OBERTHUR TECHNOLOGIES

FDTC 2013

Fault Diagnosis and Tolerance in Cryptography

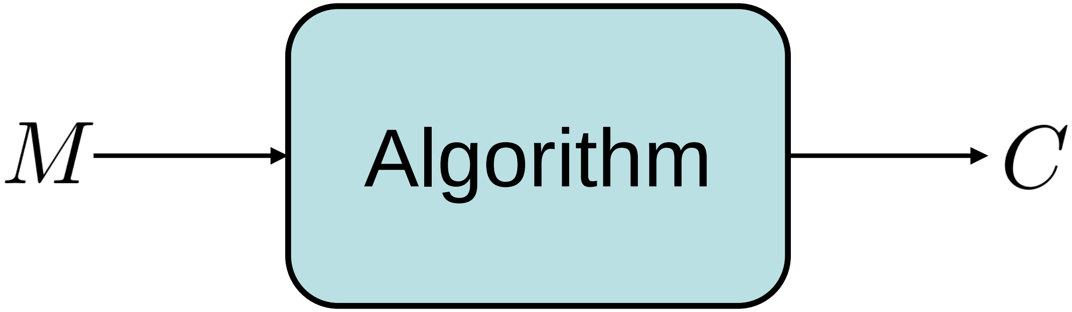
Fault Analysis of Infective AES Computations

Alberto Battistello and Christophe Giraud

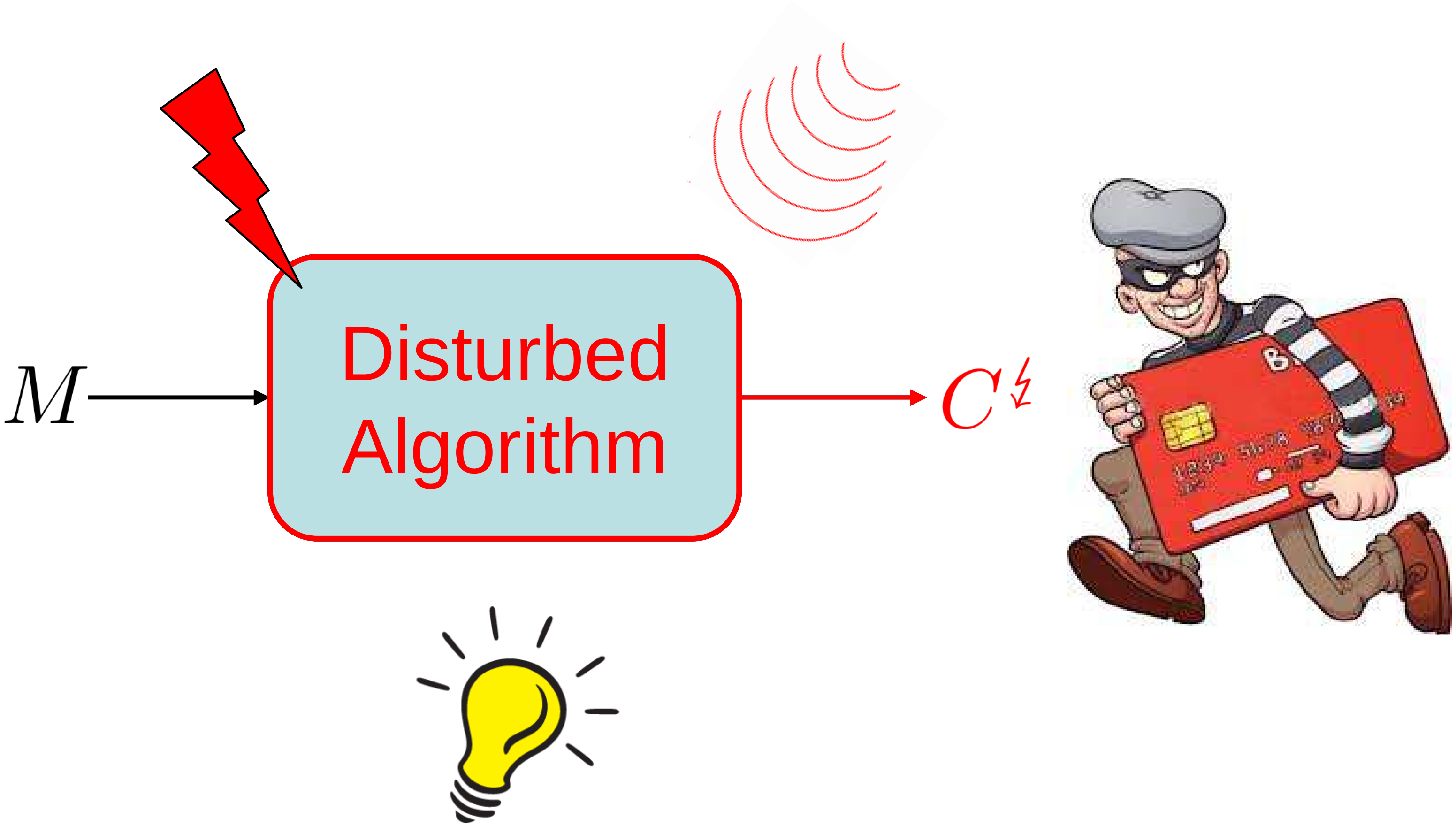
Overview

- Introduction
- Attacks
  - FDTC 2012 Countermeasure
  - LatinCrypt 2012 Countermeasure
- Conclusion

- Introduction
- Attacks
  - FDTC 2012 Countermeasure
  - LatinCrypt 2012 Countermeasure
- Conclusion

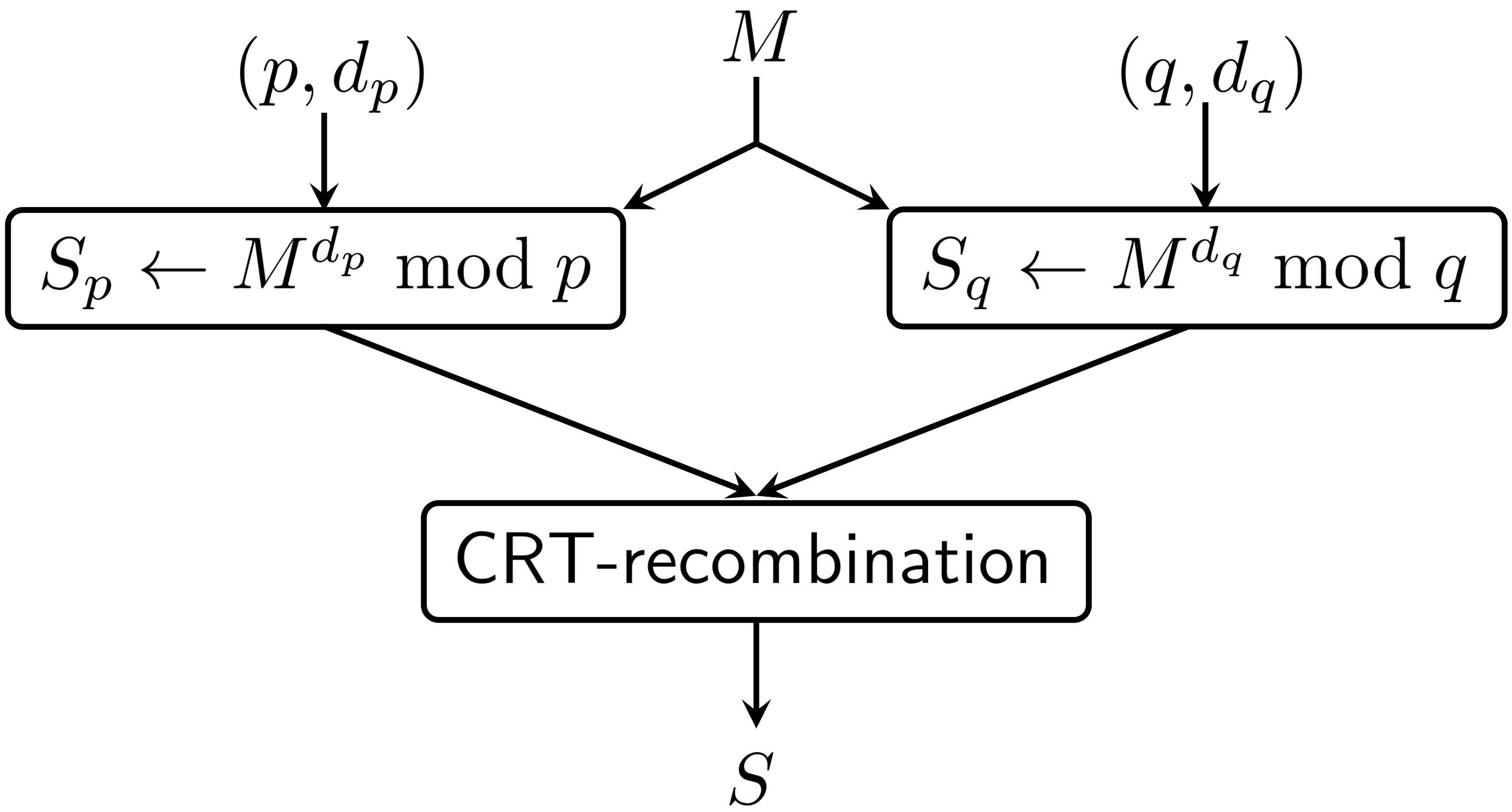

M
Algorithm
C

Fault Attacks



An example of Fault Attack

- Instead of computing $S = M^d \bmod N$

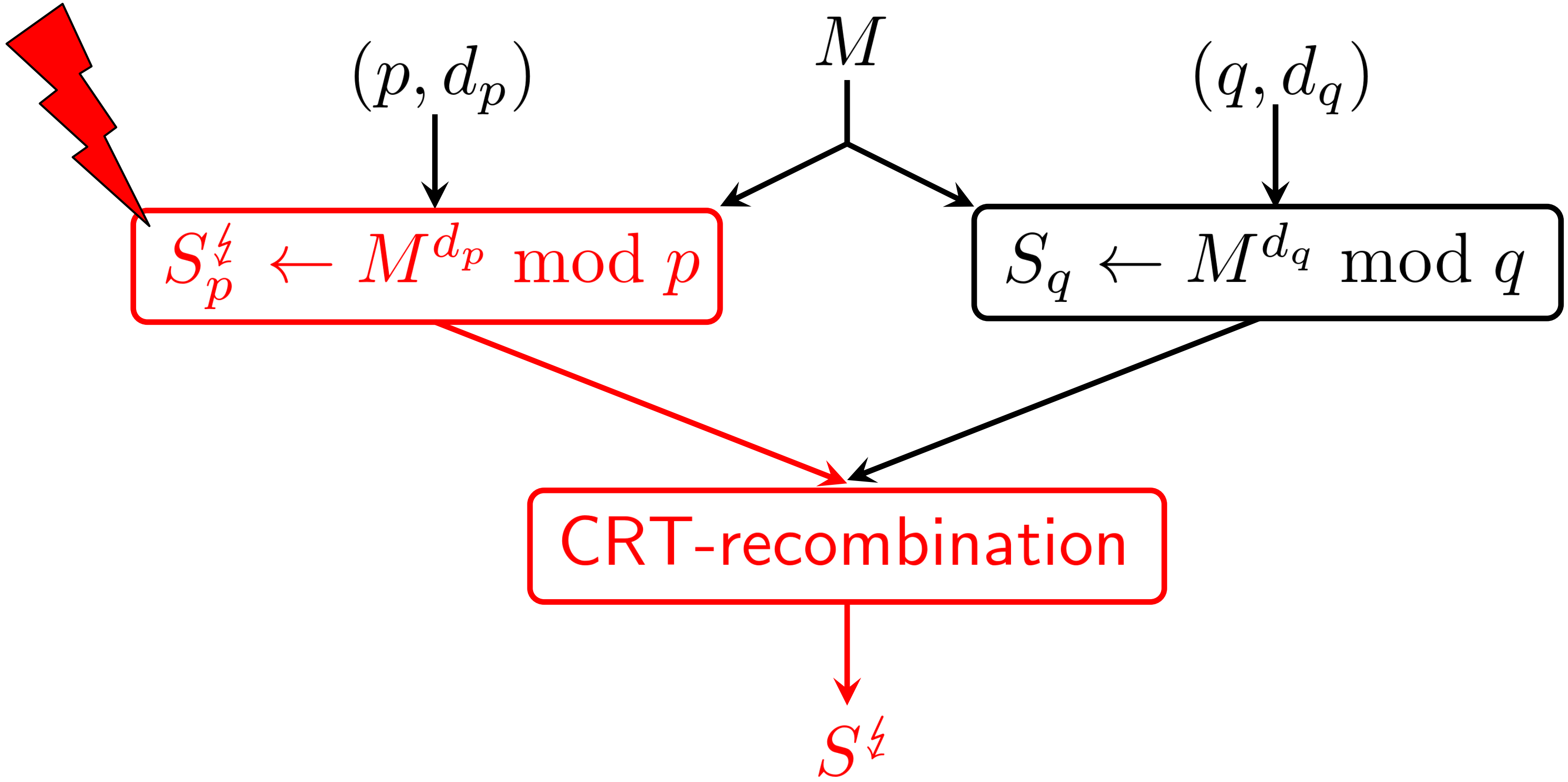


$$\begin{cases} S \equiv S_p \bmod p \\ S \equiv S_q \bmod q \end{cases}$$

OBERTHUR TECHNOLOGIES

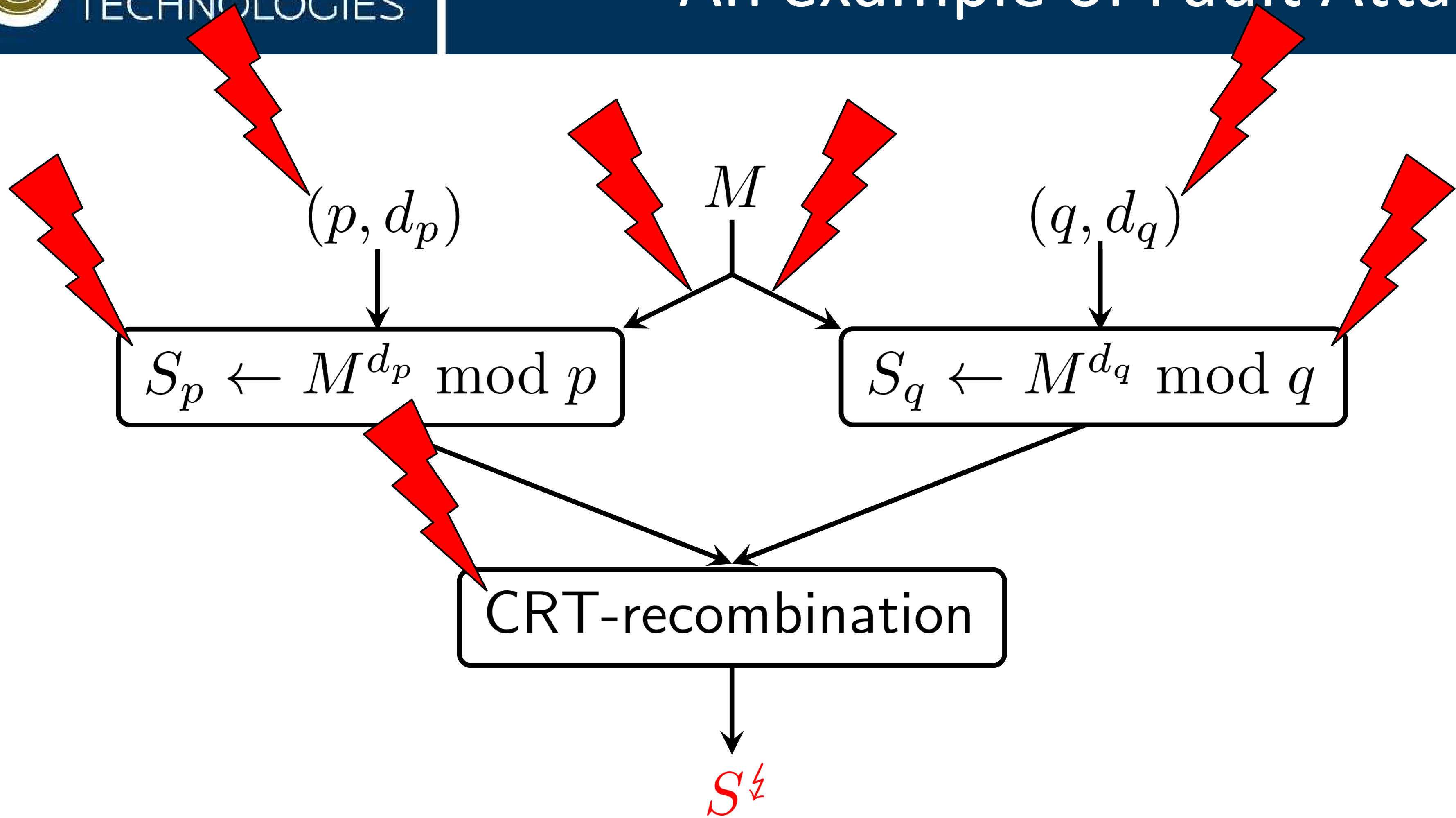
An example of Fault Attack

- Instead of computing $S = M^d \bmod N$



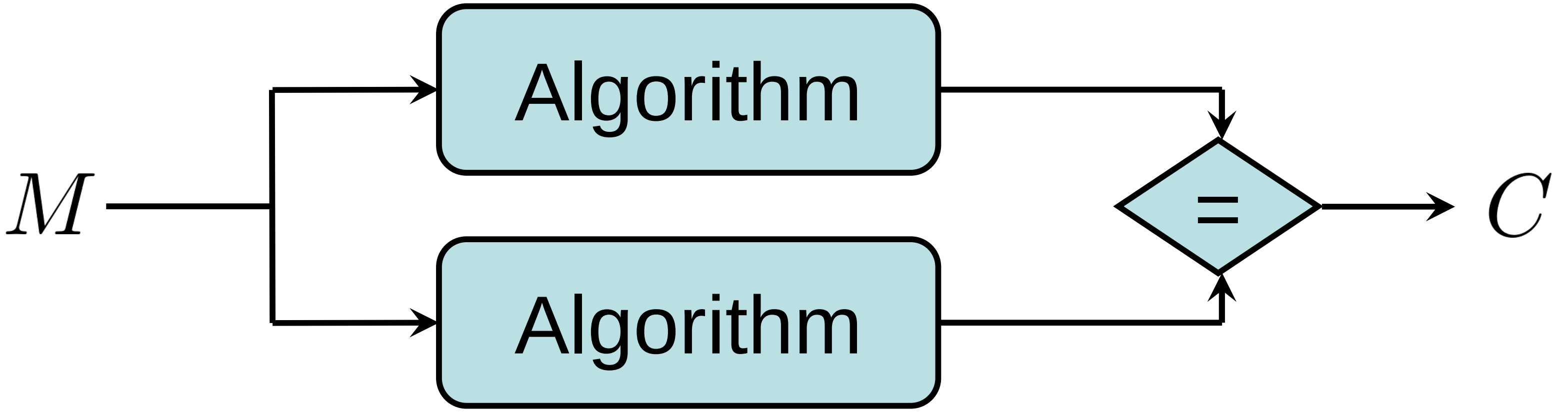
$$\begin{cases} S \equiv S_p \bmod p \\ S \equiv S_q \bmod q \end{cases} \begin{cases} S^{\lightning} \not\equiv S_p \bmod p \\ S^{\lightning} \equiv S_q \bmod q \end{cases} \Longrightarrow \text{gcd}(S - S^{\lightning}, N) = q$$

An example of Fault Attack



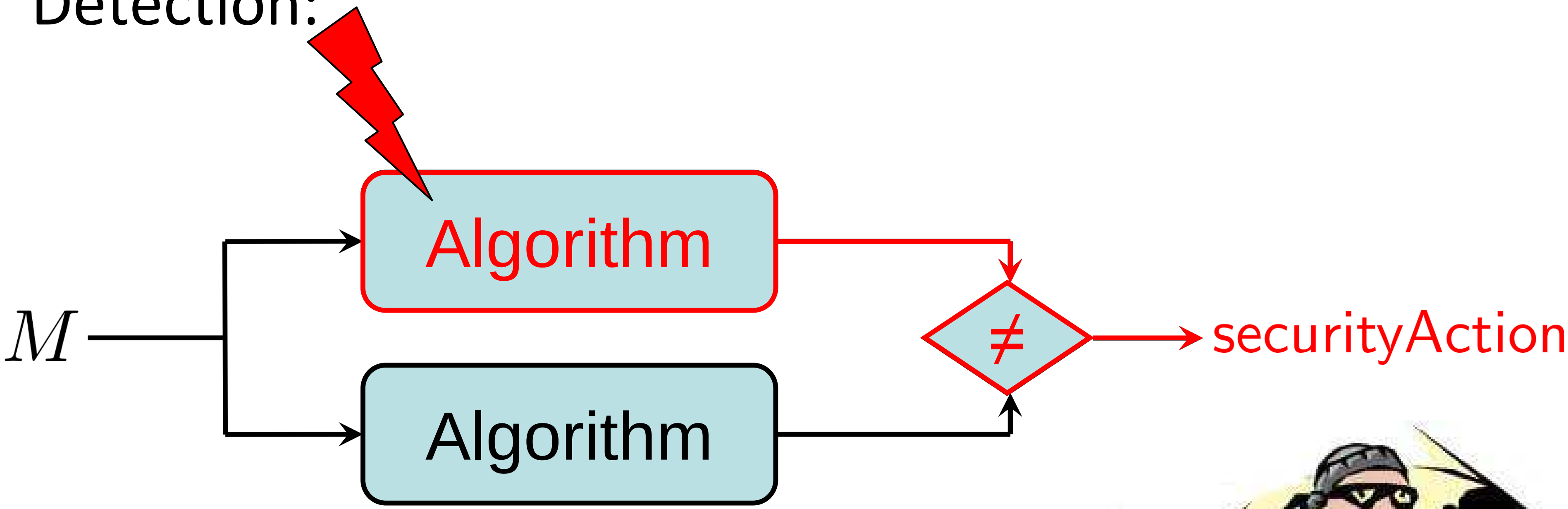
- What a challenge for the countermeasure!!!

- Detection:



Countermeasures

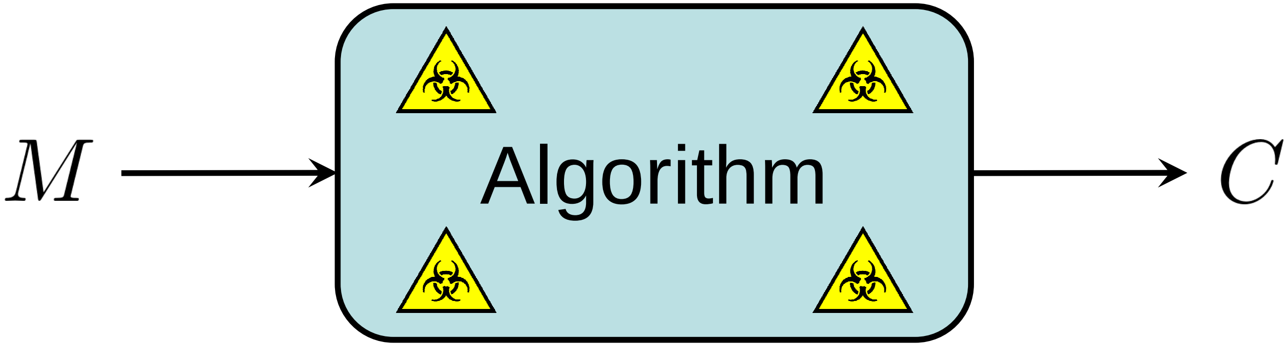
- Detection:



- Drawbacks:
  - Attacks during comparison
  - Different paths to manage

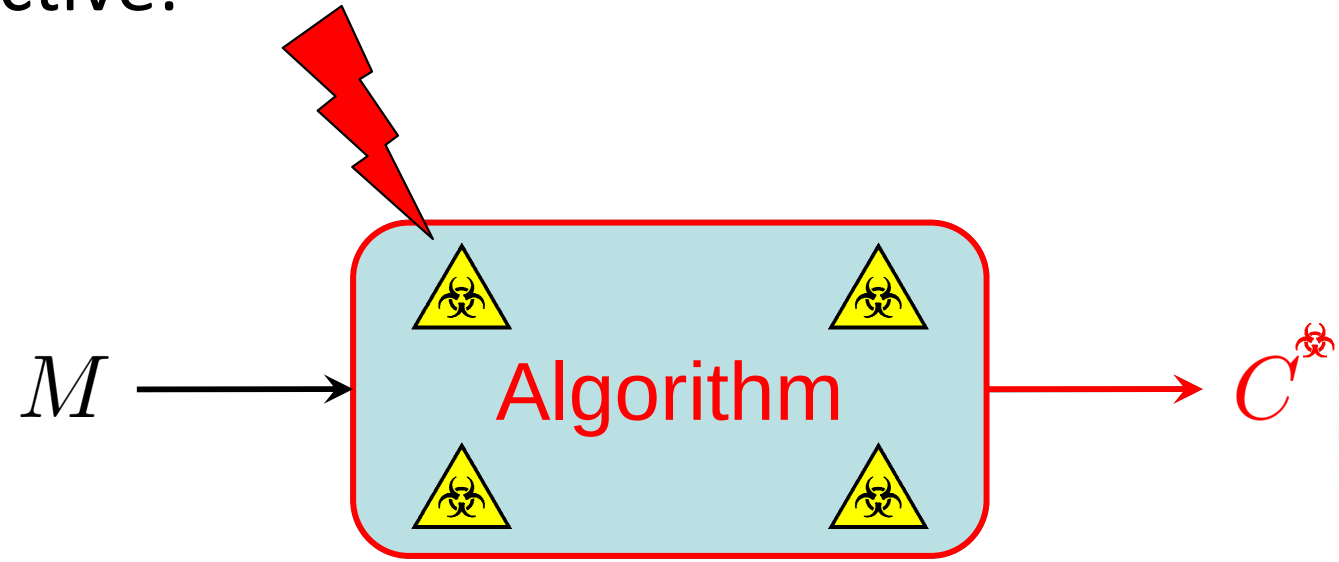
Countermeasures

- Infective:



Countermeasures

- Infective:



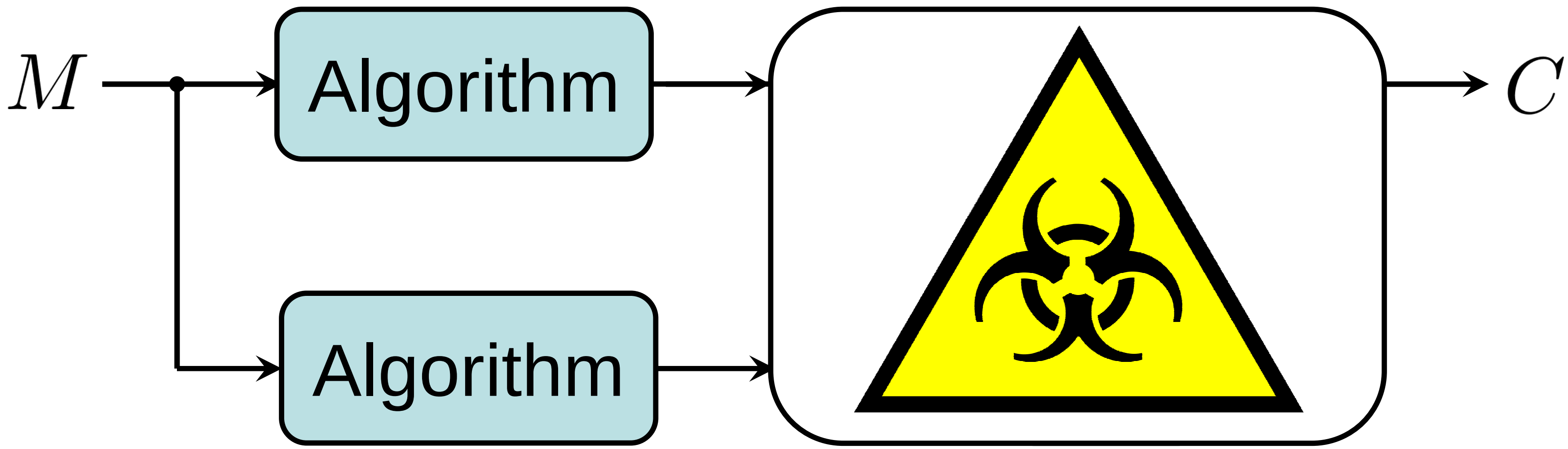
- Comparison with Detection:
  - \+ No comparison
  - \+ Single path
  - – Could be much slower

Infective Countermeasures History

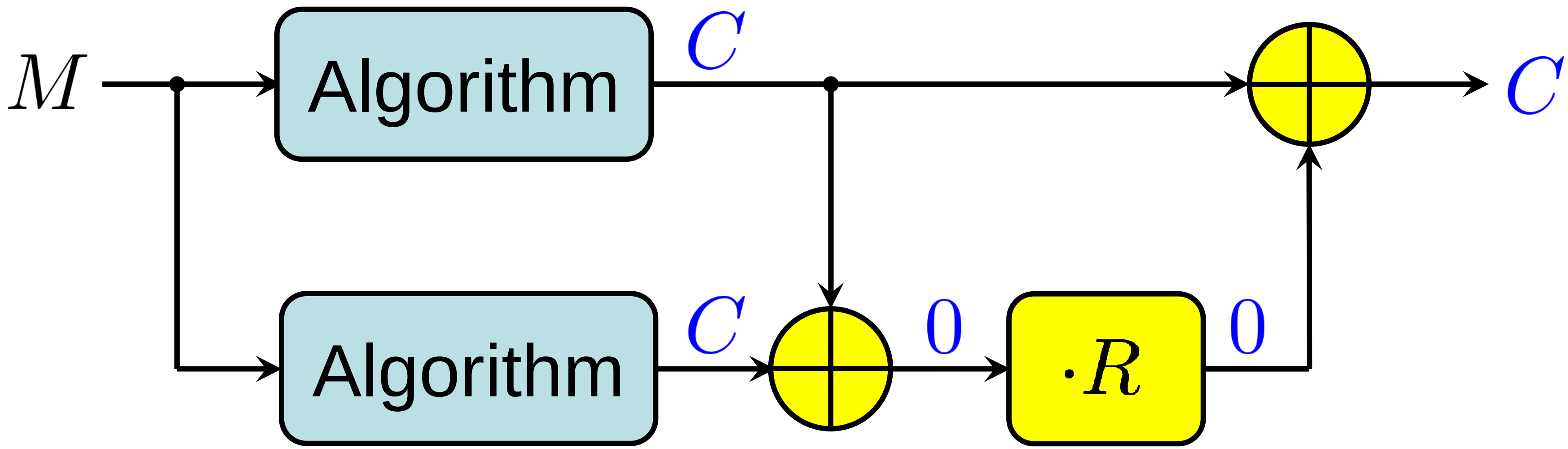
- Asymmetric:
  - [Yen, Kim, Lim, Moon] 2001 → [Yen, Kim, Moon] 2004
  - [Blömer, Otto, Seifert] 2003 → [Qin, Li, Kong] 2008
  - [Ciet, Joye] 2005 → [Berzati, Canovas, Goubin] 2008
  - [Schmidt et al.] 2010 → [Feix, Venelli] 2013
- Symmetric:
  - [Lomné, Roche, Thillard] 2012
  - [Gierlichs, Schmidt, Tunstall] 2012

- Introduction
- Attacks
  - FDTC 2012 Countermeasure
  - LatinCrypt 2012 Countermeasure
- Conclusion

FDTC 2012 Countermeasure

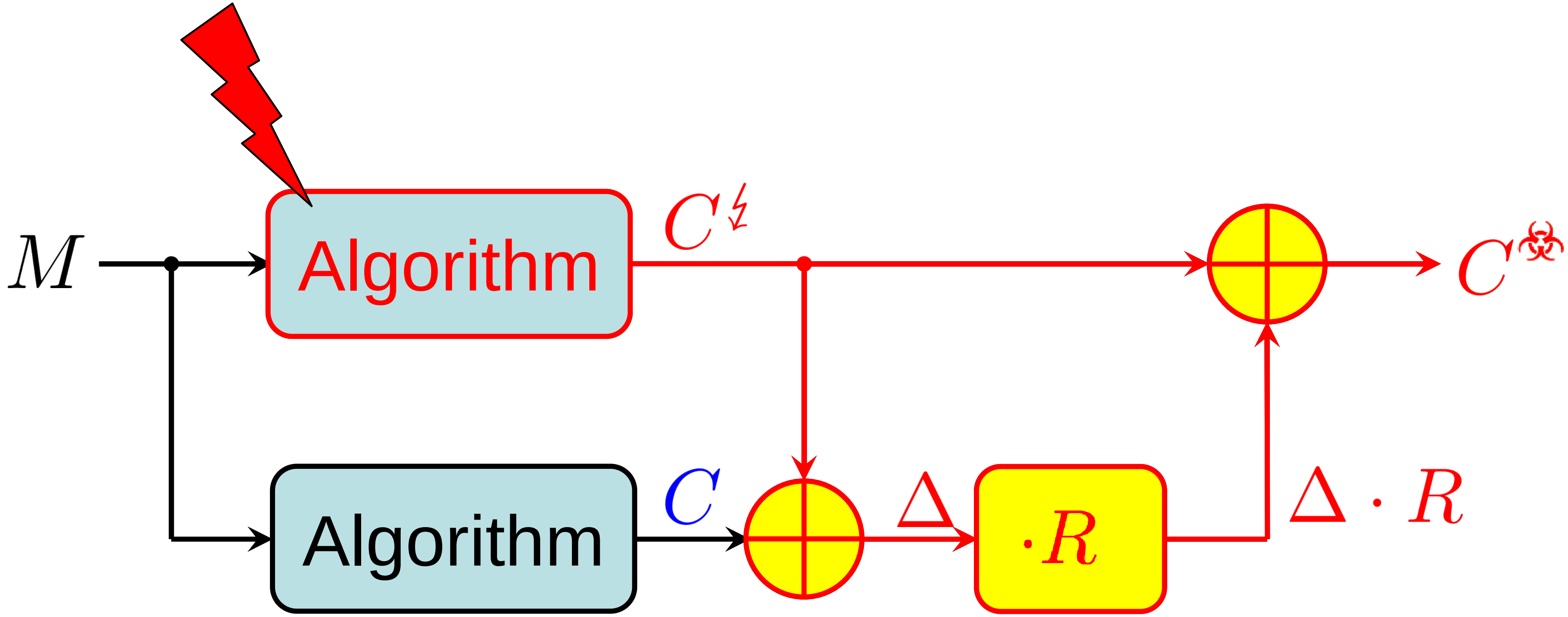


FDTC 2012 Countermeasure



- For efficiency, multiplication is performed byte per byte
- Restriction on the multiplicative mask:
  - R_i must be different from 0 and 1

FDTC 2012 Countermeasure



- For efficiency, multiplication is performed byte per byte
- Restriction on the multiplicative mask:
  - R_i must be different from 0 and 1

FDTC 2012 CM Analysis

- AfricaCrypt 2009 : Mukhopadhyay shows that:

  $$(C, C^{\lightning})$$ gives the AES-128 key

  if a byte-fault has disturbed the 8^{th} round.

$\Rightarrow$ Goal for the attacker: Recover $C^{\lightning}$ from $C^{\biohazard}$:

$$C_i^{\biohazard} = C_i^{\lightning} \oplus \Delta_i \cdot R_i$$

where $\Delta_i = C_i \oplus C_i^{\lightning}$ and R_i a random value $\neq \{0, 1\}$.

- Let us assume a constant fault model (i.e. Δ cst):

$$R_i = 2 \qquad C_i^{\biohazard} = C_i^{\lightning} \oplus 2 \cdot \Delta_i$$
$$R_i = 3 \qquad C_i^{\biohazard} = C_i^{\lightning} \oplus 3 \cdot \Delta_i$$
$$\cdots$$
$$R_i = 255 \qquad C_i^{\biohazard} = C_i^{\lightning} \oplus 255 \cdot \Delta_i$$

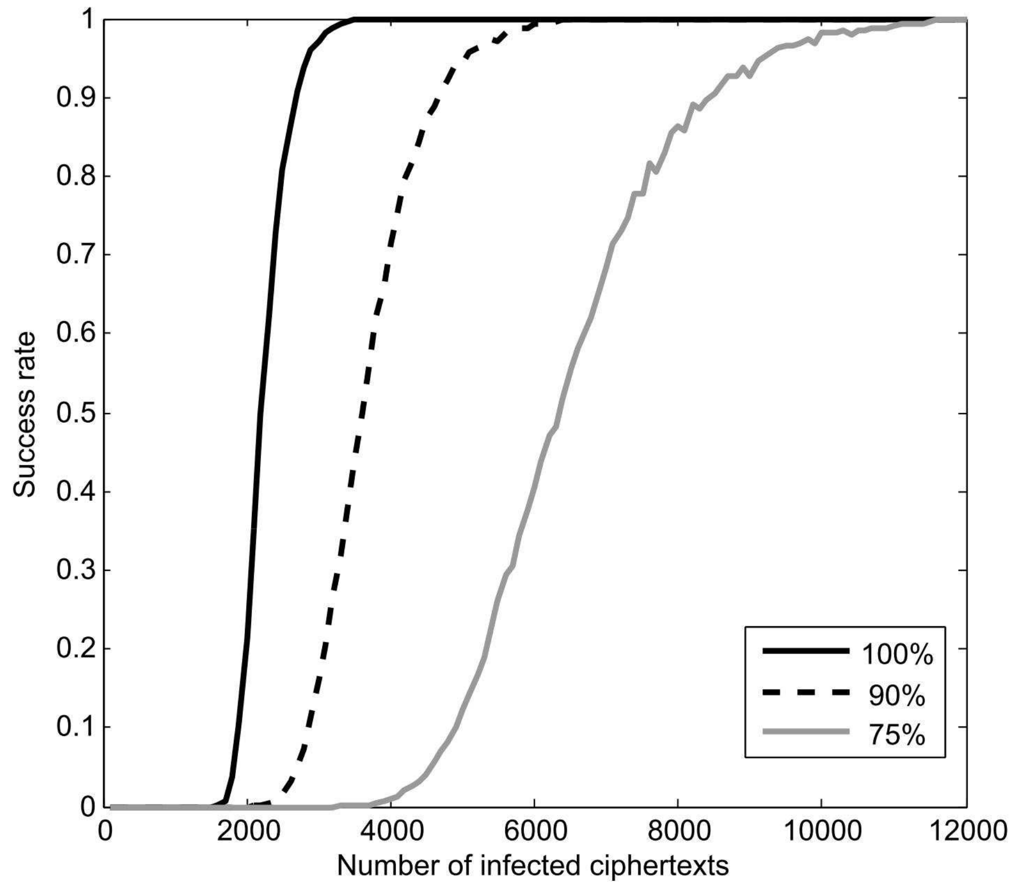
$\Rightarrow$ 2 values never appear : $C_i^{\lightning}$ and $C_i^{\lightning} \oplus \Delta_i = C_i$

OBERTHUR TECHNOLOGIES

FDTC 2012 CM Analysis

- Attack procedure:
  1. Inject a constant byte error during round 8 to obtain $C^{\biohazard}$
  2. For each byte i, remove $C_i^{\biohazard}$ from the list of possible values for $C_i^{\lightning}$
  3. If one $C_i^{\lightning}$ has more than 2 possible values, then go back to Step 1
  4. Identify each $C_i^{\lightning}$ since C_i's are known
  5. Apply Mukhopadhyay's attack to $(C, C^{\lightning})$ to recover the secret key

Simulations

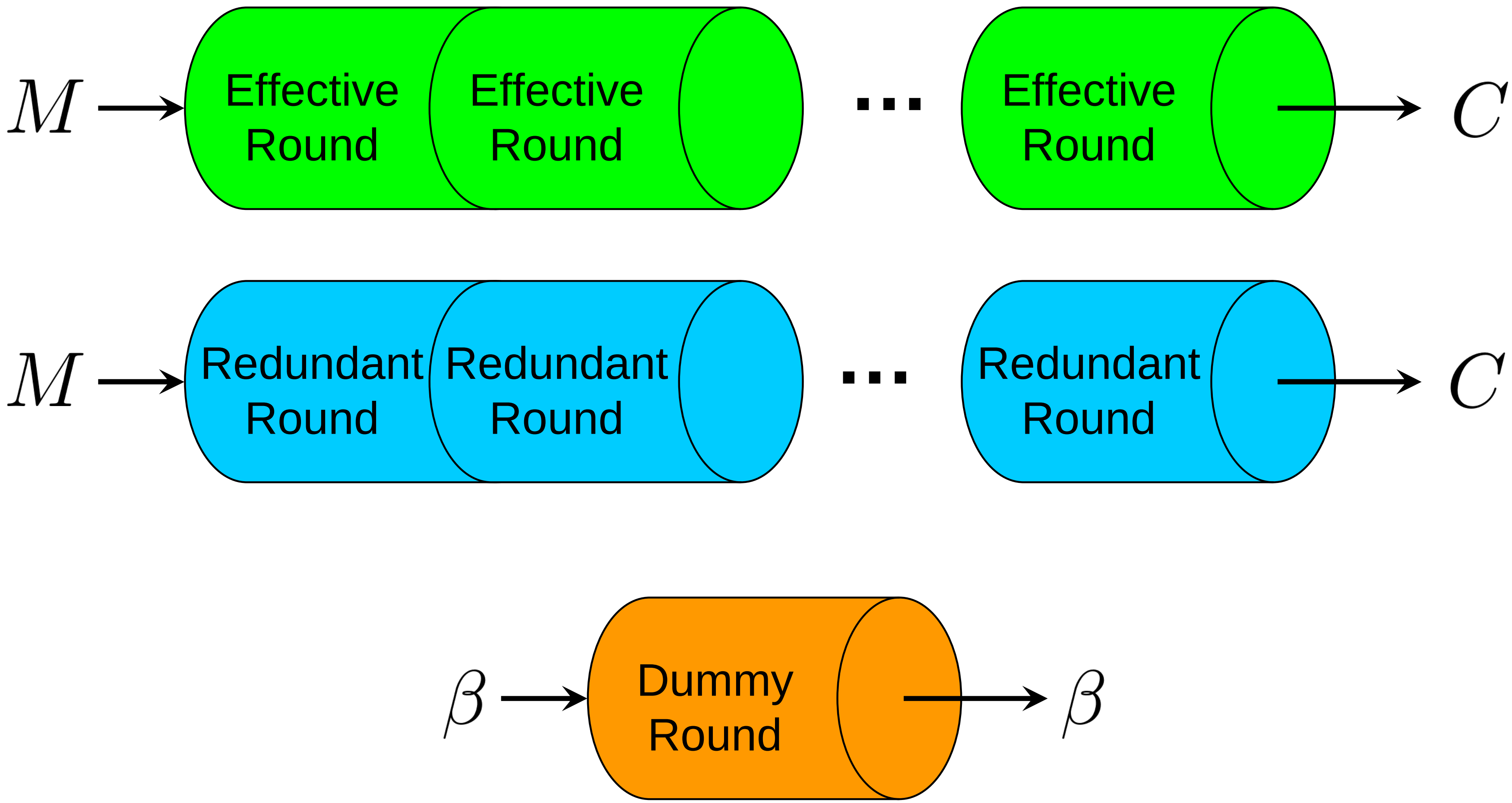


- With $3\,000$ $C^{☣}$'s, the AES key is recovered with 99% success rate

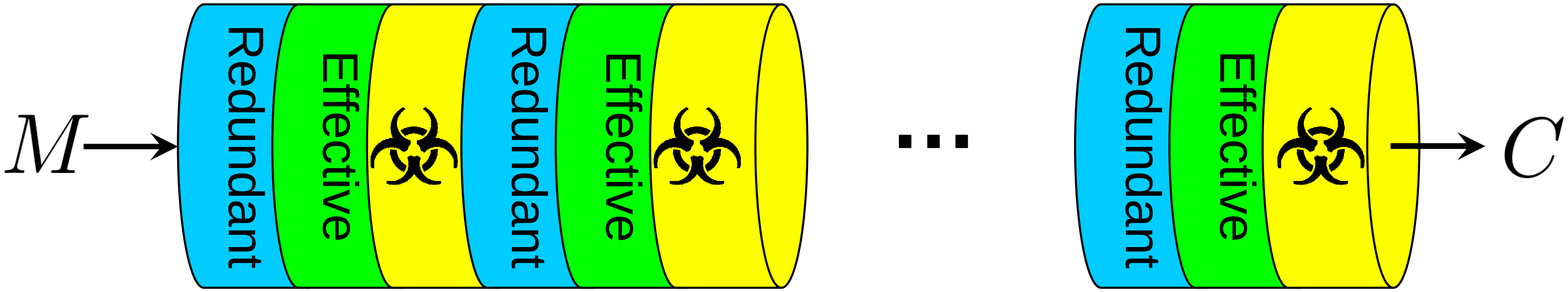
Overview

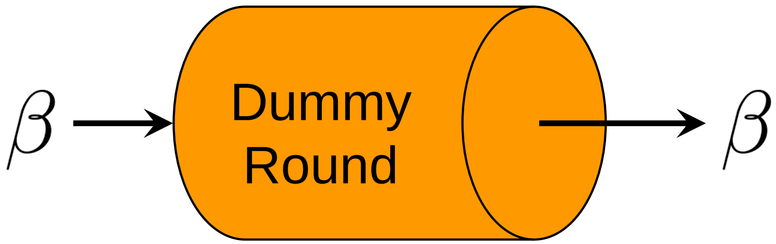
- Introduction
- Attacks
  - FDTC 2012 Countermeasure
  - LatinCrypt 2012 Countermeasure
- Conclusion

LatinCrypt 2012 Countermeasure

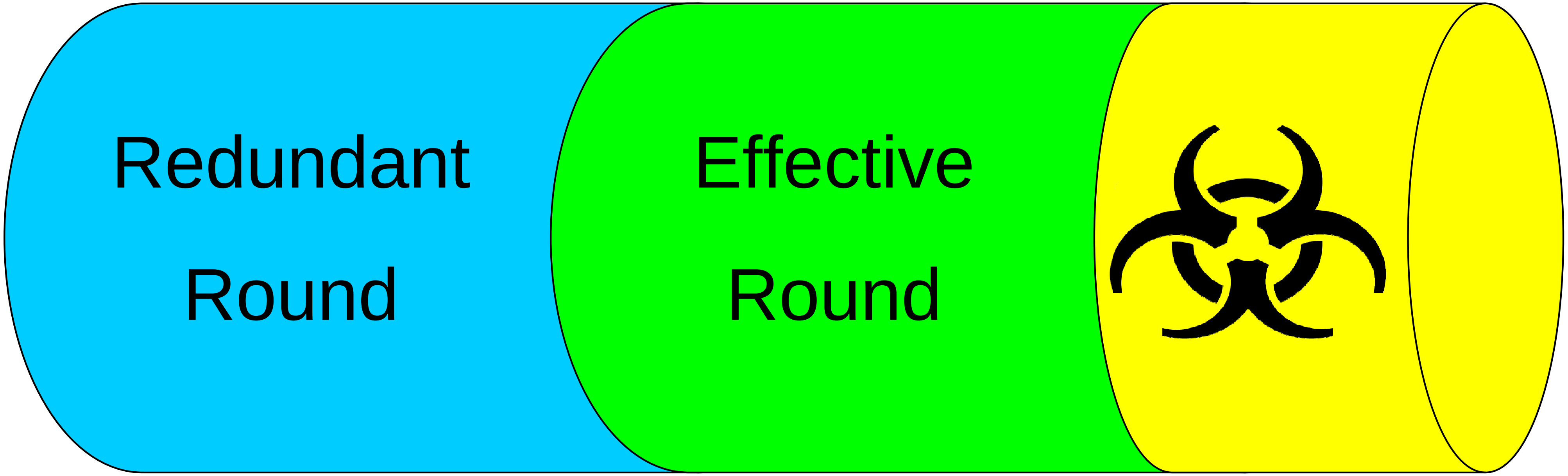


LatinCrypt 2012 Countermeasure

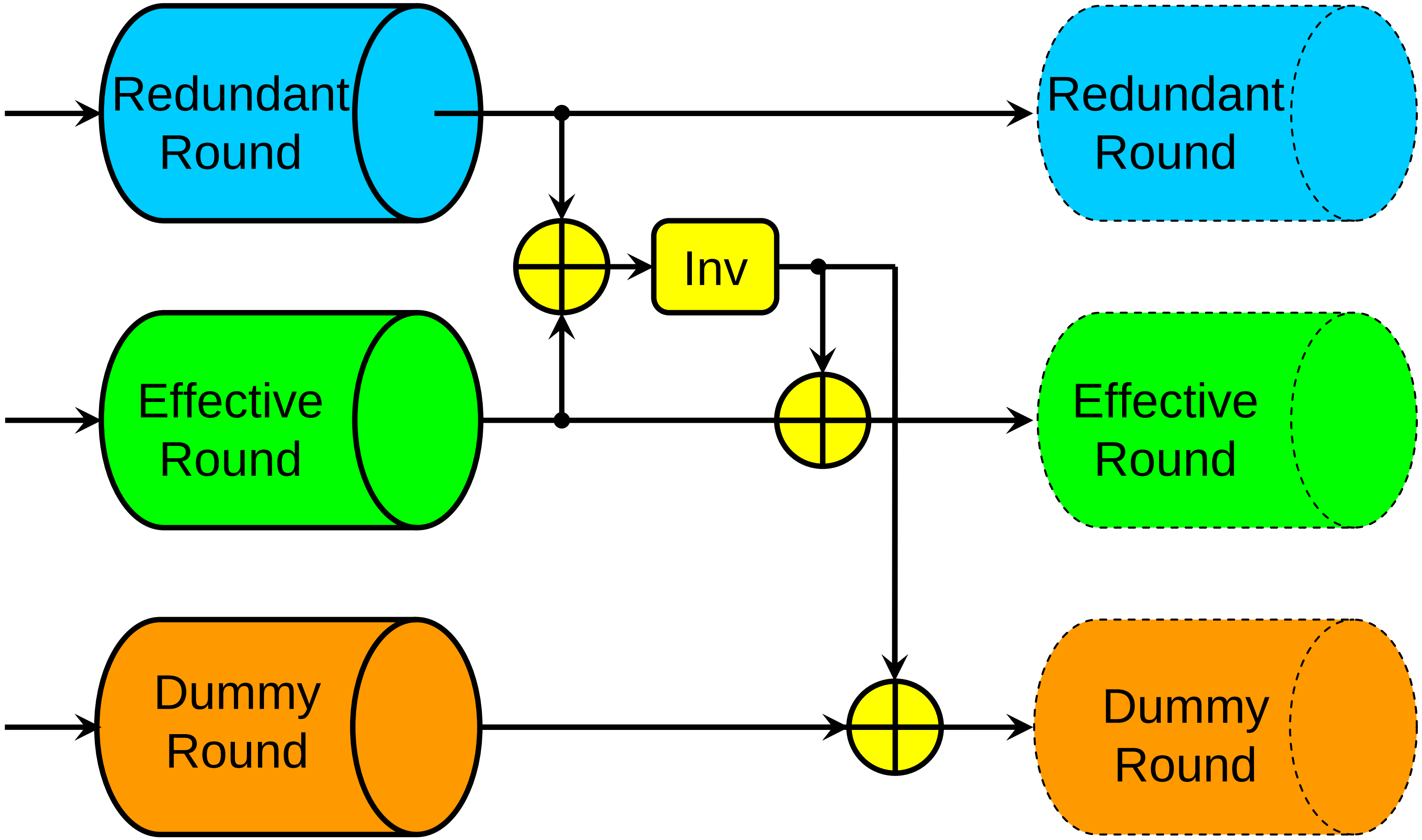




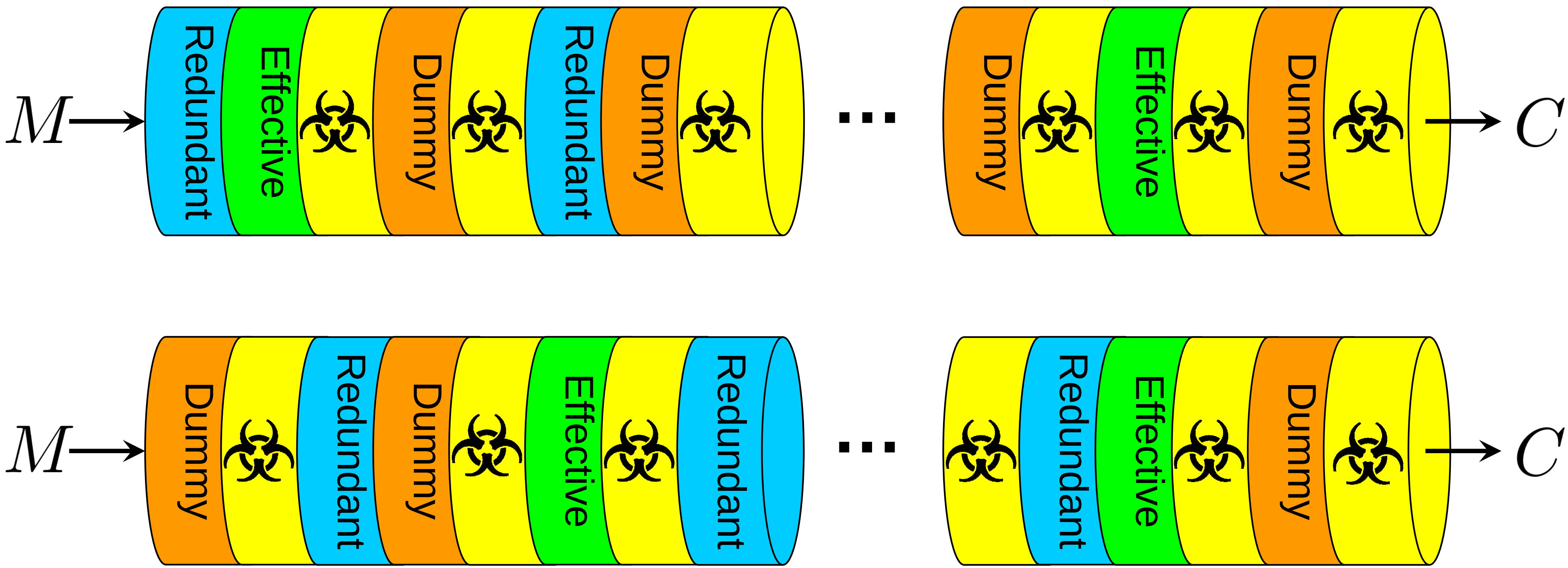
First Infective Mechanism



First Infective Mechanism

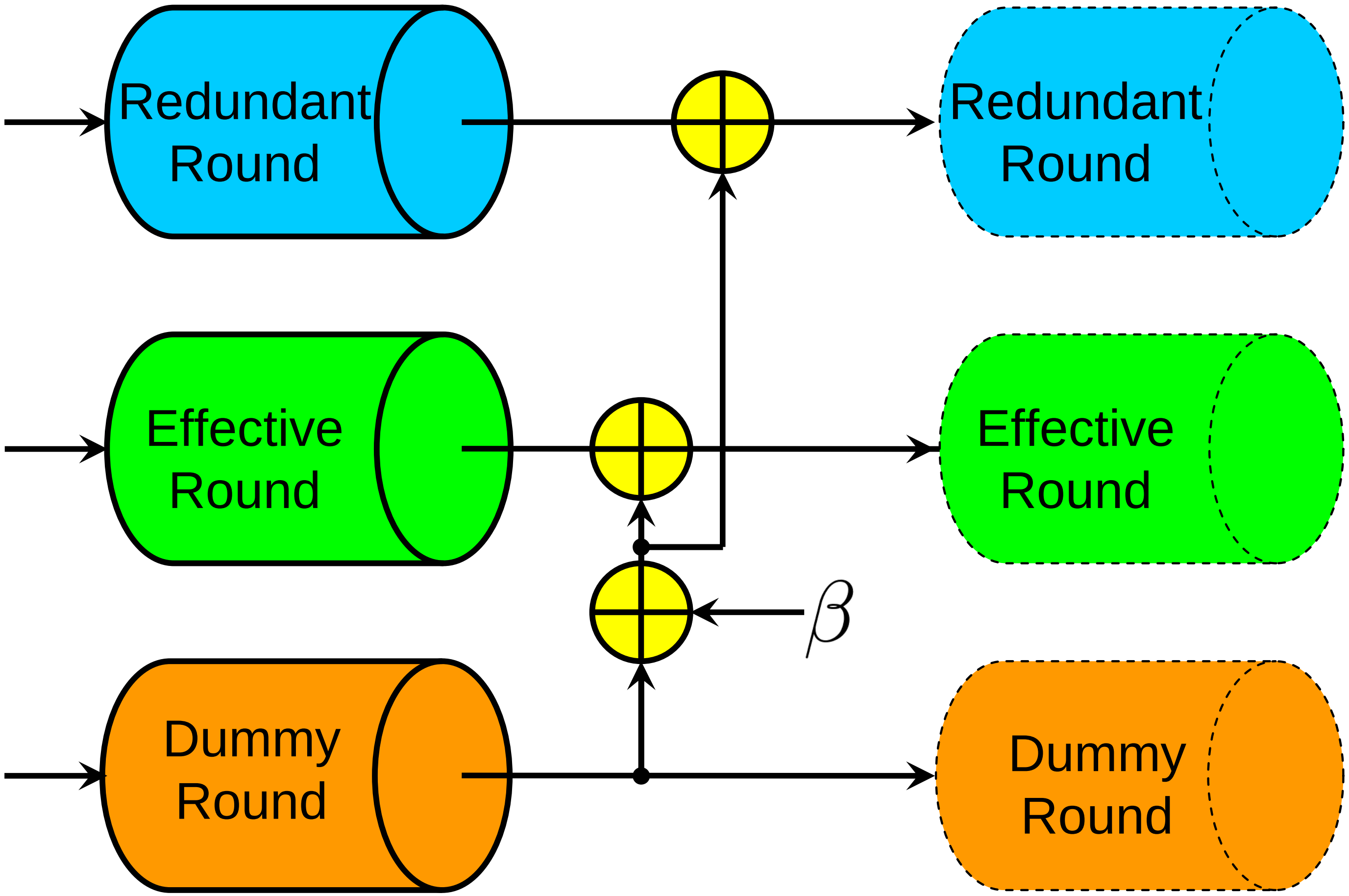


On the Use of Dummy Rounds

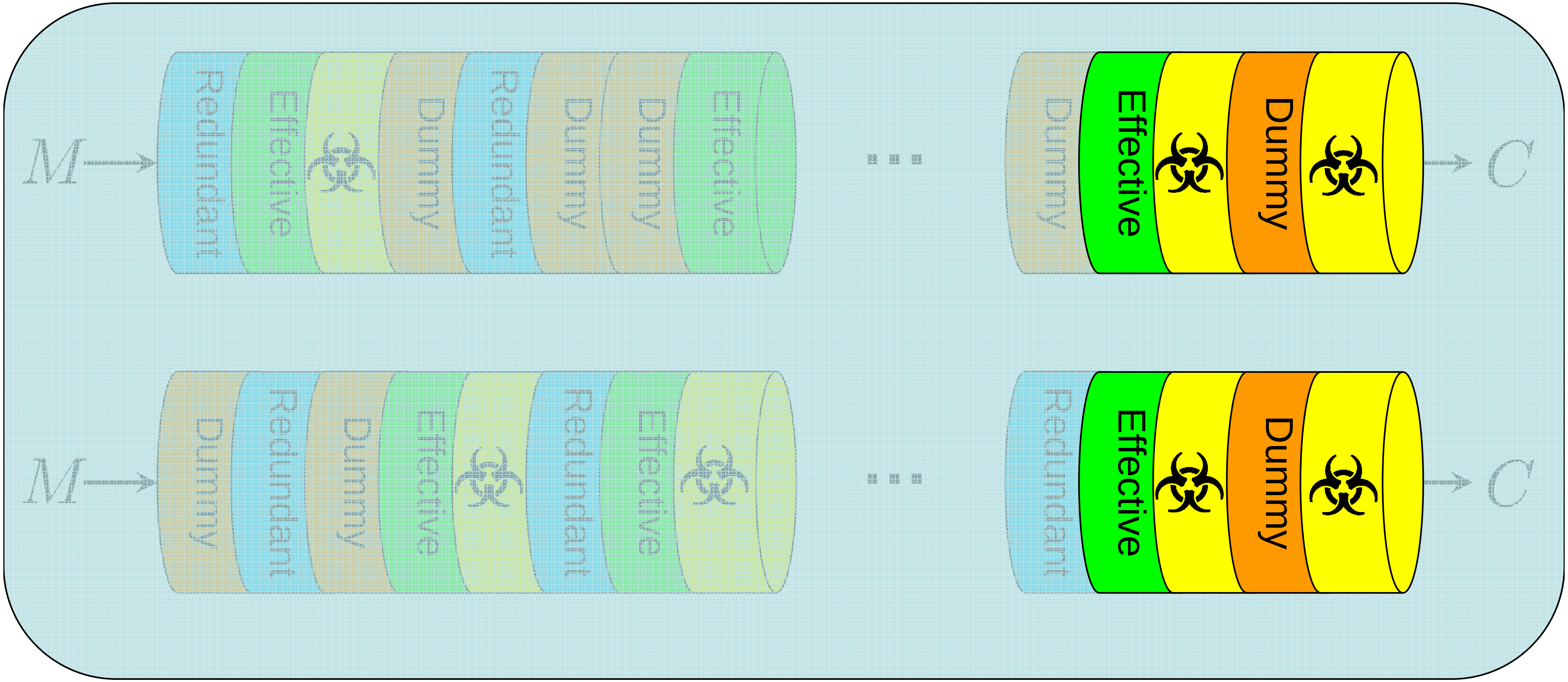


Second Infective Mechanism

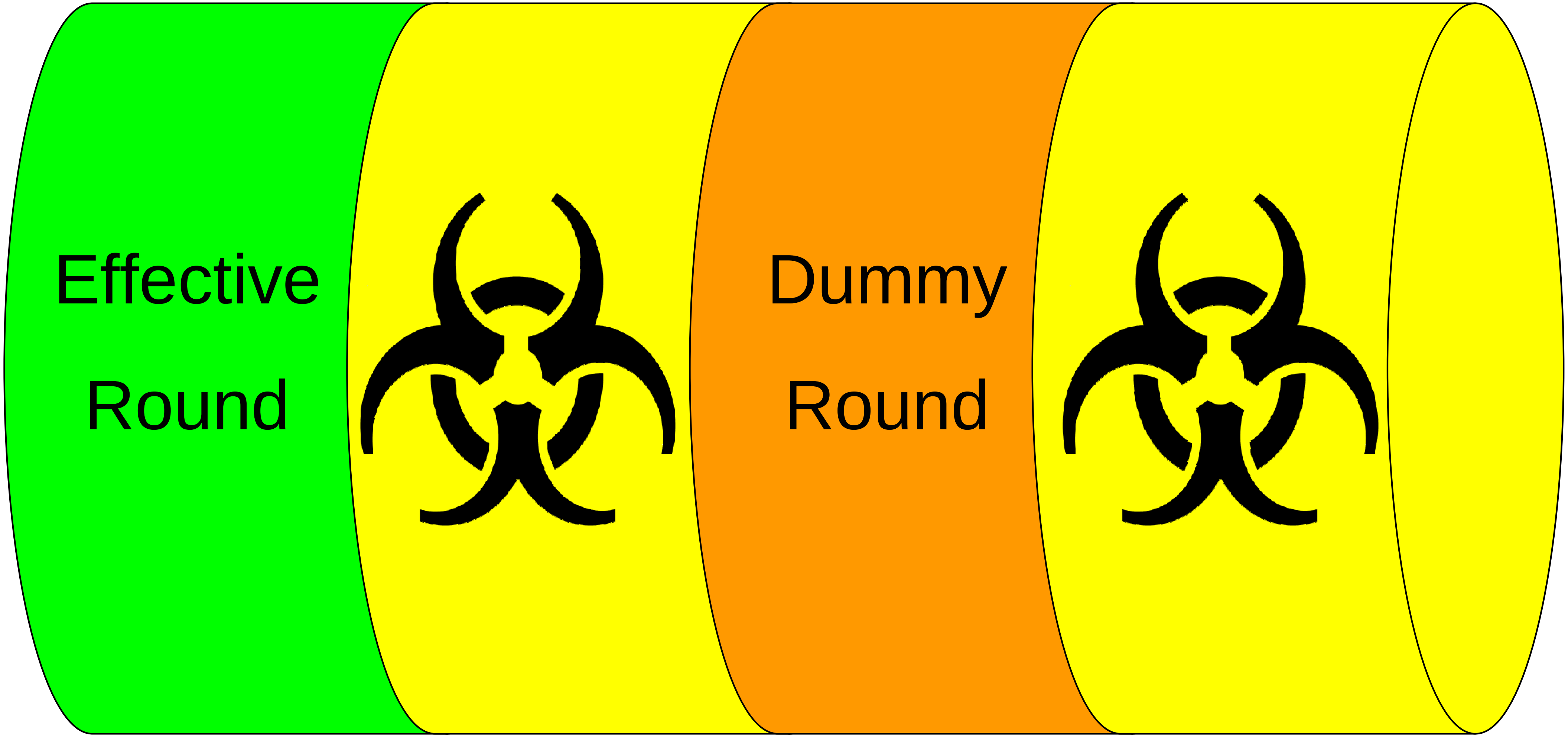
Second Infective Mechanism



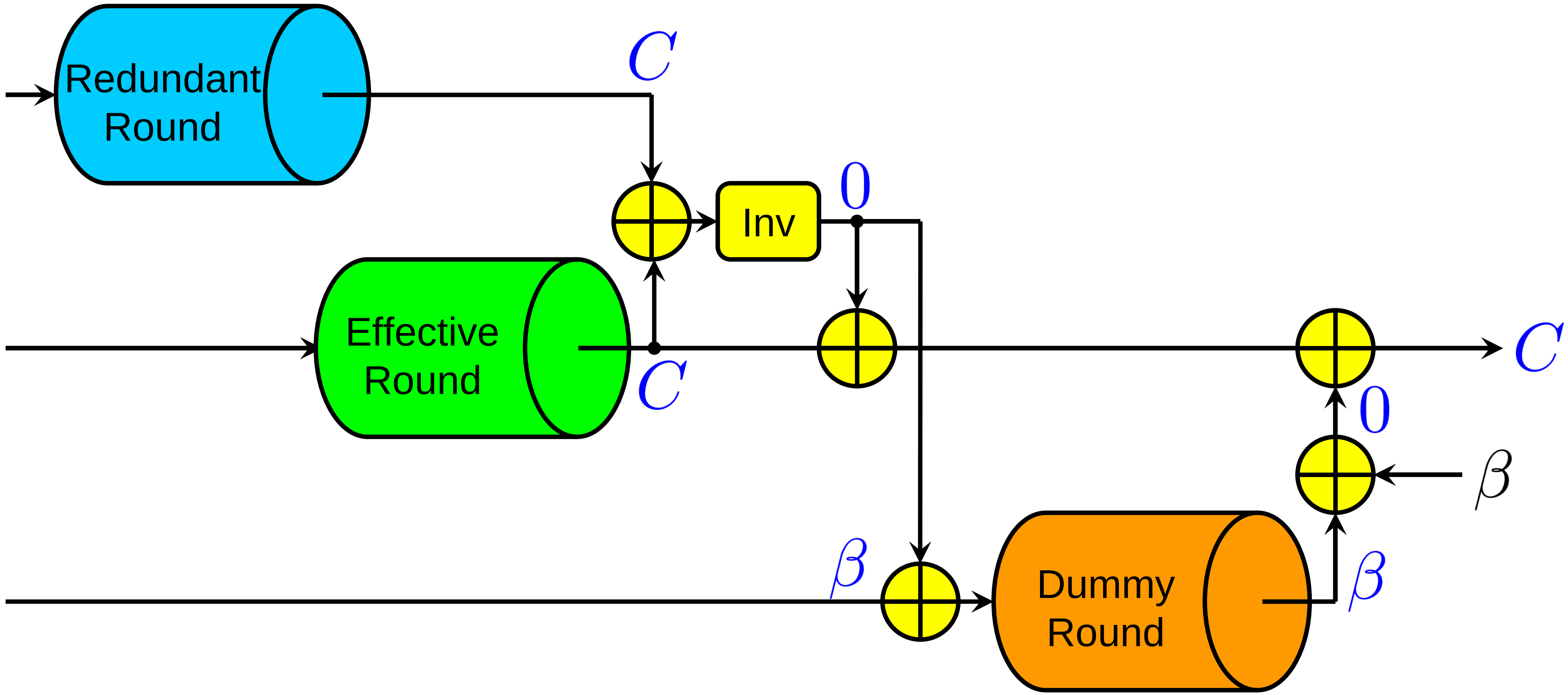
A Useful Remark



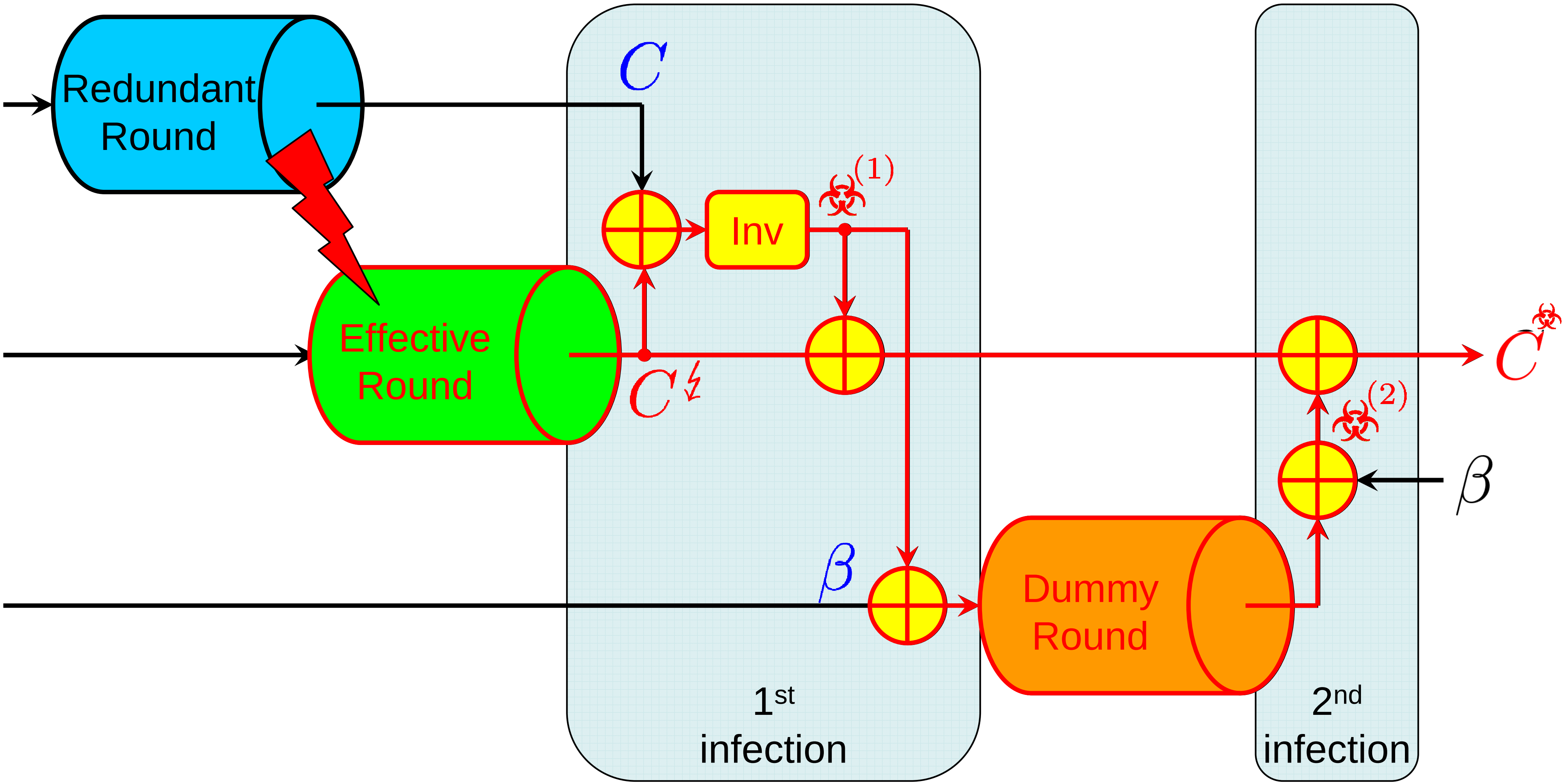
A Useful Remark



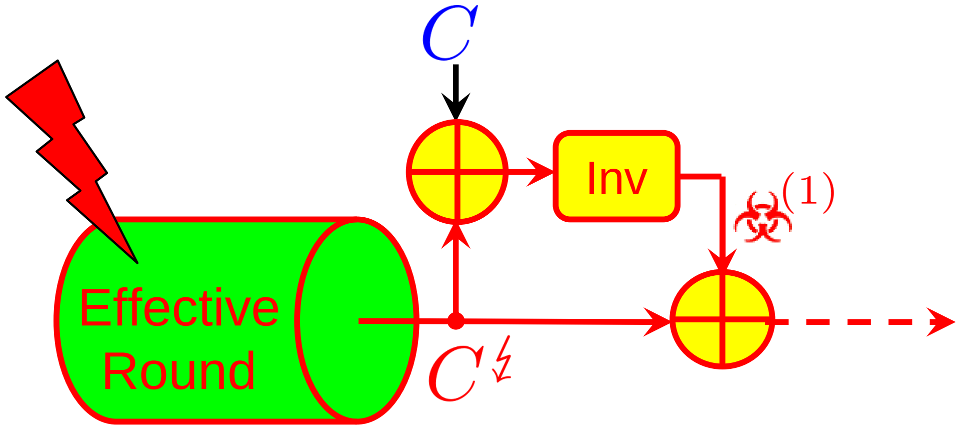
A Useful Remark



A Useful Remark



First Infection Analysis



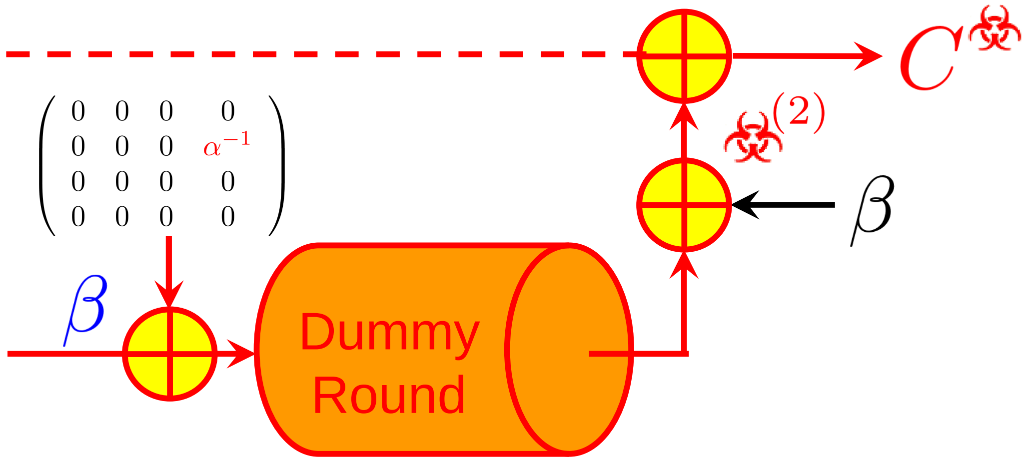
- If disturbance of a byte of the input, the differential is:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ e & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \boxed{\text{SubBytes}} \rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \boxed{\text{ShiftRows}} \rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- So the first infection is equal to:

$$☣^{(1)} = \mathrm{Inv}(C \oplus C^{↯}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Second Infection Analysis



$$\text{Round}\left(\beta \oplus \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right) \oplus \beta = \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & 0 \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix}$$

$$\text{☣}^{(2)} = \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & 0 \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix}$$

First + Second Infection

- The infected output is defined by:

$$C^{\biohazard} = C^{\lightning} \oplus \biohazard^{(1)} \oplus \biohazard^{(2)}$$

- Therefore, we have:

$$C^{\biohazard} = C^{\lightning} \oplus \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & 0 \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix}$$

which is equivalent to:

$$C^{\biohazard} = C^{\lightning} \oplus \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & \alpha^{-1} \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix}$$

Analysis

- By using:

$$C^{☣} = C^{\lightning} \oplus \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & \alpha^{-1} \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix} \quad \text{and} \quad C \oplus C^{\lightning} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

we obtain :

$$C \oplus C^{☣} = \begin{pmatrix} 0 & 0 & \delta_0 & 0 \\ 0 & 0 & \delta_1 & \alpha \oplus \alpha^{-1} \\ 0 & 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 & 0 \end{pmatrix}$$

- The byte α contains information on the key but:
  - $☣^{(1)}$ does not efficiently blind this value
  - $☣^{(2)}$ has no effect due to ShiftRows transformation

Attack Procedure

- To sum up, we have:

$$C_{13} \oplus C_{13}^{\text{☣}} = \alpha \oplus \alpha^{-1}$$

  with

$$\alpha = \mathsf{SB}(s \oplus e) \oplus \mathsf{SB}(s)$$

  where s is the second input byte of the last effective round.

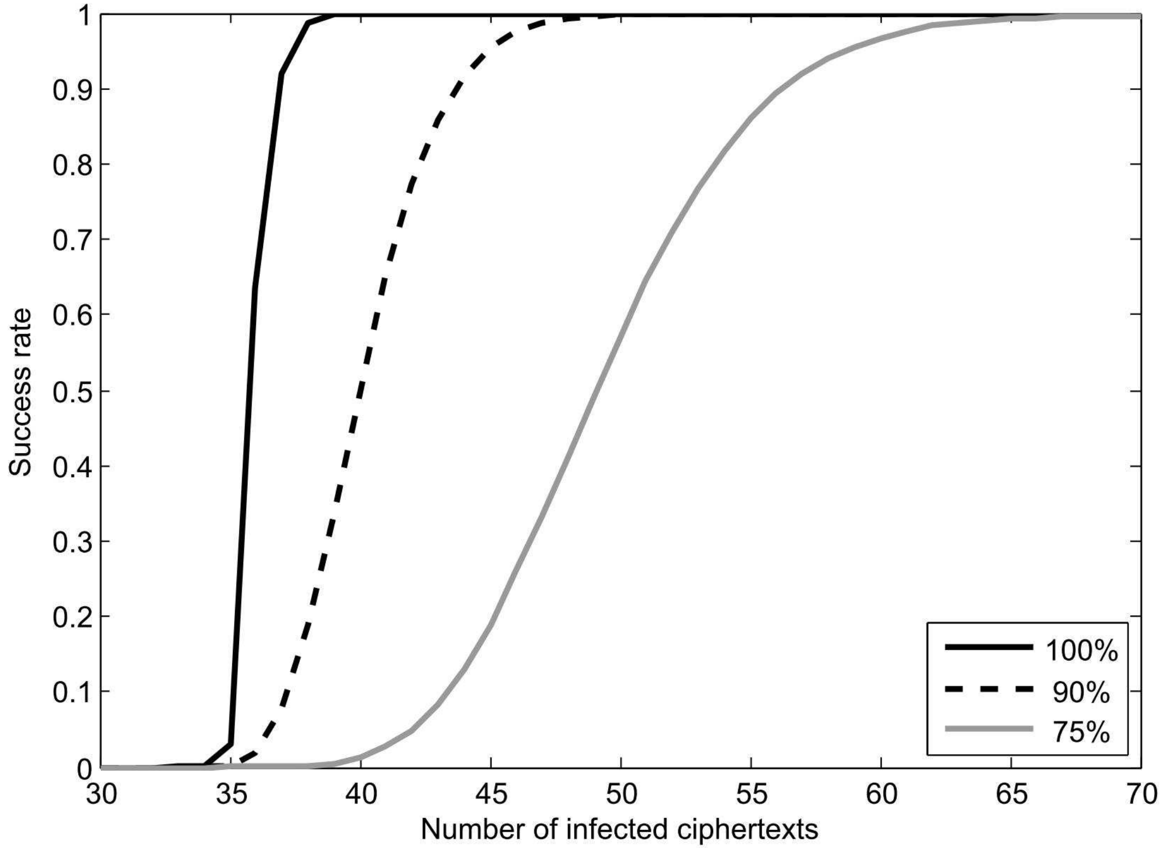
- The byte s can thus be expressed as:

$$s = \mathsf{SB}^{-1}(C_{13} \oplus k_{13})$$

- The attack process is thus the following:

  1. Guess the corresponding key byte $k_h \in \{0, \cdots, 255\}$
  2. Compute $s_h = \mathsf{SB}^{-1}(C_{13} \oplus k_h)$
  3. Guess the error value $e_h \in \{1, \cdots, 255\}$
  4. Compute $\alpha_h = \mathsf{SB}(s_h \oplus e_h) \oplus \mathsf{SB}(s_h)$
  5. If $C_{13} \oplus C_{13}^{\text{☣}} \neq \alpha_h \oplus \alpha_h^{-1}$ then discard (k_h, e_h)

Simulations



- With 37 $C^{☣}$'s, the last three rows of the AES key are recovered with 99% success rate

Overview

- Introduction
- Attacks
  - FDTC 2012 Countermeasure
  - LatinCrypt 2012 Countermeasure
- Conclusion

Conclusion

- The two existing symmetric infective countermeasures are flawed
- Easy to patch but a framework is missing to formally prove countermeasures' security
- After 10 years of research in infective countermeasures, no original proposal has survived…

  Do infective countermeasures have a future?

Any Questions?☣