Poor Man's ChipTweaker

Nous avons des moyens (abordables) de vous faire parler

Luka Matic (Croatie)

Les HSM (*Hardware Security Modules*, en français, boîtes noires transactionnelles) sont des dispositifs électroniques dotés de différents systèmes de protection matérielle contre la lecture non autorisée des données secrètes qu'ils contiennent. Ils servent généralement au chiffrement ou aux signatures numériques, mais ils peuvent aussi effectuer d'autres tâches. À titre d'exemple, les cartes bancaires ou les cartes d'accès sont des HSM. En cas de vol ou de perte, ces cartes doivent garder secrets le plus longtemps possible les codes confidentiels et les clés privées. En outre, de nombreux microcontrôleurs à usage général possèdent également des fonctions de protection de la mémoire, destinées à empêcher la copie illégale de microprogrammes exclusifs. L'outil Poor Man's ChipTweaker (PMCT) présenté ici sert à découvrir les éléments de base des attaques non invasives contre les HSM. Grâce à cet outil, vous allez pouvoir tenter de déverrouiller un microcontrôleur à mémoire protégée.

Comme nous l'avons expliqué dans un article précédent [1], nos espions à petit budget Alice et Bob ne peuvent pas compter sur la construction de leurs propres HSM, car cela nécessite un équipement spécialisé coûteux, et des connaissances qu'ils ne possèdent probablement pas. Pour leurs opérations critiques, ils utilisent leur propre matériel construit avec des composants bon marché et polyvalents auxquels ils peuvent se fier. Cela fonctionne bien pour eux, tant que les procédures de sécurité des opérations (OpSec) sont respectées. D'un autre côté, ils ne peuvent pas non plus éviter d'utiliser des HSM, les cartes bancaires par exemple, et ils peuvent tomber sur un modèle dont ils pourraient vouloir percer les secrets. L'outil PMCT leur sera donc très utile car il leur permet d'apprendre les éléments de base des attaques non invasives contre les HSM.

Attaques matérielles

Les attaques visant les HSM appartiennent pour l'essentiel à trois types :

1. **Non invasive :** Le HSM n'est ni ouvert ni décapsulé. L'attaque est réalisée à l'aide de différents signaux électriques appliqués à un port de communication ordinaire et aux broches d'alimentation du HSM. C'est ce pour quoi l'outil PMCT a été conçu. Une attaque peu coûteuse.
2. **Semi-invasive :** La couche supérieure de la puce de silicium du HSM est retirée. Les contacts électriques de la puce en silicium ne sont pas exposés. L'attaque est réalisée en illuminant certaines parties des microcircuits du HSM à l'aide d'impulsions lumineuses. Une attaque de coût modéré.
3. **Invasive :** La puce en silicium est entièrement exposée, ce qui permet de connecter les microsondes aux contacts de la puce. L'attaque est réalisée en injectant des signaux électriques dans les microcircuits du HSM. Il s'agit d'une attaque coûteuse qui nécessite un équipement hautement spécialisé.

De puissants systèmes « ChipWhisperer » numériques sont bien sûr disponibles [2]. Mais ce que je voulais faire, c'était créer un dispositif basé sur une technologie plus ancienne, similaire à ce que le Dr Skorobogatov a fait [3], avec tous les signaux analogiques et numériques entièrement accessibles, ce qui pourrait être une plateforme d'apprentissage plus performante pour comprendre les principes des attaques non invasives.

Spécifications

Les attaques non invasives se basent sur différentes méthodes d'injection de défauts, comme l'application de données mal formatées ou de signaux d'amplitude et de fréquence incorrectes à l'une des broches du HSM (dont les broches d'alimentation). Ces méthodes peuvent faire en sorte que le DUT (*Device Under Test*, en français Dispositif testé, ou plutôt attaqué) effectue de nombreuses actions incontrôlées et, si possible, révèle ses secrets.

Les attaques non invasives nécessitent beaucoup de traitements de données, d'ajustements et de recherches complexes pour chaque dispositif testé particulier, mais s'il est possible de le déverrouiller de manière non invasive, l'outil PMCT, simple et bon marché, peut vous aider à le faire. Il dispose des fonctionnalités suivantes :

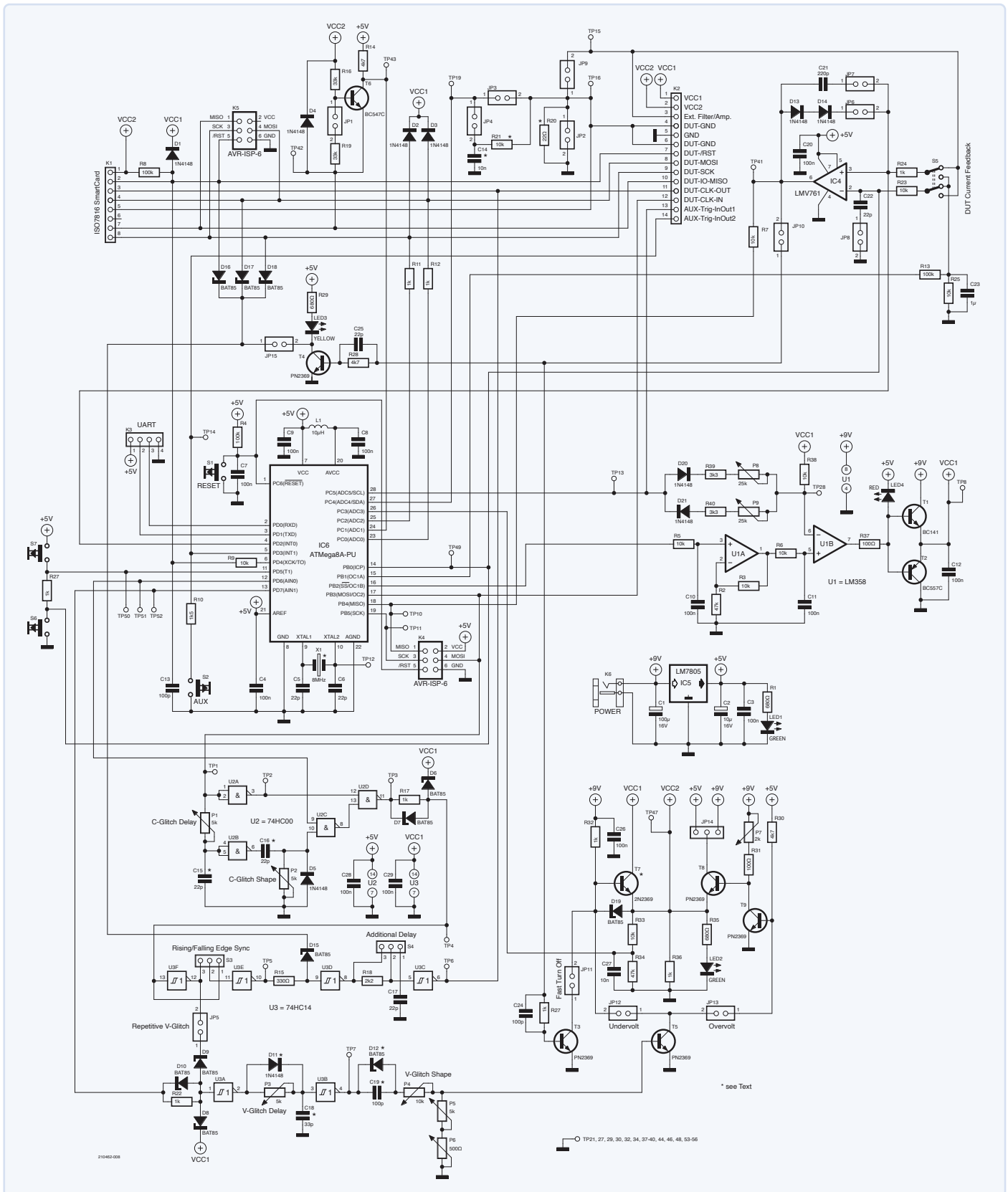


*Figure 1. Schéma de l'outil Poor Man's ChipTweaker (PMCT). À noter l'utilisation de transistors rapides de type 2369. Pour des questions de dissipation thermique, le transistor T7 doit être de type TO-18 (boîtier métallique, 2N2369). Les types PN2396 en boîtier plastique utilisés pour les autres peuvent être moins chers et/ou plus faciles à trouver. Les autres composants portant la marque « * » devraient être montés sur support pour permettre d'expérimenter facilement différentes valeurs.*

1. **Générateur de défaut de signal d'horloge :** L'insertion de signaux trop rapides sur l'entrée d'horloge (CLK) du HSM peut faire sauter une instruction machine ou l'exécuter de manière incorrecte, et donc, par exemple, ignorer une vérification du code confidentiel.
2. **Générateur de défaut de tension :** La modification de la tension d'alimentation du HSM hors de sa plage de fonctionnement nominale peut également entraîner des opérations incontrôlées. L'outil PMCT peut provoquer des défauts de tension lents, moyens et rapides, sous forme de surtensions ou de tensions insuffisantes.
3. **Alimentation variable entre 1,0 V et 6,0 V du dispositif testé :** Commandée par le microcontrôleur du PMCT.
4. **Comparateur pour désactiver rapidement un dispositif testé :** Au-dessus ou au-dessous d'un certain seuil de courant d'alimentation (par exemple, pour une attaque de sabotage d'écriture dans une mémoire EEPROM), armé-désarmé ou directement déclenché par le microcontrôleur principal.
5. **Interface SPI avec « bit-banging » (génération de signaux par logiciel) et UART bidirectionnelle mono-broche** pour la communication avec le dispositif testé (y compris tous les types de cartes à puce), comprenant des dispositifs de décalage de niveau de tension bidirectionnels couvrant la plage comprise entre 1,5 V et 6,0 V.

Au-delà des attaques citées, le PMCT peut effectuer toutes sortes d'attaques de synchronisation et d'analyse de puissance en combinaison avec un oscilloscope numérique (par exemple, un SmartScope de LabNation), un oscilloscope analogique à mémoire de 100 à 200 MHz (par exemple, le Tektronix 466) et un PC avec un logiciel comme MATLAB pour l'analyse des données hors connexion. Contrairement aux dispositifs « ChipWhisperer » intégralement numériques, tous les signaux analogiques et numériques sont accessibles pour être étudiés et analysés, et il est possible d'ajuster plus précisément la synchronisation du défaut grâce à un réglage analogique continu avec des potentiomètres dépourvus de pas de quantification. Un ATmega8 lent peut ainsi coordonner toute la procédure d'attaque, et aucun circuit FPGA n'est donc nécessaire. Il est possible de générer des défauts d'une durée inférieure à 10 ns, ce qui induit des erreurs pour les unités centrales conçues pour fonctionner jusqu'à 50 MHz, et plus. De nombreux microcontrôleurs et cartes à puce ne fonctionnent que jusqu'à 20 MHz.

Description du matériel

Au centre du schéma (**figure 1**), nous trouvons le microcontrôleur (MCU, IC6, ATmega8) qui coordonne les procédures d'attaque. Le circuit autour d'IC1 est une source de tension variable pour le dispositif testé. Sa tension de sortie VCC1 est commandée par modulation de largeur d'impulsions (PWM) par la sortie de temporisation OC1B d'IC6. Il est ainsi possible de produire des défauts de tension lents. La broche PC5 du microcontrôleur peut être dans l'un des trois états suivants : L (bas), H (haut) et high-Z (haute impédance), ce qui lui permet de modifier rapidement l'amplification d'IC1B, et donc de produire trois valeurs différentes de VCC1, définies par P8 et P9. C'est ainsi que l'on produit un défaut de tension à vitesse moyenne (mesurée en microsecondes). La LED rouge LED4 va s'allumer à titre d'avertissement et va limiter VCC1 à environ 6 V, ce qui n'est pas trop élevé pour un dispositif alimenté en 5 V.

Production de défauts

Le dispositif testé est normalement alimenté par VCC2 au travers du transistor T7 (un 2N2369 en boîtier TO-18). Le transistor T5 peut rapidement ramener sa base à l'état bas (en insérant le cavalier JP12), ce qui entraîne un défaut de sous-tension rapide (de l'ordre de 10 ns). Si le cavalier JP13 est inséré, le transistor T8 amènera rapidement la tension VCC2 au niveau haut, créant ainsi un défaut rapide de surtension. Les défauts de tension rapides sont formés par C19, P4, P5 et P6. Le retard du défaut de tension par rapport à l'impulsion CLK est ajusté avec P3. Lorsque la sortie PD7 du microcontrôleur est mise à l'état haut, un seul défaut de tension est déclenché. Si la sortie reste haute, et si le cavalier JP5 est court-circuité, les défauts de tension seront déclenchés à chaque impulsion CLK.

Le circuit IC2 sert à générer un défaut d'horloge sur le dispositif testé. La sortie en PWM OC2 du microcontrôleur produit l'impulsion CLK pour le dispositif testé. Si le port PD6 est à l'état haut, IC2C transmettra un défaut de courte durée à IC2D à chaque impulsion CLK. Il est possible de sélectionner la polarité du défaut à l'aide d'un cavalier ou d'un commutateur sur S3.

IC4 est un comparateur rapide avec retour positif (amélioré par D13, D14 et C21) destiné à éteindre le dispositif testé en quelques nanosecondes. Il est également possible de le faire à l'aide d'une commande du microcontrôleur (PD5 ou PD2 pour allumer, PB0 pour éteindre le dispositif testé) ou en appuyant sur les boutons S6 et S7.

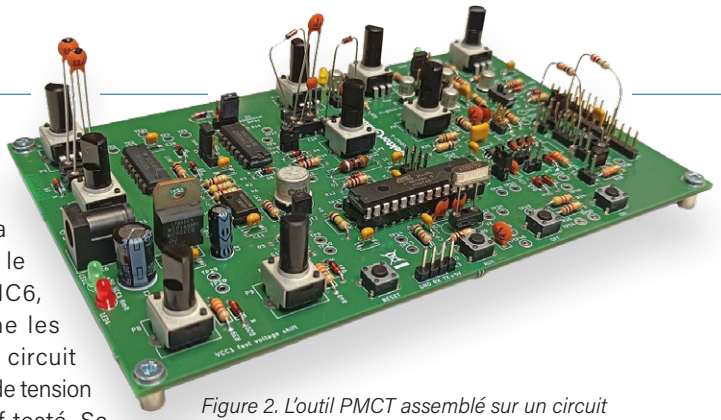


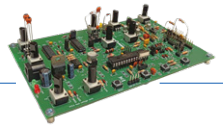
*Figure 2. L'outil PMCT assemblé sur un circuit imprimé de bonne taille. Les composants montés sur support portent la marque « * ».*

Current Consumption Attacks

IC4 peut également être configuré pour éteindre le dispositif testé si la consommation de courant de ce dernier (mesurée avec la résistance R20) est supérieure ou inférieure (selon une sélection effectuée par S5) à une tension de seuil (sur C23, contrôlée par PWM sur la broche OC1A). Cette approche est généralement utilisée pour empêcher le dispositif testé dans sa mémoire EEPROM interne (qui consomme du courant supplémentaire). La sortie d'IC4 active le transistor T3, qui, à son tour, désactive T7. Elle active également le transistor T4 (désactivation auxiliaire) pour amener toutes les lignes de la carte à puce vers le bas et éviter toute alimentation fantôme.

L'entrée analogique ADC4 surveille le courant du dispositif testé. Comme il s'agit d'un CA/N lent, le signal peut être filtré par R21/C14. Il peut aussi être traité par un filtre/amplificateur externe plus rapide (sur le connecteur K2), si nécessaire. Pour une attaque de synchronisation ou d'analyse d'alimentation, ce signal est généralement enregistré par un oscilloscope pour être traité et analysé hors ligne avec un outil comme MATLAB.

Comme le microcontrôleur fonctionne à partir de 5 V alors que le dispositif testé est alimenté par la tension VCC2 (généralement) plus faible, le dispositif de décalage de niveau bidirectionnel (construit autour du transistor T6) est nécessaire pour la broche 7 d'E/S du dispositif testé sur le connecteur K1 (un port UART bidirectionnel, qu'utilisent aujourd'hui la plupart des cartes à puce). Les broches 4 et 8 servent pour les cartes à puce avec interface SPI (par exemple, la FunCard). Comme elles sont unidirectionnelles, les dispositifs de décalage de niveau sont plus simples (diodes D2 et D3).



Circuit imprimé

Pour faciliter la construction de votre propre outil PMCT, une carte à circuit imprimé (PCB) de la taille du format Eurocard a été conçue pour lui (**figure 2**). (Voir [4] pour accéder au projet KiCad et aux fichiers gerber.) Tous les éléments sont des composants de type à trou traversant, à l'exception d'IC4, conditionné en boîtier SOIC-8. L'assemblage de la carte ne devrait pas poser de problèmes, mais il faut faire attention à certains détails :

- Les cavaliers à 3 broches « Sxx » peuvent être plus pratiques comme interrupteurs unipolaires et bidirectionnels (SPDT) à glissière ou à bascule.
- Vous préférerez peut-être utiliser des trimmers plutôt que des potentiomètres. Les trimmers sont plus faciles à trouver et offrent une plus large gamme de valeurs.
- Le transistor T7 doit être un 2N2369 dans un boîtier métallique TO-18 pour des raisons de dissipation thermique. Les transistors PN2396 TO-92 peuvent être remplacés par des types 2N2369, selon ce qui est le plus facile à trouver.
- Les composants marqués d'un symbole « * » (par exemple, D11 et C18, etc., à l'exception de T7) devraient être montés sur support pour expérimenter facilement différents types et valeurs.

La carte comporte de nombreux points de test avec une connexion de masse à proximité. N'essayez pas de comprendre leur numérotation ; elle ne répond à aucune logique.

Préparer l'attaque

Le dispositif testé pour les applications de démonstration est une FunCard standard (qui contient un microcontrôleur Microchip AVR AT90S8515 et une mémoire EEPROM AT24C). Sans aucune protection sérieuse, ce dispositif est relativement facile à attaquer. Ce sera donc la première étape si vous voulez avancer dans ce domaine. Chargez le micrologiciel dans la FunCard accessible en [4] par le biais du connecteur ISP K5. Ensuite, utilisez une des deux variantes du micrologiciel ATmega8 (concernant l'attaque d'une FunCard, également en [4]) pour programmer le microcontrôleur IC6 du PMCT à l'aide du connecteur ISP K4 (et non K5). Vous pouvez maintenant essayer quelques actions élémentaires.

Connectez un terminal série au port UART K3 et configurez-le pour 9600-8-N-1 (soit 9600n81). Après avoir actionné le bouton de réinitialisation S1, vous devriez obtenir un écran comme celui représenté sur la **figure 3**. Les options 0 et 1 permettent d'envoyer des commandes simples au microcontrôleur et à la FunCard. Toutes les commandes ont une longueur de 4 octets, le dernier étant toujours égal à 0x0d (CR). Référez-vous au **tableau 1** et au **tableau 2**.

Démonstration d'attaque par défaut d'horloge

Il s'agit d'une démonstration d'une attaque élémentaire de type défaut d'horloge. La FunCard effectue plusieurs opérations en 16 bits sur deux opérandes d'entrée, situés dans la mémoire EEPROM interne de son microcontrôleur aux adresses 0x00, 0x01, 0x02 et 0x03 au format « big-endian » (mots de poids fort en tête). Le fait d'émettre une impulsion d'horloge (impulsion trop rapide pour

```
> 0 : Send a single command to Smartcard
> 1 : Send a single command to MCU
> 2 : Find minimum Vcc2
> 3 : Adjust minimum Vt
> 5 : Funcard no glitch demo
> 6 : Funcard clock glitch demo
> 7 : Funcard volt glitch demo
> 8 : Volt glitch loop test
> 9 : Volt & Clock glitch loop test
> A : Fast Vcc2 test
> B : Brute-force PIN demo, Funcard MCU internal counter
> b : Brute-force PIN demo, Funcard AT24C external counter
> R : Reset MCU & Smartcard

Vcc2 RAW set to:0xDD    Vcc2 feedback: 5,092V
Vt RAW set to:0xFF    Vt scaled set to: 0,452V    Vi feedback: 0,166V
Smartcard fclk division factor set to RAW:0x00

> Select Option (0-X): 5

No glitch demo.

Answer To Reset: 0x45 0x67 0x78 0x9A 0xBC 0xDE 0xF0 0x12 0x34 0x56 0x78 0xDE 0xAD 0x01 0x23 0x00 0x00 0x00 0x00 0x00
```

Figure 3. Vous pouvez choisir dans un menu l'attaque que vous souhaitez.

Commande	Description
V2x	« x » est un nombre sur 8 bits définissant le ratio de PWM sur OC1B pour fixer la tension VCC2.
Vtx	« x » est un nombre sur 8 bits définissant le ratio de PWM sur OC1A pour fixer la tension Vt, un seuil sur C23 pour activer IC4.
Fxy	« x » est un facteur de division de fréquence brute, destiné à produire une impulsion CLK pour le dispositif testé sur OC2. Si le facteur est fixé à 0, OC2 fonctionnera à la moitié de la fréquence du cristal du microcontrôleur ; « y » n'est pas pris en compte.
GLx	Les 3 bits de poids le plus faible de « x » définissent les broches PD7, PD6 et PC5 du microcontrôleur ; pour le test.
onx	Mise en marche du dispositif testé ; « x » n'est pas pris en compte.
ofx	Arrêt du dispositif testé ; « x » n'est pas pris en compte.
Sxy	Enregistrement des paramètres actuels dans l'EEPROM du microcontrôleur ; « x » et « y » ne sont pas pris en compte.

Tableau 1. Commandes adressées au microcontrôleur Terminer chaque commande par 0x0d (CR).

Commande	Description
Rxy	Lecture d'un octet à l'adresse « x » de la mémoire EEPROM interne de l'AT90S ; « y » n'est pas pris en compte.
Wxy	Écriture d'un octet « y » à l'adresse « x » de la mémoire EEPROM interne de l'AT90S.
Pxy	Test d'un code confidentiel « xy » (au format BCD). Ce code est stocké dans la mémoire EEPROM interne de l'AT90S.
rxy	Lecture d'un octet à l'adresse « x » de la mémoire EEPROM de l'AT24Cv; « y » n'est pas pris en compte.
wxy	Écriture d'un octet « y » à l'adresse « x » de la mémoire EEPROM de l'AT24C.
pxy	Test d'un code confidentiel « xy » (au format BCD). Ce code est stocké dans la mémoire EEPROM de l'AT24C.

Tableau 2. Commandes adressées à la carte à puce Terminer chaque commande par 0x0d (CR).

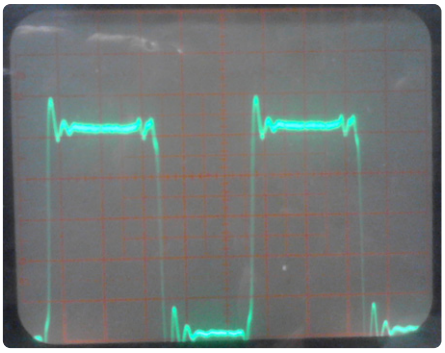


Figure 4. Un signal d'horloge propre sans défauts. L'oscilloscope a été réglé sur 50 ns/div sur l'axe horizontal et 1 V/div sur l'axe vertical.

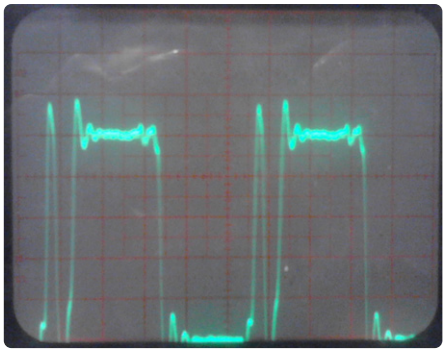


Figure 5. Un signal d'horloge assorti d'un défaut (50 ns/div axe horizontal, 1 V/div axe vertical). Réglage des potentiomètres P1 et P2 pour créer un défaut comme indiqué ici.

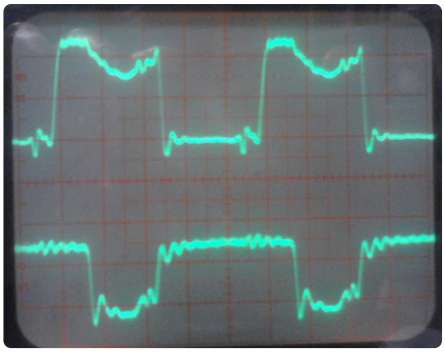


Figure 7. Le défaut s'est déclenché 30 à 40 ns suite au front montant de l'impulsion d'horloge. La trace supérieure est le signal d'horloge, la trace inférieure est la tension d'alimentation VCC2 (50 ns/div horizontal ; 1 V/div vertical).

le dispositif testé) à différentes étapes des calculs produit un certain nombre d'erreurs. Il est possible que le résultat d'une opération mathématique soit erroné, ou simplement qu'une instruction machine soit ignorée.
Se référer aux **figures 4**, **5** et **6** concernant les signaux et les résultats. Un défaut efficace pour la FunCard est une impulsion basse active, d'une durée comprise entre 10 ns et 20 ns, déclenchée environ 10 à 20 ns après le front montant de l'horloge de l'unité centrale (voir **figure 5**).

Démonstration d'attaque par défaut de tension

Il s'agit d'une démonstration d'une attaque élémentaire de type défaut de tension. La FunCard procède aux mêmes opérations que dans l'exemple précédent. Un défaut de sous-tension efficace a une durée d'au moins 50 ns et donne de bons résultats avec le timing illustré sur la **figure 7**. Ici, le défaut s'est déclenché 30 à 40 ns suite au front montant de l'impulsion d'horloge. La tension d'alimentation de la FunCard a été fixée à 2,24 V. Le défaut de tension présentait une profondeur d'environ 1,5 V.
Ajustez la forme du défaut de tension avec P4, P5 et P6. Utilisez P3, S3 et S4 pour régler le retard et la synchronisation du défaut. Les résultats sont affichés de la même manière que dans l'exemple précédent (**figure 8**).

Déverrouillage de la protection de la mémoire pour extraire le micrologiciel

Chaque carte à puce ou microcontrôleur protégé nécessite une procédure d'attaque différente. De nombreux tâtonnements et recherches complexes sont nécessaires pour trouver les points faibles possibles. Adaptée à de nombreux microcontrôleurs AVR Microchip, la procédure s'appuie sur un défaut de sous-tension lent.
Elle fonctionne comme suit. Lorsque le microcontrôleur FunCard (AT90S8515) est protégé, les deux bits de verrouillage de la mémoire sont programmés à 0. À partir de là, la mémoire flash ne peut plus être lue, mais seulement effacée. En mode de programmation série (son entrée de réinitialisation est ramenée au niveau bas 0 V), la commande *Chip Erase* va d'abord effacer la mémoire flash (en positionnant tous les octets qu'elle contient à la valeur 0xFF), puis effacer les bits de verrouillage (en les désactivant pour les ramener à 1).
Le défaut de conception présent sur de nombreux contrôleurs AVR (mais pas tous) permet de déverrouiller la protection de la manière suivante. Si la tension d'alimentation est abaissée au-dessous du minimum nominal (2,7 V), jusqu'à une valeur comprise entre 1,6 V et 1,7 V, il n'y aura pas assez de puissance pour effacer la mémoire flash, même si l'effacement des bits de verrouillage sera toujours possible. L'attaque est lancée à 1,1 V et la tension est progressivement augmentée. Les bits de verrouillage ont été supprimés avec succès à 1,62 V sans effacer la mémoire flash !
Lorsque vous essayez de lire le micrologiciel ou la signature du dispositif à partir d'un microcontrôleur AVR protégé, la réponse sera « 0x00, 0x01, 0x02, 0x03 » et vous savez donc que la mémoire est protégée (**figure 9**). Une fois que vous obtenez la signature correcte (« 0x1E, 0x93, 0x01 » pour le microcontrôleur AT90S8515), les bits de verrouillage de la mémoire ont été supprimés et la mémoire du programme peut ainsi être lue en utilisant n'importe quel programmateur ISP.

```
> 0 : Send a single command to Smartcard
> 1 : Send a single command to MCU
> 2 : Find minimum Vcc2
> 3 : Adjust minimum Vt
> 5 : Funcard no glitch demo
> 6 : Funcard clock glitch demo
> 7 : Funcard volt glitch demo
> 8 : Volt glitch loop test
> 9 : Volt & Clock glitch loop test
> A : Fast Vcc2 test
> B : Brute-force PIN demo, Funcard MCU internal counter
> b : Brute-force PIN demo, Funcard AT24C external counter
> R : Reset MCU & Smartcard

Vcc2 RAW set to:0xDD    Vcc2 feedback: 5,092V
Vt RAW set to:0xFF    Vt scaled set to: 0,452V    Vi feedback: 0,166V
Smartcard fclk division factor set to RAW:0x00

> Select Option (0-X): 6

Clock glitch demo.

Answer To Reset: 0x45 0x67 0x78 0x9A 0xBC 0xDE 0xF0 0x12 0x34 0x56 0x78 0xDE 0xAD 0x01 0x23 0x00 0x00 0x00 0x00 0x00
0xF62F 0x01 0x25 0x25  OK!
0xF34B 0x02 0x3E 0x3E  OK!
0xF62F 0x03 0x25 0x25  OK!
0xF62F 0x04 0x25 0x25  OK!
0xF5E4 0x05 0xD9 0xD9  OK!
0xF62F 0x06 0x25 0x25  OK!
0xF62F 0x07 0x25 0x25  OK!
0xF576 0x08 0x6B 0x6B  OK!
0x48E8 0x09 0x30 0x30  OK!
0x47E8 0x0A 0x2F 0x2F  OK!
0xF676 0x0B 0x6C 0x6C  OK!
0x47E8 0x0C 0x2F 0x2F  OK!
0xF62F 0x0D 0x25 0x25  OK!
0x0000 0x0E 0x00 0x00  OK!
0x5555 0x0F 0x55 0xAA  Error!
0xF62F 0x10 0x25 0x25  OK!
0xF62F 0x11 0x25 0x25  OK!
0xF62F 0x12 0x25 0x25  OK!
0xF62F 0x13 0x25 0x25  OK!
```

Figure 6. Fenêtre de terminal montrant les résultats de l'attaque par défaut d'horloge. La **colonne 1** *contient le résultat sur 16 bits de l'opération mathématique exécutée par le dispositif testé. Il sera erroné si le défaut a réussi.* **Colonne 2 :** *délai de l'attaque. L'application du défaut à des moments différents donne des résultats différents.* **Colonne 3** *: somme de contrôle basée sur la somme des deux octets du résultat sur 16 bits calculé par la FunCard.* **Colonne 4 :** *somme de contrôle calculée par le microcontrôleur du PMCT.* **Colonne 5 :** *« Error! » si les colonnes 3 et 4 ne correspondent pas.*

Attaque par analyse de la consommation d'énergie d'une carte bancaire

Les cartes bancaires sont de plus en plus sophistiquées et utilisent de multiples

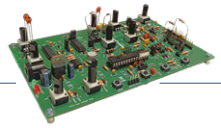
```
> 0 : Send a single command to Smartcard
> 1 : Send a single command to MCU
> 2 : Find minimum Vcc2
> 3 : Adjust minimum Vt
> 5 : Funcard no glitch demo
> 6 : Funcard clock glitch demo
> 7 : Funcard volt glitch demo
> 8 : Volt glitch loop test
> 9 : Volt & Clock glitch loop test
> A : Fast Vcc2 test
> B : Brute-force PIN demo, Funcard MCU internal counter
> b : Brute-force PIN demo, Funcard AT24C external counter
> R : Reset MCU & Smartcard

Vcc2 RAW set to:0x60    Vcc2 feedback: 2,238V
Vt RAW set to:0xFF    Vt scaled set to: 0,452V    Vi feedback: 0,053V
Smartcard fclk division factor set to RAW:0x00

> Select Option (0-X): 7

Volt glitch demo.

Answer To Reset: 0x45 0x67 0x78 0x9A 0xBC 0xDE 0xF0 0x12 0x34 0x56 0x78 0xDE 0xAD 0x01 0x23 0x00 0x00 0x00 0x00 0x00
0xF62F 0x01 0x25 0x25  OK!
0xF62F 0x02 0x25 0x25  OK!
0xF5E4 0x03 0xD9 0xD9  OK!
0xF62F 0x04 0x25 0x25  OK!
0x3F4B 0x05 0x8A 0x8A  OK!
0xF62F 0x06 0x25 0x25  OK!
0xF62F 0x07 0x25 0x25  OK!
0x482F 0x08 0x77 0x77  OK!
0x492F 0x09 0x78 0x78  OK!
0x482F 0x0A 0x77 0x77  OK!
0xF676 0x0B 0x6C 0x6C  OK!
0x482F 0x0C 0x77 0x77  OK!
0xF62F 0x0D 0x25 0x25  OK!
0x0000 0x0E 0x00 0x00  OK!
0xF642 0x0F 0x38 0x38  OK!
0x0000 0x10 0x00 0x00  OK!
0xF62F 0x11 0x2F 0x25  Error!
0xF62F 0x12 0x25 0x25  OK!
0xF62F 0x13 0x25 0x25  OK!
```

Figure 8. Fenêtre de terminal montrant les résultats de l'attaque par défaut de tension. Les colonnes sont formatées de la même façon que pour l'attaque par défaut d'horloge (voir la figure 6).

```
CLEAR  AutoScroll  Reset Cnt  13  Cnt = 1721  HEX  ASCII  StartLog  StopLog  Req

Stage:0x01 Vcc2 RAW:0x43 Init:0x53
    Reading firmware:0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
Reading signature... 0x00 0x01 0x02 0x03
Y-erase the lockbits, N-read again, F-finish?
    Vcc2 feedback: 1,575V

Stage:0x01 Vcc2 RAW:0x44 Init:0x53
    Reading firmwar03 03 x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
Reading signature... 0x00 0x01 0x02 0x
Y-erase the lockbits, N-read again, F-finish?
    Vcc2 feedback: 1,598V

Stage:0x01 Vcc2 RAW:0x45 Init:0x53
    Reading firmware:0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
Reading signature... 0x00 0x01 0x02 0x03
Y-erase the lockbits, N-read again, F-finish?
    Vcc2 feedback: 1,622V

Stage:0x01 Vcc2 RAW:0x46 Init:0x53
    Reading firmware:0x0C 0x1C 0x1A 0x19 0x18 0x17 0x22 0x2C 0x14 0x13
Reading signature... 0x1E 0x93 0x01 0xFF
Y-erase the lockbits, N-read again, F-finish?

Stage:0x01 Vcc2 RAW:0x47 Init:0x53
    Reading firmware:0x0C 0x1C 0x1A 0x19 0x18 0x17 0x22 0x2C 0x14 0x13
Reading signature... 0x1E 0x93 0x01 0xFF
Y-erase the lockbits, N-read again, F-finish?

Transmit
```

Figure 9. Déverrouillage de la protection de la mémoire flash.

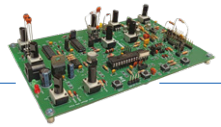
Figure 10. Courant d'alimentation (trace jaune) d'une carte bancaire lors de la vérification d'un code confidentiel incorrect.



méthodes pour protéger physiquement les zones de mémoire critiques. Elles essaient également de masquer le courant d'alimentation afin que les opérations critiques soient plus difficiles à détecter. Si certaines procédures, comme la vérification du code confidentiel saisi, peuvent être localisées avec précision dans le temps, nous savons alors à quel moment des anomalies peuvent être déclenchées pour tenter de contourner les protections.

Voici quelques-unes des méthodes connues de protection des cartes bancaires :

1. Utilisation d'une horloge RC interne pour les opérations de sécurité critiques, passage à une CLK externe uniquement pour une synchronisation précise, par exemple pour la communication via l'UART. Il est ainsi possible d'éviter les attaques de type « défaut d'horloge ».
2. Utilisation de pompes de charge très rapides sur de petits condensateurs d'alimentation internes (de l'ordre du pF) pour maintenir une tension d'alimentation stable. Ceci permet d'éviter les attaques à l'aide de défauts de tension.
3. Modification aléatoire des demi-périodes de l'horloge RC interne (à l'aise d'un générateur de nombres aléatoires interne - TRNG). Ainsi, les opérations critiques (notamment la vérification du code confidentiel) ne se produiront pas toujours au même moment, ce qui rendra les attaques plus difficiles.
4. Ajout d'un bruit aléatoire au courant de l'alimentation électrique. Il est ainsi possible d'éviter les attaques par analyse de l'alimentation (**figure 10**).
5. Lors de la vérification d'un code confidentiel, il convient d'abord de diminuer le compteur de tentatives de code dans l'EEPROM de la carte à puce, puis de vérifier le code et d'augmenter ce compteur de tentatives si ledit code est correct. Les anciennes cartes n'écrivaient que dans l'EEPROM pour diminuer le compteur en cas de saisie erronée du code confidentiel. Ainsi, les attaques par force brute pouvaient extraire le code confidentiel en coupant rapidement l'alimentation électrique après chaque tentative erronée.
6. Utilisation de fonctions très soigneusement conçues pour les opérations critiques relatives à la sécurité (programmation en assembleur requise ici !), qui nécessitent toujours le même nombre de cycles d'horloge de l'unité centrale et (éventuellement) la même quantité d'énergie, quelles que soient les variables d'entrée. L'approche permet d'éviter les attaques de synchronisation et par analyse de la consommation d'énergie.



7. Prolongation d'une opération critique pour la sécurité à 200 ms, par exemple, contre 1 ms à l'origine. Un signal utile de 1 ms est donc caché dans un délai de 199 ms servant au traitement de données inutiles.

La mise hors d'état de fonctionner d'une carte bancaire moderne de manière non invasive (si tant est que cela soit possible) nécessite une analyse approfondie des données enregistrées et un travail long et minutieux. Pour des informations plus complètes et détaillées sur les attaques par analyse de la consommation d'énergie, lisez l'ouvrage spécialisé *Power Analysis Attacks* [5].

Tester et découvrir

L'outil PMCT n'est pas un appareil high-tech, mais il peut servir à tester et découvrir des attaques non invasives sur les HSM. J'espère que vous avez trouvé cet article intéressant, au moins comme base pour vous lancer. De nos jours, la conception et l'attaque des HSM monopuce représentent un travail très difficile pour les personnes concernées. C'est ce qui explique que ce domaine reste ouvert à de nouvelles recherches.

Bon espionnage !

VF : Pascal Godart — 210462-04

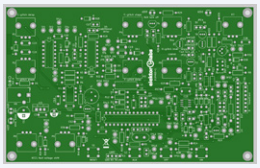
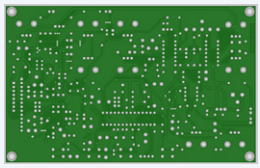
Des questions, des commentaires ?

Envoyez un courriel à l'auteur (luka.matic@fer.hr) ou contactez Elektor (redaction@elektor.fr).

Produits

- **LabNation SmartScope oscilloscope USB (SKU 17169)**
  www.elektor.fr/17169
- **Luka Matic, *Livre en anglais « Electronic Security and Espionage »*, Elektor 2021 (SKU 19903)**
  www.elektor.fr/19903

LISTE DES COMPOSANTS

Résistances

R1,R29,R35 = 680 Ω
R2,R34 = 47 kΩ
R3,R5,R6,R7,R9,R23,R25,R33,R38 = 10 kΩ
R4,R8,R13 = 100 kΩ
R10 = 1,5 kΩ
R11,R12,R17,R22,R24,R26,R27,R32,R36 = 1 kΩ
R14,R28,R30 = 4,7 kΩ
R15 = 330 Ω
R16,R19 = 33 kΩ
R18 = 2,2 kΩ
R20* = 22 Ω*
R21* = 10 kΩ*
R31,R37 = 100 Ω
R39,R40 = 3,3 kΩ
P1,P2,P3,P5 = potentiomètre, 5 kΩ, vertical
P4 = potentiomètre, 10 kΩ,vertical
P6 = potentiomètre, 500 Ω, vertical
P7 = potentiomètre, 2 kΩ, vertical
P8,P9 = potentiomètre, 25 kΩ,vertical

Condensateurs

C1 = 100 µF 16 V, pas de 3,5 mm
C2 = 10 µF 16 V, pas de 2,5 mm
C3,C4,C7,C8,C9,C10,C11,C12,C20,C26,C28,C29 = 100 nF, pas de 5 mm
C5,C6,C17,C22,C25 = 22 pF, pas de 2,5 mm
C13,C24 = 100 pF, pas de 2,5 mm
C14* = 10 nF, pas de 5 mm*
C27 = 10 nF, pas de 5 mm
C15*,C16* = 22 pF, pas de 5 mm*
C18* = 33 pF, pas de 5 mm*
C19* = 100 pF, pas de 5 mm*
C21 = 220 pF, pas de 2,5 mm
C23 = 1 µF, pas de 5 mm

Inductances

L1 = 10 µH

Semi-conducteurs

D1,D2,D3,D4,D5,D13,D14,D20,D21 = 1N4148
D11* = 1N4148*
D6,D7,D8,D9,D10,D15,D16,D17,D18,D19 = BAT85
D12* = BAT85*
IC1 = LM358
IC2 = 74HC00
IC3 = 74HC14
IC4 = LMV761, SOIC8
IC5 = 7805, TO220
C6* = ATmega8A-PU*
LED1,LED2 = LED, 3 mm, verte
LED3 = LED, 3 mm, jaune
LED4 = LED, 3 mm, rouge
T1 = BC141, TO39
T2 = BC557C
T3,T4,T5,T8,T9 = PN2369, TO92
T6 = BC547C
T7 = 2N2369, TO18

Divers

JP1-JP13,JP15 = barrette 2 broches, pas de 2,54 mm
JP14,S3,S4 = barrette 3 broches, pas de 2,54 mm
K1 = barrette 8 broches, pas de 2,54 mm
K2 = barrette 14 broches, pas de 2,54 mm
K3 = barrette 4 broches, pas de 2,54 mm
K4,K5 = barrette 2 rangées, 6 broches, pas de 2,54 mm
K6 = embase jack
S5 = commutateur à glissière (C&K JS202011CQN)
X1* = quartz 8 MHz*

* montage avec support pour faciliter l'expérimentation

Funcard. (Source : www.cellularcenter.it)

LIENS

[1] Luka Matic, « Boîte inviolable protégée par un témoin d'effraction », Elektor 5/ 2020 : https://www.elektormagazine.fr/magazine/elektor-148/58644
[2] ChipWhisperers from NewAE Technology Inc. : https://www.newae.com/chipwhisperer
[3] Sergei P. Skorobogatov, « Copy Protection in Modern Microcontrollers » : https://www.cl.cam.ac.uk/~sps32/mcu_lock.html
[4] Téléchargements pour cet article depuis Elektor Labs : https://www.elektormagazine.fr/labs/poor-mans-chipwhisperer-or-a-smartcard-tweaker
[5] S. Mangard, E. Oswald, T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards, Springer, 2007 : https://amzn.to/3RWBtuy

MagPi, le magazine **officiel** du Raspberry Pi

12 mois
Plus de
100 projets
Le prix
54,95 €

- **6 X MAGPI : ÉDITION IMPRIMÉE**
- **ACCÈS AUX ARCHIVES EN LIGNE DU MAGPI**

COMMANDEZ DÈS MAINTENANT AU
WWW.MAGPI.FR/ABO

MagPi
www.magpi.fr Magazine