Fault attacks on white-box computations

L'attaque en faute : la bête noire des boîtes blanches

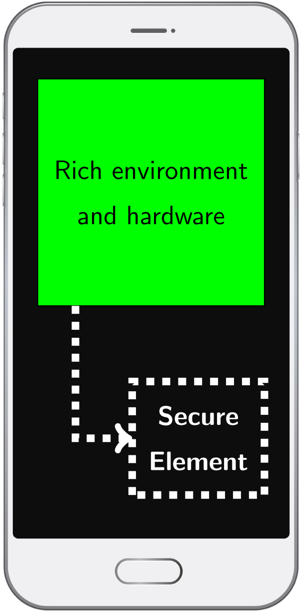
Vincent Giraud [1,2] Guillaume Bouffard [1,3]

[1]DIENS, École Normale Supérieure, Université PSL, CNRS

[2]Ingenico

[3]Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)

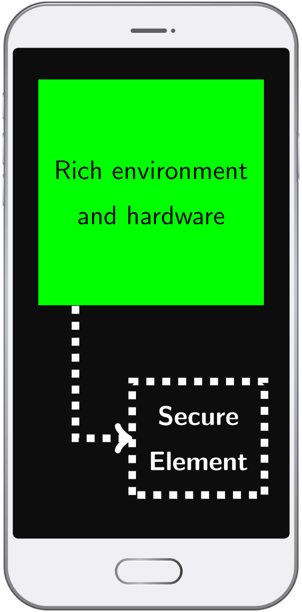
November 9, 2022



In the last twenty years, COTS (Customer Off-The-Shelf devices) became really popular and increasingly complex.

The industry has repeatedly investigated the possibility of running sensitive processes on them.

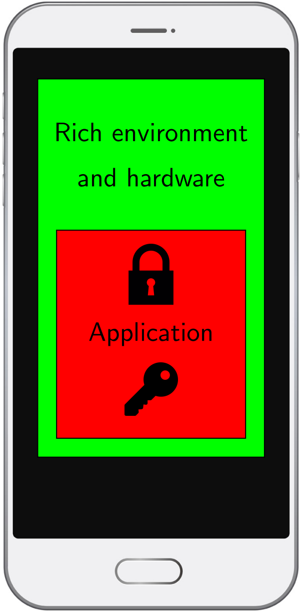
However, access to a secure environment is not ensured !



In the last twenty years, COTS (Customer Off-The-Shelf devices) became really popular and increasingly complex.

The industry has repeatedly investigated the possibility of running sensitive processes on them.

However, access to a secure environment is not ensured !



In the face of this, a lot of research has been done on the execution of sensitive operations on uncontrolled environment.

This is called the white-box model. It tries to reproduce the features of a root of trust, without it.

The considered applications are mainly found in cryptography.

As white-boxes cannot reach the security level offered by protected hardware, they require methods to complicate attacks.

Reversing and debugging are fought against using techniques including obfuscation and more advanced methods.

As white-boxes cannot reach the security level offered by protected hardware, they require methods to complicate attacks.

Reversing and debugging are fought against using techniques including obfuscation and more advanced methods.

The goal is obviously to make attacks as complicated and time-consuming as possible, even using advanced tools (such as Ghidra, IDA Pro, Radare2...). High skills in reverse engineering would also be required.

Operating in the white-box model has harsh consequences :

- while secure components can provide strong guarantees on the very long term,

  white-boxes' security are measured in days

Operating in the white-box model has harsh consequences :

- while secure components can provide strong guarantees on the very long term,

  white-boxes' security are measured in days

- performances are reduced

Operating in the white-box model has harsh consequences :

- while secure components can provide strong guarantees on the very long term,

  white-boxes' security are measured in days

- performances are reduced

- new side channels appear

Operating in the white-box model has harsh consequences :

- while secure components can provide strong guarantees on the very long term, white-boxes' security are measured in days
- performances are reduced
- new side channels appear
- sources of entropy become unreliable

Operating in the white-box model has harsh consequences :

- while secure components can provide strong guarantees on the very long term, white-boxes' security are measured in days
- performances are reduced
- new side channels appear
- sources of entropy become unreliable
- existing fault attacks can become easier to exploit, and new ones can also appear

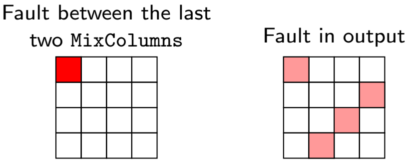
Operating in the white-box model has harsh consequences :

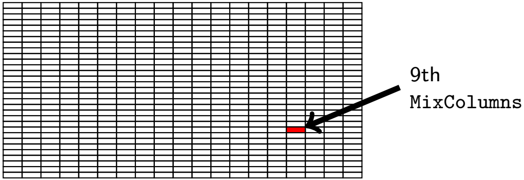
- while secure components can provide strong guarantees on the very long term, white-boxes' security are measured in days
- performances are reduced
- new side channels appear
- sources of entropy become unreliable
- existing fault attacks can become easier to exploit, and new ones can also appear

While reverse engineering such an implementation is, as intended, costly, difficult and lengthy, binary instrumentation brings another danger: automated and easy to conduct attacks.

With the help of tools such as QBDI or Rainbow, one can effectively reproduce hardware attacks on obfuscated programs or white-boxes.

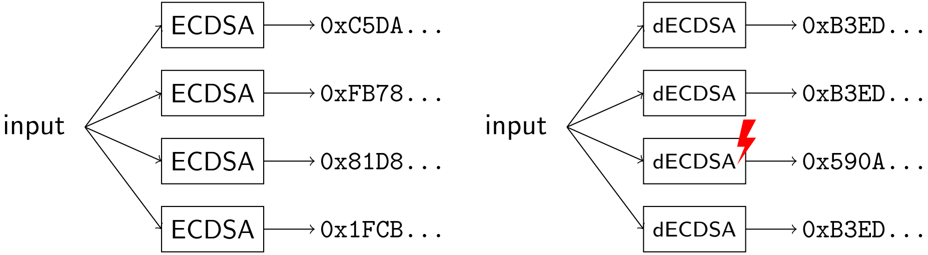
Consider the classic fault attack on AES [4].





This attack is directly transposable to the white-box model [7].

The ECDSA fundamentally relies on entropy, and operates on large values. This makes white-box implementations particularly tedious.



Having deterministic results makes differential fault attacks possible [8] [1].

White-box algorithms are doomed to face these kinds of issues.

- To properly fight against binary instrumentation, they must provide an adapted defense-in-depth.

White-box algorithms are doomed to face these kinds of issues.

- To properly fight against binary instrumentation, they must provide an adapted defense-in-depth.
- Besides, asymmetric cryptosystems are seen as handy in the industry as they provide flexible use cases.

White-box algorithms are doomed to face these kinds of issues.

- To properly fight against binary instrumentation, they must provide an adapted defense-in-depth.
- Besides, asymmetric cryptosystems are seen as handy in the industry as they provide flexible use cases.
- Finally, white-box cryptography should now also aim to address the quantum threat [3].

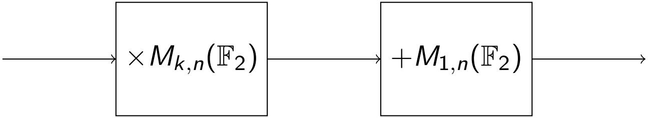
Are there any asymmetric, quantum-proof and white-boxable algorithms ?

In 2015, PQCRYPTO officially made an initial recommendation on the McEliece cryptosystem for public-key encryption, with $n = 6960$ [6].

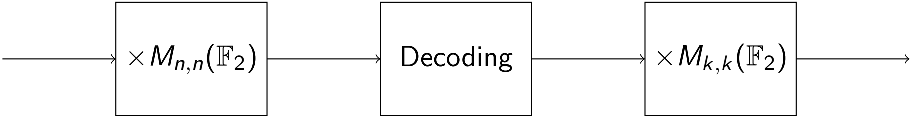
PQCRYPTO
ICT-645622

Introduced in 1978, the McEliece cryptosystem is an asymmetric algorithm heavily relying on linear error-correcting codes [5].

The encryption process uses a public key and simply consists of a matrix multiplication followed by an addition.



The decryption process uses a private key and can be summarized as a matrix multiplication, a decoding process, and another matrix multiplication.



All of these components are easily encodable, except for the decoding process.

Error-correction has a good diffusion on large data blocks, hence the difficulty to predict at compile time, and the robustness against fault attacks [2].

To make a white-boxed McEliece cryptosystem, different research axes were explored :

- explore other error-correcting codes outside of the Goppa ones ?
- modify the way codes are exploited in McEliece ?

Error-correction has a good diffusion on large data blocks, hence the difficulty to predict at compile time, and the robustness against fault attacks [2].

To make a white-boxed McEliece cryptosystem, different research axes were explored :

- explore other error-correcting codes outside of the Goppa ones ?
- modify the way codes are exploited in McEliece ?

The following task is thus to harden such an implementation in different ways:

- Adding measures against reverse-engineering and debugging
- Adding protections against binary instrumentation attacks

Error-correction has a good diffusion on large data blocks, hence the difficulty to predict at compile time, and the robustness against fault attacks [2].

To make a white-boxed McEliece cryptosystem, different research axes were explored :

- explore other error-correcting codes outside of the Goppa ones ?
- modify the way codes are exploited in McEliece ?

The following task is thus to harden such an implementation in different ways:

- Adding measures against reverse-engineering and debugging
- Adding protections against binary instrumentation attacks

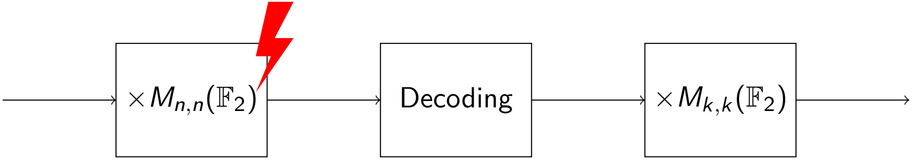
As mentioned earlier, verifying the original McEliece cryptosystem's robustness against fault attacks allows us to get a glimpse of how a white-box version can resist too.

We are looking for faults targeting the specification itself. However it's commonly recognized as strong [2].

As mentioned earlier, verifying the original McEliece cryptosystem's robustness against fault attacks allows us to get a glimpse of how a white-box version can resist too.

We are looking for faults targeting the specification itself. However it's commonly recognized as strong [2].

However, we did found a working attack based on fault injection targeting the classic way of implementing this cryptosystem.



The portability of such an attack raises many questions.

White-boxes relying on precomputation hold much better than the others against it.

The portability of such an attack raises many questions.

White-boxes relying on precomputation hold much better than the others against it.

Can the attack be successfully adapted to reach them too ?

This requires to reproduce it using data mutation.

Although the white-box model is very useful, implementations require a lot of efforts to be deployed and maintained, and provide limited resilience.

Binary instrumentation shows how threats such as fault attacks can transpose or add themselves when one switches to this model, and how a lot of the original countermeasures can become inefficient in it.

The McEliece cryptosystem, despite the size of its keys, seems to offer relevancy in this domain of research.

Sven Bauer, Hermann Drexler, Maximilian Gebhardt, Dominik Klein, Friederike Laus, and Johannes Mittmann.
Attacks against white-box ECDSA and discussion of countermeasures - a report on the WhibOx contest 2021.
https://eprint.iacr.org/2022/448.

Pierre-Louis Cayrel and Pierre Dusart.
McEliece/Niederreiter PKC: Sensitivity to Fault Injection.
In *5th International Workshop on Future Engineering, 2010*, pages 1–6, Busan, South Korea, May 2010.

Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI).
Anssi views on the post-quantum cryptography transition.
2022.
https://www.ssi.gouv.fr/en/publication/anssi-views-on-the-post-quantum-cryptography-transition/.

P. Dusart, G. Letourneux, and O. Vivolo.
Differential fault analysis on a.e.s., 2003.
https://eprint.iacr.org/2003/010.

R.J. McEliece.
A public-key cryptosystem based on algbraic coding theory.
1978.

PQCRYPTO.
Post-quantum cryptography for long-term security, 2015.
www.pqcrypto.eu.org.

Eloi Sanfelix, Cristofaro Mune, and Job de Haas.
Practical attacks against obfuscated ciphers.
2015.

Christophe Giraud Agathe Houzelot Chaoyun Li Mohammad Mahzoun Adrián Ranea Guillaume Barbu Ward Beullens, Emmanuelle Dottax and Jianrui Xie.
ECDSA white-box implementations: Attacks and designs from WhibOx 2021 contest.
https://eprint.iacr.org/2022/385.