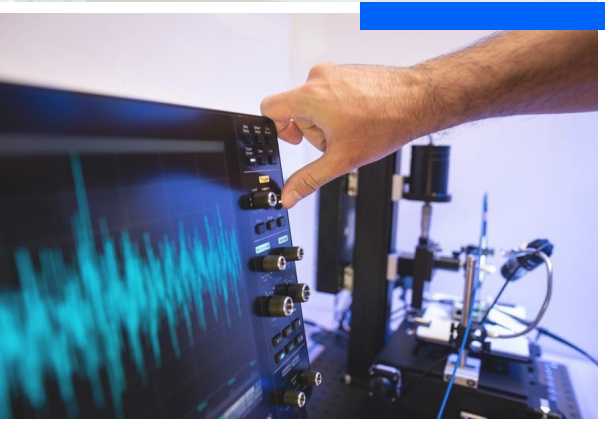
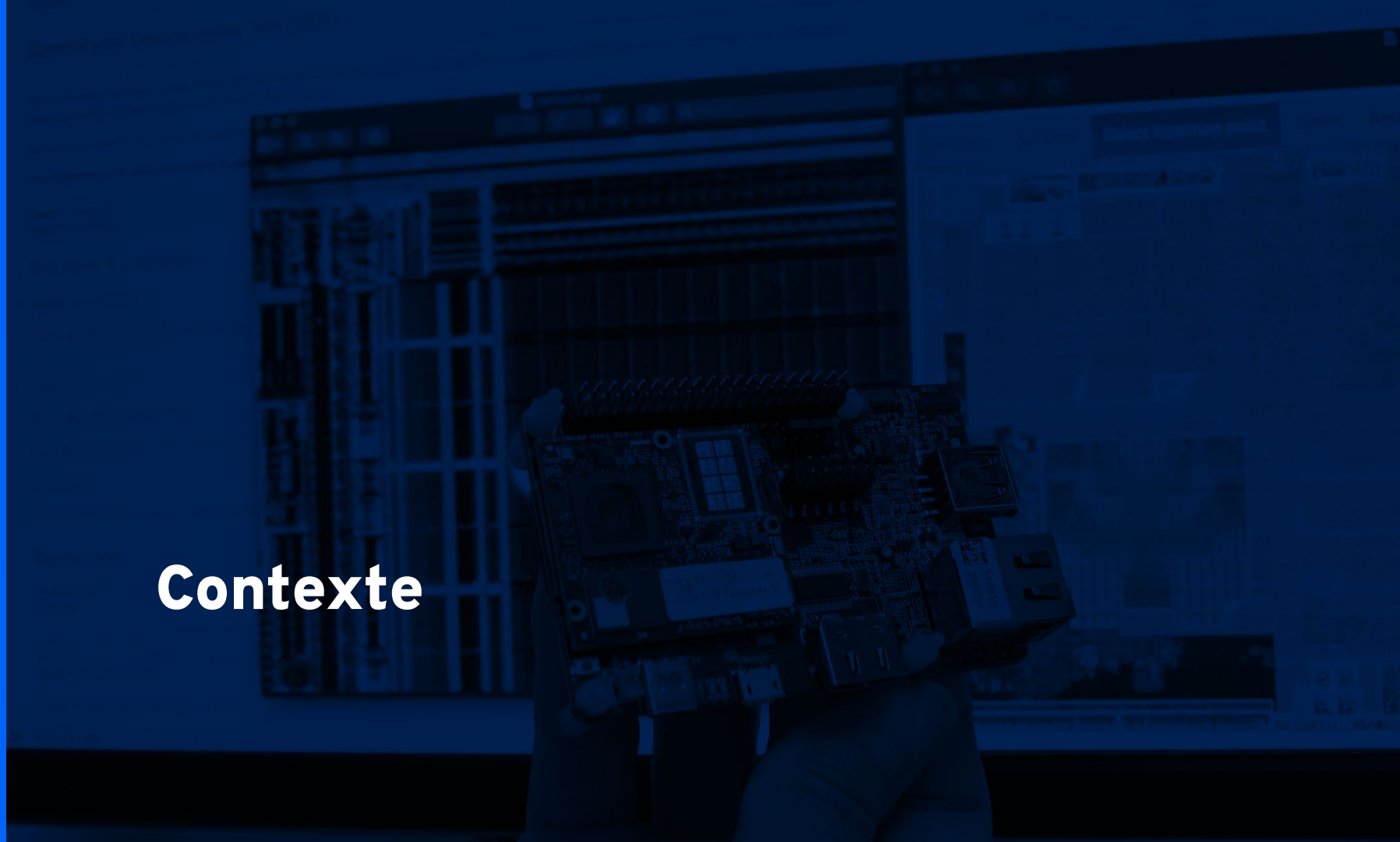


eShard Scaling Security Testing

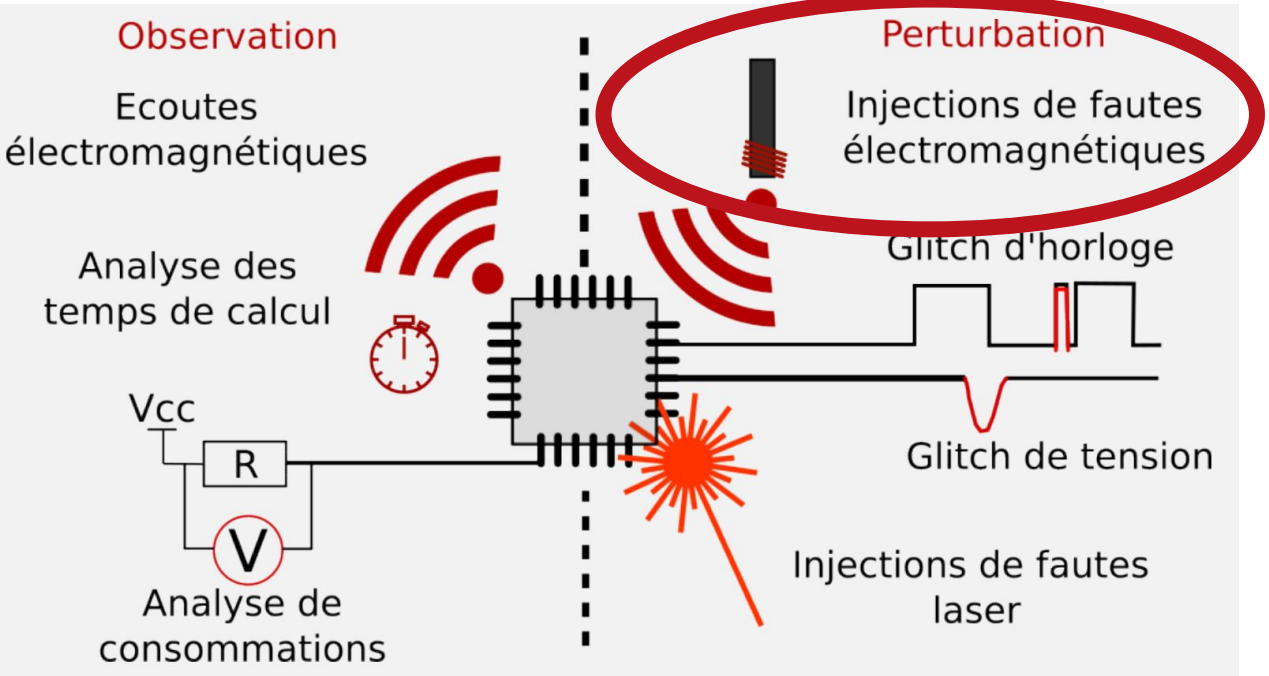
Injections de fautes électromagnétiques - D'une faute à une exploitation système

Journée thématique - Interférences ÉlectroMagnétiques Intentionnelles 26



Contexte

Attaques matérielles

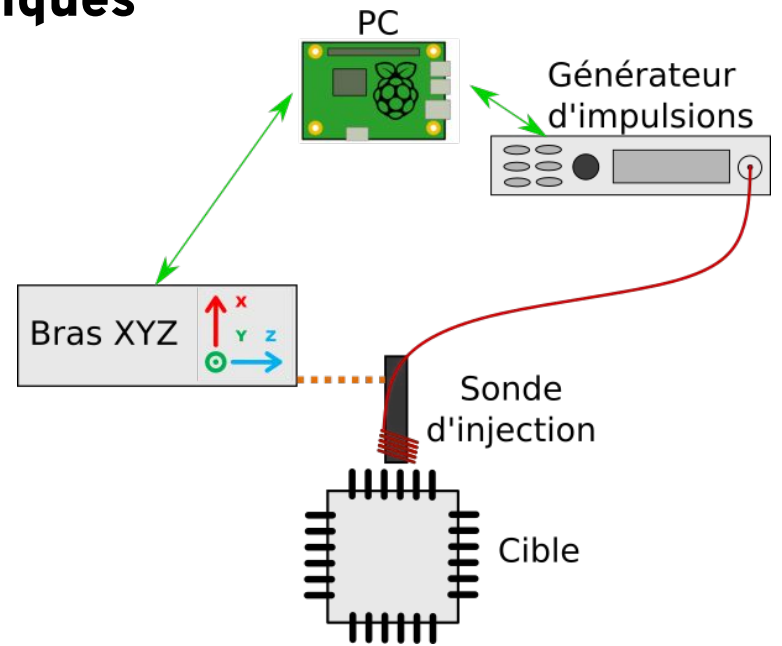


Injection de fautes électromagnétiques

Banc de test

2 éléments clés:

- Générateur d'impulsions
- Sonde



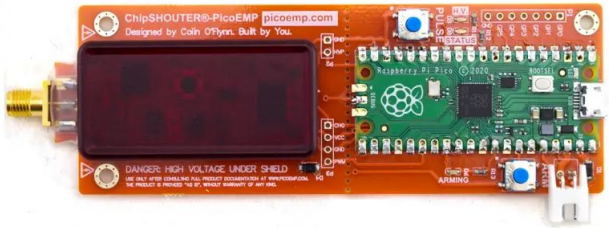
Générateur d’impulsions

Tension, largeur d’impulsion et fronts

Générateur d’impulsions de tensions:

Quelques valeurs courantes :

- 100V à quelques kV (souvent 750V/1000V) sous 50Ohm
- Largeur d’impulsion de 10 à 80ns
- Durée du front montant de 0,5ns à 5ns
- Simple ou double pulse -> 2 fautes temporellement séparées
- De 100€ à 30k€

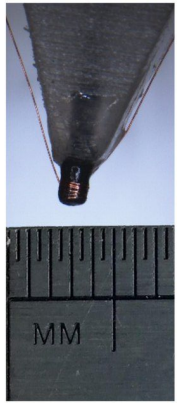


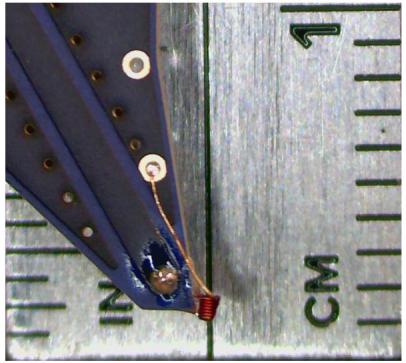
* Arrêt de la production d’Avtech en 2025

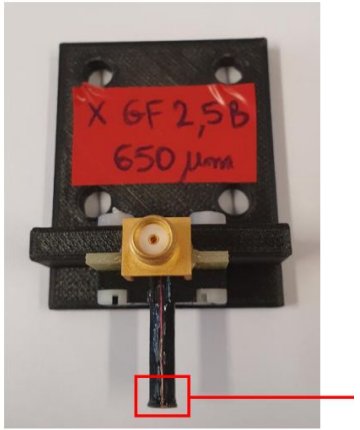
Sonde

Compromis entre résolution spatiale et puissance

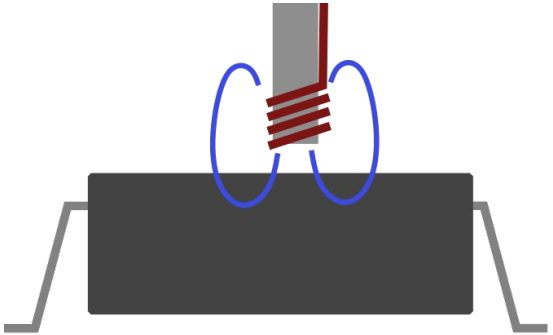
- Bobinage : 5 - 20 tours
- Diamètre : 4mm à 0,5mm
- Ferrite dans la majorité des cas







Couplage entre sonde et circuit

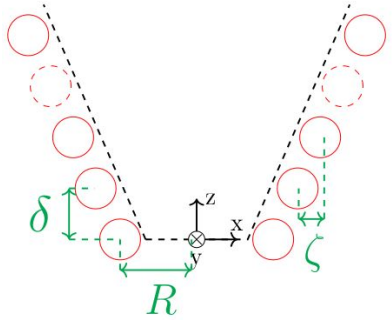


- EMFI : Couplage inductif
- Impulsion de tension -> Variation de courant -> Champ magnétique (Maxwell-Ampère)
- Variation de flux magnétique => Force électromotrice (Lenz-Faraday)
- Variations de tension internes => Faute

$$e = -\frac{d\Phi}{dt} \qquad \Phi = \iint_S B \cdot dS$$

Théorie des sondes

Quelques recommandations



Modélisation depuis Biot et Savart :

$$B_z(x,y,z) = \sum_{n=0}^{N-1} \frac{\mu_0 I}{4\pi} \int_0^{2\pi} \frac{((R+n*\zeta)^2 - (R+n*\zeta)*y\sin\theta - (R+n*\zeta)*x\cos\theta)d\theta}{[(x-(R+n*\zeta)*\cos\theta)^2 + (y-(R+n*\zeta)*\sin\theta)^2 + (z+n*\delta)^2]^{\frac{3}{2}}}$$

Règles pour maximiser la puissance transmise pour une tension donnée :

- Sonde cylindrique : Spires jointives
- Sonde conique : Angle entre 90° et 120°
- Fil de sonde fin (par ex entre 40 et 100µm)
- Présence d'une ferrite si le diamètre de la sonde le permet
- Rayon de sonde optimal pour une profondeur z : R = √2 * z
- Inductance inférieure à 100nH (Avtech)

2019. Arthur Beckers, et al. Design Considerations for EM Pulse Fault Injection. CARDIS
2019. Oualid Trabelsi, et al. Characterization at logical level of magnetic injection probes. EMC
2022: Julien Toulemont, Amélioration des résolutions spatiale et temporelle des plateformes d'analyse et d'injection électromagnétiques
2022. Clément Gaine, Evaluation et mitigation du risque d'attaque par injection de fautes électromagnétiques sur plateformes mobiles

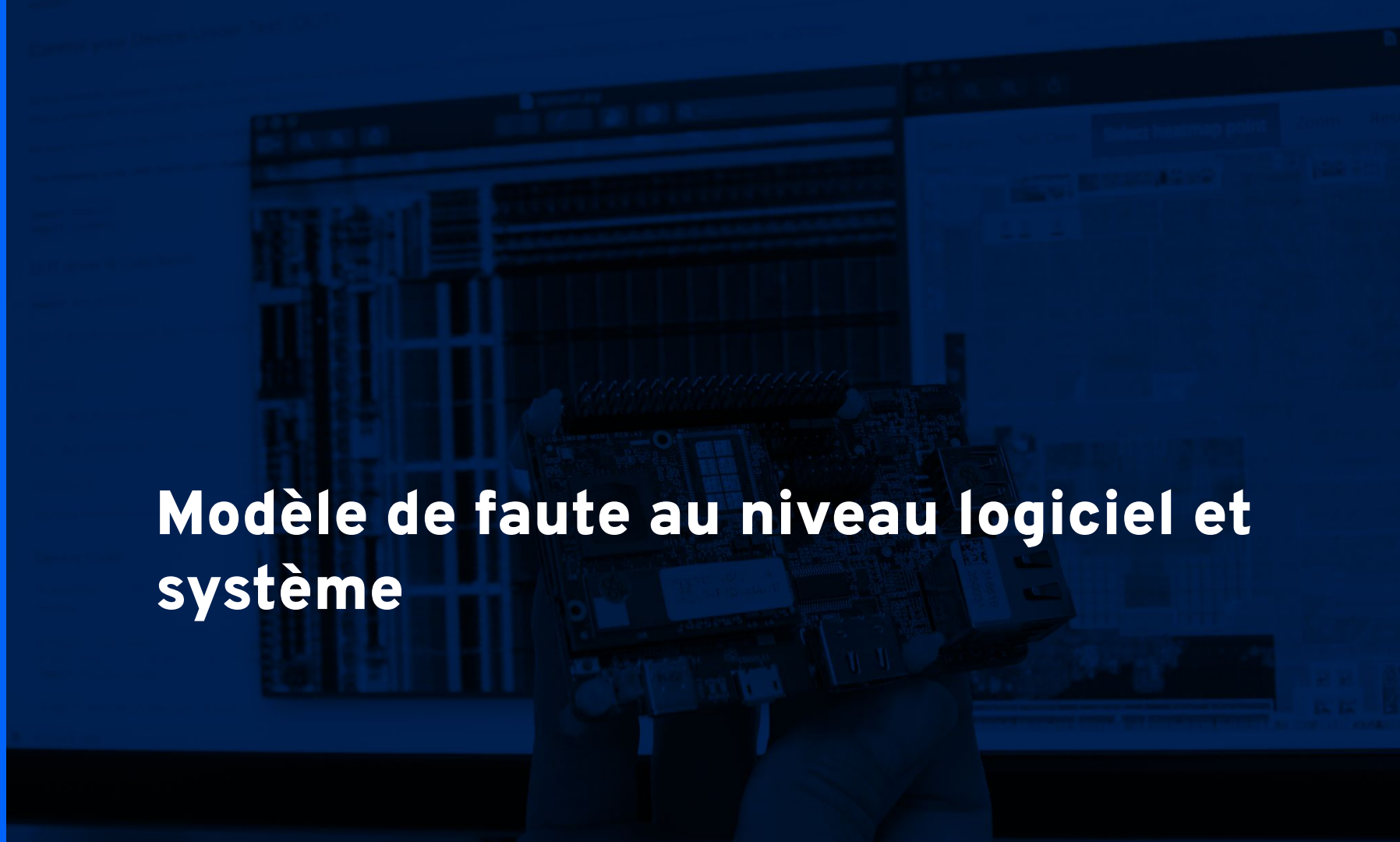
Immunité CEM et injection de fautes

Robustesse globale et exploitation d'une vulnérabilité locale

	Immunité CEM (EFT / Burst - IEC 61000-4-4)	EMFI
Objectif	Robustesse Le système ne doit *pas* planter. Conformité -> IEC	Sécurité Offensive ou Défensive Le système *doit* faillir, mais de manière contrôlée. Conformité -> Certification/CESTI
Cible	Le système complet	Une zone spécifique du Silicium (ALU, Mémoires, Registres, …)
Moyens	Salves aléatoires/périodiques (5kHz pendant 15ms, 100kHz pendant 0,75ms tous les 300ms)	Glitch unique synchronisé à quelques ns avec le CPU

Profil attaquants

Vecteur de Menace	Niveau d'Équipement	Cibles	Objectifs
Amateur (Low-Cost)	**Faible (< 200 €)** *Outils :* PicoEMP, Piézo modifié. *Sondes :* Faites main, peu précises	IoT grand public	Contournement de protections basiques Ajout de fonctionnalités
Industriel / Criminel	**Moyen (5k € - 50k €)** *Outils :* ChipSHOUTER, Oscilloscopes, Tables XY *Sondes :* Sondes de champ proche	Propriété Intellectuelle (IP) Clés de chiffrement (AES/ECC) Systèmes automobiles (ECU)	Vol de firmware Clonage de produits Contournement des modèles économiques (licences)
Laboratoire / Étatique	**Élevé (> 100k €)** *Outils :* Avtech, Générateur sur mesure, (Laser/Glitch) *Sondes :* Sondes haute résolution	Éléments Sécurisés (SE) Cartes à puce (Passeport/Bancaire)	Extraction de clés Contournement de niveaux EAL5+ / EAL6+



Modèle de faute au niveau logiciel et système

Quels effets au logiciel ? système ?

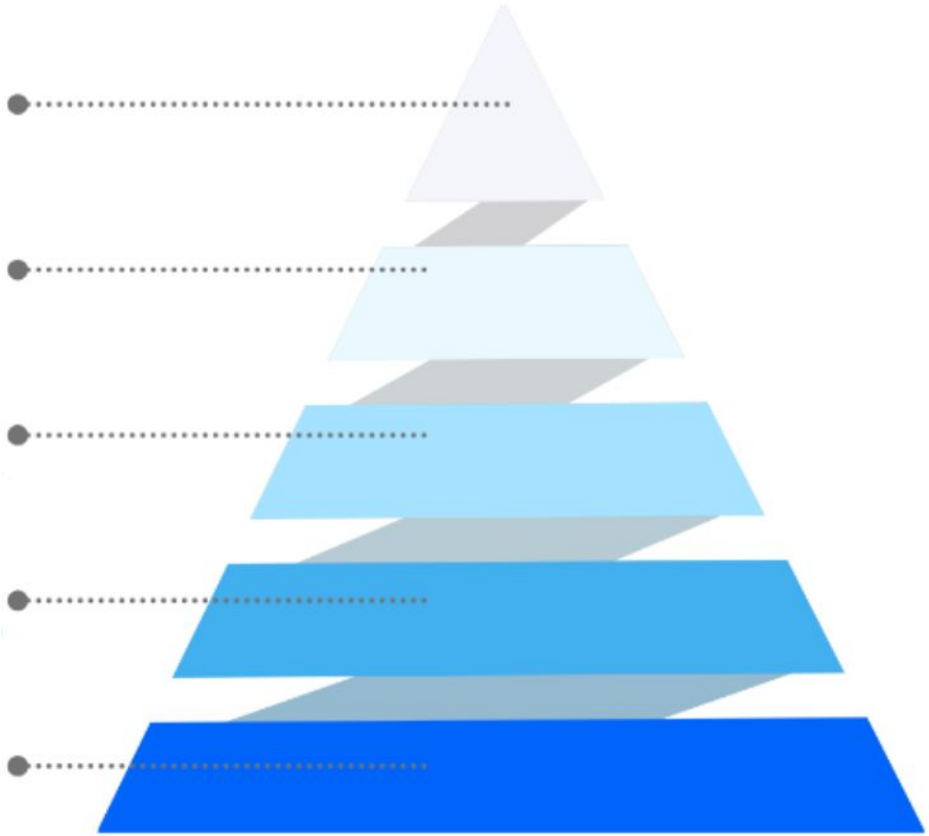
Physique
Variations de tensions électriques
Transistors

Logique
Logic Gates, Flips-flops

Binaire
Données (set/reset/flip)

Logiciel
Comportement du logiciel
Jeu d'instructions

Système
Comportement système



Modèle de fautes au niveau du jeu d'instructions

Quels effets sur les instructions ? sur les données ?

Sur microcontrôleur / processeur

Flot de contrôle :

- Saut d'instruction
- Corruption du PC

Données :

- Transfert de données depuis / vers mémoire cache/RAM/Flash
- Registres

- Corruption d'instructions

D'autres effets sur FPGA, ASIC, contrôleurs mémoires ...

Saut d'instruction

Modèle le plus observé

Instruction dont l'effet est assimilable à un NOP
- Opération corrompue => Ignorée par le pipeline (Fetch ou Decode*)
- Opération non observée => Par ex, modification d'un registre non utilisé dans le code

Exploitation au niveau logiciel:
- Modification de branchement:
  - BNE (branch if Not Equal)
  - Le branchement est "sauté" et l'exécution continue

```
if (verify_signature() != OK) {
        return ERROR;
}
```

```
BL verify_signature
CMP R0, #0
BNE return_error
```

- Sortie de boucle :
  - BNE (branch if Not Equal)
  - Tous les tours de l'AES ne sont pas exécutés

```
for (int i = 0; i < 10; i++) {
    add_round_key(state, key);
    sub_bytes(state);
    shift_rows(state);
    mix_columns(state); }
```

```
loop_start: BL aes_round
ADD R0, R0, #1
CMP R0, #10
BNE loop_start
```

Plusieurs sauts d'instructions peuvent être observés

*Moro et. al. Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller, FDTC 2013

Corruption du Program Counter (PC)

Moyen moins prédictible mais puissant

Modification directe du registre PC ou de l'adresse de retour dans la stack
- Saut arbitraire de code

Exploitation au niveau logiciel:
- Bypass d'un mécanisme de dead loop

```
while(1);
```

```
dead_loop: B dead_loop
```

Transfert de données depuis/vers mémoires cache/RAM/Flash

Destination constante ou statique

Modification de données lors de Load ou Store (depuis la Flash)

- Ecriture de 0xFF dans un registre lors d'un load

Modification de données en RAM :

- Bit flip dans une SRAM lors d'injection EM statique

Exploitation au niveau logiciel:

- Modification des bornes d'un compteur de boucle

```
for (int i = 0; i < 10; i++) {
    add_round_key(state, key);
    sub_bytes(state);
    shift_rows(state);
    mix_columns(state); }
```

```
loop_start: BL aes_round
ADD R0, R0, #1
CMP R0, #10
BNE loop_start
```

Moro et. al. Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller, FDTC 2013
O'Flynn, EMFI for Safety-Critical Testing of Automotive Systems, FDTC 2021

Registres

Bit set, bit reset, ...

Bit set ou bit reset dans les registres

Exploitation au niveau logiciel:

- Changement de la valeur d'un Flag

```
if (SecurityFlag == 1) {
   GrantAccess();
}
```

```
LDR R0, [SecurityFlag]
CMP R0, #1
BEQ GrantAccess
```

Ordas et. al., Evidence of a larger EM-induced fault model, CARDIS 2014

Corruption d'instruction

Concerne le flot d'exécution et les données

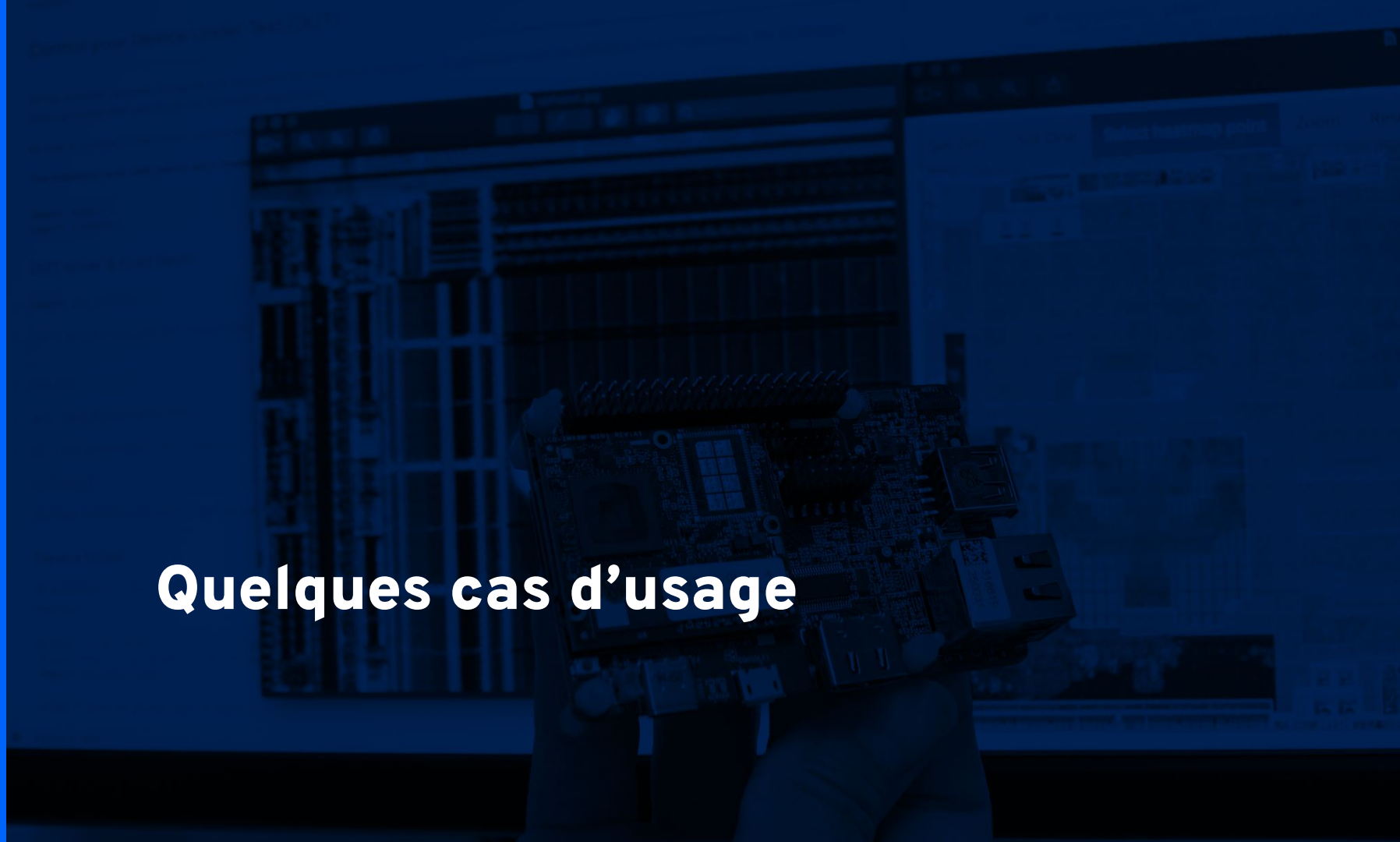
Modification d'une instruction
- Contrairement au saut d'instruction -> Executée
- Opcode
  - NOP => str r0, [r0, #0]
- Opérandes
  - add x0, x0, #0x1 => add x0, x0, #0x0

Exploitation au niveau logiciel:
- Modification de flot d'execution
  - Inversion d'un branchement

```
int verify_password(char* input) {
        if (check_crypto(input) != 0) {
                return ACCESS_DENIED;
        }
        return ACCESS_GRANTED;
}
```

```
BL check_crypto
CMP R0, #0
BNE acces_denied
B acces_granted
```

Moro et. al. Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller, FDTC 2013
Trouchkine et. al., Electromagnetic fault injection against a complex CPU, toward new micro-architectural fault models



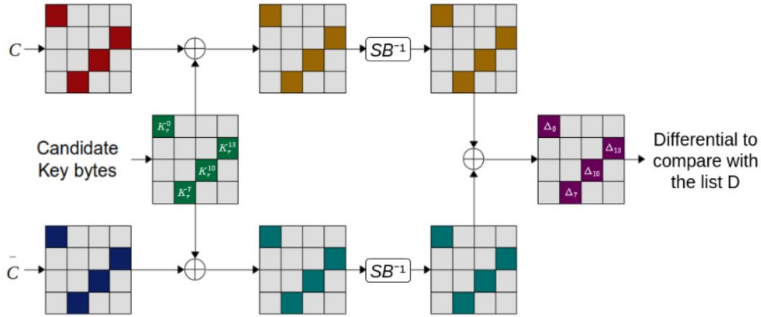
Quelques cas d’usage

Récupération de clés cryptographiques

RSA, DES, AES...

Réalisation d'une faute sur un octet avant le MixColumn du 9ème tour de l'AES-128bits

-> 2 paires de chiffré fauté/correct permettent de retrouver 4 octets de clé



- 8 Fautés seulement permettent de retrouver les 16 octets de clé d'un AES128

-> Nombreuses réalisations sur microcontrôleurs, AES matériel, FPGA, ...
-> Optimisé et étendu : 4 couples permettent de retrouver la clé d'un AES 256

Piret & Quisquater, A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad, CHES 2003

Elévation de privilèges

Extraction d'une clé sur portefeuille de cryptomonnaie/clé d'authentification

Portefeuille Trezor

- Communications entre un PC et le portefeuille via USB (Configuration, mise à jour, transfert, ...)
- Stack USB : Permet de récupérer un descripteur USB (identification de l'appareil)
  - Quantité maximale de donnée renvoyée limitée par un champ `wLength`
  - Injection EMFI permet un saut d'instruction et un bypass de cette limite
  - 64 Ko peuvent être extrait de la mémoire interne
    - Récupération de la seed

Similaire sur la clé d'authentification FIDO2 Solo Key

O'Flynn, MIN()imum Failure: EMFI Attacks against USB Stacks, WOOT 2019

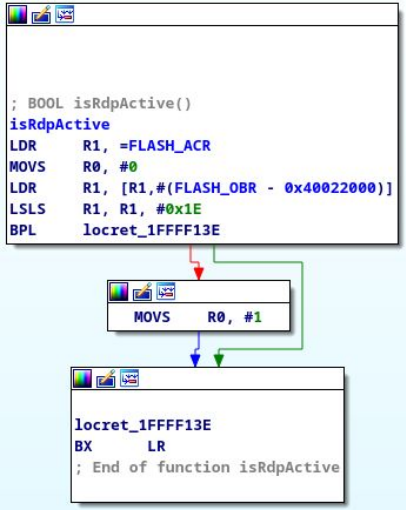
Extraction de firmware

Outils d'extraction de firmware

Firmware contient des informations importantes: Propriété intellectuelle, clés, ...
- Microcontrôleurs ont des mécanismes pour empêcher la lecture de la flash: Read Out Protection

Sur STM32F1/4 : 3 niveaux (0 protections, Protection mémoire, Protection mémoire + debug désactivé)
- Niveau 1 -> Niveau 0 efface la mémoire

Vérification de la protection RDP :
- Retour par défaut R0=0 (désactivé), R0=1 sinon.
  - Forcer un branchement à `locret_1FFFF123E` contourne la protection
  - Autorise la lecture de 255 octets de Flash par UART
  - Boucle afin de reconstruire le firmware



Voltage Glitch: Bozato et. al., Shaping the Glitch: Optimizing Voltage Fault Injection Attacks, CHES 2019

Modification du fonctionnement d'un produit

Reset d'une page mémoire via attaque combinée EMFI + Tearing

Puce avec mémoire simulant une EPROM (Uniquement décrémentation des données)
=> Objectif : Ecrire 0xFF dans une case mémoire

Étapes d'une écriture valide (d'après datasheet):
- Ecriture dans une mémoire tampon avec vérification de la légitimité des données (masquage)
- Transfert des données vers la case mémoire

Observations : EMFI permet de corrompre le contrôleur d'EEPROM
-> Retour de toutes les lectures avec les données 0x00

1er chemin d'attaque : Contourner le mécanisme de masquage ?
-> Mémoire tampon avec 0xFF
-> Ecriture échoue: Second masquage

2ème chemin d'attaque : Tromper la lecture de la configuration de la page
-> L'écriture reste masquée ...

Analyse des émanations électromagnétiques : Écriture en 2 parties
-> Effacement (à 0xFF) puis ré-écriture des données ?

3ème chemin d'attaque : Attaque combinée:
- EMFI pour modifier la lecture des permissions
- Tearing pour modifier le mécanisme d'écriture
  - Déconnexion de la puce après l'effacement et avant l'écriture

Attaques sur System-on-a-Chip

Elévation de privilèges - sudo sur SoC

Les défis d'une attaque sur SoC vs Microcontrôleur:

- Fréquences de fonctionnements élevées (>1GHz)
  - Nécessite une forte résolution temporelle, inférieur à la nanoseconde
- Couches logicielles complexes (OS, ...)
  - Ajout de sources de désynchronisation temporelle (Gigue)
- Grande surface de silicium avec un petit noeud technologique
  - Zone d'analyse plus importante avec une résolution spatiale plus faible
- Architecture matérielle complexe (cache mémoire, CPUs, ...)

Elévation de privilèges

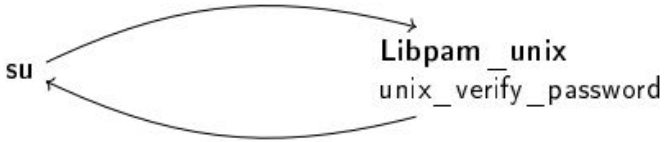
Elévation de privilèges - sudo sur SoC

Cible :
- SoC 4 coeurs A53, 64-bits, 28nm @1.2GHz

Point de départ :
- Phase d'apprentissage permettant d'obtenir des sauts d'instructions
  - Paramètres (XY, Tension)
  - Effets

Hypothèse : Accès utilisateur sans le mot de passe root
-> Commande `su` de Linux => Utilisateur à root

su :
- Authentification réussie -> Console root
- Echec -> Console utilisateur

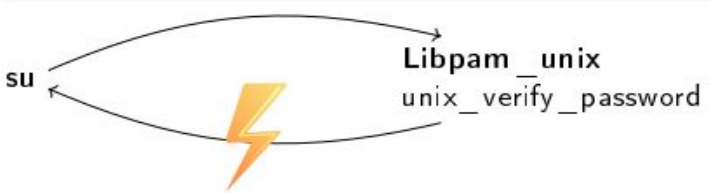


Elévation de privilèges

Elévation de privilèges - sudo sur SoC

1er chemin d'attaque : Inverser le test sur l'authentification de l'administrateur

- Protection contre les attaques par brute-force et écoutes (délai aléatoire)
  - Équivalent à cibler 1ns dans un intervalle de 1.5s



Elévation de privilèges

Elévation de privilèges - sudo sur SoC

2ème chemin d'attaque : Modifier le flot de contrôle de strcmp

- Vérification de la validité du mot de passe fourni
  - Comparaison octet par octet des hash du mot de passe entré et stocké

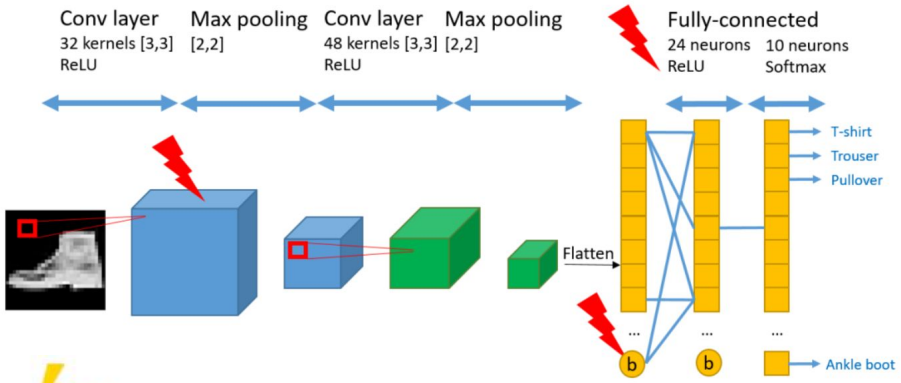
```
$6$wWxFc|tJdeOI05|KNO$IAAh|w8Th... -> Hash de "root"
$6$wWxFc|tJdeOI05|KNO$Uung|4U7s... -> Hash de "fail"
  mot 1  | mot 2 |  mot 3  | mot ...
```

  - Sortie de boucle forcée après 1ère ou 2nde vérification

=> Authentification réussie

Réseaux de neurones

Sujets de recherche



Réseaux de neurones = Grande surface de code

- Couche de convolution
  => Effet mémoire

- Biais
  => Chute de la précision (77% vers 35%)

- Fonction d'activation
  - ReLU : Ignorer la mise à zéro si la valeur est négative
  => Modification de la prédiction

=> Applications à la biométrie ?
- Reconnaissance faciale / empreinte permettant de déverrouiller un téléphone

Conclusion

- Non-invasif et accessible à faible coût (100€)
- Application sur de nombreuses cibles matérielles et logicielles
  - Du microcontrôleur simple au SoC complexe (Linux)
- Applications diverses :
  - Permet des récupérations de clés cryptographiques, élévations de privilèges, récupérations d'IP, clonage, ..
- Peut nécessiter d'être combiné à d'autres techniques : Voltage glitch, Side-channel

Thank you.
Any questions?

contact@eshard.com

www.eshard.com

/company/eshard

@eshard

France HQ

Bâtiment GIENAH
11 avenue de Canteranne
33600 Pessac, France

France R&D

7 rue Gaston de Flotte
13012 Marseille, France

Singapore

#03-07 Jit Poh Building
19 Keppel Road
Singapore 089058

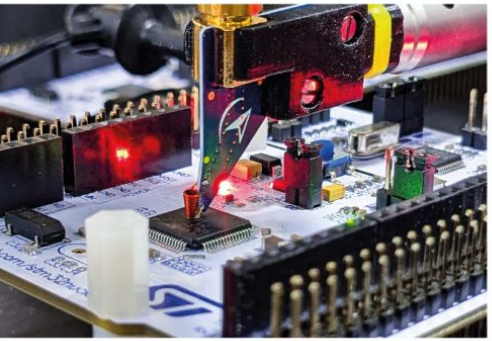
Annexes

eShard

PULSE GENERATOR

After benchmarking several models on the market, we felt limited in regard to our expectations. This is why we decided to design a specific EM pulse generator to match advanced Fault Injection applications.

High peak power, short rise time, fine pulse control and accurate triggering. This pulse generator, available both in Single and Dual mode enables effective EMFI on modern ICs.



Technical specifications

Pulse control	Programmable
Peak Voltage	up to 3000 V
Rise time	500 ps
Pulse duration	40 ns pulse duration at 90%
Repetition time	1 kHz
Trigger and delay	Internal or external
Trigger to 1st pulse delay	50ns - 100ms
1st pulse to 2nd pulse delay	1 ns to 1 µs
Delay adjustment step	Variable, down to 100 ps
Jitter	< 100 ps

EM Fault Injection Turnkey Solution