1 Séance 1

Summary
Common Criteria sont particulièrement importants et essentiel mais tous les systèmes ne peuvent pas les suivre. En effet, de nombreux systèmes sont trop complexes pour correspondre aux critères. Seul les cartes à puces correspondent le mieux (EAL5-6-7).

2 Séance 2

Les méthodes d'attaques

Types d'attaques :

Implementation based attacks Attaque au laser, rajout de composants, de sondes ...

Observation-based Mesure du courant lors d'une utilisation normale ou caractéristique pour repérer la clé de chiffrement et comprendre le fonctionnement du circuit.

Perturbation-based A partir d'une perturbation induite, et le résultat, repérer le fonctionnement du circuit.

Définir la menace (pirate dans son garage, chercheur dans une université).

2.1 Méthodes invasives

Depackaging/Repackaging (Exemple de la carte à puce) Récupérer la puce en enlevant tout le plastique de la carte et en le plaçant dans des solutions chimiques (pour dissoudre le plastique), puis obtenir le circuit. On rajoute souvent des détecteurs de lumière pour repérer l'état. En effet, cette étape est utilisée principalement pour le debuggage et la qualité, cependant en rajoutant des capteurs de lumière, on peut détecter une attaque et détruire les informations. Le pirate qui s'y connaît un peu, peut toujours déconnecter les capteurs. Il s'agit du premier niveau de protection.

Direct optical observation De nos jours, il est difficile d'analyser un circuit car les circuits sont sur plusieurs couches, et ils sont souvent très petits. Cela empêche au pirate de situer les composants dans le circuit (mémoire, chiffrement).

Méthodes physiques

Layout observation and micro-probing On peut supprimer les couches (chimiques), pour repérer ce qui est utile et ce qui n'est que du remplissage. (voir example page 26, une première image avec seulement les rails d'alimentation, et dans la deuxième, on peut localiser de nombreux composants et ensuite comprendre le circuit par l'analyse, et tester d'autres attaques sur un autre circuit). Après avoir analyser le circuit, on peut descendre des "aiguilles" (micro-probing) pour sonder les puces. Deux problèmes, il faut localiser précisément les signaux que l'on veut mesurer (le micro-probing se fait souvent sur les bus car il y a plus de chance d'obtenir le bus du processeur. Il faut aussi en poser un nombre suffisant en simultané (en étant doué, quelques dizaines de pointes) car il est difficile de les placer de façon stable. De plus, il faut pouvoir analyser, en post-traitement, les signaux récupérer. Il faut donc beaucoup de compétences, de matériels et de temps pour faire cette attaque.

Pour un FPGA classique, il suffit de moins de 10 pointes (l'horloge, entrées/sorties ...)

Reading ROM Memory contents On peut récupérer le programme du système et parfois la clé de chiffrement. Pour éviter cela, il faudrait mélanger les bits, c'est-à-dire, 8 bits qui se suivent doivent être éparpillé sur toute la surface de la ROM, cela permet d'éviter qu'une simple lecture du masque suffit à lire la ROM.

Reverse engineering A partir du matériel, on en déduit la fonction du circuit (ex : décodeur DVD).

Physical interventions On peut modifier le circuit en coupant certaines connexions et d'en recréer d'autre (pour "supprimer" certains éléments de sécurité comme des capteurs ou pour réparer un fusible qui a servi pour les tests (scanpath), un fusible qui a sauté est facilement localisable).

Méthodes sans contact

Voltage contrast microscopy (SEM) Méthode permettant d'obtenir le comportement des circuits sans abîmer le circuit en prenant des photos au microscope.

Dynamic analysis of data sent on a bus

2.2 Méthodes non invasives

Side channels ou canaux cachés

Il ne suffit pas de faire du circuit intégré pour avoir un circuit sécurisé. Il existe de nombreux éléments qui peuvent permettre d'analyser un circuit (avec des attaques qui s'y rapportent) :

Execution time : Timing Attack.

Power consumption : Power attacks (SPA / DPA / CPA).

Electromagnetic emmisions : Electromagnetic Attack (EMA)

Sound, heat, photons : analyse des secteurs du circuit qui travaille à l'instant t.

A partir de ces mesures, il faut savoir décoder les données qui correspondent. De plus, il est démontré que les algorithmes secrets sont peu fiables par rapport à ceux testés et prouvés (RSA, il y a juste la clé à protéger, l'AES est le remplaçant du DES qui est obsolète par la force brute). Car il y a beaucoup plus de gens qui peuvent le tester et le confirmer. L'inconvénient est, bien sûr, que tout le monde connaît l'algorithme et ce qui est important à mesurer.

2.2.1 Timing attacks

Selon les entrées, le temps de réponse peut aider à analyser la réponse. Une condition sur la clé par exemple. La solution est de modifier le code de tel façon que qu'on est le même temps quelque soit l'entrée (perte de performance). La "timing attack" est souvent utilisé sur le code PIN car l'aglorithme compare les chiffres un par un. Donc plus, la vérification met du temps, plus le nombre de chiffres sont bons. Cependant le fait qu'il est un nombre limité d'essais oblige un meilleur algorithme.
L'optimisation des processeurs permet d'avoir une description plus exacte dans la timing attack car il travaille uniquement quand il a besoin de faire quelque chose.

2.2.2 Power consumption

TODO

Un transitor consomme lors d'un changement d'états. Donc si on a une hypothèse ou un accès on a un ensemble réduit d'hypothèse sur la valeur finale en analysant la consommation d'un groupe de transistor. On peut donc ainsi faire des hyphotèses sur la clé jusqu'à faire une attaque par force brute.

SPA, simple power analysis

La consommation n'est pas la même selon la valeur du bit. L'exemple page 35 permet de savoir où se situe le calcul le plus complexe du RSA et on peut en déduire le type de calcul. Et ainsi, on sait quand recupèrer la clé.

DPA, differential power analysis

Moyenne de courbes de consommations

Sur un bus, on a souvent Y=aH(X)+b avec Y puissance consommée, X la donnée, et H le poids de Hamming. Peu précis mais permet d'avoir une approche. Pour être plus précis, il y a le modèle de la distance de Hamming : Y=aH(P-X)+b avec P la valeur précédente.

Sur l'image de la page 38, la mesure de la différentiel permet de deviner quand on a obtenu la bonne clé sur l'ensemble des tests.

2.2.3 Electomagnetic attacks

Même chose que la consommation mais avec un matériel différent et plus précis (méthode encore théorique et difficile en pratique). Plus d'information sur le signal enregistré. Beaucoup de contraintes et de paramètres.

2.2.4 Perturbation-based attacks

A pour méthode de changer les conditions de fonctionnement en manipulant la fréquence de la clock, l'alimentation et la temperature, causant des changements de temps de propagations. Pour lutter contre ces methodes, on rajoute des capteurs (qui peuvent être désactivé). D'autres perturbations existent tels que des "glitches" sur l'horloge ou l'alimentation, des perturbations optiques ou électromagnétiques.

Overclocking vs clock glitch :

TODO

2.2.5 Laser-based attack

Plus le circuit est petit, plus le laser doit être cher et complexe. Permet de modifier des valeurs.

2.2.6 Fault-based attacks

En créant une faute, on peut exploiter le résultat et obtenir la clé. La Bellcore attack à permis de casser la clé de 1024 bit du RSA avec un résultat crypté correct et un autre erroné (avec laser) et par un simple calcul du GCD (greatest common divisor ou PGCD) on obtient la clé (Description page 45).

2.3 Conclusion

Une attaque a réussi non pas quand on a injecté une faute mais quand on a obtenu le résultat. Donc il faut que le résultat d'une attaque par faute doit être caché ou être corrigé, ou de changer l'erreur (repmlacer par une autre erreur).
==> Attention au changement de delais.
Autres exploitations d'erreurs : changer la configuration du circuit (compteur illimter de tentatives pour le code PIN, "protection bypass", ..

3 Séance 3

www.industrie.gouv.fr/logiciel-embarque : Rapport du gouvernement sur le logiciel embarqué. Beaucoup de demandes dans l'industrie (30000 pour 5 ans).

Page 47

L'attaque par faute est surtout une erreur logiciel (commutation d'un bit (SEU) ou plusieurs (MBU)). Finalement, nous obtenons un ou plusieurs bits erronés dans un registre. Il faut faire appel aux codes correcteurs d'erreurs (le bit de parité ne suffit pas toujours). Les attaques doivent être réalistes, il faut que l'attaque corresponde au profil de l'attaquant en prévoyant ses moyens.
Au sujet du logiciel, les attaques se font de plus en plus nombreuses car les systèmes sont plus "ouverts" (ex : téléphone). Il faut se méfier :

- des failles du logiciel
- des buffers qui ne doivent pas être "overflow"
- des trojans (ex : certaines applis pour smartphones)
- du réseau, avec les attaques classiques sur un réseau.

Pour cela, plusieurs solutions pour les "smart-cards", il faut déjà connaître son matériel pour adapter le logiciel parfaitement, implémenter les dernières technologies qui ont été vérifiées par la communauté ... (exemple page 48). Les failles sont plutôt dans l'EEPROM (stockage applis) mais la partie critique est dans la ROM (où se trouve le coeur de la carte SIM). Quelques attaques :

- Attaque par downgrade en installant la version précédente pour rendre le système sensible à une faille connue.

- Les attaques à distance pour les systèmes sans fils (un mètre ou plus) en récupérant des informations ou autres ... Il faut prendre en compte les émissions EM générées par le clavier qui peuvent être récupéré à distance (dépend de l'équipement 15 ou 20 m, les pertes ne sont pas trop importantes et passe tout cryptage) ==> existe.
- Faire confiance (ou pas) aux laboratoires externes et étrangers qui peuvent rajouter des failles (pas d'exemple sur cela // voir pologne, communications bancaires).

Non seulement les produits doivent être sécurisés mais ils doivent être vérifiés et testés. Il faut vérifier que les protections n'entraînent pas des dysfonctionnements.

3.1 Attaques : impact des techniques de tests

Les attaques sur l'horloge lorsque l'horloge est externe (l'horloge doit être interne) car la synchronisation de l'attaque et la base de temps pour l'attaquant est une faille. Alors que il y a parfois besoin d'une horloge externe pour les tests. Les accès privilégiés sur le circuit sont utiles pour les tests mais peuvent se révéler être des failles. Exemple page 51 : l'implantation du "scanpath" est à surveiller, dans les exemples fournis, ils faut peu de signaux pour contrôler totalement le circuit.

Sécurisation des modes de tests

- clé secrète
- Détection de décalage lors du mode test
- Rajout d'aléatoire dans la chaîne (complique la tâche à l'attaquant et le testeur, connaissant l'algorithme, peut faire ses tests)
- Rajout de bits non utiles
- Éviter les techniques de scan en utilisant des tests auto.

En conclusion, quasiment toutes les contraintes de design ont un lien avec la sécurité.

3.2 Modélisation et caractérisation des erreurs

A une époque, il était évalué que la plupart des attaques seraient une faute sur un seul bit. Deux exemples : l'attaque de Giraud qui modifie un seul bit à un certain moment du système mais n'importe où. Et l'attaque de Blömer avec l'inversion l'une valeur qui est plus complexe mais qui change beaucoup.

Attaque par laser

Selon la taille du laser, plusieurs bits d'un registre peuvent être modifié. L'élargissement du faisceau est possible par contre la réduction l'est moins. Les solutions sont la parité d'une cellule d'un registre ... Actuellement, plusieurs bits peuvent être modifié en même temps et sur plusieurs registres.

Il faut noter que les protections coûtent très rapidement très chers, il faut donc parfois faire des compromis.

Pour éviter l'optimisation par l'outil de synthèse qui pourrait supprimer les parties de vérification et de redondance dans un bloc, il faut décrire des blocs séparés (en VHDL) pour que l'outil les synthétisent et optimisent indépendamment. Après coup, les portes sont mélangés pour éviter l'identification des blocs. L'implantation de blocs de vérifications et de redondance coûte cher en place et en argent (même pour une simple parité, il y a des codeurs et décodeurs à rajouter).

Étude d'une attaque au laser page 63-66

Étude d'une attaque basée sur la SRAM d'un FPGA : page 66-68

3.3 Protections : page 80

Compromis sécurité/coût. Les approches évoluent rapidement, des méthodes pour détruire des protections peuvent apparaître en quelques mois : 100% sécurisé n'existe pas. Les protections peuvent faire apparaître des possibilités pour d'autres.

3.4 influence du style de design

Les circuits asynchrones sont plus robustes que les synchrones (pics de consommation, émission EM ...)