Féidearthachtaí as Cuimse
Infinite Possibilities

# Week 5
# Security

Fundamentals of IoT

Dr. Eoin Rogers (eoin.rogers@tudublin.ie)

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

# Lesson Outline

- Why is security important?
- How can we implement secure IoT devices
- Emphasis on cryptography and other common-sense approaches to dealing with security

# Security in IoT devices

- Security is an essential part of modern networking, but is unfortunately often overlooked by IoT vendors!
- This has lead to major security breaches:
  - Ring cameras
  - Attacks on St. Jude’s cardiac devices
  - Mirai botnet

# Basic security practices

By following certain practices, we can increase the security of our devices

# Common IoT security issues

- Default passwords
- Single level of access control
- Large attack surfaces
- Old software

- Lack of encryption
- Trusted computing
- Poor responses from vendors when intrusions are found

# Default passwords

- When a user first logs in, **make them change the password**!
- Enable **two-factor authentication** by default
  - Do allow users to disable it, but an important security principle is to **make the defaults secure** whenever possible!

# Single level of access control

- Seperate **normal** and **administrative** users
  - This can be challenging, because IoT devices often have **limited functionality**!
- Implement stricter security for admin functions and users

# Minimise attack surfaces

- Present **fewer targets** that an attacker can exploit
  - Do you really need 20 ports open on your IoT device?
  - Do you really need to run a full operating system?
- You could argue that IoT as a whole is a violation of this principle!

# Old software

- Old software is often filled with **vulnerabilities**
- Only use old software if you are sure it is safe!
- “My device is low-end and constrained” is not a very good excuse!
- **Don’t re-invent the wheel** – writing your own security code makes your system more vulnerable, not less!

# Encryption

- Try to encrypt anything that could potentially be sensitive or private
  - Maybe even data that isn't private, in order to protect against **clever side channels or network analysis**!
- Three types of encryption: **symmetric**, **asymmetric** and **hashing**

# Encryption

Actually, encryption is important enough that we should spend some time talking about it!

# Symmetric ciphers

- Sender and reciever need to **share the same key in advance of communication taking place**
- **Same key** used to encrypt and decrypt
- Examples:
  - AES/Rijndael
  - Blowfish and Twofish

# Asymmetric ciphers

- Two keys exist: a **public key** used to encrypt data, and a **private key** to decrypt it
- The receiver generates the two keys, and **only makes the public key public**. It is useless for decrypting
- Examples:
  - RSA
  - Eliptic curve algorithms

# RSA overview

- Generate two (large) prime numbers, **p** and **q**
- **n** = p * q
- **λ** = lcm(p – 1, q – 1)
- Pick a number **e** between 1 and λ such that the gcd(e, λ) = 1
- Pick a number **d** such that (e * d) % n == 1
- Use n and e as a **public key**, and n and d as a **private key**

# RSA usage

- To **encrypt** the plaintext **p**:
  - c = pow(p, e) % n
- To **decrypt** the ciphertext **c**:
  - p = pow(c, d) % n

# Hashing

- Basically **irreversible encryption** – we can go from plaintext to ciphertext, but not the reverse
- Commonly used to store passwords
- Examples:
  - SHA-2 (128/256/512)
  - SHA-3 (256/384/512)

# Encryption on the Pico

Yes, there is a library that implements AES!

# Modes of operation

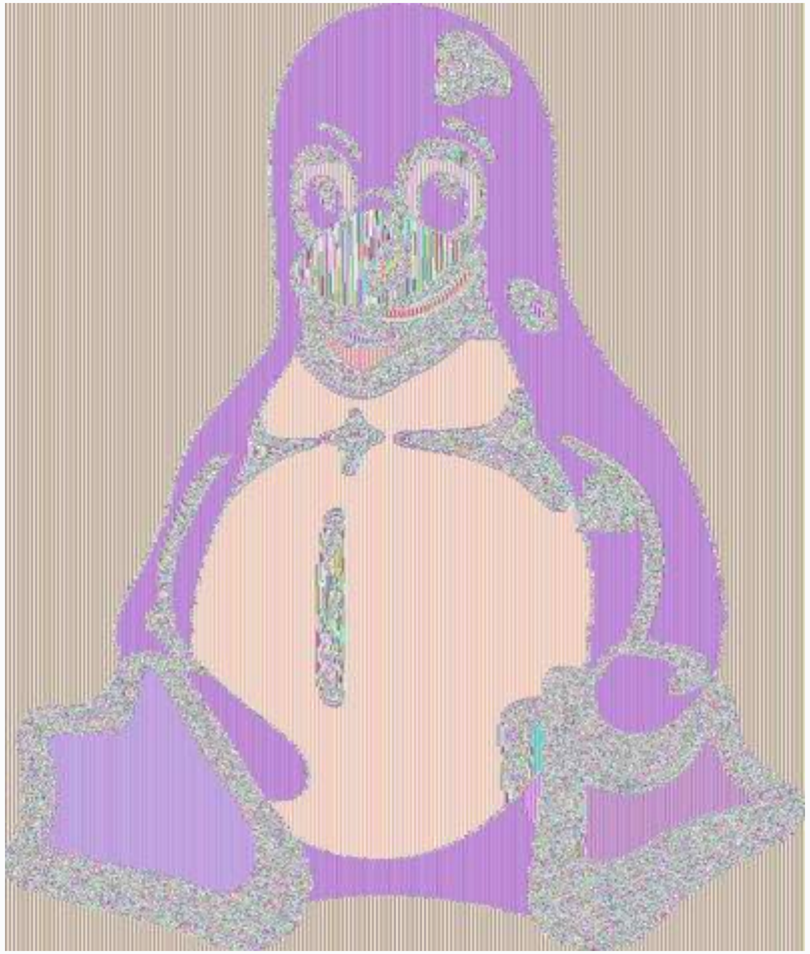
- One complexity in the real world: if we use a block cipher with a block size of 256 bits, **how do we encrypt more than 256 bits of data?**
  - You would assume we pad it to have a length which is a multiple of 256 bits and split it into blocks
  - Do people think this will work?

# Modes of operation

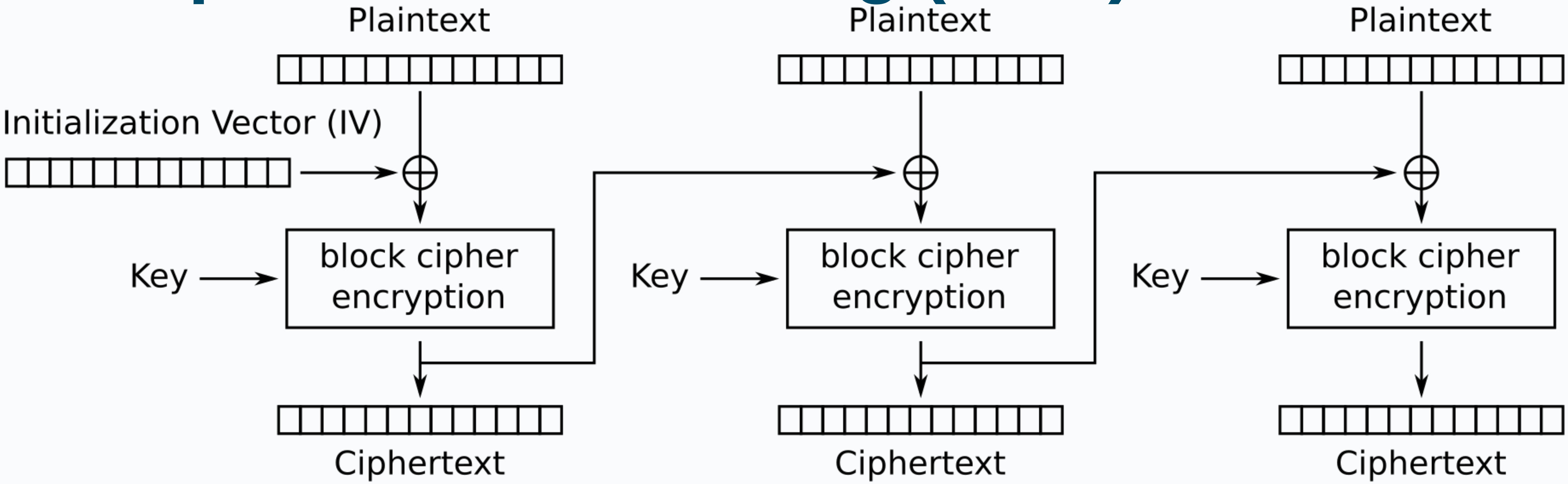
- One complexity in the real world: if we use a block cipher with a block size of 256 bits, **how do we encrypt more than 256 bits of data?**
  - You would assume we pad it to have a length which is a multiple of 256 bits and split it into blocks (ECB mode)
  - Do people think this will work?

# Electronic code book doesn’t work!

Using ECB can **leak** a lot of information about the underlying data!



# Cipher-block chaining (CBC)




Cipher Block Chaining (CBC) mode encryption

# AES in MicroPython

```python
import cryptolib

iv = b'hey!'
key = b'secret!'
data = b'Hello, World!'

def pad_128(data):
    output = data[:]
    while len(output) < 16:
        output += data

    if len(output) == 16:
        return output

    return output[:-(len(output) % 16)]
```

```python
padded_key = pad_128(key)
padded_iv = pad_128(iv)
padded_data = pad_128(data)

# The 2 means we want to use CBC mode
cipher = cryptolib.aes(padded_key, 2, padded_iv)

ciphertext = cipher.encrypt(padded_data)

cipher = cryptolib.aes(padded_key, 2, padded_iv)
plaintext = cipher.decrypt(ciphertext)

print(ciphertext)
print(plaintext)
```

# Back to the main lecture

That’s enough talk about encryption!

# Trusted computing

- Use cryptographic functions to **ensure that firmware hasn't been tampered or modified**
  - Somewhat controversial!
- **Secure Boot** on PCs – a similar system called the **Device Identifier Composition Engine** (**DICE**) is used for IoT

# Vendor responses to security incidents

- How it is **supposed** to work:
  - Have a **plan** in place detailing how to respond to security breaches
  - **Monitor network traffic** to detect suspicious activity!
  - **Inform customers** when a breach occurs
  - **Update device firmware**
  - Plan for **process changes**, deal with **human factors**!

# Summary

- Security is a common issue in IoT devices, so it is something that needs special attention
- There is no such thing as 100% secure, but there is such a thing as not worth dealing with (attackers follow the path of least resistance)
- Use proper encryption!

That’s all for this week

Thanks for your attention!