AURACAST™

Breaking Broadcast LE Audio Before It Hits the Shelves

38th Chaos Communication Congress

Frieder Steinmetz & Dennis Heinze

ERNW
providing security.

1. What is Bluetooth Auracast?

What is Bluetooth Auracast?

AURACAST™

A BLUETOOTH® TECHNOLOGY

- Bluetooth Low Energy audio broadcast
- Initially designed as hearing aid feature
- Introduced in 2019 with Bluetooth Core Specification 5.2
- Actual implementations start arriving now

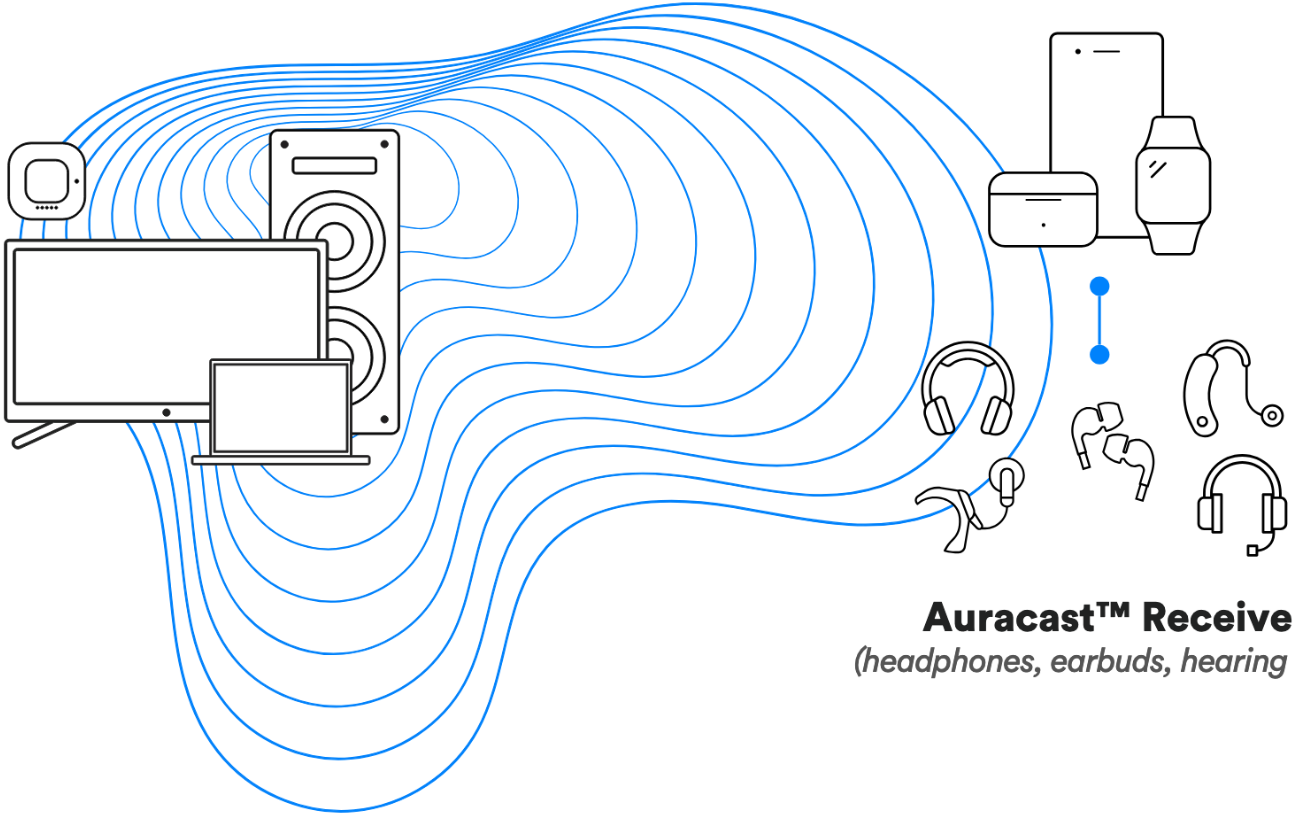
- Potential applications:
  - Replacement for hearing aid coils
  - Audio sharing
  - TVs, sports bars
  - Airports & Train stations
  - Sendezentrum @ 38c3

Auracast™ Transmitter

(television, laptop, PA system...)

Auracast™ Assistant

(smartphone, smartwatch, hearing aid remote...)



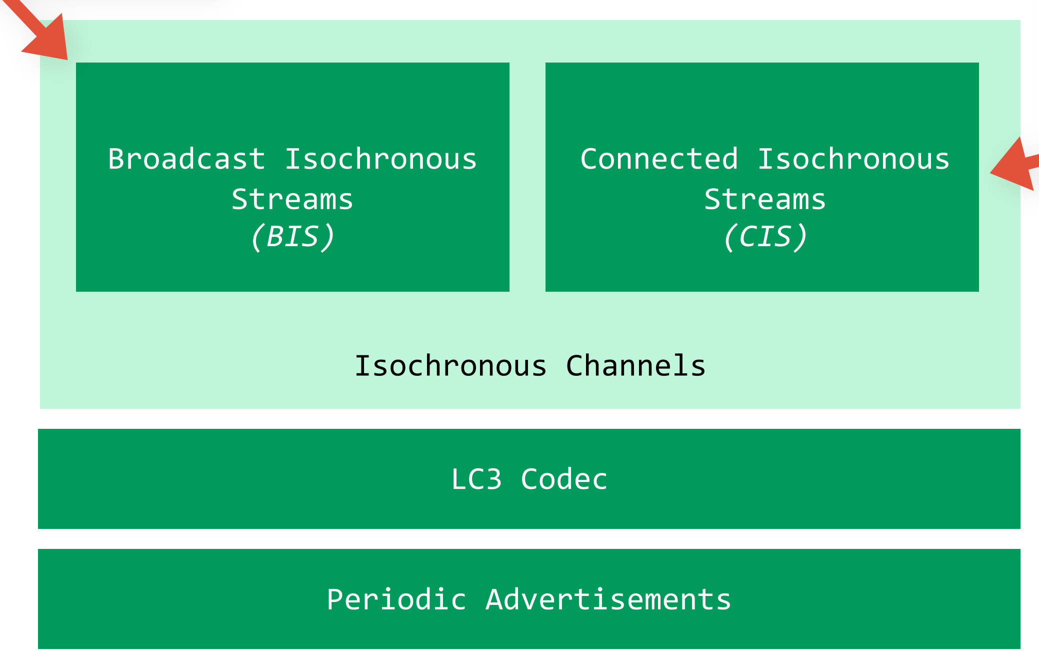
Auracast™ Receiver

(headphones, earbuds, hearing aid...)

https://www.bluetooth.com/wp-content/uploads/2024/05/2403_How_To_Auracast_Transmitter.pdf

This is broadcast audio and is, for us, the relevant feature that enables Auracast.

at is Bluetooth Auracast?



Broadcast Isochronous Streams *(BIS)*

Connected Isochronous Streams *(CIS)*

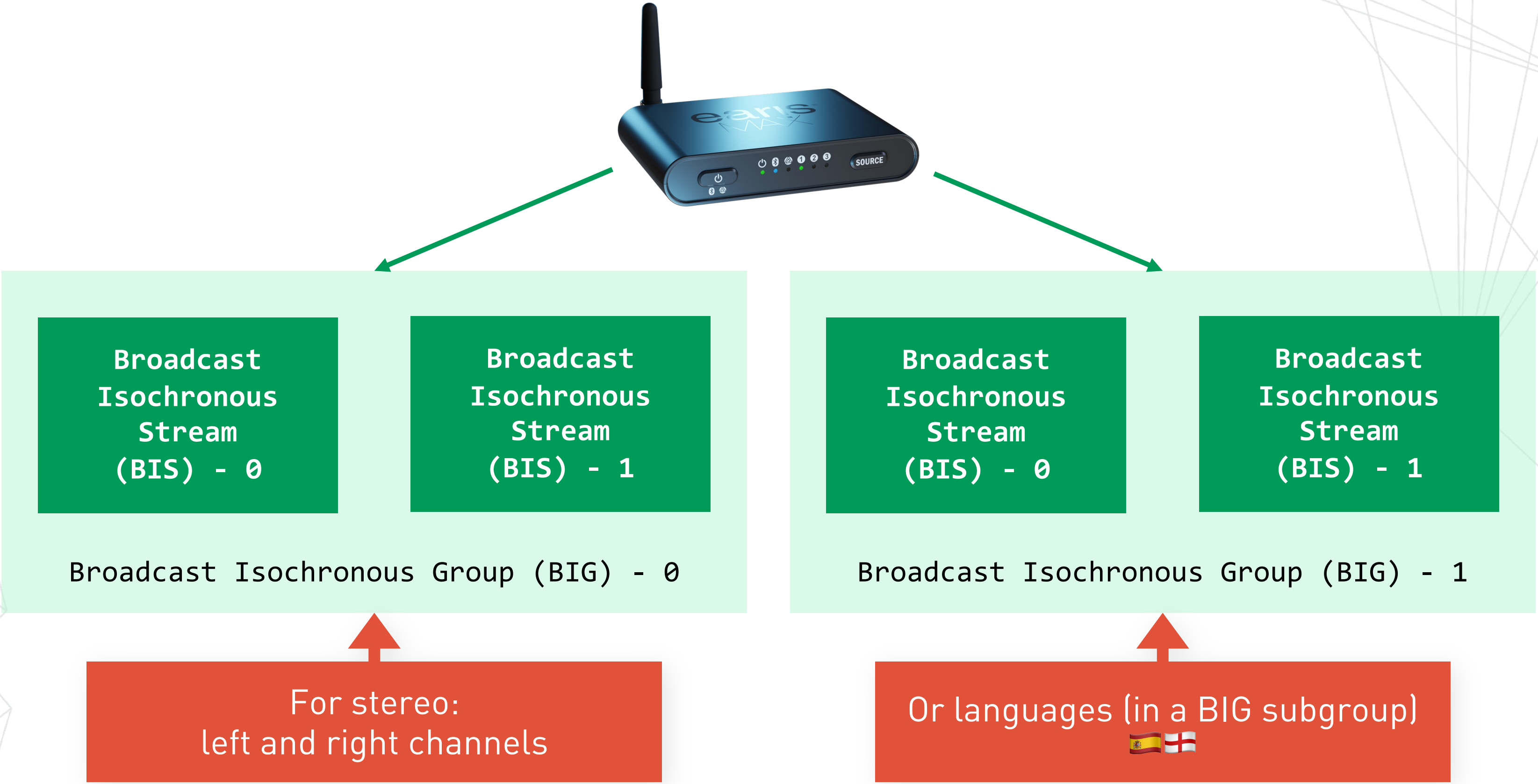
Isochronous Channels

LC3 Codec

Periodic Advertisements

This is LE Audio, which might replace "Classic Bluetooth" audio.

Auracast Transmitters



Broadcast Isochronous Stream (BIS) - 0

Broadcast Isochronous Stream (BIS) - 1

Broadcast Isochronous Group (BIG) - 0

Broadcast Isochronous Stream (BIS) - 0

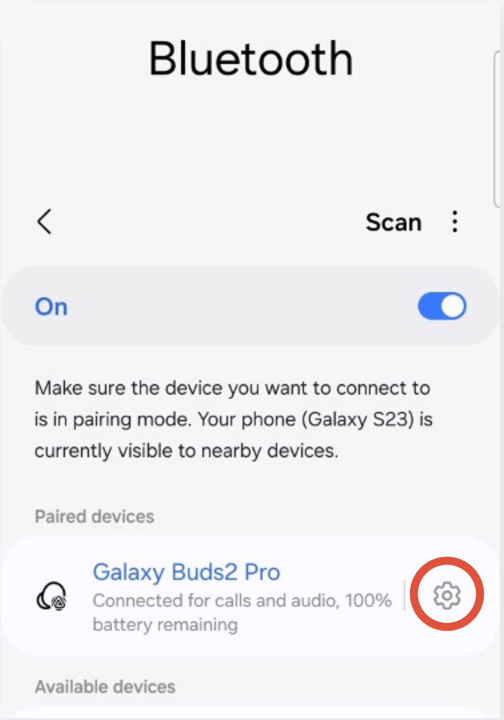
Broadcast Isochronous Stream (BIS) - 1

Broadcast Isochronous Group (BIG) - 1

For stereo:
left and right channels

Or languages (in a BIG subgroup)
🇪🇸🏴󠁧󠁢󠁥󠁮󠁧󠁿

Auracast on Samsung Galaxy



Bluetooth

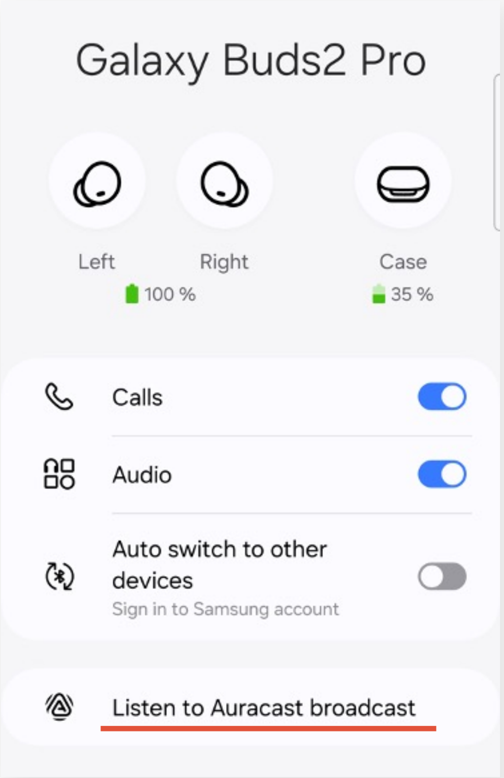
Make sure the device you want to connect to is in pairing mode. Your phone (Galaxy S23) is currently visible to nearby devices.

Paired devices

Galaxy Buds2 Pro
Connected for calls and audio, 100% battery remaining

Available devices

Auracast on Samsung Galaxy



Galaxy Buds2 Pro

Left
100 %

Right

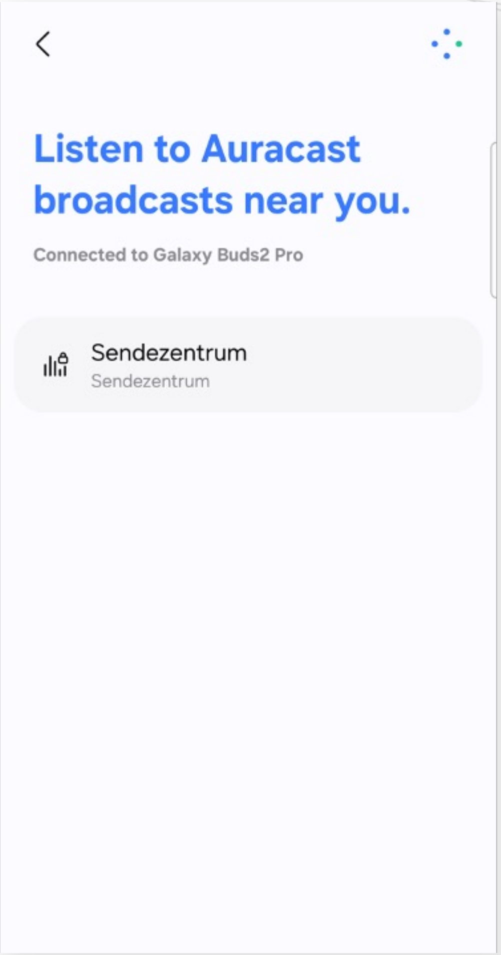
Case
35 %

Calls ●

Audio ●

Auto switch to other devices ○
Sign in to Samsung account

Listen to Auracast broadcast

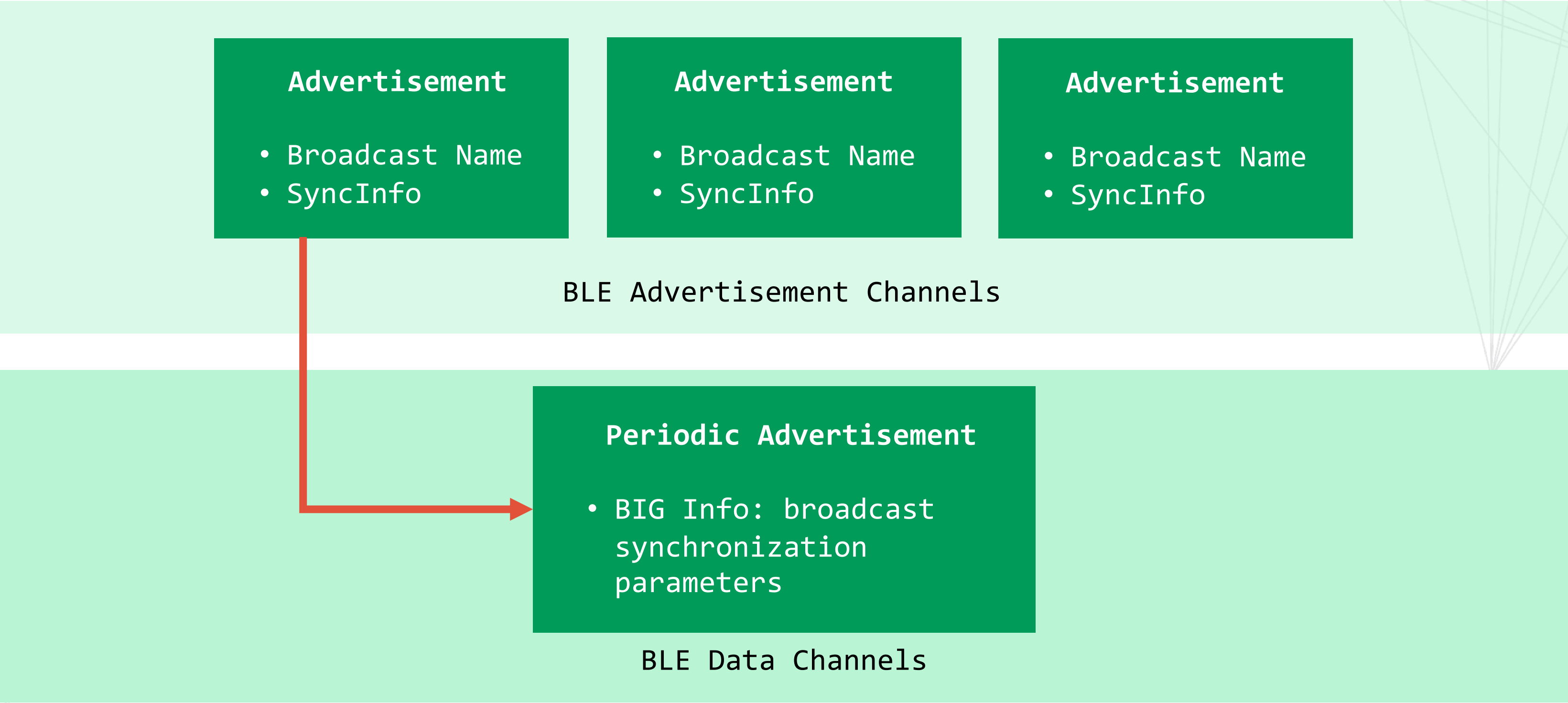


Listen to Auracast broadcasts near you.

Connected to Galaxy Buds2 Pro

Sendezentrum
Sendezentrum

BIS Synchronization



Advertisement

- Broadcast Name
- SyncInfo

Advertisement

- Broadcast Name
- SyncInfo

Advertisement

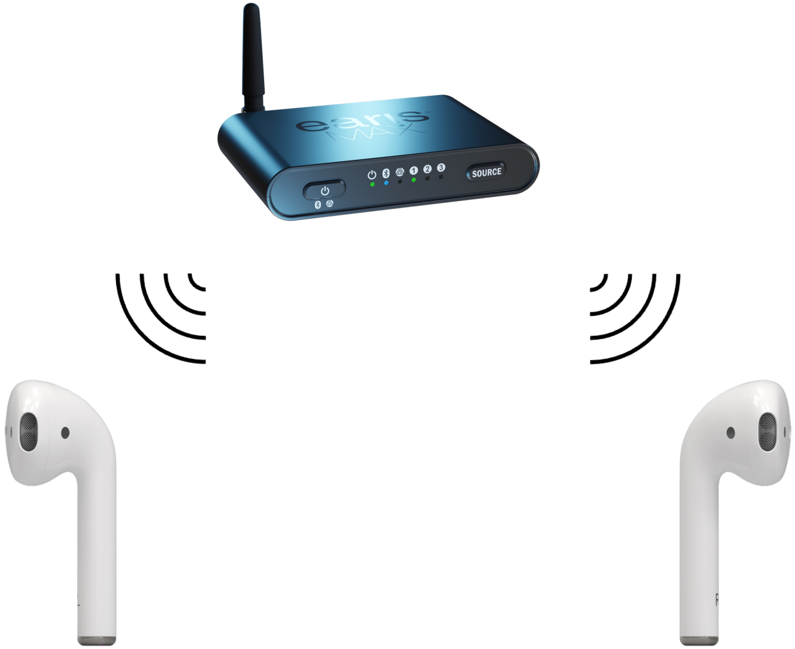
- Broadcast Name
- SyncInfo

BLE Advertisement Channels

Periodic Advertisement

- BIG Info: broadcast synchronization parameters

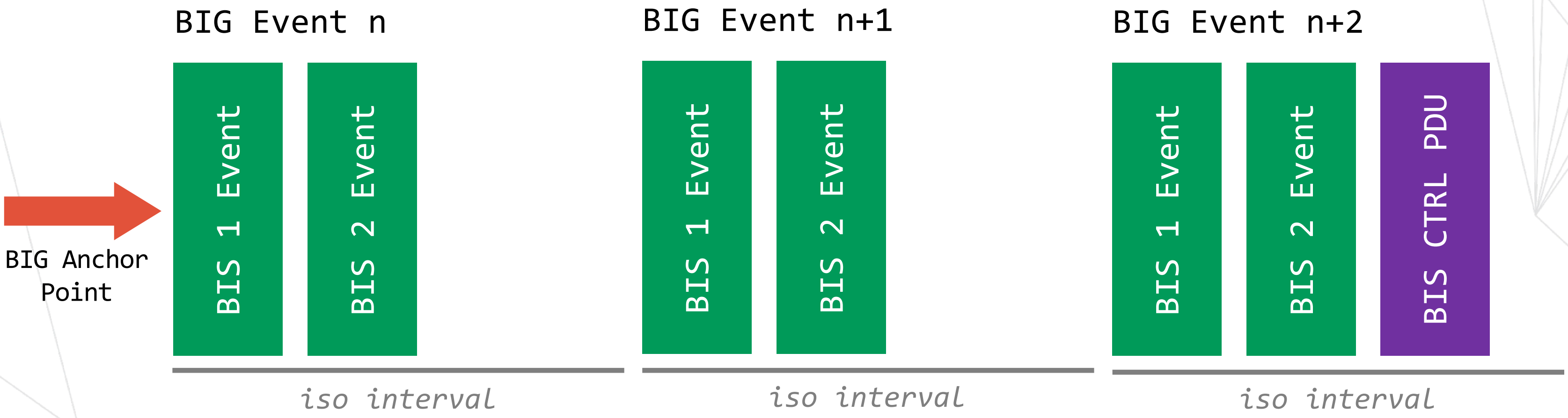
BLE Data Channels



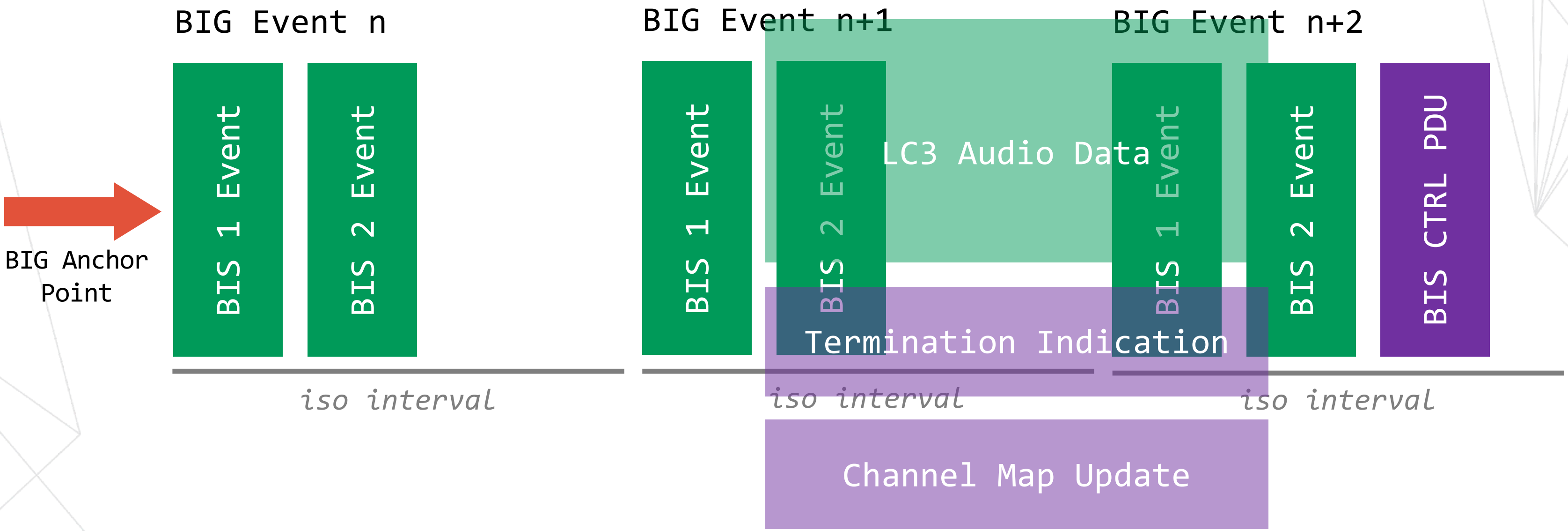
Left
(BIS 0)

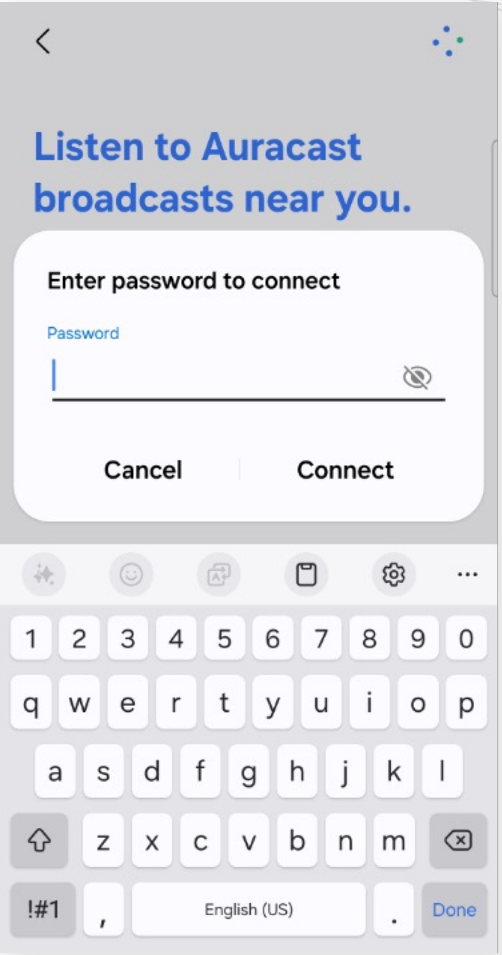
Right
(BIS 1)

Broadcast Isochronous Streams



Broadcast Isochronous Streams





Listen to Auracast broadcasts near you.

Enter password to connect

Password

2. Auracast Security

Security Concerns

○ **Privacy**
Broadcasts might still supposed to be private, only available to a select group of recipients.

○ **Authenticity**
Broadcast content should not be spoofable.

○ **Availability**
Broadcasts should be resilient to accidental and deliberate interference.

What security features does it have?

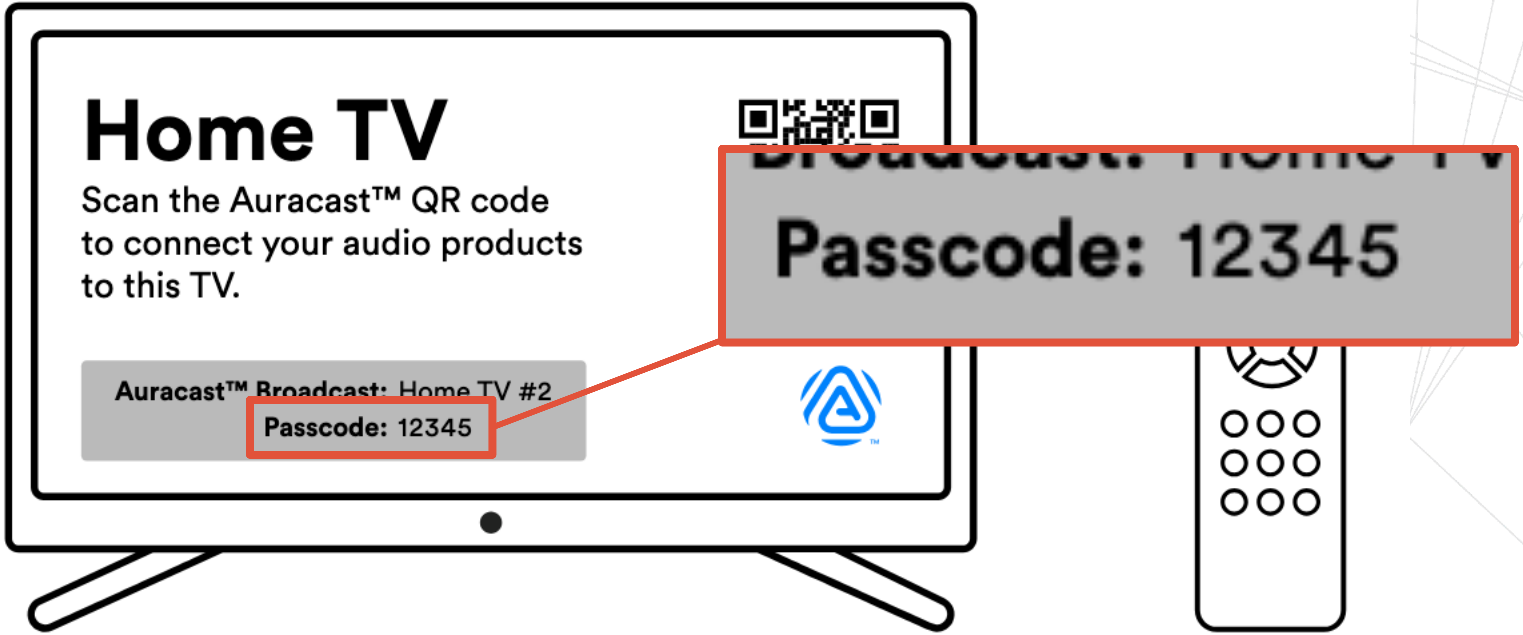
1 🔒 Encrypted Broadcasts

That’s it...

Encrypted Broadcasts

"Auracast™ broadcast audio is capable of using ***broadcast codes*** *for* ***secure conversations****. Managing output power and spillover alone does not guarantee privacy. The use of broadcast codes is recommended for* ***private access****."*

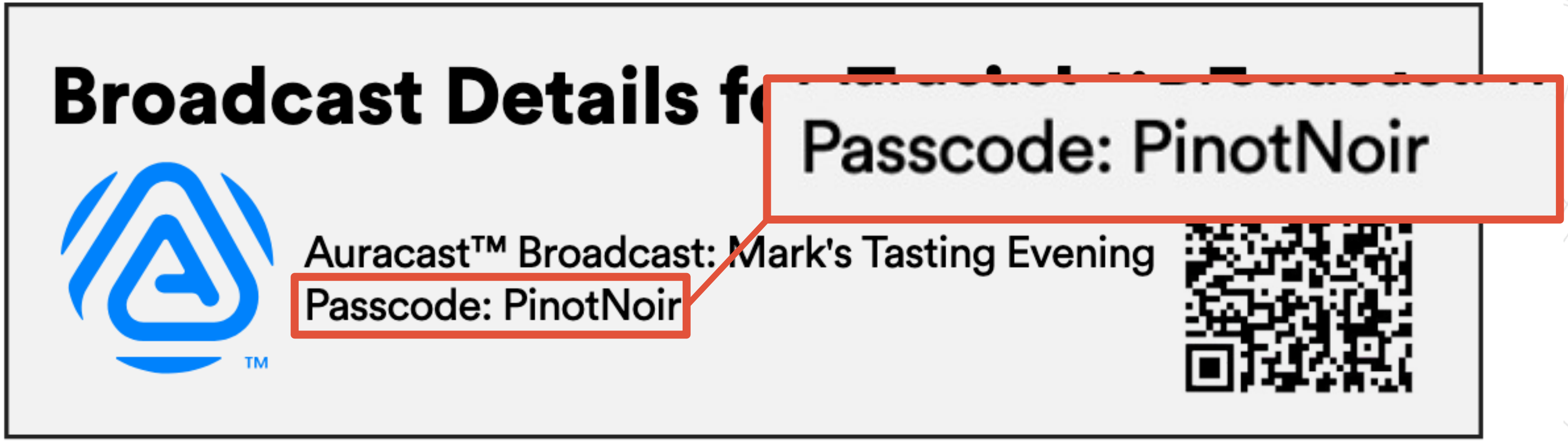
https://www.bluetooth.com/blog/answers-to-commonly-asked-questions-about-auracast-broadcast-audio/



Home TV
Scan the Auracast™ QR code to connect your audio products to this TV.
Auracast™ Broadcast: Home TV #2
Passcode: 12345
Passcode: 12345

Figure 3.8: Tiered access for home TV usage

https://www.bluetooth.com/wp-content/uploads/2024/05/2403_Auracast_Overview.pdf



Broadcast Details fo

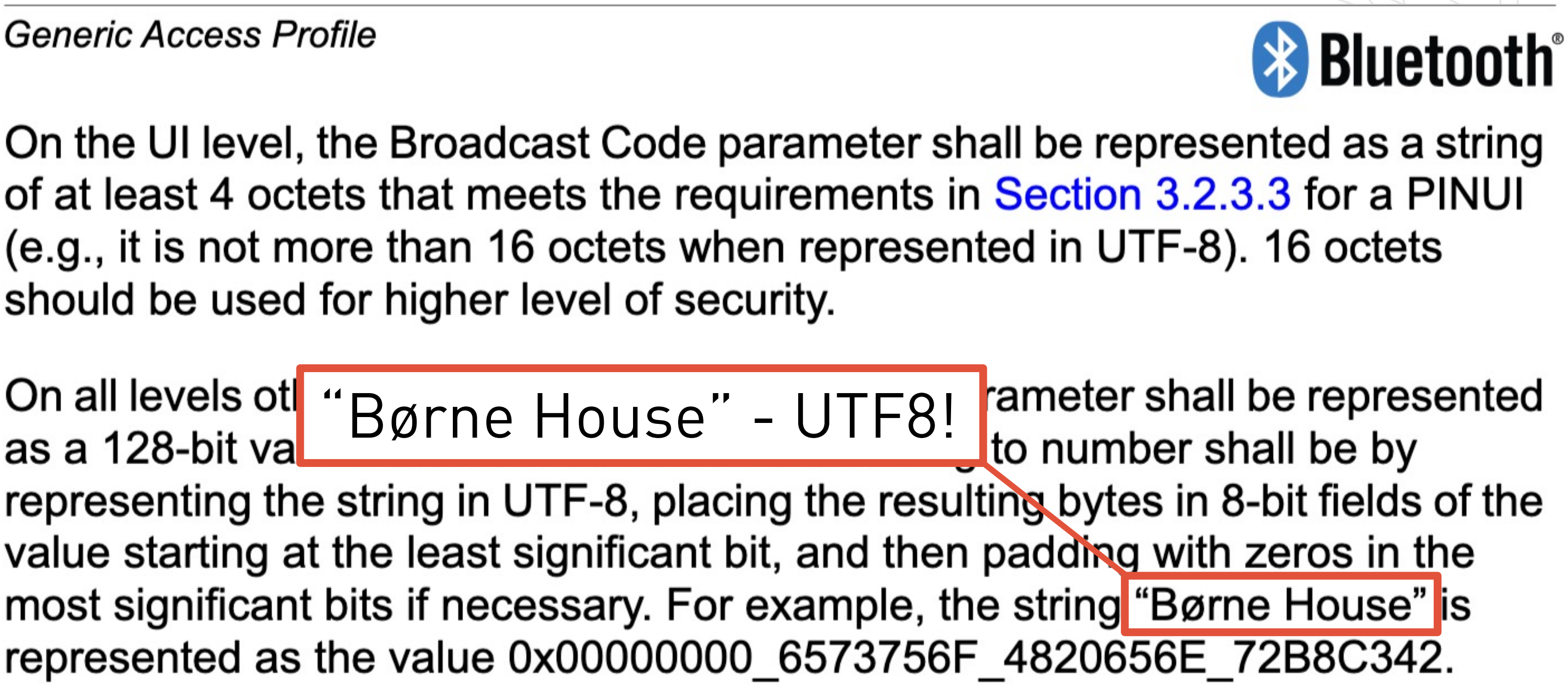
Passcode: PinotNoir

Auracast™ Broadcast: Mark's Tasting Evening

Passcode: PinotNoir

Figure 2.3 Example of a public Auracast™ broadcast audio notice with a Broadcast_Code

Auracast™ Simple Transmitter Best Practices Guide – Page 19
https://www.bluetooth.com/wp-content/uploads/2022/10/Auracast-Transmitter_Recommendations.pdf



Generic Access Profile

Bluetooth®

On the UI level, the Broadcast Code parameter shall be represented as a string of at least 4 octets that meets the requirements in Section 3.2.3.3 for a PINUI (e.g., it is not more than 16 octets when represented in UTF-8). 16 octets should be used for higher level of security.

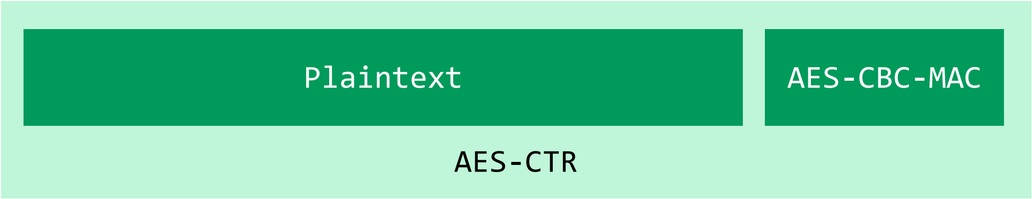
On all levels otl rameter shall be represented as a 128-bit va to number shall be by representing the string in UTF-8, placing the resulting bytes in 8-bit fields of the value starting at the least significant bit, and then padding with zeros in the most significant bits if necessary. For example, the string “Børne House” is represented as the value 0x00000000_6573756F_4820656E_72B8C342.

“Børne House” - UTF8!

BLUETOOTH CORE SPECIFICATION Version 5.4 | Vol 3, Part C page 1253

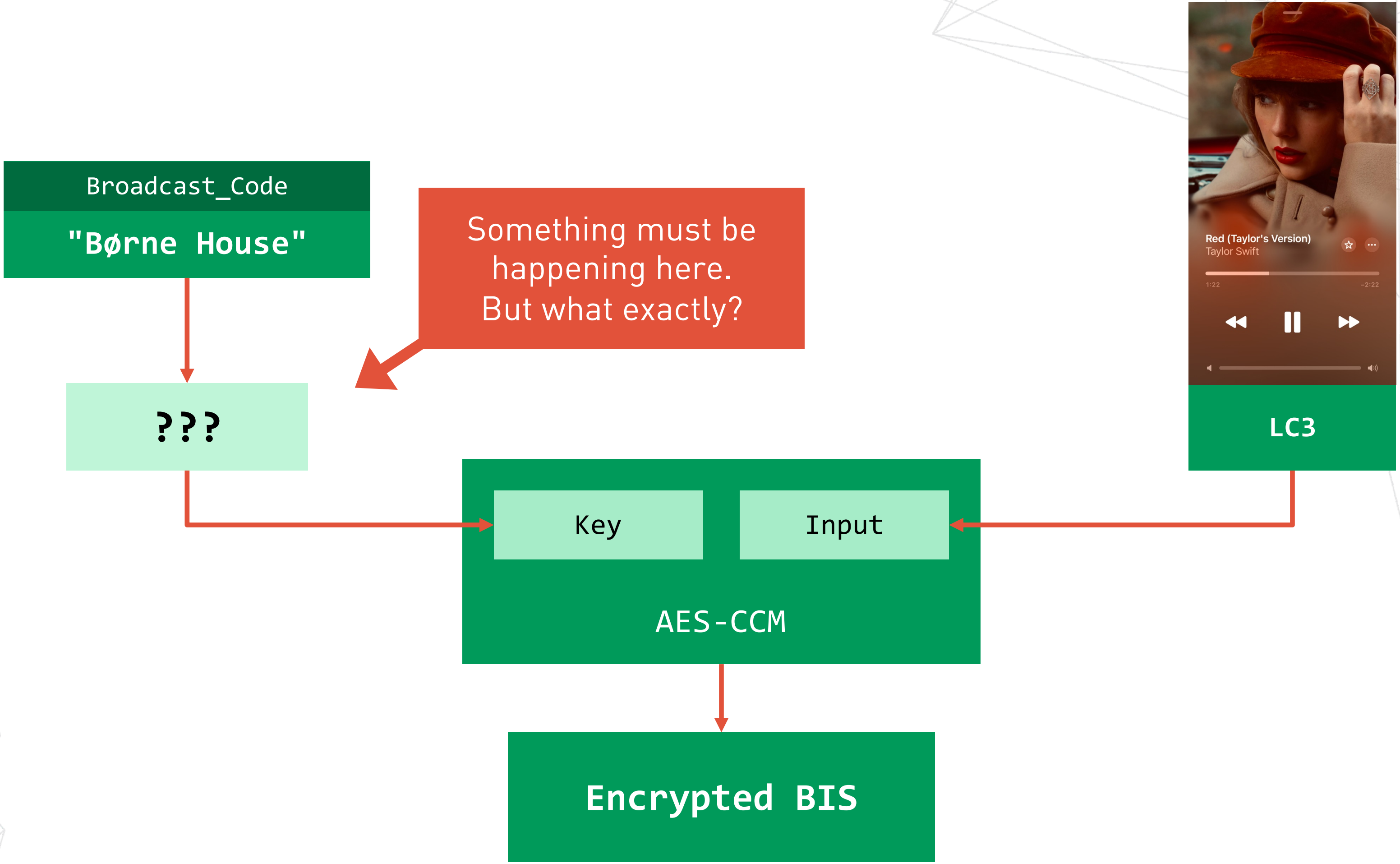
Encryption

AES-CCM:



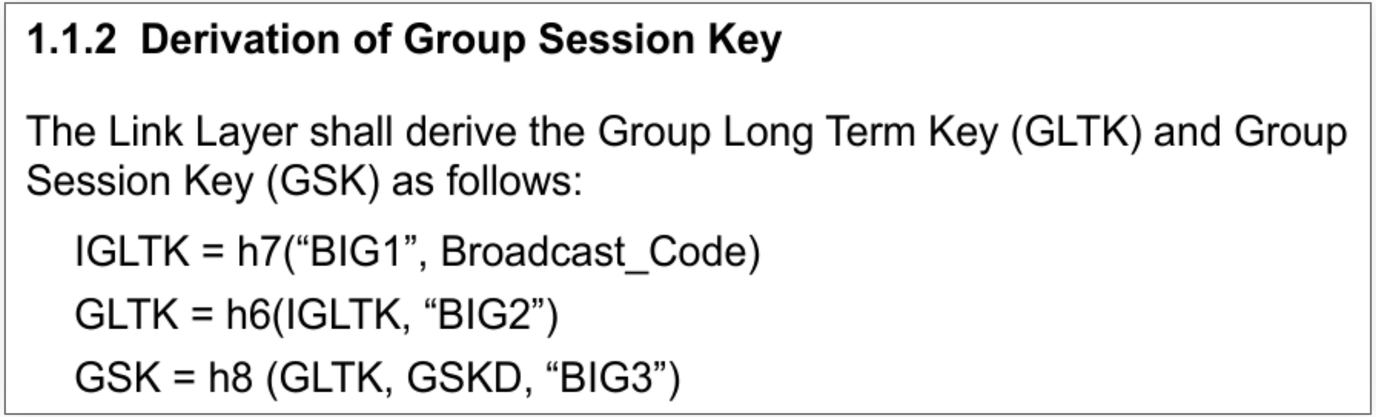
Authenticated Encryption with Associated Data (AEAD)

- **AES CTR** mode for confidentiality
- **CBC-MAC** for authentication



Something must be happening here. But what exactly?

Key Derivation



1.1.2 Derivation of Group Session Key

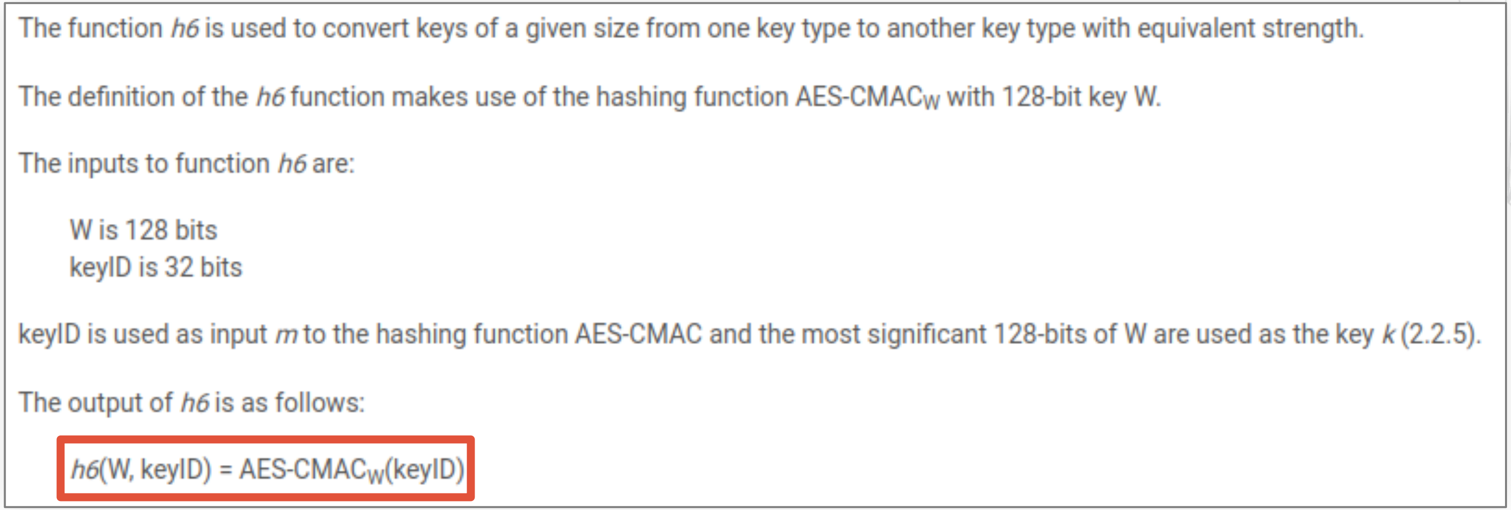
The Link Layer shall derive the Group Long Term Key (GLTK) and Group Session Key (GSK) as follows:

IGLTK = h7(“BIG1”, Broadcast_Code)

GLTK = h6(IGLTK, “BIG2”)

GSK = h8 (GLTK, GSKD, “BIG3”)

h6 – Link Key Conversion Function



The function *h6* is used to convert keys of a given size from one key type to another key type with equivalent strength.

The definition of the *h6* function makes use of the hashing function AES-CMAC$_W$ with 128-bit key W.

The inputs to function *h6* are:

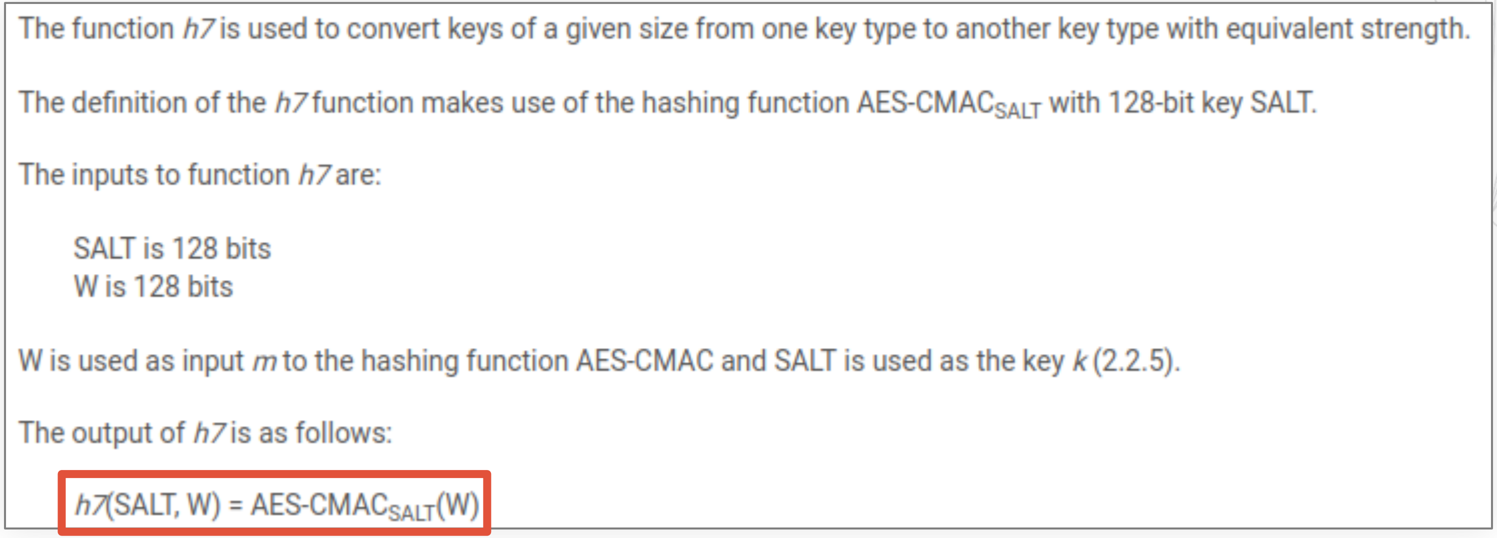
W is 128 bits
keyID is 32 bits

keyID is used as input *m* to the hashing function AES-CMAC and the most significant 128-bits of W are used as the key *k* (2.2.5).

The output of *h6* is as follows:

$h6(W, keyID) = AES\text{-}CMAC_W(keyID)$

h7 – Link Key Conversion Function



The function *h7* is used to convert keys of a given size from one key type to another key type with equivalent strength.

The definition of the *h7* function makes use of the hashing function $\text{AES-CMAC}_{\text{SALT}}$ with 128-bit key SALT.

The inputs to function *h7* are:

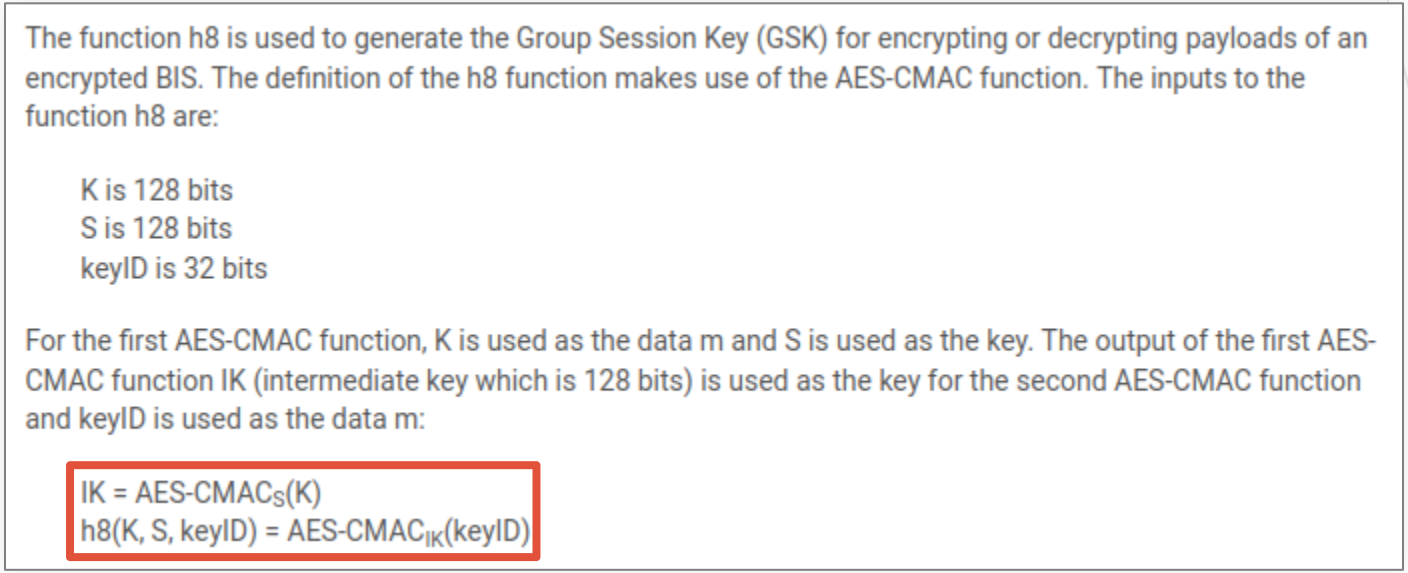
SALT is 128 bits
W is 128 bits

W is used as input *m* to the hashing function AES-CMAC and SALT is used as the key *k* (2.2.5).

The output of *h7* is as follows:

$$h7(\text{SALT}, W) = \text{AES-CMAC}_{\text{SALT}}(W)$$

h8 – Group Session Key Derivation Function

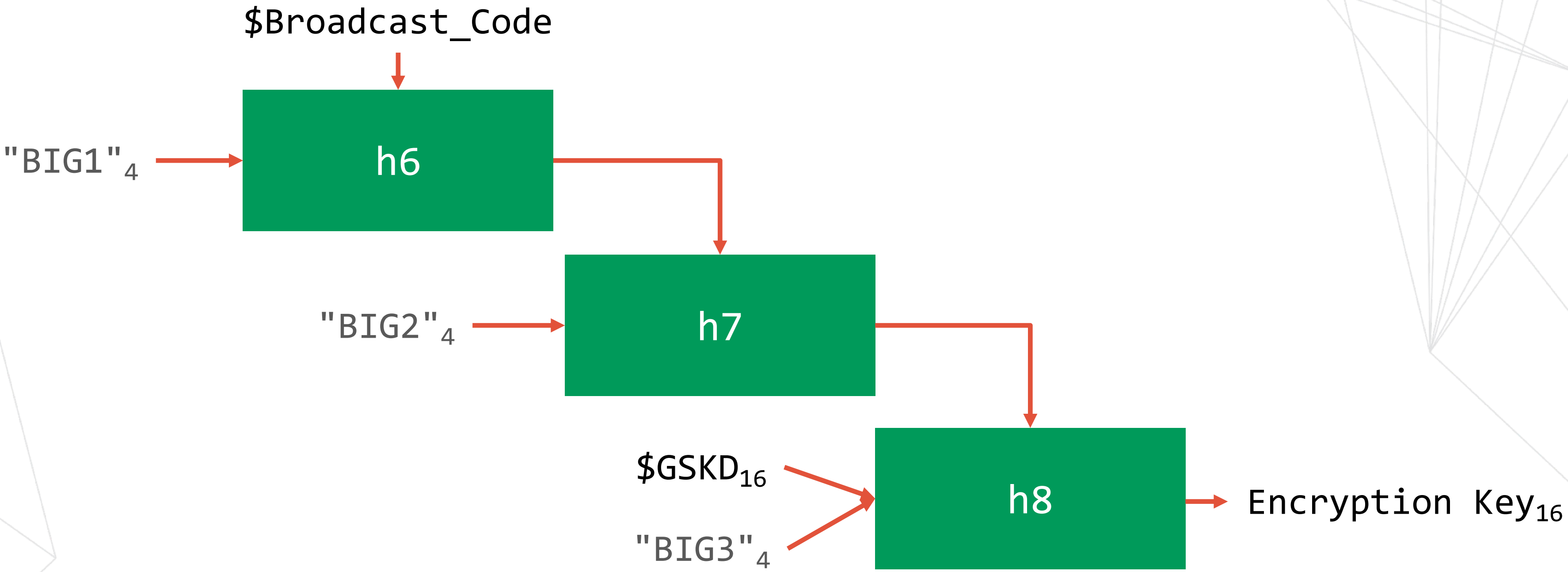


The function h8 is used to generate the Group Session Key (GSK) for encrypting or decrypting payloads of an encrypted BIS. The definition of the h8 function makes use of the AES-CMAC function. The inputs to the function h8 are:

K is 128 bits
S is 128 bits
keyID is 32 bits

For the first AES-CMAC function, K is used as the data m and S is used as the key. The output of the first AES-CMAC function IK (intermediate key which is 128 bits) is used as the key for the second AES-CMAC function and keyID is used as the data m:

$IK = \text{AES-CMAC}_{S}(K)$
$h8(K, S, keyID) = \text{AES-CMAC}_{IK}(keyID)$



$Broadcast_Code

"BIG1"$_4$

h6

"BIG2"$_4$

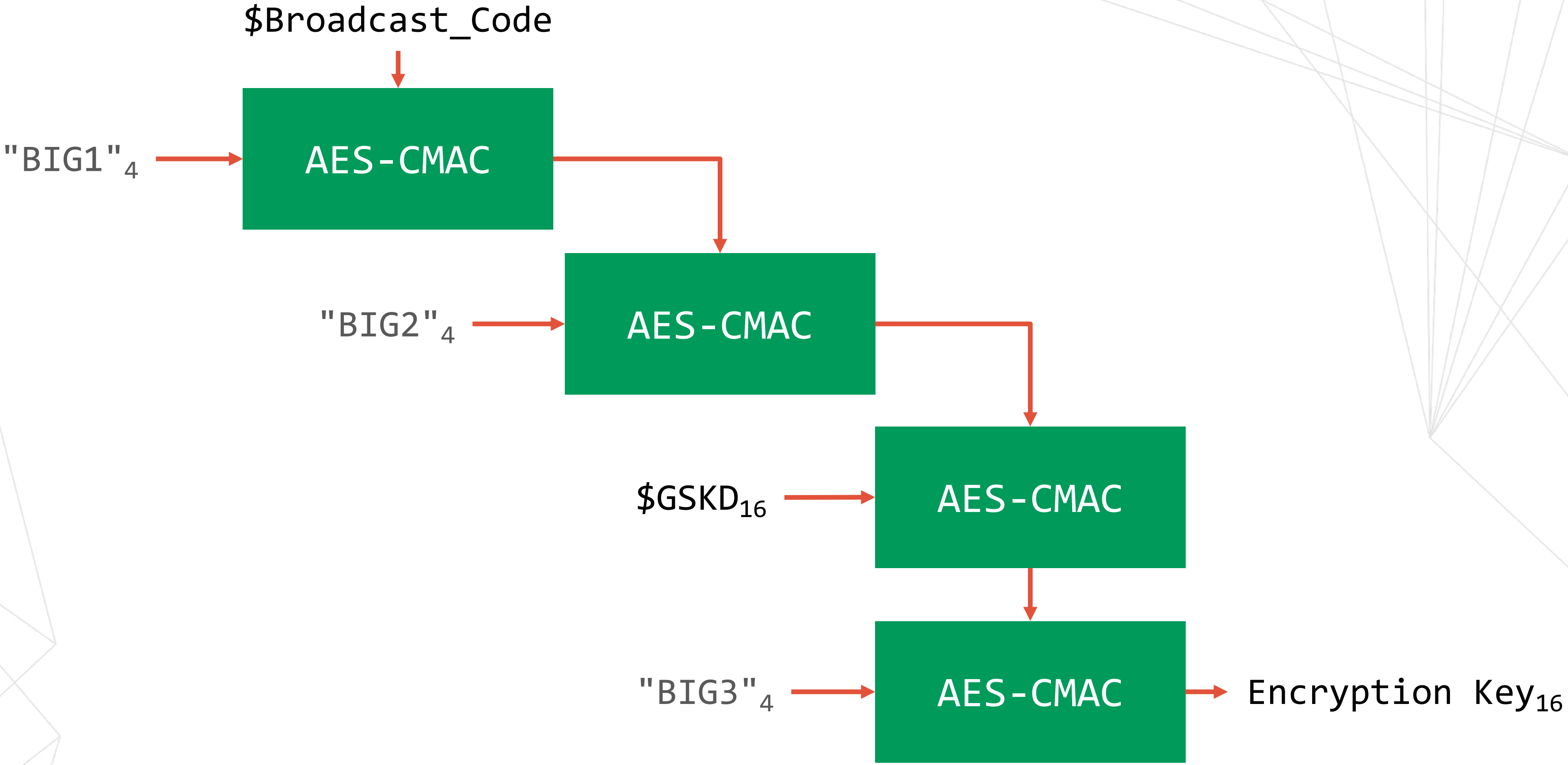
h7

$GSKD$_{16}$

"BIG3"$_4$

h8

Encryption Key$_{16}$


$Broadcast_Code
"BIG1"4
AES-CMAC
"BIG2"4
AES-CMAC
$GSKD16
AES-CMAC
"BIG3"4
AES-CMAC
Encryption Key16

What security properties does the Specification guarantee?

None

None*

*This is a bit harsh, but not entirely wrong.

What are the security properties of encrypted broadcasts?

Security Concerns

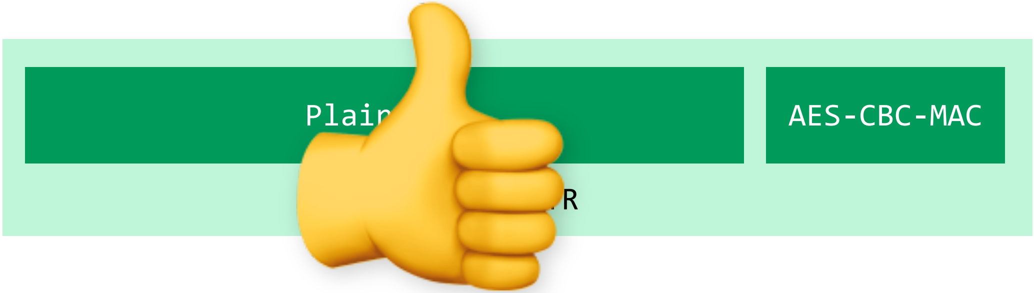
○ **Privacy**
Broadcasts might still supposed to be private, only available to a select group of recipients.

○ **Authenticity**
Broadcast content should not be spoofable.

○ **Availability**
Broadcasts should be resilient to accidental and deliberate interference.

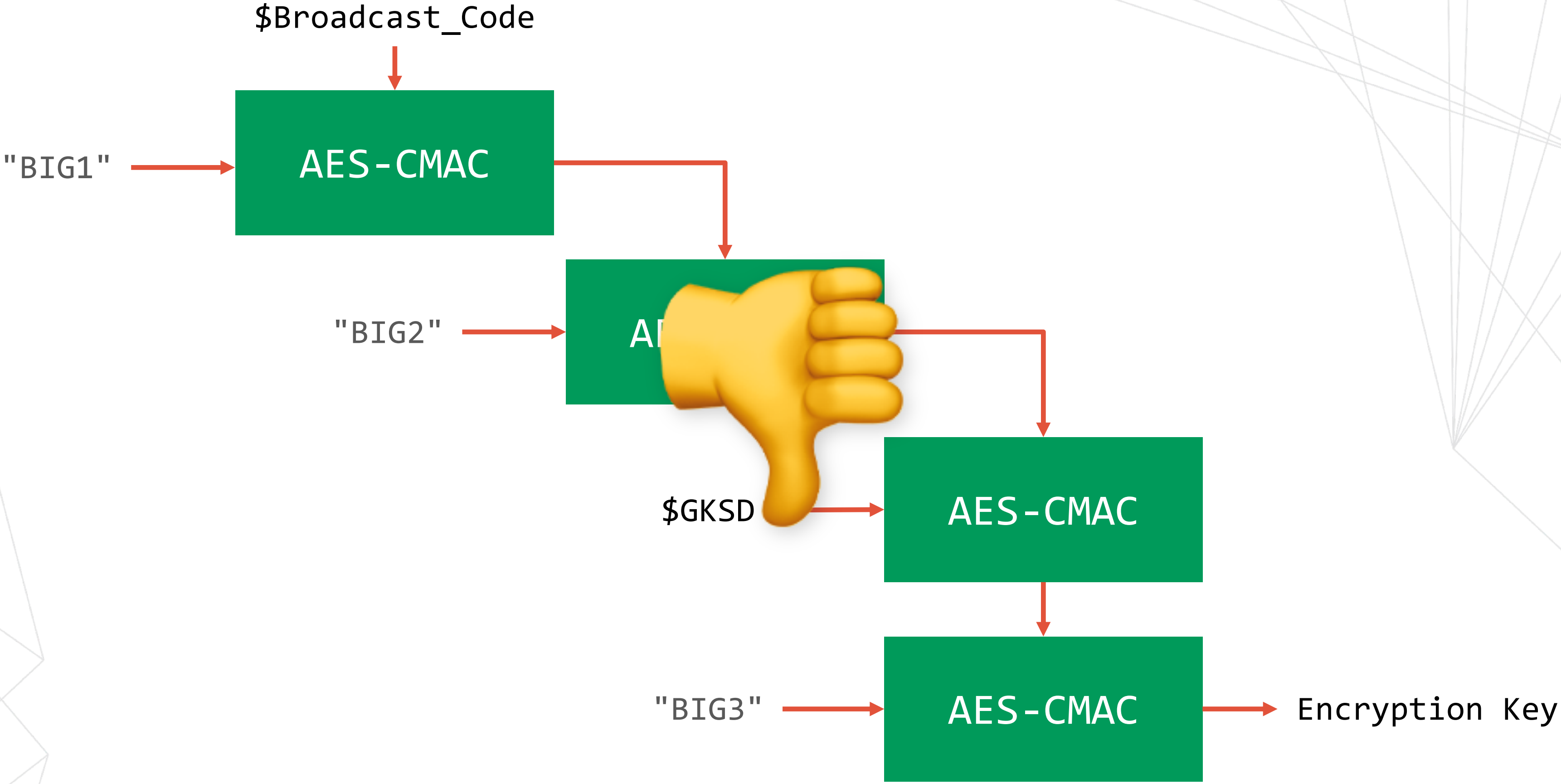
Encryption

AES-CCM:



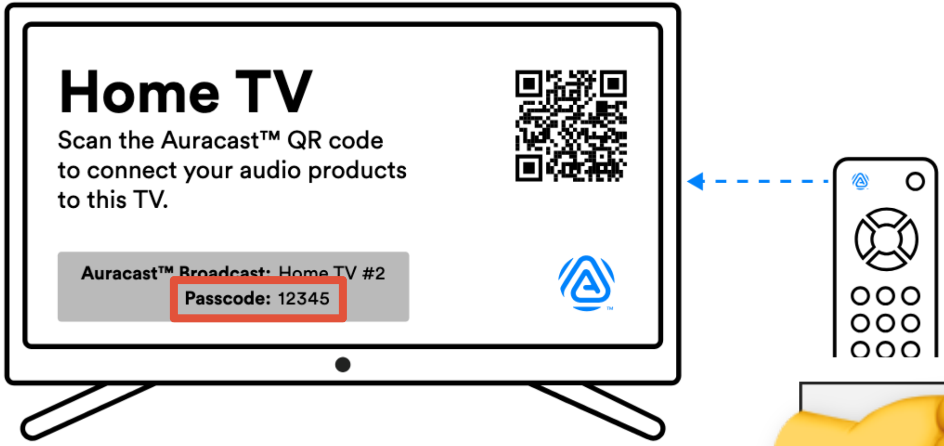
Authenticated Encryption with Associated Data (AEAD)

- **AES CTR** mode for confidentiality
- **CBC-MAC** for authentication


$Broadcast_Code
"BIG1"
AES-CMAC
"BIG2"
AE
$GKSD
AES-CMAC
"BIG3"
AES-CMAC
Encryption Key

Key Derivation

- AES-CMAC is a keyed hash function
  - Suitable wherever key derivation with a PRF is suitable
  - Deriving a key from a DH shared secret would be an application
- It is **not suitable** for "*key stretching*" which adds a known number of bits to the expected difficulty of an exhaustive search attack.



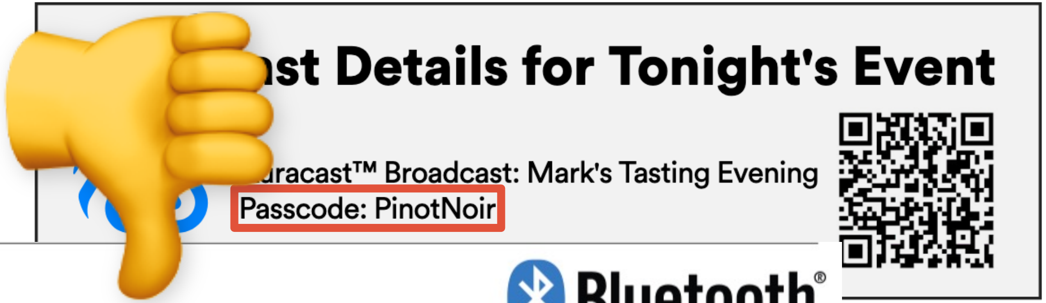
Home TV

Scan the Auracast™ QR code to connect your audio products to this TV.

Auracast™ Broadcast: Home TV #2

Passcode: 12345

Figure 3.8: Tiered access for home TV usage

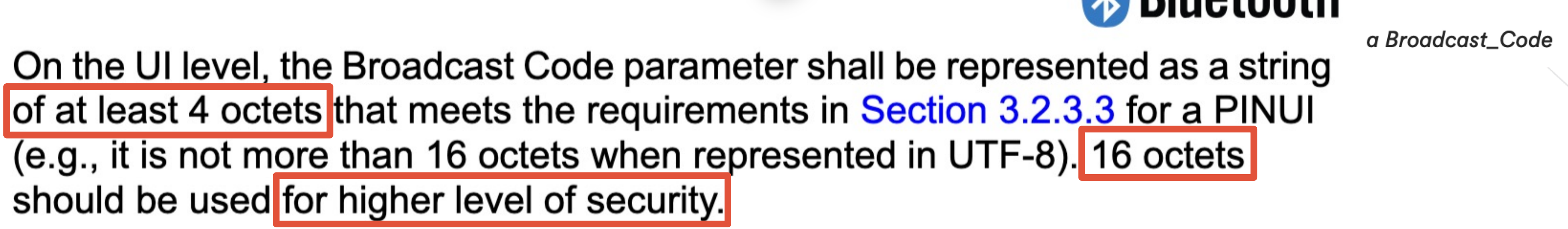


st Details for Tonight's Event

uracast™ Broadcast: Mark's Tasting Evening

Passcode: PinotNoir

Generic Access Profile



a Broadcast_Code

On the UI level, the Broadcast Code parameter shall be represented as a string of at least 4 octets that meets the requirements in Section 3.2.3.3 for a PINUI (e.g., it is not more than 16 octets when represented in UTF-8). 16 octets should be used for higher level of security.

BLUETOOTH CORE SPECIFICATION Version 5.4 | Vol 3, Part C page 1253

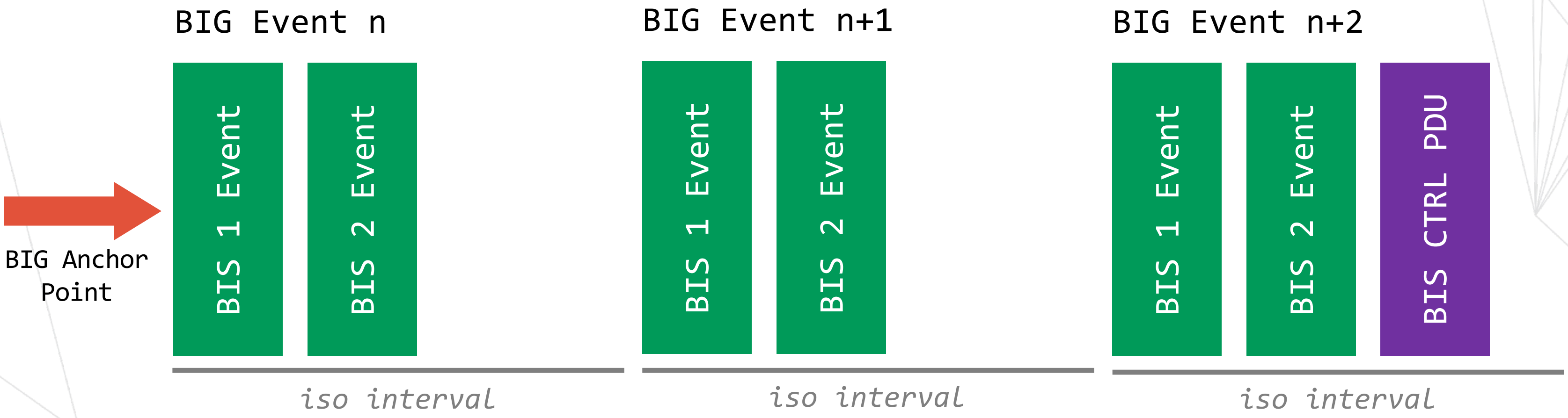
Security Concerns

● **Privacy**
Broadcasts might still supposed to be private, only available to a select group of recipients.

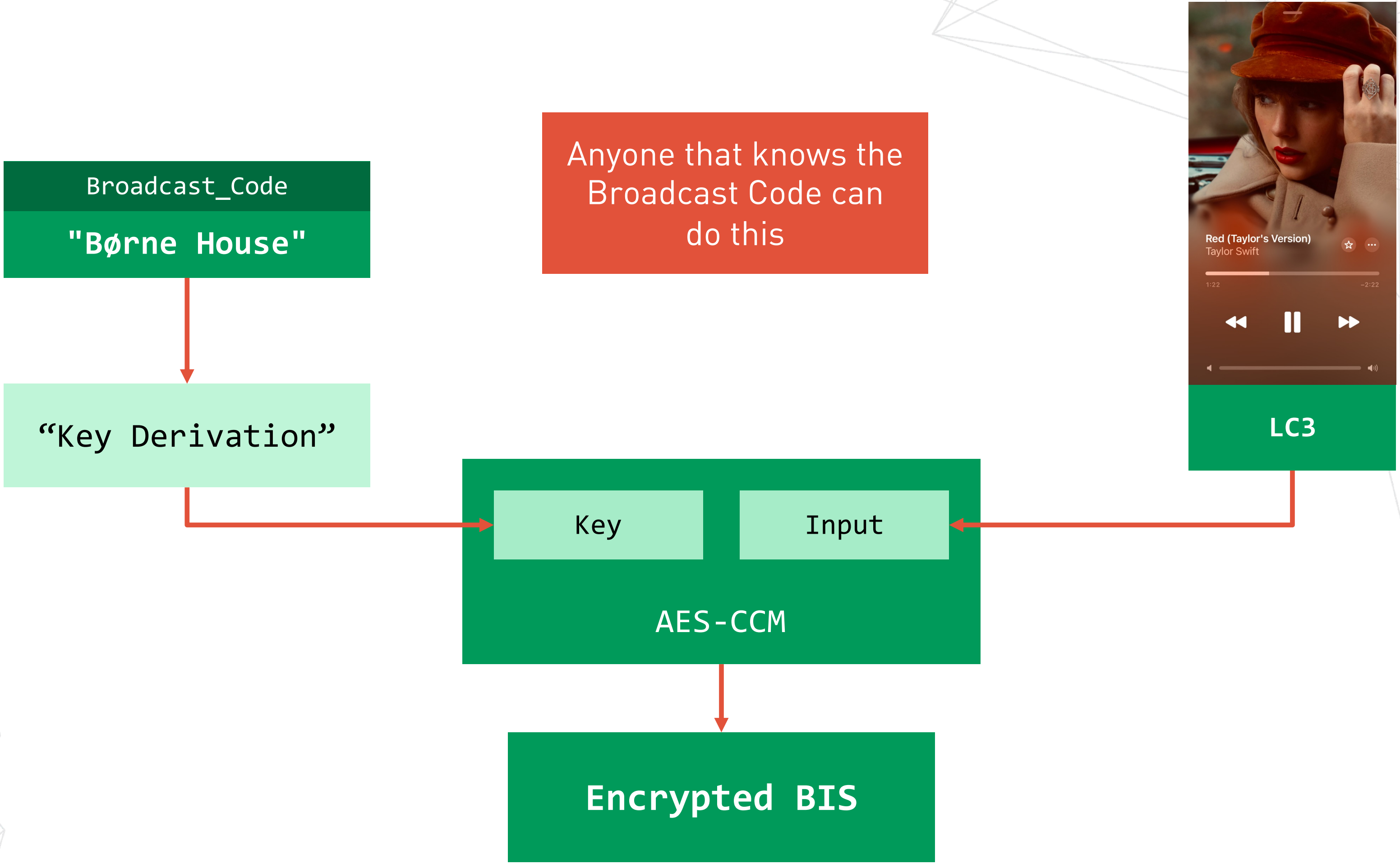
○ **Authenticity**
Broadcast content should not be spoofable.

○ **Availability**
Broadcasts should be resilient to accidental and deliberate interference.

Broadcast Isochronous Streams



Anyone that can receive these packets can also send!



Broadcast_Code

"Børne House"

“Key Derivation”

Anyone that knows the Broadcast Code can do this

LC3

Key

Input

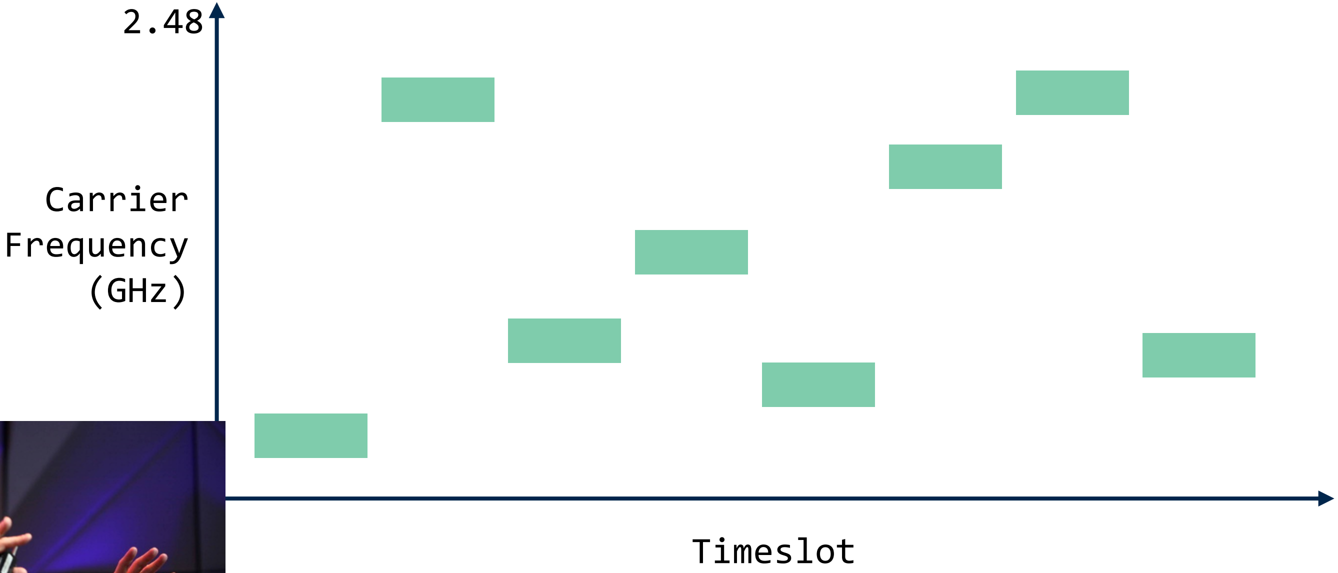
AES-CCM

Encrypted BIS

Security Concerns

- 🟡 **Privacy**
  Broadcasts might still supposed to be private, only available to a select group of recipients.
- 🔴 **Authenticity**
  Broadcast content should not be spoofable.
- ⚪ **Availability**
  Broadcasts should be resilient to accidental and deliberate interference.

Adaptive Frequency Hopping



Not a security boundary

Security Concerns

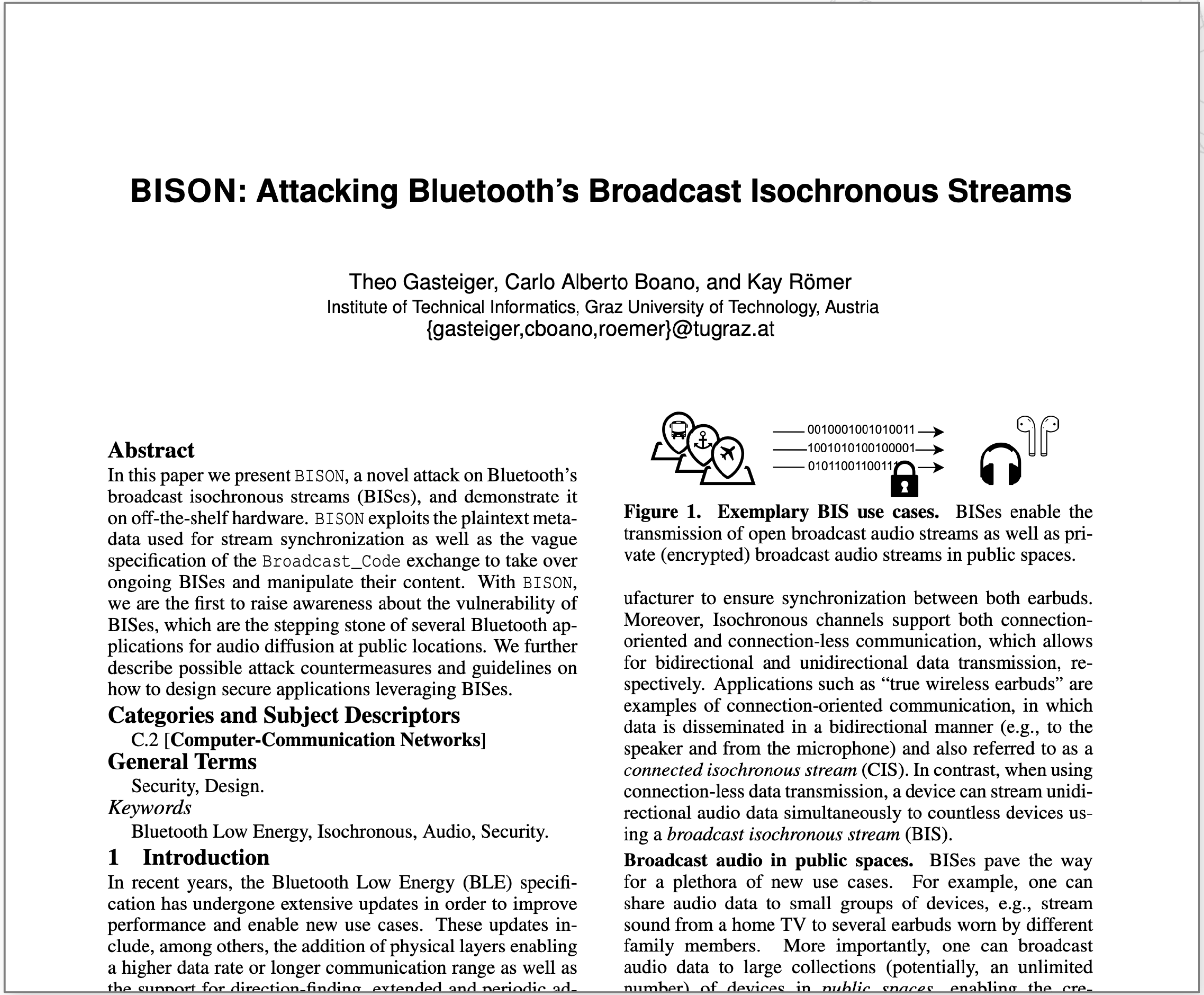
🟡 **Privacy**
Broadcasts might still supposed to be private, only available to a select group of recipients.

🔴 **Authenticity**
Broadcast content should not be spoofable.

🔴 **Availability**
Broadcasts should be resilient to accidental and deliberate interference.

3. Auracast Attacks

BISON



BISON: Attacking Bluetooth's Broadcast Isochronous Streams

Theo Gasteiger, Carlo Alberto Boano, and Kay Römer
Institute of Technical Informatics, Graz University of Technology, Austria
{gasteiger,cboano,roemer}@tugraz.at

Abstract

In this paper we present BISON, a novel attack on Bluetooth's broadcast isochronous streams (BISes), and demonstrate it on off-the-shelf hardware. BISON exploits the plaintext metadata used for stream synchronization as well as the vague specification of the Broadcast_Code exchange to take over ongoing BISes and manipulate their content. With BISON, we are the first to raise awareness about the vulnerability of BISes, which are the stepping stone of several Bluetooth applications for audio diffusion at public locations. We further describe possible attack countermeasures and guidelines on how to design secure applications leveraging BISes.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]

General Terms

Security, Design.

Keywords

Bluetooth Low Energy, Isochronous, Audio, Security.

1 Introduction

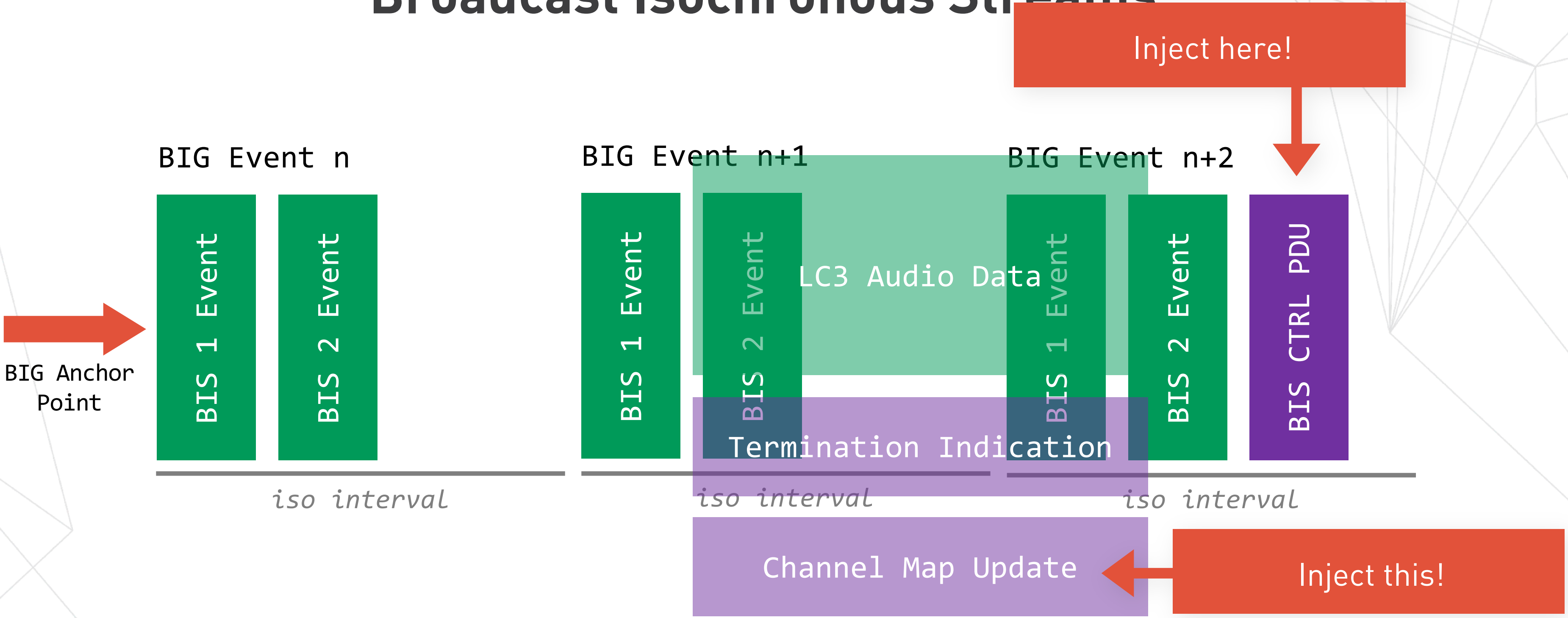
In recent years, the Bluetooth Low Energy (BLE) specification has undergone extensive updates in order to improve performance and enable new use cases. These updates include, among others, the addition of physical layers enabling a higher data rate or longer communication range as well as the support for direction-finding, extended and periodic ad-

Figure 1. Exemplary BIS use cases. BISes enable the transmission of open broadcast audio streams as well as private (encrypted) broadcast audio streams in public spaces.

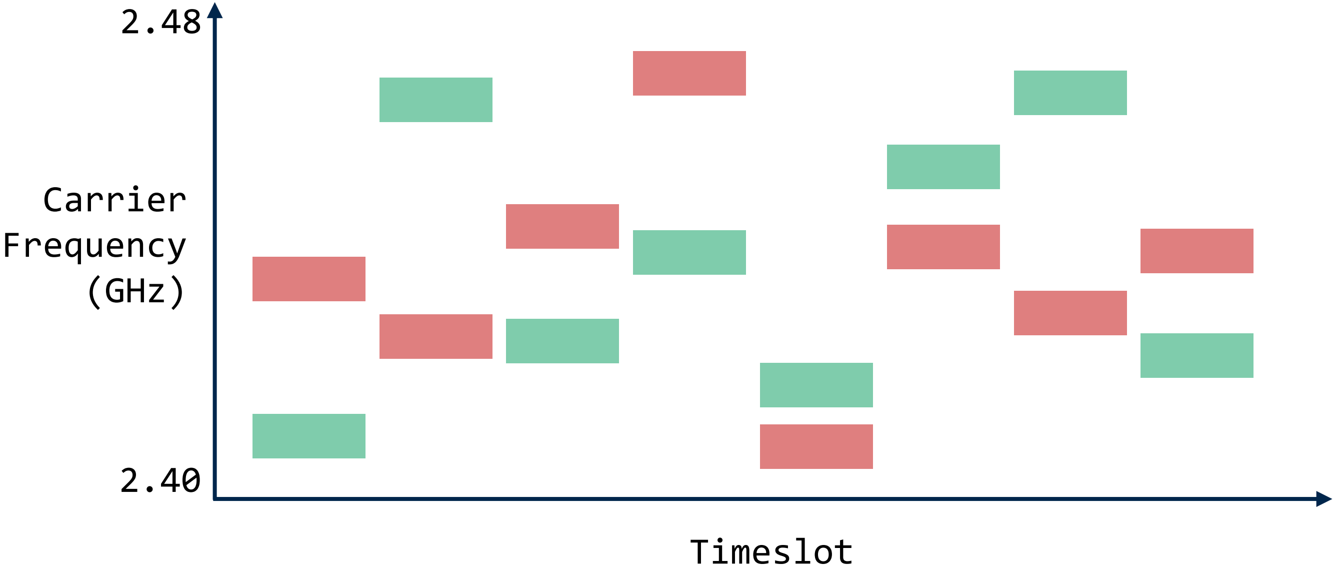
ufacturer to ensure synchronization between both earbuds. Moreover, Isochronous channels support both connection-oriented and connection-less communication, which allows for bidirectional and unidirectional data transmission, respectively. Applications such as "true wireless earbuds" are examples of connection-oriented communication, in which data is disseminated in a bidirectional manner (e.g., to the speaker and from the microphone) and also referred to as a *connected isochronous stream* (CIS). In contrast, when using connection-less data transmission, a device can stream unidirectional audio data simultaneously to countless devices using a *broadcast isochronous stream* (BIS).

Broadcast audio in public spaces. BISes pave the way for a plethora of new use cases. For example, one can share audio data to small groups of devices, e.g., stream sound from a home TV to several earbuds worn by different family members. More importantly, one can broadcast audio data to large collections (potentially, an unlimited number) of devices in *public spaces*, enabling the cre-

Broadcast Isochronous Streams

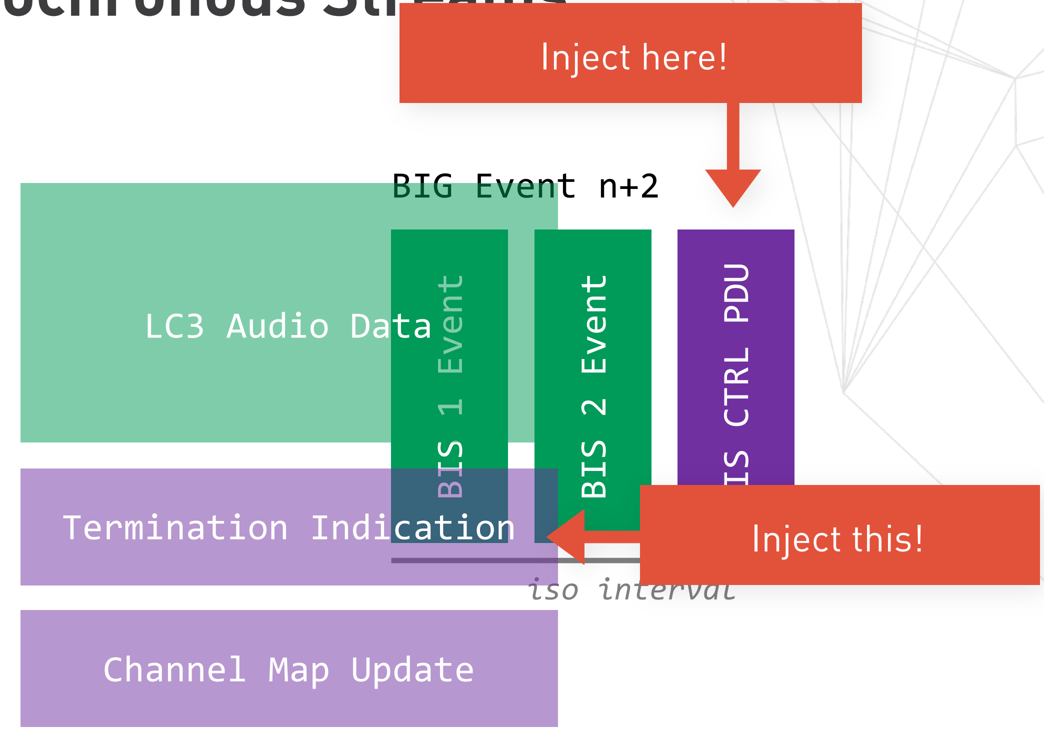


BISON: Updated Channel Map

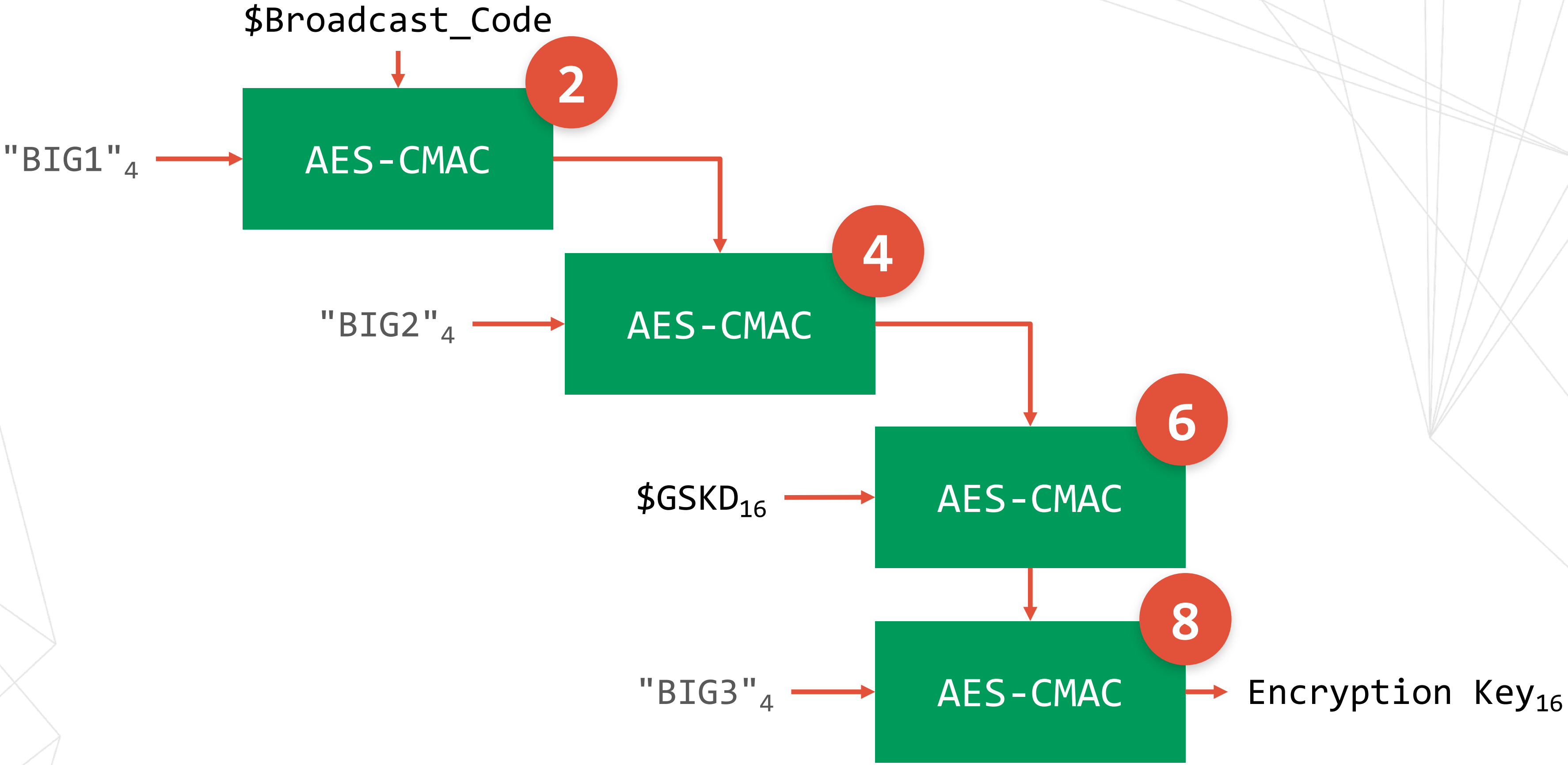


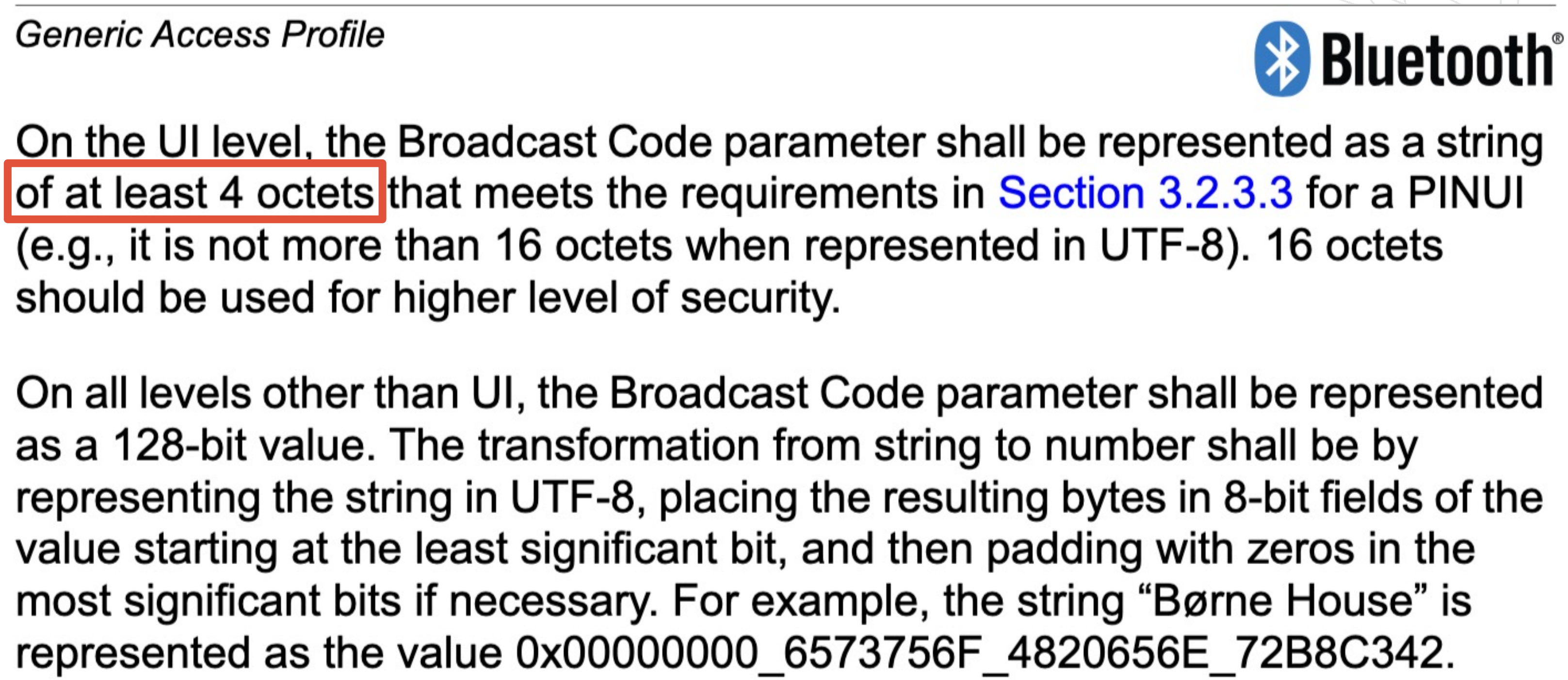
🟥 Attacker Channel Map

Broadcast Isochronous Streams



BISCrack


$Broadcast_Code
2
"BIG1"$_4$
AES-CMAC
4
"BIG2"$_4$
AES-CMAC
6
$GSKD$_{16}$
AES-CMAC
8
"BIG3"$_4$
AES-CMAC
Encryption Key$_{16}$



Generic Access Profile

On the UI level, the Broadcast Code parameter shall be represented as a string of at least 4 octets that meets the requirements in Section 3.2.3.3 for a PINUI (e.g., it is not more than 16 octets when represented in UTF-8). 16 octets should be used for higher level of security.

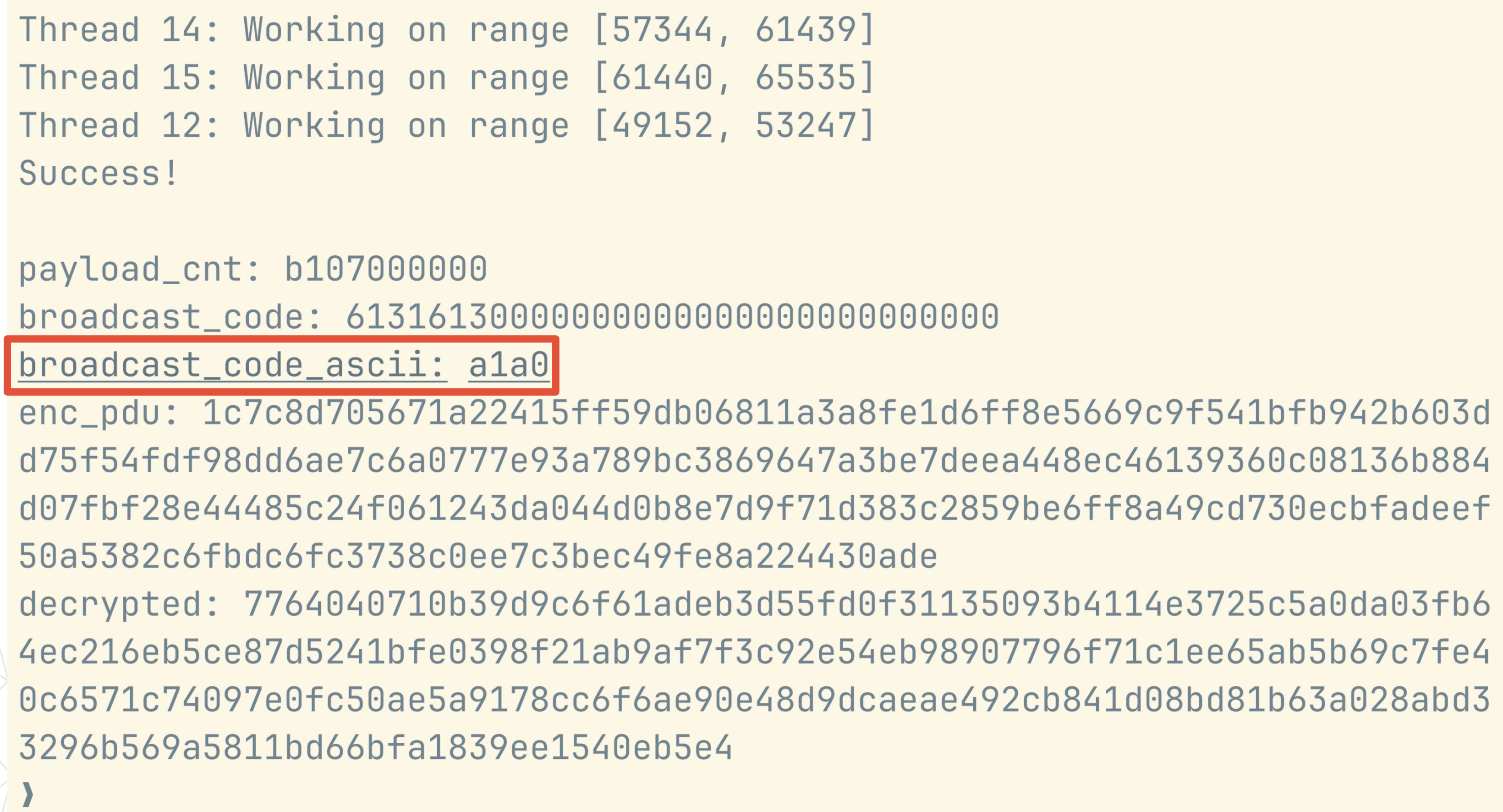
On all levels other than UI, the Broadcast Code parameter shall be represented as a 128-bit value. The transformation from string to number shall be by representing the string in UTF-8, placing the resulting bytes in 8-bit fields of the value starting at the least significant bit, and then padding with zeros in the most significant bits if necessary. For example, the string “Børne House” is represented as the value 0x00000000_6573756F_4820656E_72B8C342.

BLUETOOTH CORE SPECIFICATION Version 5.4 | Vol 3, Part C page 1253

Vendor Implementation

```java
public final String generateBroadcastCode() {
    return UUID.randomUUID().toString().substring(0, 4);
}
```

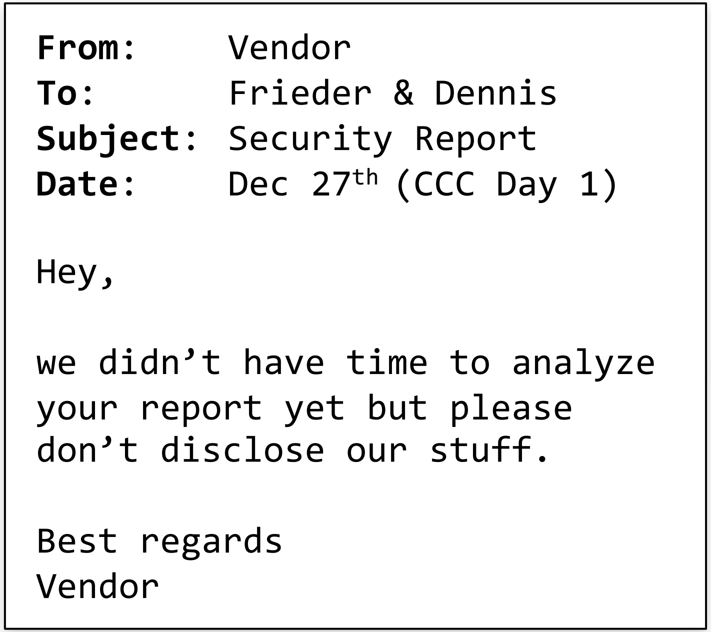
Example 5d65c2e5-7fdc-4e31-8fd1-a4c767c85480 → **5d65**



```
Thread 14: Working on range [57344, 61439]
Thread 15: Working on range [61440, 65535]
Thread 12: Working on range [49152, 53247]
Success!

payload_cnt: b107000000
broadcast_code: 61316130000000000000000000000000
broadcast_code_ascii: a1a0
enc_pdu: 1c7c8d705671a22415ff59db06811a3a8fe1d6ff8e5669c9f541bfb942b603d
d75f54fdf98dd6ae7c6a0777e93a789bc3869647a3be7deea448ec46139360c08136b884
d07fbf28e44485c24f061243da044d0b8e7d9f71d383c2859be6ff8a49cd730ecbfadeef
50a5382c6fbdc6fc3738c0ee7c3bec49fe8a224430ade
decrypted: 776404071 0b39d9c6f61adeb3d55fd0f31135093b4114e3725c5a0da03fb6
4ec216eb5ce87d5241bfe0398f21ab9af7f3c92e54eb98907796f71c1ee65ab5b69c7fe4
0c6571c74097e0fc50ae5a9178cc6f6ae90e48d9dcaeae492cb841d08bd81b63a028abd3
3296b569a5811bd66bfa1839ee1540eb5e4
❯
```

Real-World Case



From: Vendor
To: Frieder & Dennis
Subject: Security Report
Date: Dec 27th (CCC Day 1)

Hey,

we didn’t have time to analyze your report yet but please don’t disclose our stuff.

Best regards
Vendor

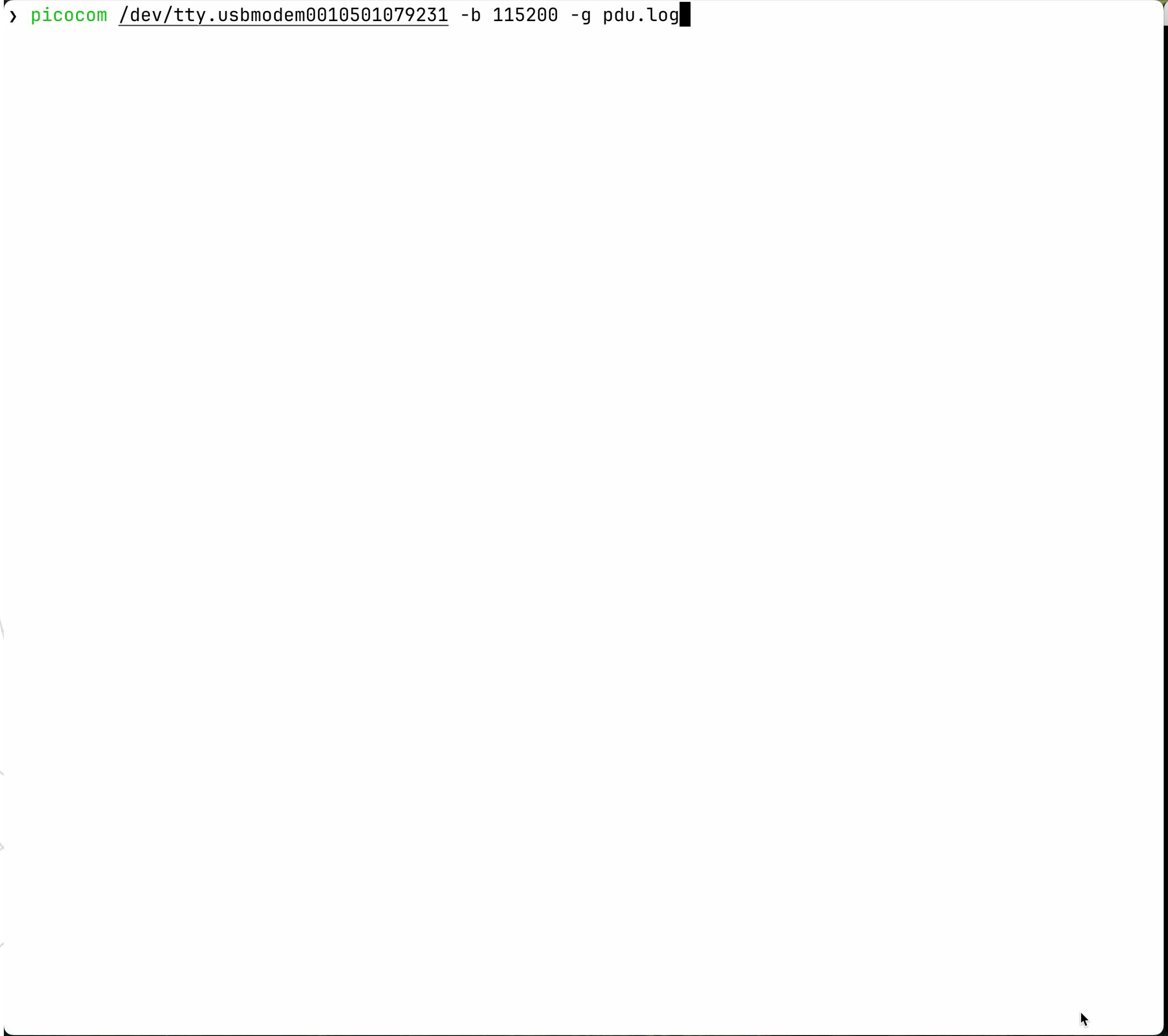
Which is fair, because 90 days are not over yet.
But still sad.

Crackability

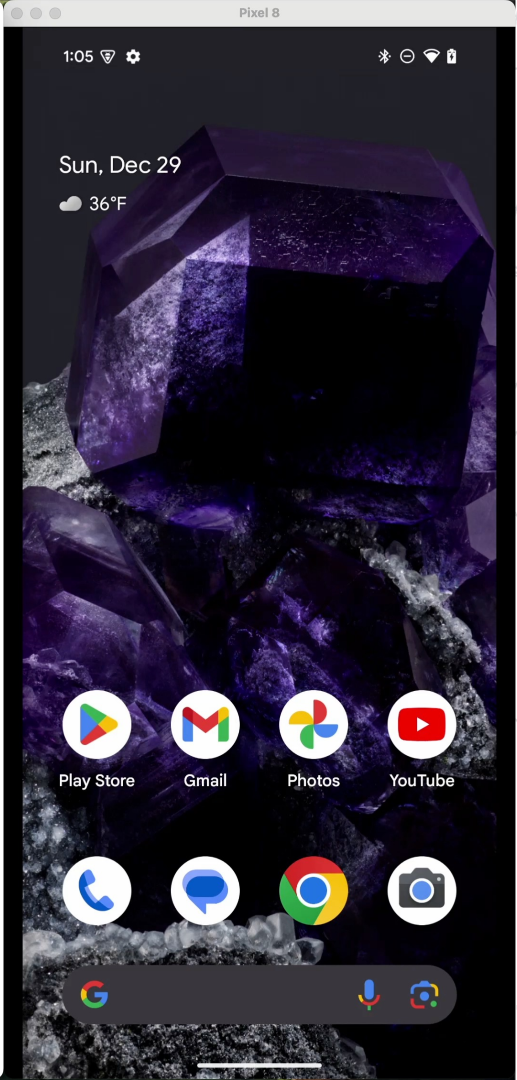
Cracking speed on a mid-range Laptop:

Target	~ Time in s
rockyou.txt	10
4 Byte Hex	0.04
6 Byte Hex	7
4 Byte Alphanumeric	0.25
6 Byte Alphanumeric	2

4. Auracast Hacker‘s Toolkit

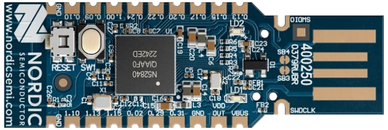


```
❯ picocom /dev/tty.usbmodem0010501079231 -b 115200 -g pdu.log
```



Auracast Hacker’s Toolkit

Contains an open-source Bluetooth Link Layer which is great for research!



nRF52480 USB Dongle

https://www.zephyrproject.org/wp-content/uploads/2023/03/Zephyr_color-13.png

https://www.nordicsemi.com/-/media/Images/Products/DevKits/nRF52-Series/nRF52840-Dongle/nRF52840-Dongle-rev2-prod-page.png

5. Conclusion

Conclusion

- Users
  - Set the Broadcast Code to something strong and try to exhaust all 16 characters.
  - Do not use Auracast for highly sensitive information.
- Vendors
  - Generate secure default broadcast codes and educate users (e.g. requirements of 16 characters).
- Bluetooth Specification
  - Improve key derivation so that a proper AES key is used for encryption.
  - Discuss security properties of Broadcasts.

- But: Try Auracast, the thechnology is cool! Great for hearing aids!
- Unicast LE Audio is (probably) fine and as secure as other LE connections.

ERNW
providing security.

- There's much more to explore in the world of Auracast!
- If you have any questions or want to talk about the topic hit us up or find us at Congress!
- For very interested people we have nRF dongles to play around with Auracast and our tools.
- A blogpost (at insinuator.net) will follow next year.

✉ fsteinmetz@ernw.de
dheinze@ernw.de

www.ernw.de

twillnix@infosec.exchange
ttdennis@chaos.social

www.insinuator.net

Auracast Hacker's Toolkit: