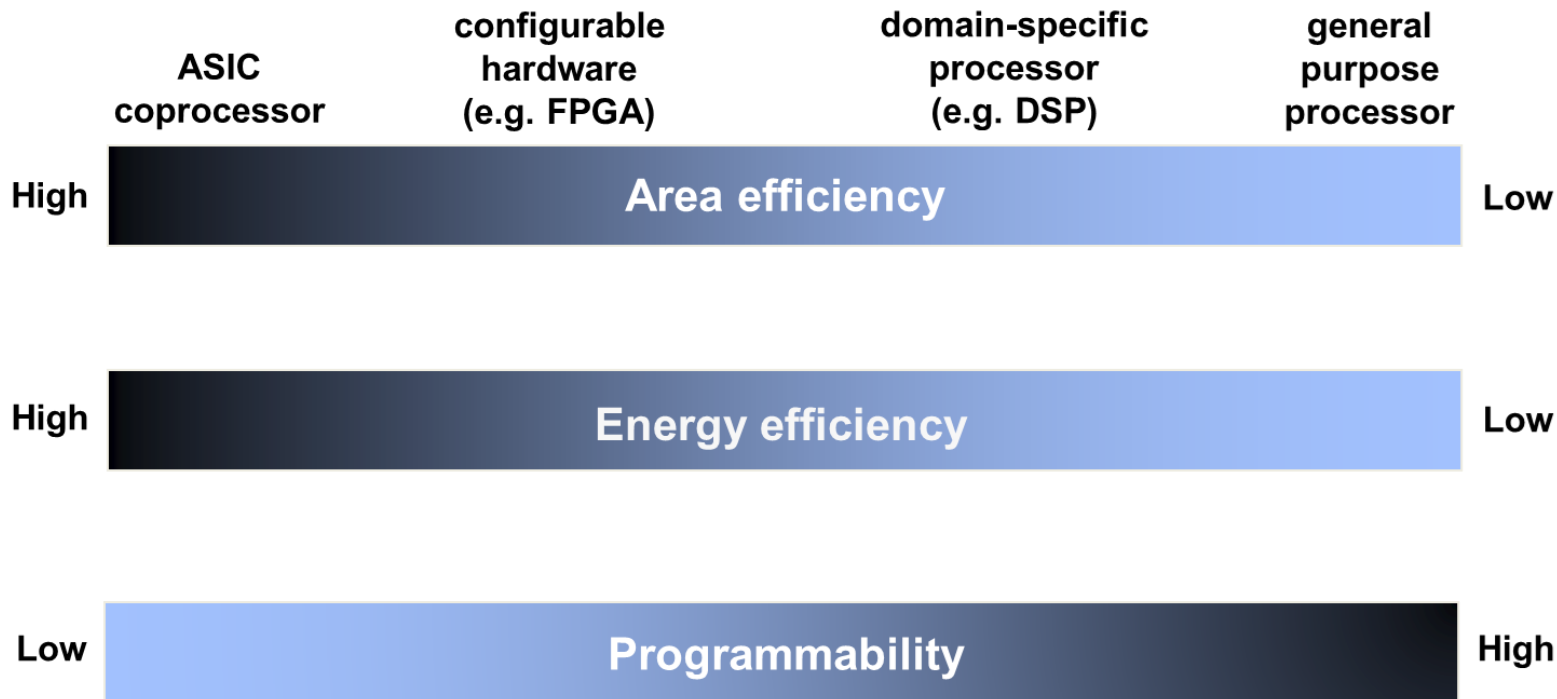
low cost – disposable medical sensors



```
each: function(e, t, n) {  
  var r, i = 0;  
  o = e.length;  
  a = M(e);  
  if (n) {  
    if (a) {  
      for (; o > i; i++)  
        if (r = t.apply(e[i], n), r === !1) break  
    } else  
      for (i in e)  
        if (r = t.apply(e[i], n), r === !1) break  
  } else  
    for  
  } else  
  }  
}  
  
return null == e ?  
} : function(e) {  
  return null == e ?  
},  
makeArray: function(e, t) {  
  var n = t || [];  
  return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e), n  
},  
isArray: function(e, t, n) {  
  var r;  
  if (t) {  
    if (n) return m.call(t, e, n);  
    for (n = t.length, n = n > 0 & n < Math.max(8, r + n) : n : 8; r > n; r++)  
      if (n in t && t[n] === e) return n  
  }  
}
```

# Efficient and secure cryptographic implementations (Lecture 5+6)

# Implementation platforms



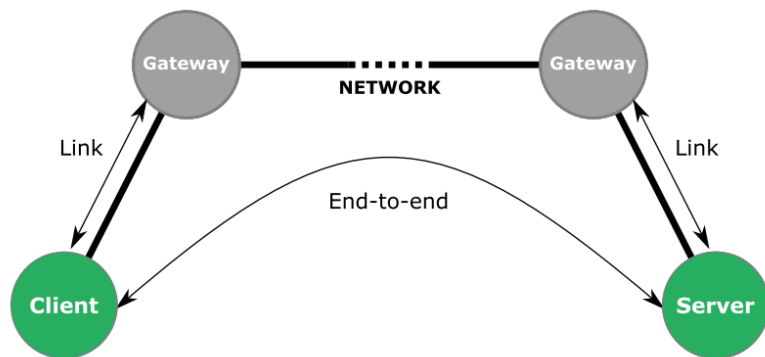
# Note on computation vs. communication overhead

- It's not only the (cryptographic) computation that counts
- Sometimes optimizing the cryptographic implementation has a negligible effect on the efficiency (performance, energy consumption) of the system
- Depending on the deployed implementation platform, the (wireless) communication standard, and the cryptographic algorithms and protocols, either the computation or the communication overhead dominates (see case study on the next slides)
- This course: we mainly look at the computation overhead of cryptographic algorithms and protocols

# Case study: computation vs. communication energy

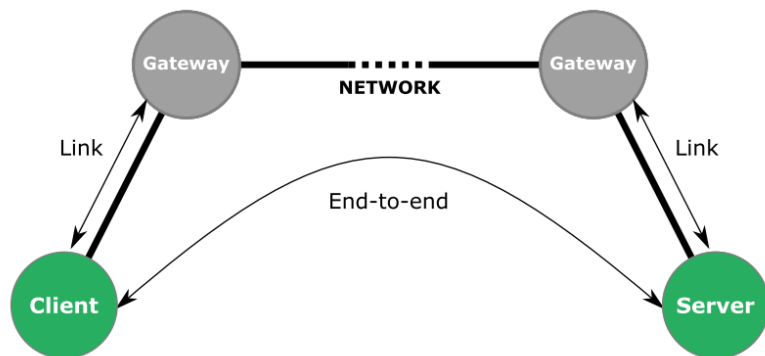
- 3 different cryptographic protocols:
  - PSK: symmetric-key crypto through a pre-shared and pre-installed key on both entities
  - PPSK: public-key crypto through a pre-installed public key of the remote entity
  - PSTTPPK: pre-shared trusted third-party public-key, used to verify the certificate that links the public key of the remote entity to that entity
- 3 different processor platforms – 3 different wireless communication standards:
  - MAX32620 (96 MHz, 2000 KB Flash, 256 KB RAM) – Bluetooth Low Energy (BLE)
  - MSP432-P401R (48 MHz, 256 KB Flash, 64 KB RAM) – WiFi
  - STM32-L073RZT6 (32 MHz, 192 KB Flash, 20 KB RAM) – Long Range Wide Area Network (LoRaWAN)

# Case study: computation vs. communication energy



| Configuration           | PSK   | PSPK  | PSTTPPK |
|-------------------------|-------|-------|---------|
| <i>nuc + LoRa (SF7)</i> |       |       |         |
| Total ( $\mu J$ )       | 3119  | 62345 | 133845  |
| Comp (%)                | 2.9%  | 75.6% | 50.2%   |
| Comm (%)                | 97.1% | 24.4% | 49.8%   |
| <i>msp + WiFi</i>       |       |       |         |
| Total ( $\mu J$ )       | 66    | 13257 | 19185   |
| Comp (%)                | 74.8% | 99.4% | 98.1%   |
| Comm (%)                | 25.2% | 0.6%  | 1.9%    |
| <i>max + BLE (4.1)</i>  |       |       |         |
| Total ( $\mu J$ )       | 68    | 3414  | 5654    |
| Comp (%)                | 22.3% | 92.3% | 79.6%   |
| Comm (%)                | 77.7% | 7.7%  | 20.4%   |

# Case study: computation vs. communication energy



| Configuration           | PSK   | PSPK  | PSTTPPK |
|-------------------------|-------|-------|---------|
| <i>nuc + LoRa (SF7)</i> |       |       |         |
| Total ( $\mu J$ )       | 3119  | 62345 | 133845  |
| Comp (%)                | 2.9%  | 75.6% | 50.2%   |
| Comm (%)                | 97.1% | 24.4% | 49.8%   |
| <i>msp + WiFi</i>       |       |       |         |
| Total ( $\mu J$ )       | 66    | 13257 | 19185   |
| Comp (%)                | 74.8% | 99.4% | 98.1%   |
| Comm (%)                | 25.2% | 0.6%  | 1.9%    |
| <i>max + BLE (4.1)</i>  |       |       |         |
| Total ( $\mu J$ )       | 68    | 3414  | 5654    |
| Comp (%)                | 22.3% | 92.3% | 79.6%   |
| Comm (%)                | 77.7% | 7.7%  | 20.4%   |



```

each: function(e, t, n) {
  var r, i = 0;
  o = e.length;
  a = M(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r === !1) break;
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
      for (; o > i; i++)

```

# Attacks on (embedded) systems

## (Lecture 2+3+4)

```

    return e
  },
  trim: b && !b.call(
    return null ==
  } : function(e) {
    return null == e ? "" : (e + "").replace(C, "")
  },
  makeArray: function(e, t) {
    var n = t || [];
    return null != e && (M(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : h.call(n, e), n
  },
  isArray: function(e, t, n) {
    var r;
    if (t) {
      if (n) return m.call(t, e, n);
      for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n) : n : 0; r > n; n++)
        if (n in t && t[n] === e) return n
    }
  }
}

```



# Main attack categories

1. Attacks on cryptographic algorithms and protocols
2. Attacks on the system and the network
3. Attacks on the implementation of algorithms and protocols

# Example 1: Tesla hack

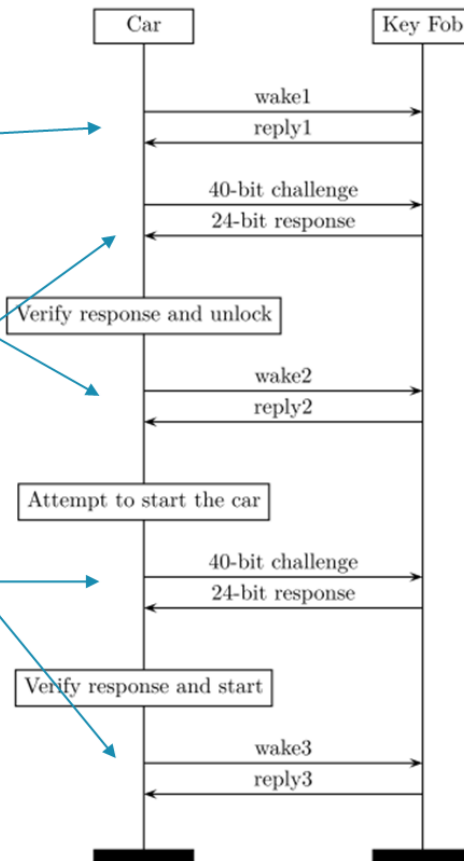
- [https://www.youtube.com/watch?time\\_continue=2&v=aVIYuPzmJoY&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=2&v=aVIYuPzmJoY&feature=emb_logo)



# Tesla hack explained



Known combinations,  
no crypto involved

Compression function  
with a 40-bit key



# Tesla hack explained

## What are the vulnerabilities?

- Proprietary crypto algorithm is used to do the compression
  - Security by obscurity 
- A 40-bit key is used
  - Brute-force attack can successfully be mounted 

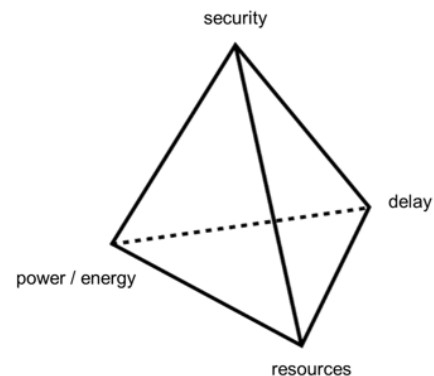
# Tesla hack explained

## Main attack categories

1. Attacks on cryptographic algorithms and protocols
2. Attacks on the system and the network
3. Attacks on the implementation of algorithms and protocols

## Secure systems

- Data security is not the core functionality of a system
- Data security needs to be added with a minimal overhead in delay, resources (cost), power/energy consumption
- One aspect of (embedded) security is to equip electronic systems with appropriate implementations of cryptographic algorithms and protocols → this course



More info on crypto algorithms  
and key lengths in the  
**Bachelor course on Security**

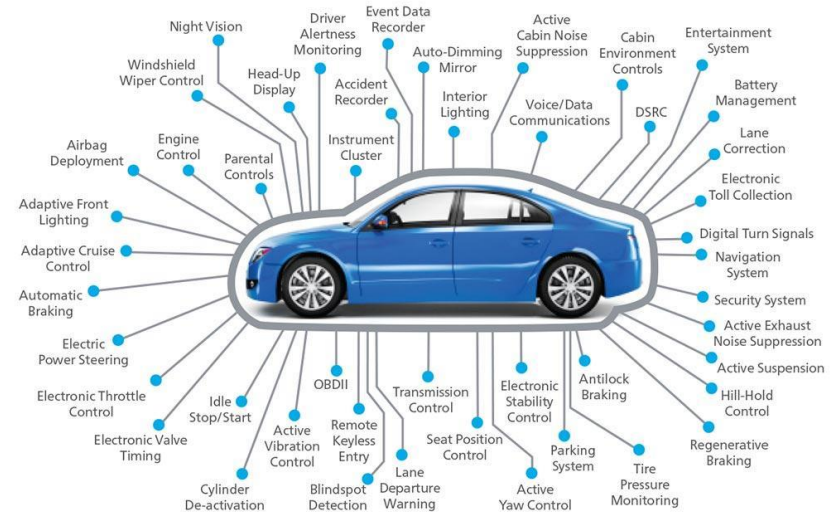
## Example 2: Remote jeep hack

ANDY GREENBERG SECURITY 03.17.16 6:58 PM

### THE FBI WARNS THAT CAR HACKING IS A REAL RISK



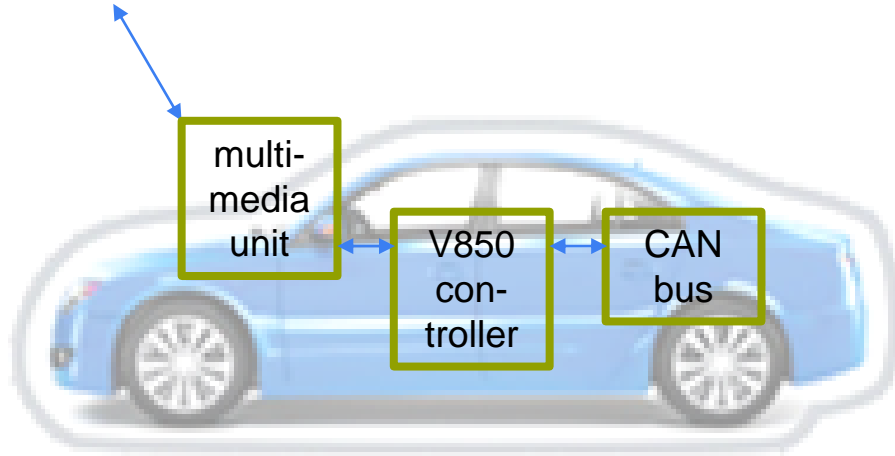
ANDY GREENBERG/WIRED



- <https://www.youtube.com/watch?v=MK0SrxBC1xs>

# Remote jeep hack explained

Wi-Fi or cellular



- Get access to multimedia unit through Wi-Fi or cellular network (through insecure password)
- Get access to V850 controller through multimedia unit
- Install malware on V850 controller such that it can control the CAN (Controller Area Network) bus - V850 controller is only supposed to read from CAN bus
- Get access to breaks, wipers, engine,... through the CAN bus

# Remote jeep hack explained

## Main attack categories

1. Attacks on cryptographic algorithms and protocols
2. Attacks on the system and the network
3. Attacks on the implementation of algorithms and protocols

More info on trusted computing  
in **Master course on System  
& Software Security**



# Example 3: Pacemaker hack


- <https://www.youtube.com/watch?v=BDtr5vixC7E>

## Security

### Fatal flaws in ten pacemakers make for Denial of Life attacks

Brit/Belgian research team decipher signals and devise wounding wireless attacks

By [Darren Pauli](#) 1 Dec 2016 at 06:02

20  SHARE ▼

A global research team has hacked 10 different types of implantable medical devices and pacemakers finding exploits that could allow wireless remote attackers to kill victims.

Eduard Marin and Dave Singelée, researchers with KU Leuven University, Belgium, began examining the pacemakers under black box testing conditions in which they had no prior knowledge or special access to the devices, and used commercial off-the-shelf equipment to break the proprietary communications protocols.

From the position of blind attackers the pair managed to hack pacemakers from up to five metres away gaining the ability to deliver fatal shocks and turn off life-saving treatment.

# Pacemaker hack explained

Combination of vulnerabilities:

- Privacy attacks: data exchanged between pacemaker and programmer is protected based on an XOR with a fixed value (always the same value)
  - This allows revealing the patient's name, treatment,... or tracking the patient by placing beacons in strategic places

# Pacemaker hack explained

Combination of vulnerabilities:

- Privacy attacks: data exchanged between pacemaker and programmer is protected based on an XOR with a fixed value (always the same value)
  - This allows revealing the patient's name, treatment,... or tracking the patient by placing beacons in strategic places
- Denial-of-Service (DoS) attacks: instead of directly going into sleep mode, the device listens for new incoming requests for five minutes after deactivation
  - This allows to drain the battery by continuously sending requests  
→ Denial-of-Life attack!

# Pacemaker hack explained

Combination of vulnerabilities:

- Privacy attacks: data exchanged between pacemaker and programmer is protected based on an XOR with a fixed value (always the same value)
  - This allows revealing the patient's name, treatment,... or tracking the patient by placing beacons in strategic places
- Denial-of-Service (DoS) attacks: instead of directly going into sleep mode, the device listens for new incoming requests for five minutes after deactivation
  - This allows to drain the battery by continuously sending requests  
→ Denial-of-Life attack!
- Replay attacks: no freshness is included in the messages
  - This allows to resend a previously transmitted message

# Pacemaker hack explained

Combination of vulnerabilities:

- Privacy attacks: data exchanged between pacemaker and programmer is protected based on an XOR with a fixed value (always the same value)
  - This allows revealing the patient's name, treatment,... or tracking the patient by placing beacons in strategic places
- Denial-of-Service (DoS) attacks: instead of directly going into sleep mode, the device listens for new incoming requests for five minutes after deactivation
  - This allows to drain the battery by continuously sending requests  
→ Denial-of-Life attack!
- Replay attacks: no freshness is included in the messages
  - This allows to resend a previously transmitted message
- Spoofing attacks: no authentication between the entities
  - This allows to send arbitrary commands to the pacemaker

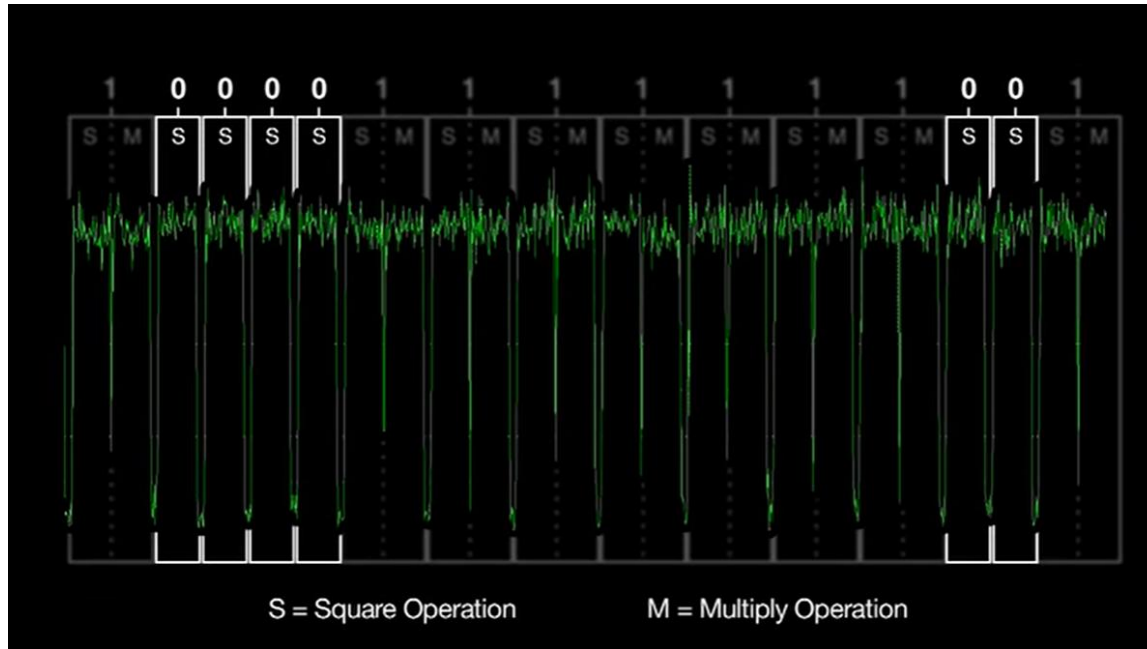
# Pacemaker hack explained

## Main attack categories

1. Attacks on cryptographic algorithms and protocols
2. Attacks on the system and the network
3. Attacks on the implementation of algorithms and protocols

## Example 4: electromagnetic side-channel analysis

- <https://www.youtube.com/watch?v=cPDDNVKo43w>



# Electromagnetic side-channel analysis explained

## Main attack categories

1. Attacks on cryptographic algorithms and protocols
2. Attacks on the system and the network
3. Attacks on the implementation of algorithms and protocols

More info on physical attacks in **Lecture 2+3+4 of this course!**