Architecture temps-réel pour l'interception de signaux Bluetooth porteurs de compromission

Corentin Lavaud, Robin Gerzaguet, Matthieu Gautier, Olivier Berder

Architecture temps-réel pour l'interception de signaux Bluetooth porteurs de compromission

Corentin LAVAUD[1] Robin GERZAGUET[1] Matthieu GAUTIER[1] Olivier BERDER[1]

[1]Univ Rennes, CNRS, IRISA
Rue Kérampont, Lannion, 22300, France

Résumé – Cet article propose une architecture temps réel dédiée à l'interception de signaux Bluetooth en vue de procéder à une analyse de vulnérabilités d'un canal caché. Ces travaux s'inscrivent dans le contexte dit de "TEMPEST télécom" où un signal légitime porteur d'information est couplé avec une compromission souvent bande étroite (signal audio, clef de chiffrement, ...). On s'intéresse aux systèmes Bluetooth dont la couche physique utilisant le saut de fréquence rend la détection et l'exploitation de ce canal caché plus complexe. Comme la séquence de saut est inconnue par le récepteur réalisant l'analyse, nous décrivons une architecture matérielle temps-réel à base de radio logicielle capable de détecter le canal Bluetooth utilisé et d'extraire la vulnérabilité. Enfin, nous dévoilons une vulnérabilité sur une puce nRF52832 où un signal issu d'une PWM interne est extrait via la réception du signal Bluetooth.

Abstract – In this article, a real-time architecture dedicated to the interception of Bluetooth signals is proposed in order to carry out a vulnerability analysis of a side channel. This work falls within the framework of "TEMPEST telecom," where a legitimate signal carrying information is coupled with a compromise, mainly a narrowband one (audio signal, encryption key, ...). We focus on Bluetooth systems whose physical layer based on frequency hopping makes the detection and exploitation of this hidden channel more complex. As the hop sequence is unknown by the receiver performing the analysis, we describe a real-time hardware architecture based on software-defined radio. This architecture is capable of detecting the used Bluetooth channel and extracting the vulnerability. Finally, we reveal a vulnerability on an nRF52832 System On Chip where a signal from an internal PWM is extracted via reception of the transmitted Bluetooth signal.

1 Introduction

Parmi la longue liste de menaces de cybersécurité potentielles, la vulnérabilité TEMPEST conduit à un des scénarios les plus problématiques : il se produit lorsque des données confidentielles sont émises involontairement en raison de la présence d'un canal non légitime. Ce canal peut avoir une nature différente (lumière, son ou électromagnétique) et être dû à différentes causes (fuites électromagnétiques, couplage, exfiltration volontaire, ...) [8].

Différentes attaques ont ciblé les canaux cachés électromagnétiques et se caractérisent généralement par une interception en champ proche. Il a récemment pu être démontré que les transmissions sans fil légitimes (par exemple, Wi-Fi ou Bluetooth) peuvent également cacher des canaux cachés, ce qui augmente le risque de compromission [2, 3] notamment car la portée d'interception se retrouve augmentée.

Au sein d'un canal caché télécom, la récupération de la vulnérabilité nécessite la capacité de détecter le signal légitime. Cette tâche peut s'avérer complexe, en particulier dans le cas des signaux à saut de fréquence. Cette méthode permet de transmettre des signaux radio en modifiant la fréquence porteuse parmi plusieurs fréquences disti

Il y a plusieurs défis liés à l'interception des signaux à saut de fréquence. Tout d'abord, la séquence de saut peut être très longue et impossible à estimer, dans le cas du TRANSEC par exemple [4]. Dans ces cas, la seule solution est de détecter le canal utilisé en temps réel. De plus, la largeur de bande de transmission peut être très grande (80 MHz pour le Bluetooth) ce qui rend l'interception en temps réel difficile ou coûteuse. Enfin, un autre défi lié au scénario TEMPEST est l'extraction du signal confidentiel.

Dans cet article, nous proposons une architecture matérielle et logicielle portée sur une radio logicielle X310 et pouvant capturer jusqu'à 160 MHz de bande, d'extraire des sous-bandes de 150 kHz et de suivre jusqu'à 50 000 sauts par seconde. Cette architecture utilise les ressources FPGA de la radio logicielle ainsi qu'un environnement logiciel basé sur le langage Julia [6]. Après avoir décrit le modèle numérique du saut de fréquence et l'architecture temps-réel, nous montrons qu'un système sur puce sur lequel une broche PWM émet un signal peut entraîner la présence d'un canal caché et que l'activité de la broche peut être retrouvée par un récepteur qui n'est pas appairé avec le système. Nous démontrons ainsi la nécessité de revoir l'indice de menace associé avec les canaux cachés télécoms.

2 Système d'interception

2.1 Modèle du signal à saut de fréquence

On considère le modèle du signal à saut de fréquence qui peut s'exprimer de la manière suivante :

$$x(t) = b(t)e^{2j\pi f_p(t)t},\ p \in [0, \dots N-1], \qquad (1)$$

où $f_p(t)$ est la fréquence porteuse occupée par le signal $x(t)$ au temps t et associée au canal d'indice p avec N le nombre total de canaux et où $b(t)$ est le signal bande de base. Ce signal peut s'exprimer dans le cas de la présence d'un canal caché

comme étant :

$$b(t) = b_b(t) + h \times r(t), \tag{2}$$

avec $b_b(t)$ est le message d'information légitime sur une bande F_b (par exemple 2 MHz pour le Bluetooth *low energy*), $r(t)$ le signal caché de bande F_r avec $F_r < F_b$ (quelque kHz pour un signal audio) et h le coefficient de couplage. La bande totale du système à saut de fréquence est $F_c = N \times F_b$. Dans le cas du Bluetooth, on a ainsi $N = 40$ canaux conduisant à une bande totale F_c de 80 MHz. Le signal est ensuite transmis sur une fréquence porteuse (typiquement 2,4 GHz pour le Bluetooth).

2.2 Système d'interception

Le signal est traité par une radio logicielle qui procède à la démodulation. On dispose donc d'un signal reçu $d(t)$ échantillonné au débit maximal F_c tel que $d[u] = d(uT_c)$ (avec $T_c = 1/F_c$). Nous proposons sur la Figure 1 un détecteur basé sur une analyse fréquentielle court terme à base de transformée de Fourier rapide (FFT) qui réalise également l'extraction du signal. Ce détecteur est basé sur une simplification du détecteur proposé dans [7] avec pour objectif de :

- s'appuyer sur une taille de transformée de Fourier Rapide (FFT) $N_{\text{FFT}} > N$ qui soit une puissance de 2 pour profiter d'une implémentation efficace et ne pas être tributaire du nombre de canaux du standard à saut de fréquence considéré,
- rendre la structure moins dépendante d'une synchronisation fine avec les intervalles de temps de saut de fréquence,
- combiner la phase d'analyse fréquentielle avec celle qui permet l'extraction du canal d'intérêt sous hypothèse que l'information cachée est mélangée au canal légitime par une modulation d'amplitude. Cette hypothèse est vraie dans le cas des canaux TEMPEST où les compromissions sont dues à la proximité des composants et se traduisent par le modèle décrit dans (2).

La structure proposée est nommée FFT-BE pour *Bin Extraction* et pour chaque intervalle de temps k le détecteur est défini par :

$$\hat{p}[k] = \underset{p\in[0;N_{\text{FFT}}-1]}{\arg\max} \left\{ \left| \sum_{m=0}^{N_{\text{FFT}}-1} d[kN_{\text{FFT}}+m] e^{-\frac{2j\pi mp}{N_{\text{FFT}}}} \right|^2 \right\}, \tag{3}$$

et le signal extrait comme étant :

$$\hat{d}_{\hat{p}[k]}[k] = \left| \sum_{m=0}^{N_{\text{FFT}}-1} d[kN_{\text{FFT}}+m] e^{-\frac{j2\pi m\hat{p}[k]}{N_{\text{FFT}}}} \right|^2. \tag{4}$$

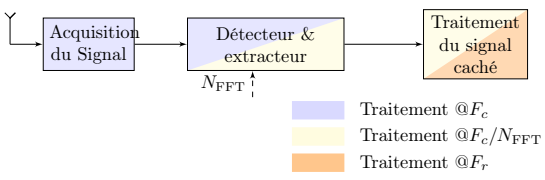

FIGURE 1 : Schéma du dispositif d'interception.

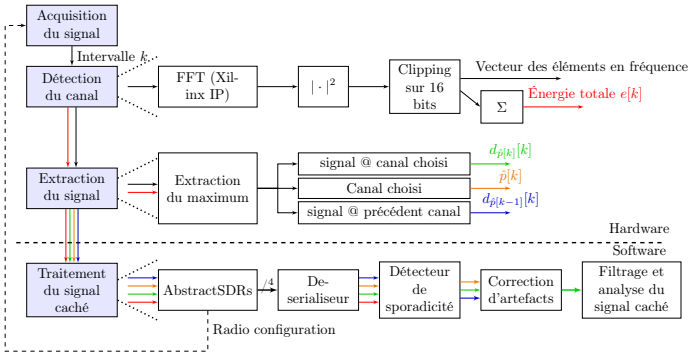

FIGURE 2 : Schéma de partitionnement proposé.

Ces deux expressions ne nécessitent qu'une seule utilisation de FFT ce qui permet de réduire drastiquement la complexité et d'envisager une implémentation matérielle qui traite de très larges bandes. Il est à noter que ces simplifications conduisent à une réduction des performances de détection théoriques, qui sont détaillées dans [5] mais qui ne sont pas l'objet de la contribution présentée dans cet article.

3 Implémentation du système d'interception

3.1 Partitionnement matériel/logiciel

Les contraintes temps réel sont strictes pour la partie de détection du canal et l'extraction de la sous bande. Par ailleurs, pour le traitement de la donnée cachée, une plus grande flexibilité est souhaitée afin d'adapter les algorithmes à la nature du signal à analyser. On se propose donc d'utiliser les ressources matérielles de la radio logicielle X310 pour accélérer la partie des traitements qui requiert la large bande et de procéder à l'analyse du signal en logicielle via une approche mêlant prototypage et bonnes performances en Julia [6].

Le partitionnement est décrit sur la Figure 2. On remarque bien les blocs matériels dédiés à l'extraction du canal d'intérêt via l'analyse fréquentielle avec $N_{\text{FFT}} = 1024$. Dans la partie matérielle, nous extrayons l'indice du canal, les informations du canal et la puissance reçue dans le canal choisi. Ceci permet de ne pas considérer des intervalles de temps où la transmission n'a pas eu lieu (détecteur de sporadicité) via un seuillage dynamique. Notre architecture est capable de détecter des sauts de durée $N_{\text{FFT}}/Fs = 6.4\,\mu s$ ce qui est bien inférieur à la durée classique de saut de fréquence (625 μs en Bluetooth). En pratique, un canal $\hat{p}[k]$ sera donc souvent identique sur plusieurs intervalles consécutifs. Afin d'améliorer les performances et de corriger des potentiels artefacts, nous décidons d'extraire également les données qui correspondent au canal précédemment sélectionné. Dans la partie *correction d'artefact* nous sélectionnons le signal associé au canal choisit précédent dans le cas où on aurait choisi un canal $\hat{p}[k]$ différent des C voisins précédents et suivants. Ceci implique une latence additionnelle négligeable pour notre architecture (25 μs pour l'implémentation considérée avec C=2) mais permet de corriger des artefacts dus à la présence du bruit.

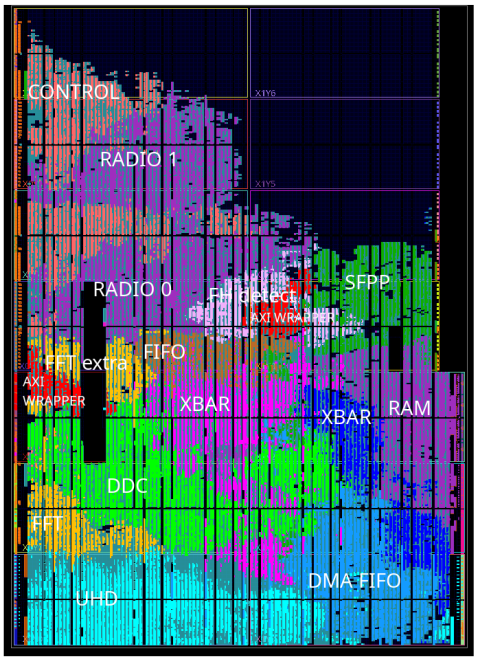

FIGURE 3 : Carte du design matériel proposé sur la X310.

3.2 Accélération matérielle via FPGA

La partie matérielle est placée sur le FPGA de la radio logicielle X310 qui est un Kintex 7-410T. Pour interfacer les blocs proposés, nous nous appuyons sur le fonctionnement de RFNoC de UHD [1]. Chaque bloc est instancié sous forme d'IP puis interfacé via les inter-connections AXI du pilote de UHD. Le design est décrit sur la Figure 3. Les deux têtes avant de la radio (en violet) sont synthétisées même si notre application utilise seulement Radio-0. L'étage de conversion de fréquence et filtrage est représenté en vert clair et est désigné sous le nom de DDC. L'interface entre la X310 et l'ordinateur hôte pour le traitement du signal caché est visible en vert foncé avec une interface SFPP. Différentes configurations et interfaces de contrôle sont visibles (en rouge clair en haut, AXI wrapper en rouge foncé, UHD en cyan, XBAR en rouge et bleu) ont également été synthétisées pour assurer la configurabilité depuis l'ordinateur hôte. De grandes zones de mémoire ont été instanciées dans le coin droit en violet et bleu (DMA FIFO et RAM) pour stocker les échantillons numériques.

La Table 1 détaille les ressources utilisées occupés par notre design avec, en particulier les LUT, les multiplieurs DSP 48 bits et les cellules. Le détecteur utilise plus de ressources de calcul (FFT oblige) mais la surface de l'extracteur est légèrement plus grande que celle du détecteur. On remarque enfin que plus de la moitié de la surface du FPGA est occupée mais principalement due au système d'interface de UHD.

	LUT	DSP48	Cellules
Détecteur	5823 (5.22%)	52 (3.37%)	21,047 (8.68%)
Extracteur	7732 (6.93%)	0 (0%)	37,766 (14.74%)
Total	111,582 (43.90%)	245 (15.91%)	242,610 (59.65%)

TABLE 1 : Utilisation des ressources FPGA.

Nous avons pu valider le bon comportement de l'architecture via l'utilisation d'un générateur de signal arbitraire.

4 Mise en évidence de vulnérabilités sur le Bluetooth

Nous avons essayé de provoquer des fuites à différents endroits dans une puce intégrée (SoC) nRF52832 avec Bluetooth et observé dans quelle mesure il était possible d'induire un canal caché et si notre système d'interception était capable de le récupérer. Le dispositif Bluetooth n'est pas modifié contrairement aux approches classiques de la littérature [2] et la compromission n'a pas été forcée par un moyen externe. Les fuites proviendront des dispositifs eux-mêmes en raison de leur comportement opérationnel. Afin de détecter une compromission, nous avons programmé le SoC pour effectuer de manière indépendante une tâche de transmission Bluetooth et des tâches courantes telles que la lecture d'un flux audio, la mise en marche d'une LED ou l'acquisition d'un signal analogique. Le signal Bluetooth est ensuite capté par notre radio logicielle et nous vérifions si une fréquence caractéristique du signal associé à ces tâches courantes est présente dans le signal Bluetooth.

4.1 Attaque sur une chaine audio

Cette première expérience illustre le scénario d'un système audio avec une connexion Bluetooth. En général, dans un tel système, un appareil externe envoie le flux audio à lire via Bluetooth. Le signal Bluetooth est envoyé périodiquement car le flux audio a un débit constant. La configuration de cette expérience est présentée dans la Figure 4. Le SoC génère une modulation de largeur d'impulsion (PWM) à 1000 Hz vers un amplificateur de puissance audio de classe D externe, ce dernier étant ensuite connecté à deux haut-parleurs de 3 W. La sortie PWM est connectée à la fois à l'amplificateur audio et à une LED pour fournir un indicateur de fonctionnement.

On observe sur la Figure 5 le spectre différentiel c'est à dire la différence entre les densités spectrales de puissance en absence de chaîne audio et en présence de chaîne audio. On observe bien la présence d'un pic à la fréquence de fonctionnement de l'audio ce qui démontre la présence du signal rouge

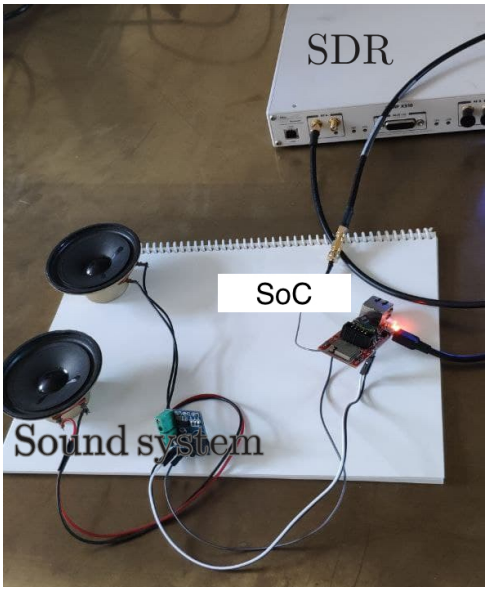

FIGURE 4 : Schéma de l'interception de signal audio.

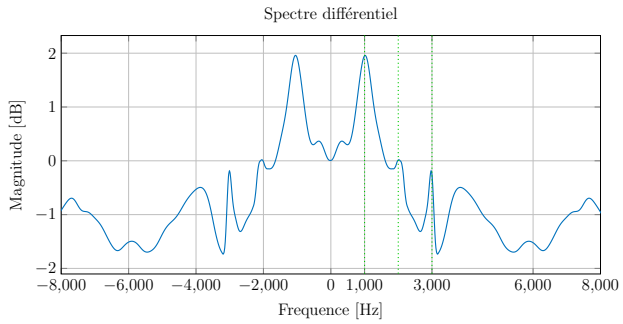

FIGURE 5 : Mise en évidence de la vulnérabilité audio.

et donc de la compromission.

4.2 Attaque sur une LED

Dans un second temps, nous déconnectons l'amplificateur mais gardons la LED connectée. On remarque un spectre différentiel tout à fait similaire à celui de la Figure 5 mais avec un pic d'amplitude à 4.2 dB, plus important qu'avec l'ampli audio. On peut supposer que l'amplificateur audio qui était précédemment connecté comprend des condensateurs à grande capacité afin de compenser le bruit généré dans l'alimentation électrique et réduit ainsi probablement l'amplitude du canal caché.

4.3 Attaque sur une patte déconnectée

Enfin, la dernière expérience ne comprend qu'un signal PWM qui est envoyé à une sortie GPIO connectée à rien. Dans ce cas, le signal récupéré correspondant est montré dans la Figure 6. Il montre qu'une fuite est toujours présente à la même position que celle avec la LED ou le signal audio de la section précédente, mais son amplitude a été considérablement réduite à 0.8 dB, ce qui indique que la LED fuyait effectivement mais n'est pas la seule à causer des fuites.

Il est à noter que cette vulnérabilité, quoique moins marquée dispose d'un indice de menace plus élevé. Ceci implique en effet que des informations peuvent être exfiltrées via une action malicieuse sur une patte qui n'est pas connectée à un dispositif externe. Elle démontre également bien que les activités de traitement couplée à une transmission Bluetooth peuvent engendrer la présence de canaux cachés télécoms.

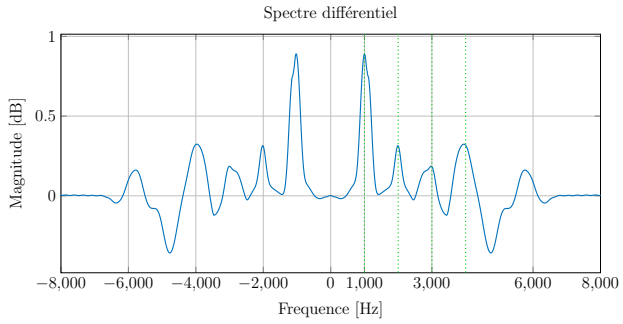

FIGURE 6 : Mise en évidence de la vulnérabilité sur une patte PWM.

5 Conclusion

Cet article présente une architecture dédiée à l'analyse de canaux cachés TEMPEST télécom. Nous étudions les signaux à sauts de fréquence car ils sont plus difficiles à suivre en raison de la fréquence changeante de leur porteuse. Après avoir dérivé les modèles de fuites et de télécommunications, un système d'interception en temps réel a été proposé en utilisant une radio logicielle. Nous montrons que notre architecture est capable d'analyser la bande passante de 160 MHz de la radio logicielle X310 en temps réel et est capable de suivre un très grand nombre de saut par secondes. Enfin, nous avons démontré que des sorties audio et des broches PWM peuvent entraîner des fuites dans les dispositifs Bluetooth, ce qui permettrait de transmettre des données sensibles. Il est donc opportun de revoir l'indice de menace induit par les canaux cachés télécom et que les systèmes à saut de fréquences peuvent être ciblés même sans connaître la séquence de sauts.

Remerciements

Ce travail a été financé par l'intermédiaire d'une bourse CREACH Lab et par le projet ANR-22-CE25-0007-01 (ANR RedInBlack).

Références

[1] M. BRAUN, J. PENDLUM et M. ETTUS : RFNoC : RF network-on-chip. *In GNURadio Conference*, 2016.

[2] G. CAMURATI, S. POEPLAU, M. MUENCH, T. HAYES et A. FRANCILLON : Screaming Channels : When Electromagnetic Side Channels Meet Radio Transceivers. *In ACM Conference on Computer and Communications Security (CCS)*, 2018.

[3] J. CHOI, H.-Y. YANG et D.-H. CHO : TEMPEST comeback : A realistic audio eavesdropping threat on mixed-signal SoCs. *In ACM Conference on Computer and Communications Security (SIGSAC)*, 2020.

[4] B.J. HAMILTON : SINCGARS System Improvement Program (SIP) specific radio improvements. *In IEEE Tactical Communications Conference. Ensuring Joint Force Superiority in the Information Age*, 1996.

[5] C. LAVAUD : *Reconfigurable systems for the interception of compromising sporadic signals*. Thèse de doctorat, Univ Rennes, 2022.

[6] C. LAVAUD, R. GERZAGUET, M. GAUTIER et O. BERDER : AbstractSDRs : Bring down the two-language barrier with Julia Language for efficient SDR prototyping. *IEEE Embedded Systems Letters*, 13:166–169, Dec. 2021.

[7] C. LAVAUD, R. GERZAGUET, M. GAUTIER, O. BERDER, E. NOGUES et S. MOLTON : Toward Real time interception of Frequency Hopping Signals. *In Proc. IEEE International Workshop on Signal Processing Systems*, 2020.

[8] C. LAVAUD, R. GERZAGUET, M. GAUTIER, O. BERDER, E. NOGUES et S. MOLTON : Whispering devices : A survey on how side-channels lead to compromised information. *Journal of Hardware and Systems Security*, 2021.