Secure Messaging, Key Management & Device Identity for the IoT

Andrew John Poulter
Steven Johnston
Simon Cox

IoT Security Foundation Conference
26th November 2019

About this work

- The Defence Science and Technology Laboratory (Dstl) is an executive agency of the United Kingdom Ministry of Defence (MOD); our role is to ensure that innovative science and technology contribute to the defence and security of the UK
- AJ Poulter is a principal computer scientist with Dstl; and the lead engineer for the Dstl AI Lab
- The work being presented here is output from PhD research conducted in conjunction with the University of Southampton, and is sponsored by Dstl. As such. **this presentation refers to research work; and nothing about current or future MOD policy should be inferred from this presentation**

“The S in IoT stands for security…”

- IoT security failings are (very) well publicized and documented…
- The security threat is high – both to domestic & commercial devices; as well as industrial, infrastructure, and even higher when deployed on military devices…
- Security is hard – typically application developers don’t think like security professionals & they **shouldn’t need to**…

https://www.flickr.com/photos/devdsp/6999839463/

The threat…

- In contrast to more traditional cybersecurity, the threat is not just to the data…
- Potential attack targets:
  - Data
    - Interception of sensitive data
    - Spoof or modify data messages…
  - Device
    - Damage (destroy) the physical device
    - Deny or Degrade service
    - Device being co-opted into a botnet
  - Network
    - As gateway to attack local network

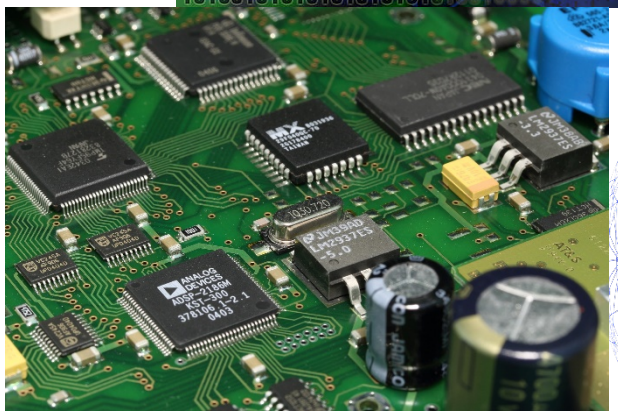


Image by olafpictures from Pixabay

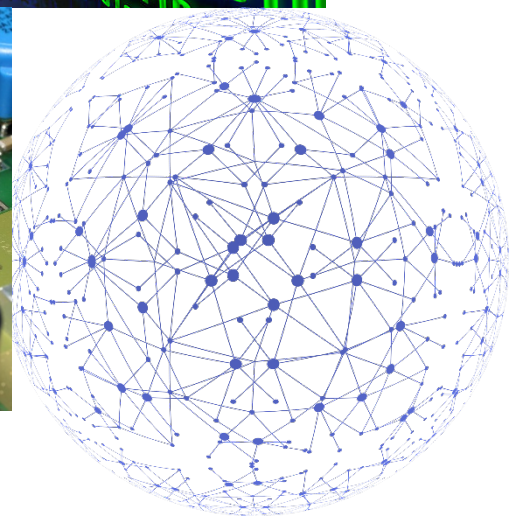
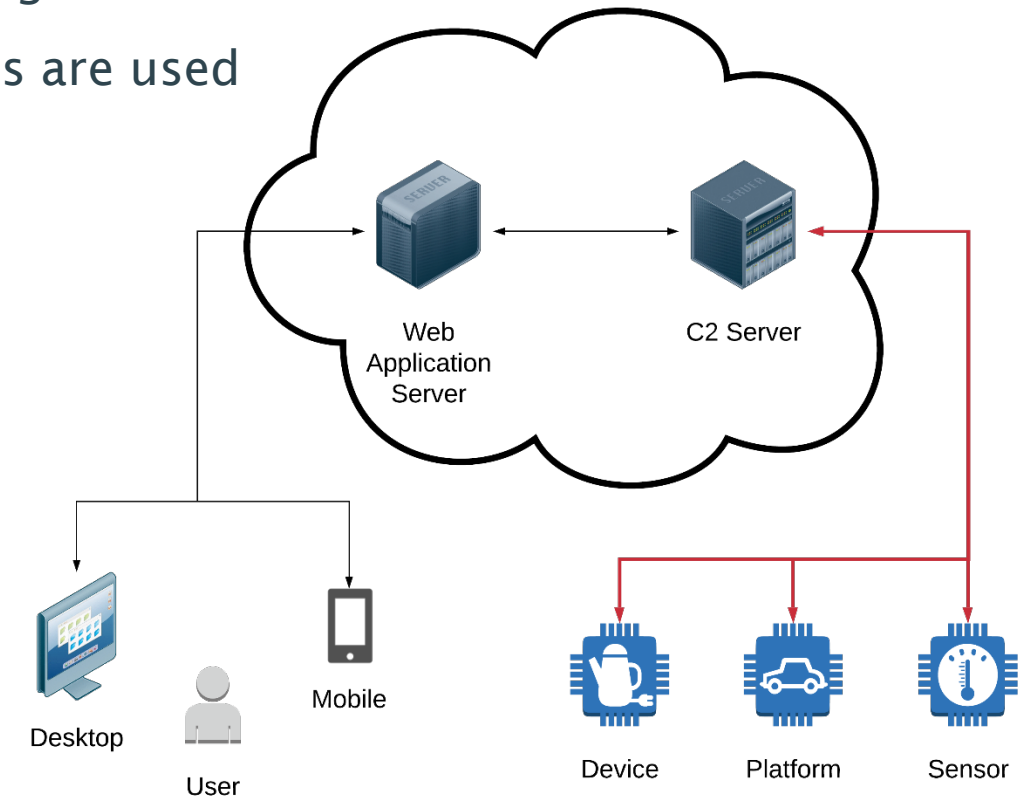


Image by Gerd Altmann from Pixabay

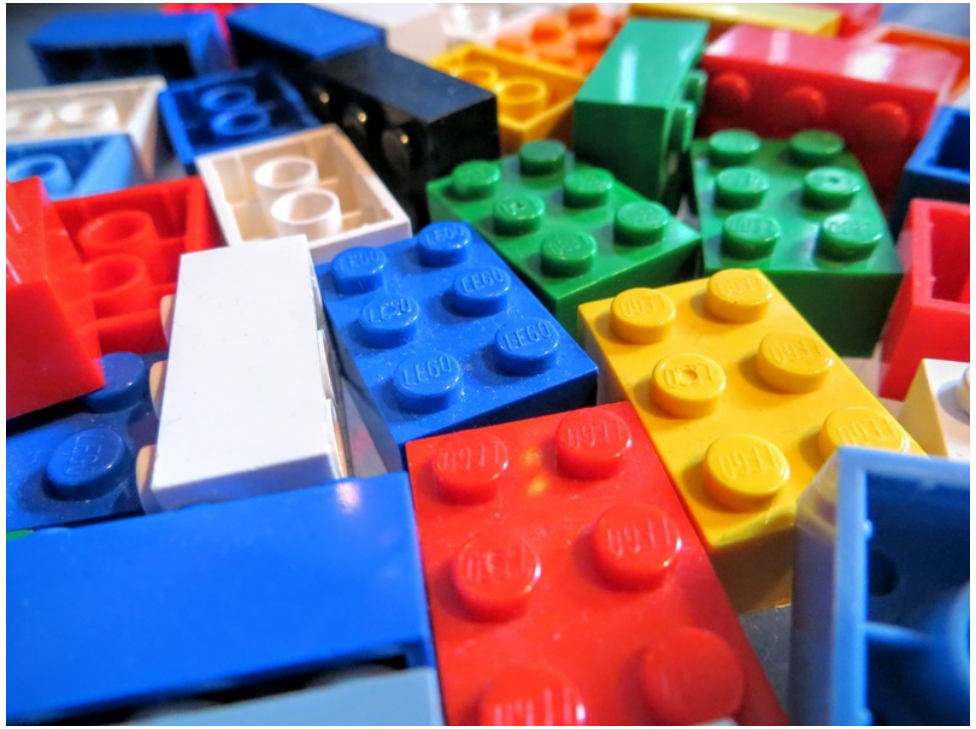
C2 Oriented Approach to IoT

- The research work has been conducted with a background assumption of a fundamentally *centralized* Command-and-Control (C2) IoT system
  - This obviously reflects a military way of thinking…
  - It also reflects how many *real-world* IoT devices are used

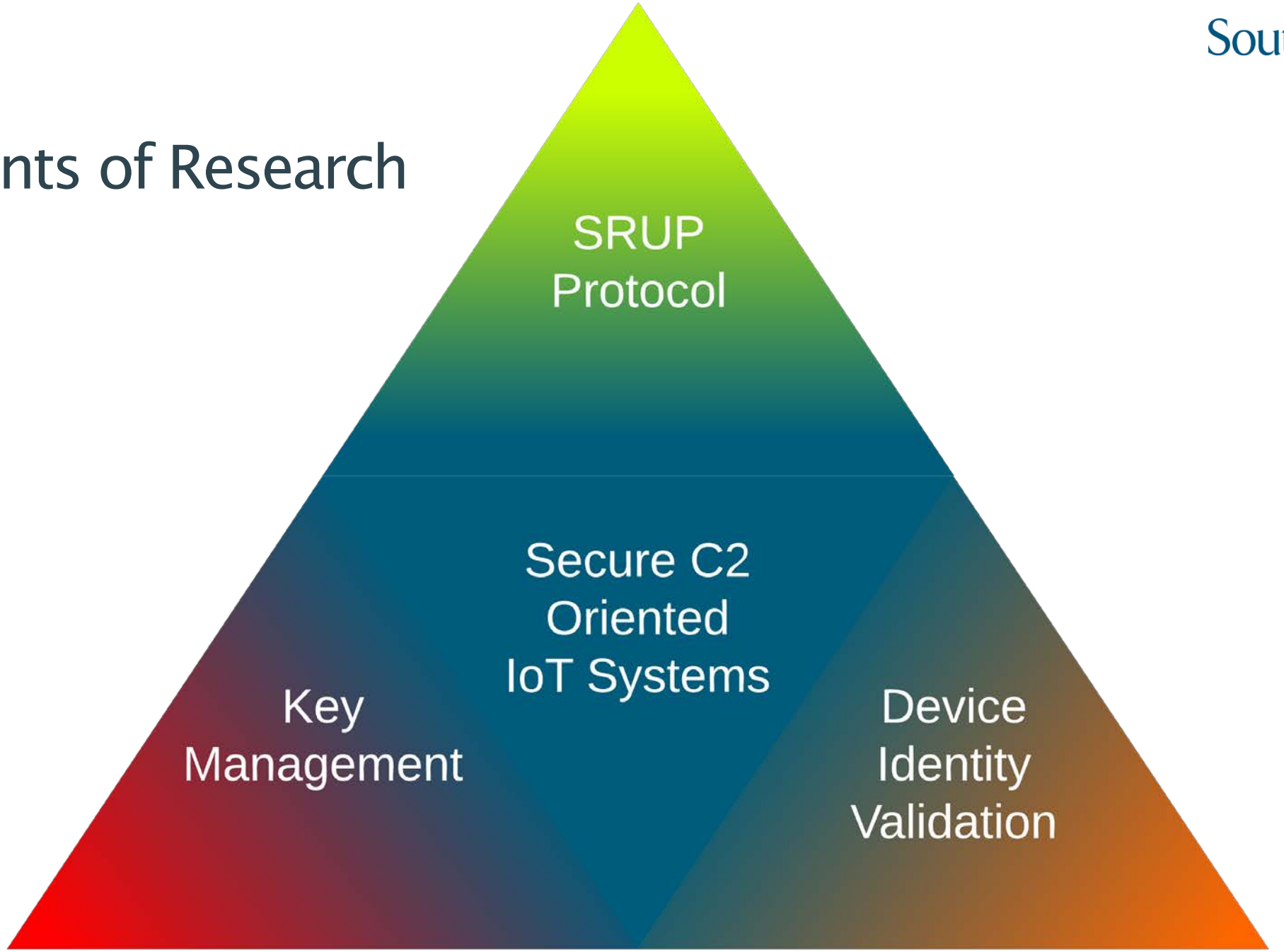


Commodity Software Components

- The idea of building hardware and software commodity components isn't new…
  - HPC Clusters…
  - Python, JavaScript, R, Go, Perl, etc…
- This work is built on the same conceptual idea generalized for a systems context
- The objective was to build something that reuses existing software components wherever possible; and to provide a component solution that others can utilize

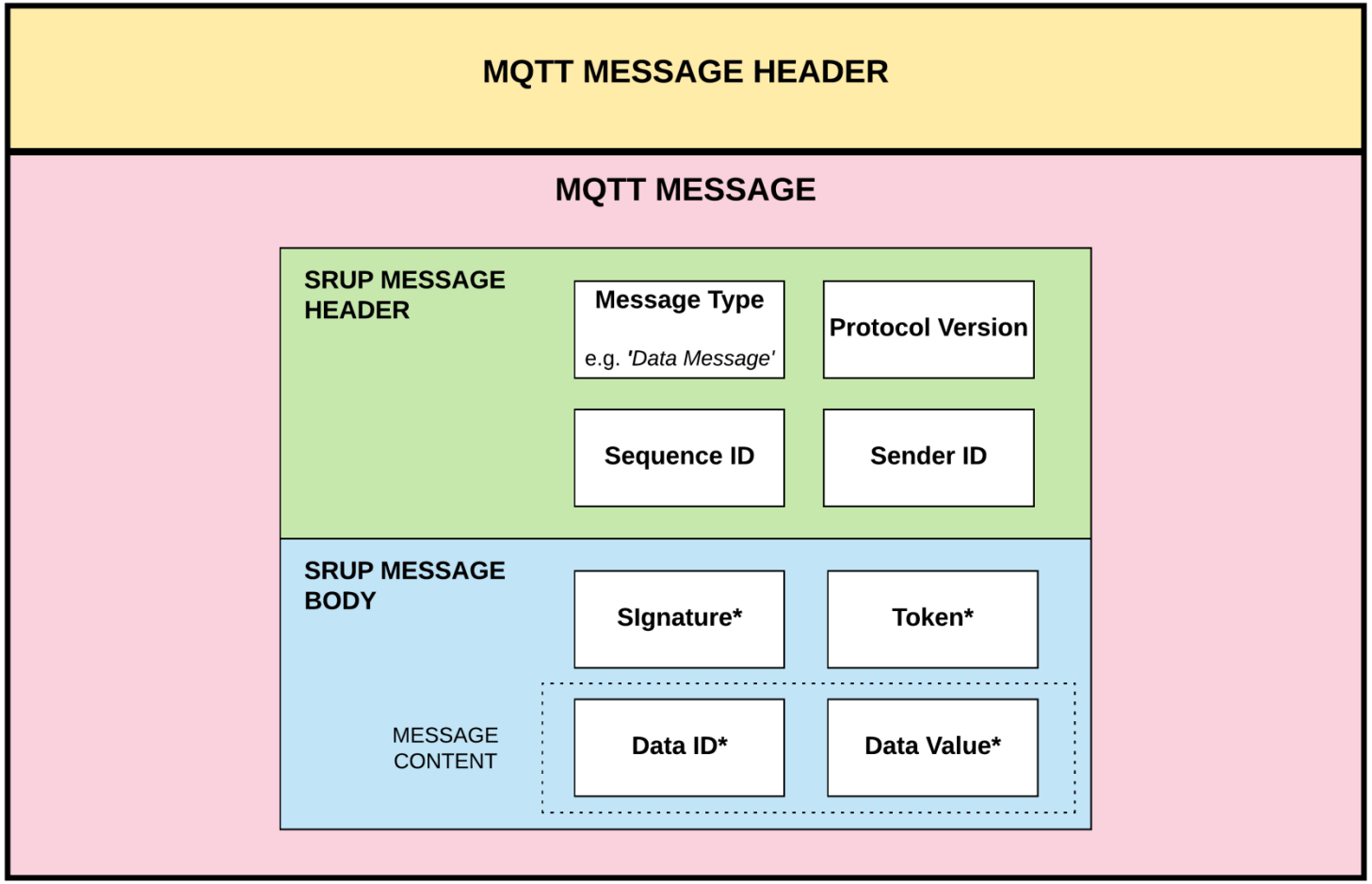


Elements of Research



SRUP – the Secure Remote Update Protocol

- Built on top of MQTT: a widely-used topic-based, *publish & subscribe* messaging protocol
  - SRUP provides an efficient binary message format for C2 messaging within the IoT
  - SRUP messages are delivered as the payload of an MQTT message; with the MQTT topic being used to denote the device to which the messages are being addressed…
  - The SRUP Messages consist of a number of fields:
    - Header
    - Signature
    - Token
    - Message content



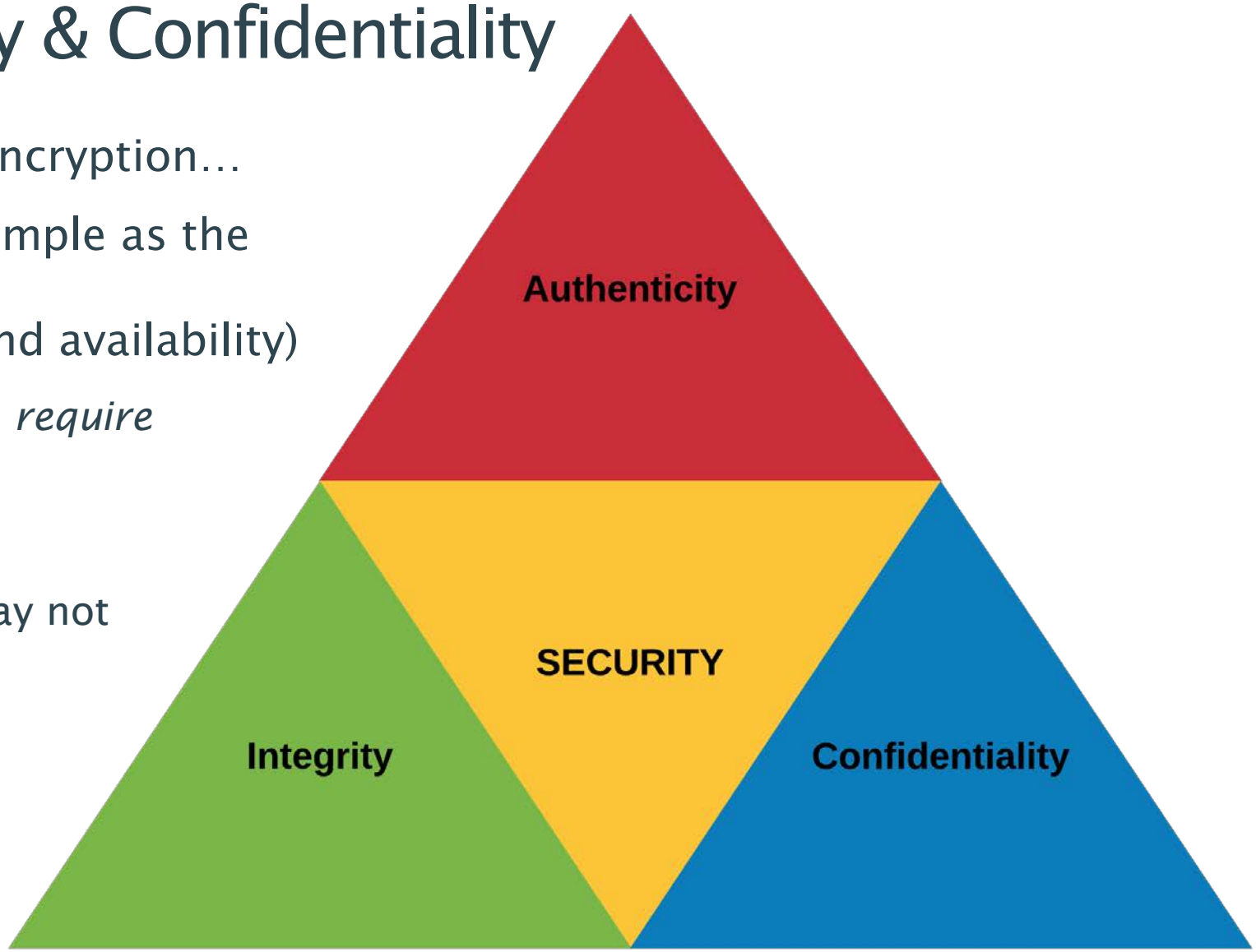
* Denotes variable length field – so an additional byte per field is also sent containing a field length…

Crypto Agility

- The current implementation of SRUP uses RSA_{2048} + SHA-256 for cryptographic signatures…
- The protocol has been designed to be inherently *crypto-agile*; enabling easy substitution for other algorithms in the future
  - Replacement with future *quantum-safe* cryptographic algorithms (once standardized)
  - Key lengths in given system implementations can be varied to provide best balance of security / speed…
- The *application developer* doesn't need to worry about the details of the implementation – even when using the low-level version of the library…

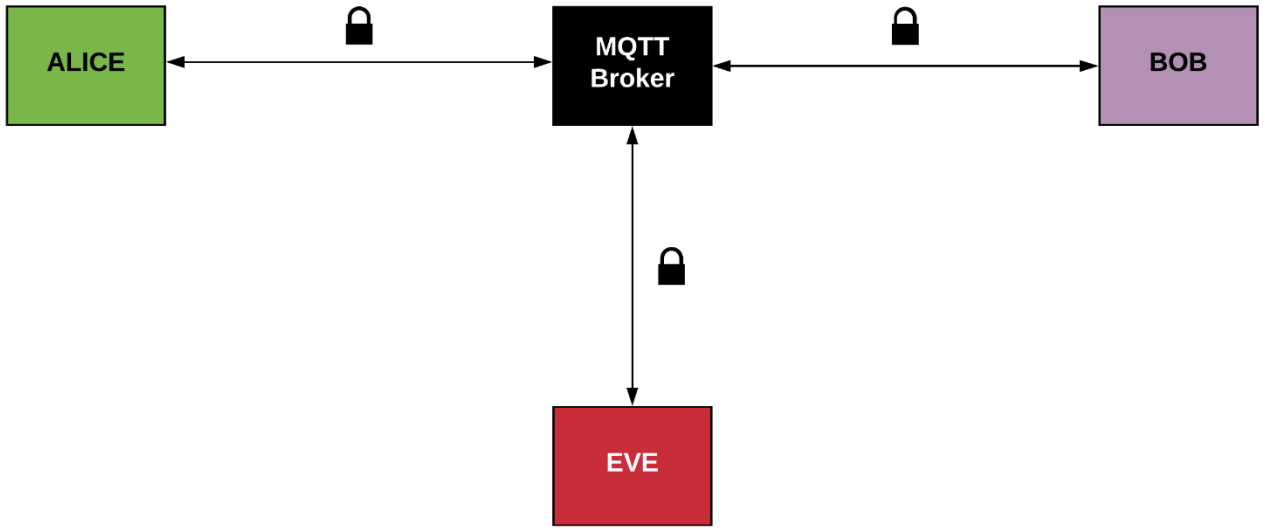
Authenticity, Integrity & Confidentiality

- Security is more than just encryption…
- Security within IoT isn't a simple as the *traditional 'CIA'* (confidentiality, integrity, and availability)
  - Some applications may not *require* message encryption
    - e.g. low-power devices
  - For the IoT – availability may not be a requirement
    - Or possible?
  - Authenticity & Integrity of messages is **essential** for IoT security…



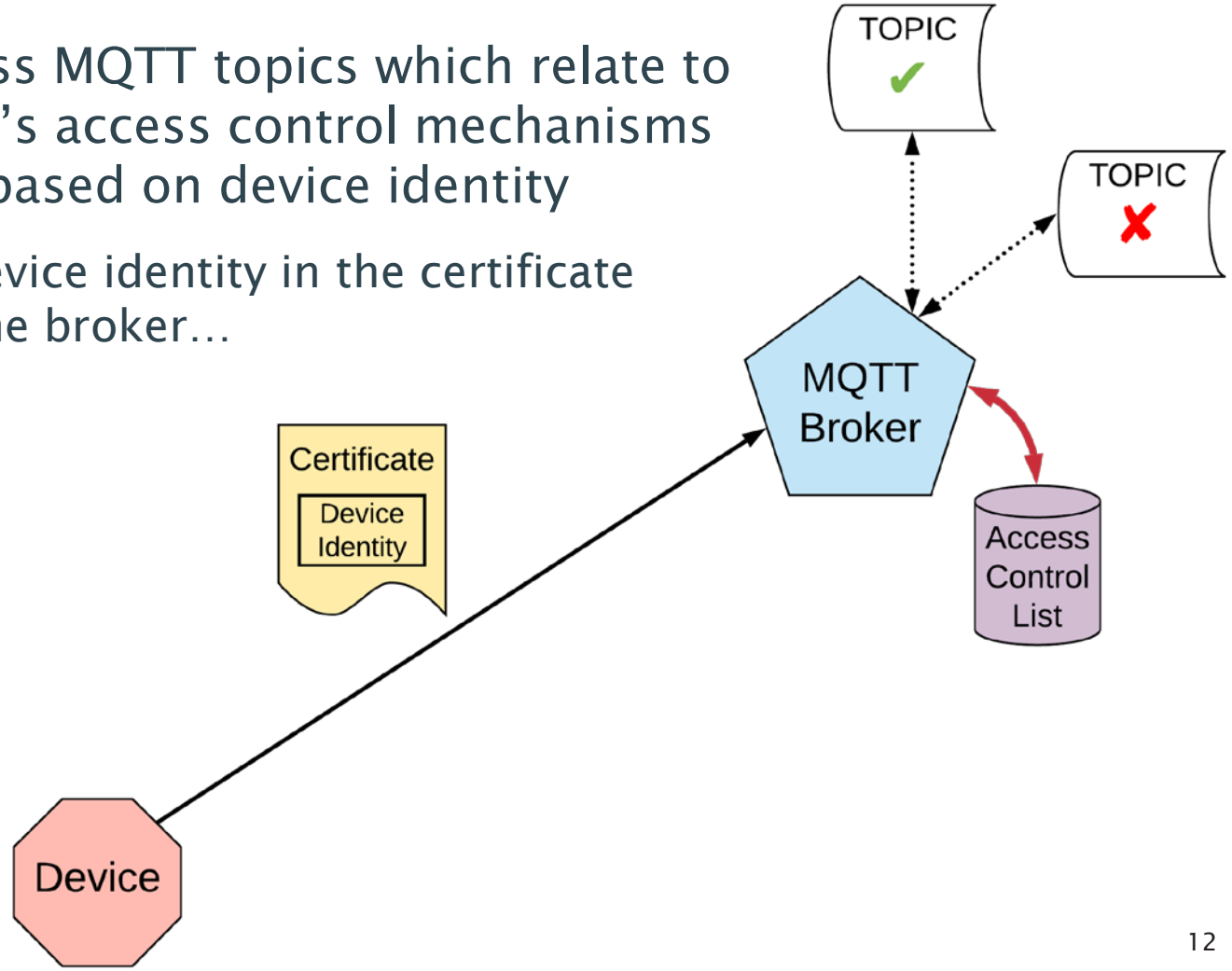
MQTT over TLS

- Within SRUP, the Sender ID, Message Signatures & Sequence IDs provide guarantees of the message authenticity & integrity…
- Where required, confidentiality can be provided using MQTT over Transport Layer Security (TLS) – as used for secure web-browsing.
- This requires a careful implementation to ensure that devices cannot subscribe to topics that are outside of their 'need to know'…



Identity, Access Control, and Certificates

- To ensure that devices can only access MQTT topics which relate to themselves, we use the MQTT broker's access control mechanisms to limit publication & subscriptions, based on device identity
  - This is enforced by embedding the device identity in the certificate used to establish the connection to the broker…



Static Identity vs. Dynamic Identity

- Traditional approach to device identity is to *fix* identity within the device at the time of manufacture
  - Provides a fixed relationship between *physical device* and *logical* device identity
  - Keys embedded in device at construction; can also be used to enable secure boot…
- This approach is expensive, and requires secure distribution to prevent tampering and potential extraction of the key
- Device identity is unrevocable – without decommissioning the device…

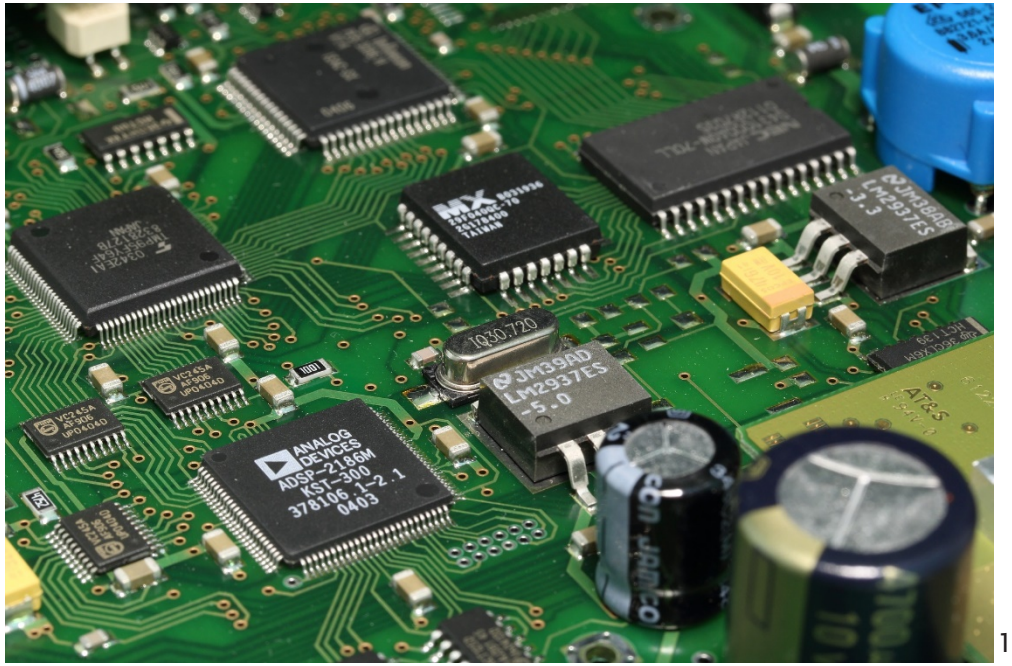
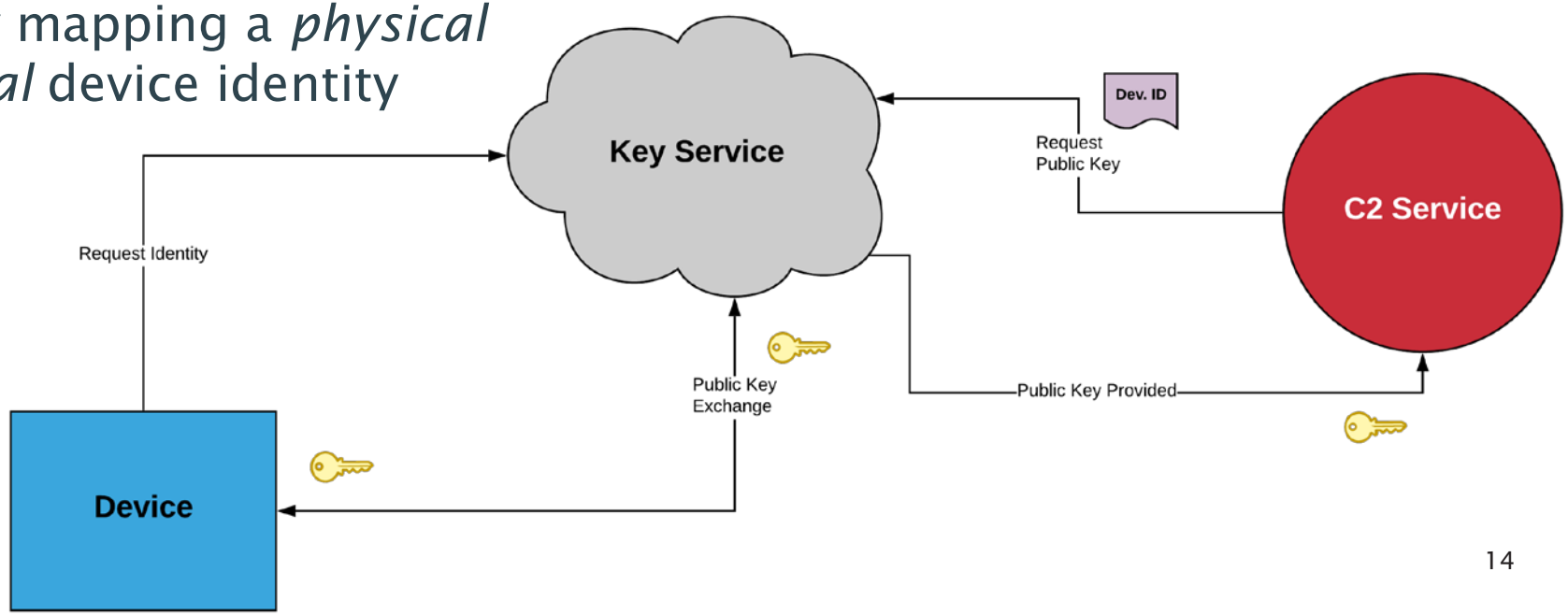


Image by olafpictures from Pixabay

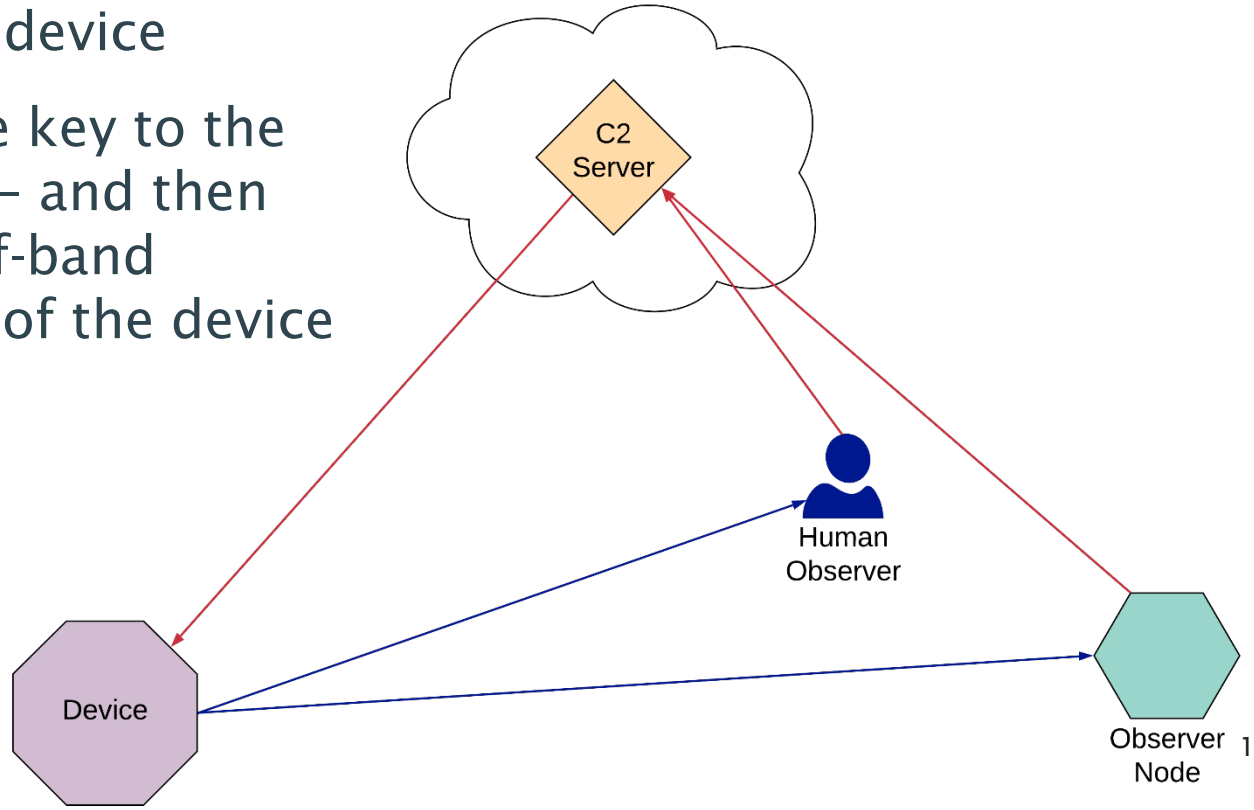
Dynamic Identity – Key Allocation & Revocation

- An alternative approach is to allocate identity dynamically
  - Anyone can request an identity at any time – via a secure webservice…
  - Easy to revoke access by removing identity from C2 service
    - Device can re-connect by requesting a new identity…
- The challenge is securely mapping a *physical* device to any given *logical* device identity



Implications of Dynamic Identity

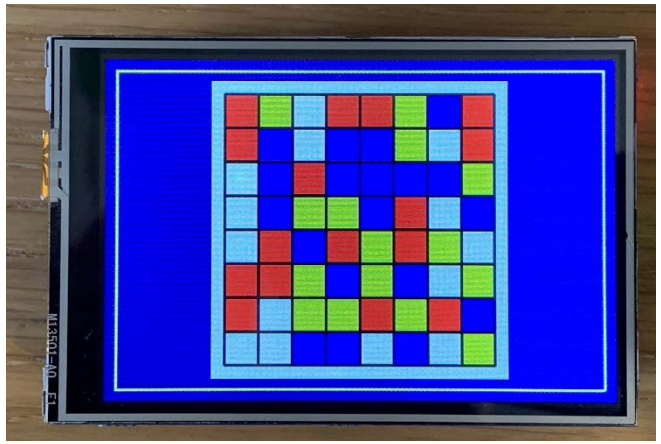
- Using dynamic identity removes the simple (fixed) link between physical & logical identity…
- Consequently alternative approaches are required to provide trusted link to determine identity of specific physical device
- A solution to this is to send a one-time key to the device, encrypted using its public key – and then to use a trusted third-party & an out-of-band channel to communicate confirmation of the device identity, back to the C2 server…

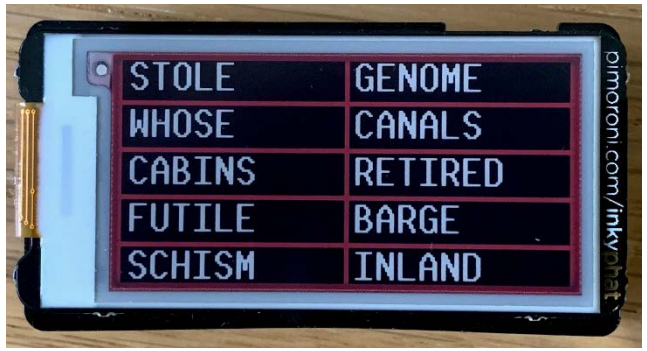


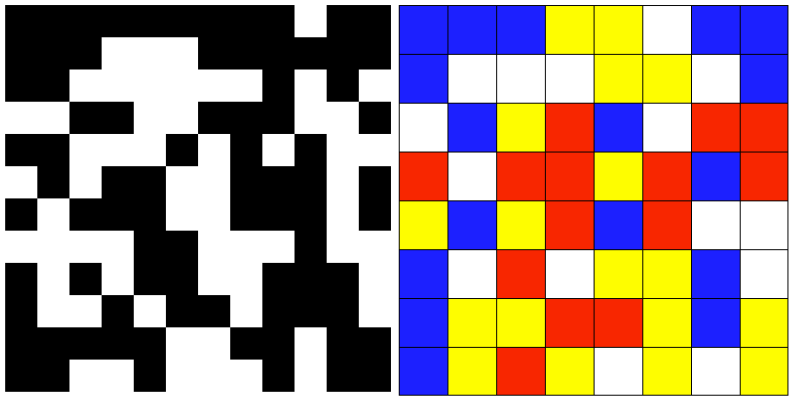
Human-Moderated Proof of Identity

- Presentation of high-entropy 128-bit values to humans requires careful comparison
  - Even when expressed in hexadecimal notation…
- Alternative methods of presentation can be used to make human comparison easier to carry-out
  - Examples include pictograms, and word-lists…
- These can be displayed by the device & the C2 server's user-interface…

```
b08c51d6-6eb8-4d67-9118-0e6da6f2eddd
```



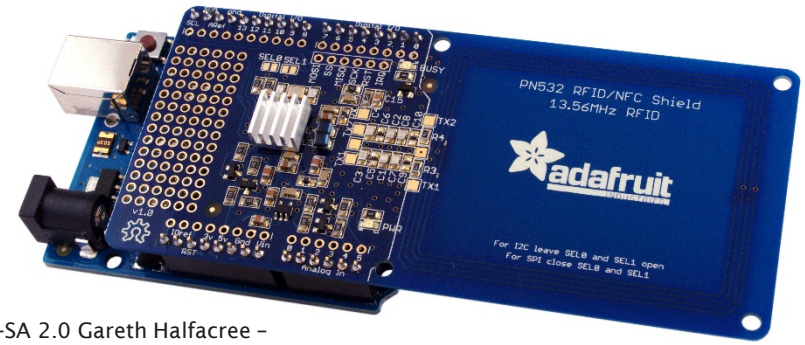




TAXI	GETS
TAG	PIE
POOL	BITTEN
CORK	BUSY
CLERICS	MITT

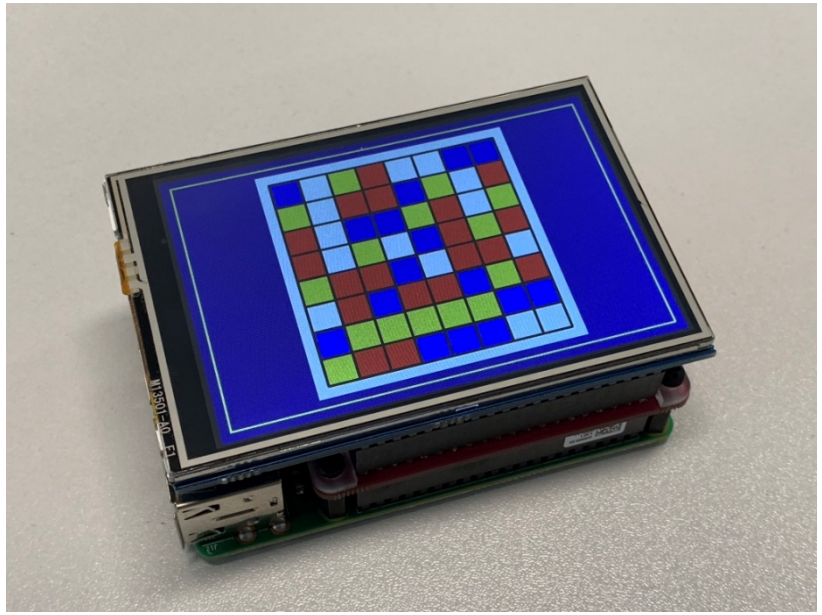
Machine-Moderated Proof of Identity

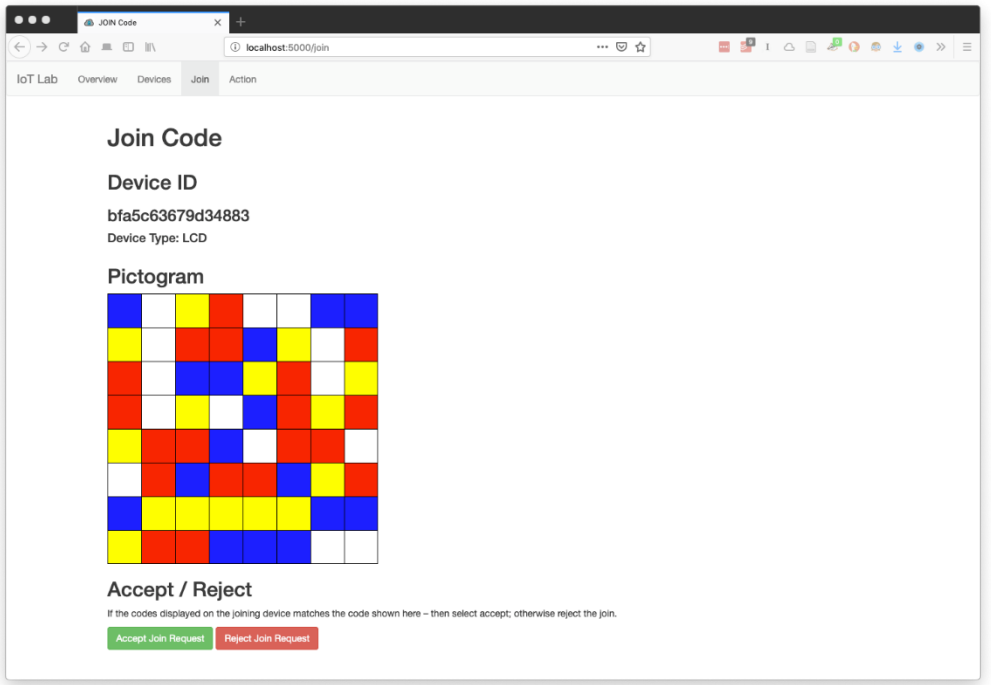
- Research is currently on-going to establish candidate mechanisms for a machine-moderated proof of identity
- Technologies being considered for this include:
  - Camera-based visual recognition using QR Codes
  - Very short-range RF links, (e.g. NFC / RFID) & Bluetooth (BLE)
  - Short-range directional RF links



pySRUP – a Python wrapper for ease of use…

- As an example of how a protocol such as SRUP could be *packaged* for more general use: a Python Wrapper (pySRUP) has been produced…
- Using pySRUP enabled a developer to build a system using the SRUP protocol, with a minimum of code; enabling them to focus on the application they're building – not the *plumbing* to make it work securely…



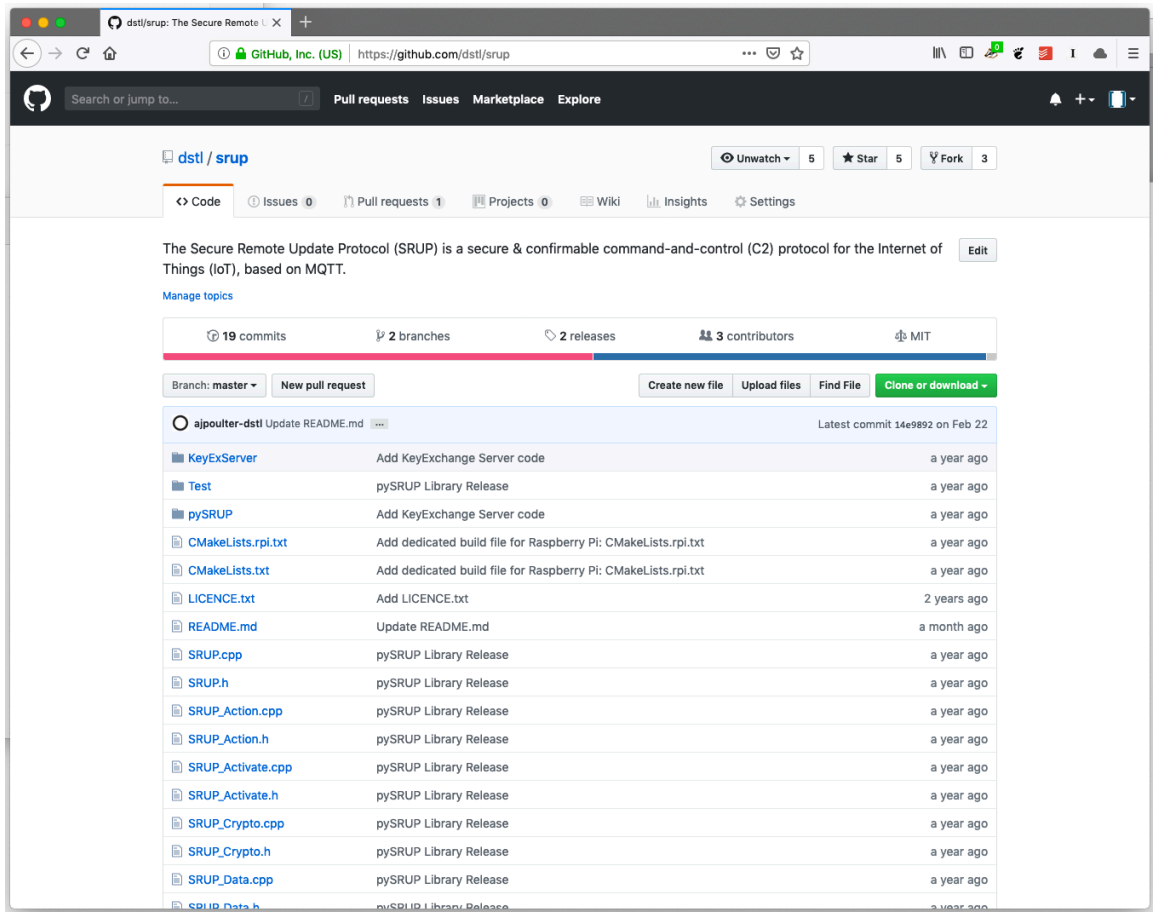


Future Work

- In addition to completing research on, and an implementation of, machine-moderated observation; a *demonstration experiment* is planned for summer 2020
  - Details are still being finalized – but the intent is to demonstrate the use of the protocol, and other technologies discussed in this presentation in the context of a scenario based around real-world application of these technologies

Example Implementation

- Still in development with new functionality being added over the next 6-9 months…
- Released under MIT licence
- Source code: https://github.com/dstl/srup
- To find out more; please get in touch:
  - ajpoulter@dstl.gov.uk



UNIVERSITY OF Southampton

Questions?