LES ATTAQUES PAR FAUTE

MESSOUS TILILI
ARABI WALID

PLAN DE TRAVAIL

- 1.Introduction

1.1 Circuits sécurisés

1.2 Techniques d'injections de fautes

- 2. Attaques par fautes

2.1 Types d'attaques

2.1 Exemple

- 3.Contre mesures
- 4.Conclusion

1.INTRODUCTION :

Les circuits sécurisés traitent des informations confidentielles et peuvent être soumis a divers types d'attaques visant à révéler le secret, c'est-à-dire la clef permettant de chiffrer et déchiffrer un message. L'une de ces attaques consiste à provoquer une

erreur de calcul. Par comparaison avec un calcul sans erreur, l'attaquant peut remonter jusqu'au secret, ou du moins une partie du secret. Ce type d'attaques nommé attaques en faute est référencé sous le terme générique de DFA pour Differential Fault Analysis.

Pour produire une attaque DFA, il faut d'une part manipuler le circuit de façon à provoquer une erreur exploitable et d'autre part définir la méthode qui, à partir d'un calcul « fauté », permet de calculer une partie du secret (la clef).

Les contre-mesures associées à ce type d'attaques consistent naturellement a révéler la présence d'une erreur en cours de calcul.

Le principe de ce type d'attaque repose sur le fait que l'attaquant peut perturber le fonctionnement normal d'un circuit pour en révéler les données confidentielles.

1.1 Circuit sécurisés :

La sécurité des systèmes est une contrainte de conception de plus en plus répandue et critique pour de nombreuses applications. Au-delà des problèmes généraux de sécurité informatique et des réseaux, de nouvelles menaces sont apparues assez récemment.

Il faut s'assurer que l'implantation est réalisée de manière à éviter des fuites d'informations sensibles. En raison de ces nouvelles menaces, la seule implantation dans un circuit d'un algorithme cryptographique standardisé, ce qu'on va appeler un circuit sécurisé, n'est plus suffisant pour assurer l'échange d'information sécurisé et confidentiel

Des mécanismes de sécurité supplémentaires doivent alors être implantés. Ils reposent sur différents principes: le secret de design et d'implémentation, les opérations de chiffrement des données confidentielles et les contre-mesures logicielles et matérielles mises en place pour la détection d'attaques.

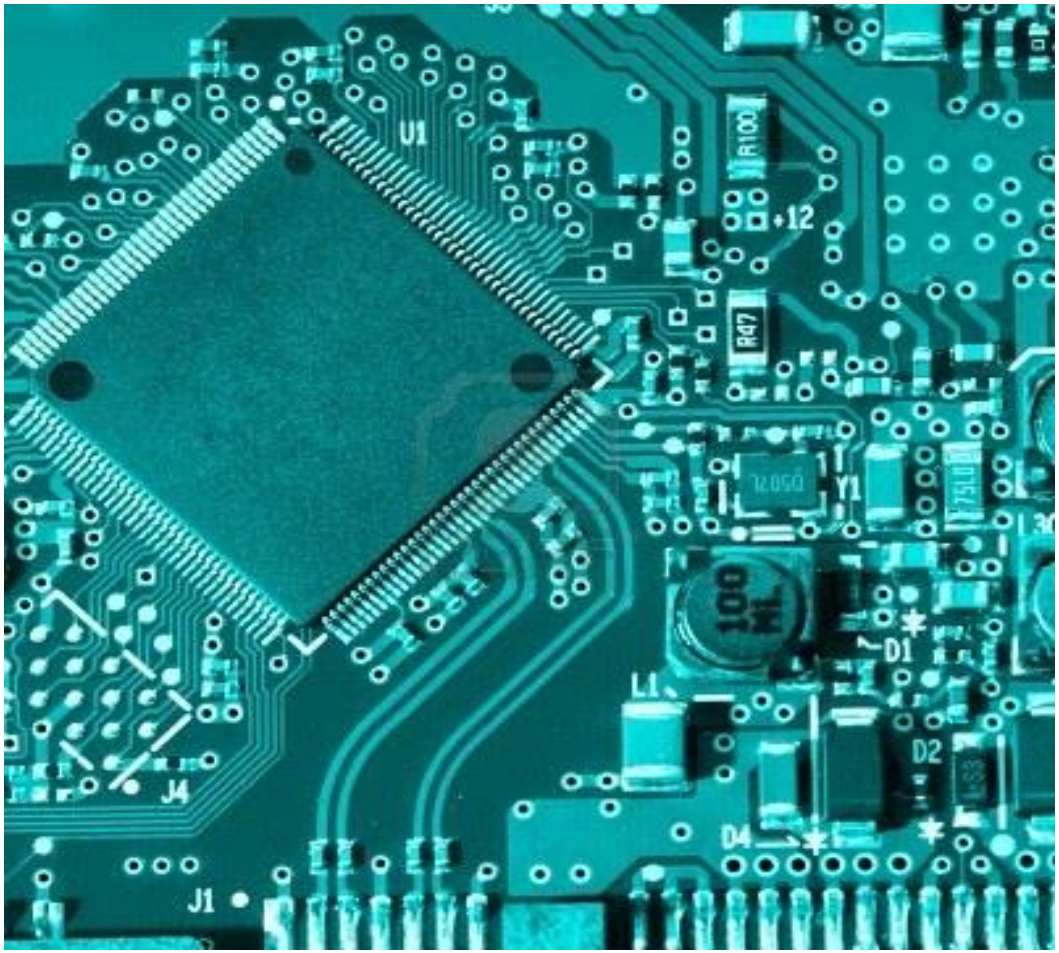


Figure 1. model d’un circuit

1.2 Techniques d'injection de fautes :

Certains des moyens d'attaques décrits ci-après sont très facilement exploitables si aucune contremesure n'est mise en œuvre. Elles sont extrêmement puissantes et présentent une menace réelle ce qui nous amène à en définir trois catégories :

- **Attaque invasives :** Le circuit est souvent détruit dans ce type d'attaques qui nécessitent d'abord une préparation de circuit en enlevant le boitier, ou reconstruire le circuit ou extraire des données secrètes.

- **Attaques semi-invasives:** Dans ce type d'attaques la puce n'est pas détruite mais le package contenant la puce est enlevé afin de pouvoir réaliser l'attaque et observer de plus près le comportement de la puce. Les attaques semi-invasives peuvent être effectuées en utilisant des outils tels que la lumière UV, les rayons X et d'autres sources de rayonnements ionisants, les lasers et les champs électromagnétiques Ils peuvent être utilisés individuellement ou en combinaison les uns avec les autres.

- **<u>Attaques non invasives :</u>** Les attaques non invasives sont considérées comme une menace réelle. En plus, elles nécessitent souvent beaucoup moins d'équipement que les attaques invasives. L'attaquant observe ici les paramètres externes ou les phénomènes physiques liés au fonctionnement de la puce et aux données traitées. L'attaquant peut ainsi en déduire des informations sur la clef.

  Il existe deux types d'attaques non invasives :

1) *Les attaques par canaux cachés.*
2) *Les attaques par fautes .*

LES ATTAQUES MATÉRIELLES

INVASIVES

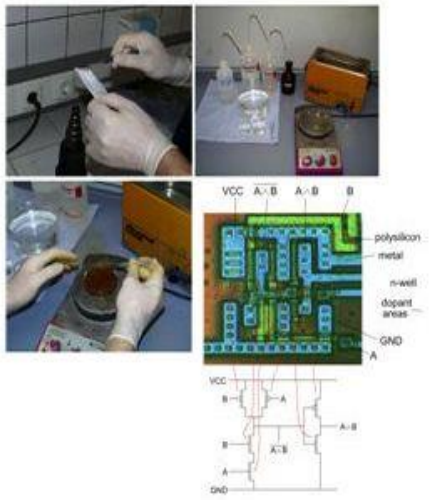
- **décapsulation** + contact CI
- **Ingénierie inverse** possible
- **Coût** de l'attaque **: élevé**

SEMI INVASIVES

- **décapsulation** sans contact CI
- Attaque par **laser** possible (amincis.)
- **Coût** de l'attaque : **moyen**

NON INVASIVES

- **sans décapsulation**
- difficiles à détecter (**mesures EM**)
- nécessitent beaucoup de mesures
- **Coût** de l'attaque : **réduit**



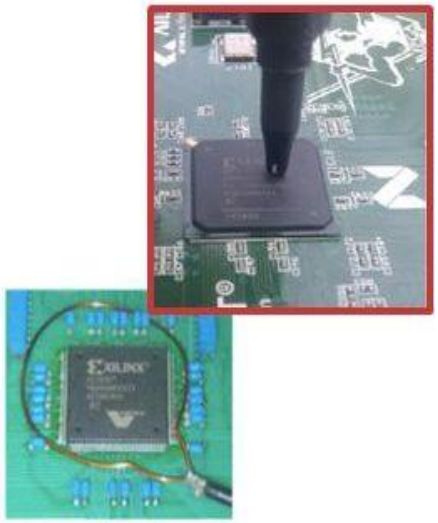
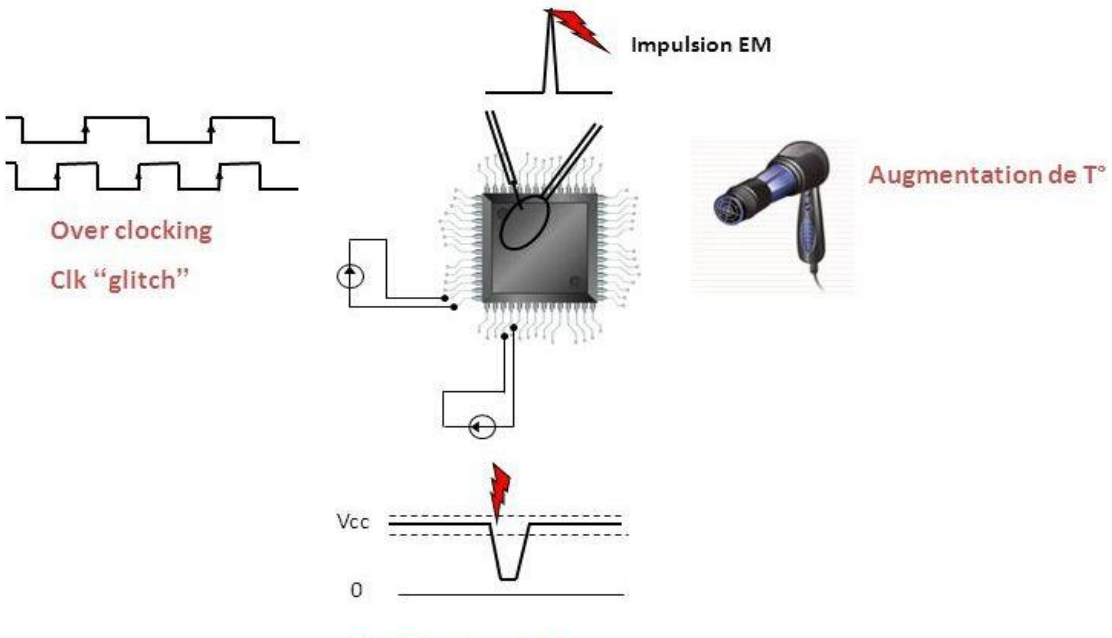


Figure 2. techniques d'injection de fautes

2. ATTAQUES PAR FAUTES :

Les implémentations logicielles (les séquences assembleurs, les registres, les variables …), peuvent être attaqués.

Les implémentations matérielles (les registres, les bascules, les mémoires …) , sont des potentiellement vulnérables aux fautes. Les vérifications logicielles et matérielles sont des points très sensibles aux fautes, par exemple la vérification du mot de passe, de code PIN ou la vérification de la signature.



2.1 Types d’attaques :

- **Attaques par perturbation de la tension d’alimentation**

  Tout circuit est conçu de façon à tolérer une certaine variation d'alimentation électrique. En dehors de cette marge de tolérance, le circuit ne fonctionne plus correctement. Un moyen de "fauter" un circuit consiste donc à perturber l’alimentation.

  Plus l’alimentation est faible et plus les temps de propagation sont importants. Lorsque les temps de propagation d’un bloc combinatoire deviennent supérieurs a la période d’horloge, la valeur capturée en sortie du bloc dans l’élément de mémorisation (registre) est erronée.

  Un moyen de contrôler l’instant d’injection est l’utilisation de pics de tensions.

- **Attaques par perturbation de l'horloge :**

Dans ce cas c'est le signal d'horloge qui est manipulé. Une augmentation de la fréquence permet de capturer des valeurs erronées dans les registres.

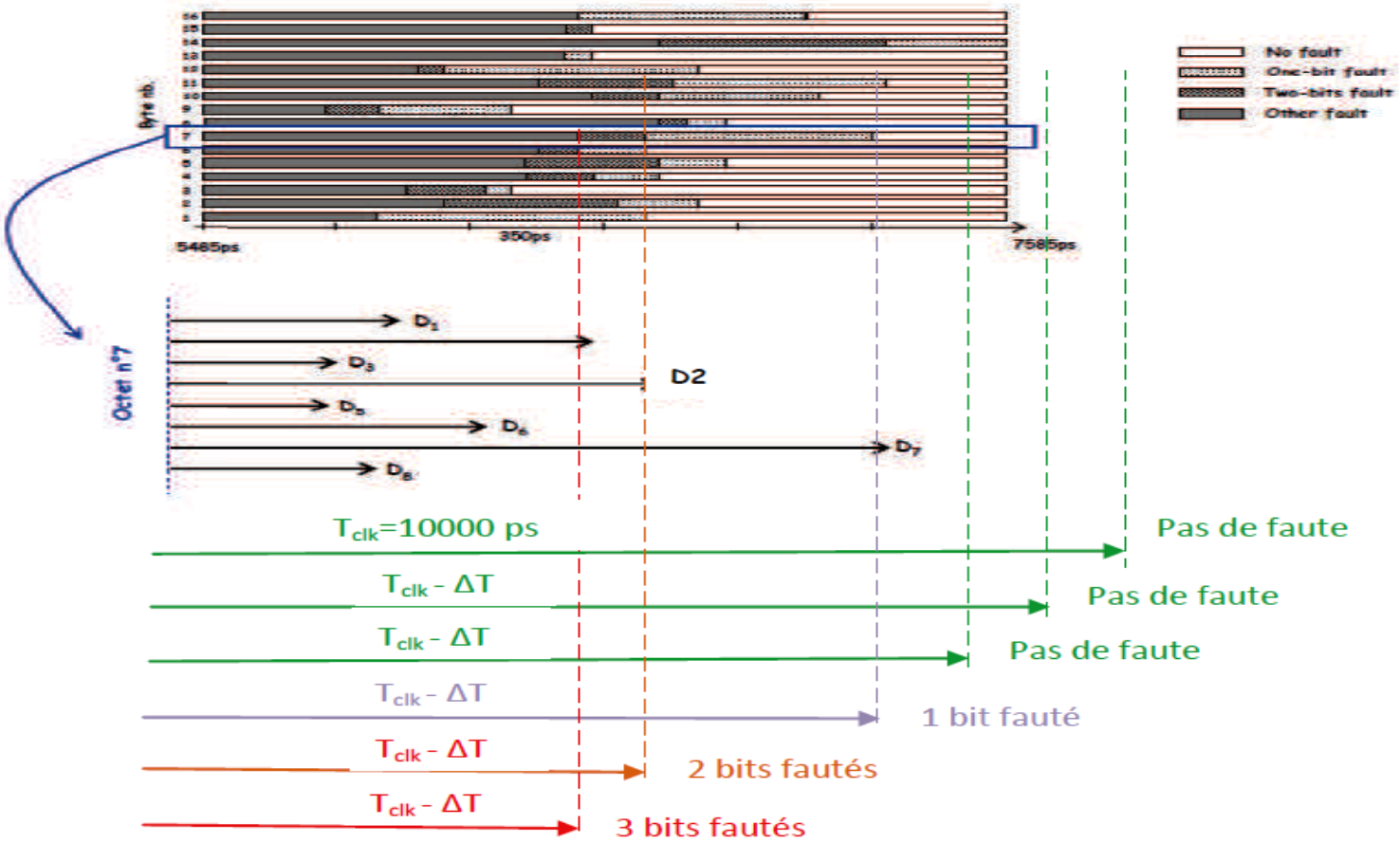


Figure 3. exemple

La figure 3 montre comment une diminution progressive de la période d'horloge conduit a différents profils d'erreur sur les octets de sortie d'une ronde d'AES. La période idéale pour cet exemple particulier est de 350 ps, en particulier sur l'octet n° 7. Le nombre de bits erronés après diminution de la période d'horloge est indique par la couleur de la barre. Barre verte : pas d'erreur ; barre mauve : 1 bit erroné/8 ; barre orange : 2 bits erronés/8 ; barre rouge : plus de 2 bits erronés/8.

Dans un premier temps, une diminution de la période n'entraine pas d'erreur sur l'octet 7. Après plusieurs diminutions successives on obtient une première erreur sur un seul bit D7 de l'octet 7, ce bit correspondant au chemin le plus long dans le calcul de l'octet.

Apres une nouvelle diminution de la période d'horloge, on obtient 2 bits erronés D4 et D7, puis 3 bits erronés D7, D4, D2 et ainsi de suite.

- **Attaques par perturbation de la température :**

  De même, une augmentation de la température produit une augmentation des temps de propagation. L'effet escompté dans ce cas est le même que celui recherché avec une baisse de l'alimentation ou de l'overclocking.

  Si les manipulations précédentes nécessitent une parfaite localisation dans le temps les attaques par perturbation de la température demandent une localisation géométrique.

  Une expérience montre qu'il est possible d'induire une faute de commutation sur un bit unique au moyen d'un spot lumineux utilisé pour augmenter la température. Les auteurs précisent d'ailleurs que, sans une maitrise expérimentale parfaite, tout le système attaqué devient défaillant. Il est donc extrêmement difficile de ne produire que des erreurs exploitables.

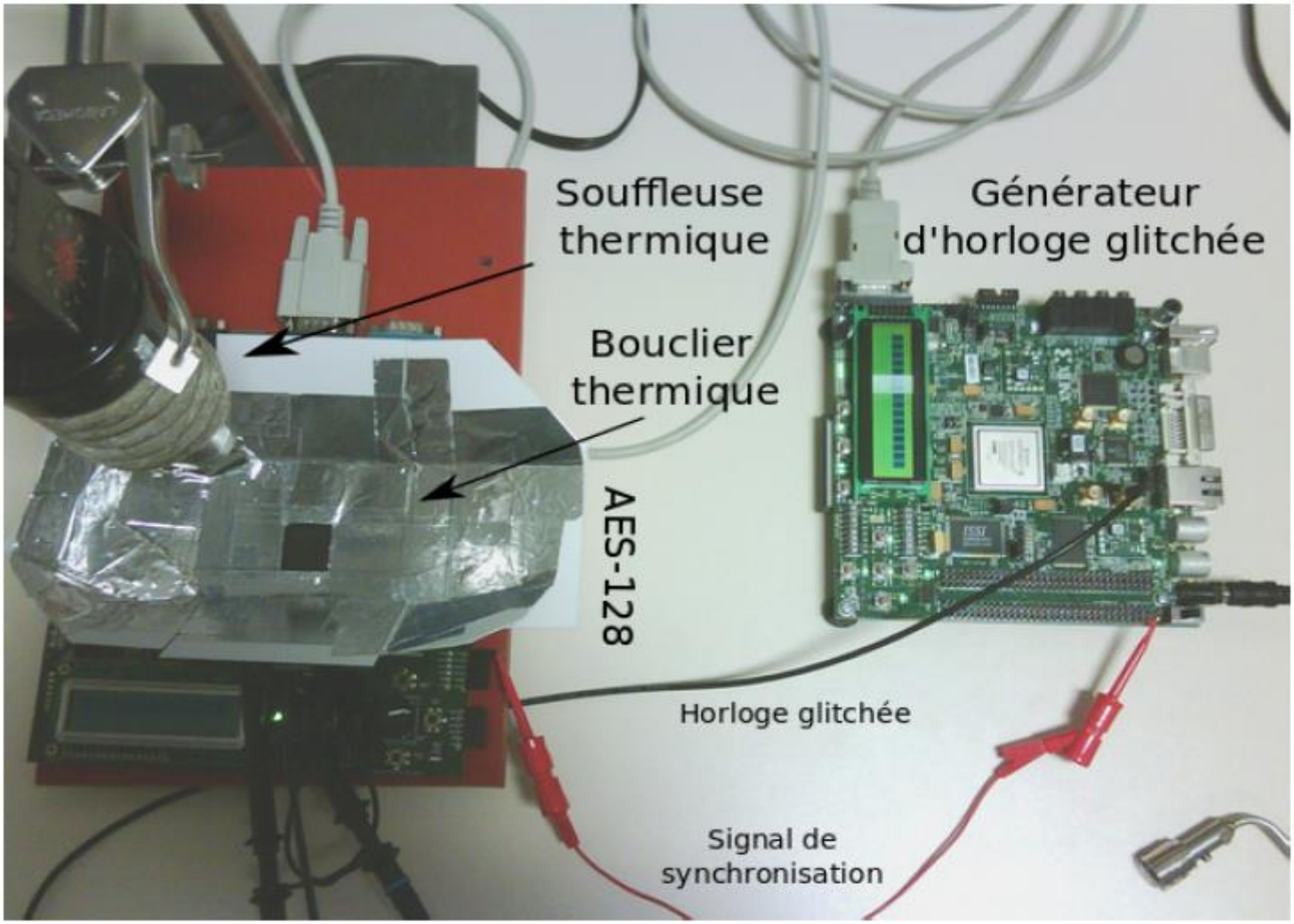


Figure 4. banc de chauffe

- **Attaques optiques :**

  Ce type d’attaques semi-invasif nécessite l'ouverture du boitier et une diminution de l'épaisseur du silicium pour avoir une bonne pénétration des éléments perturbateurs tels que la lumière.

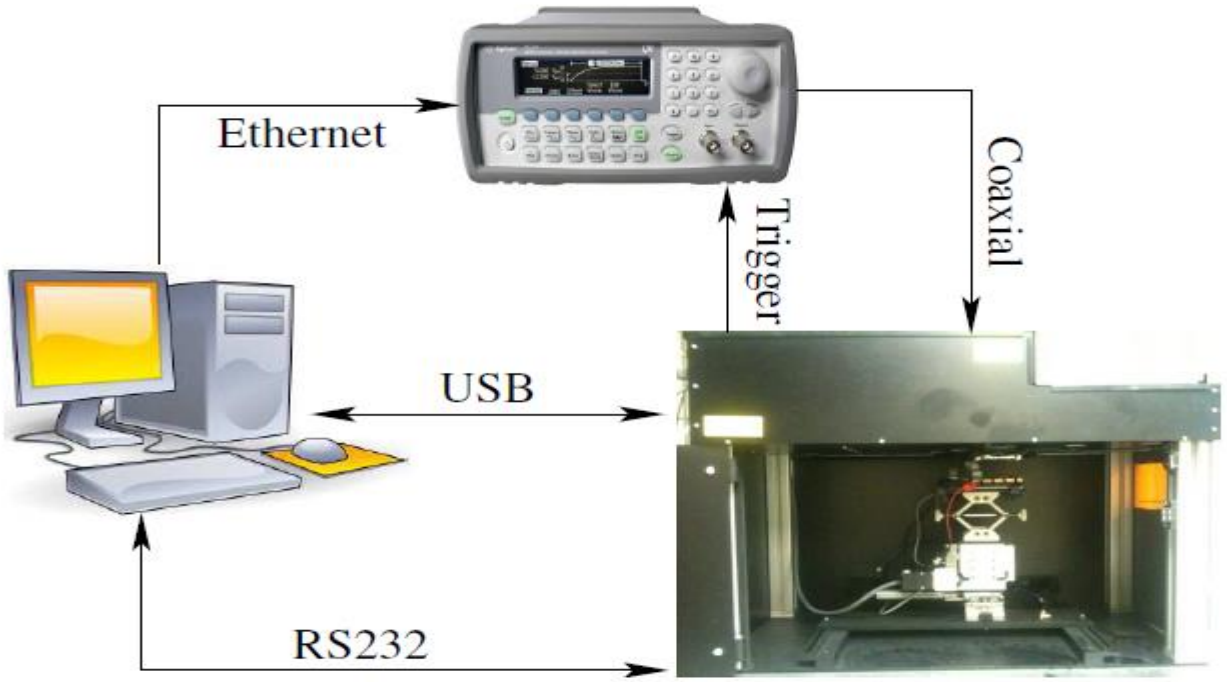


Figure 5. plate-forme d'injection optique

1.Injection de lumière

En 2002, Sergei Skorobogatov et Ross Anderson présentèrent un équipement assez simple et peu couteux permettant d'injecter en pratique des fautes en utilisant un simple flash d'appareil photo. Cela leur a permis de modifier la valeur d'un bit d'une cellule mémoire.

Le principe est d'utiliser l'énergie d'une émission lumineuse pour perturber le silicium du composant , cette énergie est en fait absorbée par les électrons du silicium, créant des paires électron-trou qui forment un courant photoélectrique local.

Le flash lumière ne permet généralement pas une localisation précise de l'injection de faute et provoque donc un disfonctionnement en de nombreux points du circuit attaqué.

. *2. Injection de particules*

Le principe est d'injecter des fautes en utilisant des ions lourds ou des particules ionisantes.

Ces injections de fautes peuvent être réalisées en utilisant des accélérateurs de particules. Cependant, cet équipement coûte cher et reste réservé à certains laboratoires.

Il est difficile d'assurer un bon control spatial et temporel avec cette technique.

3. Injection laser

Pour présenter l'effet d'une injection laser on va considérer l'exemple d'une cellule mémoire SRAM (1 bit). Une cellule classique SRAM (1 bit) est composée de deux inverseurs la cellule mémoire admet deux états stables : « 0 » ou « 1 ». Quand un faisceau laser traverse le silicium l'energie amenée par le faisceau laser peut créer des paires électrons-trous. Ces charges créent un courant transitoire qui inverse logiquement la tension de sortie de l'inverseur. Cette tension inversée serait donc appliquée a son tour sur le deuxième inverseur qui va commuter vers son état.

2.2 Exemple illustré :

1. Une implémentation particulière de RSA:

On va utiliser pour signer, un système RSA, de module n produit des deux nombres premiers p et q, dont la clé publique est e et la clé privée est d. Ainsi:

$$e\,d \equiv 1\ (\lambda(n))\ ,\quad \text{où } \lambda(n) = PPCM(p-1,\ q-1)$$

est la fonction de Carmichael.

Lors de la signature le propriétaire de la clé doit calculer une expression du type :

$$S = c^{d} \bmod n.$$

Pour cela il peut, soit faire un calcul direct, soit calculer :

$$S1 = c^{d} \bmod p$$

$$S2 = c^{d} \bmod q$$

et reconstituer S grâce au théorème des restes chinois :

$S = uS2p + vS1q \bmod n$, où (u, v) vérifie : $up + vq = 1$.

Cette dernière méthode est plus rapide. À cause du petit théorème de Fermat on peut écrire :

$$S1 = c^{d1} \bmod p.$$

$$S2 = c^{d2} \bmod q.$$

avec :

$$d1 = d \bmod (p - 1).$$

$$d2 = d \bmod (q - 1).$$

Cette méthode est à peu près 4 fois plus rapide que le calcul direct pour effectuer une signature. Mais elle est sensible à l'attaque par faute suivante:

2. Attaque par faute:

Supposons que l'un et un seul des deux calculs de S1 ou S2 soit faux. Supposons par exemple que ce soit celui de S1. Alors le résultat S′ faussement calculé avec S1′ au lieu de S1 et S2 vérifie :

$$S' \equiv S2\ (q).$$

$$S' \ddagger S1\ (p).$$

Si bien que :

$$(S'\ \hat{}e - c) \bmod n \equiv 0\ (q).$$

$$(S'\ \hat{}e - c) \bmod n \ddagger 0\ (p).$$

En conséquence,

$$pgcd(n, (S'e - c) \bmod n) = q.$$

Ainsi, grâce à la signature erronée S′ du message connu c, et à la clé publique (e, n), un attaquant est capable de factoriser n. Pour monter cette attaque il suffit de provoquer une erreur de calcul au bon moment. Sur une carte à puce cela est relativement simple.

3.CONTRE – MESURES :

Les protections contre les attaques par fautes sont de nature très variées. Elles ont pour but soit de détecter l'attaque soit de la rendre inefficace, et peuvent être déployées à tous les niveaux entre le matériel et l'application. Les protections matérielles se répartissent en trois grands groupes : certaines visent à rendre inefficaces les méthodes d'induction de fautes, d'autres essaient de détecter ou de corriger les fautes qui ont été induites et les dernières sont basées sur des modifications de l'algorithme pour rendre l'analyse plus complexe.

1.Contre-mesures matérielles :

- ✓ Détecteurs de glitchs (overcloncking)
- ✓ Détecteurs de lumière
- ✓ Dédoublement et masquage des registres
- ✓ Mécanismes d'intégrité des mémoires (RAM et EEPROM/FLASH)
- ✓ Boitiers des composants

2.Contre-mesures logicielles :

- ✓ Duplication de code
- ✓ "Pistage" du code a l'exécution

3.CONCLUSION :

Les fautes affectant les circuits sécurisés peuvent être permanentes ou transitoires. Ces fautes peuvent être intentionnellement injectées dans le circuit afin de contourner des éléments de vérification ou de récupérer des clefs secrètes de chiffrement. Certaines fautes affecteront le contrôle ,elles sont en théorie très efficaces mais difficiles à mettre en œuvre puisqu'elles demandent à l'attaquant une parfaite connaissance de l'implantation du circuit. D'autres ne demandent que d'obtenir un résultat faute de chiffrement mais l'erreur doit être suffisamment localisée pour qu'elle soit exploitable, et nécessite un moyen extrêmement précis d'injection de la faute pour ne produire que l'effet escompté. Les dysfonctionnements permanents sont rapidement signalés sans recourir à un arrêt du système pour test de maintenance.

Les dysfonctionnements transitoires affectant momentanément le circuit seraient eux aussi détectés, ce qui devient d'autant plus important avec les technologies actuelles et à venir qui sont de plus en plus sensibles aux phénomènes naturels.