# How to decrypt without keys with GlobalPlatform SCP02 protocol

Gildas Avoine[1,2] Loïc Ferreira[3,1]

Univ Rennes, INSA Rennes, CNRS, IRISA, France

Institut Universitaire de France

Orange Labs, Applied Cryptography Group, Caen, France

July 6, 2018


# SCP02

## Context

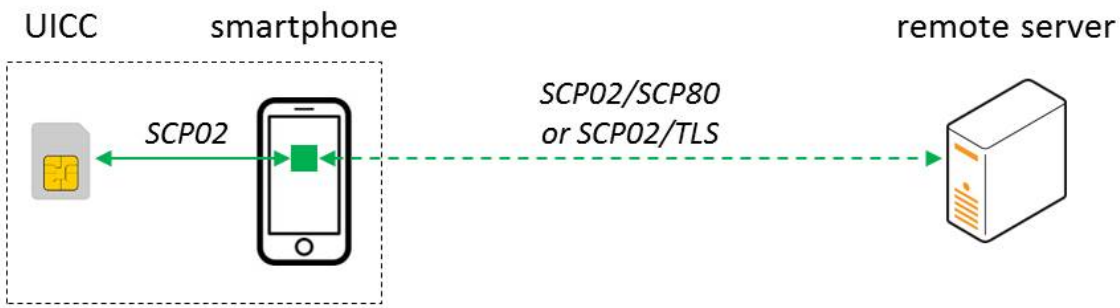
- Security protocol promoted by GlobalPlatform
- Element of a set of security protocols
  - "Symmetric" based: SCP02, SCP03, SCP80
  - "Asymmetric" based: SCP81, SCP10, SCP11
- Likely the most widely used SCP protocol

## Cryptographic functions

- Based on DES/3DES (encryption and MAC; cf. [ISO9797-1] and [ISO10116])

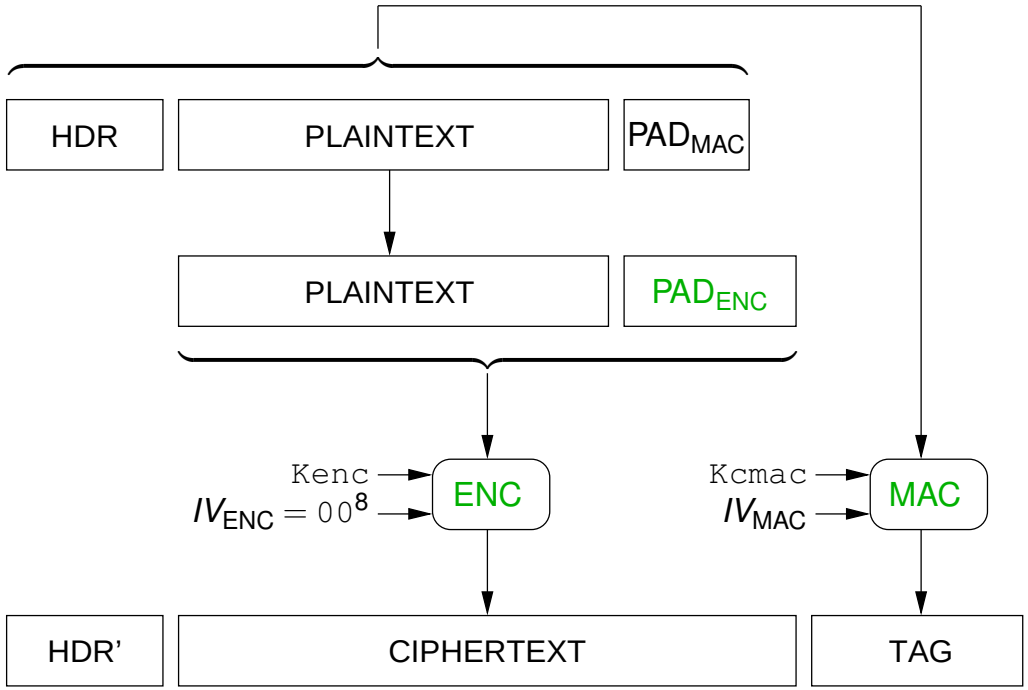
## Purpose

- Secure channel between an "*off card entity*" and a card
- Different security levels: integrity, confidentiality, both
- Remote card management (e.g., applet upload)



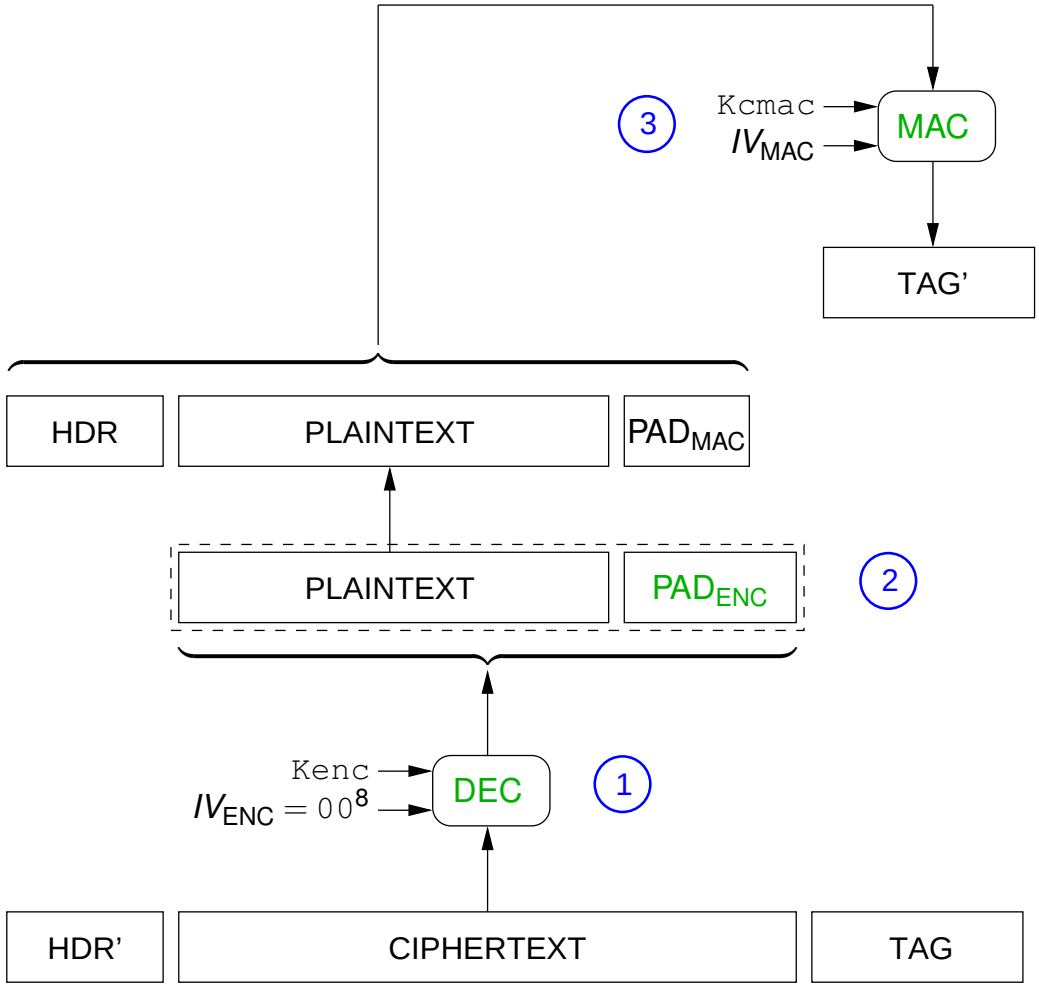
# Encryption and MAC

MAC **and encryption of a command frame**



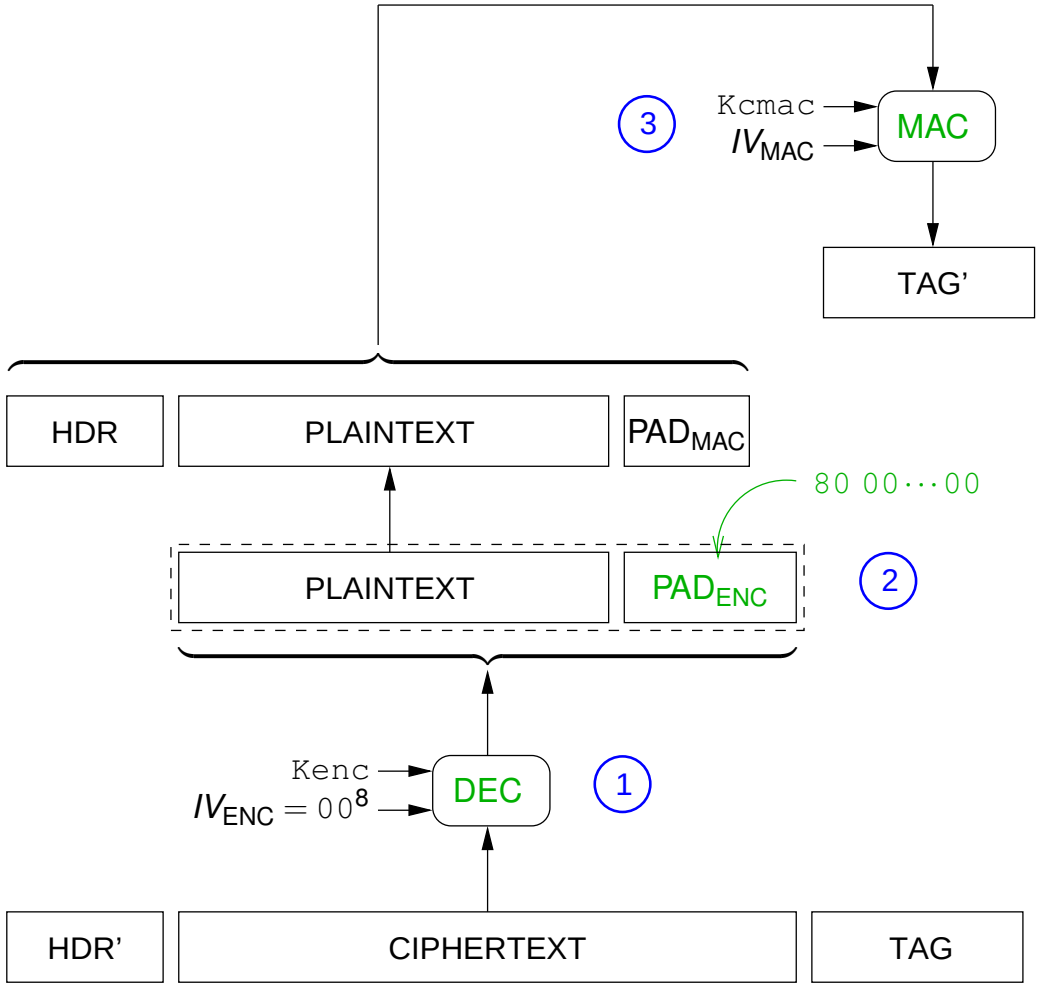
# Encryption and MAC

**Decryption of a command frame**



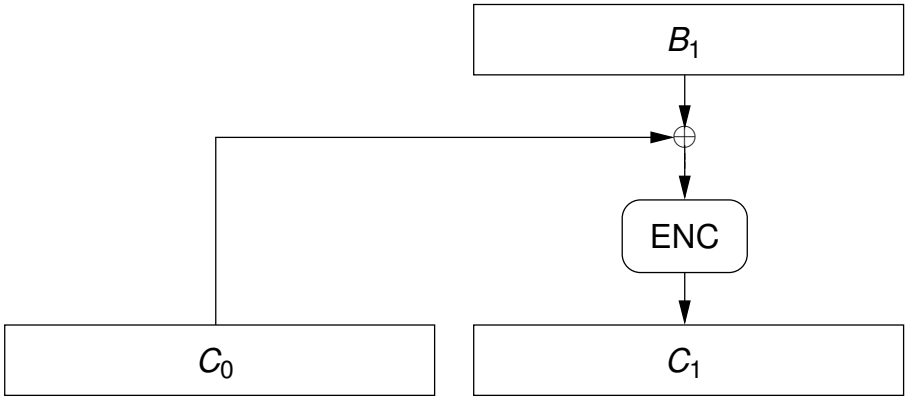
# Encryption and MAC

**Decryption of a command frame**



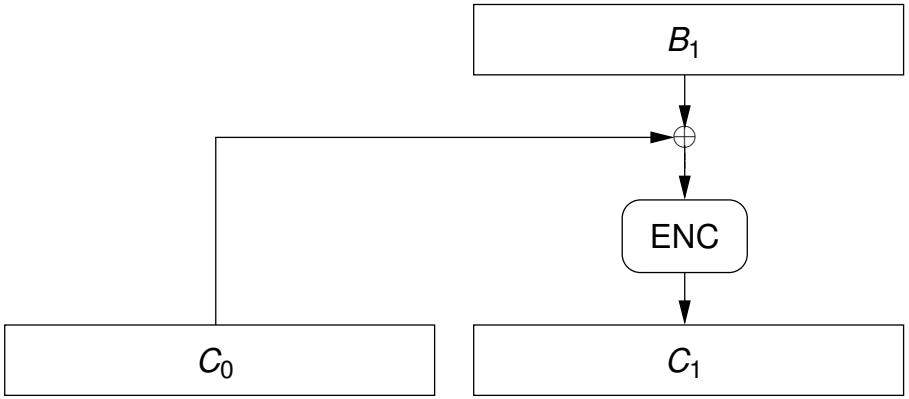
# Encryption and MAC

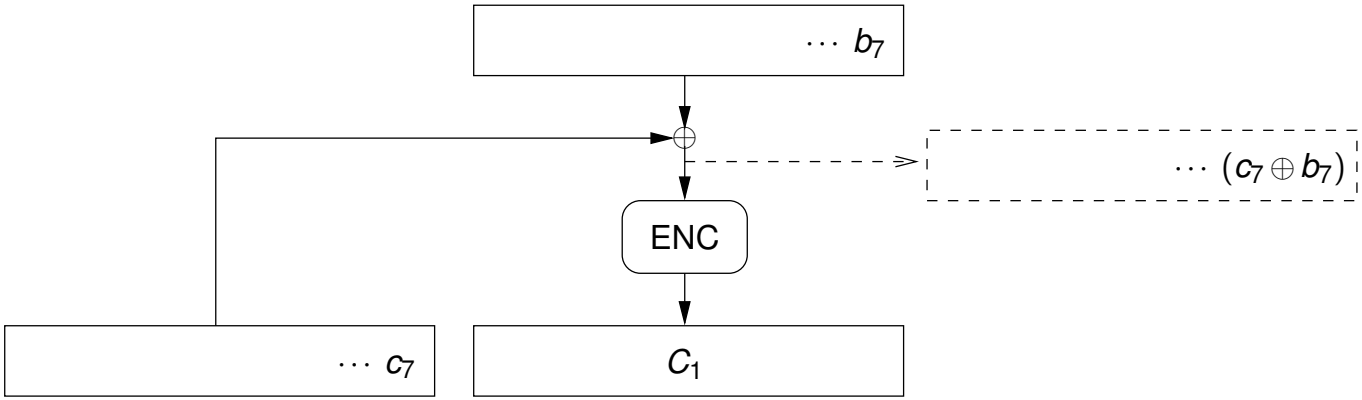
CBC **encryption**



# Encryption and MAC

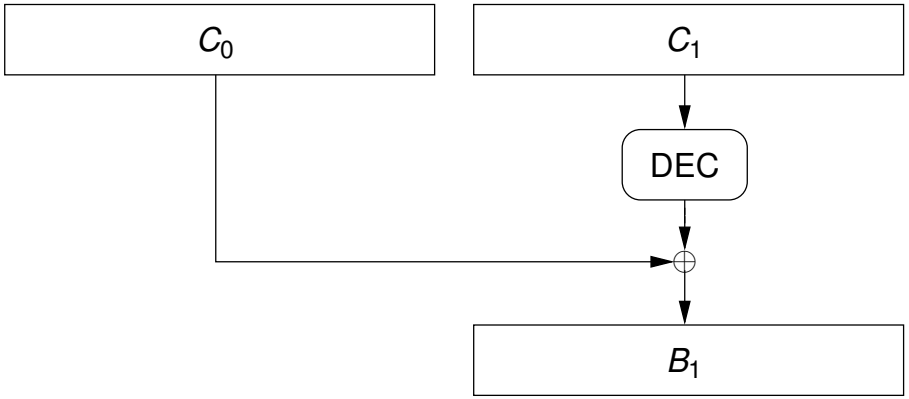
CBC **encryption**



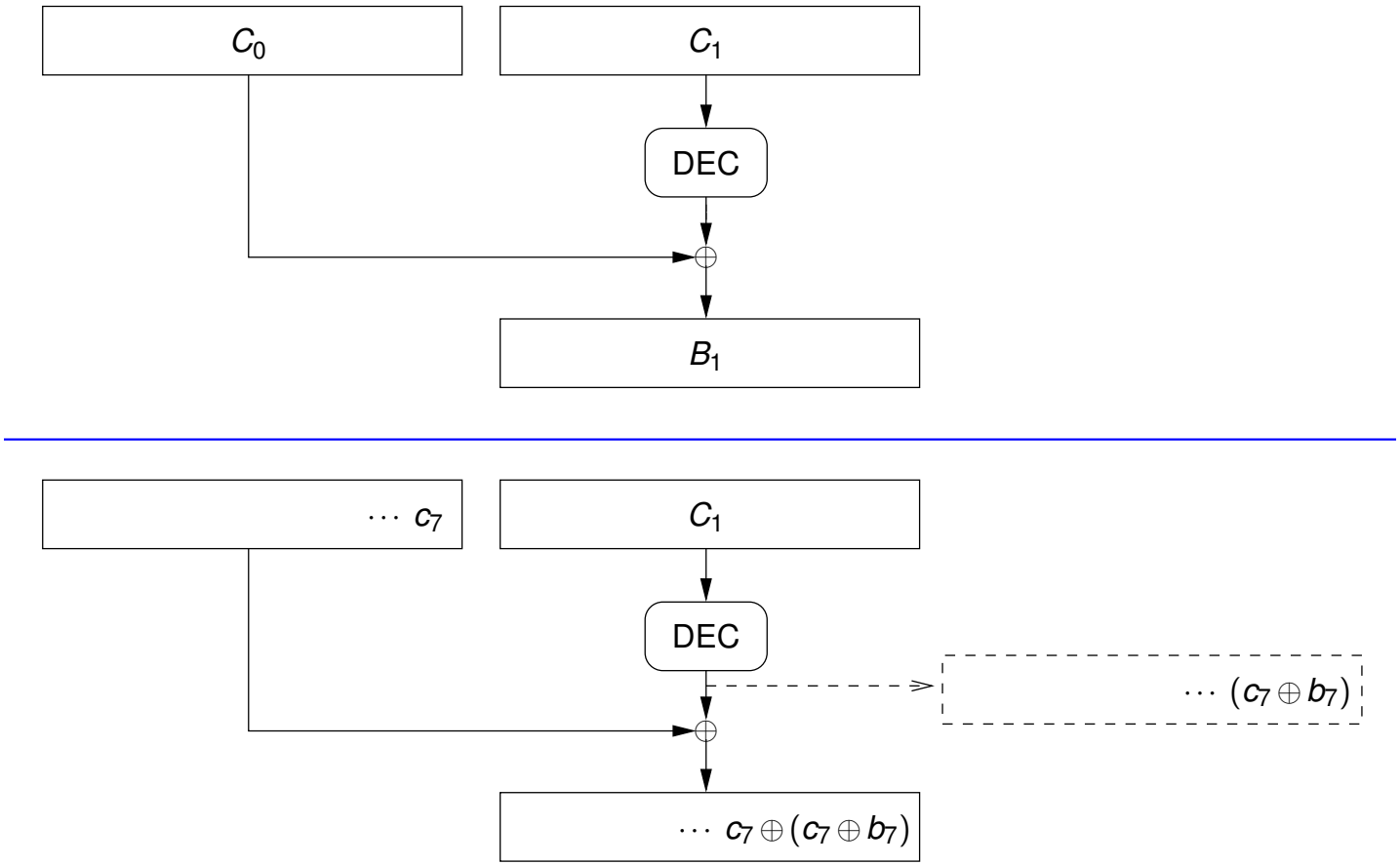
# Encryption and MAC

CBC **decryption**



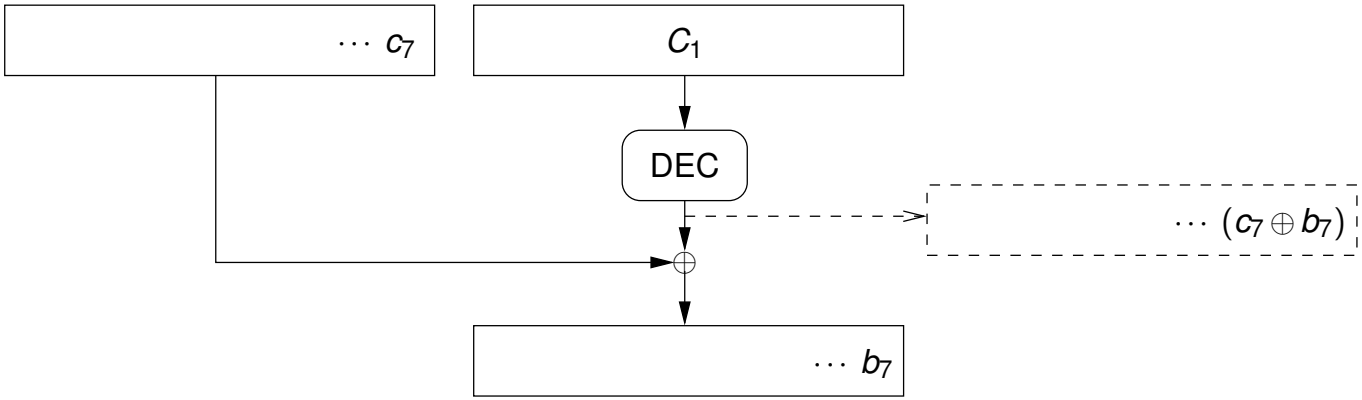
# Encryption and MAC

CBC **decryption**



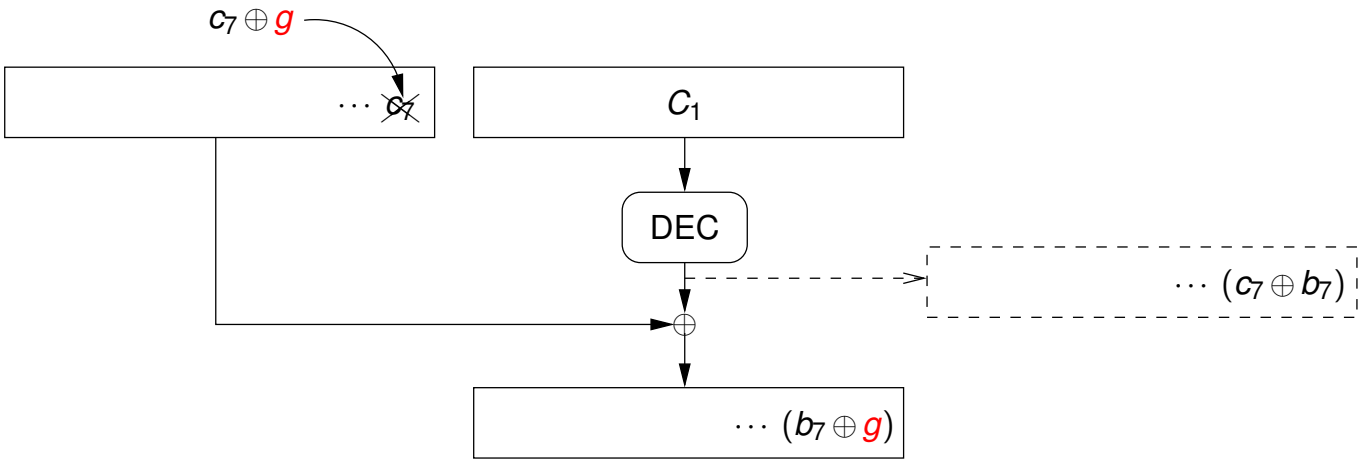
# Encryption and MAC

CBC **decryption and malleability**



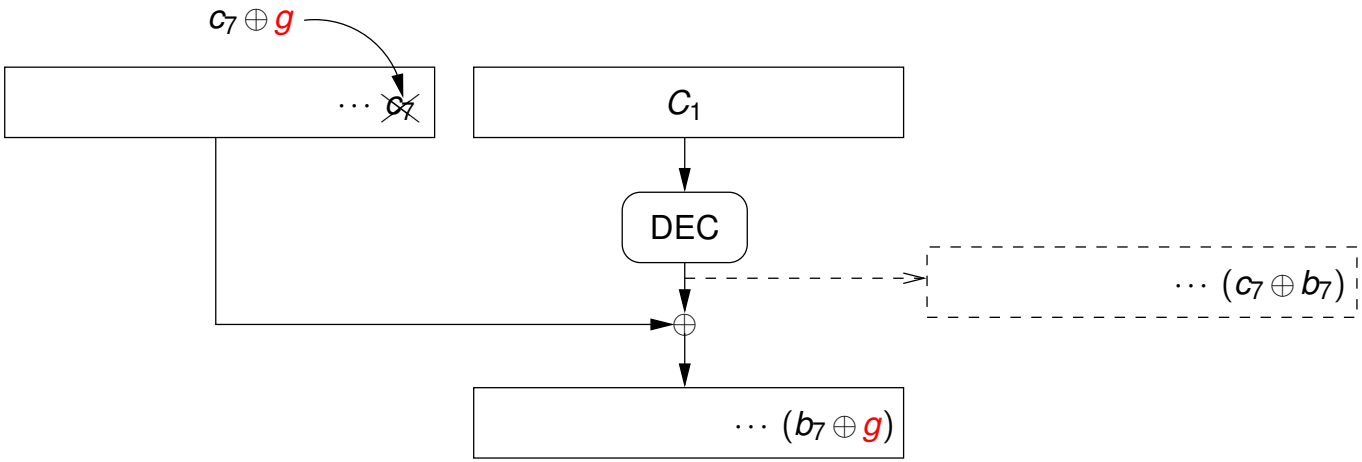
# Encryption and MAC

CBC **decryption and malleability**



# Encryption and MAC

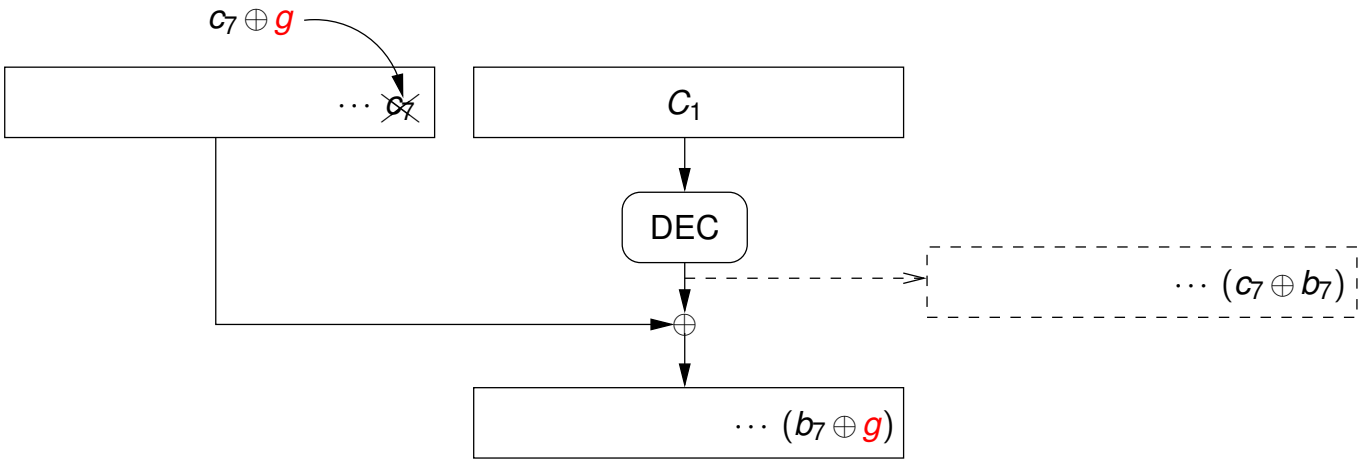
CBC **decryption and malleability**




$g = 13$ $\Rightarrow$ $\Rightarrow$ invalid padding

# Encryption and MAC

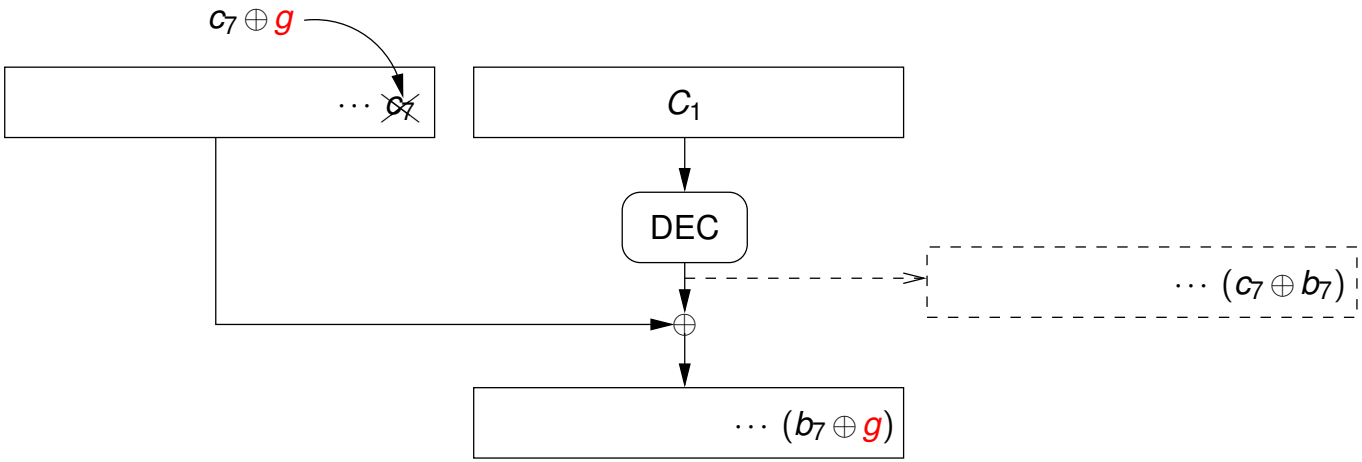
CBC **decryption and malleability**




$g = 13 \quad \Rightarrow \quad \Rightarrow$ invalid padding
$g = 14 \quad \Rightarrow \quad \Rightarrow$ invalid padding

# Encryption and MAC

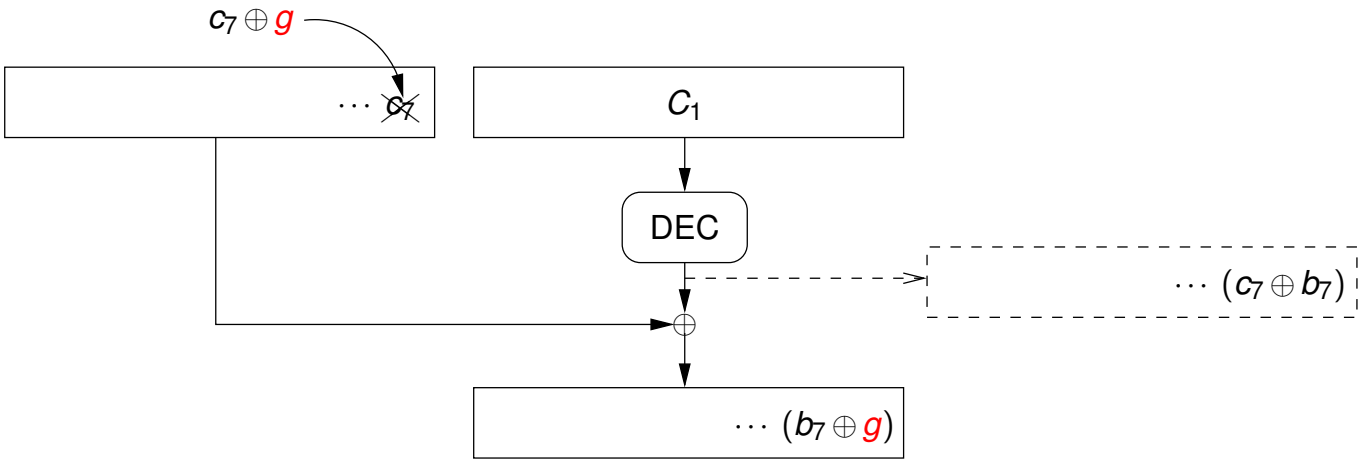
CBC **decryption and malleability**




| | | | |
|---|---|---|---|
| $g = 13$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 14$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 15$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |

# Encryption and MAC

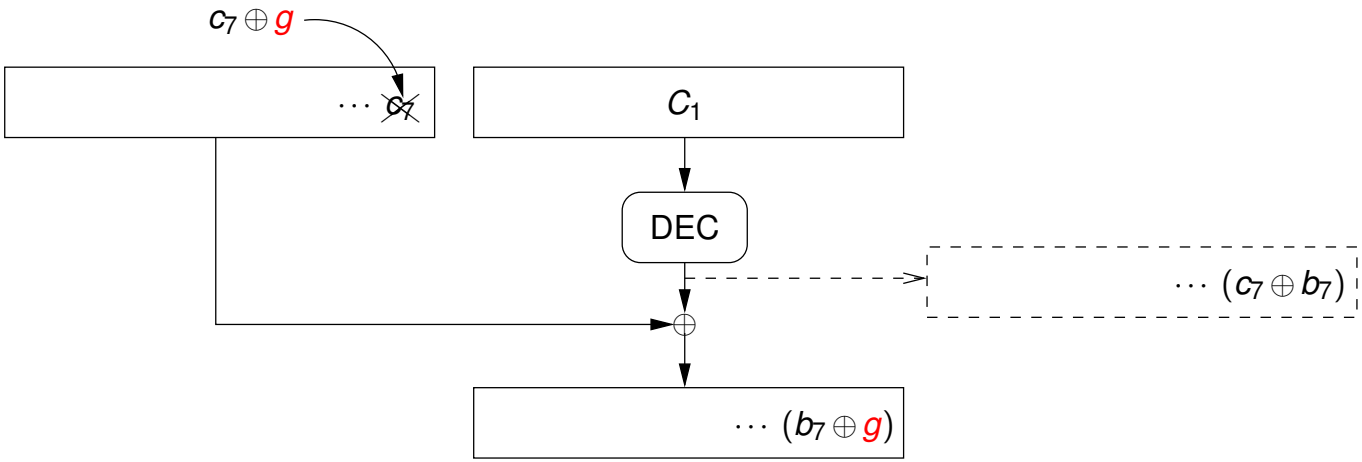
CBC **decryption and malleability**




| | | | |
|---|---|---|---|
| $g = 13$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 14$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 15$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 16$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |

# Encryption and MAC

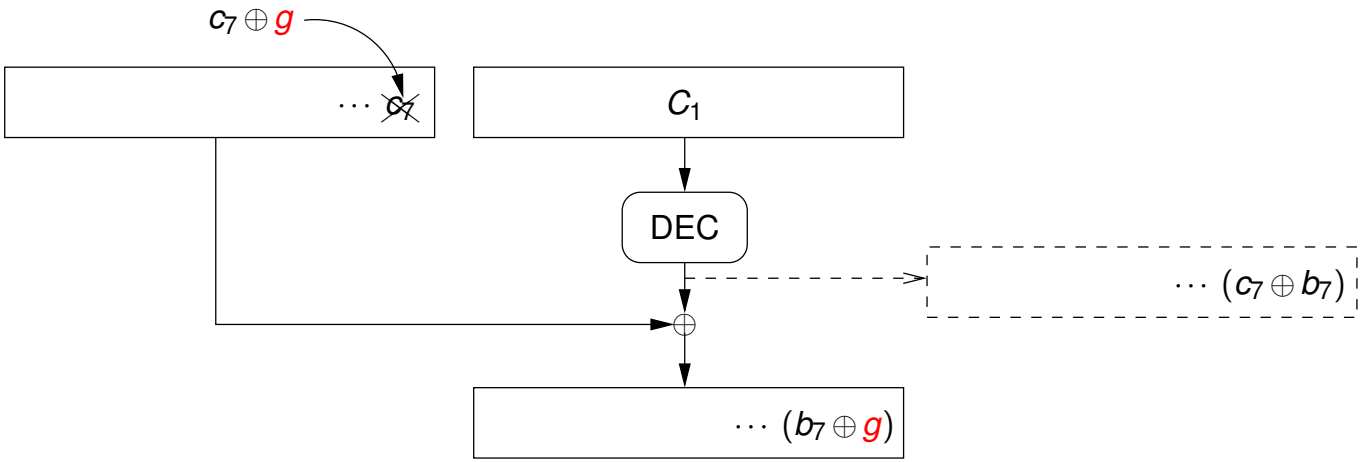
CBC **decryption and malleability**




| | | | |
|---|---|---|---|
| $g = 13$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 14$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 15$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 16$ | $\Rightarrow$ | $\Rightarrow$ | invalid padding |
| $g = 17$ | $\Rightarrow$ | $\Rightarrow$ | valid padding |

# Encryption and MAC

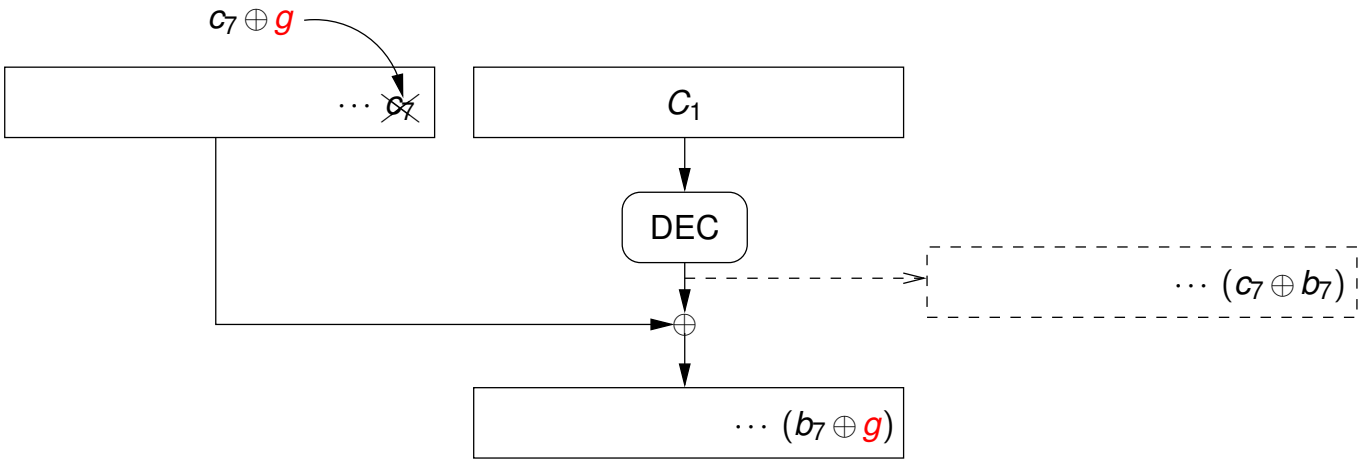
CBC **decryption and malleability**




| | | | | | | |
|---|---|---|---|---|---|---|
| $g = 13$ | $\Rightarrow$ | $b_7 \oplus g = 76$ | $\Rightarrow$ | invalid padding | | |
| $g = 14$ | $\Rightarrow$ | $b_7 \oplus g = 77$ | $\Rightarrow$ | invalid padding | | |
| $g = 15$ | $\Rightarrow$ | $b_7 \oplus g = 78$ | $\Rightarrow$ | invalid padding | | |
| $g = 16$ | $\Rightarrow$ | $b_7 \oplus g = 79$ | $\Rightarrow$ | invalid padding | | |
| $g = 17$ | $\Rightarrow$ | $b_7 \oplus g = 80$ | $\Rightarrow$ | valid padding | $\Rightarrow$ | $b_7 = g \oplus 80 = 97$ |

# Encryption and MAC

CBC **decryption and malleability**



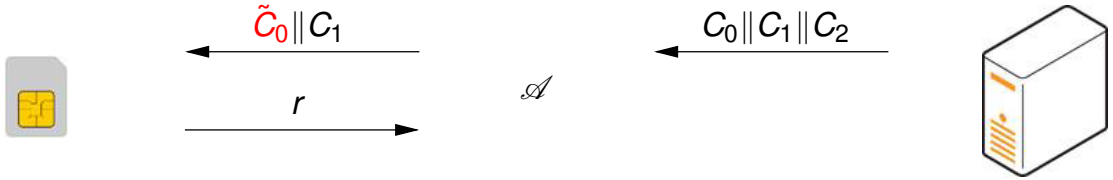

$$
\begin{array}{lcll}
g = 13 & \Rightarrow & b_7 \oplus g = 76 & \Rightarrow \text{ invalid padding} \\
g = 14 & \Rightarrow & b_7 \oplus g = 77 & \Rightarrow \text{ invalid padding} \\
g = 15 & \Rightarrow & b_7 \oplus g = 78 & \Rightarrow \text{ invalid padding} \\
g = 16 & \Rightarrow & b_7 \oplus g = 79 & \Rightarrow \text{ invalid padding} \\
g = 17 & \Rightarrow & b_7 \oplus g = 80 & \Rightarrow \text{ valid padding} \Rightarrow b_7 = g \oplus 80 = 97
\end{array}
$$

- The validity of padding data indicates whether $b_7$ can be found or not.
- Technique called “padding oracle attack” due to Vaudenay in 2002 [V02].

# Generic padding oracle attack

**Padding oracle**

- The remote server wants to send secret data $B_1 = S$ to the card.
- The remote server embeds $B_1 = S$ in an encrypted SCP02 command $C_0 \| C_1 \| C_2$.
- The attacker $\mathscr{A}$ gets $C_0 \| C_1 \| C_2$, replaces it with $\tilde{C}_0 \| C_1$ (adding $g$), and sends the result to the card.

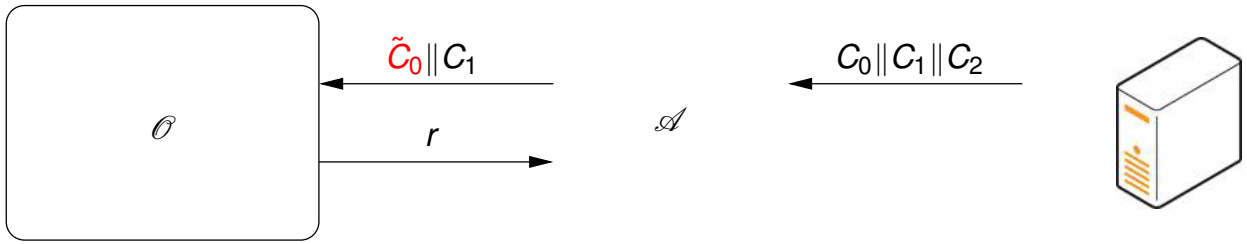



a. if $r =$ "valid padding" then $\mathscr{A}$ decrypts one secret byte
b. if $r =$ "invalid padding" then $\mathscr{A}$
   (i) picks another value $g$
   (ii) goes to a.

# Generic padding oracle attack

**Padding oracle**

- The remote server wants to send secret data $B_1 = S$ to the card.
- The remote server embeds $B_1 = S$ in an encrypted SCP02 command $C_0 \| C_1 \| C_2$.
- The attacker $\mathscr{A}$ gets $C_0 \| C_1 \| C_2$, replaces it with $\tilde{C}_0 \| C_1$ (adding $g$), and sends the result to the card.




a. if $r =$ "valid padding" then $\mathscr{A}$ decrypts one secret byte
b. if $r =$ "invalid padding" then $\mathscr{A}$
   (i) picks another value $g$
   (ii) goes to a.

# Generic padding oracle attack

**Building the padding oracle $\mathscr{O}$**

- Operations during decryption process

| (A) | (B) |
| --- | --- |
| 1. decryption<br>2. padding data: invalid<br>3. ~~MAC~~ | 1. decryption<br>2. padding data: valid<br>3. MAC |

# Generic padding oracle attack

**Building the padding oracle $\mathscr{O}$**

- Operations during decryption process

| (A) | (B) |
|---|---|
| 1. decryption<br>2. padding data: invalid<br>3. ~~MAC~~ | 1. decryption<br>2. padding data: valid<br>3. MAC |

- Oracle $\mathscr{O}$ based on error message (e.g., WTLS, see [V02])

| | | | |
|---|---|---|---|
| ERR_DEC | $\Rightarrow$ | invalid padding data | (A) |
| ERR_MAC | $\Rightarrow$ | valid padding data | (B) |

# Generic padding oracle attack

**Building the padding oracle $\mathscr{O}$**

- Operations during decryption process

| (A) | (B) |
|---|---|
| 1. decryption<br>2. padding data: invalid<br>3. ~~MAC~~ | 1. decryption<br>2. padding data: valid<br>3. MAC |

- Oracle $\mathscr{O}$ based on error message (e.g., WTLS, see [V02])

| | | | |
|---|---|---|---|
| ERR_DEC | $\Rightarrow$ | invalid padding data | (A) |
| ERR_MAC | $\Rightarrow$ | valid padding data | (B) |

- Oracle $\mathscr{O}$ based on computation time (e.g., TLS 1.0, see [CHVV03])

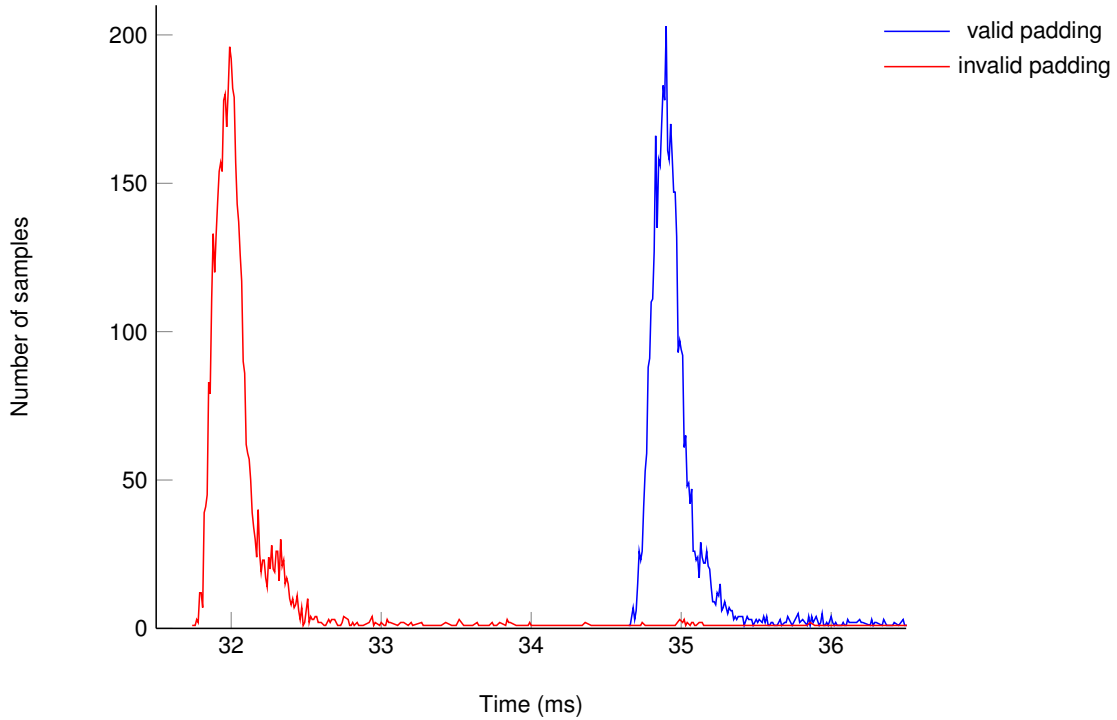
| | | | | | |
|---|---|---|---|---|---|
| time $\searrow$ | $\Rightarrow$ | ~~MAC~~ | $\Rightarrow$ | invalid padding data | (A) |
| time $\nearrow$ | $\Rightarrow$ | MAC | $\Rightarrow$ | valid padding data | (B) |

# Padding oracle in SCP02-compliant smart cards

- The smart card sends *always* a response (status word).
- Invalid padding data or invalid MAC $\Rightarrow$ same error code

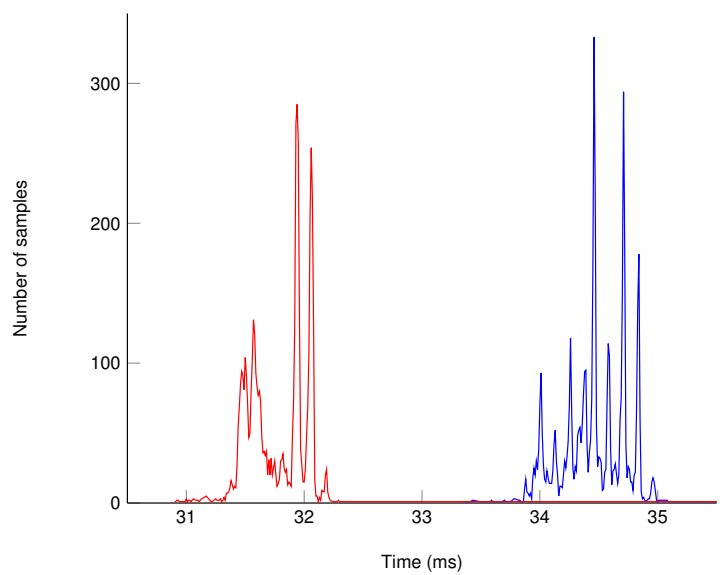
# Padding oracle in SCP02-compliant smart cards

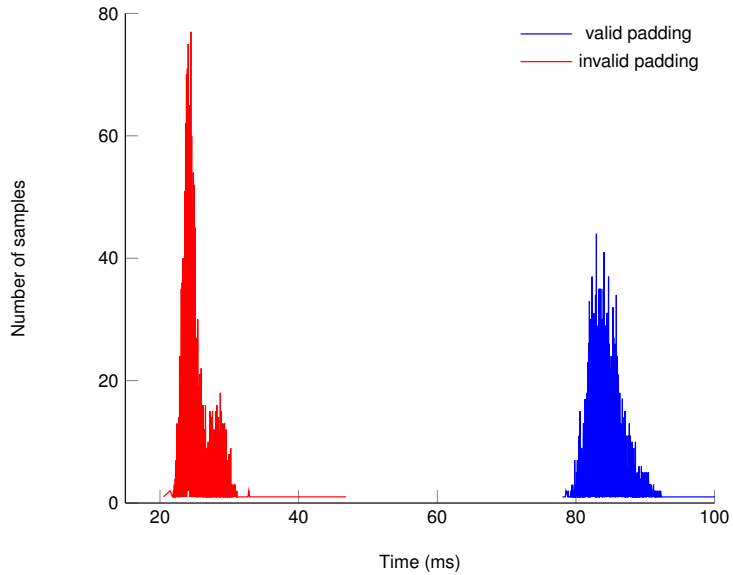
- The smart card sends *always* a response (status word).
- Invalid padding data or invalid MAC $\Rightarrow$ same error code

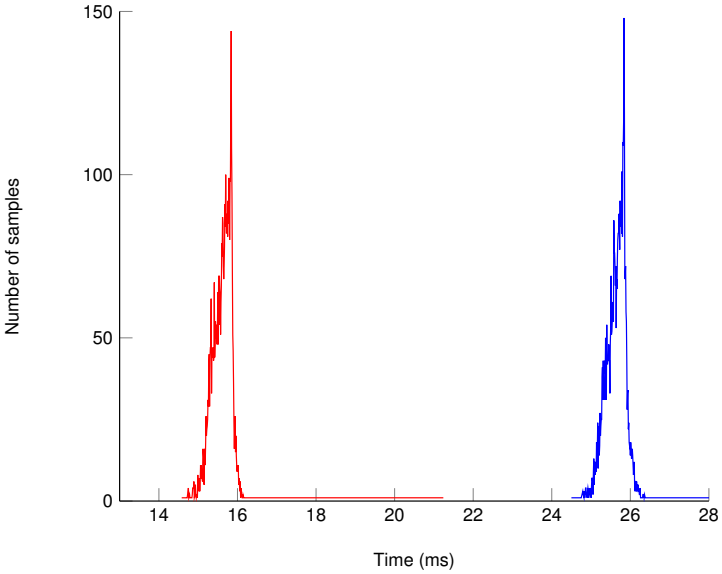


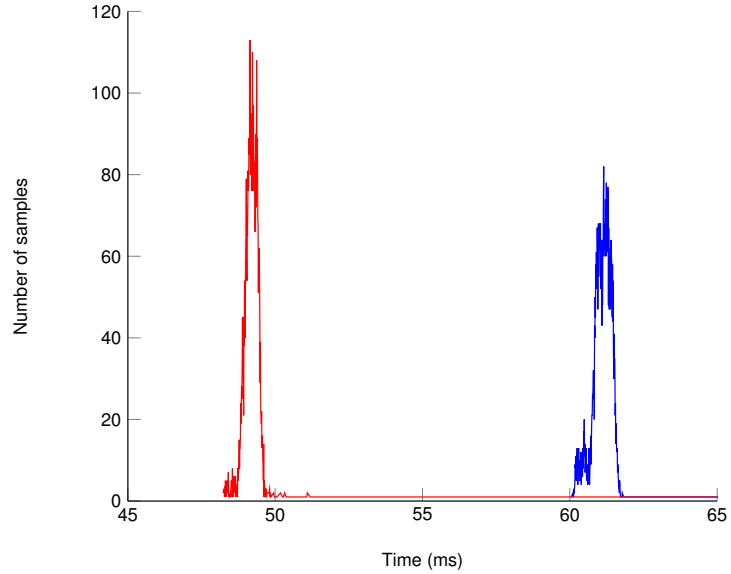

- The card response time reflects the card computation time $\Rightarrow$ suitable padding oracle $\mathcal{O}$

# Padding oracle in SCP02-compliant smart cards



# Padding oracle attack against SCP02-compliant smart cards

**Practical experiments**

- Experimental setting: card connected to a card reader (4 card readers, wired and wireless)
- 10 smart cards from 6 card manufacturers
- SIM cards, generic Java cards

- Experiment: find a 16-byte secret key sent to the smart card in an encrypted SCP02 command
- 300 experiments/card $\Rightarrow$ 100 % success
- Practical complexity $\in [127.75, 133.38]$ close to best average case (128)
- Time to find 16 bytes: 2.7 mn to 11.4 mn (variable response time from the smart card)

| M | C | $\mu_W$ (ms) | $\mu_R$ (ms) | $t_{min}$ (ms) | $m$ | $\tau_+$ (%) | $K_W$ | $K_R$ | $Z$ | $Z/n$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | A | 39.60 | 42.59 | 41.00 | 28 | 0.16 | 1 | 3 | 2055.71 | 128.48 |
| | B | 40.19 | 43.94 | 42.00 | 28 | 0.44 | 1 | 3 | 2077.78 | 129.86 |
| 2 | C | 25.17 | 84.34 | 75.00 | 0 | 0.00 | 1 | 2 | 2043.95 | 127.75 |
| | D | 26.64 | 34.36 | 32.00 | 0 | 0.00 | 1 | 2 | 2066.54 | 129.16 |
| 3 | E | 15.61 | 25.65 | 23.00 | 0 | 0.00 | 1 | 2 | 2134.03 | 133.38 |
| 4 | F | 31.81 | 34.48 | 33.00 | 28 | 0.48 | 1 | 3 | 2109.71 | 131.86 |
| | G | 15.64 | 18.53 | 17.00 | 0 | 0.28 | 1 | 3 | 2103.62 | 131.48 |
| 5 | H | 25.18 | 84.86 | 72.00 | 0 | 0.00 | 1 | 2 | 2048.34 | 128.02 |
| 6 | I | 25.90 | 35.85 | 32.00 | 0 | 0.06 | 1 | 3 | 2108.60 | 131.79 |
| | J | 14.32 | 19.92 | 17.50 | 0 | 0.10 | 1 | 2 | 2094.85 | 130.93 |

# Padding oracle attack against SCP02-compliant smart cards

**Requirements**

1. The attacker sits between the remote server and the card at a point where she can directly eavesdrop on SCP02 encrypted commands and send modified commands to the card.
2. The attacker is able to discriminate response times corresponding to a valid and an invalid padding.
3. The remote server repeatedly sets up a (new) secure channel with the card.
4. The same secret information is sent through each such secure channel.
5. The secret information is sent at a predictable position.

NB: req. 4 $\Rightarrow$ req. 3 (and 5)

# Attack scenario

**Trojan in a smartphone**


1. The victim downloads from a popular store an infected application (embedding a Trojan ✹) into his smartphone.
2. The Trojan gets access to the memory space of the legitimate application ■ (through privileges escalation).
3. The Trojan can apply the attack: it reads, and modifies the encrypted SCP02 commands received by the legitimate application.

# Countermeasures for SCP02-compliant smart cards

- Correct implementation (not possible for deployed cards)
- Use additional security mechanisms
- Use `PUT_KEY` command to send sensitive data
- Do not send too many times the same data (server side)

# Responsible disclosure

**Disclosure (October 2017-April 2018)**

- Card manufacturers
- GlobalPlatform

- ANSSI has been also informed.

**Publication**

- Paper accepted at conference CHES 2018

  *Attacking GlobalPlatform SCP02-compliant Smart Cards Using a Padding Oracle Attack*,
  Gildas Avoine (INSA Rennes), Loïc Ferreira

- Published in IACR Transactions on Cryptographic Hardware and Embedded Systems [AF18].
  To be presented at CHES conference in September 2018.

# Take-away

- The padding oracle attack against SCP02-compliant smart cards is possible because of
  - a theoretical flaw lying in the SCP02 protocol (Encrypt-and-MAC scheme),
  - exploited by means of a timing side-channel provided by the smart cards (implementation).
- Several requirements to be fulfilled in order for the attack to be successful.
- Practical attack
  - Experimental setting: 10 smart cards from 6 manufacturers.
  - How many smart cards impacted in real life?
- SCP02 is now deprecated (March 2018): use SCP03 instead.

# Thank you

Questions?


# References


[SCP02] GlobalPlatform. *GlobalPlatform – Card Specification*, version 2.3.1, ref. GPC_SPE_034, March 2018.

[SCP03] GlobalPlatform. *GlobalPlatform Card Technology – Secure Channel Protocol '03' – Card Specification v2.2 – Amendment D*, version 1.1, ref. GPC_SPE_014, July 2014.

[ISO9797-1] ISO/IEC JTC 1/SC 27. *ISO/IEC 9797-1:2011 – Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*, 2011.

[ISO10116] ISO/IEC JTC 1/SC 27. *ISO/IEC 10116:2017 – Information technology – Security techniques – Modes of operation for an n-bit block cipher*, 2017.

[ISO7816-4] ISO/IEC JTC 1/SC 17. *ISO/IEC 7816-4:2013 – Information technology – Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*, 2013.

[CHVV03] B. Canvel, A. Hiltgen, S. Vaudenay, M. Vuagnoux. *Password interception in a SSL/TLS channel*. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*. LNCS, vol. 2729, pp. 583-599. Springer, 2003.

[V02] S. Vaudenay. *Security Flaws Induced by CBC Padding – Applications to SSL, IPSEC, WTLS...* In L. R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*. LNCS, vol. 2332, pp. 534-545. Springer, 2002.

[ST16] M. Sabt, J. Traoré. *Cryptanalysis of GlobalPlatform Secure Channel Protocols*. In L. Chen, D. McGrew, C. Mitchell, editors, *Security Standardisation Research – SSR 2016*. LNCS, vol. 10074, pp. 62-91. Springer, 2016. `https://eprint.iacr.org/2017/032`

[AF18] G. Avoine, L. Ferreira. *Attacking GlobalPlatform SCP02-compliant Smart Cards Using a Padding Oracle Attack*. In IACR Transactions on Cryptographic Hardware and Embedded Systems, Issue 2, pp. 149-170, May 2018. `https://tches.iacr.org/index.php/TCHES/article/view/878`