Sécurité des SoC complexes hétérogènes – de la TEE au matériel

Technologie ARM TrustZone® et attaques combinées sur SoC

GDR SoC2 – Jeudi 20 septembre 2018, Campus de Jussieu, Paris

Fabien MAJÉRIC
20.09.2018

ARM TrustZone® et attaques combinées sur SoC

Sommaire:

1. Vue d'ensemble de la technologie ARM TrustZone®
   - 1.1 Principe de la TrustZone®
   - 1.2 Architecture de la TrustZone®
2. Expérimentations d'attaques
   - 2.1 Perturbation de la TrustZone®
   - 2.2 Attaque d'un Trusted Execution Environment basé sur de la TrustZone®

1. Vue d'ensemble de la technologie ARM TrustZone®

1.1 Principe de la TrustZone®

1.2 Architecture de la TrustZone®

TrustZone® - Principe

TrustZone® - Principe

- **<u>Contexte</u>:** la protection de données est généralement effectuée par des blocs matériels dédiés + cryptographie.

→ Or, un attaquant peut prendre le contrôle des **entrées/sorties** de ces blocs (ex: sans accéder aux clefs, ils accèdent aux messages déchiffrés)

- **<u>Solution proposée par ARM</u>:** Un état de sécurité appliqué à l'ensemble des ressources matérielles.

→ Une sécurité de bout-en-bout.

→ Des zones du système sécurisées durant un processus.

→ Protocoles de sécurité adaptés (ex: Secure Boot, authenticated debug)

TrustZone® - Principe

- **(Système)** Partitionner l'attribution des ressources matérielles et logicielles en fonction des niveaux de privilèges.
  → Existence de deux niveaux, appelés « mondes »:
  - Monde *Secure* (« *Secure World* ») pour les sous-systèmes sécurisés
  - Monde *Normal* (« *Normal World* ») pour tout le reste
  - Aucune ressource du *Secure World* ne peut être accédée par un composant du *Normal World*

- **(Processeur)** Un unique processeur physique fait fonctionner deux processeurs virtuels isolés matériellement (*Secure/Non-Secure*).

- **(Debug)** Contrôle d'accès au débogage du *Secure World* sans impacter la visibilité du débogage du *Normal World.*

1. Vue d'ensemble de la technologie ARM TrustZone®

1.1 Principe de la TrustZone®

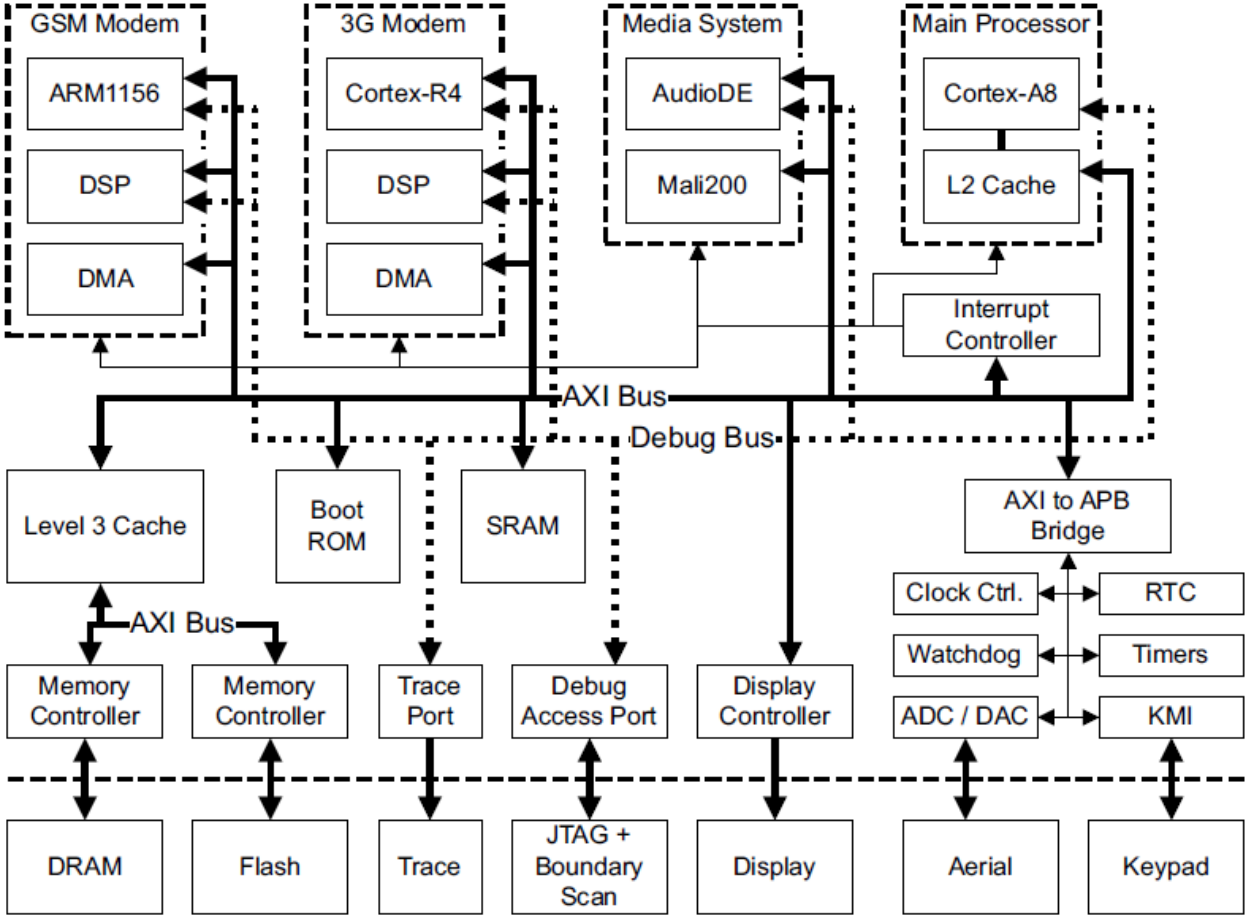
1.2 Architecture de la TrustZone®

TrustZone® - Architecture

- **(Système)**
- **(Processeur)**
- **(Debug)**

TrustZone® - Architecture

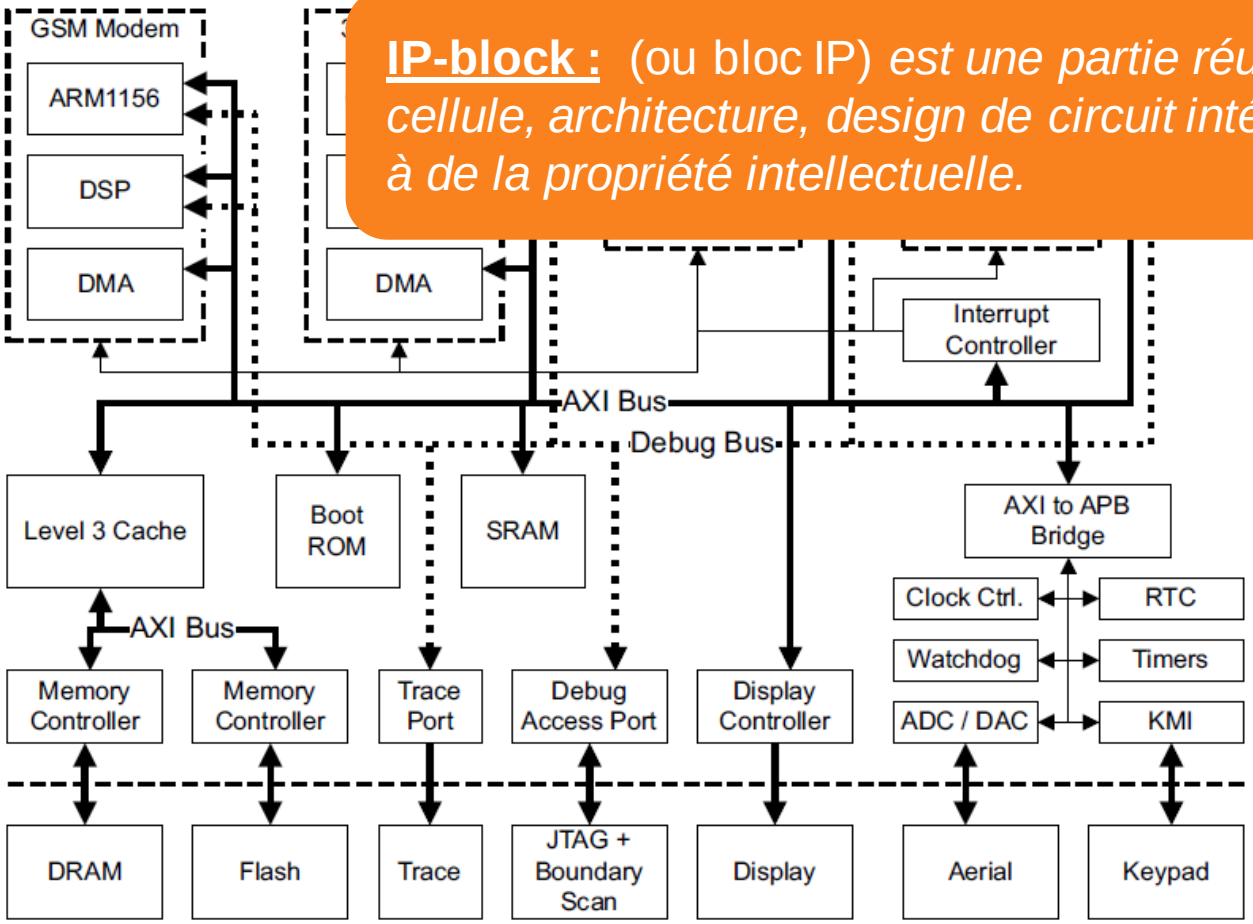
(Système) Ajout d'un mécanisme aux IP-block pour séparer les ressources matérielles en deux mondes.



TrustZone® - Architecture

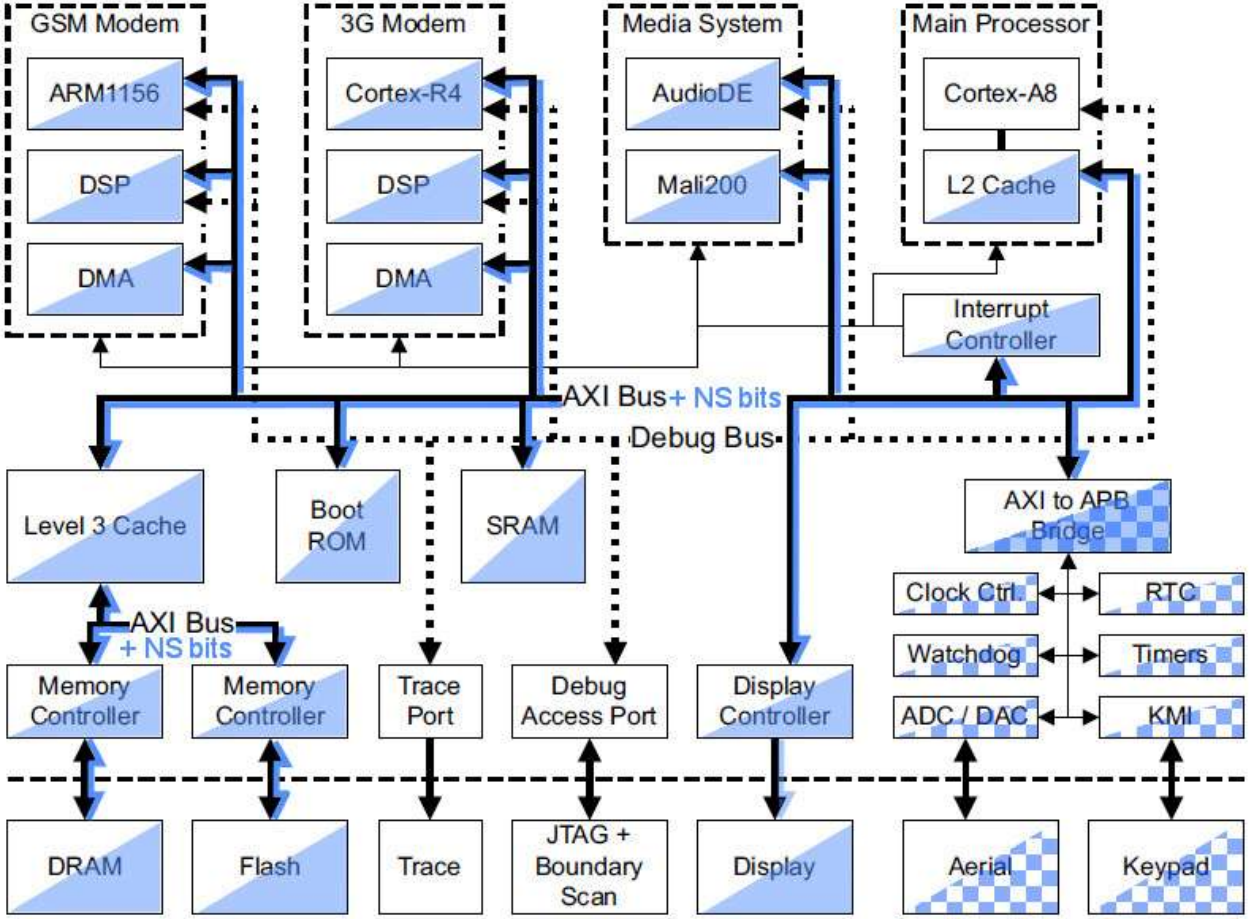
(Système) Ajout d'un mécanisme aux **IP-block** pour séparer les ressources matérielles en deux mondes.

IP-block : (ou bloc IP) *est une partie réutilisable, cellule, architecture, design de circuit intégré, soumis à de la propriété intellectuelle.*



TrustZone® - Architecture

(Système) Ajout d'un mécanisme aux IP-block pour séparer les ressources matérielles en deux mondes.



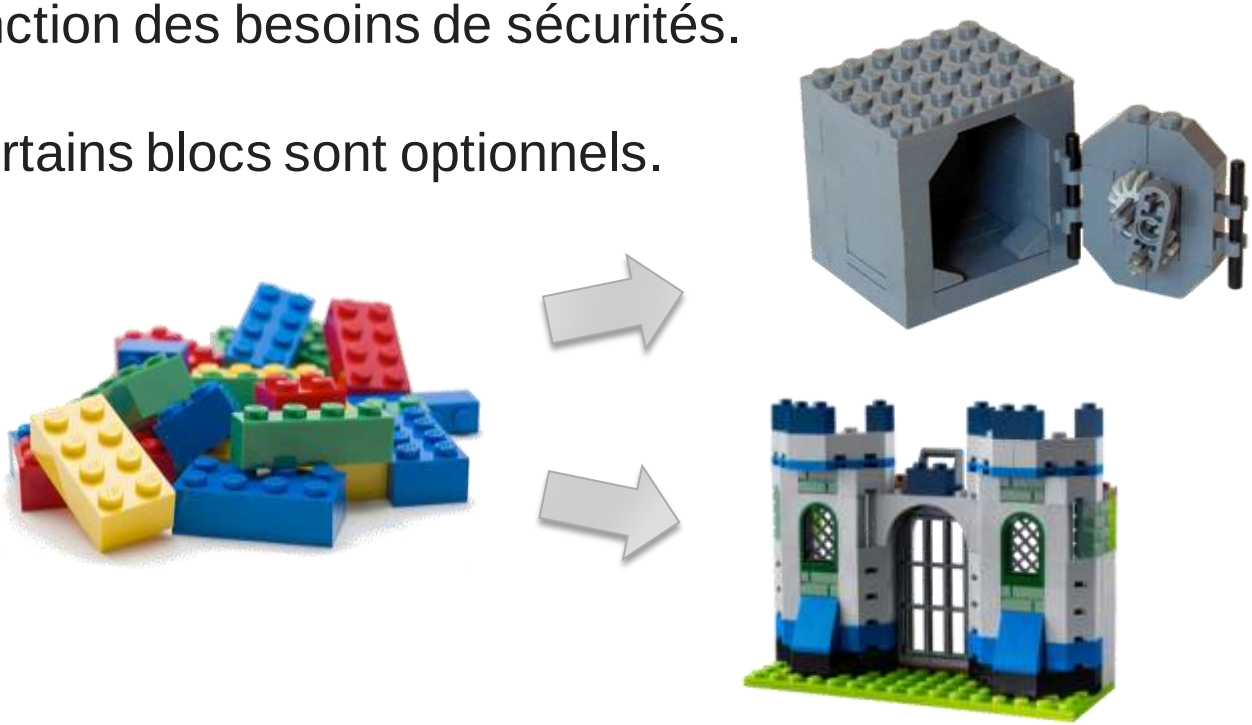
AMBA3 AXI Bus

AMBA3 APB

TrustZone® - Architecture

(Système) – blocs IP

- ARM fournit un ensemble de blocs IP qui permettent de concevoir différentes architectures matérielles.
- Ces blocs peuvent êtres combinés selon différentes manières en fonction des besoins de sécurités.
- Certains blocs sont optionnels.



TrustZone® - Architecture

(Système) – blocs IP caractéristiques:

- **AMBA3 AXI system bus** (Advanced eXtensible Interface).
- **AMBA3 APB** (Advanced Peripheral Bus).
- **Mémoires**

TrustZone® - Architecture

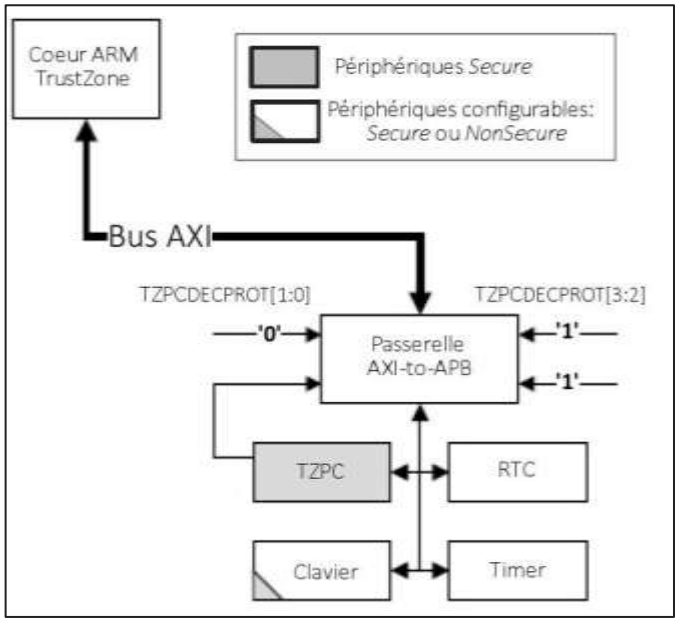
(Système) – blocs IP caractéristiques:

- **AMBA3 AXI system bus** (Advanced eXtensible Interface).
  - Ajout de 2 signaux de contrôles sur le bus principal du système
  - 2 bits Non-Secure (NS):
    - **AWPROT[1]**: Write transaction – bas pour Secure, haut pour NS.
    - **ARPROT[1]**: Read transaction – bas pour Secure, haut pour NS.
  - Les maîtres NS ont leur bit NS au niveau haut.
    (Impossible pour eux d'accéder à un esclave Secure. Erreur dans le décodage d'adresse.)
  - Les erreurs **SLVERR** (slave error) ou **DECERR** (decode error) peuvent être générées.

TrustZone® - Architecture

(Système) – blocs IP caractéristiques:

- **AMBA3 APB** (Advanced Peripheral Bus).
  - Le bus APB ne transporte pas les signaux NS. *(Assure la compatibilité avec les périphériques AMBA2 APB).*
  - Possibilité de sécuriser les périphériques, tels que les contrôleurs d'interruptions, timers et I/O utilisateurs (ex. claviers).
  - Une passerelle AXI-to-APB gère la sécurité des périphériques connecté à l'APB. Elle rejette les transactions avec une configuration de sécurité inappropriés.



TrustZone® - Architecture

(Système) – blocs IP caractéristiques:

- **Memoires.**

→ Un système 32 bits peut être vu comme avec un système d'adresses 33 bits.

- 32 bits d'adresses physiques pour les transactions ***Secure***.
- 32 bits d'adresses physiques pour les transactions ***Non-Secure***.

→ Aliasing mémoire. Les maitres *Secure* peuvent accéder aux esclaves *NS*

- Les concepteurs des systèmes doivent prendre en compte la cohérence des données,

TrustZone® - Architecture

(Système) – autres blocs IP

- **Les principaux:**
  - AMBA3 AXI to AMBA 2 AHB Bridge
  - Level 2 Cache Controller
  - DMA Controller (DMAC)
  - TrustZone Address Space Controller (TZASC)
  - AMBA3 AXI TrustZone Memory Adapter (TZMA)
  - Generic Interrupt Controller (GIC)
  - AMBA3 TrustZone Protection Controller (TZPC)

(Source [1])
Plus d'infos → https://developer.arm.com/technologies/trustzone

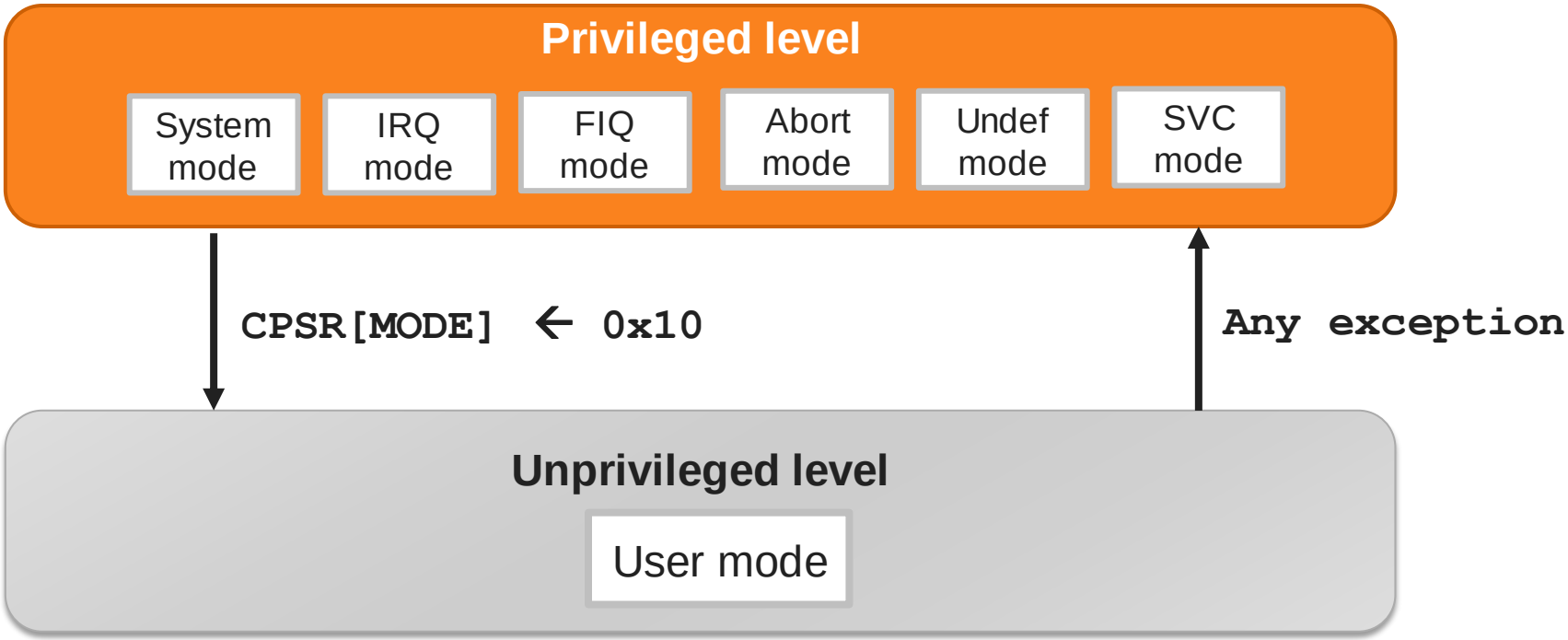
TrustZone® - Architecture

(Processeur) - avant le TZ, avant ARMv6

TrustZone® - Architecture

(Processeur) - avant le TZ, avant ARMv6

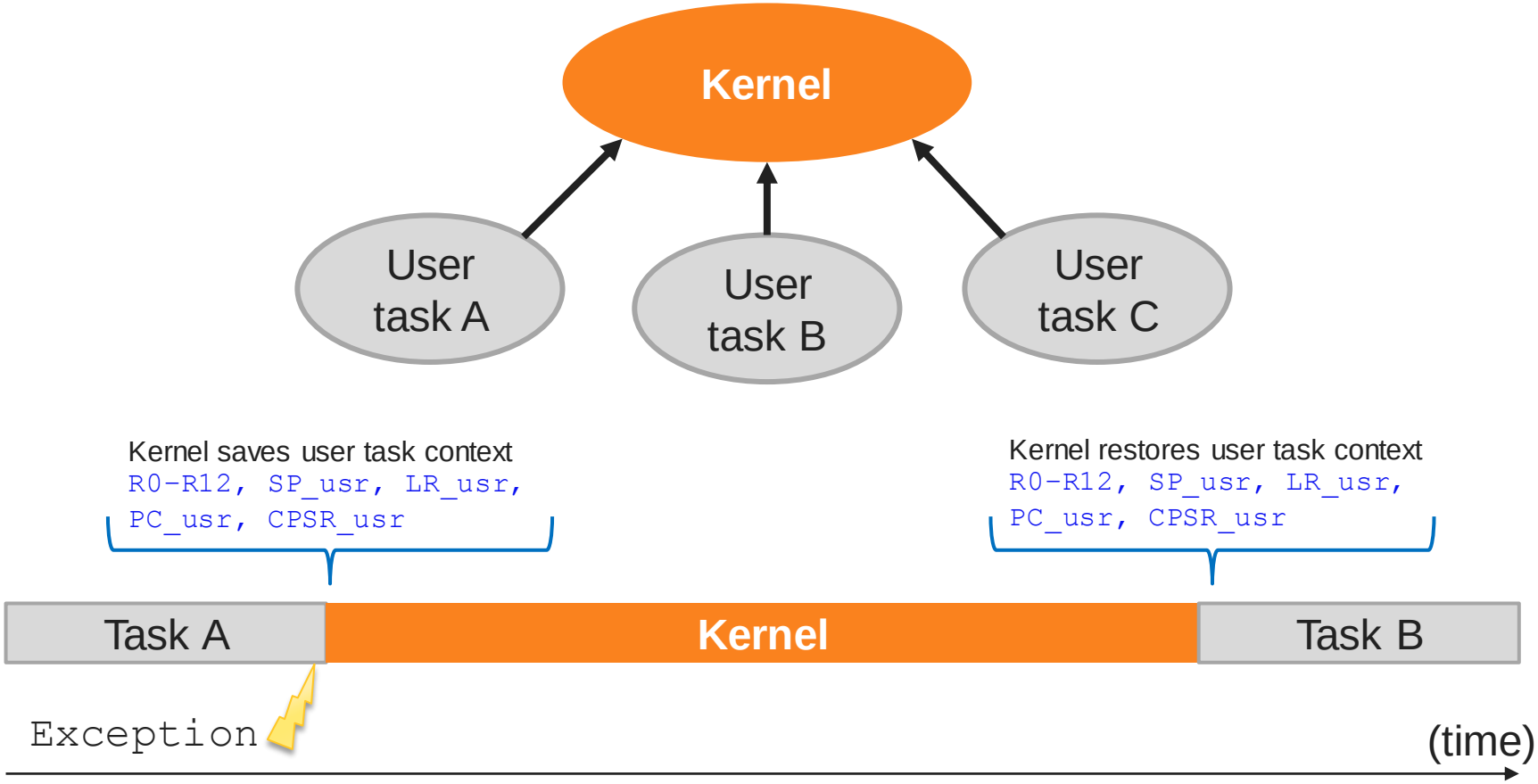
- Deux niveaux de privilèges:
  - Non-privilégié : «user mode »
  - Privilégié: « mode Kernel »



TrustZone® - Architecture

(Processeur) - avant le TZ, avant ARMv6

- Deux niveaux de privilèges:



TrustZone® - Architecture

(Processeur) - TZ

- ARM1176 est le premier CPU intégrant le TrustZone (2003).

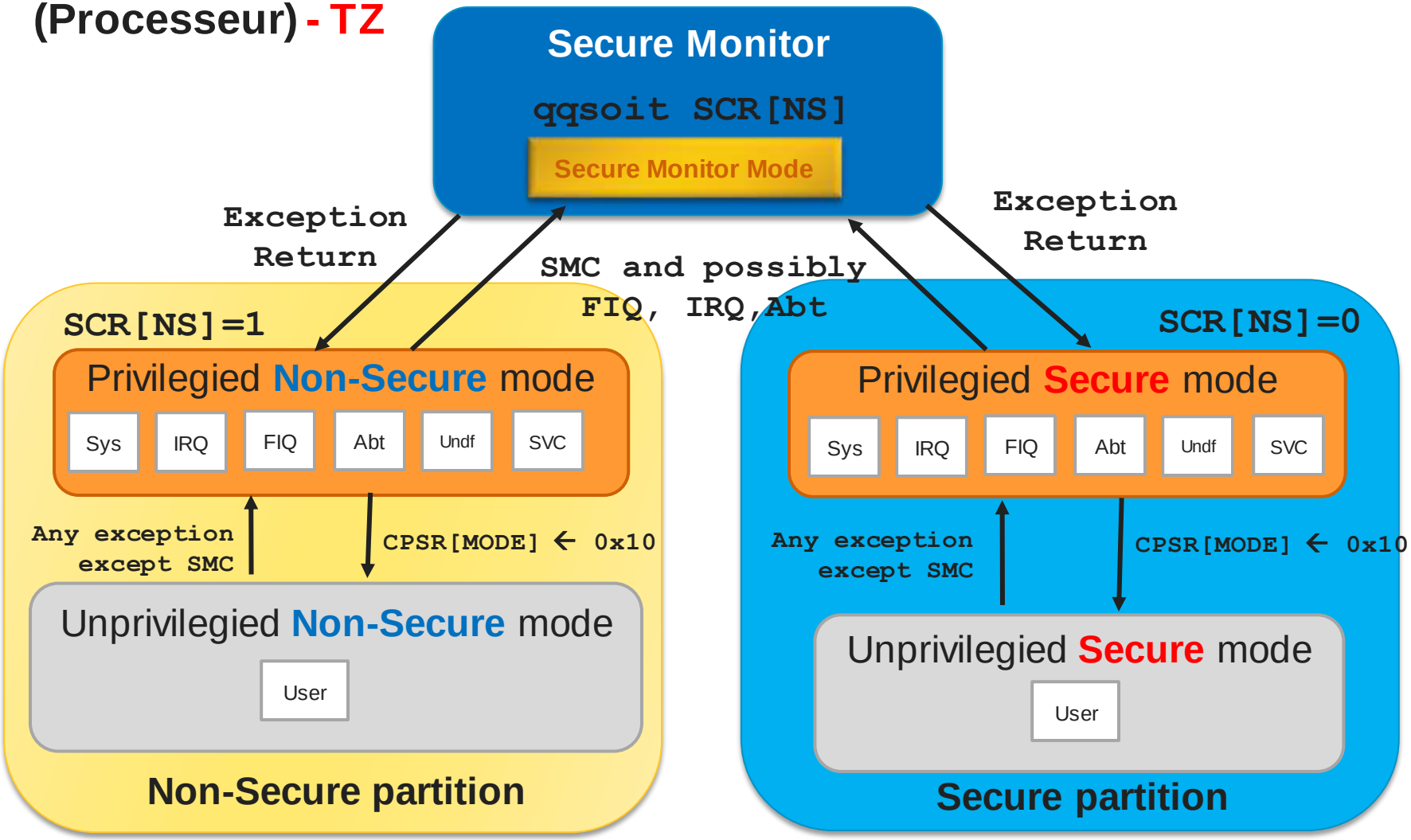
- Un nouveau niveau de privilège est ajouté: « *Secure* »

- Le programme nommé « *Secure Monitor* » contrôle la transition de l'état « *Non-Secure* » à l'état « *Secure* »

- Deux registres définissent les 4 états du CPU:
  - Le Secure Configuration Register (Secure / NonSecure)
  - Le Current Status Program Register (Privilégié / NonPrivilégié)

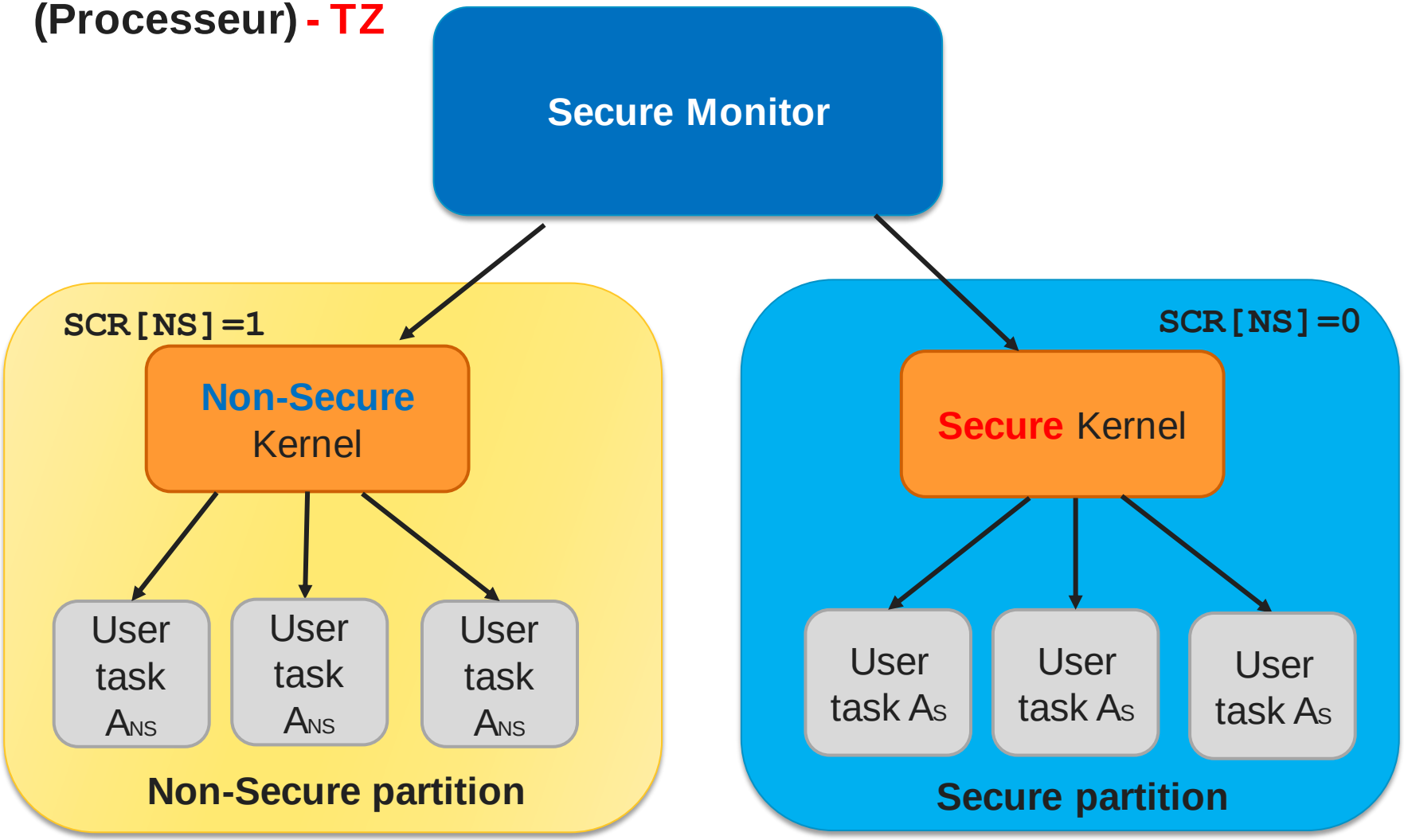
TrustZone® - Architecture

(Processeur) - TZ



TrustZone® - Architecture

(Processeur) - TZ



TrustZone® - Architecture

(Processeur) - TZ - Changement de contexte

Secure Monitor saves NS partition context
All registers of all modes, except Secure Monitor mode

Secure Monitor restores partition context
All registers of all modes, except Secure Monitor mode

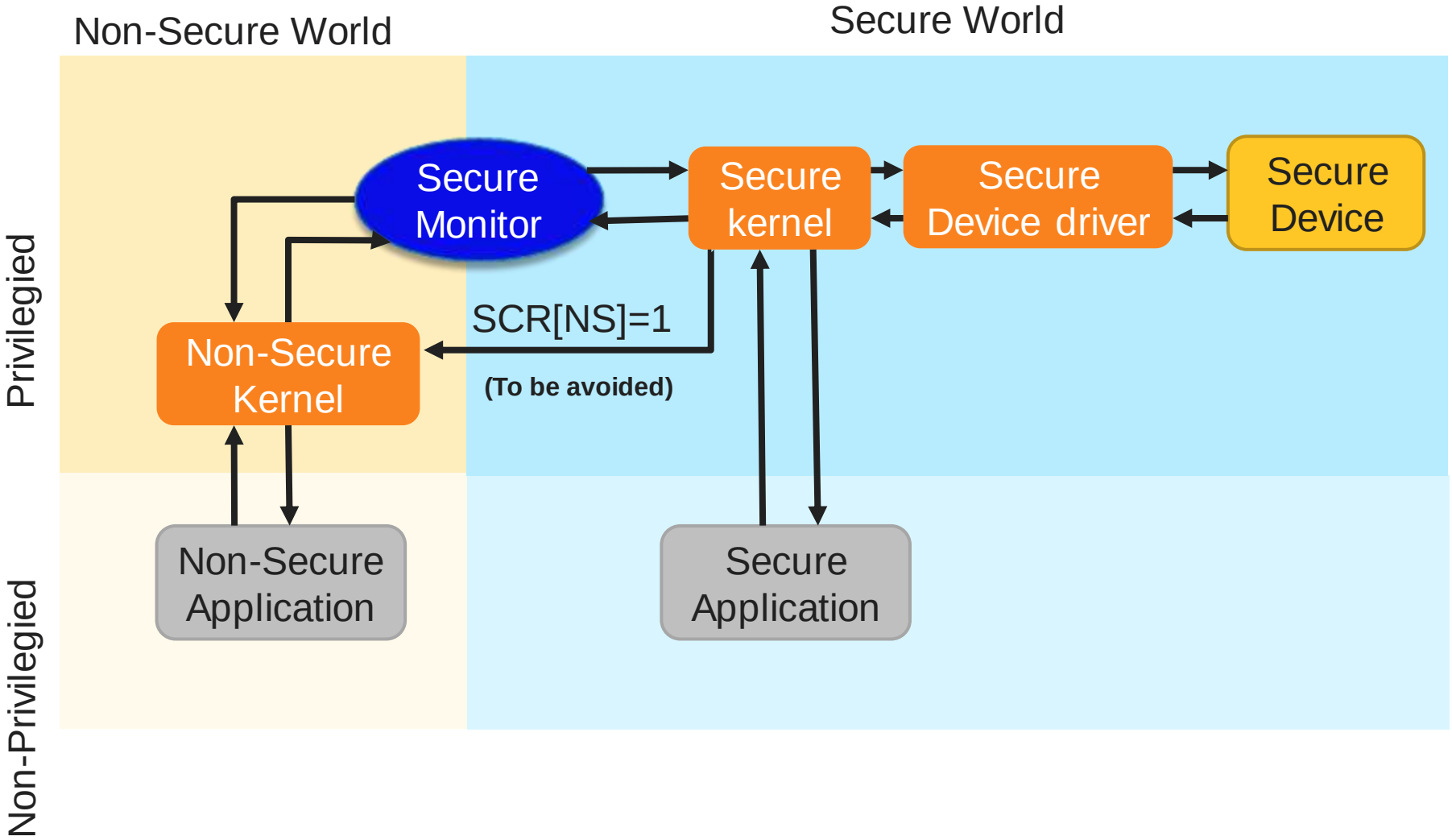
Non-Secure partition	Secure Monitor	Secure partition

SMC and possibly IRQ, FIQ, Abort

(time)

TrustZone® - Architecture

(Processeur) - TZ - Changement de contexte



TrustZone® - Architecture

(Processeur) - TZ

- La valeur du bit *NS* est propagée à l'ensemble du système par le bus AXI. Cette valeur dépend du processeur virtuel qui effectue le processus (instruction ou accès à des données)
- Le processeur virtuel *Non-Secure* n'a accès qu'aux ressources *Non-Secure*.
- Le processeur virtuel *Secure* peut accéder à l'ensemble des ressources.
- La cohérence des données est laissé au soin des concepteurs du système (différents niveaux de mémoires)

TrustZone® - Architecture

(Debug)

TrustZone® - Architecture

(Debug)

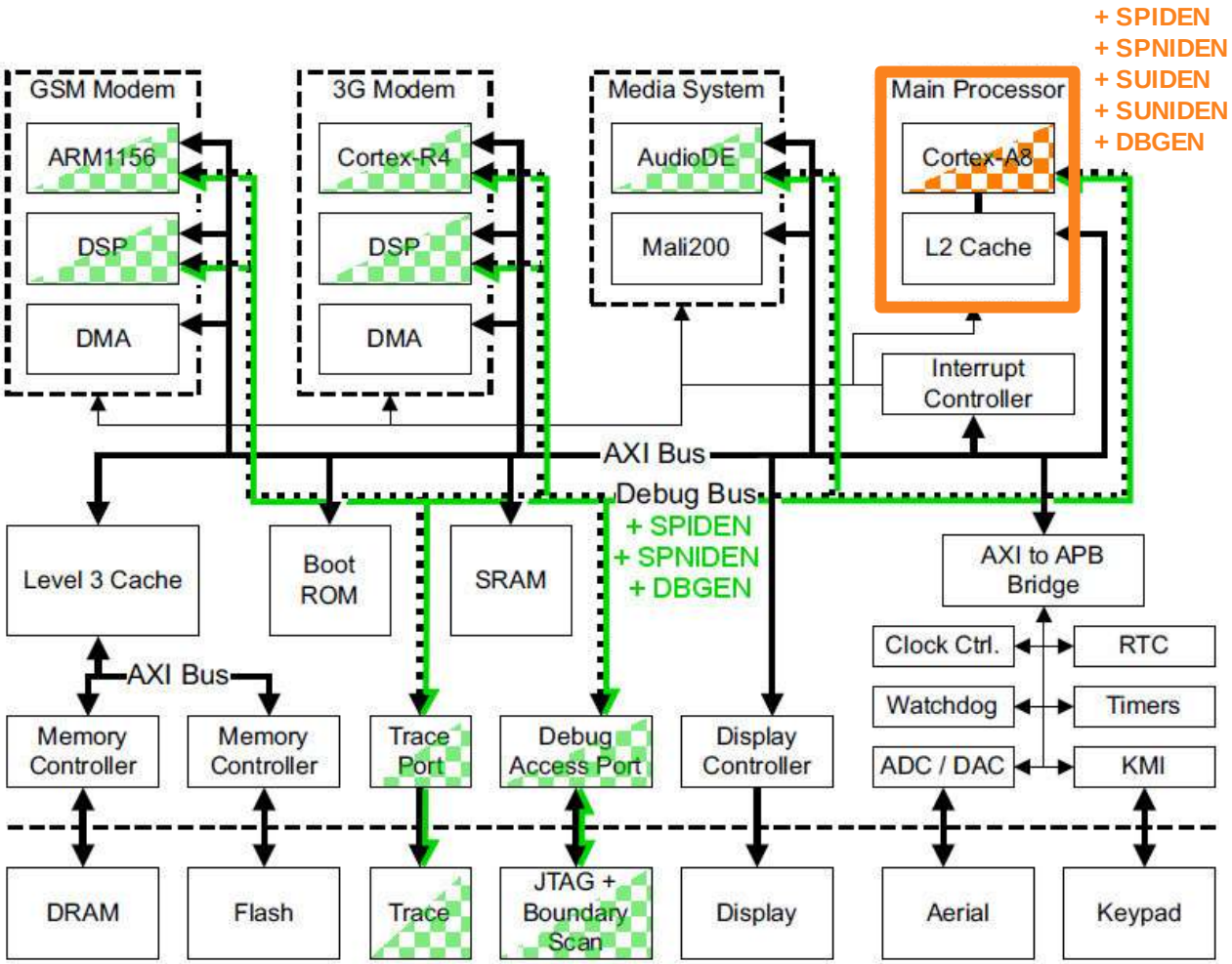
- Les systèmes ARM intègrent la technologie de débogage ARM CoreSight™ (Débogage et Traces pour les SoC)
- CoreSight™ peut être utilisé on-chip / off-chip.
- 6 signaux de contrôles sont utilisés pour sécuriser le débogage:

		Secure	***Non-Secure***
Invasif	DBGEN	SPIDEN	SUIDEN
Non invasif	NIDEN	SPNIDEN	SUNIDEN

TrustZone™

TrustZone® - Architecture

(Debug)



Signaux CoreSight™ de contrôle du débogage processeur(s)

Signaux CoreSight™ de contrôle du débogage système

2. Expérimentations d'attaques

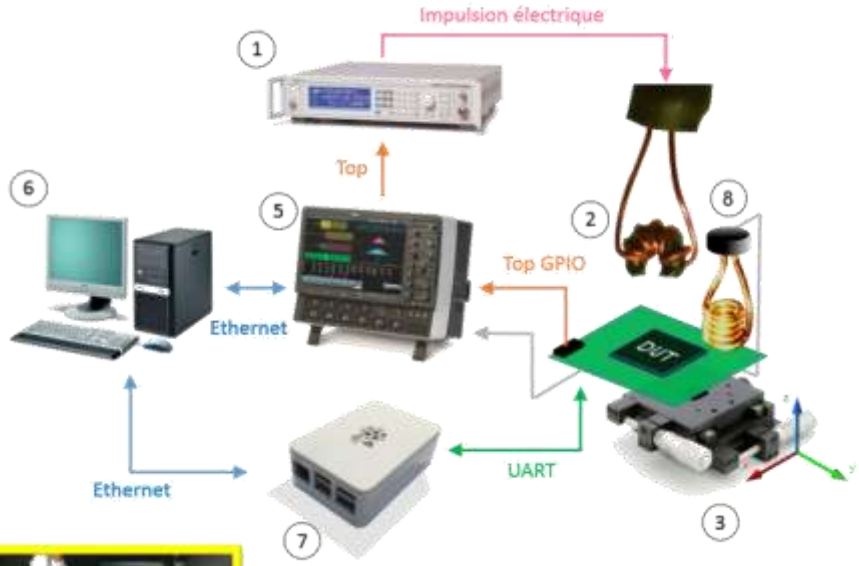
2.1 Perturbation de la TrustZone®

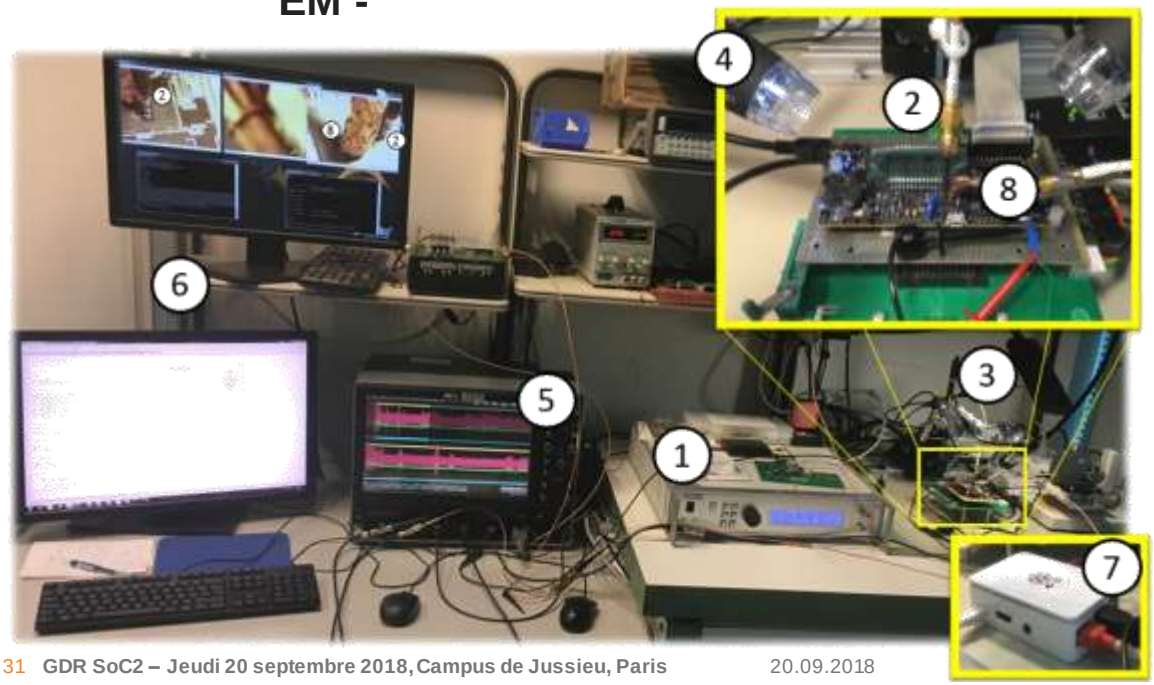
2.2 Attaque d'un Trusted Execution Environment basé sur de la TrustZone®

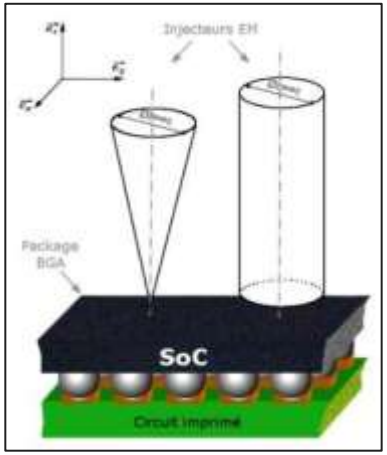
Expérimentations d'attaques

- **Setup** (Banc LIRMM [2])

- Injecteurs EM -







2. Expérimentations d'attaques

2.1 Perturbation de la TrustZone®

2.2 Attaque d'un Trusted Execution Environment basé sur de la TrustZone®

Expérimentations d'attaques - Perturbation TZ

Objectif

- Plusieurs blocs IP sont impliqués dans la configuration du niveau de sécurité de la TZ.

- Banques de registres qui définissent et transmettent le niveau de sécurité à l'ensemble du SOC. Ex :

  CPU, SCR[NS] → bus AXI, A(R/W)PROT[1] → TZPC, TZPCDECPROT

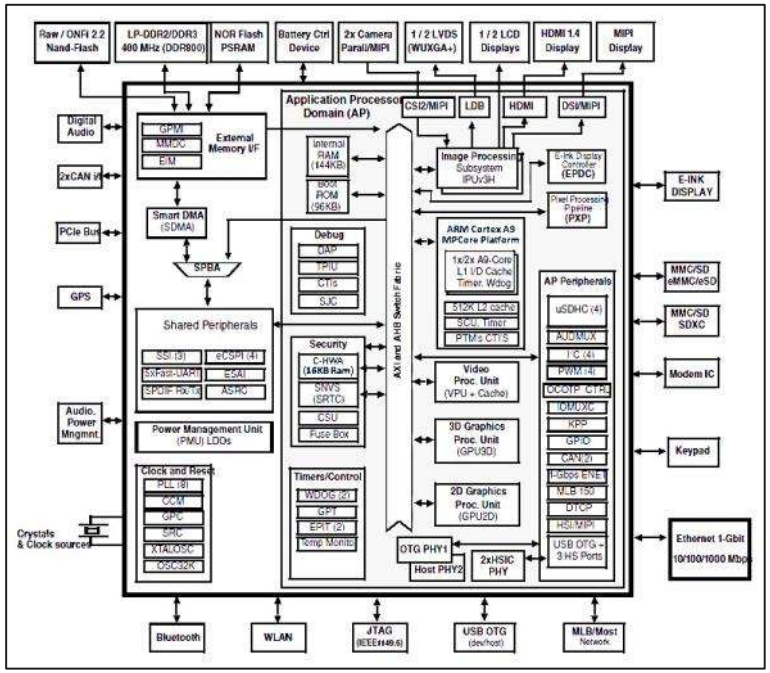
→ Est-t-il possible de modifier la configuration des registres TZ par perturbation EM ?

Expérimentations d'attaques - Perturbation TZ

Véhicule de test

- SoC très répandu dans l'automotive et l'IoT. Dédié au multimédia.1 CPU ARM Cortex-A9 MPCore™, architecture ARMv7-A, cadencé jusqu'à 1,3GHz, 32 bit. Technologie CMOS 40 nm.

- 34 périphériques programmables par banques de registres, dont ceux liés à la TZ.



Expérimentations d'attaques - Perturbation TZ

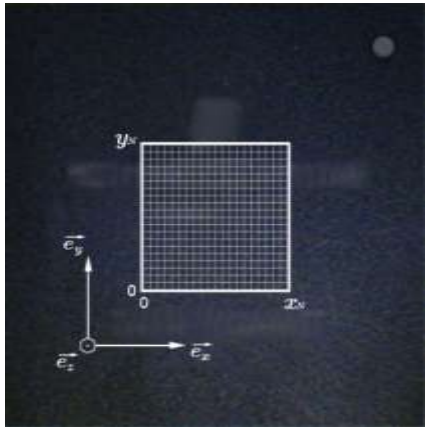
Procédure

- On ne se limite pas qu'aux registres liés à la TZ (vérification de faisabilité)

- Lecture des valeurs des checksums des 34 contrôleurs
- Injection EM
- Contrôle des valeurs des checksums

Paramètre	Valeur
Surface considérée	6000 µm²
Pas (Δx,Δy)	(60 µm, 60 µm)
Nb d'essais/pt	1
Amplitude	+400V
Durée d'impulsion	6ns

Matrice (101 x 101)
10201 essais



Expérimentations d'attaques - Perturbation TZ

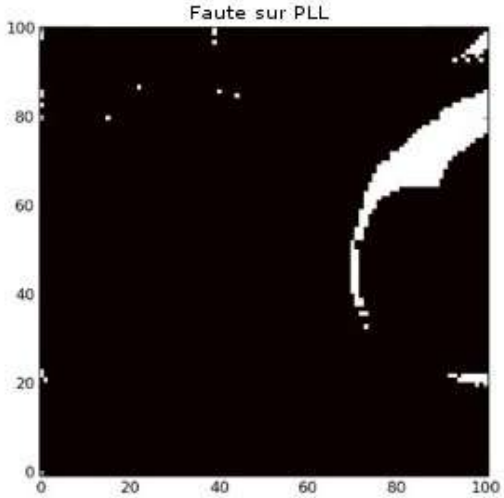
- Checksum calculés sur chaque banque de registres

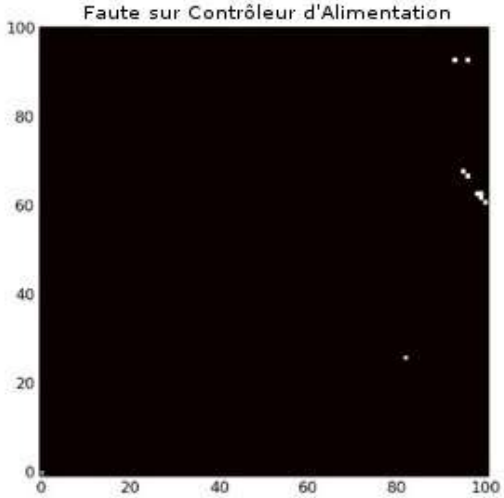
43% des injections

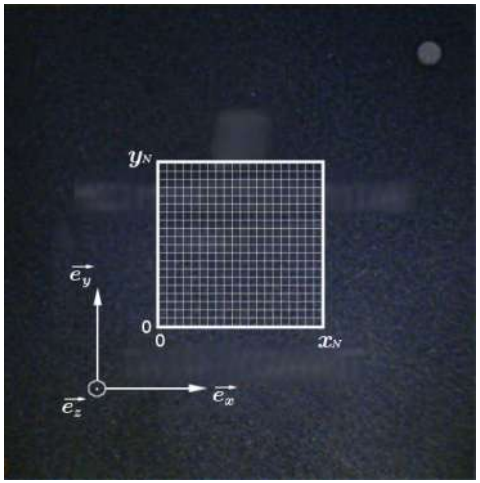
Contrôleur	Nb. de Checksum modifiés
PLL	4235
PULSE_EM	108
Alimentations	15
Module de sécurité 1/2	14
UART 2	12
Power Gating CPU	10
USB	9
Horloges	5
I/O multiplexeur	4
<reset> du système	4
Digital Host	3
Module de sécurité 2/2	3
Adaptateur de données vidéos	2
Passerelle AHB-to-IP 1	2
Passerelle AHB-to-IP 2	2
DDR	2
Ethernet	2

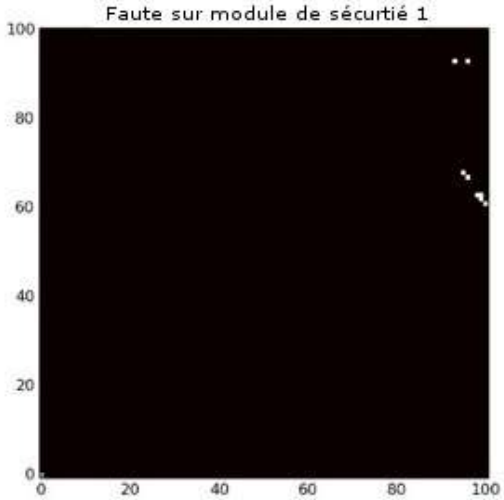
Contrôleur	Nb. de Checksum modifiés
GPIO	2
Interface modules externes	2
Interface audio	2
Module traitement vidéo	2
Module de sécurité 3	2
Multiplexeur Audio	2
OTP	2
Périphériques partagés	2
Pulse Width Modulation	2
ROM	2
SPI	2
Synchronous Serial Interface	2
Timer interruptions	2
UART 1	2
USBphy	2
MIPI_CSI	0
MIPI_DSI	0

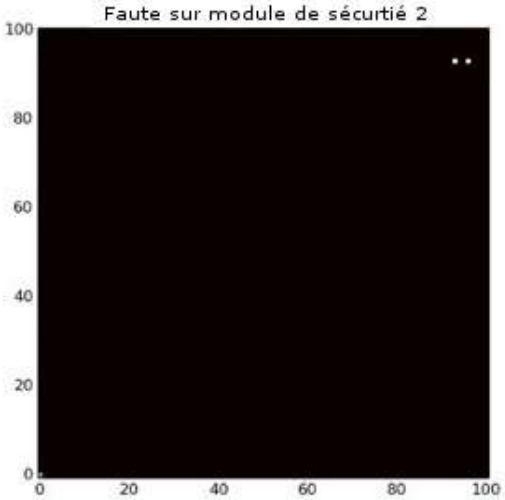
Expérimentations d'attaques - Perturbation TZ

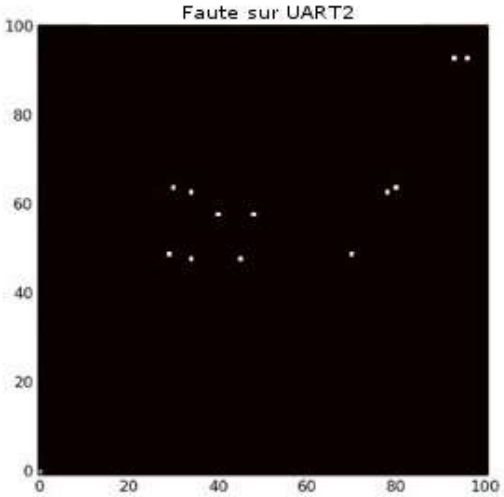












Expérimentations d'attaques - Perturbation TZ

Bilan:

- Sur 34 contrôleurs, 32 ont eu les valeurs de registres modifiés
- 2 points spatiaux produisent des modifications pour la majorité des contrôleurs. Détecteur ? Artéfact ?
- Les registres les plus impactés (43% des injections) sont liés aux PLL et à l'alimentation.

  Perturbation ⇒ modification directe des valeurs ?
  Perturbation ⇒ prise en compte du contrôleur ⇒ ajustement des valeurs ?
  (ex. DVFS)

- L'injection EM semble ne pas être adaptée pour modifier des valeurs de registres en statique
  ⇒ Pas un bon vecteur d'attaque pour modifier des configuration TZ.

2. Expérimentations d'attaques

2.1 Perturbation de la TrustZone®

2.2 Attaque d'un Trusted Execution Environment basé sur de la TrustZone®

Expérimentations d'attaques - Attaque TEE/TZ

Objectif

→ Evaluer la faisabilité de perturber le fonctionnement d'un TEE/TZ (en boite noire).

- Expérimentations sur une application particulière du TEE/TZ
- **Difficultés**: Localisation spatiale et temporelle d'un processus dans un SoC.
- **Méthodologie**:
  1. Analyse *Side-channel* EM
     - Documentation publique
     - Localisation temporelle
  2. Perturbations
     - Balayage spatial

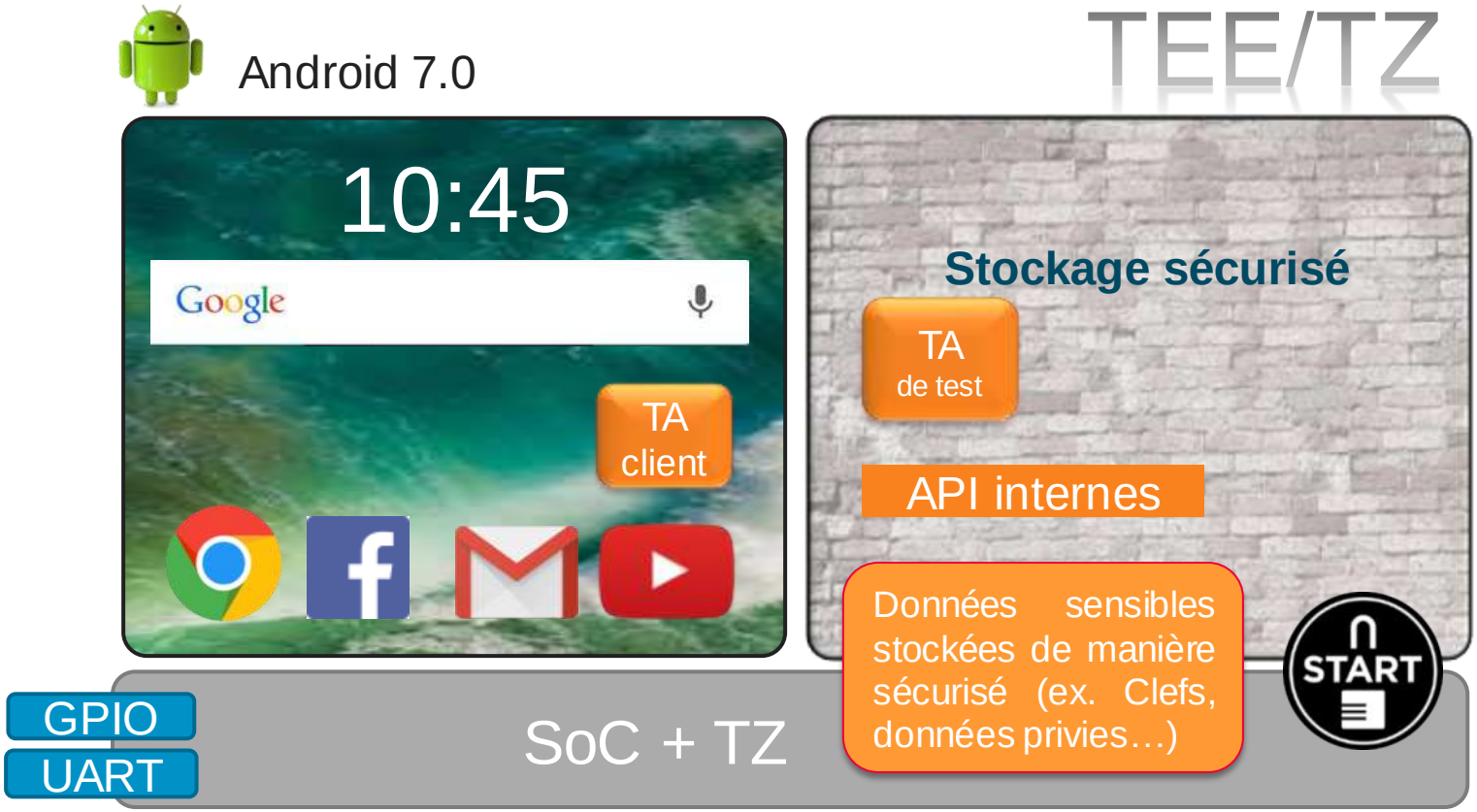
Expérimentations d'attaques - Attaque TEE/TZ

Véhicule de test

- SoC 8 CPU ARM Cortex-A53, architecture ARMv8-A, cadencé jusqu'à 1,2GHz, 64 bits.
- Android v7.0, TEE basé sur de la TrustZone, conforme aux recommandations GlobalPlatform [3].
- Carte de test « classique » (GPIO, UART, etc…)

Expérimentations d'attaques - Attaque TEE/TZ

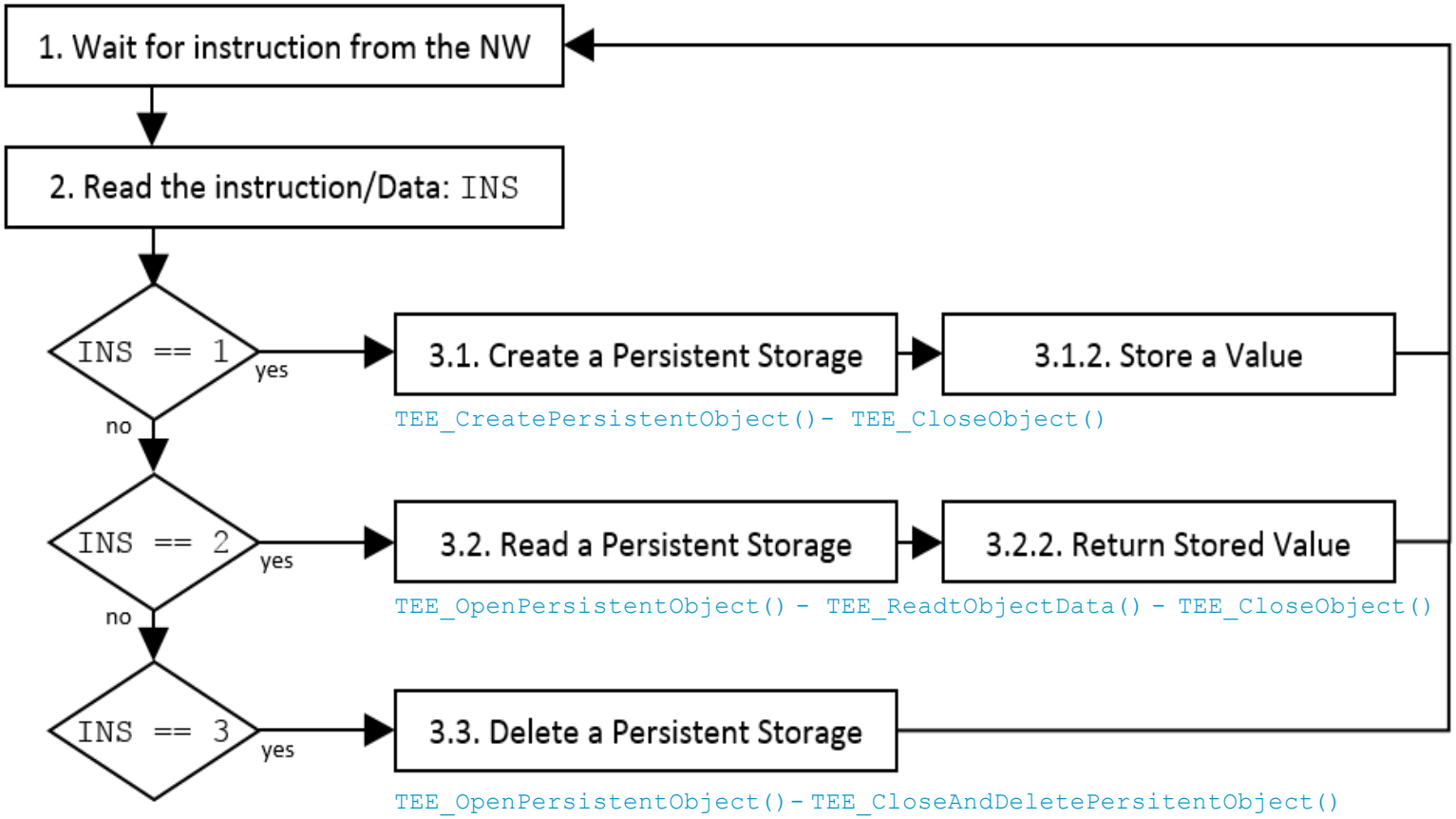
- Cible des expérimentations – API interne au Secure Storage



- TA de Test -

Expérimentations d'attaques - Attaque TEE/TZ

- Cible des expérimentations – API interne au Secure Storage

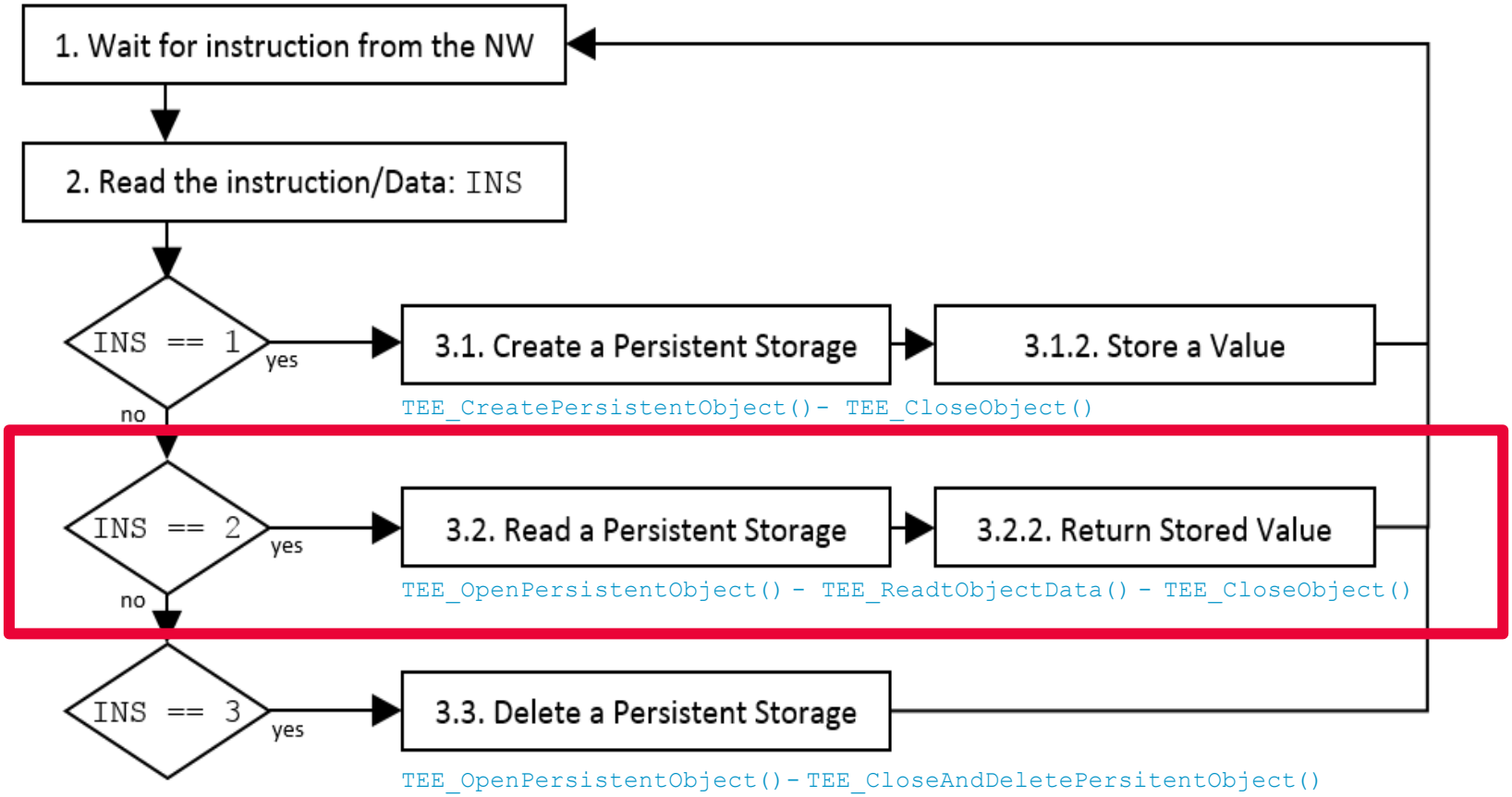


`TEE_xxxx()` : TEE Internal API involved

- TA de Test -

Expérimentations d'attaques - Attaque TEE/TZ

- Cible des expérimentations – API interne au Secure Storage



`TEE_xxxx()` : TEE Internal API involved

- TA de Test -

Expérimentations d'attaques - Attaque TEE/TZ

EM – complexité de localisation spatiale et temporelle

Pour injecter une faute:

- Besoin d'identifier le processus ciblé
- Besoin d'être très précis
- Besoin d'une référence pour trigger. Mais il n'y a ni GPIO, ni `printf()` de disponible depuis Secure World

→

- Utilisation d'un GPIO depuis le NW pour encadrer l'exécution de la TA
- Opérations avec des signatures EM caractéristiques
  → Fonction `MarkerLoop()`

Expérimentations d'attaques - Attaque TEE/TZ

Fonction `MarkerLoop()`

```
1  int MarkerLoop(unsigned char P1, unsigned char P2)
2  {
3    int i = 0;
4    int j = 0;
5    uint8_t Buffer[2048] = {0};
6    size_t randomLen[1] = {0};
7    int tmpVar = 1; //to avoid 0*N = 0...
8    tlApiResult_t tlRet = TLAPI_OK;
9
10   //RANDOM multiplication :
11   while(i<P1)
12   {
13      randomLen[0]= P2;
14      tlRet = tlApiRandomGenerateData(TLAPI_ALG_SECURE_RANDOM,Buffer,randomLen);
15      if(TLAPI_OK != tlRet)
16      {
17         return tlRet;
18      }
19      for(j=0;j<P2;j++)
20      {
21         tmpVar *= Buffer[j];
22      }
23      i++;
24   }
25   return tlRet;
26 }
27
```

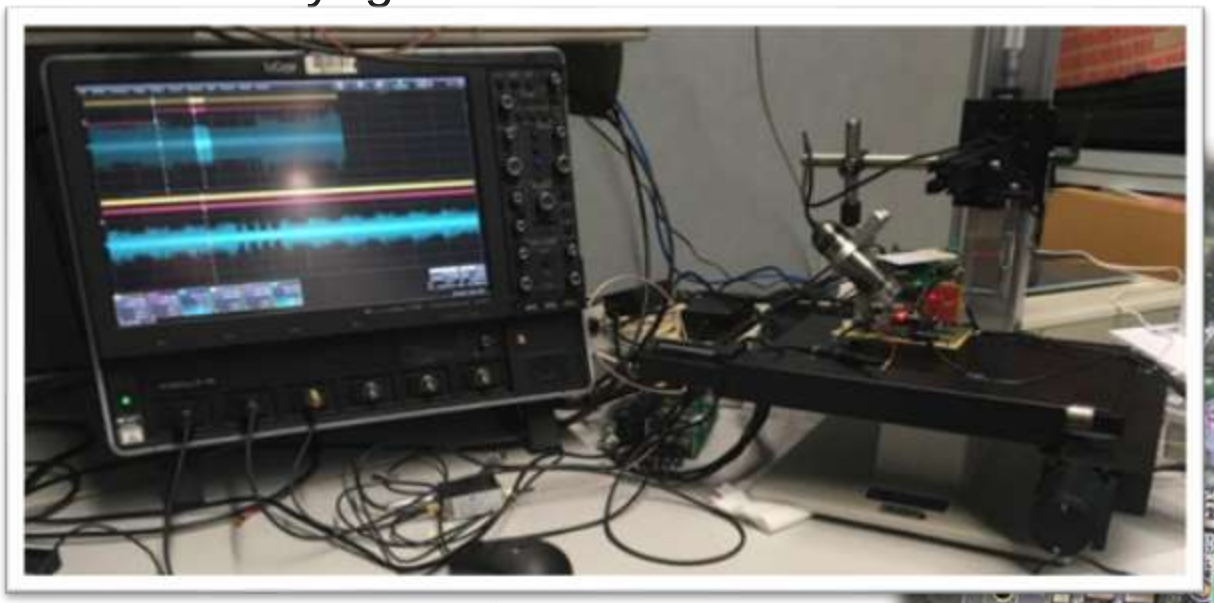
Expérimentations d'attaques - Attaque TEE/TZ

Analyse *Side-channel*

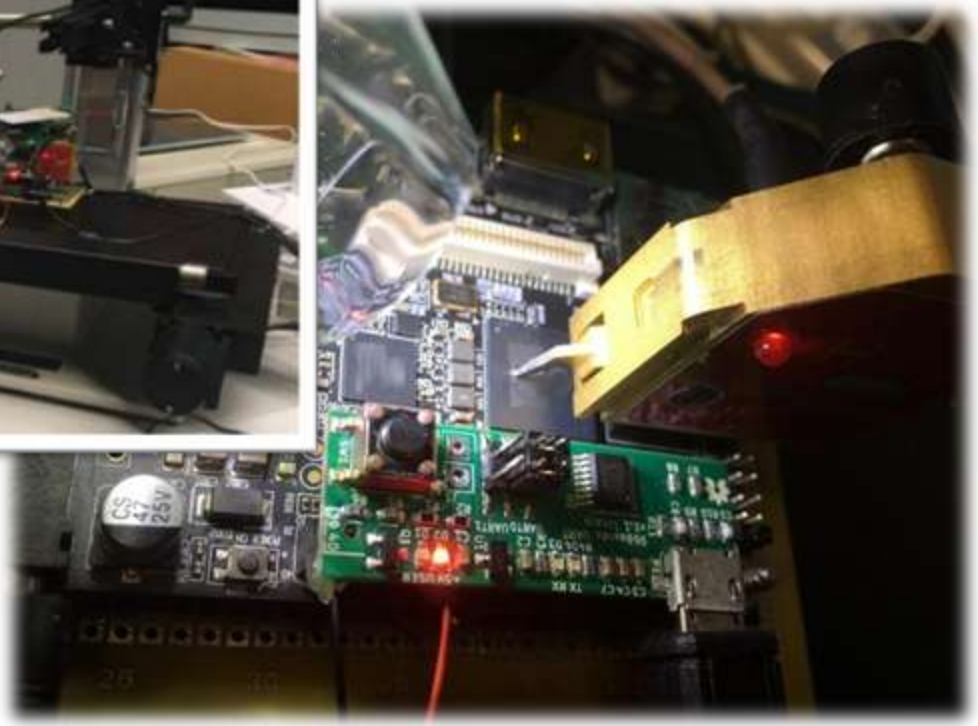
Expérimentations d'attaques - Attaque TEE/TZ

Analyse *Side-channel*

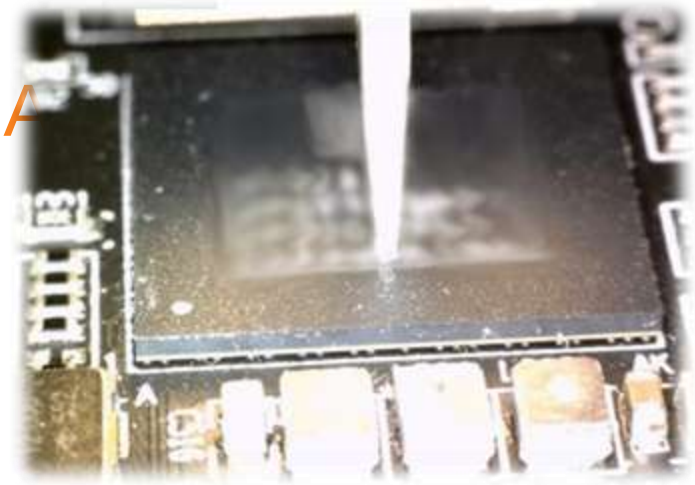
- Exécute `MarkerLoop(P1,P2)` avec différents paramètres `P1`
- Balayage de toute la surface



- Banc d'analyse EM -



Expérimentations d'attaques - A



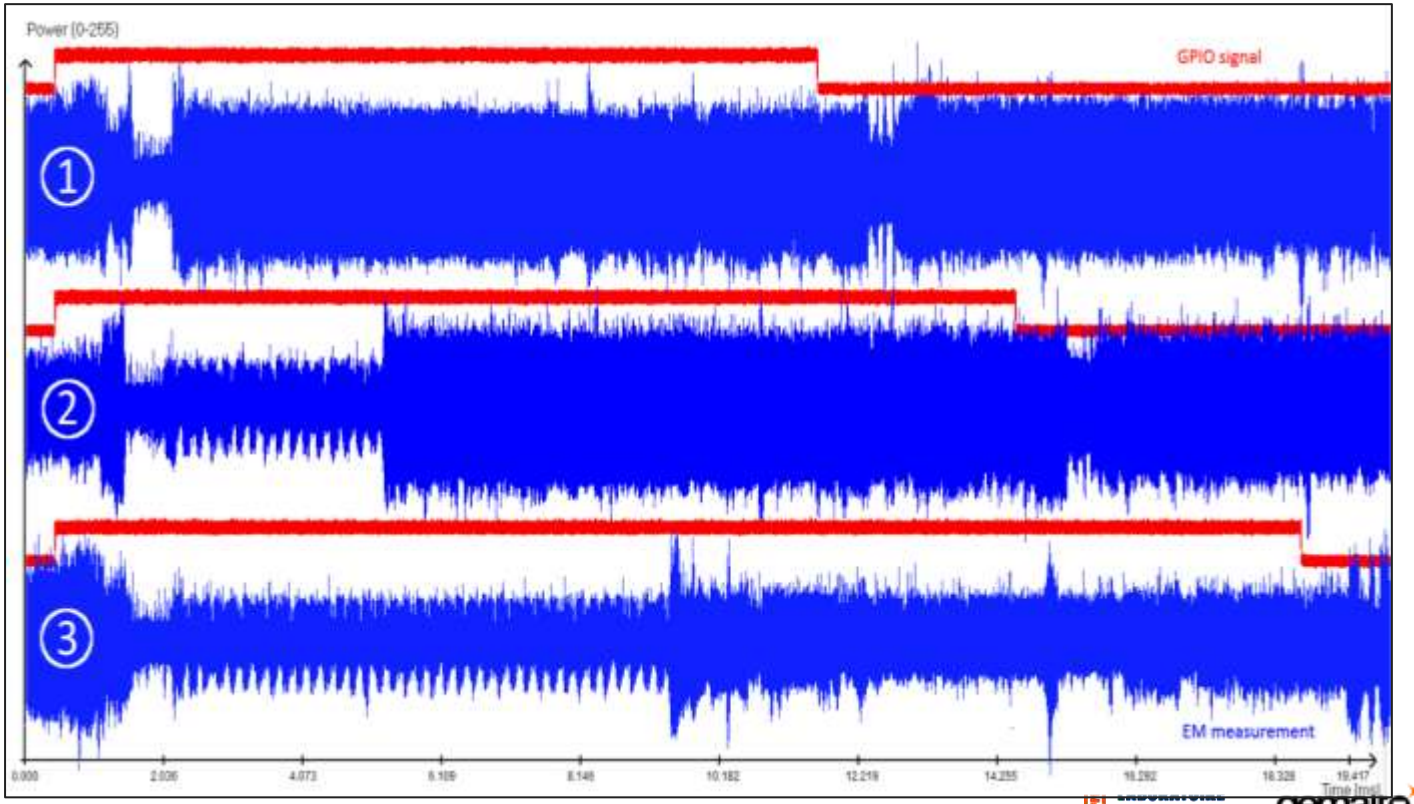
Analyse *Side-channel*

- Exécute `MarkerLoop(P1,P2)` avec différents paramètres `P1`
- Signaux caractéristiques

①: `P1` = 0, `P2` = 1

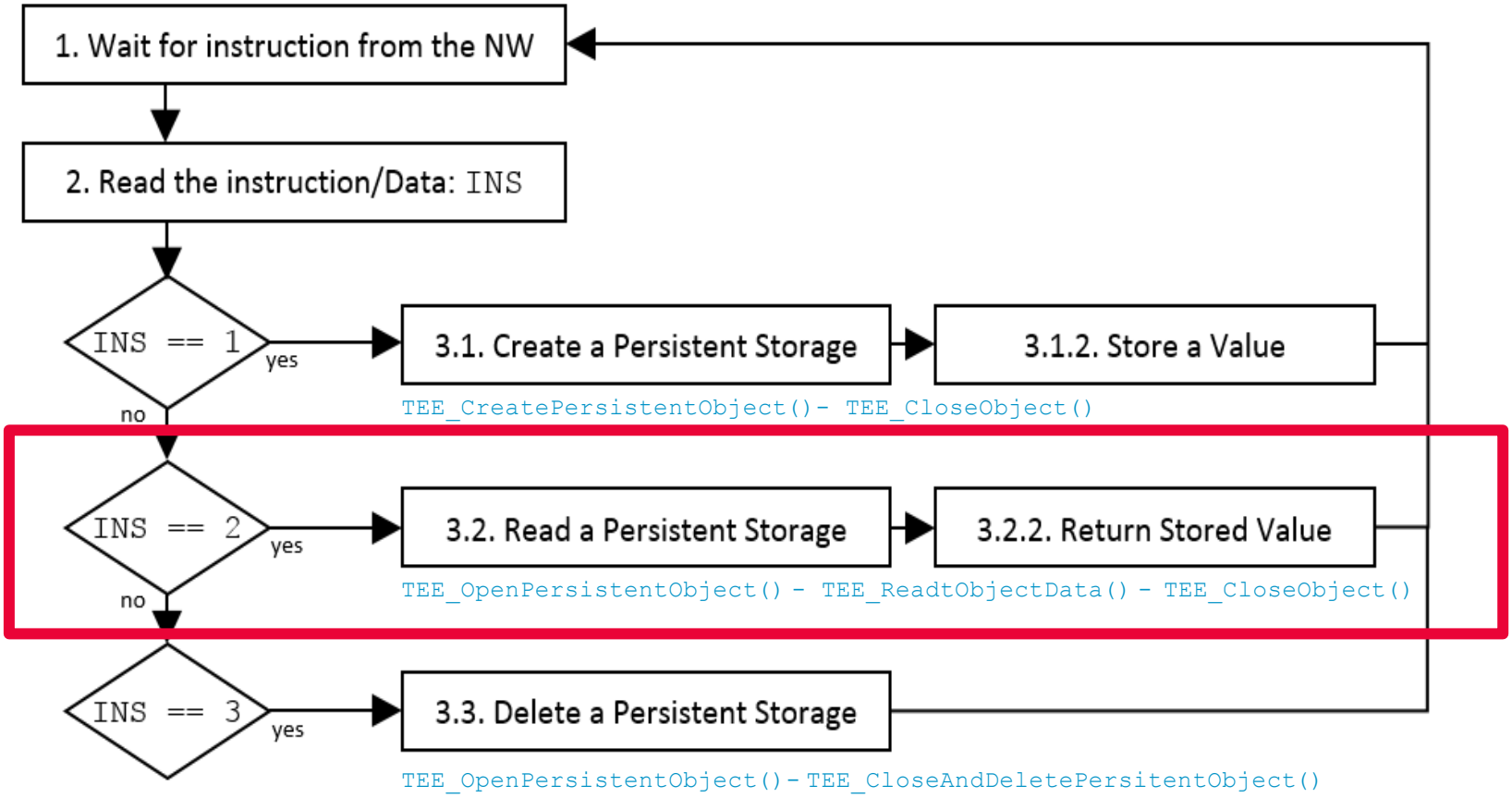
②: `P1` = 10, `P2` = 1

③: `P1` = 25, `P2` = 1



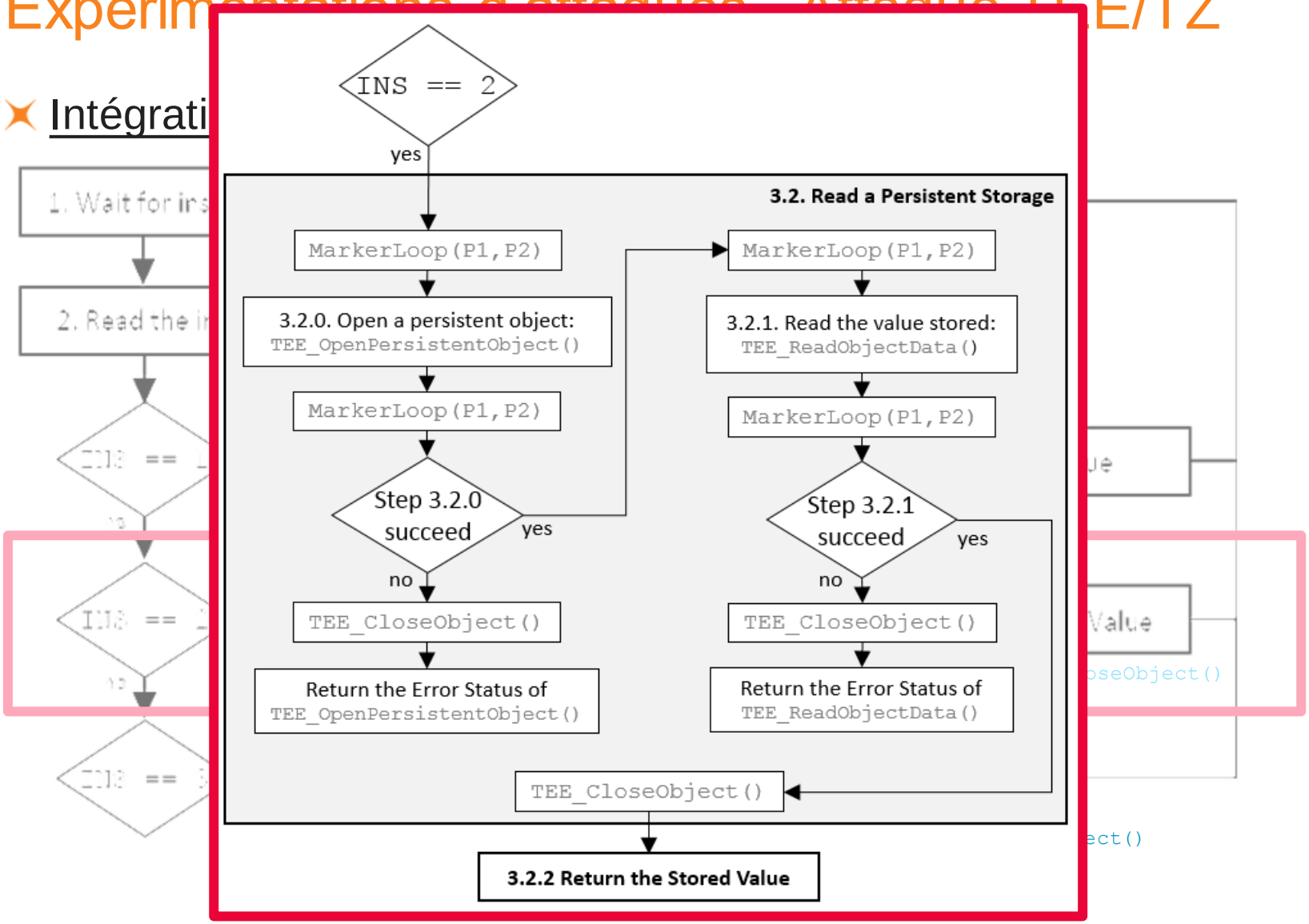
Expérimentations d'attaques - Attaque TEE/TZ

- Intégration de `MarkerLoop()` dans la partie ciblée:



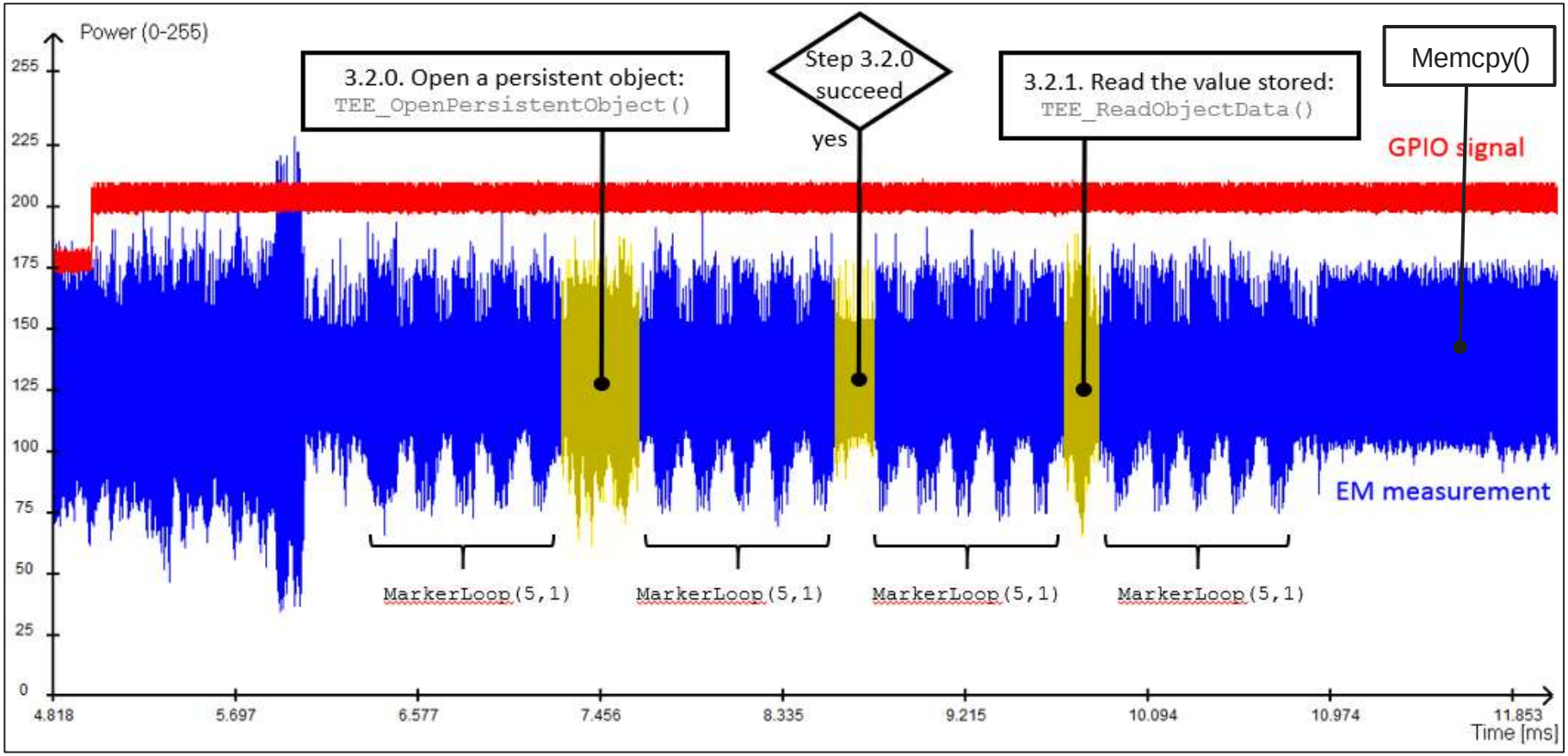
Expérimentations d'attaques – Attaque TEE/TZ

- Intégrati



Expérimentations d'attaques - Attaque TEE/TZ

Analyse *Side-channel*



Expérimentations d'attaques - Attaque TEE/TZ

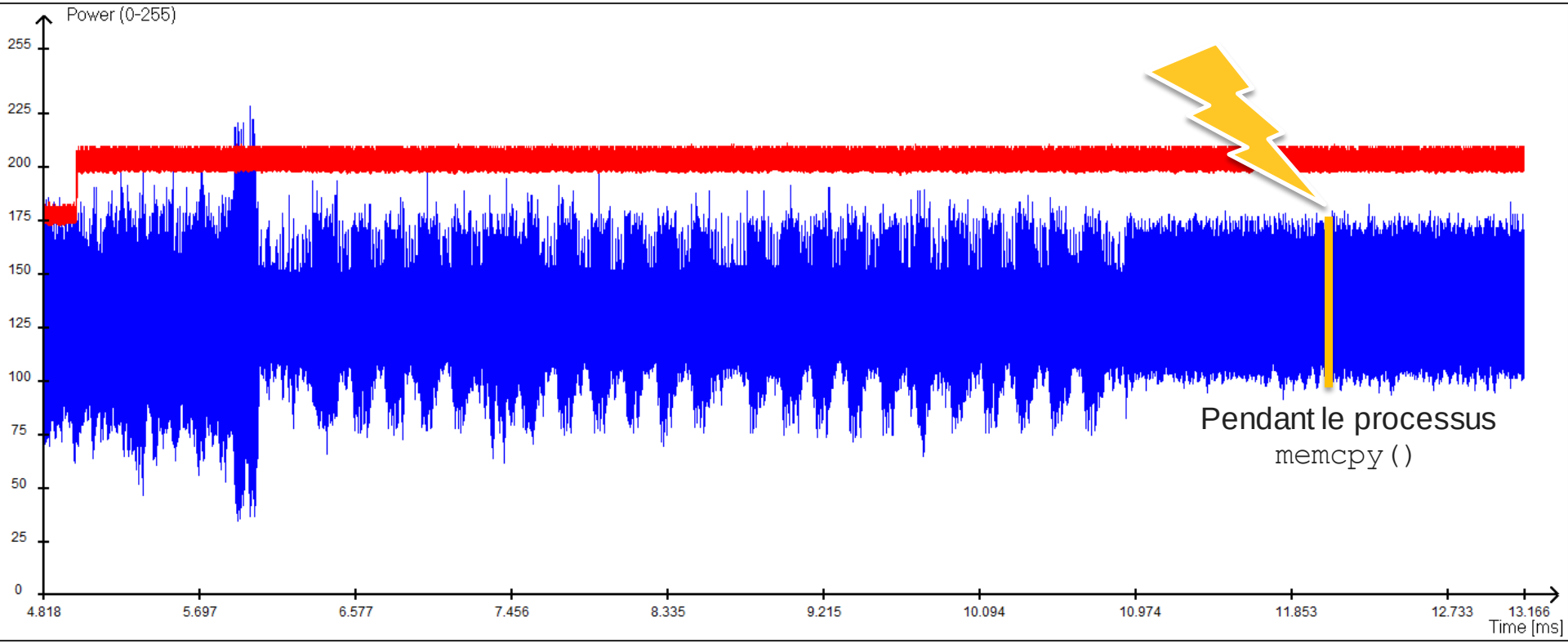
Détection des points sensibles

Expérimentations d'attaques - Attaque TEE/TZ

Détection des points sensibles

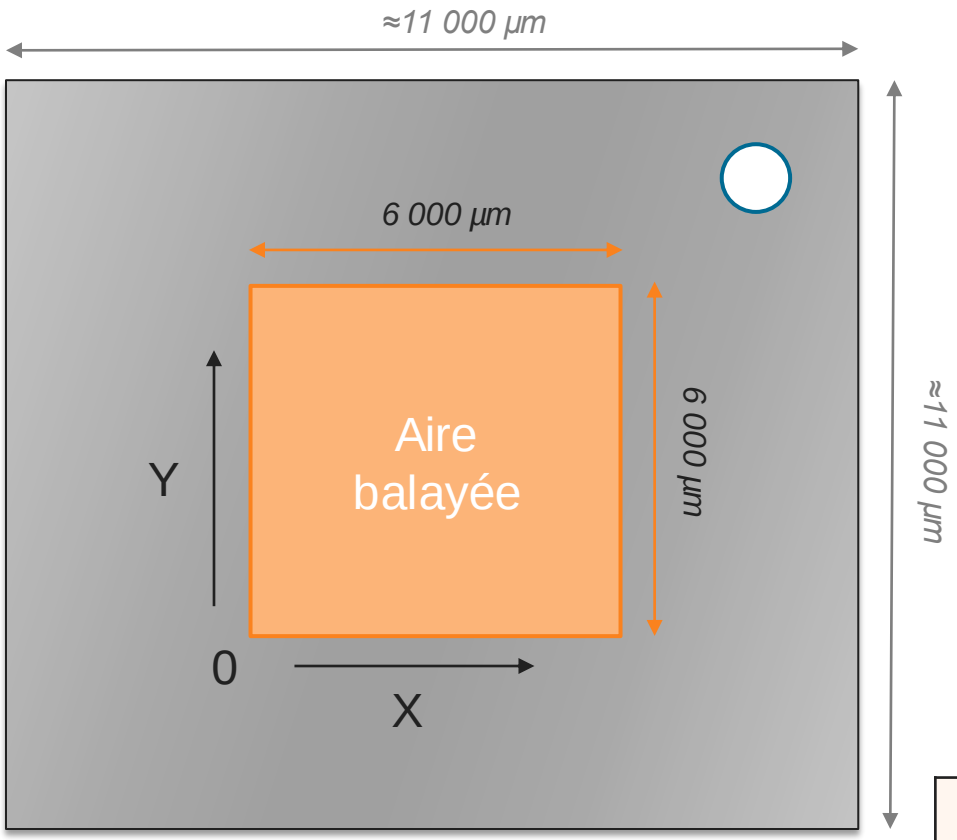
- Paramètres des impulsions EM (voltage, durée, instant)

(400V, 6ns)



Expérimentations d'attaques - Attaque TEE/TZ

Détection des points sensibles



Matrice XY : (101,101)
Nb d'essai: 10

Nb Total d'essai: 102010

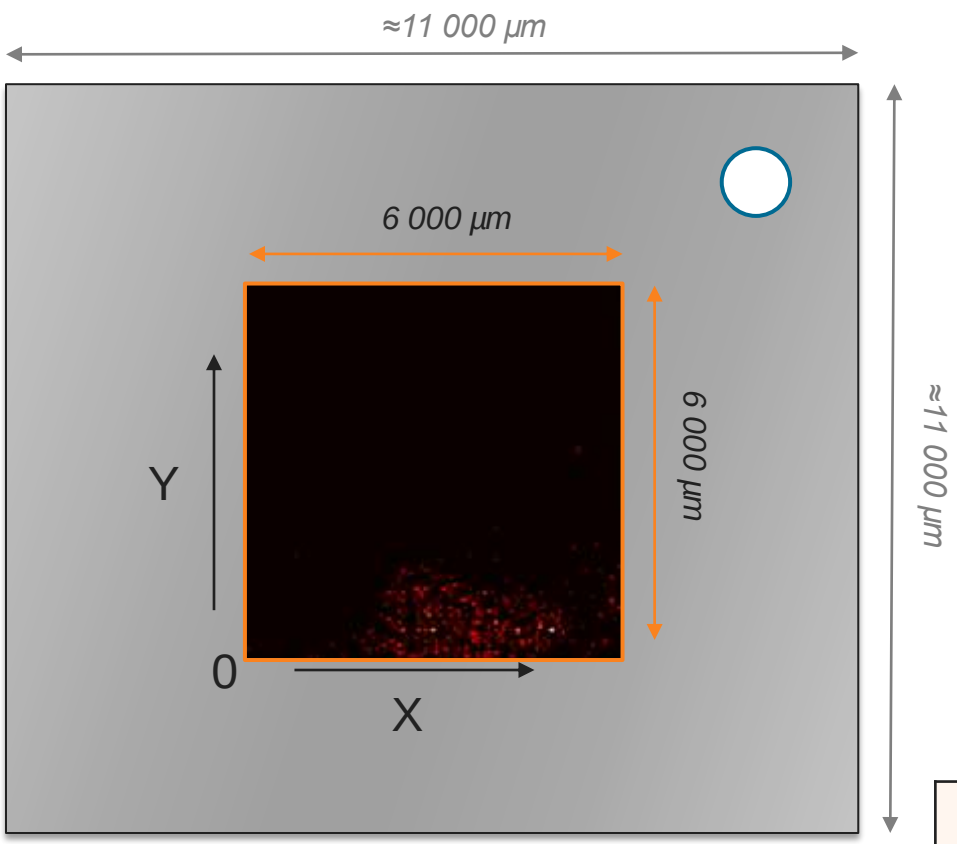
Balayage: Δx = 60µm , Δy = 60µm,
10 injections par point spatial

Expérimentations d'attaques - At

Nb total d'essai: 102010

Nb de fin correct: 100030 (98%)
Nb de Card Is Mute: 1980 (2%)
Nb de DATA erronées: 0 (0%)

Détection des points sensibles



Balayage: Δx = 60µm , Δy = 60µm,
10 injections par point spatial

Matrice XY : (101,101)
Nb d'essai: 10

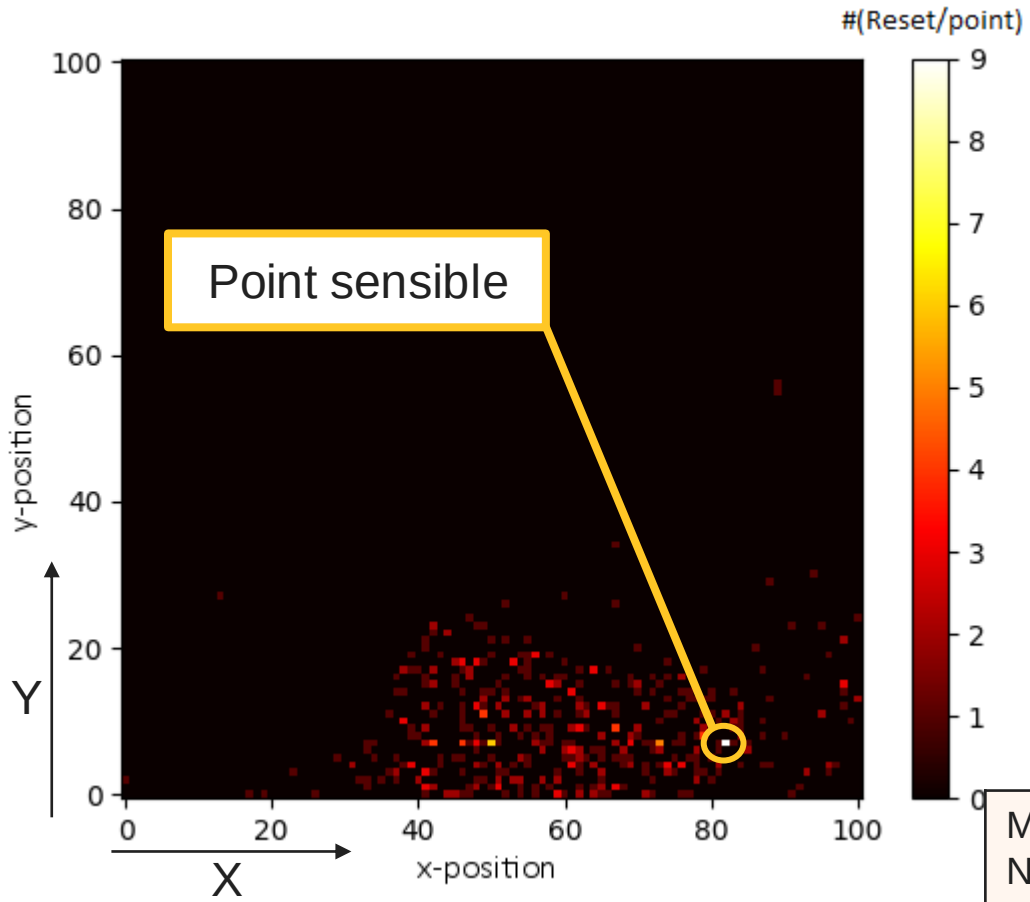
Nb Total d'essai: 102010

Expérimentations d'attaques - At

Nb total d'essai: 102010

Nb de fin correct: 100030 (98%)
Nb de Card Is Mute: 1980 (2%)
Nb de DATA erronées: 0 (0%)

Détection des points sensibles



Balayage: Δx = 60µm , Δy = 60µm,
10 injections par point spatial

Matrice XY : (101,101)
Nb d'essai: 10

Nb Total d'essai: 102010

Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

- Création de **2 TAs** différentes avec stockage sécurisé de **4096** octets:

TA_1	
ID	data
0x00	[0x0, 0x0, 0x0, … , 0x0]
0x01	[0x1, 0x1, 0x1, … , 0x1]
...	
0x80	[0x80, 0x80, 0x80, … , 0x80]
...	
0xFF	[0xFF, 0xFF, 0xFF, … , 0xFF]

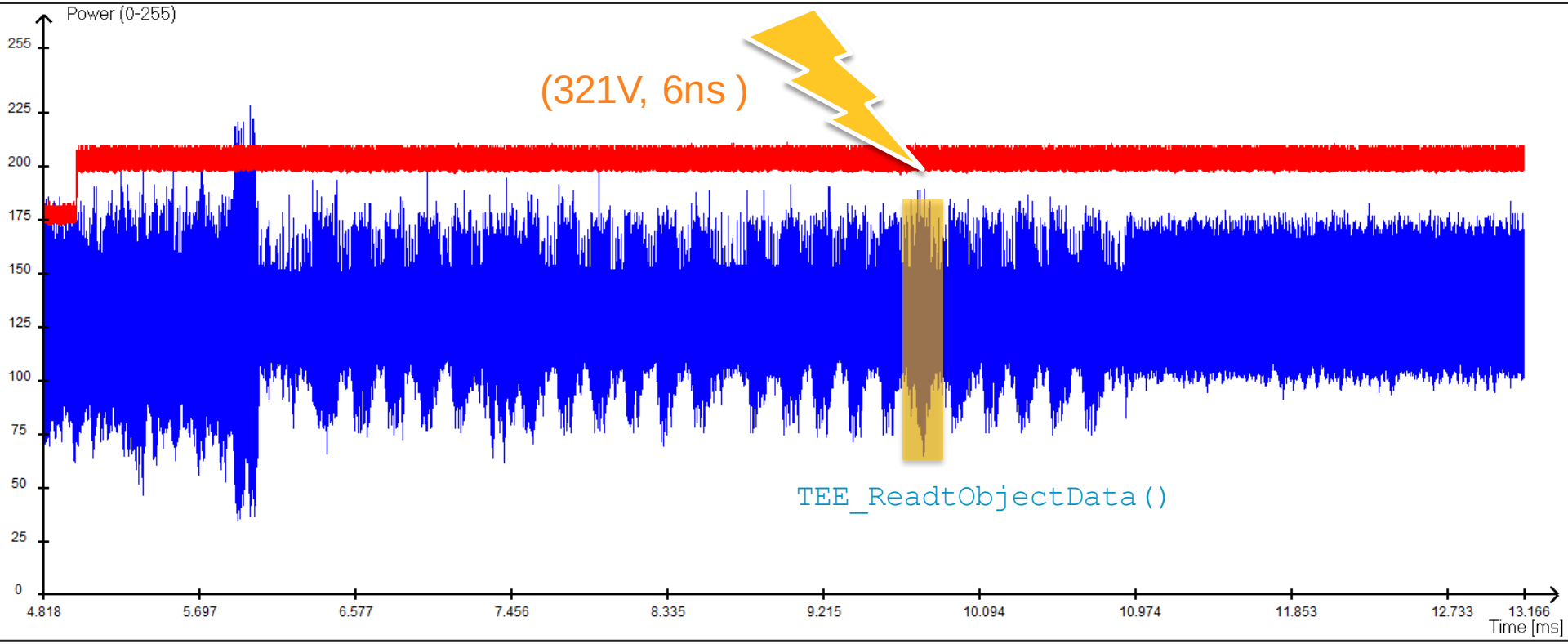
TA_2	
ID	data
0x80	[0xFA, 0xBE, 0xFA, … , 0xBE]

- Lecture des 4096 Octets d'un stockage sécurisé d'ID: `0x80` de la `TA_2`

Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

- Injection de perturbations durant l'API `TEE_ReadtObjectData()`



Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

- Répétition de la lecture et de l'injection de perturbation.
- Synthèse des résultats:

Erreurs retournées par les API:

Source : t-sdk-r11\t-sdk\TlSdk\Public\GPD_TEE_Internal_API\tee_error.h

```
22 00 : TEE_OpenPersitentObject()
23 00 : TEE_ReadObjectData()
```

```
22 00 F0 10 00 01 (TEE_ERROR_CORRUPT_OBJECT) .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 FF FF 00 03 (TEE_ERROR_ACCESS_CONFLICT)  .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 FF FF 00 06 (TEE_ERROR_BAD_PARAMETERS) .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 FF FF 00 08 (TEE_ERROR_ITEM_NOT_FOUNDS).. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 FF FF 00 0C (TEE_ERROR_OUT_OF_MEMORY) .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 00 01 7F 7C (status not found).. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
22 00 00 FF C6 85 (status not found).. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00

23 00 FF FF 00 06 (TEE_ERROR_BAD_PARAMETERS) .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. | 90 00
```

Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

Erreurs sur les donnés (extraits):

Réference :

```
FA BE FA BE FA BE FA BE FA BE FA BE [... 4096 Bytes ...] BE FA BE FA BE FA BE FA BE FA BE FA BE | 90 00
```

Fautes:

→ Données mises à '0'

```
FA BE [...] FA BE [64-bytes set to '0'] FA BE [...] FA BE [64-bytes set to '0'] FA BE [...] BE FA BE | 90 00
FA BE FA [...] BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE [3072-bytes set to '0'] | 90 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [...] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | 90 00
```

→ Mots de 32 bit modifiés

```
FA BE [...] FA BE FA BE FA BE FA BE FA BE 58 F8 00 00 BE FA BE FA BE FA BE FA BE FA[...] FA BE | 90 00
FA BE [...] FA BE FA BE FA BE FA BE FA BE 58 FC 00 00 BE FA BE FA BE FA BE FA BE FA[...] FA BE | 90 00
FA BE [...] FA BE FA BE FA BE FA BE FA BE 58 00 01 00 BE FA BE FA BE FA BE FA BE FA[...] FA BE | 90 00
FA BE [...] FA BE FA BE FA BE FA BE FA BE 58 04 01 00 BE FA BE FA BE FA BE FA BE FA[...] FA BE | 90 00
```

→ Large bloc de donnée modifié avec la même valeur

```
FA BE FA BE FA BE FA BE FA BE FA BE FA [...] BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE | 90 00
00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA | 90 00
00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA | 90 00
00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA 00 FA FF FA | 90 00
00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA | 90 00
00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA 00 FA | 90 00
BE FA BE FA BE FA BE FA BE FA BE FA BE FA [...] BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA | 90 00
```

Expérimentations d'attaques - Attaque TEE/TZ

Perturbations de TA

Erreurs sur les données (extraits):

→ Données ?

```
FA BE FA BE FA BE FA BE FA BE FA BE FA [...] BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE
73 AB 63 28 96 A0 F9 16 56 DF 65 F2 AC AD 23 2B D1 D3 5C D8 1C FA 83 36 DD 74 D2 20 6B 0A 46 4F
A0 1B 76 0C 2F A8 27 DA 68 A5 45 B2 CD 2C D5 84 B4 07 BB 55 D2 AC 1E 41 46 A7 A7 74 95 0B BF 21
90 D4 44 0B 38 98 5B 97 38 AA AF 92 81 69 7D CB 3E 92 9B E6 FE 79 13 BC 42 F2 40 A4 D2 CE 2D B3
05 76 02 05 1A 98 5F 71 51 27 A5 C3 8E 5D 19 C3 B6 FD 5E 9F 01 00 01 00 0B 00 00 00 00 00 00 00
24 43 46 47 43 52 45 00 07 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FE FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FE 30 21 30 09 06 05 2B 0E
03 02 1A 05 00 04 14 30 2D 30 0D 06 09 60 86 48 01 65 03 04 02 04 05 00 04 1C 30 31 30 0D 06 09
60 86 48 01 65 03 04 02 01 05 00 04 20 30 41 30 0D 06 09 60 86 48 01 65 03 04 02 02 05 00 04 30
30 51 30 0D 06 09 60 86 48 01 65 03 04 02 03 05 00 04 40 00 02 00 00 00 03 00 00 00 05 00 00 00
07 00 00 00 0B 00 00 00 0D 00 00 00 11 00 00 00 98 2F 8A 42 91 44 37 71 CF FB C0 B5 A5 DB B5 E9
5B C2 56 39 F1 11 F1 59 A4 82 3F 92 D5 5E 1C AB 98 AA 07 D8 01 5B 83 12 BE 85 31 24 C3 7D 0C 55
74 5D BE 72 FE B1 DE 80 A7 06 DC 9B 74 F1 9B C1 C1 69 9B E4 86 47 BE EF C6 9D C1 0F CC A1 0C 24
6F 2C E9 2D AA 84 74 4A DC A9 B0 5C DA 88 F9 76 52 51 3E 98 6D C6 31 A8 C8 27 03 B0 C7 7F 59 BF
F3 0B E0 C6 47 91 A7 D5 51 63 CA 06 67 29 29 14 85 0A B7 27 38 21 1B 2E FC 6D 2C 4D 13 0D 38 53
54 73 0A 65 BB 0A 6A 76 2E C9 C2 81 85 2C 72 92 A1 E8 BF A2 4B 66 1A A8 70 8B 4B C2 A3 51 6C C7
19 E8 92 D1 24 06 99 D6 85 35 0E F4 70 A0 6A 10 16 C1 A4 19 08 6C 37 1E 4C 77 48 27 B5 BC B0 34
B3 0C 1C 39 4A AA D8 4E 4F CA 9C 5B F3 6F 2E 68 EE 82 8F 74 6F 63 A5 78 14 78 C8 84 08 02 C7 8C
FA FF BE 90 EB 6C 50 A4 F7 A3 F9 BE F2 78 71 C6 22 AE 28 D7 98 2F 8A 42 CD 65 EF 23 91 44 37 71
2F 3B 4D EC CF FB C0 B5 BC DB 89 81 A5 DB B5 E9 38 B5 48 F3 5B C2 56 39 19 D0 05 B6 F1 11 F1 59
9B 4F 19 AF A4 82 3F 92 18 81 6D DA D5 5E 1C AB 42 02 03 A3 98 AA 07 D8 BE 6F 70 45 01 5B 83 12
8C B2 E4 4E BE 85 31 24 E2 B4 FF D5 C3 7D 0C 55 6F 89 7B F2 74 5D BE 72 B1 96 16 3B FE B1 DE 80
35 12 C7 25 A7 06 DC 9B 94 26 69 CF 74 F1 9B C1 D2 4A F1 9E C1 69 9B E4 E3 25 4F 38 86 47 BE EF
FA BE FA BE FA BE FA BE FA BE FA BE FA [...] BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE FA BE   90 00
```

Expérimentations d'attaques - Attaque TEE/TZ

Bilan

- Perturbation non expliquées. Effet de cache ?
- TEE testé semble robuste face à notre approche.
- Besoin d'approfondir la compréhension des API :
  - `TEE_CreatePersistentObject()`
  - `TEE_OpenPersistentObject()`
  - `TEE_ReadObjectData()`

→ Rétroconception du source code / boîte blanche.

Références

Références

[1] ARM Security Technology. Building a secure system using trustzone technology. PRD29-GENC-009492C. Copyright c 2005-2009 ARM Limited. All rights reserved.

[2] Ordas, Sébastien, Ludovic Guillaume-Sage, and Philippe Maurine. "EM injection: Fault model and locality." *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2015 Workshop on*. IEEE, 2015.

[3] GlobalPlatform white paper, “The Trusted Execution Environment : Delivering Enhanced Security at a Lower Cost to the Mobile Market”, février 2011.

fabien.majeric@gemalto.com