A High-Order Infective Countermeasure Framework

G. Barbu, L. Bettale, L. Castelnovi, T. Chabrier,
N. Debande, C. Giraud and N. Reboud

IDEMIA

September 17, 2021

Context

Faults are a serious threat in cryptographic implementations.

Attacker's goal: getting an erroneous output that leaks the secret key.

Dealing with **block ciphers**, two strategies in state of the art to avoid it:

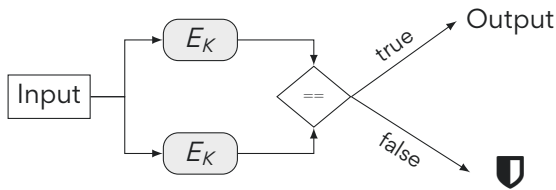
- Detection
- Infection

Detection

Principle

The algorithm is run twice and the outputs are compared.

If different, an appropriate measure is taken (for instance, no output).

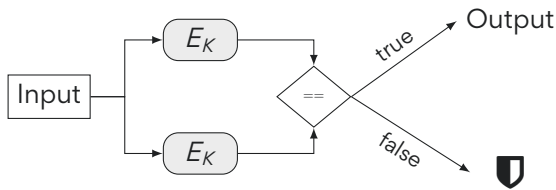


Detection

Principle

The algorithm is run twice and the outputs are compared.

If different, an appropriate measure is taken (for instance, no output).



But...

Comparison can be corrupted by an extra fault.

Infection

Principle

The algorithm's output is corrupted by an amplified error.

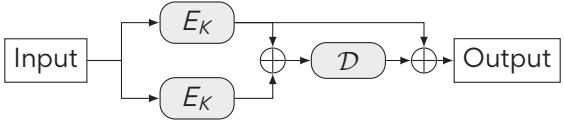
No need for comparison and non-informative output.

But...

How to amplify the error in practice?

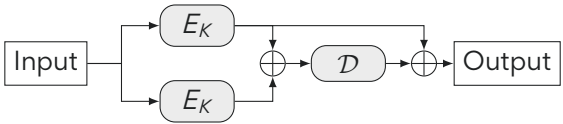
Infection today

- External infection

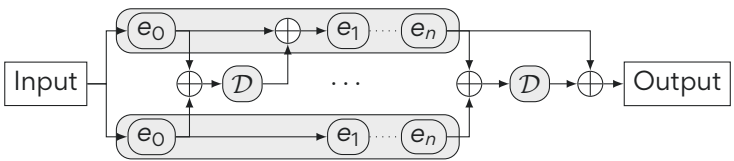


Infection today

- External infection

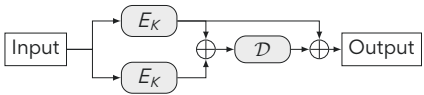


- Internal infection

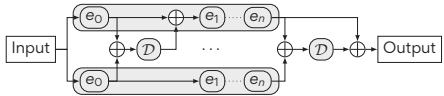


Infection today

- External infection



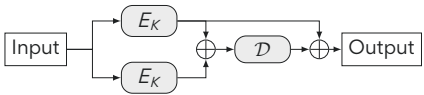
- Internal infection



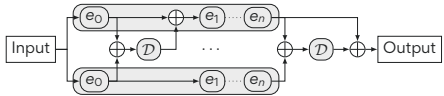
Almost all propositions up to now are broken.

Infection today

- External infection



- Internal infection



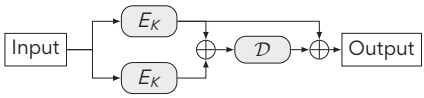
Almost all propositions up to now are broken.

Either because of:

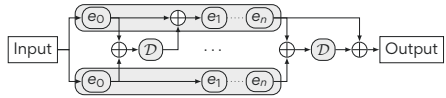
- A deterministic $\mathcal{D}$,
- Or an invertible $\mathcal{D}$,
- Or a low-diffusion $\mathcal{D}$.

Infection today

- External infection



- Internal infection

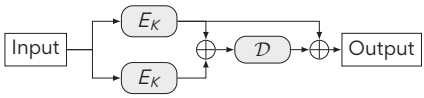


In a secure scheme, $\mathcal{D}$ should be:

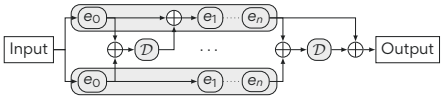
- ✓ *Non-deterministic*
- ✓ And *non-invertible*
- ✓ And with *high-diffusion* capacity

Infection today

- External infection



- Internal infection



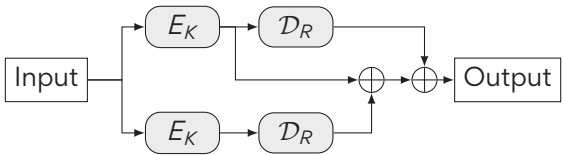
In a secure scheme, $\mathcal{D}$ should be:

- ✓ *Non-deterministic*
- ✓ And *non-invertible*
- ✓ And with *high-diffusion* capacity

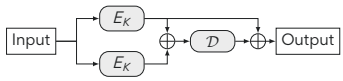
Hard to find such $\mathcal{D}$ with the constraint $\mathcal{D}(0) = 0$

Principle of our framework

External infection but the infective value is $\Delta\mathcal{D}(E_K)$ instead of $\mathcal{D}(\Delta E_K)$



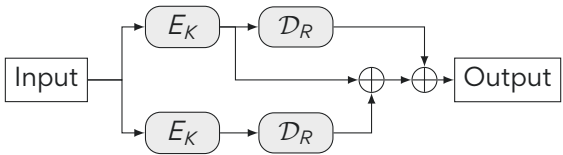
Our proposal



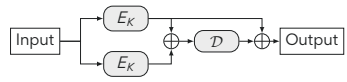
SoA external infection

Principle of our framework

External infection but the infective value is $\Delta\mathcal{D}(E_K)$ instead of $\mathcal{D}(\Delta E_K)$



Our proposal

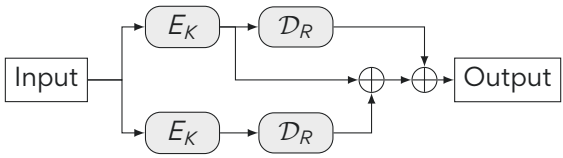


SoA external infection

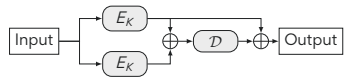
- Constraint $\mathcal{D}(0) = 0$ removed
  $\Longrightarrow$ $\mathcal{D}$ can be a hash function: *non-invertibility* and *high diffusion* achieved

Principle of our framework

External infection but the infective value is $\Delta\mathcal{D}(E_K)$ instead of $\mathcal{D}(\Delta E_K)$



Our proposal

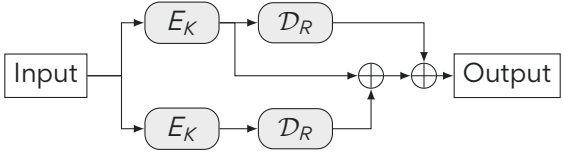


SoA external infection

- Constraint $\mathcal{D}(0) = 0$ removed
  $\Longrightarrow$ $\mathcal{D}$ can be a hash function: *non-invertibility* and *high diffusion* achieved
- R: random value seeding $\mathcal{D}$: *non-determinism* constraint fulfilled

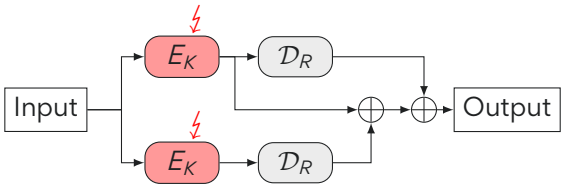
Principle of our framework

Secure only against one fault!



Principle of our framework

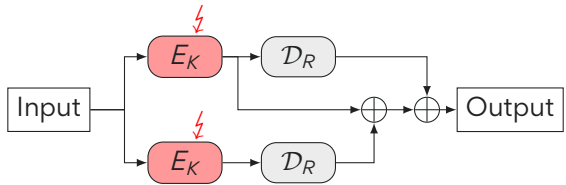
Secure only against one fault!



$$\begin{aligned}\text{Output} &= E_K^{\ell} \oplus \mathcal{D}(E_K^{\ell}) \oplus \mathcal{D}(E_K^{\ell}) \\ &= E_K^{\ell}\end{aligned}$$

Principle of our framework

Secure only against one fault!

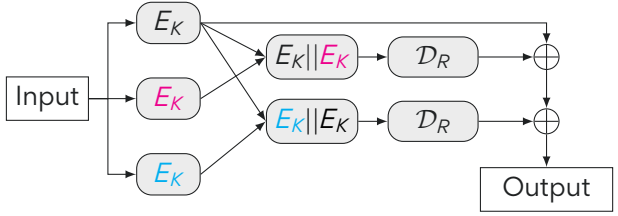


$$\begin{aligned}\text{Output} &= E_K^i \oplus \mathcal{D}(E_K^i) \oplus \mathcal{D}(E_K^i) \\ &= E_K^i\end{aligned}$$

How to get secure against several faults?

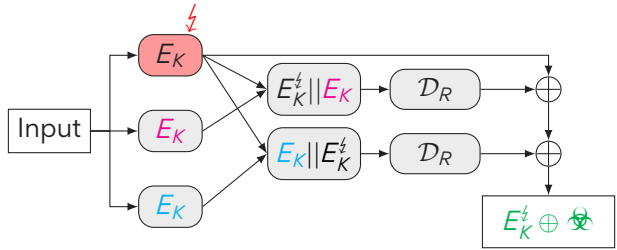
Improved construction

Construction to defend oneself against **two** faults:



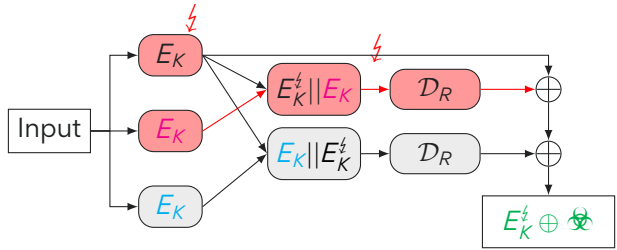
Improved construction

Construction to defend oneself against **two** faults:



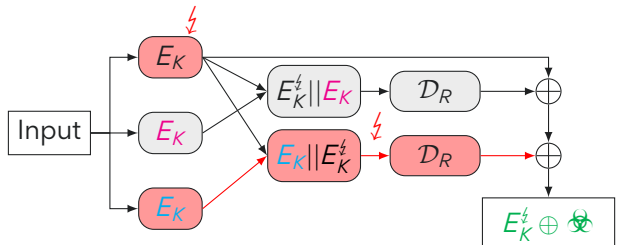
Improved construction

Construction to defend oneself against **two** faults:

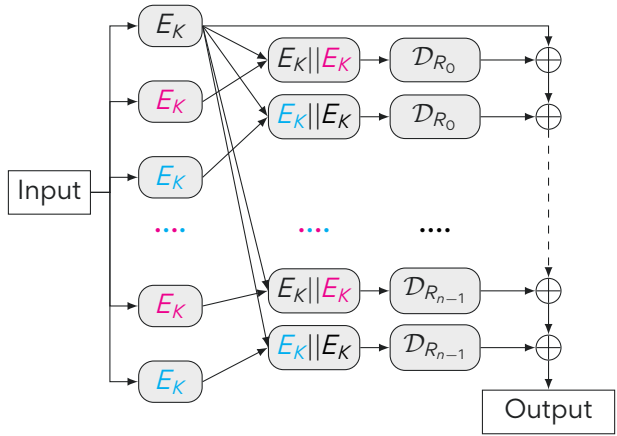


Improved construction

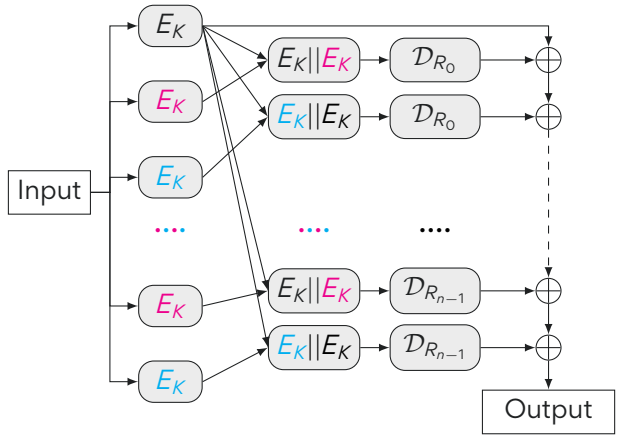
Construction to defend oneself against **two** faults:



Extension against $2n$ faults

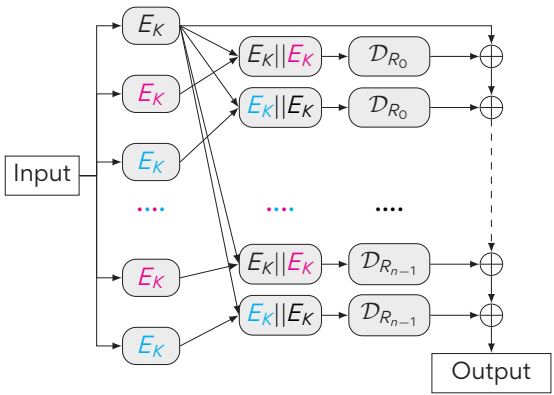


Extension against $2n$ faults



- R is changed from a couple of $\mathcal{D}$'s to another

Extension against $2n$ faults

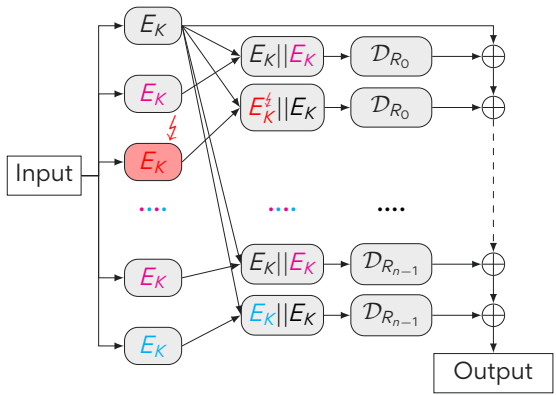


Scheme is **proven secure** in the paper

Attacker model

Per fault, the attacker:

Extension against $2n$ faults



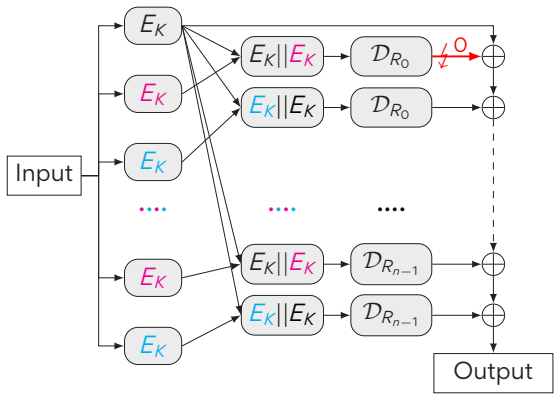
Scheme is **proven secure** in the paper

Attacker model

Per fault, the attacker:

- Can corrupt one E_K,

Extension against $2n$ faults



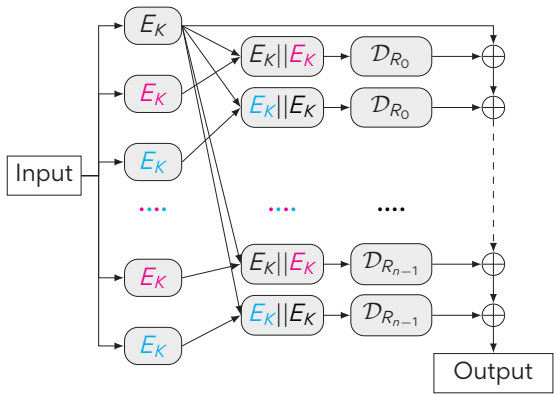
Scheme is **proven secure** in the paper

Attacker model

Per fault, the attacker:

- Can corrupt one E_K,
- Or stick at 0 one input of one XOR.

Extension against $2n$ faults



Scheme is **proven secure** in the paper

Attacker model

Per fault, the attacker:

- Can corrupt one E_K,
- Or stick at 0 one input of one XOR.

And $\mathcal{D}$ is supposed:

- ✓ *Non-invertible*,
- ✓ To have a *high-diffusion* capacity.

Conclusion

- Identification of some common flaws in the propositions of the state of the art
- Proposal of a **new solution** taking into account our observations
- **First proposal** of an infective scheme allowing one to **resist several-fault attacks**
- **Security proof** of our solutions provided in the paper
- Open question: find the best suited $\mathcal{D}$ that meets the scheme's constraints