Breaking Matter: Vulnerabilities in the Matter Protocol

Speaker: Béla Genge

Contributor: Ioan Pădurean

Béla GENGE

Senior Security Researcher @ Bitdefender

Research on Matter for ~two years

University teaching / research background for ~20 years

bgenge@bitdefender.com

AGENDA

- Intro & motivation
- Background on Matter protocol
- Security findings
- Way forward

Modern world is about automation

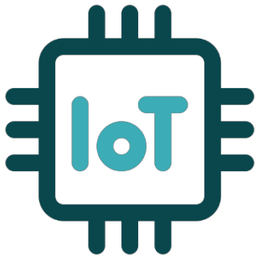
Smart Factory

Smart Home/Building

4th industrial revolution

INDUSTRY ~~4.0~~ 5.0

Digital Transformation



<accepting applications for cool buzzwords>

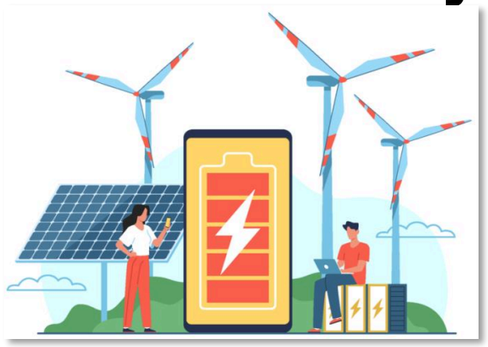
Modern home is connected

Inverter

Solar panels

EV Charging

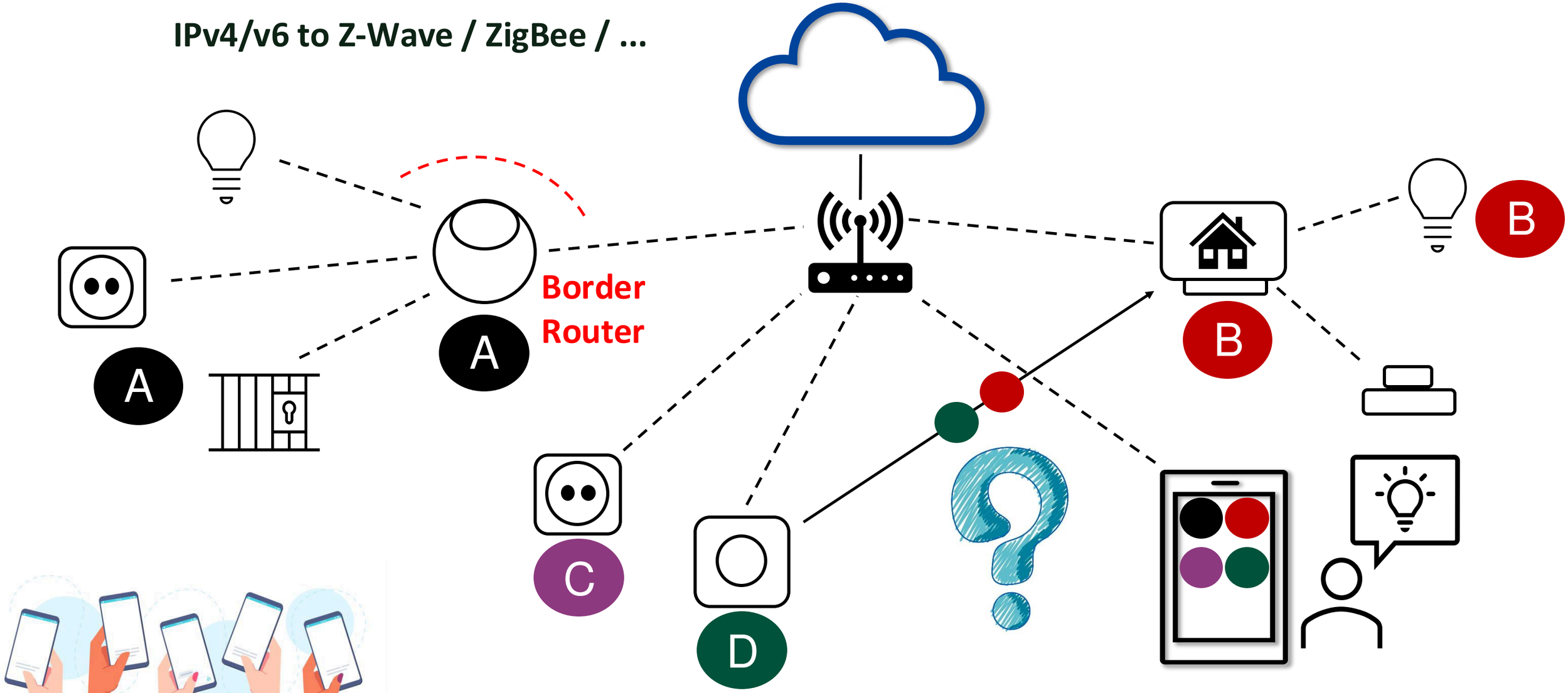
Smart battery



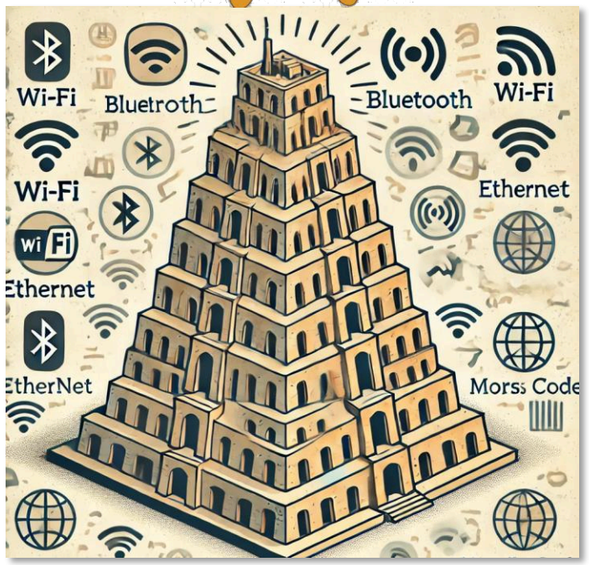
Heat pump

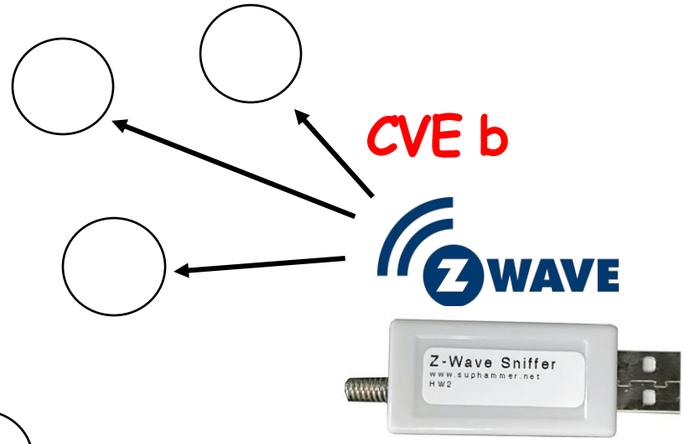
Not isolated but connected to the power grid!

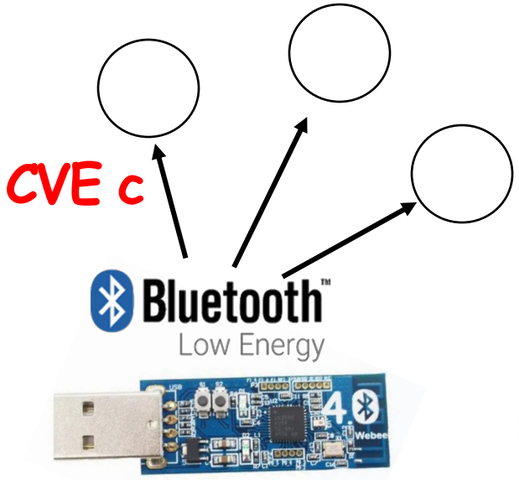
Traditional IoT ecosystem

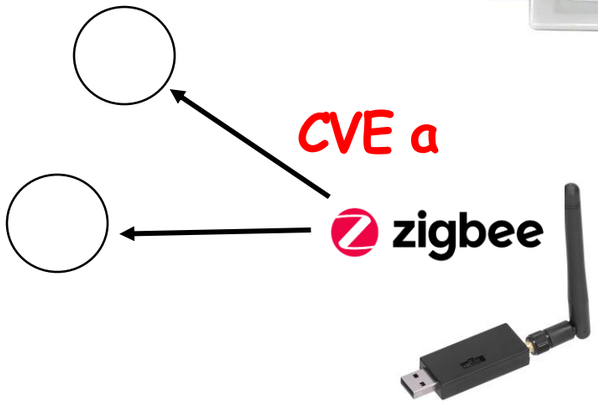


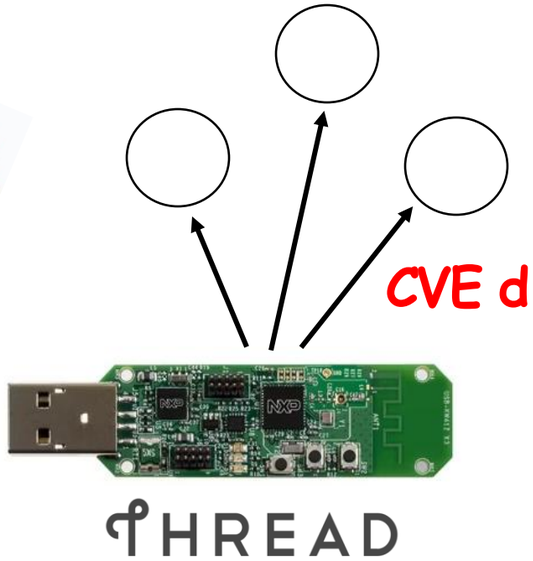
IoT ecosystem: Attacker's perspective











(Attacker's) Life is about to get easier

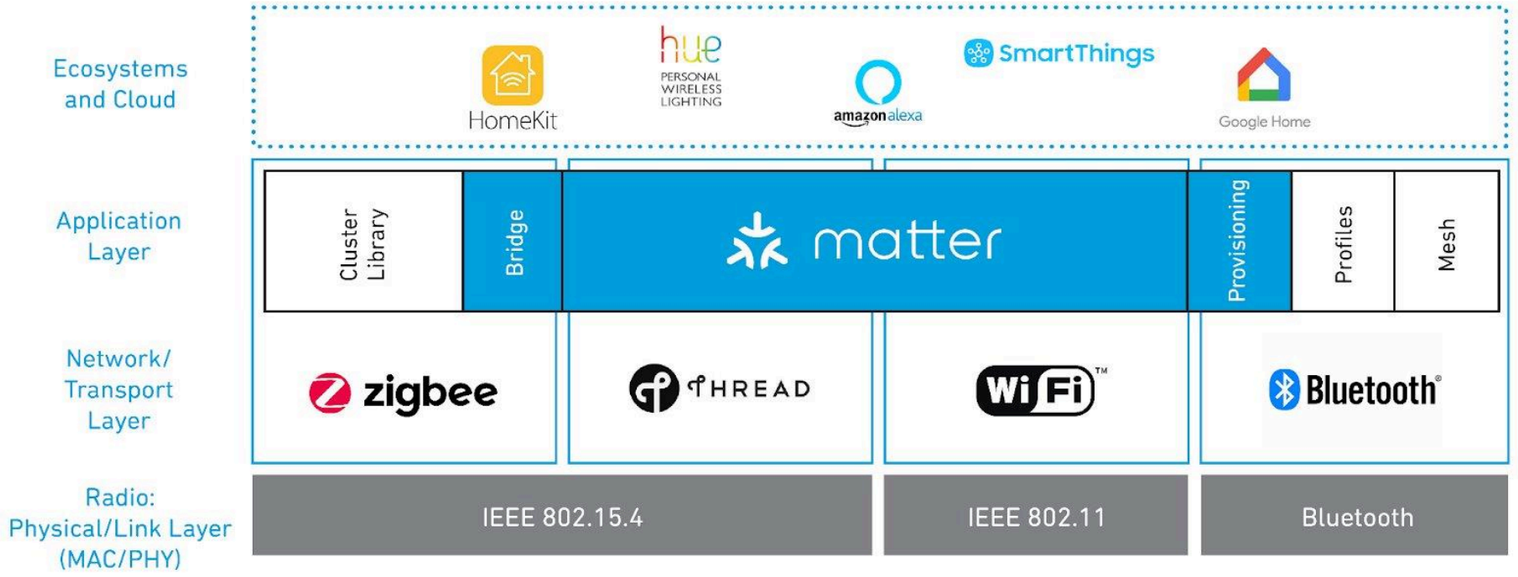
matter

Interoperability through industry-unifying standard

👍

Matter: the beginning

- December 2019: Work Group on Connected Home over IP
- CSA: **Connectivity Standards Alliance**
- Provides a secure and interoperable solution



Source: https://bytebeam.io/blog/what-is-the-matter-protocol/

Secure

IPv6

Open

Interoperable

Multi-admin

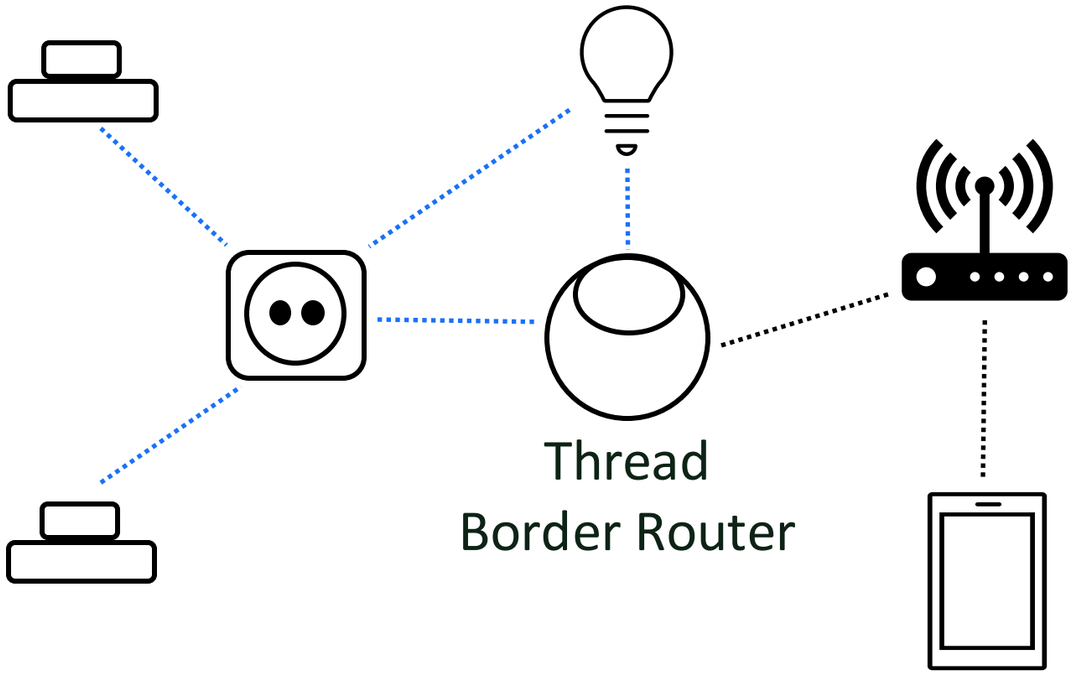
Established protocols

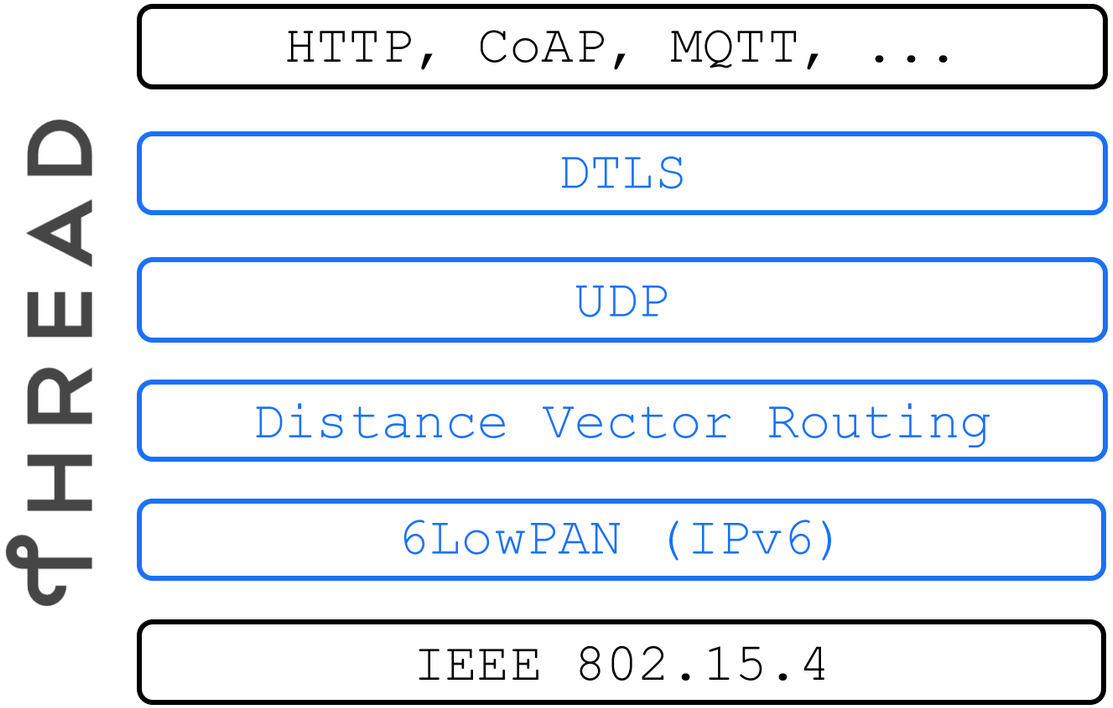
Shift in communication

Understanding Matter (1.0) and its Significance (bytebeam.io)

Thread anyone?

- IPv6-based protocol for low-power, mesh networks
- It uses 6LoWPAN on top of IEEE 802.15.4 wireless protocol





Matter: today

matter

- Global collaboration (**600+**):
  - 32 promoters
  - 284 participants
  - 288 adopters

- Device certification programs aligned with several directives

Matter is becoming
the (single?)
established standard for IoT

Matter: versions

Amazon, Apple, Google, Samsung SmartThings and the Zigbee Alliance

v1.0:

- Lighting: power plugs, electric lights, switches
- Door locks
- Thermostats
- Heating
- Ventilation
- Air conditioning
- Blinds & shades
- Motion sensors
- TV
- Video games

v1.1:

- Bug fixes
- Enhanced SDK API

v1.2:

- Refrigerators
- Portable air conditioning
- Dishwashers
- Laundry washers
- Robotic vacuum cleaners
- Monoxide alarms
- Air quality sensors
- Air purifiers
- Fans

v1.3:

- Water management
- **Energy management**
- **Electric Vehicle Charging**
- Microwave ovens
- Ovens
- Cooktops
- Extractor hoods
- Laundry dryers
- Media players

v1.4:

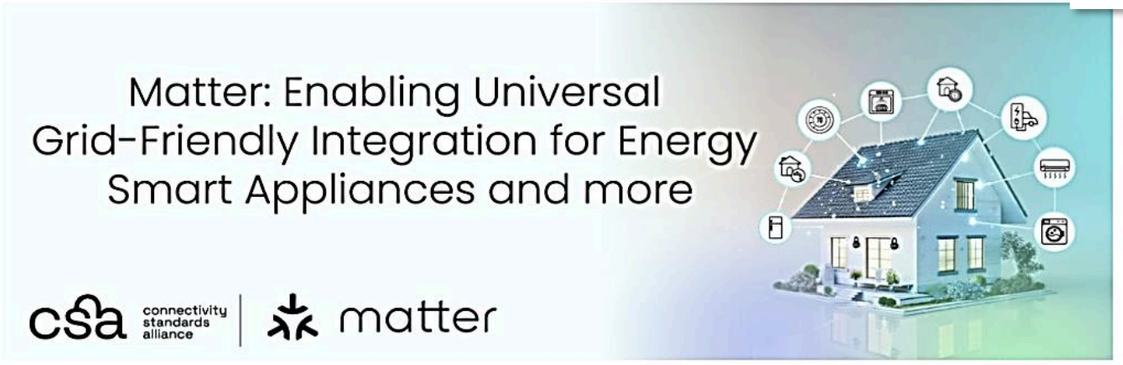
- **Home Router Access Point**
- **Solar Power**
- **Batteries**
- **Heat Pumps**
- **Water Heaters**
- Enhancements

Matter: latest news

BLOGS

Matter: Enabling Universal Grid-Friendly Integration Energy Smart Appliances and more

10/1/2024



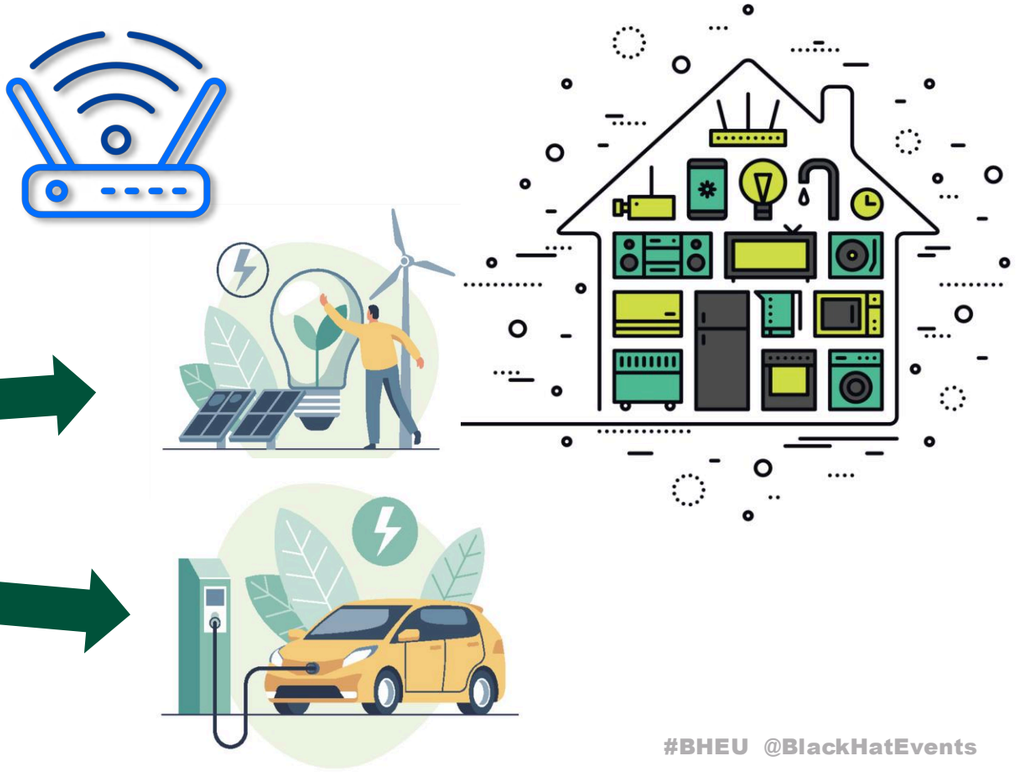
PRESS RELEASES

Matter 1.4 Enables More Capable Smart Homes

11/7/2024

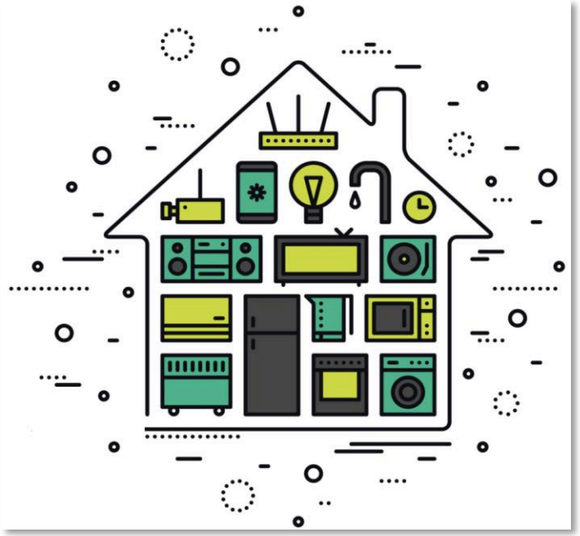
Enhanced Network Infrastructure with Home Routers and Access Points (HRAP)

New Energy Device Types and Capabilities



Matter: attacker's perspective

Find (vulnerability) Once, Run Everywhere!



Matter: my perspective

WIP: Security Vulnerabilities and Attack Scenarios in Smart Home with Matter (2024)

Seamlessly Insecure: Uncovering Outsider Access Risks in AiDot-Controlled Matter Devices (2024)

mDNS in action with the home automation Matter protocol

Real life example of mDNS being used to find and add devices to a home automation network.

Paul Otto · Follow
7 min read · Jul 4, 2024

First questions

Up to 78 million batteries will be discarded daily by 2025, researchers warn

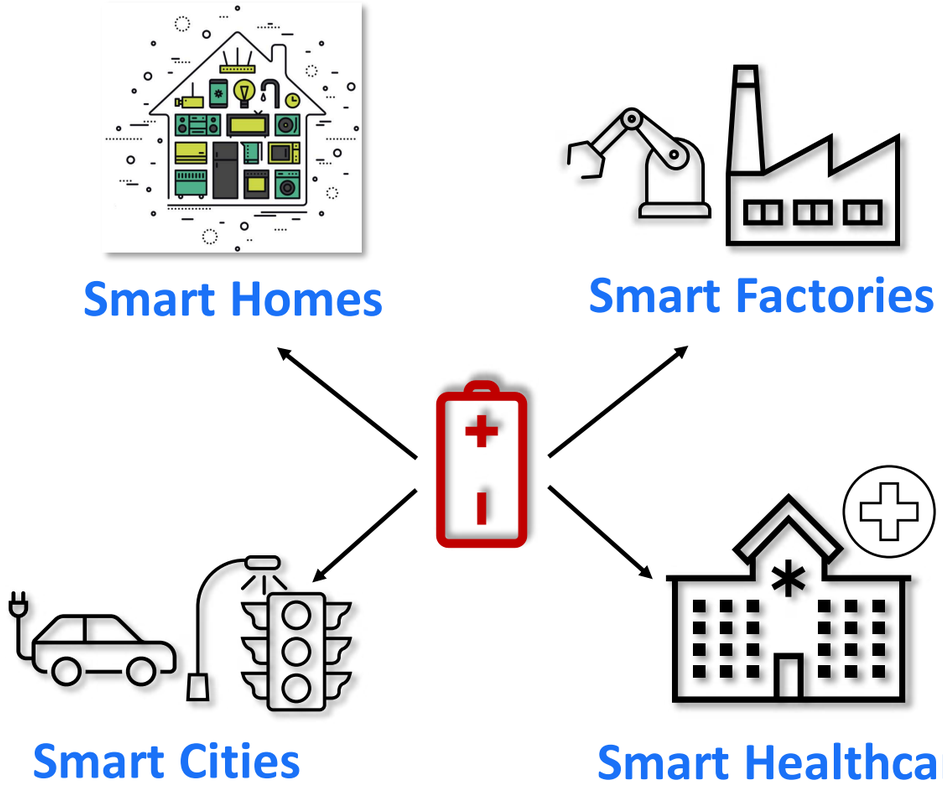
130 million batteries will be both manufactured and disposed of every single day by 2025.

https://cordis.europa.eu/article/id/430457-up-to-78-million-batteries-will-be-discarded-daily-by-2025-researchers-warn

https://www.enables-project.eu/outputs/position-paper/

- How resilient are battery-powered devices?
- While privacy is protected by the fabric, how can one still discover devices?
- What about running disruptive DoS attacks?

?



What can possibly go wrong?

No router

Connectivity Standards Alliance : Security Vulnerabilities, CVEs

Published in: 2024 January February March April May June July August September October

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9 In CISA KEV Catalog

Sort Results By : Publish Date Update Date CVE Number CVE Number CVSS Score EPSS Score

Copy

CVE-2024-3454

An implementation issue in the Connectivity Standards Alliance Matter 1.2 protocol as used in the connectedhomeip SDK allows a third party to disclose information about devices part of the same fabric (footprinting), even though the protocol is designed to prevent access to such information.

Source: Bitdefender

Max CVSS	3.5
EPSS Score	0.04%
Published	2024-07-24
Updated	2024-09-10

CVE-2024-3297

An issue in the Certificate Authenticated Session Establishment (CASE) protocol for establishing secure sessions between two devices, as implemented in the Matter protocol versions before Matter 1.1 allows an attacker to replay manipulated CASE Sigma1 messages to make the device unresponsive until the device is power-cycled.

Source: Bitdefender

Max CVSS	6.5
EPSS Score	0.04%
Published	2024-07-24
Updated	2024-09-10

No detection

No alarm

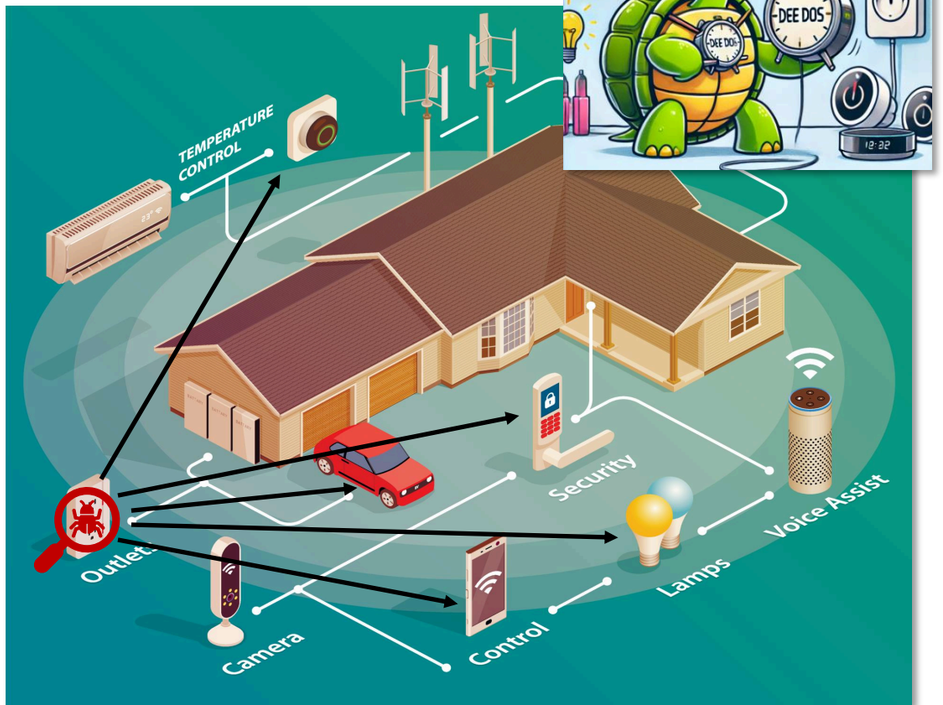
No charging

Today

- Learn about the **game changer Matter** standard

matter

- The **First** reported **vulnerabilities** in Matter **SDK** (Software Development Kit):
  - CVE-2024-3297 – **Delayed Denial of Service** (DeeDoS) [di: dos]
  - CVE-2024-3454 – **Device feature scanning**



Connectivity Standards Alliance : Vulnerability Statistics

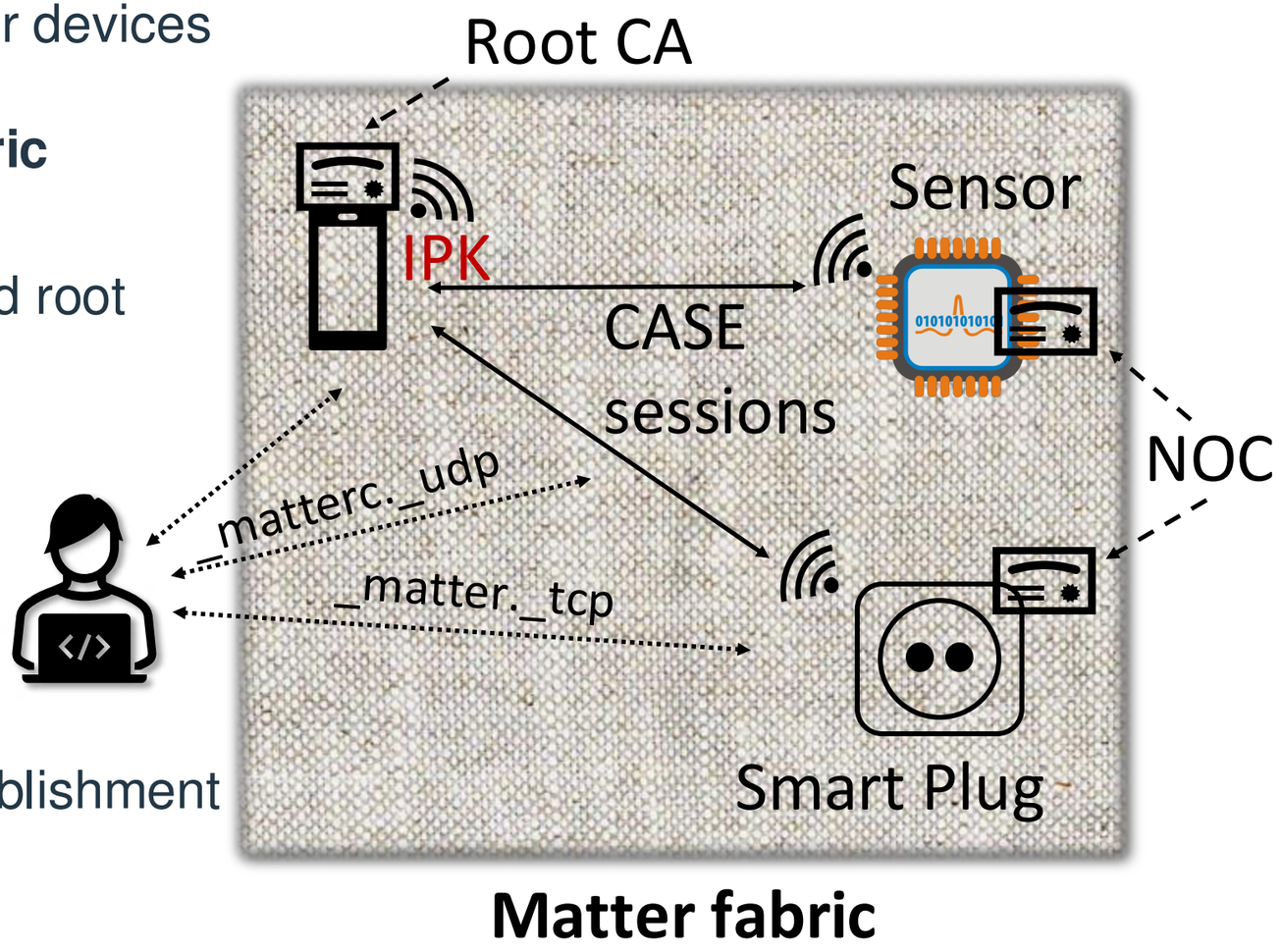
Products (2) | Vulnerabilities (2) | Search products | CVSS Report | Metasploit Modules

https://www.cvedetails.com/vendor/35076/

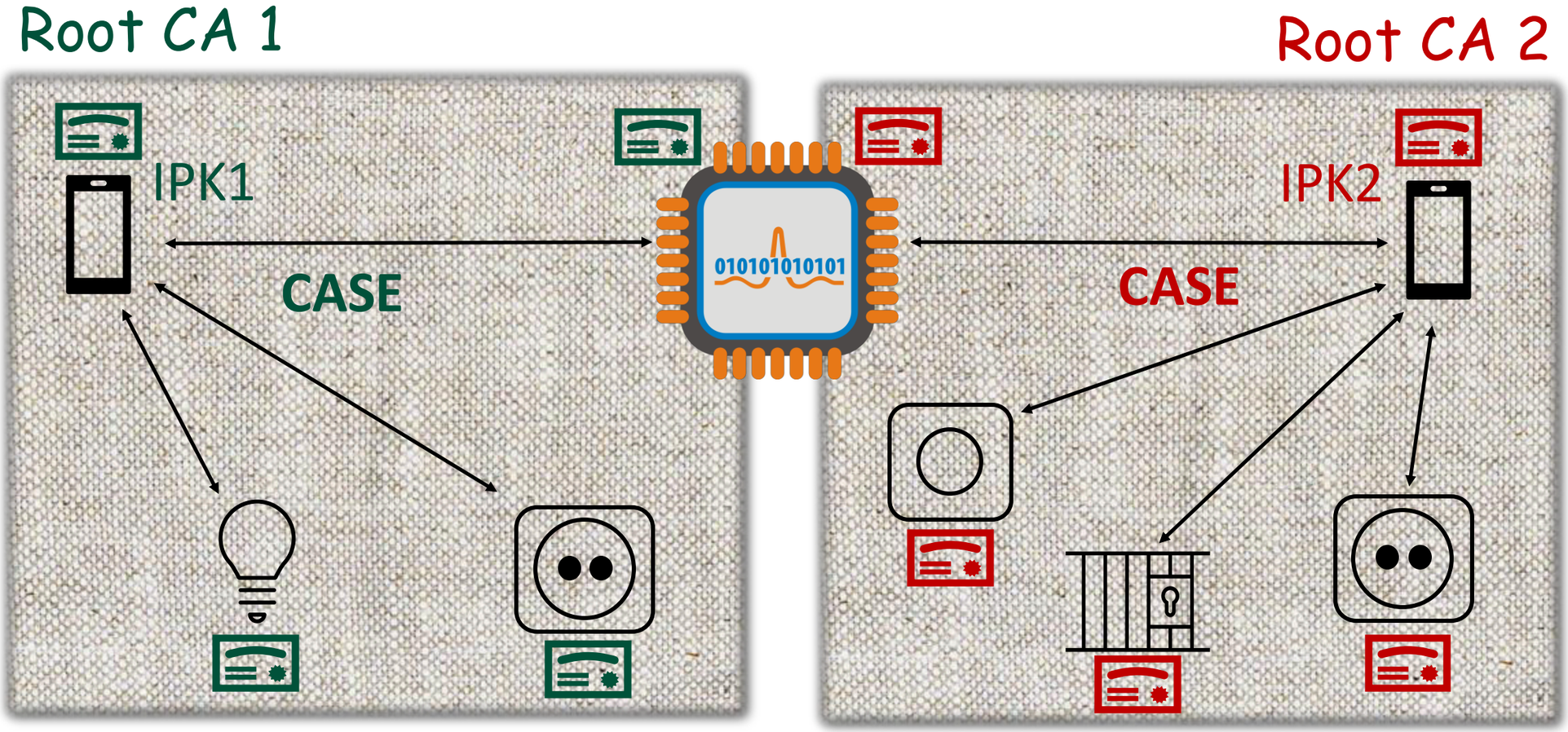
(few) Fundamental concepts

Basic terminology

- Matter uses Multicast DNS (mDNS) to discover devices
- Devices are added (commissioned) into a **fabric**
- **Fabric:** collection of devices sharing a trusted root certificate
- Operational **root of trust**: the root certificate authority responsible for:
  - Allocating fabrics
  - Issuing node operational certificates (NOC)
- **CASE:** Certificate Authenticated Session Establishment
- **IPK**: Integrity Protection Key



Multi-fabric support



SDK vs. device (product)

Commissioning

Data Model

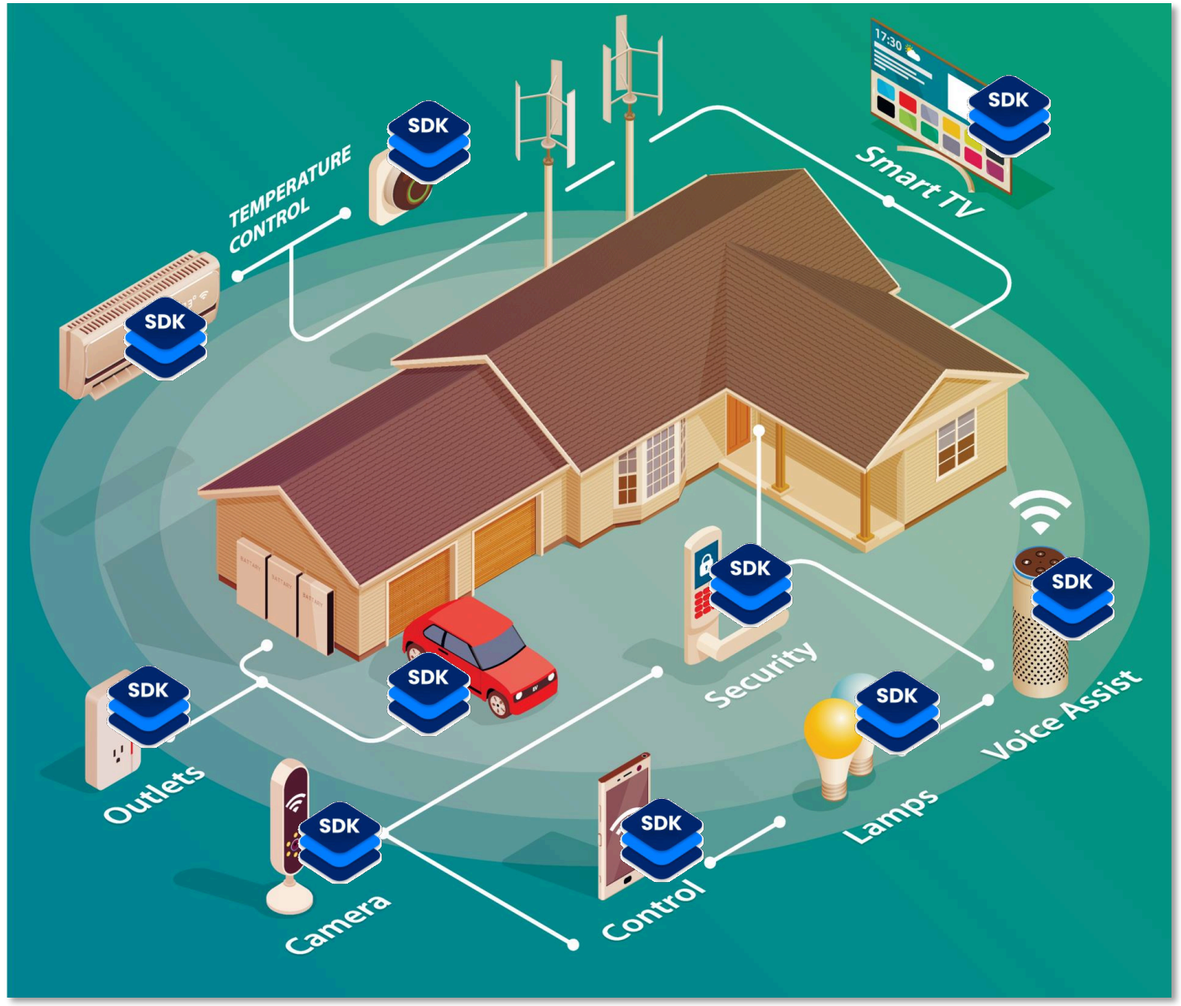
Fabrics

CASE

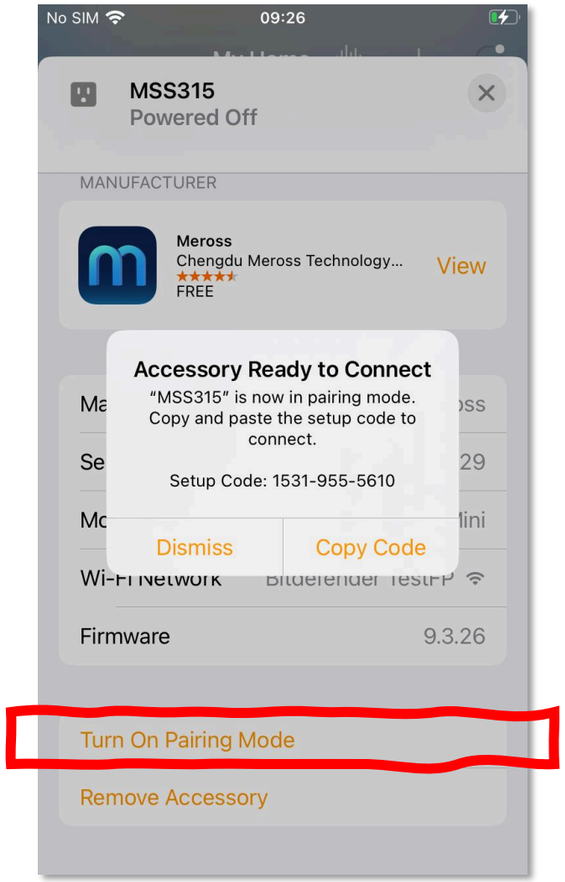
PASE

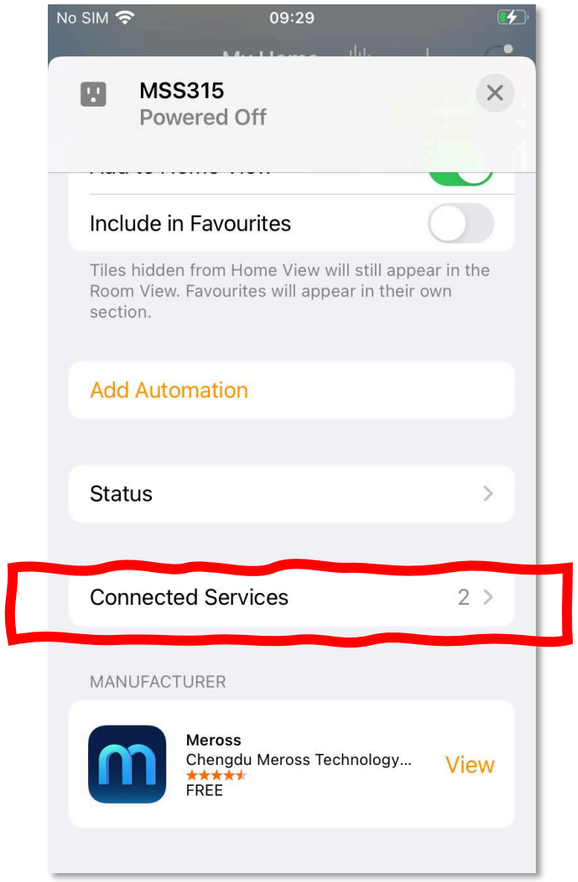
Interaction Model

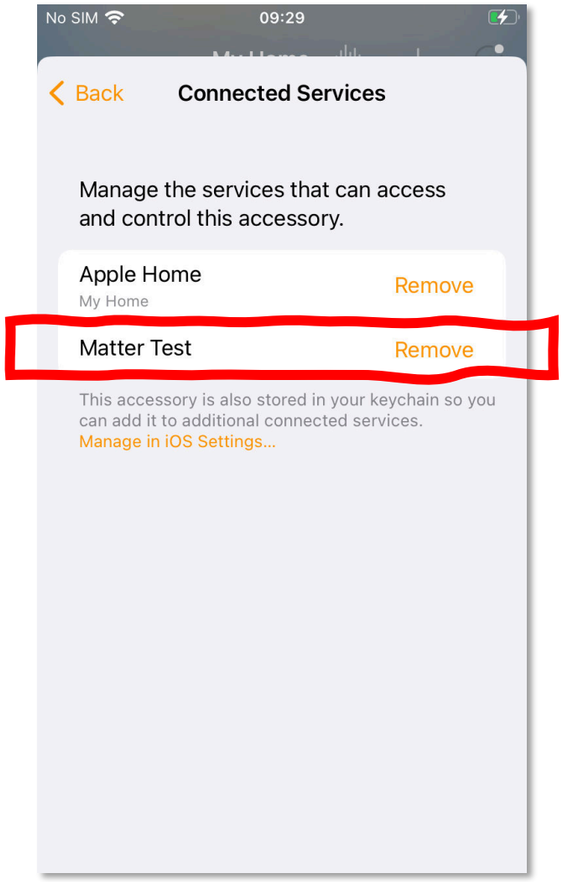
SDK: development tools for Matter devices



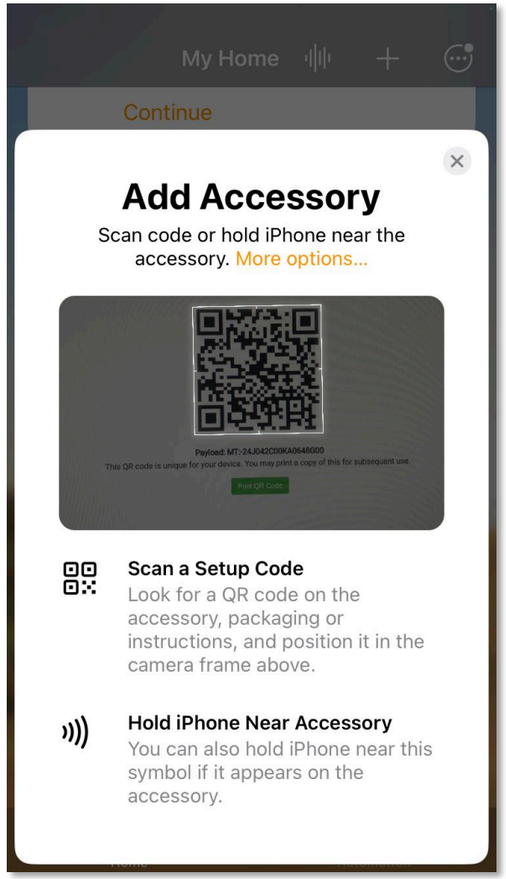
Commissioning: certified device

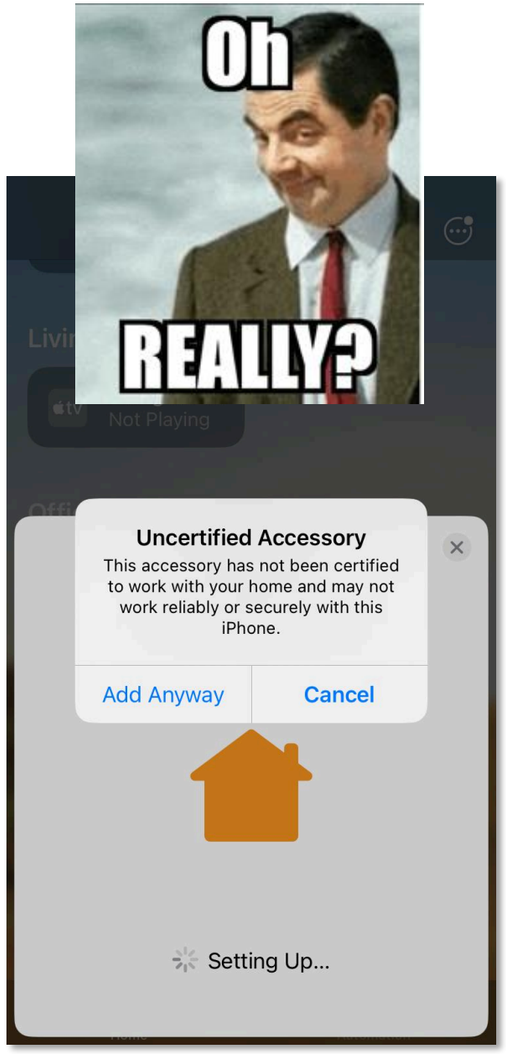


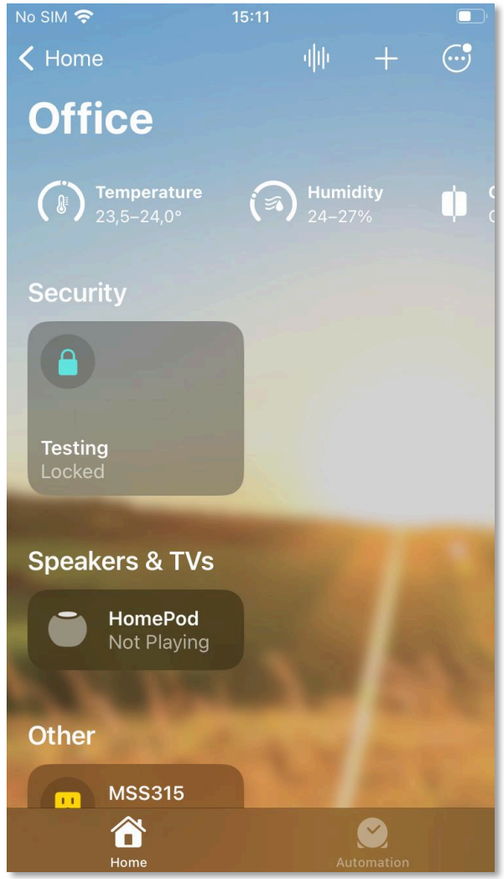




Non-certified device



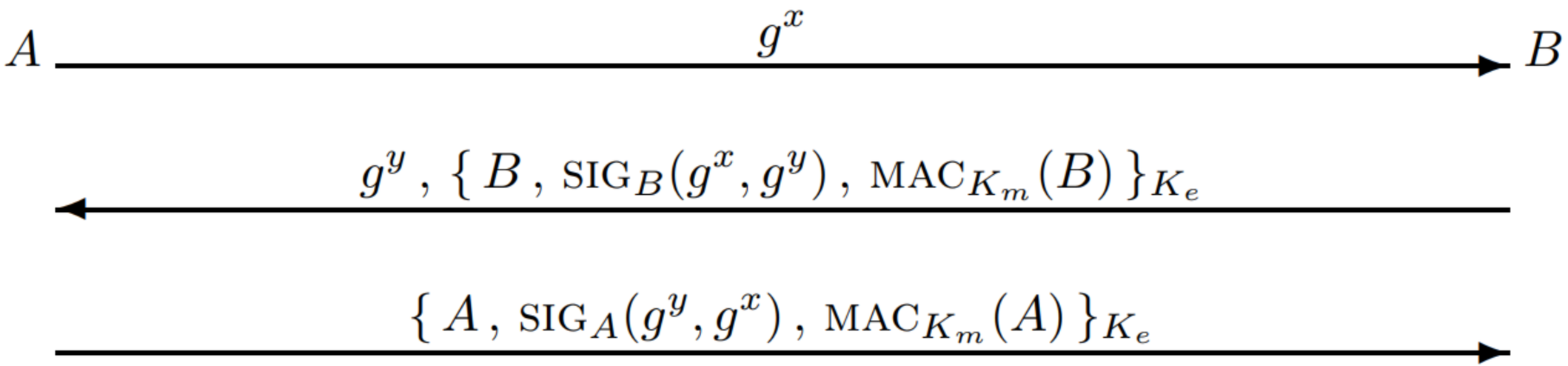




CASE protocol

- Certificate Authenticated Session Establishment (**CASE**):
  - Mutual authentication
  - Negotiate new session keys

- Built on the SIGn-and-MAc (**SIGMA**) family of protocols (Krawczyk, 2003)



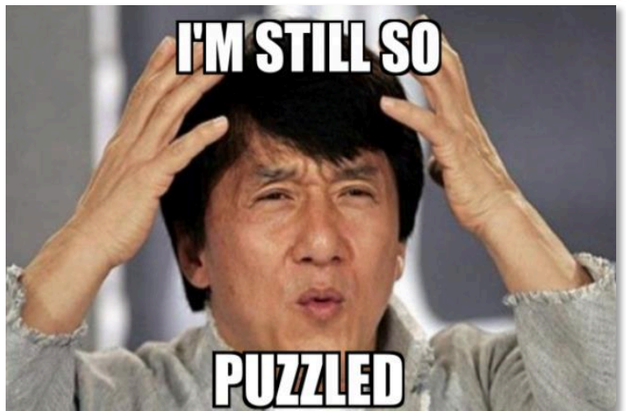
Original SIGMA-I protocol Krawczyk, 2003

CASE protocol: Sigma2, Sigma3 spec


```
Msg2 =
{
  responderRandom         (1) = Random,
  responderSessionId      (2) = ResponderSessionId,
  responderEphPubKey      (3) = ResponderEphKeyPair.publicKey,
  encrypted2              (4) = TBEData2Encrypted,
  responderSessionParams (5) = session-parameter-struct (optional)
}
```

```
byte TBEData2_A[] = {}
byte TBEData2_Nonce[13] = /* "NCASE_Sigma2N" */
       {0x4e, 0x43, 0x41, 0x53, 0x45, 0x5f, 0x53, 0x69,
        0x67, 0x6d, 0x61, 0x32, 0x4e}

TBEData2Encrypted = Crypto_AEAD_GenerateEncrypt(
     K = S2K,
     P = TBEData2,
     A = TBEData2_A,
     N = TBEData2_Nonce
```

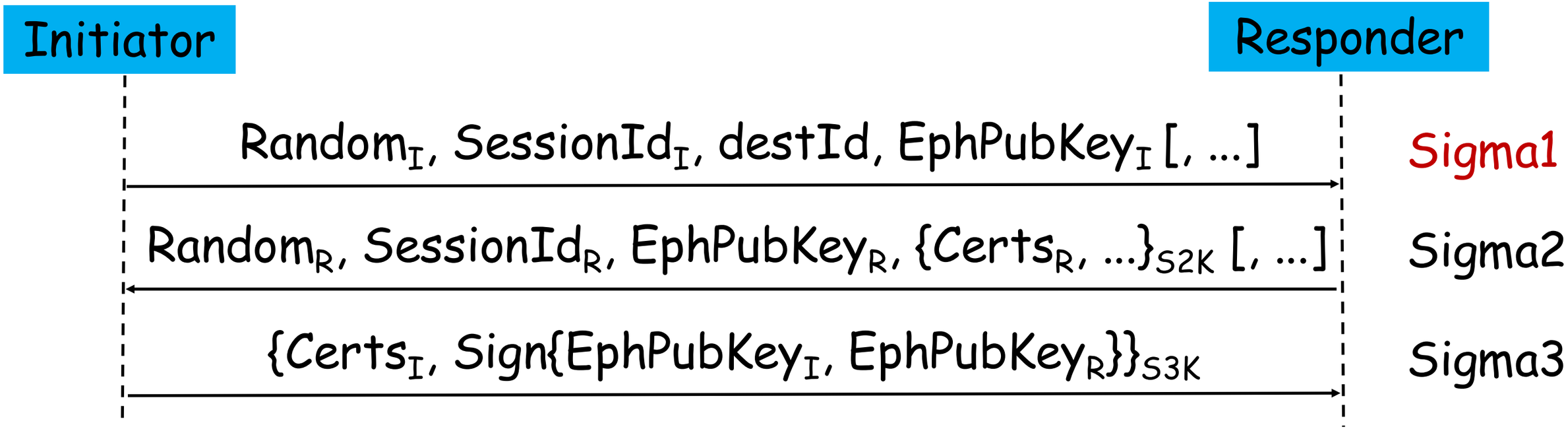
8. The initiator SHALL send a message with Secure Channel Protocol ID and Sigma3 Protocol Opc
from Table 18, “Secure Channel Protocol Opcodes” whose payload is the TLV-encoded Sign
`Msg3 = { encrypted3 (1) = TBEData3Encrypted }` with an anonymous tag for the outerm
struct.



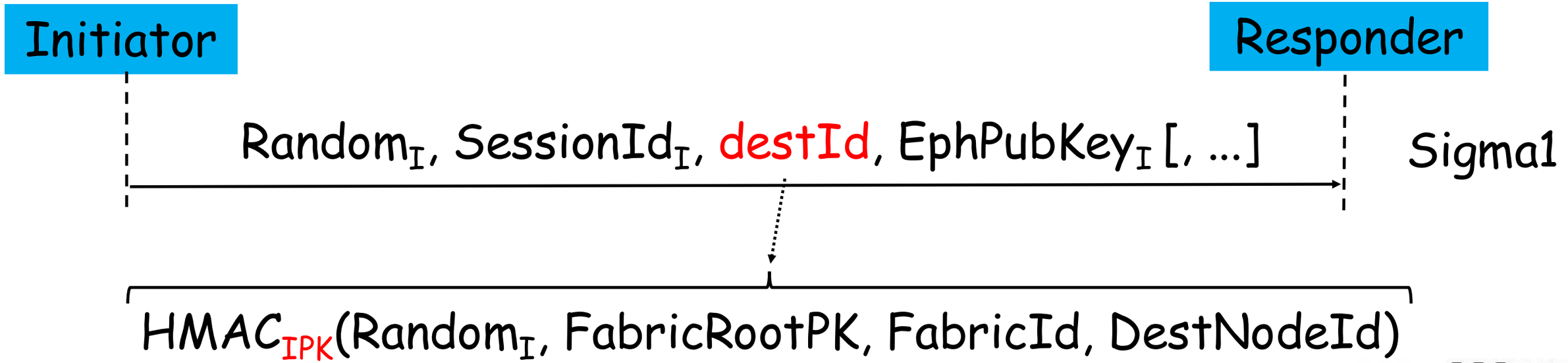

```
byte TBEData3_A[] = {}
byte TBEData3_Nonce[13] = /* "NCASE_Sigma3N" */
       {0x4e, 0x43, 0x41, 0x53, 0x45, 0x5f, 0x53, 0x69,
        0x67, 0x6d, 0x61, 0x33, 0x4e}

TBEData3Encrypted = Crypto_AEAD_GenerateEncrypt(
     K = S3K,
     P = TBEData3,
     A = TBEData3_A,
     N = TBEData3_Nonce
)
```

Error 404: Not found
(not defined)

CASE protocol: unfolded

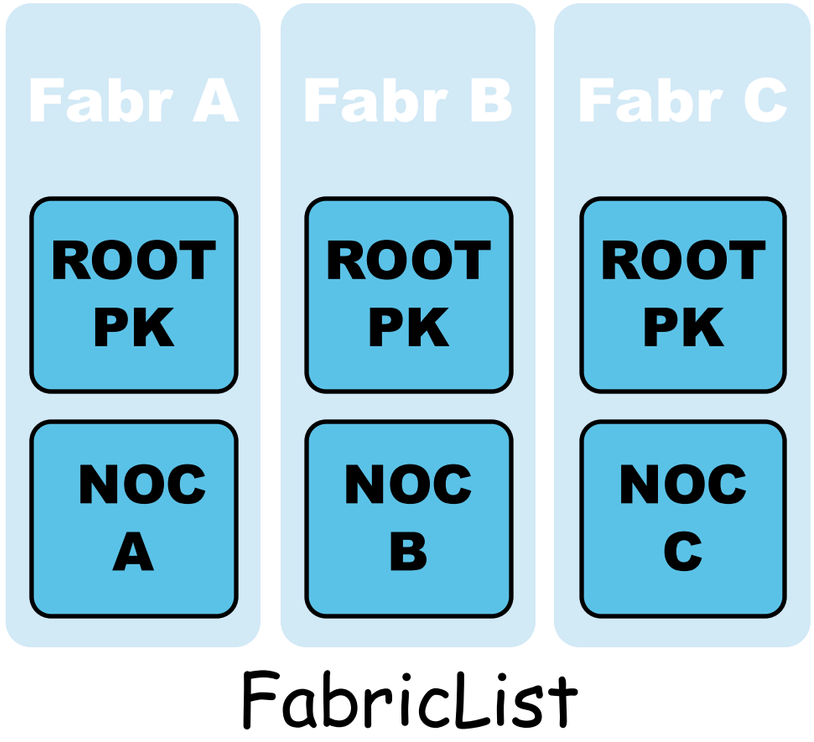


CASE Sigma1



IPK: Integrity Protection Key

CASE Sigma1 validation



FabricList

For-each (FabricId, NodeId, RootPK) in FabricList:

destId' = $HMAC_{IPK}(Random_I, RootPK, FabricId, NodeId)$;

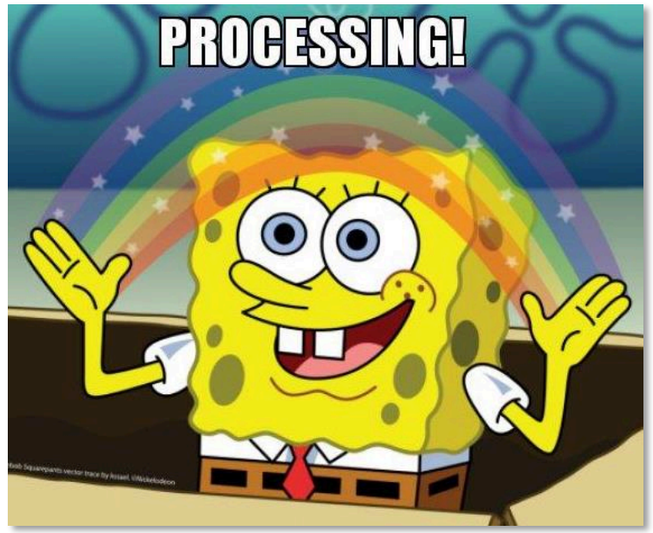
If (destId' == destId)

ValidateSigma1();

break;

EndIf

EndFor-each



CASE Sigma1 replay protection


```
Ethernet II, Src: Intel_09:e3:4a (c4:23:60:09:e3:4a), Dst: MerossTechno_c3:d7:4e (48:e1:e9:c3:d7:4e)
Internet Protocol Version 6, Src: fe80::84a:d458:42d:50f2, Dst: fe80::4ae1:e9ff:fec3:d74e
User Datagram Protocol, Src Port: 5540, Dst Port: 5540
Matter
  Message Flags: 0x04, Has Source ID, Destination ID Type: Not present
  Session ID: 0x0000
  Security Flags: 0x00, Session Type: Unicast Session
  Message Counter: 0x00df7287
  Source Node ID: 0xb88ecafd71aed5cd
  Protocol Payload
    Exchange Flags: 0x05, Initiator, Reliability
    Protocol Opcode: 0x30
    Exchange ID: 0xdc7f
    Protocol ID: 0x0000
    Application payload (144 bytes)
```

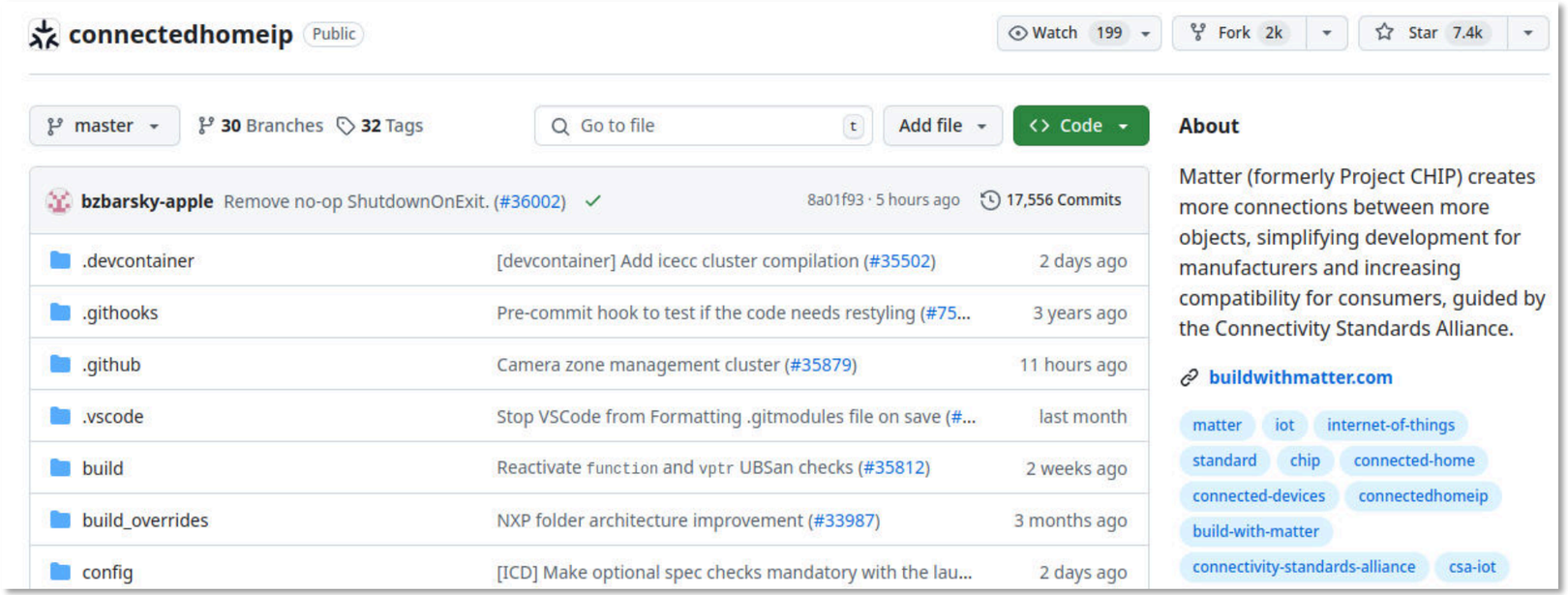
Sigma1

Counter verified for freshness
Not cryptographically protected!

Testbed setup

Controllers and devices

- Controllers (administrative domains) with Google and Apple's technologies
- For a controlled experiment we also added our own controller and Lock-App Matter Wi-Fi device
- Used the Matter reference implementation
- Other variants available:
  - RUST
  - JavaScript



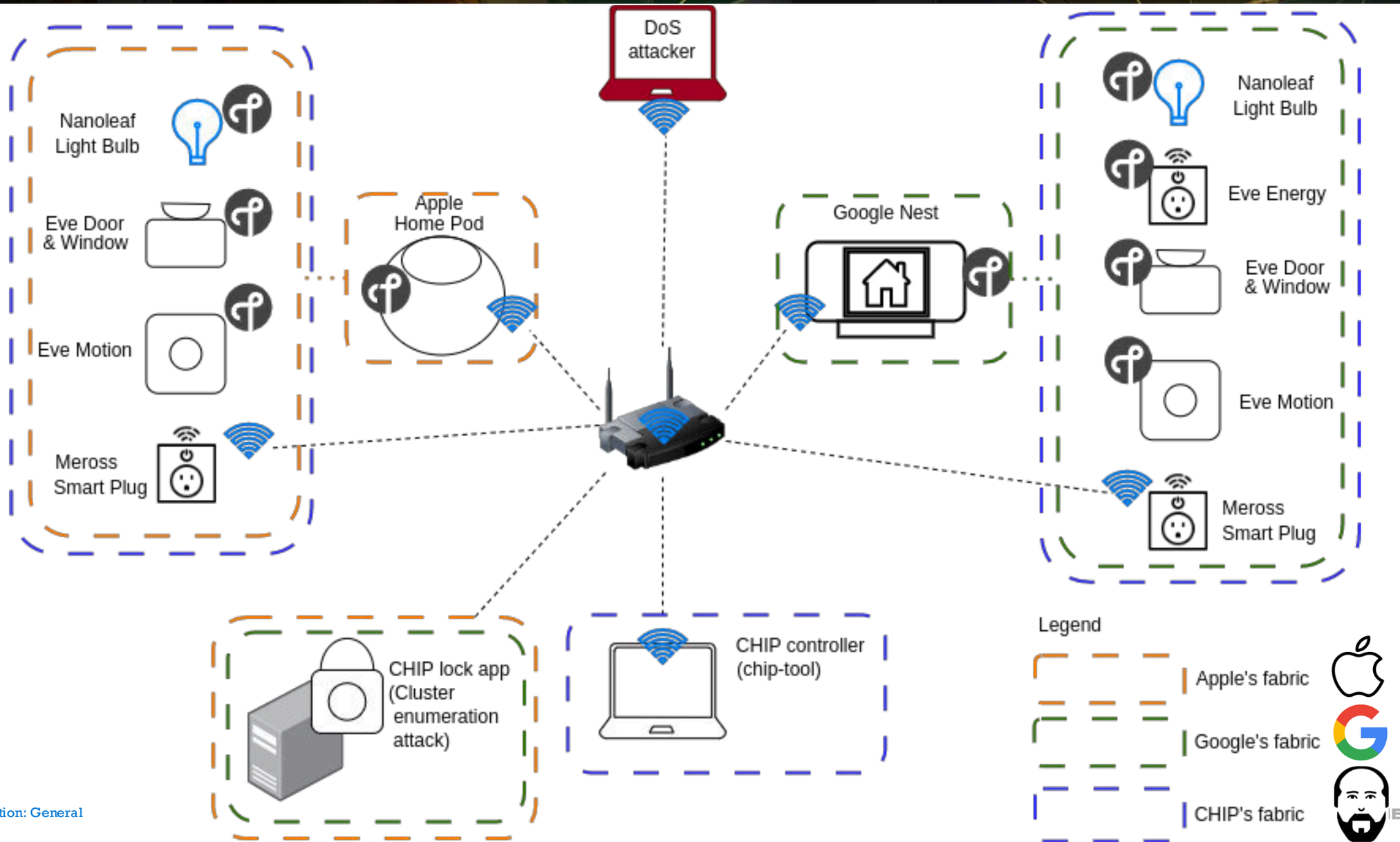
Testbed and components

Border Routers

Matter/Wi-Fi Devices

Matter/Thread Devices

Testbed fabrics and connectivity

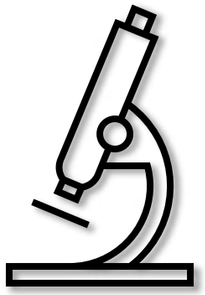


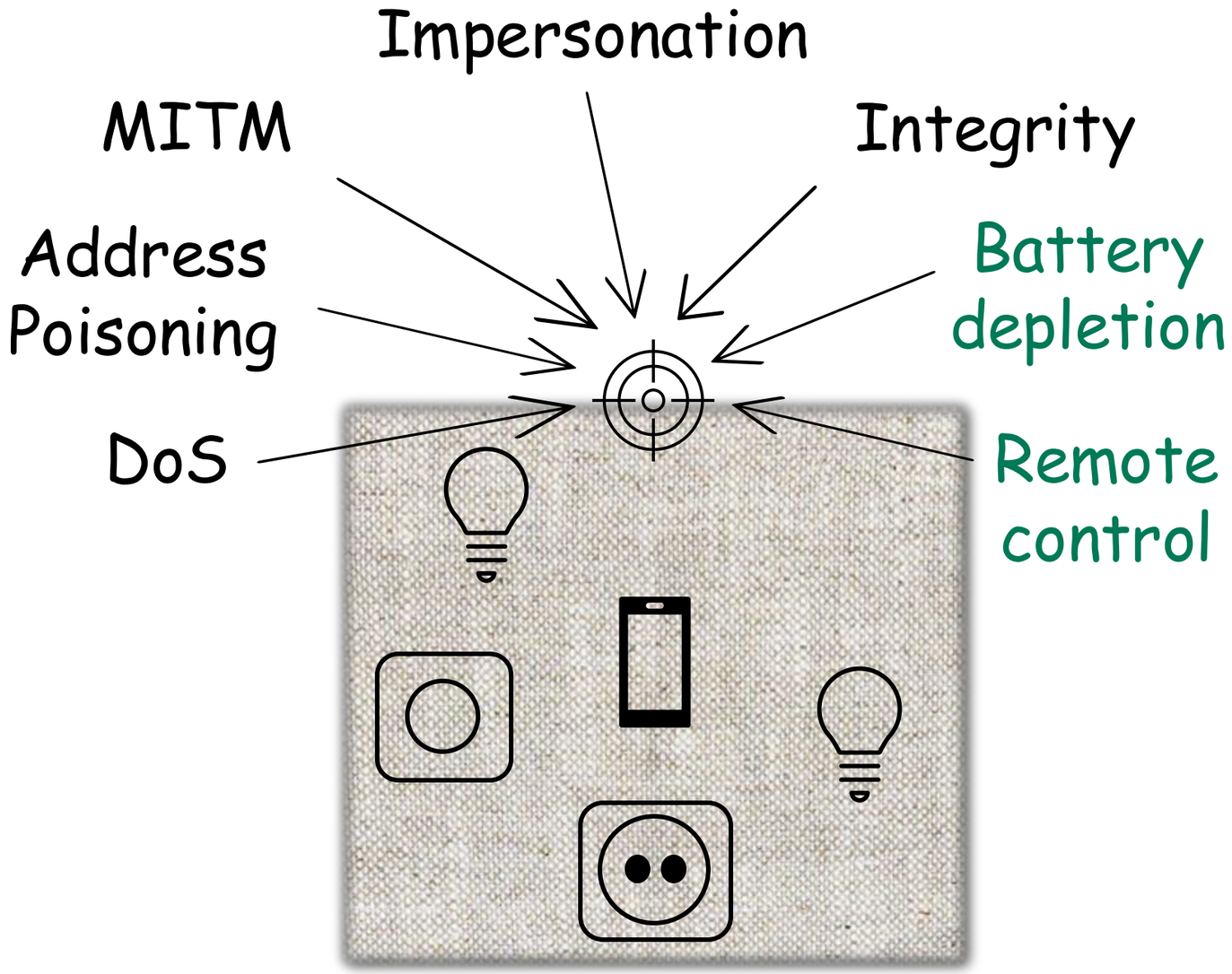
Novel attack class

Delayed DoS: CVE-2024-3297

Start of research

- **Motivating facts** for investigating the CASE protocol:
  - Protocol runs on IPv6/UDP
  - `messageCounter` is not cryptographically protected
  - Handling of `Sigma1` messages is complex





Incentive to test for a DoS attack

Initiator

Responder

Generate: 8 steps

Sigma1

Validate: 7 steps

Generate: 14 steps

Sigma2

Validate: 13 steps

Generate: 9 steps

Sigma3

Validate: 15 steps

structure

destinationId

Gen. Rand resumption ID

Load NOC

Gen. Ephemeral Keys

Gen. Shared Secret

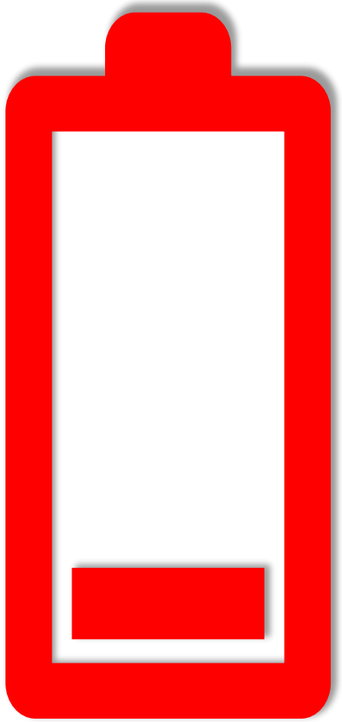
Sign (Certs, PubKeys)

Gen. Random

Gen. S2K

Encrypt (Certs, Sign)

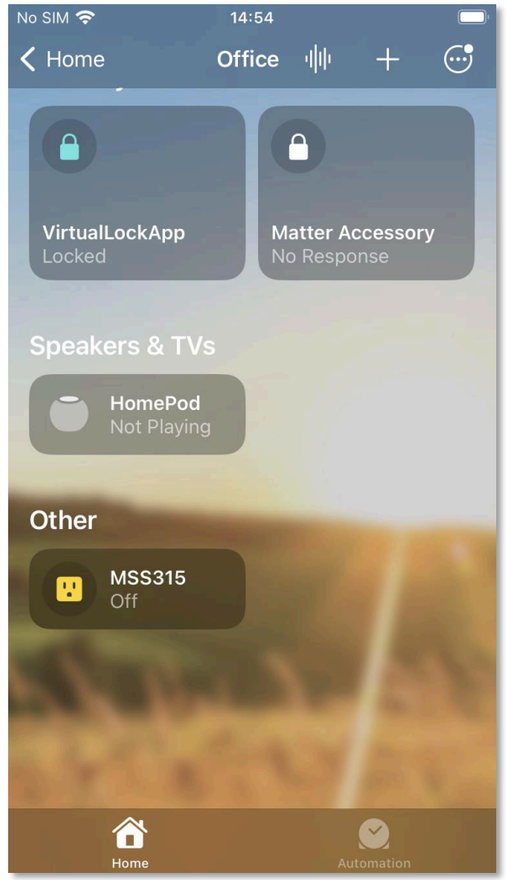
What we expected



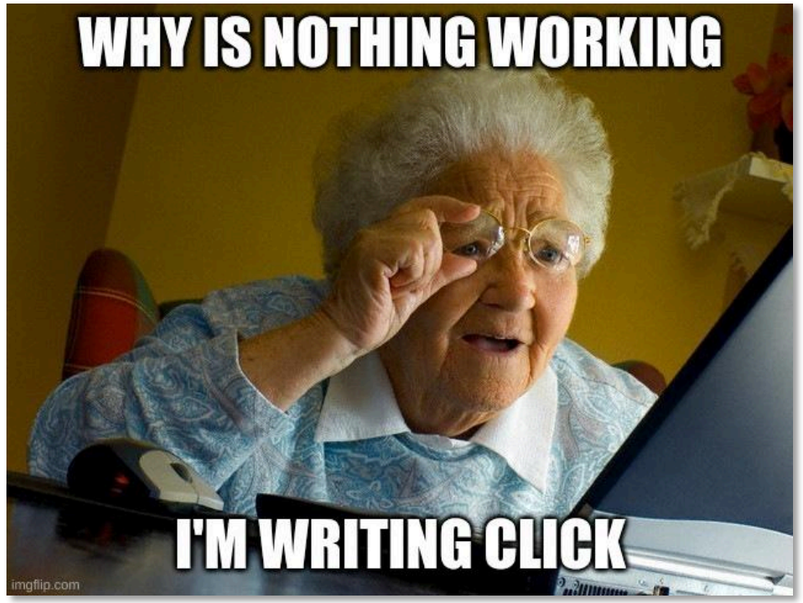
What we got

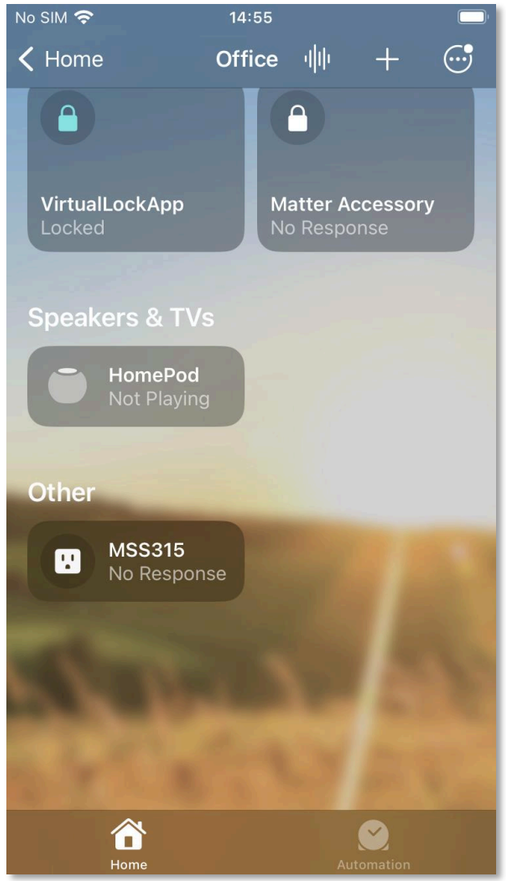
Delayed Denial of Service: DeeDoS [di: dos]

Attack impact: first day



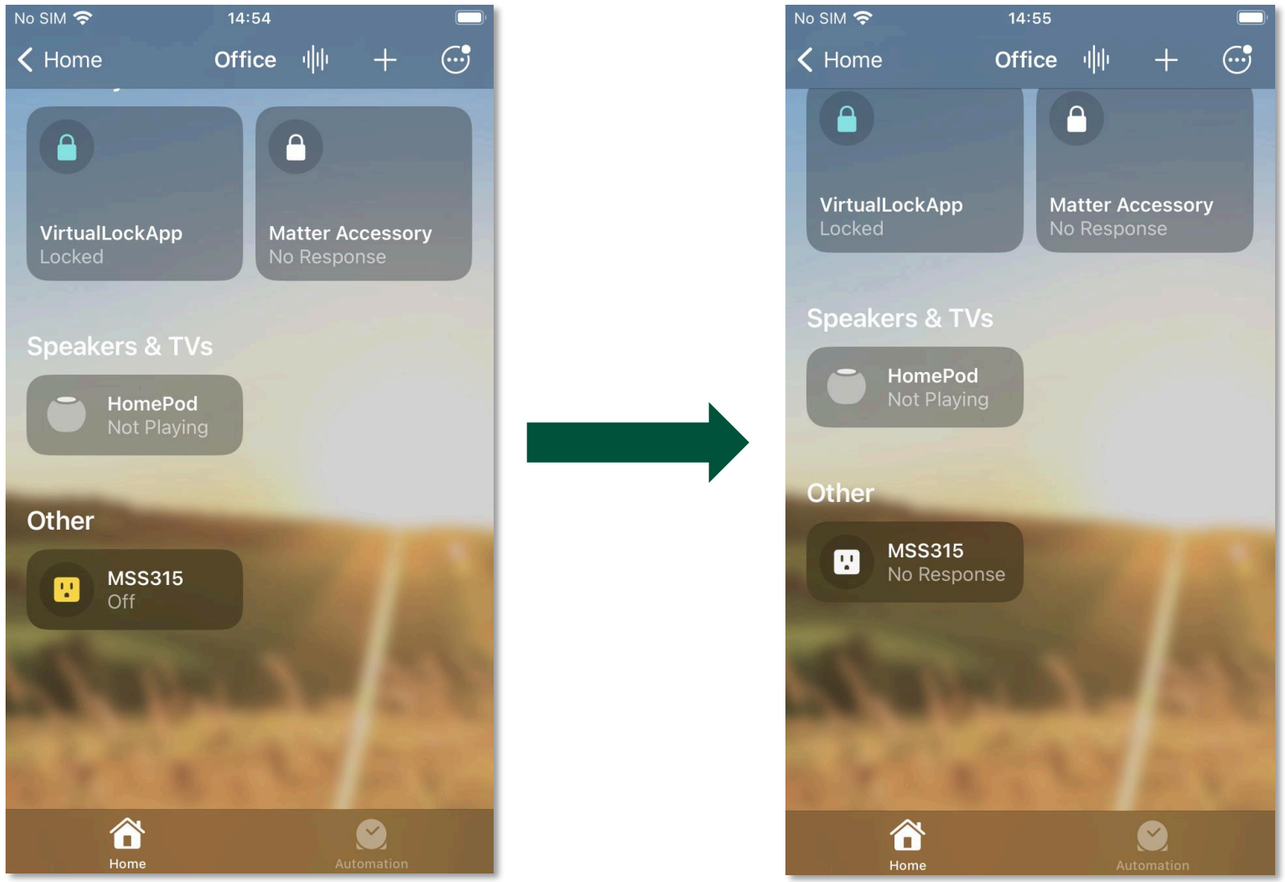
Attack impact: next days





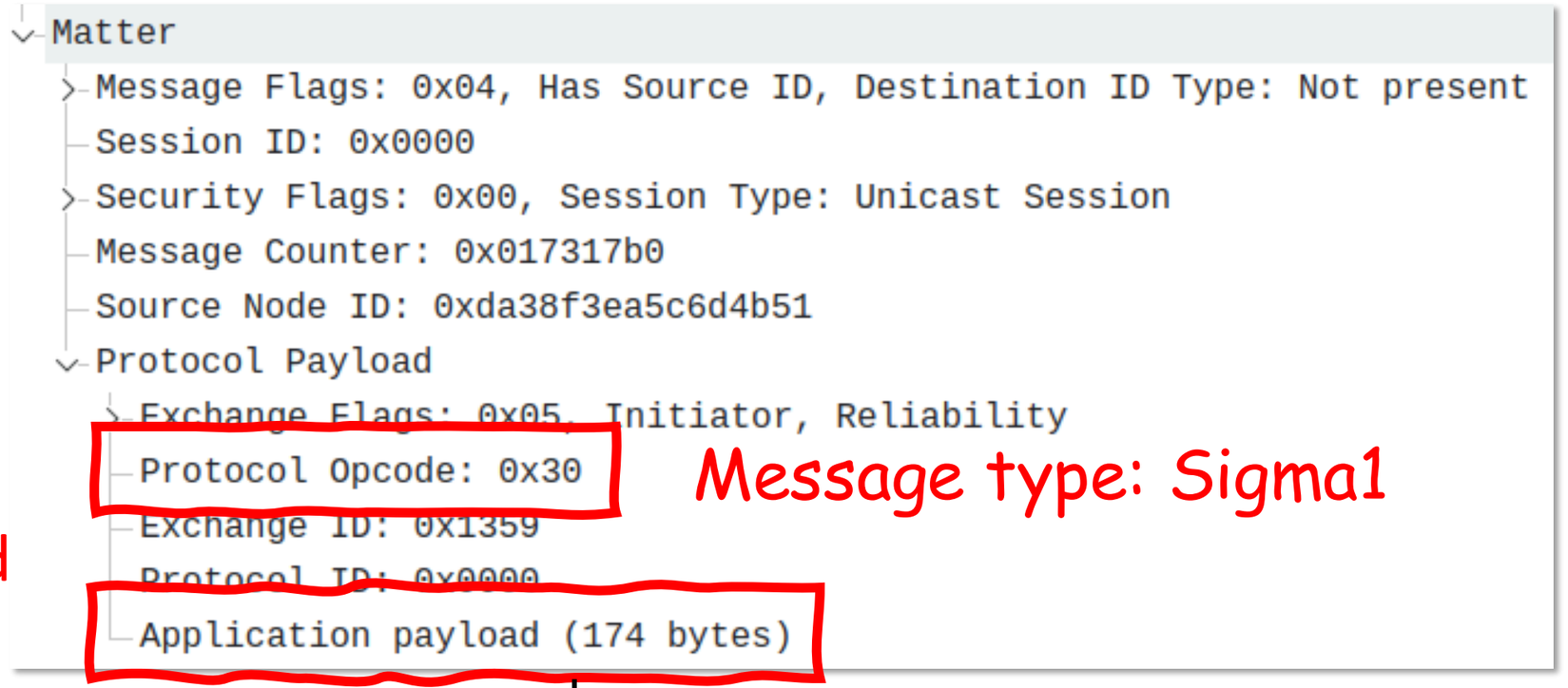
We called this: DeeDoS

- Devices have limited session slots:
  - Session handling is complex, it involves timeouts/retransmissions
  - DeeDoS **depletes session slots**
- Why Delayed?
  - It does not affect existing CASE
- Controllers **are unable to create new CASE**
  - Devices display "No Response"



Step1: get a CASE Sigma1 message

Remember!
CASE Sigma1 is not encrypted

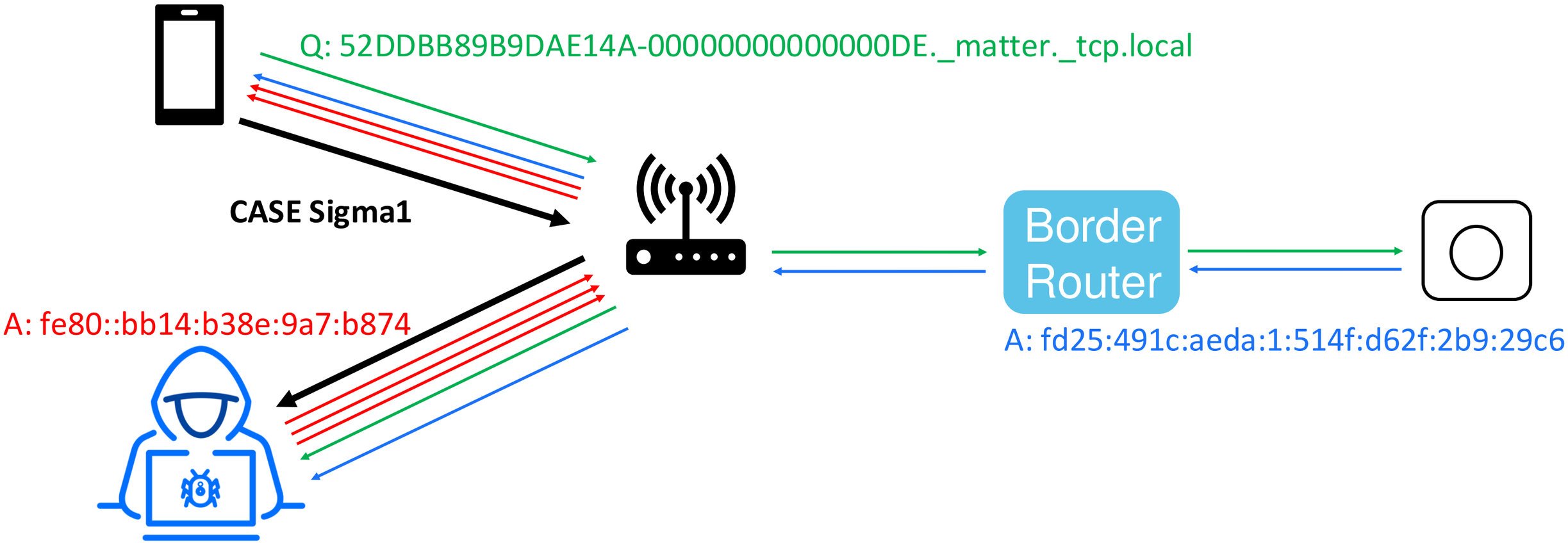


Message type: Sigma1

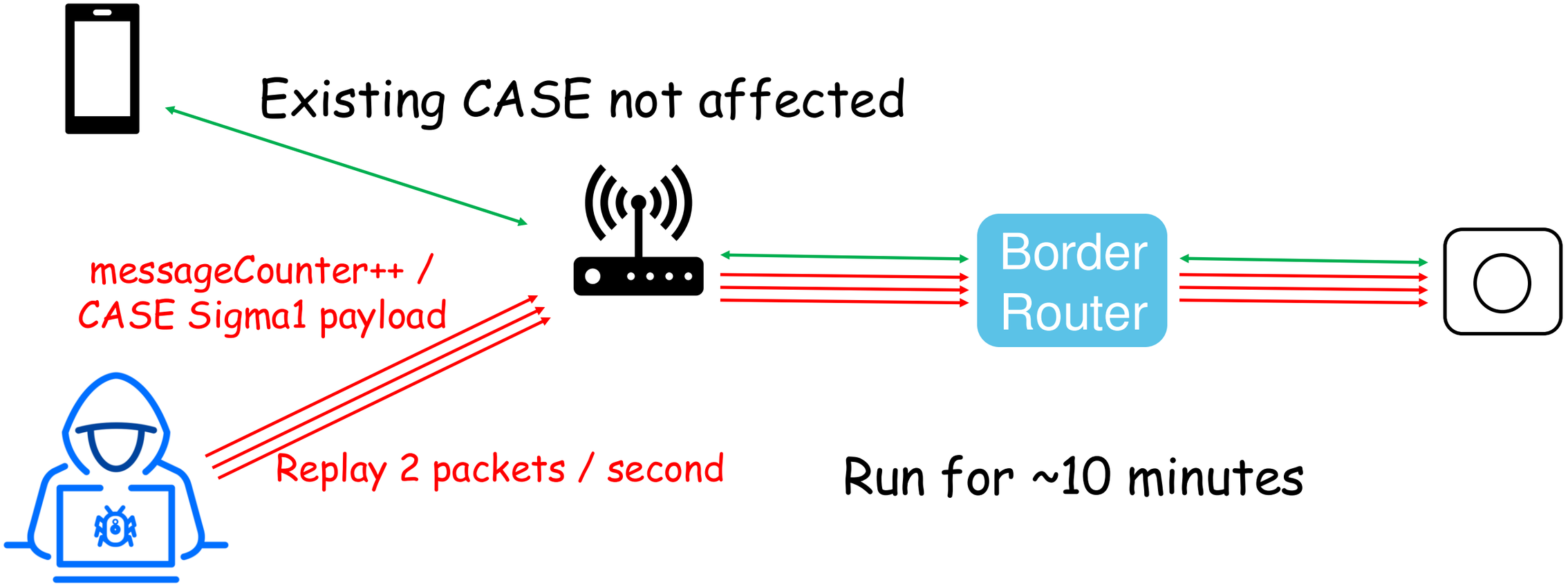
$Random_I$, $SessionId_I$, destId, $EphPubKey_I$ [, ...]

But CASE Sigma1 is not broadcasted...

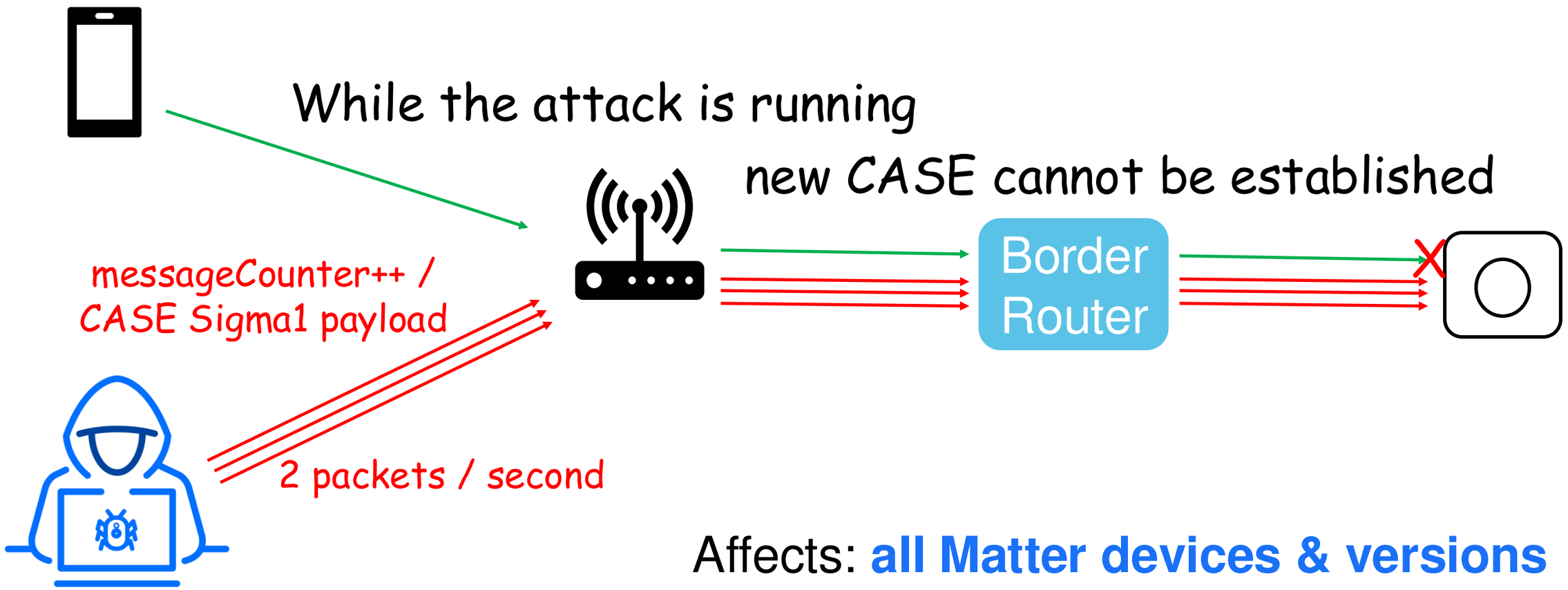
Step1a: mDNS-SD poisoning



Step2: replay

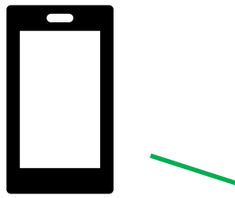


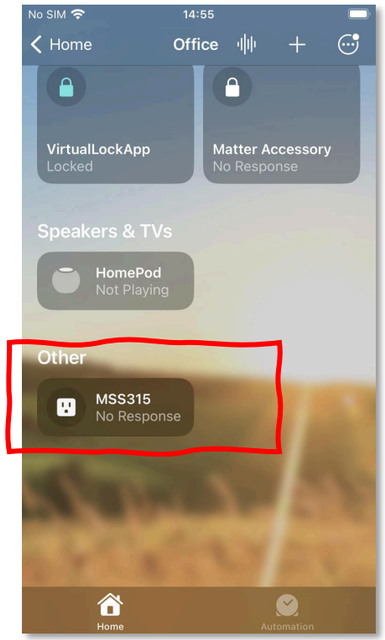
Impact 1



Affects: **all Matter devices & versions**

Impact 2

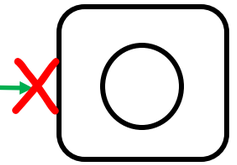




After the attack is stopped

new CASE cannot be established

Border Router



Affects: **all Matter devices with SDK version < 1.1**

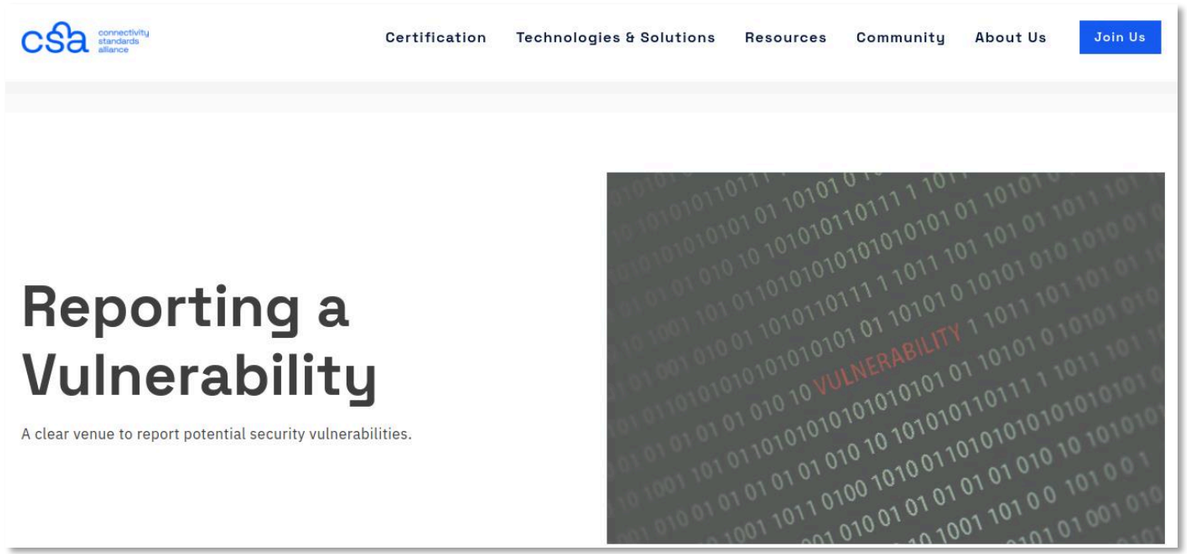
Vulnerability report

- Vulnerability first reported to device manufacturers: Meross, Nanoleaf
  - Response: **issue is in the SDK**

- Vulnerability reported to the Connectivity Standards Alliance (CSA)

meross

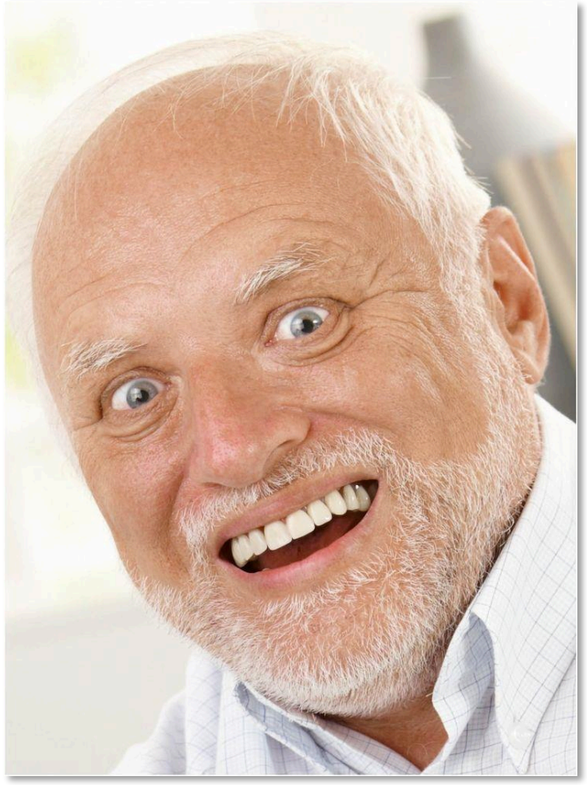
nanoleaf®
Smarter by Design



Response from CSA & solutions

- Vulnerability is indeed in the SDK
  - Affects: **all Matter devices running SDK version < 1.1**
- Solution(s):
  - Update to (at least) Matter 1.1
  - Patch the code (PR #32990)
- While the attack is running, **new CASE cannot be established**
  - Affects: **all Matter devices (all versions)**

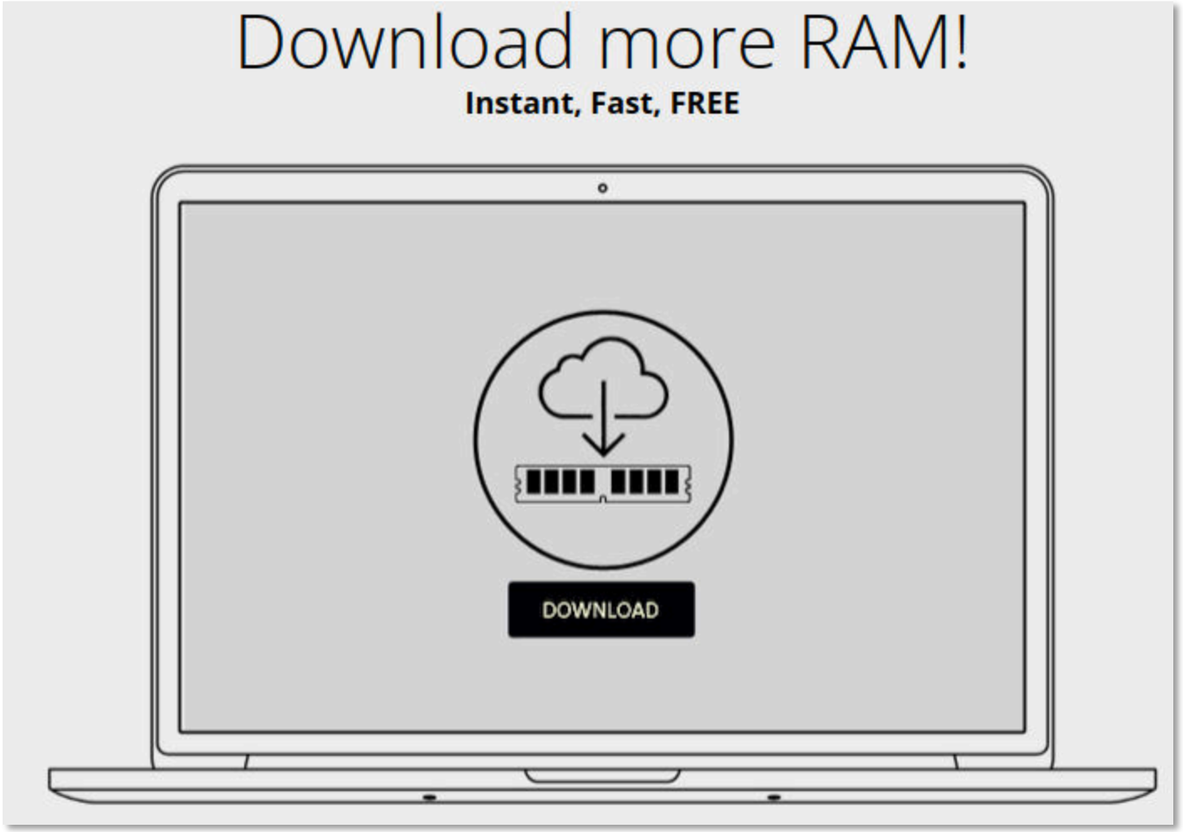
Upgrade to superior Matter version



Upgrade to superior Matter ver...

- Upgrade to Matter 1.1+ leads to improved CASE
- Matter 1.1 compliance: **additional resource requirements**

Matter Version	Access Control Limits	Group limits	Group key limits
1.0	Min. 3 entries / supported fabric	NO minimum req.	1 group key / fabric
1.1	Min. 4 entries / supported fabric	Min. 4 groups / fabric	Min. 3 group keys / fabric


Download more RAM!
Instant, Fast, FREE
DOWNLOAD

Protect the messageCounter (1)

$HMAC_{IPK}(Random_I, FabricRootPK, FabricId, DestNodeId)$

↓

$HMAC_{IPK}(messageCounter, Random_I, FabricRootPK, FabricId, DestNodeId)$

Protect the messageCounter (2)

`messageCounter`: **+4 bytes**

1 Sigma1 packet
(2-3 days)

$HMAC_{IPK}$(messageCounter,
$Random_I$,
FabricRootPK,
FabricId,
DestNodeId)

1200 Sigma1 packets
(2400 days ~ 6.5 years)

Protect the messageCounter (3)

- Matter does not have version negotiation!
- This would break backward-compatibility

So what is the solution?

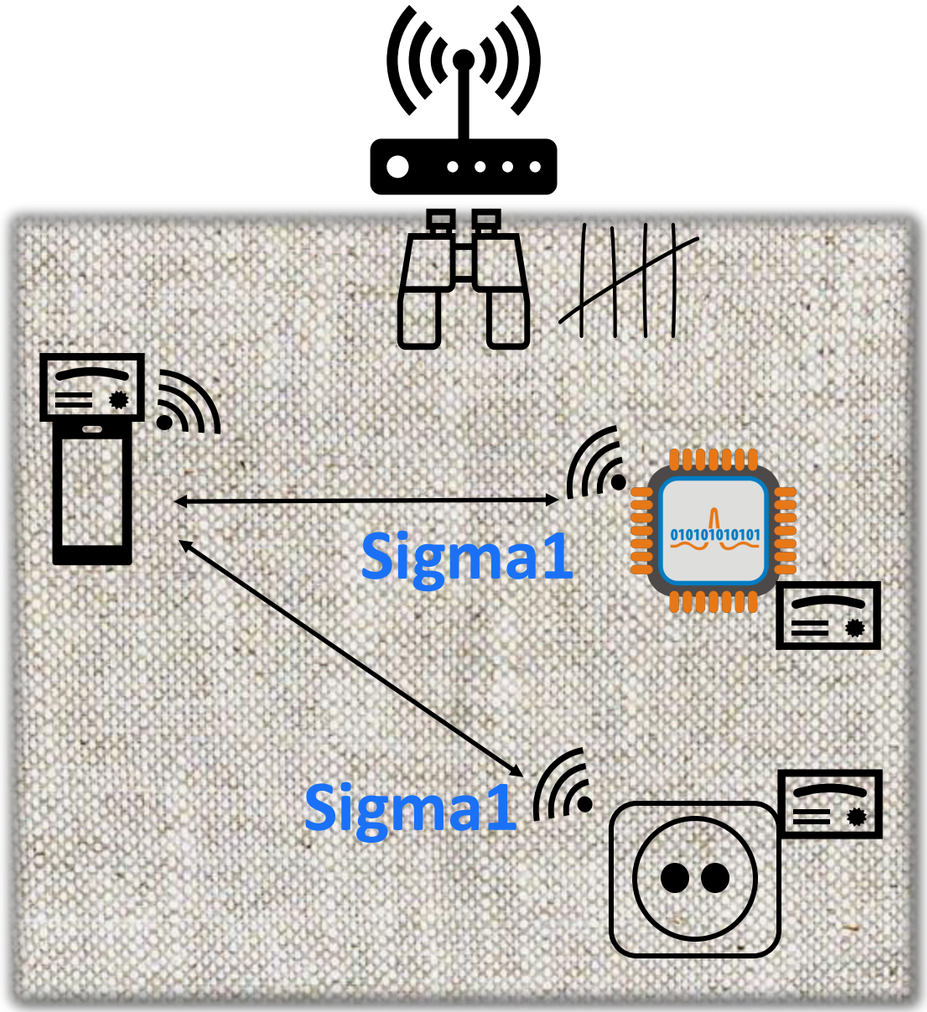
- Unable to upgrade?
- Unable to change the specification & integrate counter protection?

- **Monitor & detect!**

Monitor & detect

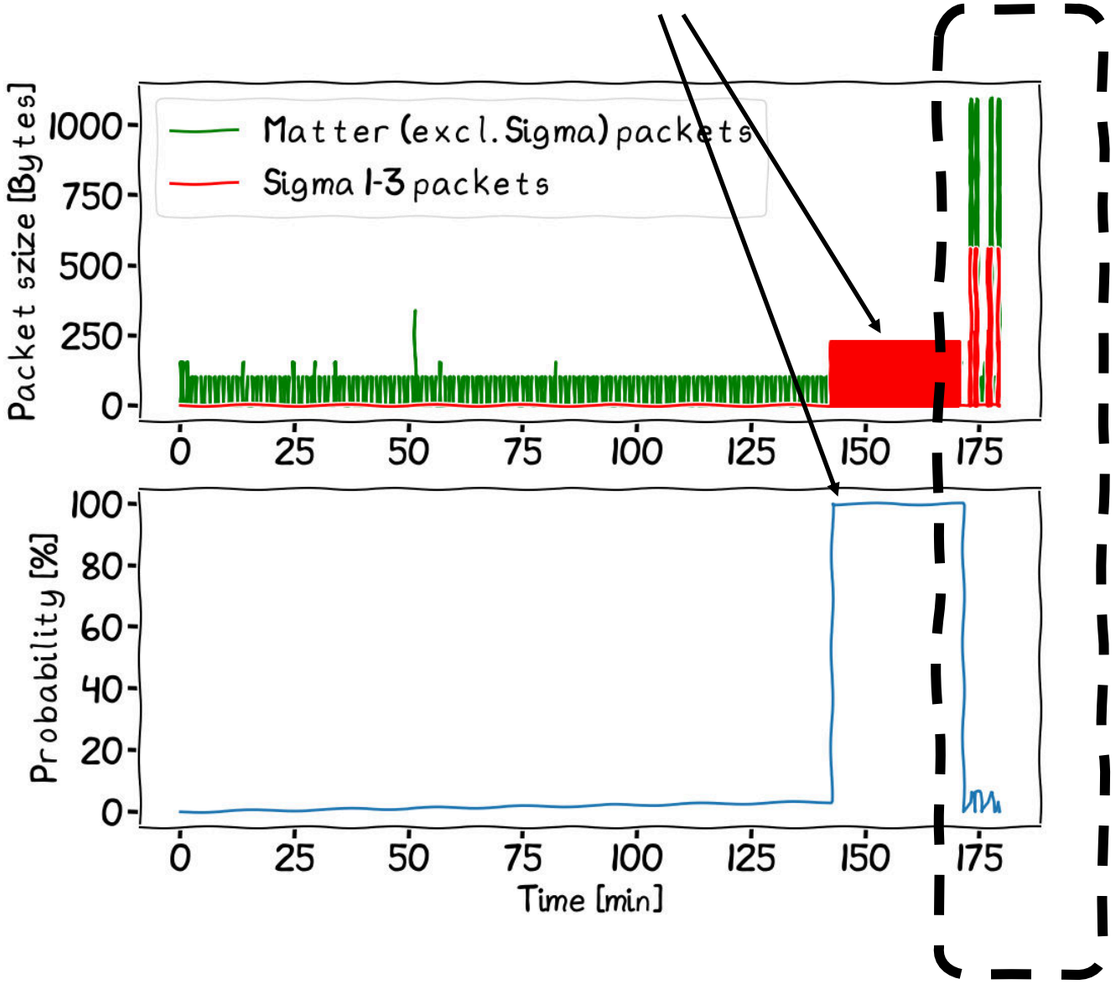
- Matter packet headers are not encrypted
- We can **count CASE Sigma1** packets
- Simple packet statistics can **detect the DeeDoS attack**

```
Foreach (TimeWnd)
    If (count(Sigma1, TimeWnd) >= Th)
        Raise_DeeDoS();
    Endif
Endforeach
```

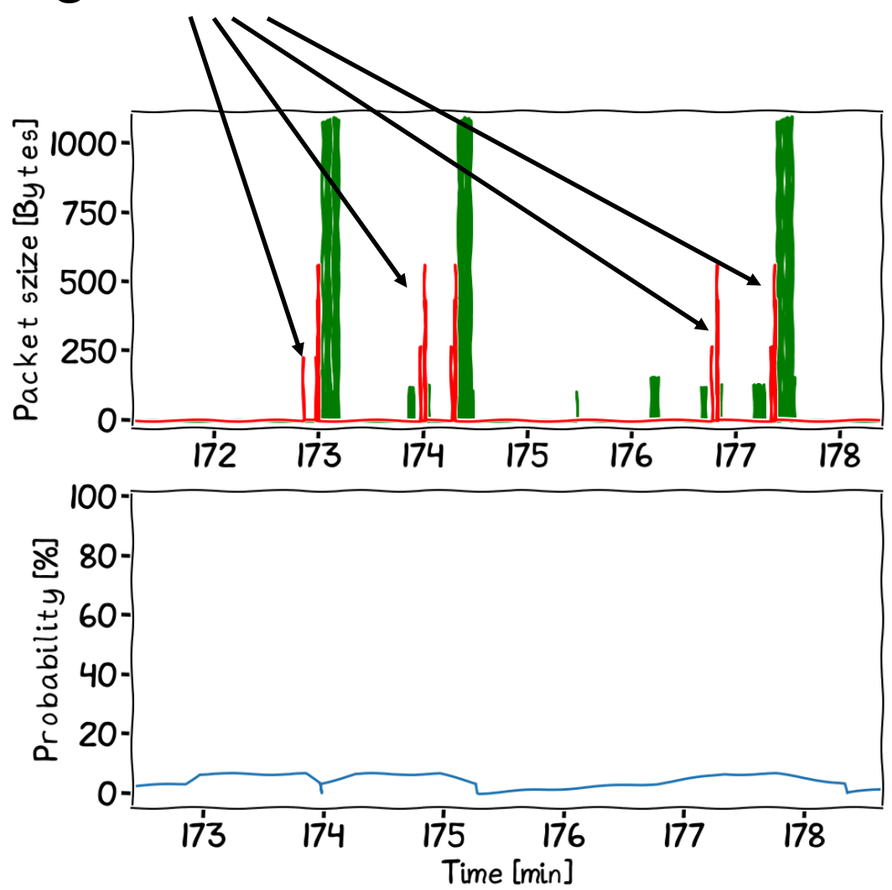


Count CASE Sigma1 packets

DeeDoS attack & detection

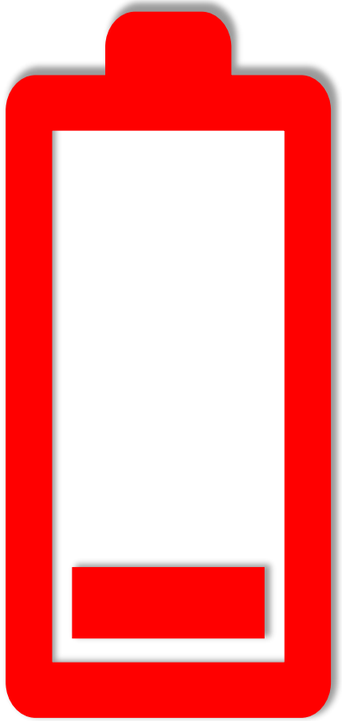


CASE Sigma1 on normal reconnection



What about battery depletion?

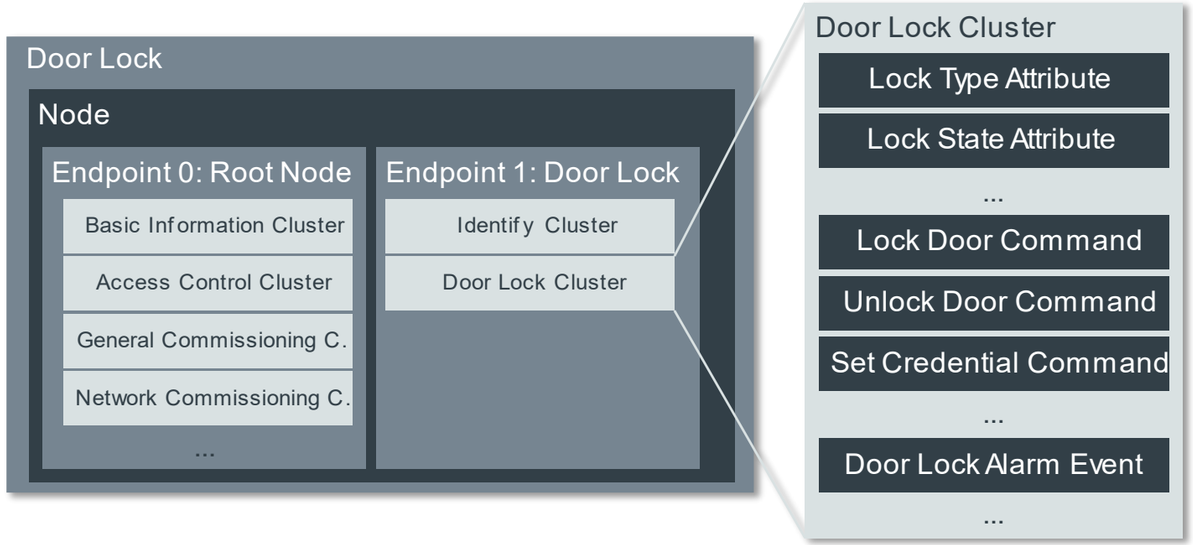
- Feasible, it takes **~ 6 – 12 days** (depending on device)
- Tested with several devices by replaying **15 Sigma1 packets / second**
- Attack has an immense **impact on usability**, important that it is stopped early!



Feature enumeration: CVE-2024-3454

Starting point

- The Matter fabric is a closed and protected ecosystem
- Within a fabric, devices do not have access to each other's **clusters** and **attributes**



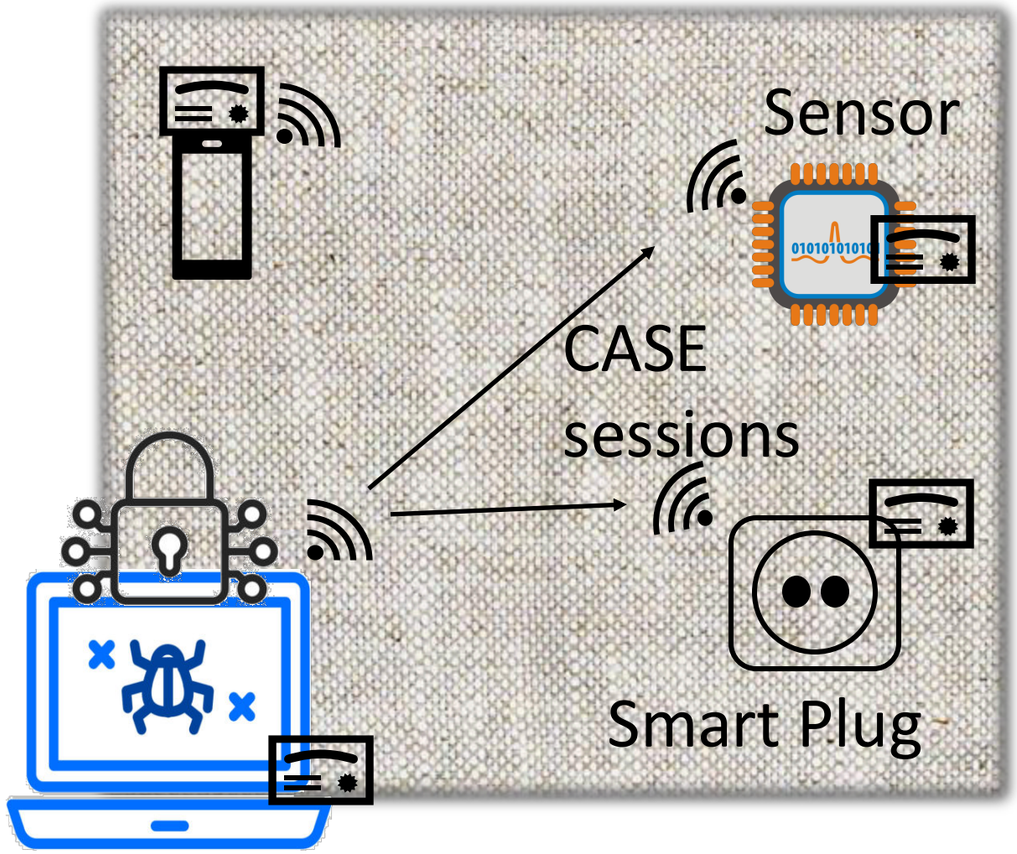
Source: https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/data_model.html

Steps to follow

1. Create a Virtual device

2. Add to target fabric

3. Interrogate other devices



Step #1: create a virtual device

connectedhomeip / examples / lock-app / **linux** /

README.md

Lock Application for Linux

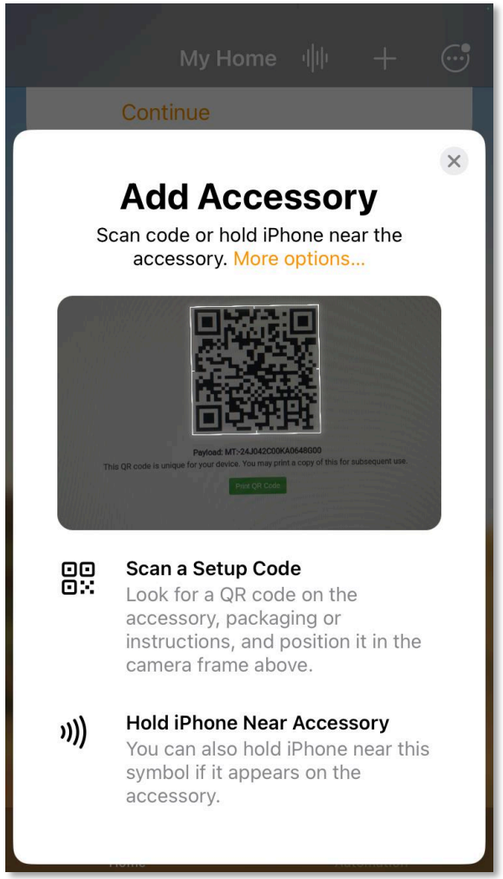
Application that showcases abilities of the Door Lock Cluster.

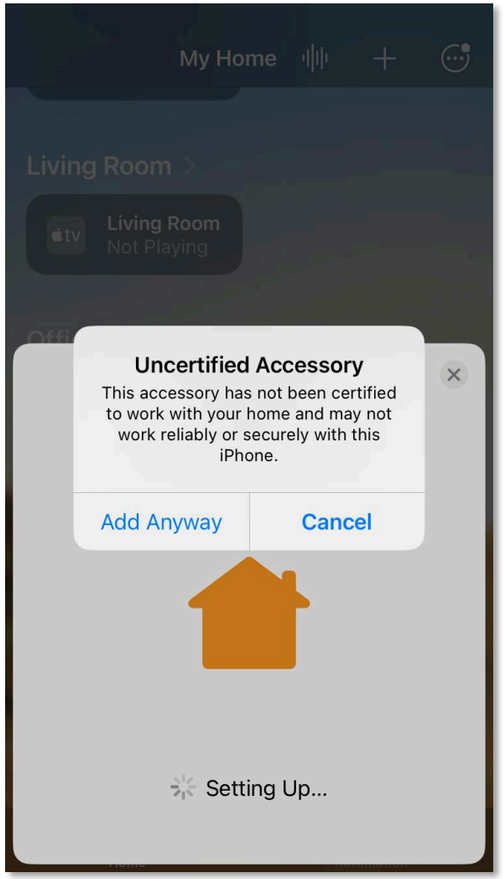
```
ndalone/chip-lock-app
nuxStorage::Init: Using KVS config file: /tmp/chip_kvs
nuxStorage::Init: Using KVS config file: /tmp/chip_kvs
nuxStorage::Init: Attempt to re-initialize with KVS config file: /tmp/chip_kvs
nuxStorage::Init: Using KVS config file: /tmp/chip_factory.ini
nuxStorage::Init: Using KVS config file: /tmp/chip_config.ini
nuxStorage::Init: Using KVS config file: /tmp/chip_counters.ini
g settings to file (/tmp/chip_counters.ini-HiMisk)
d tmp file to file (/tmp/chip_counters.ini)
t: chip-counters/reboot-count = 5 (0x5)
hernet interface: eno1
the primary Ethernet interface:eno1
[1729250960.663978][198718:198718] CHIP:DL: Failed to get WiFi interface
[1729250960.663982][198718:198718] CHIP:DL: Failed to reset WiFi statistic counts
[1729250960.663987][198718:198718] CHIP:SPT: *** WARNING: Using temporary passcode 20202021 due to no neither --passcode or --spake2p-
y and will disappear. Please update your scripts to explicitly configure onboarding credentials. ***
[1729250960.663990][198718:198718] CHIP:SPT: *** WARNING: Using temporary test discriminator 3840 due to --discriminator not given on
ase update your scripts to explicitly configure discriminator. ***
[1729250960.663993][198718:198718] CHIP:SPT: PASE PBKDF iterations set to 1000
[1729250960.663996][198718:198718] CHIP:SPT: LinuxCommissionableDataProvider didn't get a PASE salt, generating one.
[1729250960.665414][198718:198718] CHIP:DL: Device Configuration:
[1729250960.665420][198718:198718] CHIP:DL:   Serial Number: TEST_SN
[1729250960.665424][198718:198718] CHIP:DL:   Vendor Id: 65521 (0xFFF1)
[1729250960.665427][198718:198718] CHIP:DL:   Product Id: 32769 (0x8001)
[1729250960.665430][198718:198718] CHIP:DL:   Product Name: TEST_PRODUCT
[1729250960.665433][198718:198718] CHIP:DL:   Hardware Version: 0
[1729250960.665435][198718:198718] CHIP:DL:   Setup Pin Code (0 for UNKNOWN/ERROR): 20202021
[1729250960.665437][198718:198718] CHIP:DL:   Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0xF00)
[1729250960.665441][198718:198718] CHIP:DL:   Manufacturing Date: (not set)
[1729250960.665443][198718:198718] CHIP:DL:   Device Type: 65535 (0xFFFF)
[1729250960.665445][198718:198718] CHIP:-: ==== Onboarding payload for Standard Commissioning Flow ====
[1729250960.665450][198718:198718] CHIP:SVR: SetupQRCode: [MT:-24J042C00KA0648G00]
[1729250960.665453][198718:198718] CHIP:SVR: Copy/paste the below URL in a browser to see the QR Code:
[1729250960.665456][198718:198718] CHIP:SVR: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A-24J042C00KA0648G00
```

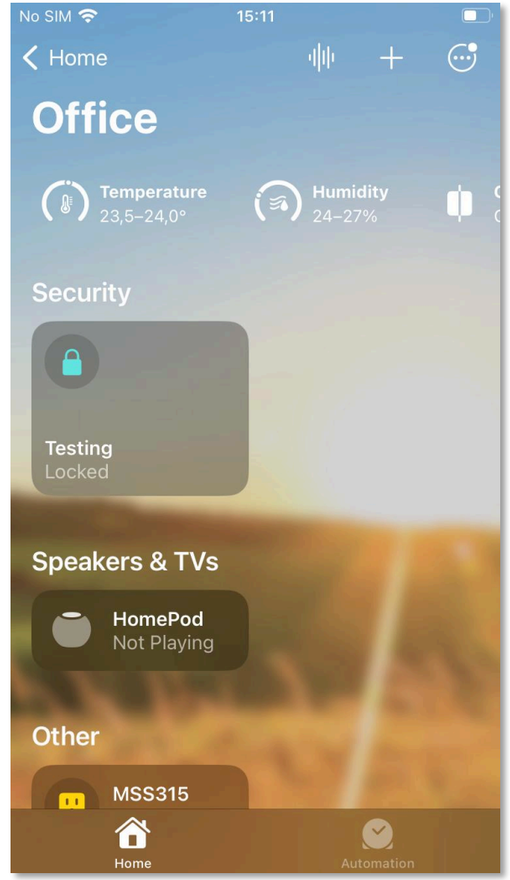
Step #2: add to target fabric


```
bgenge@bgenge-d:~/connectedhomeip$ ./out/debug/standalone/chip-lock-app
[1729250960.662813][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_kvs
[1729250960.663152][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_kvs
[1729250960.663158][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Attempt to re-initialize with KVS config file: /tmp/chip_kvs
[1729250960.663275][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_factory.ini
[1729250960.663314][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_config.ini
[1729250960.663328][198718:198718] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_counters
[1729250960.663443][198718:198718] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-HiMisk)
[1729250960.663547][198718:198718] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1729250960.663554][198718:198718] CHIP:DL: NVS set: chip-counters/reboot-count = 5 (0x5)
[1729250960.663707][198718:198718] CHIP:DL: Got Ethernet interface: eno1
[1729250960.663829][198718:198718] CHIP:DL: Found the primary Ethernet interface:eno1
[1729250960.663978][198718:198718] CHIP:DL: Failed to get WiFi interface
[1729250960.663982][198718:198718] CHIP:DL: Failed to reset WiFi statistic counts
[1729250960.663987][198718:198718] CHIP:SPT: *** WARNING: Using temporary passcode 20202021 due to no neither
y and will disappear. Please update your scripts to explicitly configure onboarding credentials. ***
[1729250960.663990][198718:198718] CHIP:SPT: *** WARNING: Using temporary test discriminator 3840 due to --dis
ase update your scripts to explicitly configure discriminator. ***
[1729250960.663993][198718:198718] CHIP:SPT: PASE PBKDF iterations set to 1000
[1729250960.663996][198718:198718] CHIP:SPT: LinuxCommissionableDataProvider didn't get a PASE salt, generati
[1729250960.665414][198718:198718] CHIP:DL: Device Configuration:
[1729250960.665420][198718:198718] CHIP:DL:   Serial Number: TEST_SN
[1729250960.665424][198718:198718] CHIP:DL:   Vendor Id: 65521 (0xFFF1)
[1729250960.665427][198718:198718] CHIP:DL:   Product Id: 32769 (0x8001)
[1729250960.665430][198718:198718] CHIP:DL:   Product Name: TEST_PRODUCT
[1729250960.665433][198718:198718] CHIP:DL:   Hardware Version: 0
[1729250960.665435][198718:198718] CHIP:DL:   Setup Pin Code (0 for UNKNOWN/ERROR): 20202021
[1729250960.665437][198718:198718] CHIP:DL:   Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0xF00)
[1729250960.665441][198718:198718] CHIP:DL:   Manufacturing Date: (not set)
[1729250960.665443][198718:198718] CHIP:DL:   Device Type: 65535 (0xFFFF)
[1729250960.665445][198718:198718] CHIP:-: ==== Onboarding payload for Standard Commissioning Flow ====
[1729250960.665450][198718:198718] CHIP:SVR: SetupQRCode: [MT:-24J042C00KA0648G00]
[1729250960.665453][198718:198718] CHIP:SVR: Copy/paste the below URL in a browser to see the QR Code:
[1729250960.665456][198718:198718] CHIP:SVR: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A-24J042C00KA0648G00
```

Step #2: add to target fabric



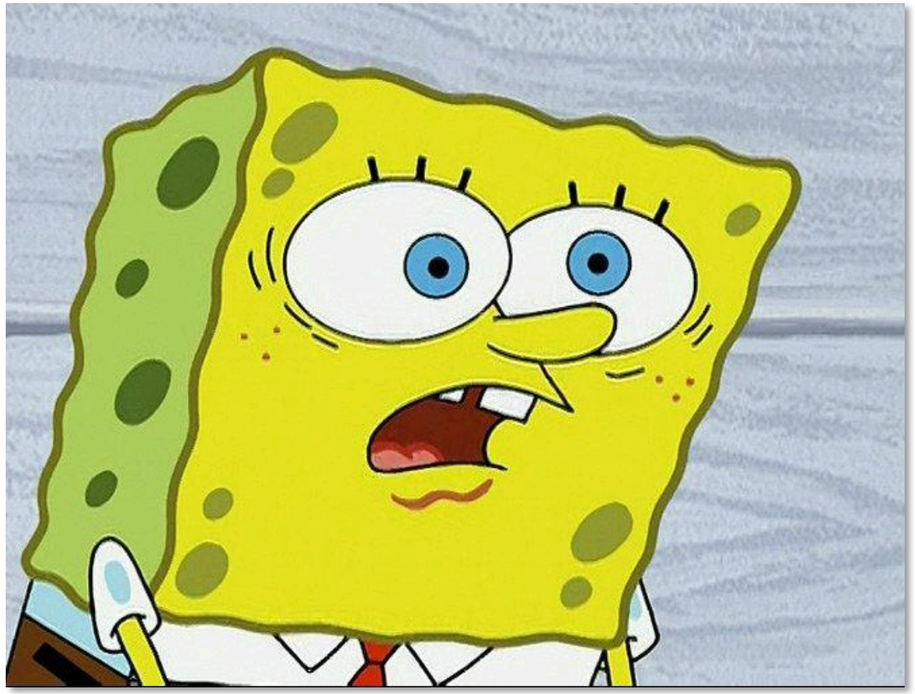




Step #3: interrogate other devices

Should be straightforward, right?

Step #3: interrogate other devices



- **No documentation** on API to open CASE
- **No documentation** on API to interrogate clusters / attributes

Change the lock application (1)

1. Add necessary include files

```
#include <app/EventManagement.h>
#include <app/InteractionModelEngine.h>
#include <app/server/Server.h>
#include <messaging/ExchangeContext.h>
#include <messaging/ExchangeMgr.h>
#include <controller/InvokeInteraction.h>
#include <controller/ReadInteraction.h>
```

2. Add global variables

```
static Messaging::ExchangeManager *gXMgr;
static unsigned long long gTargetNodeId = 0;
```

Change the lock application (2)

3. Add callback functions

```
void HandleDeviceConnected(void *context,
                           Messaging::ExchangeManager &exchangeMgr,
                           const SessionHandle &sessionHandle)
{
    // Cluster / Attribute interrogation
}

void HandleDeviceConnectionFailure(void *context,
                                   const ScopedNodeId &peerId,
                                   CHIP_ERROR err)
{
    // Error handling
}

Callback::Callback<OnDeviceConnected>
    gOnConnectedCallback(HandleDeviceConnected, nullptr);

Callback::Callback<OnDeviceConnectionFailure>
    gOnConnectionFailureCallback(HandleDeviceConnectionFailure, nullptr);
```

Change the lock application (3)

4. Change FromJSON()

```
if (params.isMember("NodeId")) {
    gTargetNodeId = strtoull(params["NodeId"].asString().c_str(),
                             NULL,
                             16);
}
```

5. Change HandleCommand()

```
if (self->mCommandName == "RunScan") {
    gXMgr = InteractionModelEngine::GetInstance()->GetExchangeManager();
    Server::GetInstance().GetCASESessionManager()->FindOrEstablishSession(
        ScopedNodeId(gTargetNodeId, 0x01),
        &gOnConnectedCallback,
        &gOnConnectionFailureCallback);
}
```

Change the lock application (4)

6. Add scan code

```
auto onSuccess = [](const ConcreteDataAttributePath &attributePath, const auto &dataResponse) {
    ChipLogProgress(NotSpecified, "Read attribute successful!");
};

auto onFailure = [](const ConcreteDataAttributePath *attributePath, CHIP_ERROR error) {
    ChipLogError(NotSpecified, "Read attribute failed: %" CHIP_ERROR_FORMAT, error.Format());
};

Controller::ReadAttribute<Clusters::OnOff::Attributes::OnOff::TypeInfo>(
    gXMgr, sessionHandle, 0x01, onSuccess, onFailure);

Controller::ReadAttribute<Clusters::LevelControl::Attributes::CurrentLevel::TypeInfo>(
    gXMgr, sessionHandle, 0x01, onSuccess, onFailure);

Controller::ReadAttribute<Clusters::ColorControl::Attributes::CurrentHue::TypeInfo>(
    gXMgr, sessionHandle, 0x01, onSuccess, onFailure);

Controller::ReadAttribute<Clusters::OccupancySensing::Attributes::Occupancy::TypeInfo>(
    gXMgr, sessionHandle, 0x01, onSuccess, onFailure);

Controller::ReadAttribute<Clusters::BooleanState::Attributes::StateValue::TypeInfo>(
    gXMgr, sessionHandle, 0x01, onSuccess, onFailure);
```

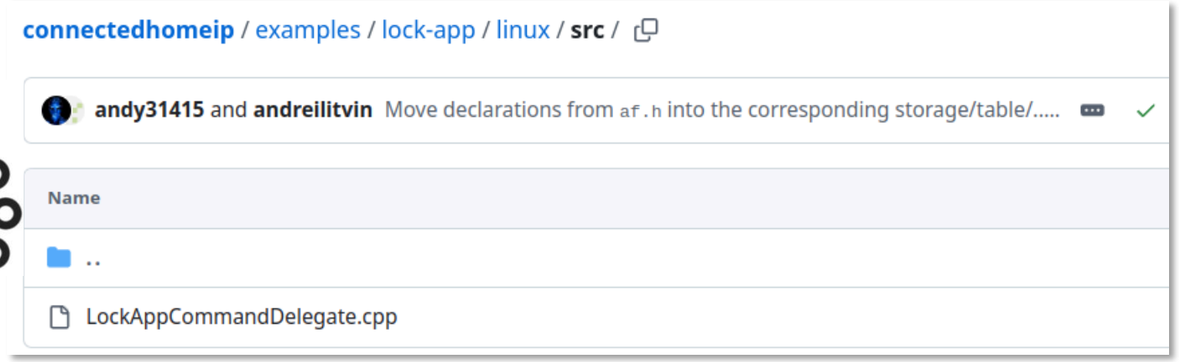
Run the lock application

1. Run lock app

```
./chip-lock-app &
```

2. Issue Scan command

```
pid=$(pidof ./chip-lock-app)
NODE_ID="00000000DA41E4CF"
CMD="{\"Cmd\": \"RunScan\", \"Params\": { \"EndpointId\": 1, \"Node\":
\"$NODE_ID\" } }"
echo $CMD > /tmp/chip_lock_app_fifo-"$pid"
```



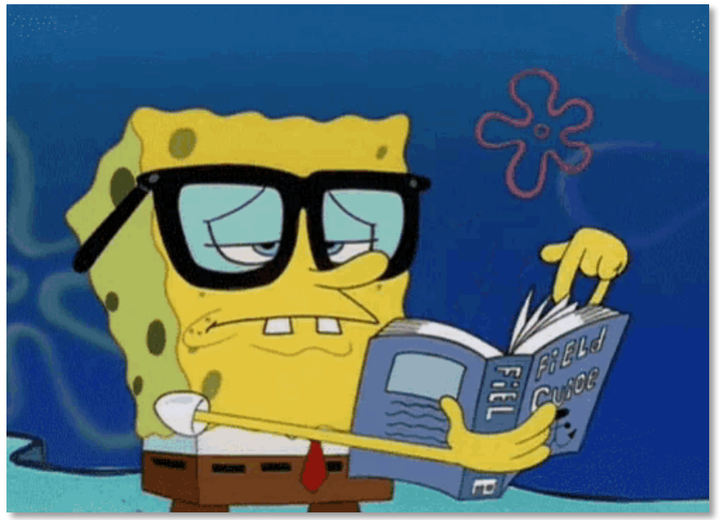
Did you expect to get access?

- Unfortunately, we only get ERROR after ERROR!
- Matter's built-in Access Control List denies, by default, all access to clusters and attributes

```
CHIP:-: Read attribute failed: IM Error 0x000005C3: General error: 0xc3
(UNSUPPORTED_CLUSTER)
CHIP:-: Read attribute failed: IM Error 0x000005C3: General error: 0xc3
(UNSUPPORTED_CLUSTER)
CHIP:-: Read attribute failed: IM Error 0x0000057E: General error: 0x7e
(UNSUPPORTED_ACCESS)
CHIP:-: Read attribute failed: IM Error 0x000005C3: General error: 0xc3
(UNSUPPORTED_CLUSTER)
...
```

All is not loosed!

- Observe the two errors:
  - `UNSUPPORTED_CLUSTER`
  - `UNSUPPORTED_ACCESS`



- If the path indicates a node that is unsupported, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_NODE Status Code.
- Else if the path indicates an endpoint that is unsupported, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_ENDPOINT Status Code.
- Else if the path indicates a cluster that is unsupported, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_CLUSTER Status Code.
- Else if the path indicates an attribute or attribute data field that is unsupported, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_ATTRIBUTE Status Code with the Path field indicating the first unsupported data field (not the entire attribute data path).
- Else if the path indicates attribute data that is not readable, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_READ Status Code.
- Else if reading from the attribute in the path requires a privilege that is not granted to access the cluster in the path, an AttributeStatusIB SHALL be generated with the UNSUPPORTED_ACCESS Status Code.

Source: Matter 1.2 specification

Order of verifications

What can we infer?

- Cluster: `OnOff`
  - Attribute: `OnOff`

- Cluster: `BooleanState`
  - Attribute: `StateValue`

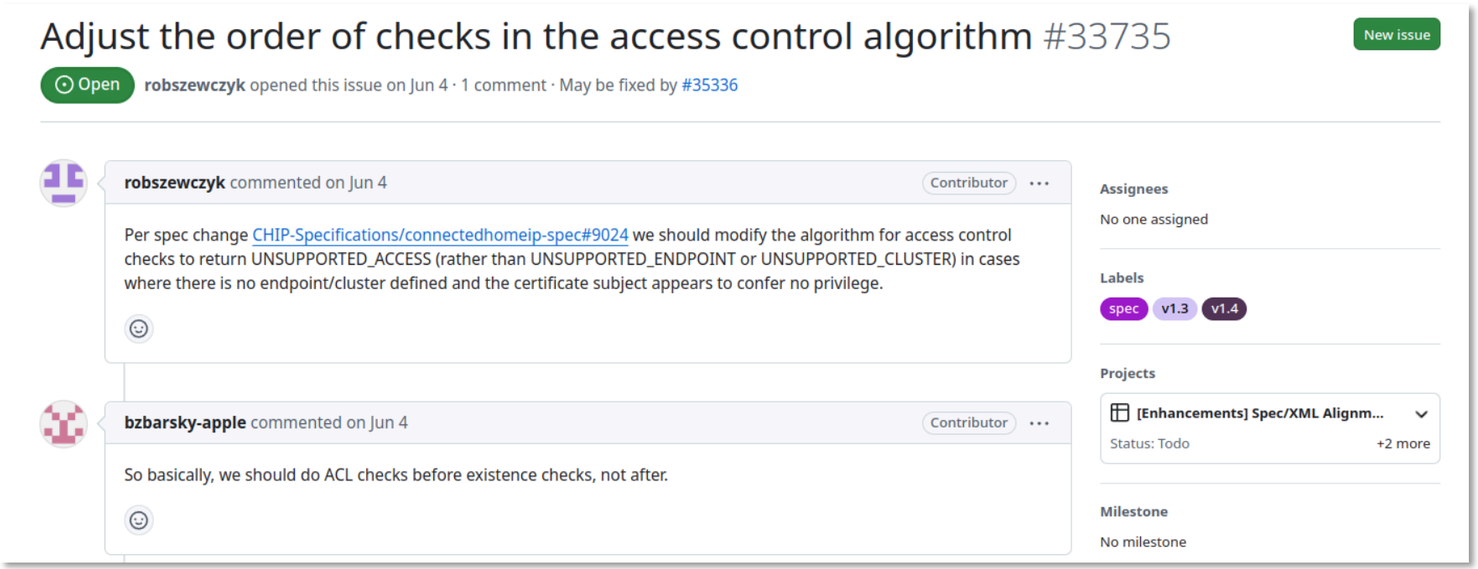
- Cluster: `OnOff`
  - Attribute: `OnOff`
- Cluster: LevelControl
  - Attribute: CurrentLevel
- Cluster: ColorControl
  - Attribute: CurrentHue

- Cluster: `OnOff`
  - Attribute: `OnOff`

Vulnerability report & response

- Detailed report submitted to CSA
  - Response: vulnerability is applicable to **all Matter versions and devices on the market**

- Resulted in a change in the Matter specification!



https://github.com/project-chip/connectedhomeip/issues/33735

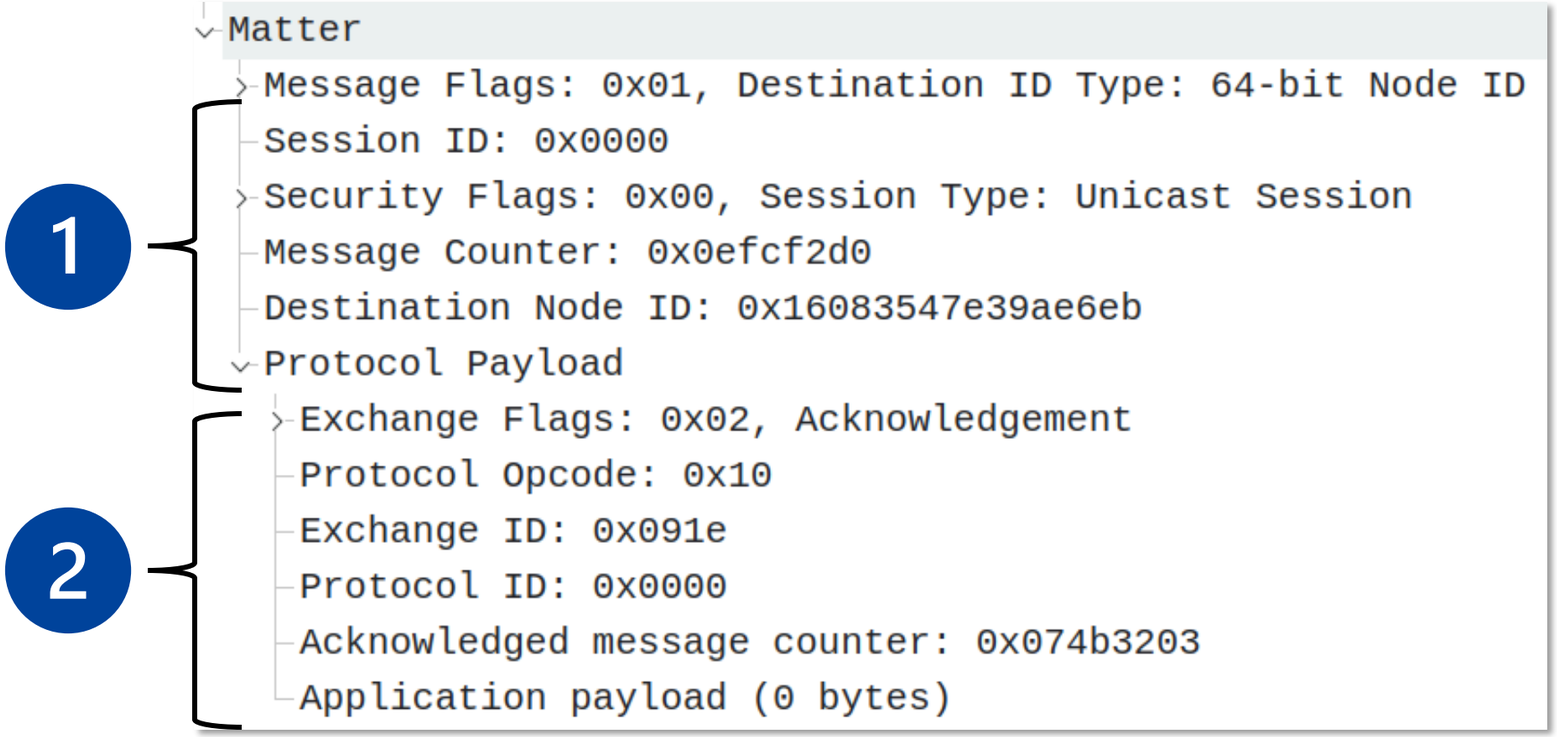
Alternative solution: packet analysis


```
Matter    239 5540 → 5540 Len=177
Matter     88 5540 → 5540 Len=26
Matter    572 5540 → 5540 Len=510
Matter    435 5540 → 5540 Len=373
Matter     88 5540 → 5540 Len=26
Matter     96 5540 → 5540 Len=34
Matter    113 5540 → 5540 Len=51
Matter    113 5540 → 5540 Len=51
Matter    114 5540 → 5540 Len=52
Matter    114 5540 → 5540 Len=52
Matter    113 5540 → 5540 Len=51
Matter    113 5540 → 5540 Len=51
Matter    113 5540 → 5540 Len=51
Matter    113 5540 → 5540 Len=51
```

```
Matter
  Message Flags: 0x01, Destination ID Type: 64-bit Node ID
  Session ID: 0x0000
  Security Flags: 0x00, Session Type: Unicast Session
  Message Counter: 0x0efcf2d0
  Destination Node ID: 0x16083547e39ae6eb
  Protocol Payload
    Exchange Flags: 0x02, Acknowledgement
    Protocol Opcode: 0x10
    Exchange ID: 0x091e
    Protocol ID: 0x0000
    Acknowledged message counter: 0x074b3203
    Application payload (0 bytes)
```

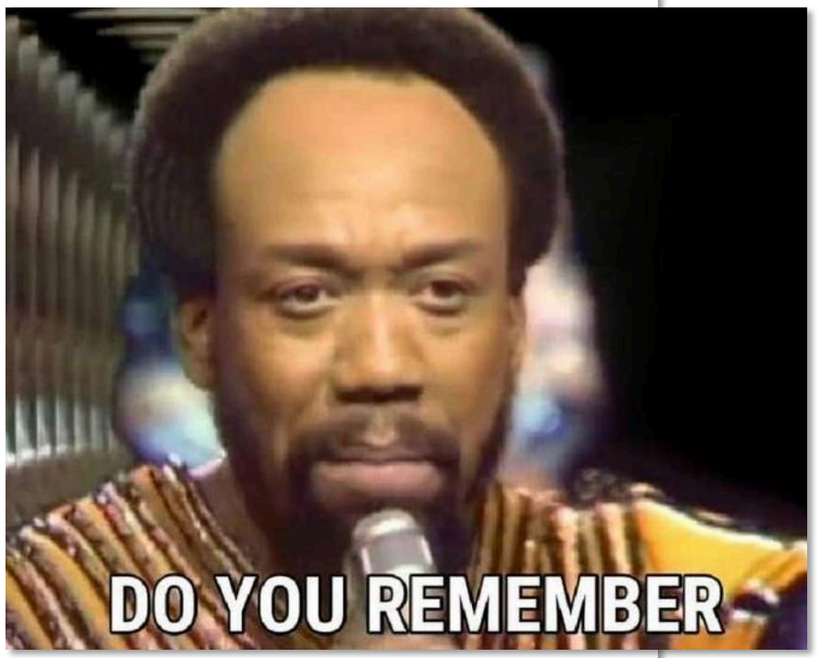
Matter packet fingerprint

1. <matter_msg_len>-<msg_flags>-<security_flags>-<enc_payload_len>;
2. <exch_flags>-<proto_opcode>-<proto_id>-<app_payload_len>;

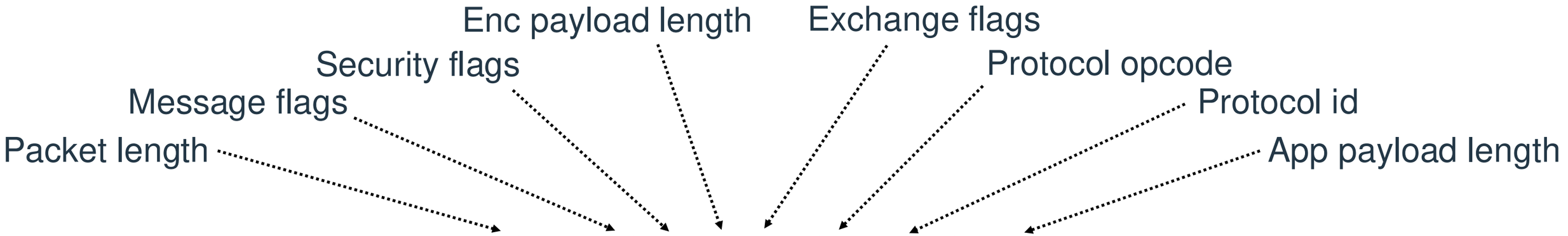


Fingerprint similar to JA3/JA4


```
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 707
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 703
      Version: TLS 1.2 (0x0303)
      Random: 0b279f74d6e4f62f53fde2c93f71837576dce0b4c55f29a1462b7548a9360929
      Session ID Length: 32
      Session ID: c8e57809ac4f3e1e46dc92660653c3d63bd65520ec01350f3db9c39d5aef4b34
      Cipher Suites Length: 32
      Cipher Suites (16 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 598
      Extension: Reserved (GREASE) (len=0)
      Extension: key_share (len=43) x25519
      Extension: ec_point_formats (len=2)
      Extension: application_settings (len=5)
      Extension: renegotiation_info (len=1)
      Extension: signed_certificate_timestamp (len=0)
      Extension: server_name (len=21) name=webmail.umfst.ro
      Extension: extended_master_secret (len=0)
      Extension: psk_key_exchange_modes (len=2)
      Extension: session_ticket (len=208)
      Extension: supported_groups (len=10)
      Extension: compress_certificate (len=3)
      Extension: status_request (len=5)
      Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
      Extension: application_layer_protocol_negotiation (len=14)
      Extension: encrypted_client_hello (len=186)
      Extension: signature_algorithms (len=18)
      Extension: Reserved (GREASE) (len=1)
      [JA4: t13d1516h2_8daaf6152771_02713d6af862]
      [JA4_r: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,cca9_0005,000a,000b,000d,0012,0017,001b,0023,002b,002d,0033,4469,fe0d,ff01_
      [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53,51-11-17513-65281-18-0-23-45-35-10-27-5-43-16-65037-13,29-23-24,0]
      [JA3: 351edb9670cb8a3fd330e2811cb787e4]
```

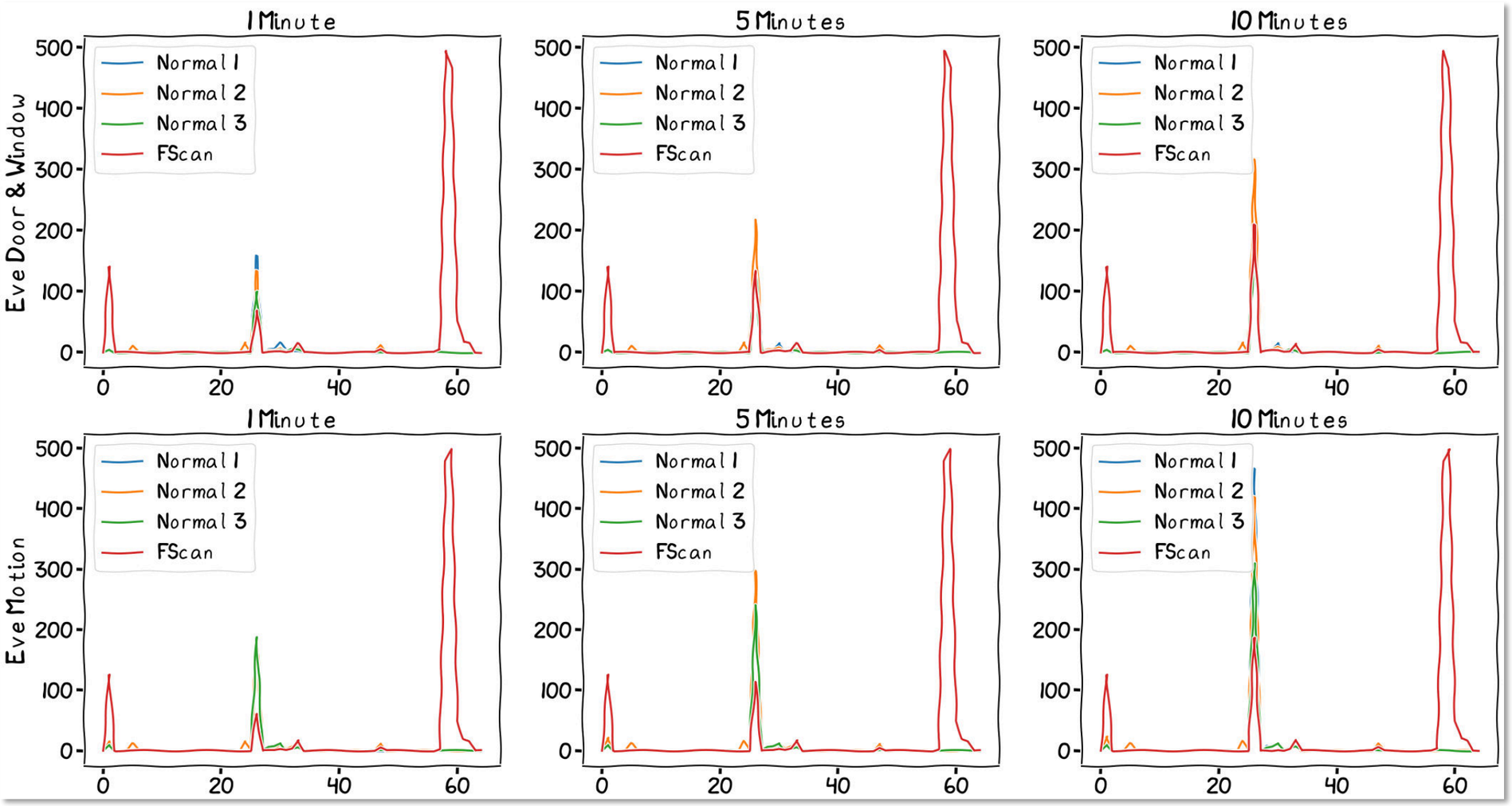


Fingerprint example

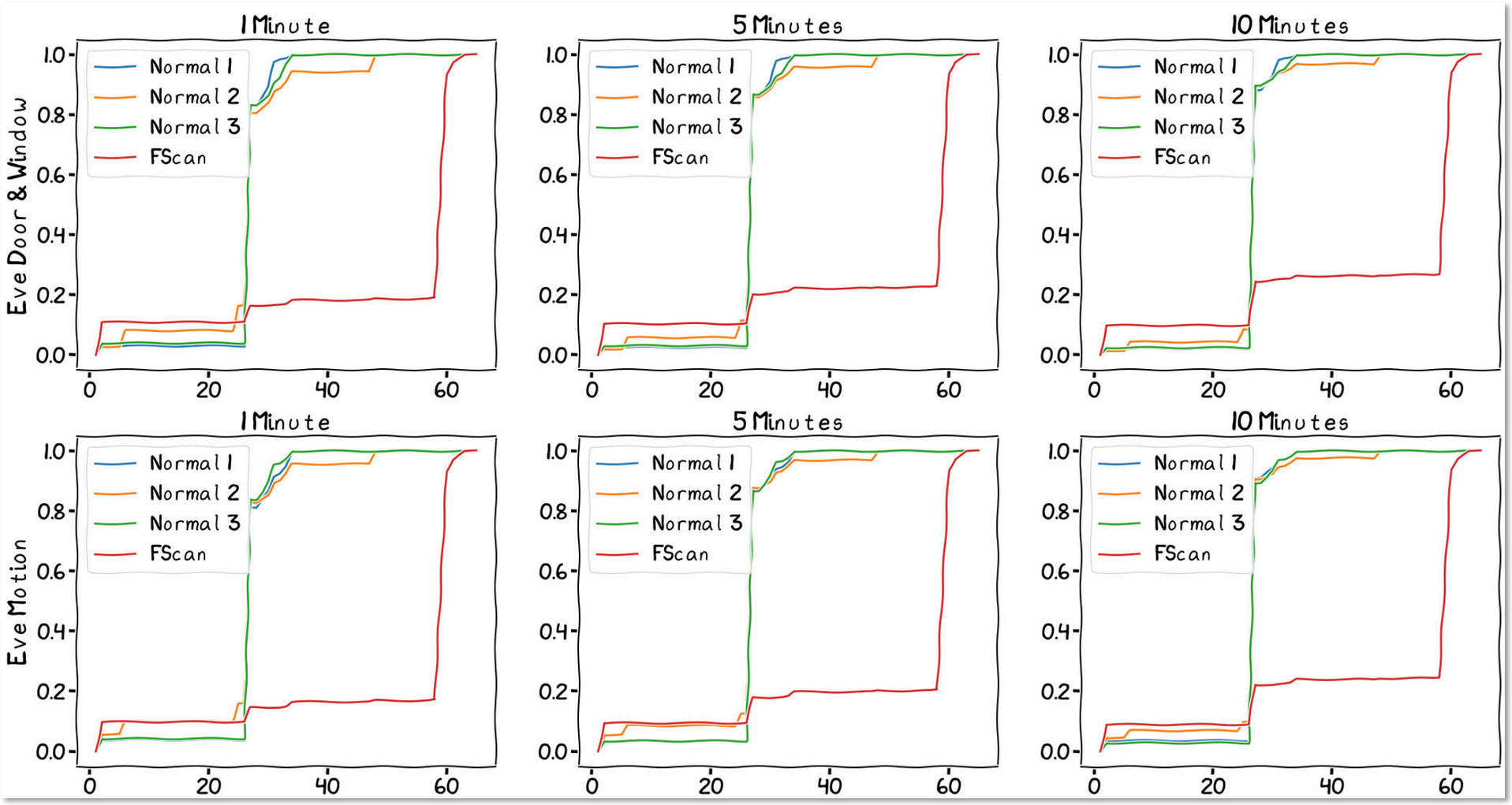



```
Message Flags: 0x04, Has Source ID, Destination ID Type: Not present
Session ID: 0x0000
Security Flags: 0x00, Session Type: Unicast Session
Message Counter: 0x074b3202
Source Node ID: 0x16083547e39ae6eb
Protocol Payload
  Exchange Flags: 0x05, Initiator, Reliability
  Protocol Opcode: 0x30
  Exchange ID: 0x091e
  Protocol ID: 0x0000
  Application payload (155 bytes)
```

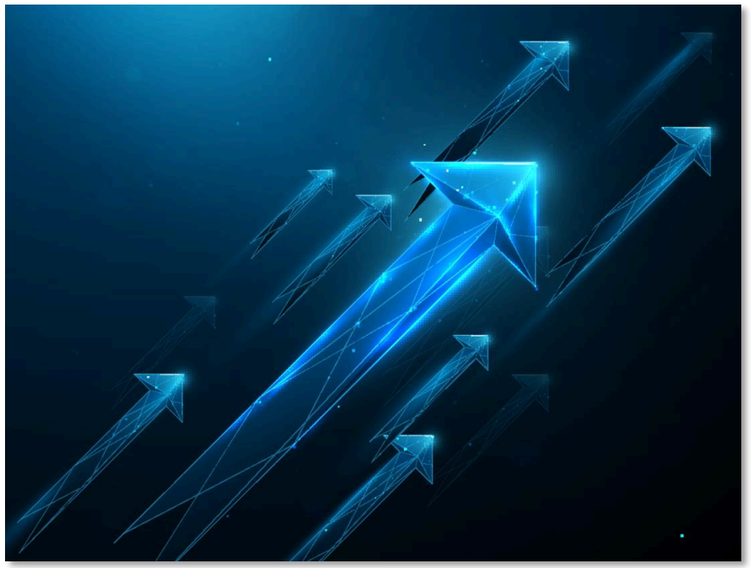
Matter packet analysis



Packet cumulative distributions



Way forward



Call for action

- Call for both **offensive** and **defensive** security research
- Matter is heavily anchored into legislative initiatives worldwide
- Security researchers may shape the evolution of the Matter IoT standard

PRESS RELEASES

The Connectivity Standards Alliance and the Cyber Security Agency of Singapore Sign Mutual Recognition Arrangement on Cybersecurity Labels for Consumer IoT

3/18/2024

Source: https://csa-iot.org/newsroom/the-connectivity-standards-alliance-and-the-cyber-security-agency-of-singapore-sign-mutual-recognition-arrangement-on-cybersecurity-labels-for-consumer-iot/

One cybersecurity mark to rule them all

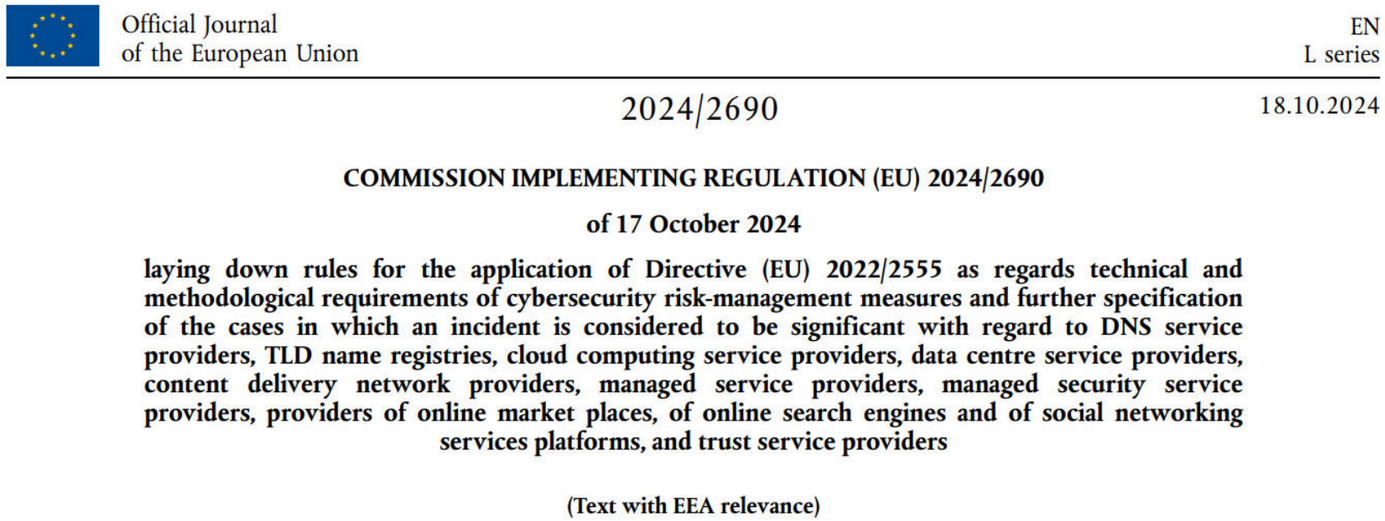
The CSA's announcement on March 18th follows last week's news that the FCC has approved implementing its new cybersecurity labeling program for consumer IoT devices in the US. Both programs are voluntary, and the CSA's label doesn't compete with the US Cyber Trust Mark. Instead, it goes a step further, taking all of the US requirements and adding cybersecurity baselines from similar programs in Singapore and Europe. The end result is a single specification and certification program that can work across multiple countries (see sidebar).

Source: https://www.theverge.com/2024/3/18/24104906/csa-iot-device-security-specification-product-security-verification-mark

Call for action – NIS-2 & CRA

- Network and Information Security (NIS) Directive 2 (NIS 2 (EU) 2022/2555) outlines a wide variety of security requirements
- The EU Cyber Resilience Act (2024) provides harmonized rules for all connected devices

- **Monitoring can help** with early detection of compromise
- Implement mandatory **vulnerability management processes**



Official Journal of the European Union

EN
L series

2024/2690

18.10.2024

COMMISSION IMPLEMENTING REGULATION (EU) 2024/2690

of 17 October 2024

laying down rules for the application of Directive (EU) 2022/2555 as regards technical and methodological requirements of cybersecurity risk-management measures and further specification of the cases in which an incident is considered to be significant with regard to DNS service providers, TLD name registries, cloud computing service providers, data centre service providers, content delivery network providers, managed service providers, managed security service providers, providers of online market places, of online search engines and of social networking services platforms, and trust service providers

(Text with EEA relevance)

REGULATION (EU) 2024/…

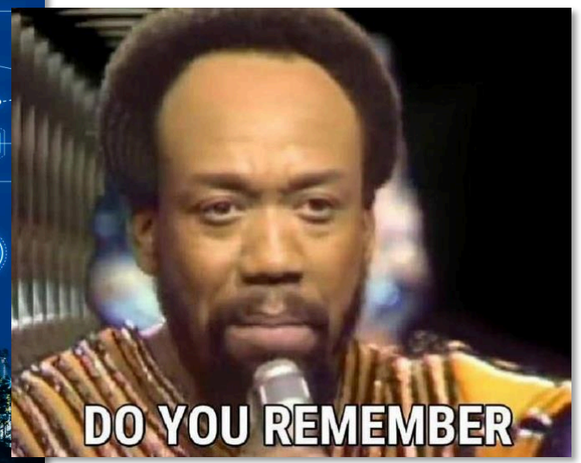
OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 23 October 2024

on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act)

Remember the OPC UA path?

- Similarly to Matter, OPC UA promised **unification** and **robust security features**
- Vulnerabilities still arise due to massive protocol complexities, implementation errors, misconfigurations, evolving cyber threats
- Let's not go down the same path again!



matter

A Broken Chain: Discovering OPC UA Attack Surface and Exploiting the Supply Chain - Black Hat USA 2021

Resting on Feet of Clay: Securely Bootstrapping OPC UA Deployments - Black Hat Europe 2021

Exploiting OPC-UA in Every Possible Way: Practical Attacks Against Modern OPC-UA Architectures - Black Hat USA 2023

Lessons learned

- The **description of security protocols** must be improved to facilitate analysis
- **Offensive security** investigations are needed (e.g., hackathons, bounty-hunting) to ensure a robust, bullet-proof standard
- **(encrypted) Matter traffic** must be **monitored** in order to detect (new) attacks

Black Hat Sound Bytes

1. **Matter is a nuclear blast protocol** from ALL perspectives!

2. After almost 30 years from Gavin Lowe's pioneering work on security protocol analysis*, **packet replay attacks are still the #1 hit**!

3. There is still time! The standard is being shaped, **be the one that improves it (all stakeholders)** by challenging the protocol's security!

* Lowe, G. (1996). Breaking and fixing the Needham-Schroeder Public-Key Protocol using FDR. In: TACAS, LNCS, vol 1055. Springer, Berlin, Heidelberg 1996. https://link.springer.com/chapter/10.1007/3-540-61042-1_43

black hat EUROPE 2024

Thank you!

Béla Genge

bgenge@bitdefender.com

Bitdefender