Reconfigurable systems for the interception of compromising sporadic signals

Corentin Lavaud

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Télécommunications*

Par
Corentin LAVAUD

Reconfigurable systems for the interception of compromising sporadic signals

Thèse présentée et soutenue à Lannion, le 06 janvier 2022
Unité de recherche : IRISA (UMR 6074)

Composition du Jury :

Rapporteurs :	Tanguy RISSET Inbar FIJALKOW	Professeur des universités - INSA Lyon Professeur des universités - ENSEA
Examinateurs :	Aurélien FRANCILLON	Professeur des universités - EURECOM
Directeur de thèse :	Olivier BERDER	Professeur des universités - Université Rennes 1
Co-encadrants de thèse :	Robin GERZAGUET Matthieu GAUTIER	Maître de conférences - Université Rennes 1 Maître de conférences (HDR) - Université Rennes 1
Invité :	Erwan NOGUES	DGA-MI

Contents

Chapitre **ZÉRO**

RÉSUMÉ ÉTENDU

1 Contexte général

La sécurité des informations est une problématique multi-latérale, multi-modale et multi-cible. Limiter les risques de fuites d'information passe par le suivi de bonnes pratiques comme le chiffrement des données (données *noires*), la vérification des droits d'accès ou encore la gestion de crises. La majorité des préoccupations en termes de sécurité se concentre sur les risques numériques. Cependant d'autres risques souvent ignorés existent, tels que ceux introduits par les canaux auxiliaires. Un canal auxiliaire consiste en la présence d'une information (que l'on considérera confidentielle et que l'on qualifiera de *rouge*) dans un canal de propagation illégitime. Une représentation de ce type de fuite est visible dans la Fig. 0-1 où l'information atteint non seulement son destinataire légitime, mais également un attaquant, sous une forme altérée. Ce dernier doit alors réaliser des traitements spécifiques afin de reconstituer la donnée initiale.

Cette thèse est financée par le Pôle d'excellence Cyber (PEC) en collaboration avec la DGA. Les recherches qui y sont menées résultent d'un besoin inhérent aux audits de sécurité menés par la DGA. Ces derniers visent à rechercher des fuites dans des systèmes afin d'éliminer les risques de compromission. Si de nombreuses études portent sur la détection de fuites statiques, les fuites dynamiques ou sporadiques, c'est-à-dire celles pouvant changer de fréquence ou disparaître temporairement, sont beaucoup moins traitées. Il est cependant nécessaire d'avoir des méthodes spécifiques permettant d'intercepter ce type de signaux afin de s'assurer qu'aucune compromission n'est présente. Le but de cette thèse et de répondre à ce besoin en réalisant un système d'interception de signaux sporadiques.

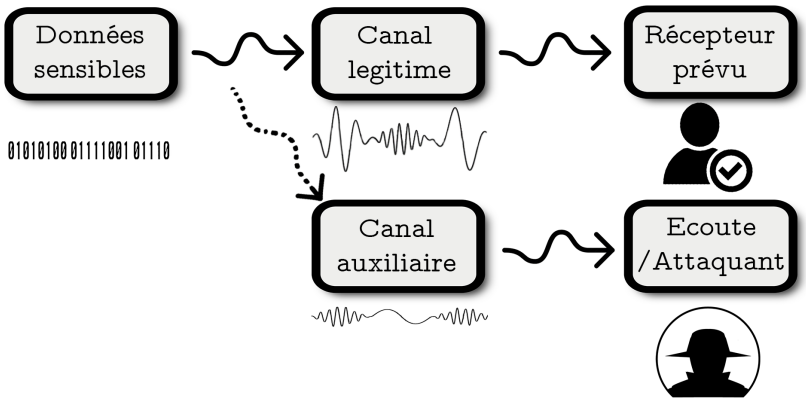


FIGURE 0-1 – Description d'une attaque par canal auxiliaire.

Après avoir proposé dans la Section 2 une classification des canaux auxiliaires, la Section 3 aborde l'état de l'art des détecteurs de signaux pouvant être adaptés au système d'interception. Elle est ensuite dédiée à l'évaluation des performances des détecteurs en considérant leur capacité à détecter un canal et à correctement extraire une donnée rouge suivant plusieurs scénarii. La Section 4 aborde les problématiques matérielles sur l'interception de signaux large bande et proposera une architecture à base de radio logicielle permettant l'interception de signaux en temps réel. Enfin, la Section 5 aborde l'utilisation du système d'interception sur diverses cibles matérielles afin de tester à la fois ses performances, son adéquation avec les expériences simulées, mais aussi la détection de signaux compromettants placés au sein de dispositifs à saut de fréquence. Pour finir la Section 6 résume les différents éléments présentés dans ce manuscrit.

2 Les canaux auxiliaires

Les canaux auxiliaires représentent un risque pour la sécurité de l'information et peuvent apparaître sous plusieurs supports : une vibration, de la lumière, le temps entre le début et la fin d'une opération, un rayonnement électromagnétique, une consommation de courant … Parmi tous les canaux auxiliaires possibles, deux catégories peuvent se dégager vis-à-vis des moyens nécessaires à un attaquant pour les exploiter :

— La catégorie *logicielle* où ces failles sont caractérisées par leur confinement au sein d'un dispositif et sont initiées et exploitées par le biais d'un programme. Des exemples connus de ce type sont les attaques Spectre [Koc18] et Meltdown [Lip20] qui exploitent une

multitude de particularités des architectures des CPU modernes telles que le délai de réponse d'un système ou sur la rémanence des données.

— La catégorie *émanation* où ces failles sont caractérisées par le fait qu'elles sortent des frontières du dispositif fautif. L'attaquant doit alors disposer de dispositifs d'acquisition spécifique au support du canal auxiliaire.

Les travaux effectués dans cette thèse s'intéressent aux canaux auxiliaires à émanation. Ces derniers peuvent résulter de plusieurs causes. Elles peuvent être issues du fonctionnement normal d'un système tel un courant passant dans un câble qui va générer un rayonnement électromagnétique. Si ce courant est un flux vidéo passant dans un câble, il est alors possible d'en reconstituer l'image en interceptant le rayonnement [Kuh04].

Par définition, toutes les données peuvent être considérées comme sensibles ou confidentielles et ne doivent donc pas être récupérées par un adversaire. On peut néanmoins nuancer ce fait et définir certaines informations comme étant moins critiques que d'autres. La faible criticité peut être attribuée soit en raison de la nature de l'information, soit parce que l'information est censée être protégée (par exemple, en utilisant le chiffrement). C'est le cas des données dites **noires** à l'opposé des données **rouges** qui représentent un risque plus élevé si elles sont interceptées par un attaquant.

Les canaux auxiliaires partagent le point commun qu'ils n'ont pas été prévus initialement lors de la conception des systèmes, ce qui implique que les fuites ont des portées relativement faibles. Il y a cependant un type de fuite qui ne possède pas ce défaut : celui affectant les transmetteurs radio. En effet ces fuites se propagent sur les ondes émises et par conséquent ont une portée importante.

3 Étude des intercepteurs de signaux FH et récupération de données sensibles

Afin de détecter des canaux auxiliaires qui affectent des transmetteurs radio, il faut commencer par acquérir le signal radio, puis procéder à une analyse des informations qu'il contient afin de s'assurer que des signaux supplémentaires sont présents. Cette récupération peut se révéler très complexe dans le cadre d'un signal sporadique, c.-à-d. un signal qui change de fréquence porteuse et/ou n'est pas présent continuellement du fait de la large bande de fréquence à observer et de la potentielle disparition du signal. Ces types de signaux sont appelés signaux à

saut de fréquence (ou *Frequency Hopping* FH) : ces signaux sont présents dans un canal (une sous partie de la bande totale d'apparition) et ce canal change (ou disparaît) en suivant une séquence pseudo-aléatoire.

3.1 Comment détecter un signal à saut de fréquence ?

Il existe plusieurs moyens d'intercepter des signaux FH, notamment les techniques de sondage de spectre (méthodes souvent utilisées dans les radios cognitives [Mit02]). Pour ces dernières méthodes, leur adaptation à la détection de canal FH revient à les utiliser de manière duale, au sens où on cherche un canal utilisé, là où le sondage de spectre recherche plutôt les bandes où il y a le moins d'interférences.

Afin de détecter un signal FH, il faut subdiviser le spectre radio écouté en sous-bandes et on considère qu'un signal FH peut être émis dans n'importe laquelle de ces sous-bandes. Idéalement, il convient d'aligner ces sous-bandes avec les canaux du système FH. Il existe plusieurs méthodes permettant de détecter la présence d'un signal dans un canal. La méthode la plus simple et rapide est la détection par énergie [Urk67] où un canal est sélectionné à partir d'un seuil statique ou dynamique. La détection utilisant les propriétés statistiques sur les signaux comme la covariance [Ber00] et la cyclostationarité [Gar91] nécessite de connaître des informations sur le signal FH. S'il y a des parties du signal émis qui sont récurrentes (comme un préambule par exemple), il est alors possible d'utiliser ces récurrences comme une signature afin de les détecter [Zha14a]. Il est également possible d'utiliser la forme d'onde propre à un signal FH afin de le différencier [Tan05]

Ces méthodes nécessitant des connaissances *a priori* sur les signaux ne sont pas adaptées à des méthodes à l'aveugle. La méthode par énergie est donc, pour notre cas, la plus pertinente.

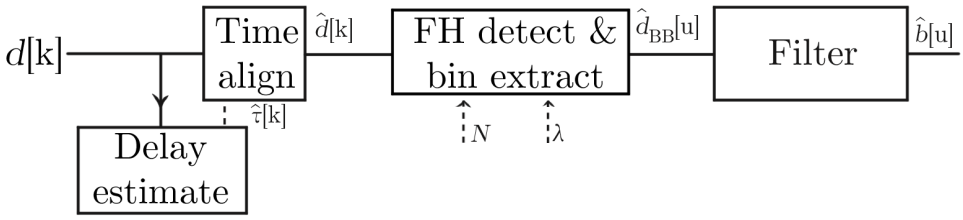


Figure 0-2 – Algorithme de détection DFT-BE.

Le système d'interception proposé est visible dans la Fig. 0-2 où la détection de canaux (*FH*

detect & bin extract) est effectuée en réalisant une transformée de Fourier de même taille que le nombre de canaux possible N. Dans cette configuration, l'énergie de chaque bin de FFT est alors l'énergie d'un canal et l'argument maximal du spectre correspond alors au canal d'intérêt sélectionné $\hat{p}$:

$$\hat{p} = \underset{p\in[0;N-1]}{\arg\max} \left\{ \sum_{l=0}^{\lambda-1} \left| \sum_{m=0}^{N-1} \hat{d}[lN+m] e^{\frac{-j2\pi mp}{N}} \right|^2 \right\}. \tag{0-1}$$

L'algorithme travaille donc avec des fenêtres successives de N échantillons. Si cette acquisition est plus rapide que la vitesse de saut, il est possible d'effectuer un moyennage de taille λ permettant d'améliorer les performances à faible rapport signal sur bruit (Signal-to-Noise Ratio (SNR)). Le signal bande de base est obtenu en extrayant la valeur du bin de la DFT où un canal est détecté (d'où le nom de notre algorithme : DFT-BE). Cette méthode permet de réaliser une décimation du signal (de rapport N). Afin d'améliorer les performances de détection, une étape de synchronisation est ajoutée pour aligner la fenêtre de la DFT avec les instants de saut du signal FH.

Une autre version de cet algorithme, le FFT-BE, est optimisée pour la vitesse de traitement afin de pouvoir être utilisée au sein d'un FPGA au débit maximal de la radio. Les mêmes principes de détection sont utilisés, avec une transformée de Fourier rapide (FFT) (la taille de la FFT est une puissance de deux et une architecture à base de Radix 4 est choisie). Le moyennage sur λ est retiré et avec lui la nécessité de réaliser une synchronisation temporelle. Les performances à bas rapport signal à bruit seront donc impactées, mais on a pu montrer que l'impact sur les performances (dû la désynchronisation entre les sauts et la fenêtre de la FFT) est négligeable dans notre contexte d'utilisation.

3.2 Performances des détecteurs

Une comparaison des performances de la DFT-BE est effectuée dans le Chapitre 4 avec deux autres algorithmes issus de l'état de l'art. Le premier utilise une strucutre PPN (*Polyphase network*) [F-B08] pour séparer le spectre en sous-bandes. Le second repose sur la transformée en ondelettes [Zha14b] pour détecter la présence de signaux FH .

Pour estimer ces performances nous définissons le taux d'erreur canal (*Channel Error Rate*

(CER)) : il s'agit pour chacun des échantillons traités de vérifier si le canal estimé $\hat{p}$ est identique au canal effectif p. Il s'en déduit alors une probabilité qui doit tendre vers zéro en absence d'erreurs. Les performances sont évaluées par simulation sur une bande de 80 MHz et en fonction de nombreux paramètres comme l'erreur de synchronisation, le type de modulations, et la vitesse de saut. Ces simulations ont permis de montrer que les performances de notre méthode DFT-BE sont légèrement supérieures à celle de l'algorithme du PPN (tout en étant moins complexe) et bien supérieures à celles de l'algorithme de transformée en ondelettes.

3.3 Performances de l'extraction de signaux audio

L'objectif final de notre système est la vérification de la présence de données sensibles au sein d'une transmission en saut de fréquence. Pour cela, nous utilisons comme donnée sensible un signal audio dont les fréquences sont aisément perceptibles et discernables. L'évaluation des performances de la reconstruction du signal rouge est réalisée grâce à l'algorithme SBOS (Selected Bands Opinion Score) qui est une adaptation à notre usage de ViSQOL [Chi20] (Virtual Speech Quality Objective Listener). Ce dernier réalise une comparaison entre deux signaux audio en compensant les éventuels phénomènes de décalages temporel et fréquentiel. Des simulations ont été effectuées afin de déterminer les performances des algorithmes sous les mêmes conditions que pour les estimations de CER. Il en est ressorti que la DFT-BE dispose de résultats similaires au PPN alors que l'algorithme FFT-BE est meilleur en haut rapport signal à bruit mais bien inférieur en présence de bruit. Ces résultats sont doublement intéressants, car ils montrent l'intérêt d'une solution de type FFT-BE dans la gamme d'utilisation que l'on va traiter (rapport signal sur bruit important) et des résultats en SBOS qui ne sont pas directement induits (ni directement reliés) à celui du CER. Ainsi, il est complémentaire de caractériser les performances des algorithmes au niveau de la récupération du signal rouge et noir (CER) et au niveau du signal rouge uniquement (pour lequel la métrique choisie doit directement liée au modèle de fuite considéré et à la nature du signal fuité).

4 La radio logicielle dans le contexte des canaux auxiliaires

Le concept de radio logicielle (*Software Defined Radio* (SDR)) prend racine dans le domaine militaire afin d'assurer une utilisation sur le long terme des équipements tout en gardant la possibilité d'intégrer de nouvelles formes d'ondes et normes de communication grâce à une

simple reprogrammation. Les SDR ont été introduites dans les années 90 [Mit92] et consistent à déplacer au maximum les parties analogiques dans le domaine numérique afin d'obtenir une plus grande flexibilité inhérente aux applications numériques.

4.1 Radio logicielle

Une radio logicielle idéale équivaudrait à placer directement des convertisseurs analogiques-numériques après les antennes et à réaliser tous les traitements au sein d'un processeur. Bien qu'extrêmement flexible, cette structure requière une importante fréquence d'échantillonnage des convertisseurs et l'utilisation de processeurs généralistes seuls n'est pas réaliste du fait de limites dans les débits atteignables et d'une très mauvaise efficacité énergétique. C'est pour cette raison que des accélérateurs matériels sont utilisés afin d'alléger la charge des processeurs. Dans cette thèse, une architecture mixte de SDR utilisant un FPGA et un processeur est utilisée afin d'atteindre des débits importants en temps réel.

L'utilisation d'une architecture mixte soulève des contraintes de langages de programmation. En effet, de multiples langages sont nécessaires afin de développer, tester, et intégrer les différentes parties du système d'interception. Une approche utilisant le langage Julia [Bez17a] est proposée afin de permettre de conjuguer une approche de prototypage (syntaxe de haut-niveau) et un débit de traitement important. Le projet *AbstractSDRs.jl*, accessible en *open-source* est introduit afin de disposer d'une interface de programmation commune à plusieurs architectures de radio logicielle différentes [Lav21a].

4.2 Architecture mixte d'interception et de récupération de données sensibles

La Fig. 0-3 présente une vue d'ensemble du flot de traitement de l'algorithme FFT-BE. Les parties soumises à débit important (*FH detect*) sont placées dans le FPGA afin d'obtenir un débit de traitement adéquat. La partie gérant la récupération de données rouge et son analyse est placée dans un processeur généraliste afin de profiter de sa plus grande flexibilité. En plus de l'algorithme FFT-BE décrit précédemment, le design réalisé ajoute un mécanisme permettant d'estimer l'utilisation du spectre écouté par la SDR et un système permettant de corriger des erreurs de mauvaise détection.

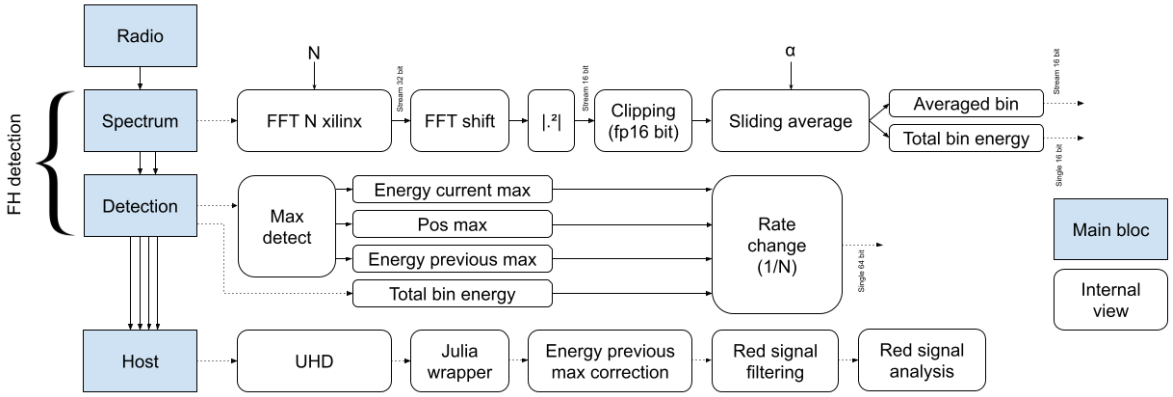


FIGURE 0-3 – Chaîne de traitement FFT-BE et partitionnement logiciel/matériel.

5 Vers une application matérielle

La dernière étape de ces travaux se déroule en deux temps : il convient dans un premier temps de valider le système de détection, en s'assurant que les canaux d'intérêt sont détectés et extraits en temps-réel sur la bande la plus large possible pour notre radio. Puis, dans un second temps, le système est utilisé afin de vérifier la présence de compromission au sein de systèmes sur puce (*System on Chip* (SoC)) Bluetooth.

5.1 Validation de l'architecture matérielle

Tout d'abord des tests unitaires sont réalisés grâce à du matériel dédié. Une génération d'un signal rampe de fréquence comportant une modulation AM permet de valider qu'il est possible de détecter un signal changeant de fréquence, d'extraire une donnée rouge (la modulation AM), mais également qu'un débit de 200 MHz est tenu par le système en temps réel sans perte d'échantillons. Le test suivant consiste à utiliser le langage Julia et de soumettre le système d'interception aux mêmes échantillons que lors de la phase de simulation afin de valider que les performances sont identiques.

5.2 Recherche de compromissions au sein de SoC Bluetooth

Le système d'interception est ensuite utilisé afin de détecter des compromissions dans des systèmes à saut de fréquence du commerce. Pour commencer, des fuites sont forcées au sein de dispositifs Bluetooth grâce à une bobine et un mélangeur analogique, ce qui permet de valider que des canaux auxiliaires peuvent se former au sein de SoC. Cette expérimentation

permet également de démontrer que ces canaux auxiliaires peuvent impacter un transmetteur radio au point qu'il est possible de récupérer les données sensibles. Le système a ensuite été utilisé comme il le serait lors d'audits de sécurité sur des dispositifs non-modifiés. Nous avons ainsi démontré que certaines activités du SoC, notamment au niveau des interfaces PWM, se retrouvent dans un canal auxiliaire porté par la transmission Bluetooth ! Ces vulnérabilités ont été mises en évidence et exploitées pour plusieurs types de puces Bluetooth du commerce et ce dans le cadre d'un fonctionnement non altéré (Bluetooth non modifié et fonctionnement du système nominal).

La Fig. 0-4 montre un exemple de canal auxiliaire récupéré au travers d'une communication Bluetooth. Dans cet exemple, des LED clignotent à 1 kHz et cette fréquence caractéristique (et certaines de ses harmoniques) est visible dans le spectre radio intercepté.

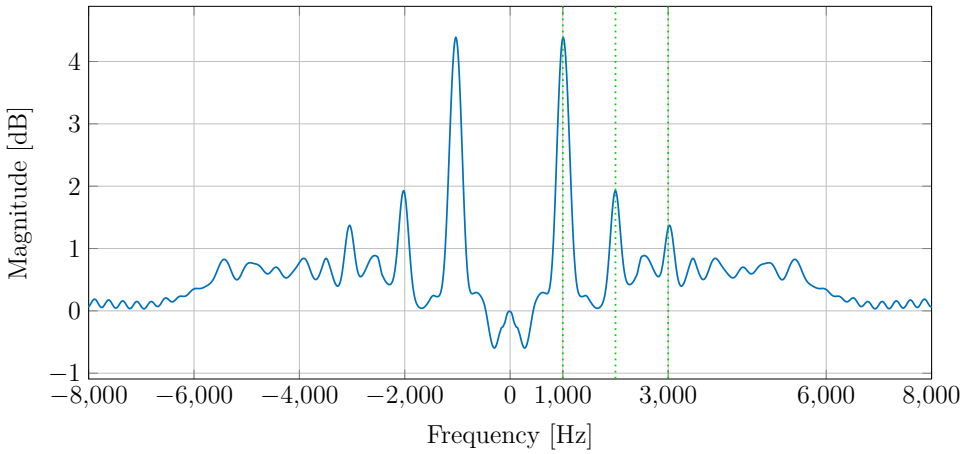


Figure 0-4 – Recherche de canaux auxiliaires dus à des LED.

6 Conclusion

Nous avons donc abordé dans cette thèse la problématique des fuites d'information par émanation. Une étude des canaux auxiliaires a d'abord été effectuée afin d'en connaître les caractéristiques principales, et de trouver des points communs ainsi que des liens entre les compromissions de natures différentes. Il est apparu que les canaux auxiliaires electromagnétiques portés par un signal radio sporadique (signaux à saut de fréquence) présentent un indice de menace important et sont difficiles à détecter. Il s'agit donc de pouvoir intercepter ces signaux et d'extraire des potentielles compromissions. Dans le cadre de ces travaux, un système d'interception de signaux à saut de fréquence, temps réel et large bande a été proposé.

Ce système a d'abord été caractérisé en simulation et comparé à des algorithmes de l'état de l'art. L'intégration du système dans une radio logicielle embarquant un FPGA a nécessité une étude du fonctionnement des SDR et des divers accélérateurs matériels utilisables. Une étude des moyens de programmation a également été réalisée pointant les limites dans le paradigme de programmation des radios logicielles. Un nouvel écosystème logiciel, *AbstractSDRs* basé sur le langage Julia a été proposé. Il permet d'allier un prototypage rapide et un débit de traitement important. Ainsi, une architecture complète couplant ressources matérielles (FPGA) et logicielles (Julia) a été proposée pour l'interception et l'extraction de canaux cachés portés par des signaux à saut de fréquence. Pour terminer, le bon fonctionnement du système a été validé grâce à du matériel spécialisé. S'en est suivi une phase expérimentale où le système a été utilisé comme il le serait dans des audits de sécurité afin de détecter des canaux auxiliaires : dans un premier temps sur des dispositifs où une fuite a été forcée puis sur des dispositifs fuitant d'eux-mêmes. Ces expérimentations ont prouvé que le système d'interception proposé est capable de les détecter, et également que des fuites peuvent se produire à de multiples endroits dans des SoC Bluetooth du commerce. Nous avons ainsi montré la nécessité de rehausser le niveau de menace induit par les canaux cachés télécoms, c'est-à-dire porté par un signal légitime de transmission.

Chapter **ONE**

INTRODUCTION

General context

In recent decades, organizations, firms, states and administrations have sought to become more efficient and productive by adopting computer technologies, thus engaging in the process of *digital transformation*. This new era of digital services and globalization has made the movement of goods, services, finance and information easier, and large companies can manage their international operations in a leaner, more efficient way.

Information security is based on three key principles: confidentiality, integrity and availability:

- **Confidentiality** aims to prevent unauthorized access to sensitive information. It can be achieved through encryption (renders the information unreadable to anyone without the decryption key) and access control (restricting exchanges with authorized entities)
- **Integrity** protects the information from unauthorized modifications by an entity and ensures that it remains accurate (nothing has been lost or degraded). This can be achieved through a hash (signature made from the data) or through a reference (mirror of the data considered as unaltered).
- **Availability** allows authorized users of a system to have fast and uninterrupted access to the information contained in that system. This is achieved through the use of redundant systems (several copies of a system can be substituted), distribution of tasks in multiple entities and the management of degraded modes (the system is able to operate in spite of system failures).

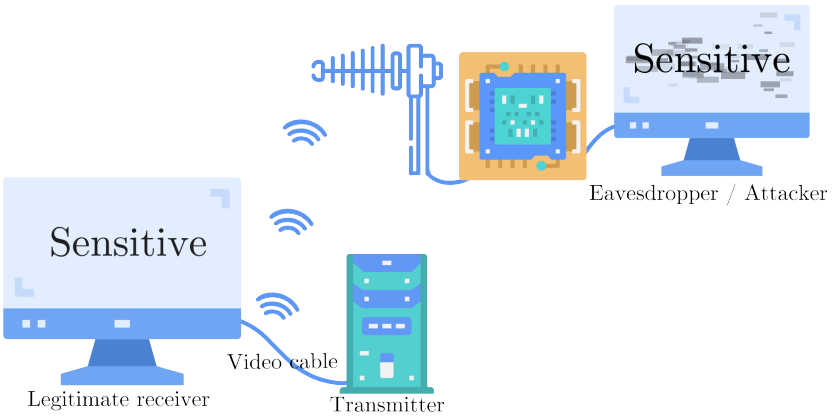


Figure 1-1 – Side-channel attack example.

The growing use of information technology increases the proportion of sensitive information transiting in digital media, consequently increasing the risk of information leakage. It makes information security a more and more complex discipline that requires increasingly wide-range of skills. One can think that computer skills are sufficient to understand and circumvent all possible leaks but it is not enough. There are indeed leaks called side-channels which are based on the effect of one or more physical phenomena causing information to deviate from its original path.

Side-channel context

The side-channel requires an understanding of the electro-magnetic, sound, light phenomena and also device hardware architecture. This type of leakage can appear at any place of a system and an example is shown in Fig. 1-1. Sensitive information passes through its legitimate channel to its destination (the computer to the screen in the Fig. 1-1), this channel can be of several kinds, a processing, storage or even transport system. However, a so-called side-channel has been formed and the sensitive information has also taken a new path through it (the video cable generates an Electro-Magnetic (EM) field as current pass through it and an eavesdropper can reconstruct the image by intercepting the EM signal). A side-channel can be the result of a system design flaw, be created by an external phenomenon, be the result of a tampering or a mixture of all three. A side-channel does not usually pose a threat to availability and integrity of information but is a great threat to confidentiality.

Indeed, if someone manages to intercept the side-channel, he is able to recover the sensitive

information it encloses. The consequence is to bypass some security set up in the legitimate channel. Therefore, it can be easier for an attacker to intercept information contained in a side-channel than in a legitimate channel [Gur18b].

The study of the presence of side-channels in a device is rarely studied except in the military and security fields, which results in numerous devices being vulnerable to side-channel attacks and can therefore have important consequences both at individual, industrial and state levels. A sensitive information can leak on several different side-channels, therefore to make a complete study it is necessary to be able to access several propagation methods. For example, a leak in a power supply can be propagated in the electrical network, in the devices that it supplies, but also by electromagnetic waves and sounds due to the vibrations produced in the coils and capacitors.

Among the different side-channels, one in particular stands out for being more complex to detect and prevent, it is the electro-magnetic side-channel, which can be generated at any location where there is a current. The study of these side-channels has several names, TEMPEST or in general ElectroMagnetic SECurity (EMSEC) and consists mainly in ensuring that no leakage is escaping from equipment. Since EM can be issued from a large number of components, it results in a large variety of sensitive information which could be leaking via EM [Bre16].

How can a radio signal be concealed?

An EM signal that emits continuously on the same frequency can be detected by scanning the radio spectrum around a targeted device. However, not all emitted signals contain hidden information. Clock signals are usually easy to find due to their waveform which generates a lot of radiations in the cables but contains little information. On the contrary, there are side-channels that are more difficult to find, e.g., when they affect a radio transceiver and especially with frequency hopping mechanism.

Frequency hopping is a technique to limit radio interference and make several signals coexist on the same radio band by emitting at distinct frequencies, different at each transmission. A radio band is divided into channels (as shown in Fig. 1-2), a message is transmitted partly on one channel and then continues to be broadcast on another and switches continuously. The channel switching complicates the message detection because it is necessary to listen to the right part of the spectrum in order to detect it. In addition, the message may not be transmitted

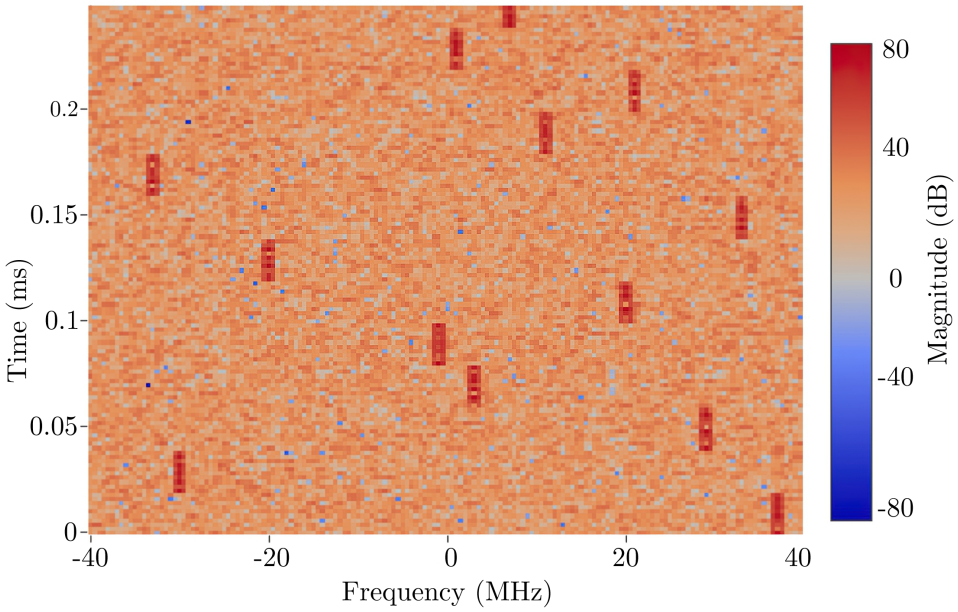


Figure 1-2 – Examples of frequency Hopping issued from a Bluetooth device in the 2.4 GHz ISM band, red areas correspond to data exchange.

continuously, and therefore, in addition to finding the right frequency, it is necessary to find the right moment. The knowledge of the frequency hopping pattern is an essential element to receive an FH signal and to be able to check the presence of additional data contained in the baseband signal (a sensitive information that propagates on the transceiver through a side-channel in our case). However, this knowledge is not always available when a device is to be checked, especially in the case of secured communication systems (TRANSEC [Bej12]). Moreover, knowing the hopping sequence only allows one to observe the channels used by the device in its normal operation, signals may be sent on channels outside the normal hopping pattern due to a malfunction or tampering of the device (covered-channel [Gur18b]).

In order to be able to detect these particular cases, it is necessary to dispose of a system that is able to observe a large band of the spectrum at the same time, then to detect if a signal is present in this spectrum and to extract the baseband signal in order to investigate if a sensitive data is hidden in it. All these requirements can be fulfilled with the use of a Software Defined Radio (SDR).

How frequency hopping signals can be intercepted?

SDR [Mit92] depicts a flexible Digital Signal Processing architecture with very high reconfiguration capabilities allowing the same architecture to be used in a multitude of applications. Such an architecture shifts most of the processing to programmable digital components such as microprocessors. An SDR converts a signal as close as possible of the antenna by a Digital to Analog Converter (DAC) and an Analog to Digital Converter (ADC), respectively of transmitting and receiving signals.

First designed to maintain the interoperability of military radio with the addition of new protocols without having to deploy new hardware, the SDR has rapidly taken an important place in the telecommunication field, allowing to get rid of the long and heavy cost of development of specialized radio.

With the emergence of SDR, cognitive radios [Mit02] have become realizable. These radios are able to adapt to their environment. They can change the type of modulation according to the traffic and noise level, avoid interference... Among their capabilities are methods for probing the spectrum, which initially are designed to prevent collisions with other radios. However, these methods can be applied to the frequency hopping detection because both seek to determine if a transmission is present in a part of the spectrum.

An important aspect for the interception of an FH signal, in addition to the algorithm and the used hardware is how it is actually realized. An FH signal imposes constraints on the reception architecture:

- **Wideband** because we cannot know in advance at which frequency the signal will appear, it is necessary to take a band wide enough to be sure to have all signal possible frequencies.
- **Time accurate** in order to detect fast frequency hopping in the case of secured radio (in order of μs in our case).
- **Real-time** because the large bandwidth results in a large data flow that most storage media are unable to sustain, therefore the processing must be done without having to store the raw data.

The interception algorithm, the language used to implement it and the SDR must therefore be in line with these criteria. An additional constraint specific to our thesis and not due to the FH signal is the mobility of the material. This research is realized in collaboration with

the DGA-MI which would use the interception architecture in security audits to validate that a device does not contain sensitive information carried by an FH signal. It is necessary that the system is easily transportable typically in a bag, and does not require internet connection. The use of cloud computing for intensive calculations is not possible and consequently all the processing must be realized locally.

How to detect if the FH device is leaking?

After having successfully intercepted an FH signal, the next step is to analyze the recovered baseband signal to determine whether it has been compromised. This analysis is specific to the nature of the data being verified. In this thesis we will focus on the audio signals. This type of side-channel is a real case and is for example found in airplanes where the limited space on board leads to mix-ups between wires and sensitive audio signal that can appear on the outgoing radio signal.

An audio signal is characterized by a low bandwidth, and as an extensive state of the art on how to evaluate its presence and degradation is available, this makes it an excellent choice for the study of side-channel presence.

As we will see during this thesis, a device alone does not automatically lead to compromise, but in conjunction with another device, a side-channels can occur. Therefore, in addition to analyzing the presence of side-channels in an isolated device, it is necessary to check whether its interaction with a third-party device will not lead to a compromise. A typical case is the use of a loudspeaker in a radio system, where the audio signal will be broadcast on the transmitted signals due to the side-channel.

Contributions and outline

Contributions of the Thesis

This thesis proposes a methodology and tools to address the task of checking the presence of a side-channel in a frequency hopping radio-based device. For this purpose, we first start by studying the scope of side-channel leaks to explore the many existing side-channels and sort them according to their characteristics, their propagation method, their origin, their range. This study of the state of the art on side-channel emanation allows to better understand the

background context and also to link several propagation methods together with the aim of creating a classification of these different side-channels. Among these, the side-channels propagating within a radio transceiver are rather interesting because their range is much higher than a conventional side-channel. Moreover, their detection is a much more complex task when the signal emitted by the radio is based on a frequency hopping mechanism.

Usable methods to detect an FH signal, their complexity, the data used, and their operation are discussed with the aim of choosing the best candidates. These candidates will then be evaluated by simulation according to several radio signals and errors while considering their capacity to detect effectively an FH signal and their capacity to extract a red signal issued from a side-channel.

A methodology to take advantage of SDR capabilities and FPGA-based processing acceleration will be given. The goal is to realize a system that intercepts FH signals and retrieves red data in real-time over a wide band (200 MHz).

The interception system will then be used on several hardware targets featuring side-channels to first validate its proper functioning in real-time, then to characterize which leaks the system will be able to detect by forcing leaks on devices and evaluate which external interaction could compromise a device. Finally, to conclude a listening of unmodified targets featuring side-channels will be performed.

Thesis outline

Chapter 2 - Side-channels state of the art. The chapter aims to review the wide range of existing emanation side-channels. For each of them, a description of their main characteristics will be given, with particular attention to their dangerousness, their ability to carry sensitive information as well as the methods and equipment used to intercept them. The first contribution of this thesis is a classification of the various side-channels considering their characteristics.

Chapter 3 - Interception methods of frequency hopping signals. The chapter aims to introduce frequency hopping transmission as well as the means to intercept them. A particular attention will be given to the metadata to be provided in advance to the algorithm for its operation, in order to draw up a list of usable methods in accordance with the constraints (in real time and blind).

Chapter 4 - Evaluation of the proposed approaches in the TEMPEST context. The chapter will show the results of simulations on selected interception algorithms, considering both commercial and TRANSEC FH cases. The ability to intercept an FH signal as well as the ability to recover a red signal are discussed through several simulated signals.

Chapter 5 - On the use of software defined radio for TEMPEST. The chapter highlights the potential of SDR use in TEMPEST scenario. Thanks to their great flexibility, rapid programming and available hardware acceleration which will be presented throughout the chapter, a real-time FH signal interception and audio red signal recovery system based on the methods defined in the previous chapter is given.

Chapter 6 - Toward real eavesdropping. The chapter will present the final use of the previously designed system, from the verification on synthesized signals, through the side-channel issued by tampered devices, to the real exploit with an untampered device that leaks sensitive data on its radio transceiver.

List of publication

National conferences

— Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Interception of Frequency-Hopping Signals for TEMPEST Attacks », *in Proc. Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, Virtuelle, France, Dec. 2020

This conference article published in *Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information* introduces the DFT-BE interception system as detailed in the Chapter 3.

International conferences

— Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Towards Real Time Interception of Frequency Hopping Signals », *in Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, Sept. 2020, pp. 1–6

This conference article published in *IEEE Workshop on Signal Processing Systems* deals with the time synchronization mechanism required by the DFT-BE as detailed in the Chapter 3.

Journal papers

— Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Whispering Devices: A Survey on How Side-channels Lead to Compromised Information », *in Journal of Hardware and Systems Security (HASS)*, Mar. 2021

This journal article published in the *Journal of Hardware and Systems Security* summarizes the different emanation side-channels with an emphasis on countermeasures, presented in Chapter 2.

— Lavaud, C., Gerzaguet, R., Gautier, M., and Berder, O., « AbstractSDRs: Bring down the two-language barrier with Julia Language for efficient SDR prototyping », *in IEEE Embedded Systems Letters* (2021)

This journal article published in *IEEE Embedded Systems Letters* deals with the efficient use of SDR and in particular with the Julia programming language, this subject will be detailed in Chapter 5.

Thesis founding

This thesis was founded by pôle d'excellence cyber (PEC) in collaboration with the French ministry of defense (DGA).

Chapter **TWO**

SIDE-CHANNELS STATE OF THE ART

Contents

The large-scale deployment of digital technologies has paved the way to new types of threats. Potentially confidential data may be carried within the same information system as ordinary content, blurring the boundary between the sensitive and non-sensitive area and leading to potentially malicious security breaches. This flaw has driven a dramatic increase in high-risk incidents and failures in information security, resulting in significant revenue losses, ecological disasters and trust issues. Billions of wireless objects are disseminated to monitor urban and rural areas, as well as animals and human beings. While most of the data produced and transmitted does not contain any sensitive information, the criticality of some applications

is beyond doubt. Moreover, the multiplication of devices offers new transmission channels and by extension more possible paths for sensitive information to reach an attacker, which is particularly the case for leaks resulting from the mixing of several signals (see 2-3.2). The search for the source of a leak is consequently rendered even more complex.

In the first section of this chapter, basic notations and classification of side-channels are depicted. A brief overview of the side-channel categories is carried out and a focus on emanation side-channels is made. Then, in Section 2-2 we address the non-electromagnetic side-channel by raising the particularities and weaknesses of each of them and addressing a sample of targets where an attack is performed, through a review of various publications of the state of the art. In Section 2-3 we specifically focus on our main subject of interest which is the electromagnetic side-channel and do an analog work of reviewing state-of-the-art paper.

1 Definition and taxonomy

Confronted with such a variety of attacks, companies tend to focus on protecting data based on their confidentiality (limited access to information), integrity (the equipment used is reliable and accurate) and finally availability (information only accessed by authorized people). To that end, several defense mechanisms have been proposed such as: limited use of control access and systems, security watchdogs at every network access point, prevention methods to enhance security and confidentiality through training (awareness campaigns and good practice workshops [Sul15] [Pfl08]) and widespread use of cryptography. Although most of these securities relate to software, if the physical layer is not secured, all the upper layers are exposed to major security flaws. Side-channel attacks rely on these hardware vulnerabilities to retrieve confidential data.

1.1 Definition of a side-channel

A side-channel denotes the presence of information in an illegitimate channel, whether at hardware or software levels. Illegitimate means that the information should not be in the channel given the normal operation of the involved device. A channel can be of a diverse nature: electromagnetic wave, light are channels that can be easily pictured but they can be more exotic like heating pipes, response times or power consumption.

It should be noted that a side-channel designates the channel itself (i.e., the propagation

medium) and that a side-channel attack designates a complex process, involving stages of acquisition, processing, with the aim of intercepting a side-channel and to extract sensitive information.

Side-channel attack seek the deviations in the implementation of a task in a device (how the task is done) rather than the implemented algorithm weaknesses themselves (what the task does). Several types of side-channels exist and they can be classified into two main categories depending on the nature of the leakage.

— A *software side-channel* is contained within a device is based on hardware vulnerabilities (e.g. RowHammer [Aic15]) or firmware vulnerabilities (e.g. Meltdown [Ge16], Spectre [Koc18]) and its operation requires mostly software skills [Sze18].

— On the other hand, as an *emanation side-channel* crosses the device boundary, its exploitation needs specific acquisition equipment according to the nature of the side-channel and also requires deep hardware knowledge as well as knowledge about the side-channel nature. It is the result of one or more physical phenomena that deviate information from its proper path to reach an unintended one.

The present chapter focuses on the latter side-channels as with the rest of the manuscript.

1.2 Taxonomy of interceptable data

Essentially, all data may be considered sensitive or confidential and therefore should not be recovered by an opponent. Some information can nevertheless be defined as less critical than others. Low criticality may be ascribed either due to the nature of the information, or because the information is supposed to be safe (e.g. using encryption). In NACSIM report [NSA82], the NSA (National Security Agency) provided the first definition of TEMPEST (see Section 2-3) and introduced the concept of red and black signals.

A **red signal** is an unencrypted signal that must be treated as highly sensitive material. In the NSA specification, this type of signal must be contained within a specific perimeter to ensure security. To that end, security measures [NSA] have to be taken into account such as shielding the location or ensuring a minimum physical distance between wires (to avoid coupling).

A **black signal** is considered to be a safe signal, i.e., it does not directly carry a compromising signal, using encryption or other methods to render interception impossible. Unlike red signals, black signals are not defined by their secure perimeter.

It should be noted that breaking cryptography does not form part of the work carried out in the present manuscript. Hence cryptography systems will be addressed from a side-channel attack perspective, i.e., recovery of a secret key used during ciphering due to side-channel leakage. Generally speaking, all the compromising signals studied in this thesis will fall into the red family i.e., unencrypted signals. While modern systems heavily rely on encryption to secure their data, it is generally during the encryption/decryption process that attacks are carried out, and so the study of side-channel is still a key element in security nowadays.

1.3 Class of side-channel attacks

Side-channel attacks can be classified based on the activeness of the attacker (active or passive) and whether the target intentionally or accidentally generates the leak (intentional or non-intentional).

- *Active vs. passive*: An active attack emits a carrier signal from a common active system towards/within the target in order to either tamper with the device proper functioning (e.g. fault-induction attacks aim to induce errors in computation to exploit unconventional target behavior) or cause the target to leak a signal (e.g. emitting a radio wave passing through a target to modulate a red signal). In contrast, passive eavesdropping is simply used to observe the device behavior without disturbing it.
- *Intentional vs. non-intentional*: Non-intentional leakages can be naturally present and therefore carry a red signal resulting from normal functioning. However, leakages can be intentionally emitted by a target on a side-channel, which implies that the attacker can access the target to force the leakage.

Examples of attacks can be found in Table 2-1 which provides a classification based on the previously mentioned classes.

1.4 Taxonomy of side-channel

Emanation side-channels include fortuitous emanations caused by the normal functioning of a device as well as the alteration and amplification of an internal signal to produce an emanation.

	Intentional	**Non-intentional**
Active	Ultrasound mesh Illumination	Mixing Broadcast
Passive	Air-gap bridging	Video eavesdropping TEMPEST attacks

Table 2-1 – Examples of attack per activeness and intentionality.

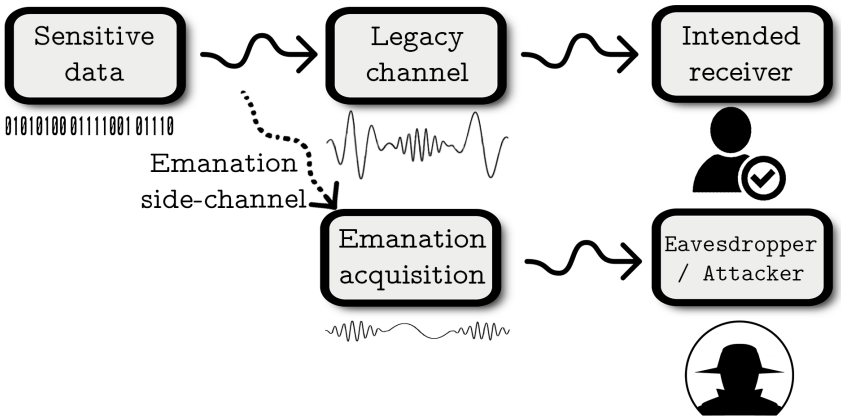


Figure 2-1 – Overview of emanation side-channel attack.

A common feature of these side-channels is their non-intrusive nature, i.e., a direct contact with the target is not needed to get the side-channel, in opposition to software side-channel which requires to be inside the target (mostly in the form of a software program). The typical case of a side-channel attack is shown in Fig. 2-1. In the normal flow, sensitive data is processed and this process leads them to be present in a legitimate channel. However, either the processing system or the legacy channel leaks to another medium i.e., the side-channel. A person gathering this leaked data can recover sensitive information. This person may either be an active attacker if he launches the attack (emits a signal toward the system to generate the side-channel emanation) or an eavesdropper if he only passively intercepts the information.

Since the expected path of the information to the intended receiver has been studied by its designer to be secure, it may be difficult to extract the sensitive information from it, but since the side-channel is not intended, it may be easier for an attacker to extract the information from a side-channel. Despite awareness of the potential risks posed by such emanations since the early days of ElectroMagnetic Compatibility (EMC) in the mid-20^{th} century, proof of their actual use has only recently been provided with disclosure of certain tools from the NSA [NSA08].

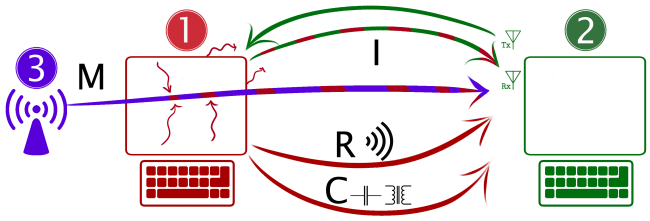
The development of new attacks is linked to the development of the targets that occurred in several phases. Analog signals were firstly used to communicate, before the transition to the digital world. This changed the paradigm of the emissions, shifting from low frequency and high amplitude to high frequency and low amplitude (due to the increase in clock frequencies and the overall reduction of operating voltages). Then mobile devices appeared and with them so-called System on Chips (SoC) allowing a size reduction, which obviously brings an additional risk of internal mixing (and thus leaking). Moreover, the worldwide competition led to overall reduction of equipment costs, sacrificing some filtering tasks and shielding that were preventing leaks proliferation. But attacks are also benefiting from the improvement of the hardware devices involved in the interception. These new devices reach higher bandwidths, are more robust against noise and the increased computation capabilities allow efficient but energy demanding techniques such as emerging machine learning. Because many attacks rely on physical or remote access to a vulnerable target, one straightforward way to counterbalance such attacks is to work within isolated systems (also referred to as air-gapped systems). Isolated systems (or networks) do not allow external access and thus maintain the sensitive data in a closed area. As such side-channel attacks have evolved to address these systems by using a leakage as a fully controllable communication medium instead of simply extracts information from it. These attacks are the so-called air-gap bridging [Gur18a].

Fig. 2-2 summarizes the different physical propagation side-channels in which a compromising emanation may occur. These side-channels may either be non-electromagnetic or electromagnetic (using radio-frequency waves).

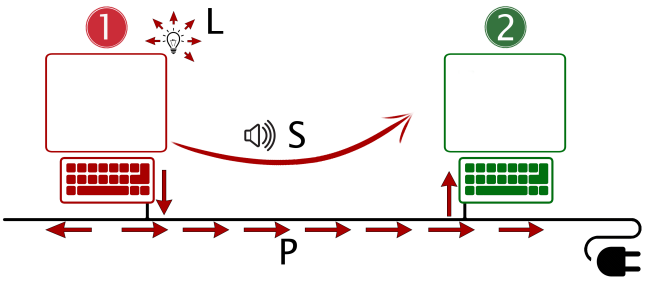
Electromagnetic side-channel is directly derived from an electronic component. This type of side-channel offers useful propagation characteristics and is currently widely used as a communication medium. Electromagnetic leakages share a common side-channel but are distinguishable by their different origins. As shown in Fig. 2-2a, we can classified these origins as illumination (I), mixing (M), radiation (R) and coupling (C).

(I) Illumination, the attacker sends a radio beam towards the target. This radio beam enforces a leak of information by acting as the side-channel carrier, therefore the attacker needs to monitor the beam that comes back and contains the information. In the mixing case, the radio beam comes from a legitimate source and not from an attacker.

(R) Radiation, probably the most common case of EM side-channel, is due to the generation of a current-induced electromagnetic field. This EM field comes from the different electronic



(a) Electromagnetic side-channel origins: illumination (I), leak by mixing (M), leak by radiation (R), leak by coupling (C).



(b) Non-electromagnetic side-channels: sound leak (S), leak by the power line (P), light leak (L).

Figure 2-2 – Typical side-channels for an emanation scenario composed of a sensitive target ❶, an attacker ❷ and a neutral access point ❸.

parts, and is directly correlated with the information passing through the device.

(C) Coupling is similar to radiation but propagates in a nearby conductor (metallic housing, power line, water pipe, etc.) and not in the air.

Fig. 2-2b highlights three main non-electromagnetic side-channels: power line (P), sound (S) and light (L) (light is not referred to as an electromagnetic side-channel in the literature).

(P) Power line includes electrical consumption of all the parts of a system. In a processor, the instantaneous energy consumption depends on the data being processed and the operation performed. Clearly, an element with high-power consumption will generate a higher trace on the overall consumption. It should be noted that the power line side-channel relates only to electrical consumption and no other leaks such as cross-talk phenomena on power lines (which falls into the electromagnetic class detailed in Section 3).

(S) Information processing devices may also emit other types of emanation such as sound, which may be generated by mechanical components (motors, buttons, etc.) or directly by

the electronics (capacitor, coil whine, etc.). In precisely the same way as radiation (R), the generated sounds are strongly correlated with the voltage that generates them and therefore with information.

(L) The final reviewed leakage uses light as its side-channel. This leakage has a significant advantage that is also its major drawback: human eyes are sensitive to some of the light spectrum, which increases the risk of detection. On the other hand, many systems include light sources as an interface (screen, state diode, etc.) that can be maliciously used to extract information. As a side note, we have chosen to consider light as a specific side-channel and not a specific case of electromagnetic side-channel as it is customarily done in the literature (see for instance [Gur18b]).

2 Non-electromagnetic side-channels

An overview of the proposed classification and the associated paper is shown in Fig 2-3. A distinction between the targets of each attack is made in the following section and is also visible in the figure. Each side-channel will be presented in greater detail, considering both what information can be reconstructed for each type of emanation and the resources needed for their application.

2.1 Power line

Du et al. [Du13], have demonstrated that it is possible to recover the keyboard data bus by studying its fingerprint on a power line. The measured consumption is a combination of keyboard consumption and consumption of other components.

Riccardo et al. [Spo17] introduced a way to exfiltrate data via a Universal Serial Bus (USB) cable without the use of the embedded data link. Instead, the power supply provided by the USB cable is used. By creating bursts of current, an On-Off Keying (OOK) modulation was achieved, allowing a 2 bit/s link using a smartphone and a tampered phone charger.

Using the same approach, other operations performed on the processor may be retrieved. Hence, retrieving all the instructions would constitute a very powerful attack. However, due to the rate of instruction execution, this type of attack requires immeasurable resources. Some specific instructions may, however, be targeted such as cryptographic operations [Agr03]. By eavesdropping the power consumption of a cryptographic-device, power traces can be

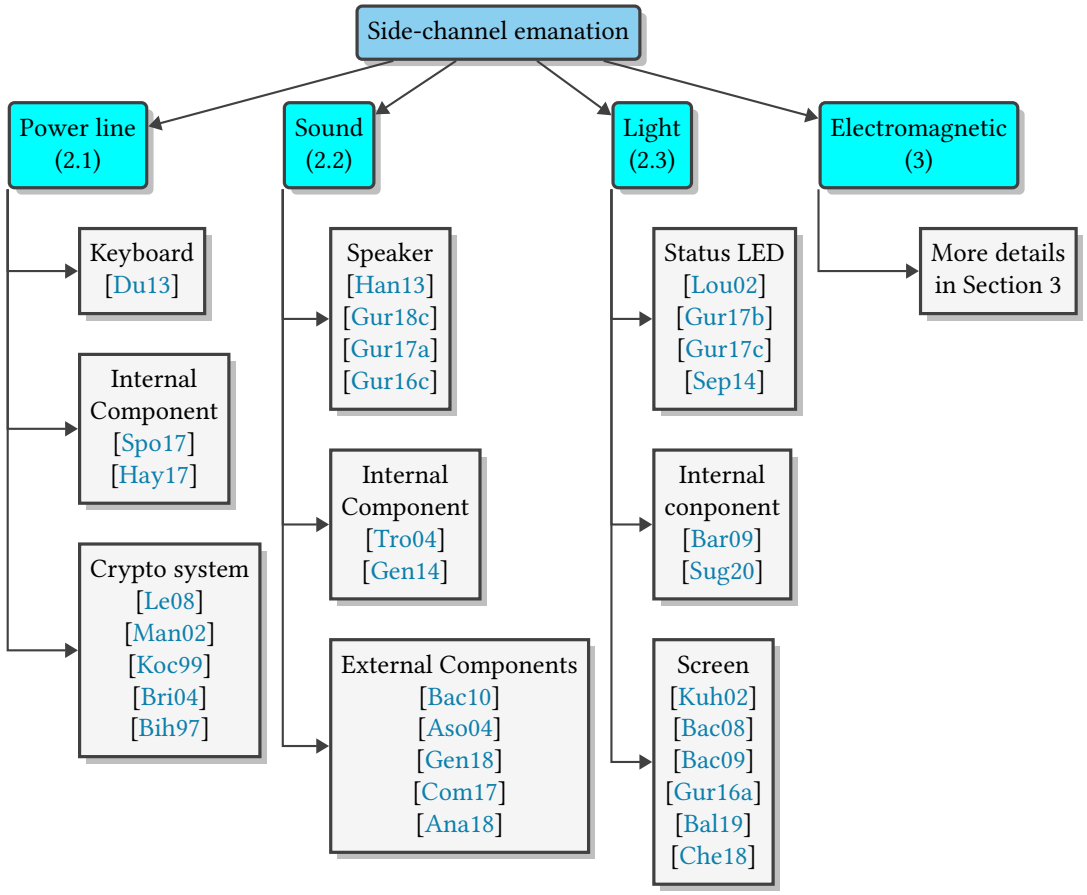


Figure 2-3 – Classification of the emanation side-channels according to propagation nature.

analyzed using an algorithm such as Simple Power Analysis (SPA) [Man02], Differential Power Analysis (DPA) [Koc99], Correlation Power Analysis (CPA) [Bri04] or Differential Fault Analysis (DFA) [Bih97] to derive the system secret key. The remainder of the section will focus on the specificities of these methods.

The total power consumption of a CMOS circuit is composed of two terms: the static power and dynamic power.

— Static power is due to the transistor internal leakage current and is therefore dependent on circuit design.

— Dynamic power is due to transistor activity (i.e., transistor switching) which depends on the actual operation being performed and the data being processed. Since analysis is aimed at determining a link between power consumption and the data being processed, only dynamic power is relevant.

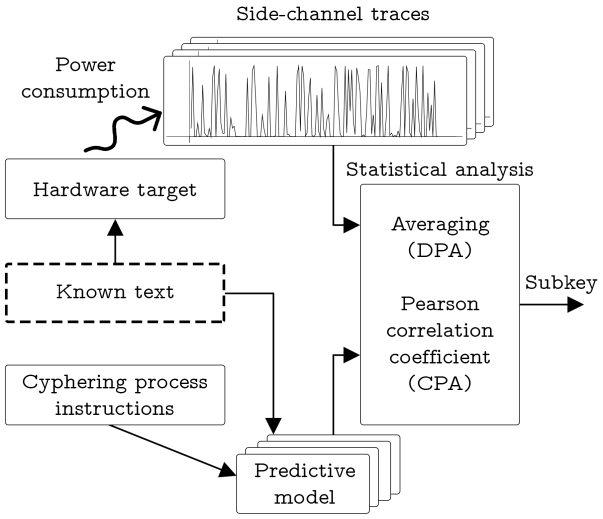


Figure 2-4 – Outline of the DPA and CPA analysis.

Separating the static and dynamic power is not necessary because the static is mostly an offset on the total power consumption of a device, it is straightforward to use total power as a possible side-channel.

SPA is based on the direct interpretation of power consumption measurements. The method used is quite basic, since each processor operation has a different trace: for example, multiplication and square operations involve the same processing. The main difference is the operand loading stage that shall be twice longer for a multiplication (indeed, a multiplication needs two inputs while a square operation only requires one). As a consequence, the multiplication and square patterns, even if they are quite similar, can be distinguished by their width which is larger for the multiplication one. In the case of an unprotected implementation of cryptographic algorithms, the sequence of operation that is performed is linked to the secret key. Therefore, if one is capable of discriminating operations then he can find out this sequence and finally recovers the secret key [Man02].

For more powerful attacks, once the observational data have been collected, they must be processed in order to retrieve information. Various methods exist: firstly, DPA combines consumption analysis (SPA) with statistical methods (see Fig. 2-4). It breaks down the secret key into blocks to compute each block individually to reduce the number of unique combinations of bits. Indeed, an Advanced Encryption Standard (AES-128) standard key has 2^{128} combinations. This key is divided into 16 bytes which only contain 2^8 combinations with a total of 16×256 or 4,096 secret key combinations. DPA requires the system power trace and its associated text

(either input or output text). It works on making hypotheses about internal variable values (each block at a time) and separates them into baskets depending on the specific bit value of the known text. An average is reached for each basket and if the difference between these two is greater than the standard deviation of the measured trace noise, the hypothesis is validated and thus the secret key block is estimated. The remaining blocks can be found through an iterative approach using the same method. DPA attacks have error correction properties which can extract keys from measurements that are too noisy for SPA to work [Koc99].

CPA is a statistical algorithm that uses the Pearson correlation coefficient to correlate data. The subkey is estimated by choosing the one with the highest correlation. This entire process is performed with each subkey, enabling the complete secret key to be collected. Compared to DPA, the CPA method requires fewer power traces. However, it requires an accurate power consumption model in accordance with the targeted hardware.

Finally, DFA works by injecting a fault into the ciphering process. DFA requires the output of the ciphering process, and the execution of the same ciphering process on the same input (potentially unknown). Different fault injection would lead to different propagation error in the ciphering process. For these different rounds, statistical assumptions can be made on secret keys. For example, by injecting eight faults in total into specific rounds of AES execution, the full key can be recovered [Bih97]. A more in-depth study on attacks is available in [Le08], including an explanation of attacks with device awareness (template attack and stochastic model attack).

2.2 Sound

The main purpose of sound channel attacks is to recover information passed through acoustic waves due to mechanical or sometimes electronic effects. Different papers focus on this approach and all highlight the need to train the system for identification. This leads to certain drawbacks such as limited recognition capacity (less than 80% without strong a priori hypothesis).

In 2010, Backes et al. proposed acoustic side-channel attacks on printers [Bac10], that were able to recover up to 72% of printed words. By assuming the language context, the attack achieves recognition rates up to 95% of whole texts. Sound acquisition was performed in the [20;48] kHz band and uses the Hidden Markov Model and the Viterbi algorithm which is regularly used in speech recognition, to determine the most likely sequence of printed

words.

Asonov et al. proposed a PC keyboard attack [Aso04] where they eavesdrop on the sound made by each pressed key. This attack is based on the hypothesis that the sound of clicks differs slightly from key to key. A neural network was used to classify clicks. The sound was acquired using a standard microphone and recorded with a standard PC sound card. 79% of the keys pressed were correctly guessed on the keyboard where the neural network was trained. It should also be noted that this ratio falls to 28% on an untrained keyboard. Sound can also be acquired from a legitimate microphone, for instance during a voice over IP call (VoIP). If the attacker can attend the call or access the audio content, he could pick up the keyboard sounds and therefore retrieve the typed message [Com17] [Ana18]. With a minimum audio rate of 20 kbit/s, 40% of the characters can be recovered, this number increases to 92% with the use of machine learning techniques trained with the writing habits of the person being listened to.

Sound can also be produced by electronic rather than mechanical origins. On modern computers, these acoustic emanations, typically caused by power regulation circuits, are correlated with CPU activity since the power draw is strongly affected by the current executed operation. Shamir et al. [Tro04] and Genkin et al. [Gen14] have shown that when ciphering is performed, the different RSA keys have distinguishable acoustic fingerprints. The full recovery of the key was successful using a cellular phone placed next to the computer or with a directive microphone from a distance of 10 meters. In [Tro04] and [Gen14], this was achieved by repeating the encryption process over many thousands of iterations, which is difficult to assess in an unsupervised attack. Although these methods are promising, due to the low microphone bandwidth (under 20 kHz using common microphones, and a few hundred kHz using ultrasound microphones [Tro04]), it is difficult to distinguish the different instructions.

To prevent keyboard sound eavesdropping, an on-screen keyboard can be used where the key selected is chosen by a mouse or a touch screen. However, another side-channel still exists as Genkin et al. [Gen18] overcame this countermeasure by listening to the power supply noise of a screen in order to recover displayed information (using the same method as a screen power consumption attack). When displaying an on-screen keyboard, word guessing accuracy of 75% was demonstrated using a convolutive neural network-based classifier after a training phase on the same screen.

Many computers have built-in microphones and speakers. Although designed to operate on

audible audio, these devices can also emit with reduced performance on the ultrasound band (and thus be undetectable to the human ear). All these elements make the use of ultrasound a way of achieving air-gap bridging. Hanspach et al. [Han13] in addition to ultrasound point-to-point communications, they also successfully created a mesh network with an error correction layer and a frequency-hopping spread spectrum transmission (in order to make the system more difficult to intercept). An experiment demonstrated a transmit message rate of approximately 20 bit/s up to a range of 19.7 m between two connected nodes.

Speakers are generally used to generate sound, but can work in reverse. By connecting a speaker to an audio input, it is possible to turn it into a microphone with much lower sensitivity than conventional microphones. Modern audio chips are capable of reversing the direction of connected headphones from output devices into input devices using software functionality. Guri et al. [Gur18c] use these special features to create a way of exfiltrating data from a computer which does not contain a microphone (for confidentiality reason) using the near-ultrasonic domain (18 kHz to 24 kHz). A range of about 9 m was achieved with a data rate of 10 bit/s. However, such acoustic communication relies on the availability of speakers on a computer which may not be available on the targeted device. To tackle this challenge, Guri et al. [Gur17a] also developed a way of generating an acoustic communication channel using a hard disk drive. By generating a specific pattern of reading operations on the disk, sound can be modulated. A data rate of 3 bit/s was demonstrated up to a range of 2 m. A similar approach can be taken by controlling the computer fan speed [Gur16c] for a data rate of 0.25 bit/s up to a range of 8 m.

2.3 Light

In order to spy on content displayed on a screen, a very basic approach proposed might be to spy on the surface of the screen with a telescope. This method, although very effective, requires a perfect line of sight between the screen and the observed device. Cathode Ray Tube screens (CRT) display their content by emitting an electron beam toward phosphor elements. This phosphor emits light according to a decay law specific for each phosphor type (i.e., for each chemical composition). The intensity of the light emitted by a CRT screen as a function of time corresponds to the video signal convolved with the impulse response of the phosphors. It has been demonstrated that sufficient high-frequency content remains in the emitted light (when the pixel is turned off) to enable reconstruction of the displayed content by deconvolving the signal received with a fast photosensor [Kuh02]. Khun was able to reconstruct a displayed

image from an out-of-sight CRT surface (i.e., from a reflection against a wall). The periodic nature of the signal can be used to reduce noise via periodic averaging.

Backes et al. [Bac08] extended this type of attack to a more modern screen using the low reflections of the screen image on eyeglasses, teapots, and even the user eye. Image acquisition was possible at up to 30 m. Later they extended their work to incorporate reflections on walls or clothes [Bac09] .

Light-emitting diodes (LEDs) are used in nearly every branch of electronics and in any situation where a fast, bright, highly visible indicator is required. Furthermore, LEDs are very fast, offering rapid response (tens of nanoseconds). In fact, common visible LEDs are sufficiently fast to be used as transmitters in fiber-optic data links. In some networking equipment, a LED is used to show line activity by flashing at the same rate as the information is transmitted. Loughry et al. [Lou02] demonstrated that besides echoing the same flow rate, the LEDs of some equipment are often directly connected to the data line they monitor and may leak into the optical domain. Since these leakages contain all the transmitted data, all parity checking and error correction features embedded in the data stream are available to the eavesdropper too. An error-free reconstruction from emitted light was possible at a distance of 38 m on various equipment such as routers, fax, modems, network cards.

Although humans can see luminous emissions, they are unable to perceive their rapid variations. It was, therefore, logical to use LED variations in order to intentionally exfiltrate information. Guri [Gur17b] successfully demonstrated that status LEDs placed on routers can exfiltrate information. He was able to establish a 1000 bit/s per LED communication channel. He shows that this bandwidth can be increased with the use of multiple LEDs. [Gur17c] takes a similar approach with a hard disk status LED with a final data rate of 4000 bit/s. [Sep14] uses a monitor status LED with a data rate of 20 bit/s.

Another method of using slow human visual perception was developed, in which data is leaked through hidden images displayed on a computer screen [Gur16a]. With this method, a nearly visible QR code (Quick Response code) is embedded on the computer screen. The nearly visible feature is achieved by very low contrast or fast flickering images.

In [Aso04] [Com17] [Ana18], the authors have shown the retrieval of a keyboard entry with a microphone. But the keyboard sounds can also be recorded with a so-called laser microphone. These microphones pick up the vibrations of a surface by projecting a laser on it and measuring the light time of flight, which depends on the slightest vibration. In [Bar09],

these laser microphones were used to listen on a surface close to a keyboard, achieving a 30 m range.

Lasers can also be used for injecting a sound into a microphone because of the photoacoustic effect. Sugawara et al. [Sug20] used a 5 mW laser to deliver voice commands to a voice assistant at 100 m. In such a device, no voice authentication is present, which results in a proximity-based threat model, where all nearby users are considered as legitimate. For mobile devices such as a smartphone or a tablet, a laser higher than 60 mW is required to inject audio content due to the thickest top layer protecting the microphone. It is noted that the laser does not produce any audible sound and is therefore perfectly undetectable by humans.

A video recording can be used to identify a password by looking at his arms movements [Bal19]. By knowing the keyboard layout and using the time elapsed between each keystroke, it is possible to estimate the distance between the used keys and a list of potential passwords is then deduced. In [Che18], the goal is also to retrieve a password but with recording eye movements. As a human eye naturally focuses on the pressed keys, it is possible to translate the eye movements into typed passwords.

It is to note that there is a side-channel due to thermal emissions of a target. However, this threat does not present a significant risk as it has a low bandwidth and a low practical range. Moreover, it is easy to detect due to the changes required at the target: change in the cooling strategy, additional computational load to increase the temperature of a component, etc.

3 Electromagnetic side-channels

The previously mentioned side-channel exhibit several weaknesses:

— The first one is their relatively short range, which is due in particular to their inability to pass through walls.

— The second drawback relates to signal quality. Combined with the short range, the signal can be distorted on the interception side, making interception difficult and sometimes impossible, even in nominal interception conditions. In addition, several devices can leak on the same side-channel rendering the isolation of a specific leak more complex, which is the case for power line.

— The third drawback relates to the decoding methods themselves: the equipment required for interception at longer distances is expensive and requires extensive expertise in signal

processing in order to be functional.

Electromagnetic waves, on the other hand, do not exhibit the defects mentioned above. They can pass through walls and do not require a line of sight. The modulation techniques and signal processing for these radio waves are very widely spread, making their access easier for everyone and allowing a long range. The usable radio spectrum is quite large, making it easy to hide some radio emissions. Although the cost of equipment can be high, very low-cost alternatives are available as we will discover in the next section. These specificities demonstrate a greater threat of attacks based on electromagnetic side-channels. The next section focuses on this type of side-channel. A synopsis of the studied papers and categorization of the attack targets are provided in Fig. 2-5.

Electromagnetic Emission Security (EMSEC) refers to the various compromising emanations and their uses for eavesdropping as well as information leaked through emanation evaluation. The NSA TEMPEST program is the first known one to address the risk of electromagnetic emanations in terms of security threats. Some of its documents have been declassified, i.e., TEMPEST basics and fundamental principles [NSA82], and the methods and equipment required in order to detect any compromising leak [NSA92].

EMSEC is linked to EMC [Jie13] insofar as both are based on the study of various electromagnetic emissions of a system. However, while EMC aims to protect equipment and people from electromagnetic radiation, EMSEC transcends EMC as it analyzes these emissions to verify that compromising information is not being emitted, and aims to eliminate them or, at least, make them unusable by an opponent.

We have separated the RF radiation into two aspects depending on the activeness (as described in 1.3). Although they share many common points, their threat is quite different, the active attack has a longer range consequently increasing their potential to be used by an attacker

3.1 Passive radio frequency side-channels

An electromagnetic field may be emitted from a conductor due to transitions in its state (i.e., applied voltage). In this section, conductors are a generic concept not limited to simple cables or Printed Circuit Board (PCB) track, but also including system components such as inductors and capacitors. These various leaks are directly correlated with the transmitted information on the conductor and therefore anyone recovering these leaks may be able to reconstruct the

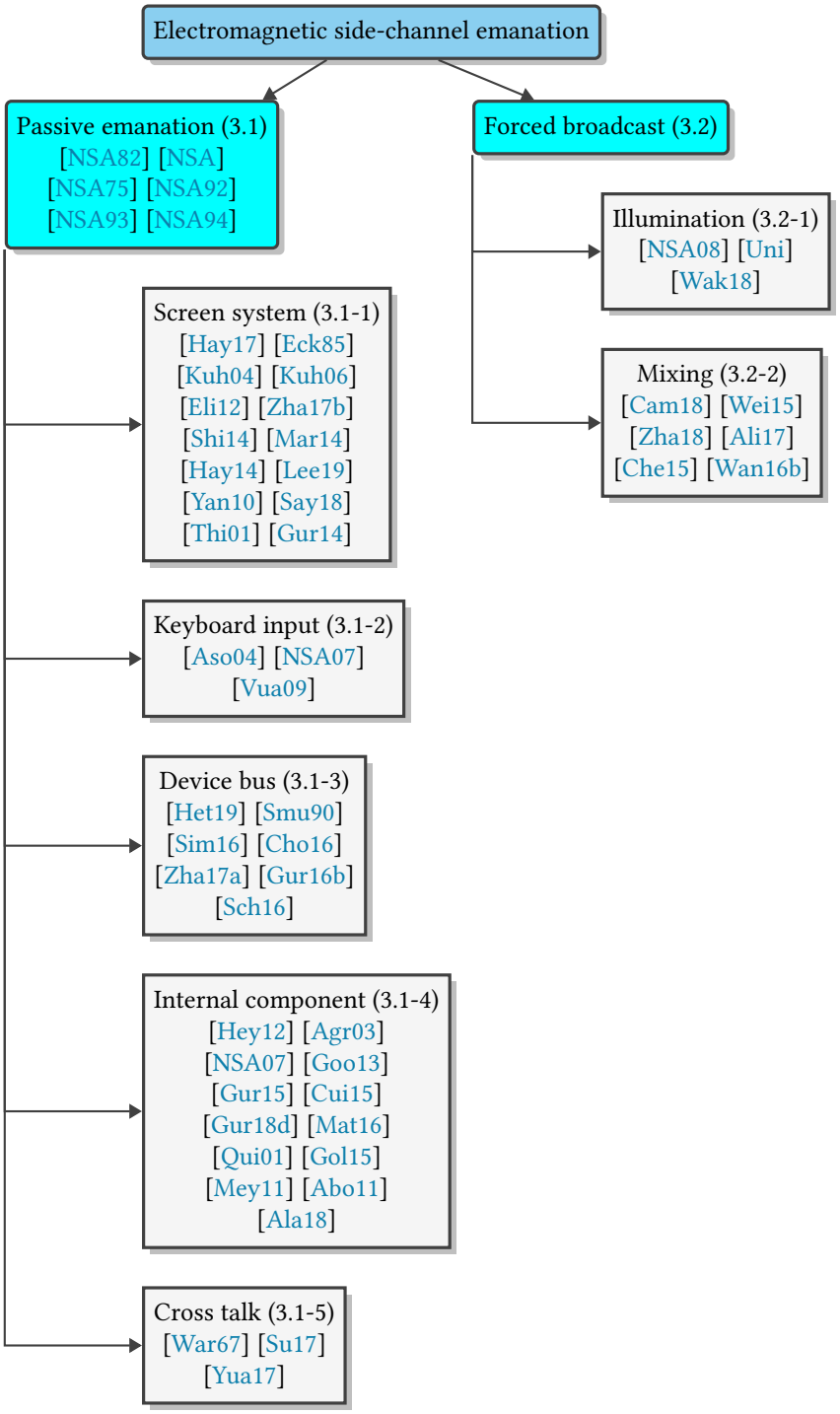


Figure 2-5 – Overview of the Electromagnetic side-channel categories.

original information. Fig. 2-6 illustrates two types of radiation leakages. The source signal (a) generates an electromagnetic field that can be directly picked up by an eavesdropper (b).

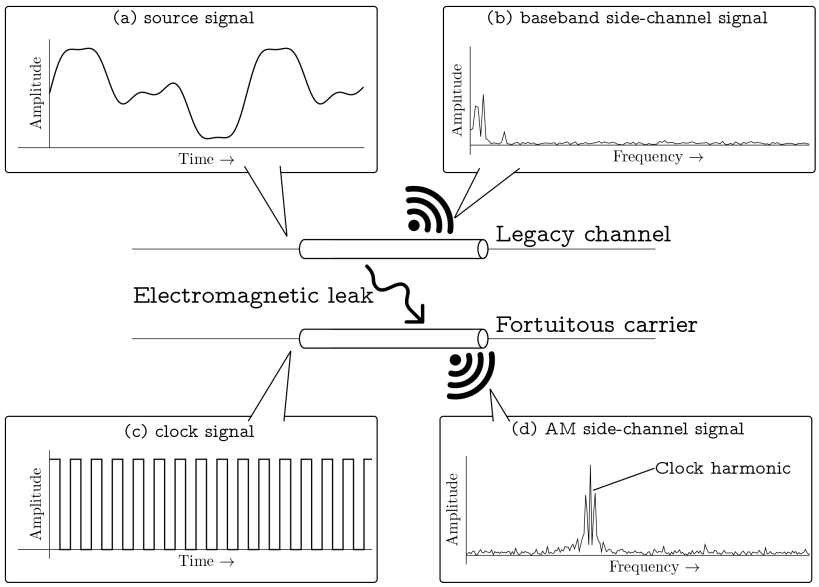


Figure 2-6 – Examples of electromagnetic radiation side-channel: AM and baseband side-channels.

The resulting side-channel is, however, in baseband and therefore has a short radio range. Nevertheless, the electromagnetic field of the source also influences the surrounding signals, for example a high frequency clock signal (c) and an Amplitude Modulation (AM) (d) is issued from the mixing of (c) and (b). This AM side-channel can be received at long radio range by an eavesdropper due to the high carrier frequency. Generally, the higher the transition rate (i.e., a data bus or clock signal), the greater the leaks. The following section will be segmented according to the nature of the device from which the source signal originates, to better stress out what hardware phenomenon is involved.

3.1-1 Leaks induced by screen systems

Screens are key elements of modern computer systems as they provide almost all the information to users. They are therefore ideal targets for an attacker because all the elements displayed are red data. A significant number of publications address this target.

Although CRT screens are now considered obsolete, they have laid the foundations for modern screens that still use some standards created for CRT monitors. Since a CRT screen does not store any data, a continuous stream of data is used (composed of one data line for grayscale images or three data lines for color). However, as screens have different resolutions,

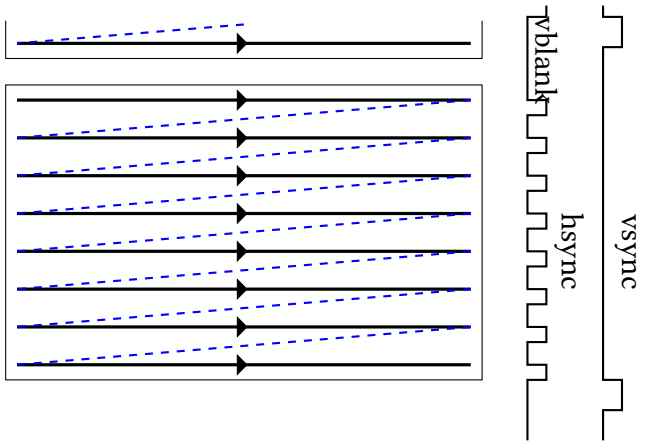


Figure 2-7 – CRT screen rendering strategies.

control signals must be added for line and image synchronizations, labeled *hsync* and *vsync* in Fig. 2-7.

A gap interval labeled *vblank* (vertical blanking interval) can be shown in Fig. 2-7 between the end of the *vsync* pulse and the start of a new image frame. During this interval, no image is displayed but data are still sent to the pixel lines all with the same value. There is also a small horizontal blanking interval after the *hsync* pulse for the same reason. Modern CRT screens do not require such long blanking intervals, and LCD displays require none, but the standards were established when the blanking was needed and thus still remain.

In 1985, Van Eck et al. [Eck85] described a method to infer the output of a CRT monitor at a distance of hundreds of meters using cheap off-the-shelf equipment.

Eavesdropping is achieved by bringing down to baseband a leaking frequency containing video information and sending it to a standard screen. The received signal does not contain the control lines that allow image synchronization. As it stands, the image is received, but moves horizontally and vertically. The quality of reception can be improved by externally generating the necessary synchronization signals and feeding them into the receiving screen.

Kuhn et al. [Kuh04] applied similar principles to flat-panel displays in 2005. The nuance is that these screens do not operate with such a high-voltage level, making the interception range shorter (around 10 meters). Moreover, the image is rendered row by row and not pixel by pixel like in CRT, so it is not possible to use the same process as Van Eck (because the energy of the pixels on the same row is all mixed at the same time). Kuhn uses another target in which the video data transits: the cable. This cable can leak outside the screen enclosure but also

inside in the path which leads to the electronic board. With modern eavesdropping systems, the two synchronization signals do not need to be physically synthesized. However, to produce a stable image, the vertical and horizontal resolutions as well as the refresh rate must be known. Eavesdropping reception is successfully achieved at a 10 m distance through two intermediate offices and without using a directional antenna [Kuh06]. With directive log-periodic antenna, the interception range was extended to 46 m [Eli12]. It should be noted that desktop or laptop screens are not the only displays on which it is possible to eavesdrop, as [Hay14] proves that smartphone or tablet screens are also likely to be intercepted.

A leakage channel can be found by checking the presence of periodic signals after an AM demodulation of the received signal [Hay17]. This periodicity is attributable to the redundancy of the information sent to the screen, since an image is generally very close to the one that precedes it. In addition, *vblank* gaps also generate strong redundancy that can be seen when autocorrelation is performed after AM demodulation on the leaked signal. Video signals can be easily separated from background noise because they are highly redundant. Displayed information usually remains unchanged for many seconds and, as a result, periodic averaging at vertical frequency (screen refresh rate, frequency between the *vblank* gaps) is very effective.

In [Zha17b], Zhang et al. have shown that more leaks appear when there are structural asymmetries or impedance discontinuities (e.g. in a poorly designed differential pair or cable connector). [Shi14] introduces a more powerful procedure than autocorrelations to detect a leakage frequency leveraging the frequency estimation of the spectral centroid. This algorithm can detect the horizontal and vertical synchronization frequencies of display video signal in a noisy environment.

Video eavesdropping is not completely blind, and some assumptions can be made to accelerate the estimation of screen parameters. Synchronization frequencies should be limited to certain specific values (as they are closely related to the display resolution) in order to reduce the kernel of frequencies to be checked. This is the method proposed by the open-source software TempestSDR [Mar14], which enables a display to be intercepted in real-time. Moreover, some parts of the image may be known in advance, e.g. task bars or application launchers that are always placed at specific locations on the screen. These elements can be used as a reference point to adjust the synchronization frequencies more quickly and accurately [Hay14].

Video leaks usually appear in the VHF band (30-300 MHz) but they can also be found in the UHF band (300-3 GHz). Hence, using an appropriate antenna, data acquisition can be achieved

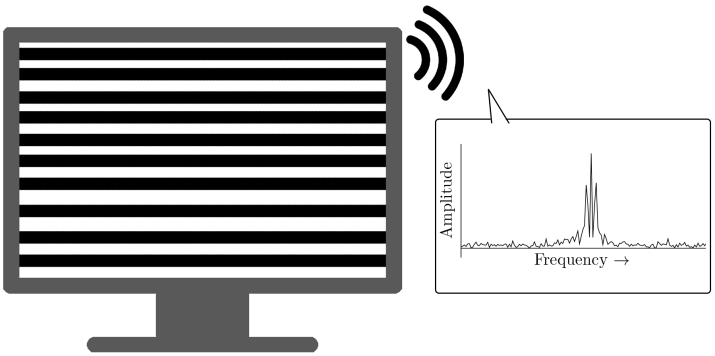


Figure 2-8 – "TEMPEST for Eliza" operating principle [Thi01].

with a commercially available dedicated TEMPEST spectrum analyzer (e.g. R&S FSET, Dynamic Sciences) but their price are very expensive, especially for hackers. A more affordable choice is to use Software-Defined Radio (SDR) to intercept the signal [Yan10] [Hay14] [Lee19] (the specific application of SDRs on signal interception is discussed in chapter 5). A bandwidth of 20 to 50 MHz is required to achieve the best video quality [Kuh06] but a lower-quality alternative is possible with a tunable TV receiver [Say18] which is in fact a small SDR with low bandwidth (around 2 MHz).

In 2001, a hacker [Thi01] released software enabling video cable leakage to be used to transmit data, which was audio content in this case. By displaying black and white strip patterns with specific sizing and spacing (see Fig. 2-8), it is possible to perform AM modulations that a simple AM short-wave radio receiver can pick up. This program not only generates a radio broadcast, but the strip pattern causes major fluctuations in the power supply and the audio content can be heard near the screen with a coil whine phenomenon. In 2014, Guri et al. [Gur14] adapted this transmission method to send more traditional data. Data are first encoded using Audio Frequency-Shift Keying (A-FSK) modulation and then, software converts the modulated symbols into pictures for the display. The resulting Frequency Modulation (FM) signal is carried in the 76-87.5 MHz band and the practical range was about 7 m at 480 bit/s.

3.1-2 Leaks induced by keyboards

An old example of keyboard eavesdropping occurred during the Second World War when the U.S. military used encrypted teletypewriter communications such as the Bell 131-B2. In a Bell laboratory, a researcher noticed by chance that every time the machine stepped, a peak appeared on an oscilloscope in a remote part of the laboratory. To prove the vulnerability of

the device, Bell engineers captured compromising emanations emitted by a Bell 131-B2, located 25 meters away. They were able to recover up to 75% of the plain text [NSA07].

More recently, in addition to sound fingerprint shown by Asonov et al. [Aso04], Vuagnoux et al. demonstrated the possibility of using electromagnetic radiation to recover transcribed information [Vua09]. The first approach involves listening to the wired data communication sent by the keyboard each time a key is pressed. Wired communication is achieved with the PS/2 protocol, in which each key is sent with a 11-bit word at 10-16.7 kHz. This retrieval method raises some difficulties because it targets baseband signals with a low propagation range. However, this signal can also be unintentionally modulated internally by the keyboard electronics, e.g. by the clock of the internal microcontroller. In this case, AM or FM modulation is generated with the clock (or its harmonics) as carriers. These transmissions are generally less disrupted by noise and obstacles such as walls and floors than baseband signals.

While the two previously mentioned methods target keyboard communication protocols that are now old and little used, the third method may be successfully used with modern wired and wireless keyboards. In order to reduce costs, each key of a keyboard is not directly connected to a microcontroller input but is placed at an intersection between a row and a column of a matrix. When a key is pressed, the matrix leak is altered at the column where the key is placed. Since keyboard clocks are not the same and not synchronized with each other, it is possible to distinguish between multiple keyboards in the same room. The maximum successful (more than 95% of keystrokes correctly guessed) eavesdropping range is about 20 m, with compromising emanations found in frequency bands between 25 MHz and 300 MHz and recovered with an SDR.

3.1-3 Leaks induced by device buses

The higher the throughput of a bus, the more transitions are generated and therefore the more radiation is emitted [Mol03]. However, the higher the throughput, the faster the receiver must be to reconstruct data successfully.

Smulders et al. [Smu90] showed the feasibility of eavesdropping on content passing through an RS-232 bus. The flow rate of this bus is relatively low (less than 200 kbit/s), and therefore its acquisition in baseband is relatively impractical. However, as with a keyboard, the communication bus can be found modulated with a system clock. During experimentation, the bus was found on the FM band at harmonics of the system clock signal (more precisely in 10-130 MHz).

The small bandwidth of the RS-232 link reduces the need for very high bandwidth reception materials. In fact, interception is feasible with standard AM/FM radio receivers (intended for broadcasting music) with successful reception at 9 m.

More recent communication buses are also sensitive to eavesdropping. In [Sch16] Schulz et al. have shown the feasibility of listening over a 10 Mbps ethernet cable. They used a near-field probe and a 20 MHz SDR to intercept the information passing through the cable. The ethernet forward error correcting codes can furthermore be used by the eavesdropper to increase the range of the attack. For higher bandwidths, more probes are needed since Ethernet standards then use multiple twisted pairs at the same time.

USB is one of the most widely used systems present on all computers and several attacks are described in the open literature [Sim16]. The principal difficulty of decoding USB lies in the Non-Return to Zero Inverted (NRZI) coding used. This coding does not code each bit independently, but codes them according to the previous bits. The NRZI creates a voltage transition if the bit is '1' and otherwise remains in the previous state. It is therefore necessary to intercept information bit-perfect [Cho16]. Zhang et al. [Zha17a] use neural networks (an echo state network) to classify received signals according to their Hamming weight (number of '1' for NRZI) and then select the most likely choice from the coding possibilities, leading to a complex yet functional system.

Apart from eavesdropping, it is also possible to use a USB bus to intentionally leak information. [Gur16b] found that transmitting a sequence full of '1' bits to a USB device can generate a detectable emission between 240-480 MHz. This is due to the fact that '1' bits generate a rapid voltage change on each clock cycle. They performed Binary FSK (B-FSK) modulation where the frequency was adjusted with the addition of '0' between '1' (to prevent a voltage shift) in order to lower the carrier leakage frequency and enable the modulation of the data stream, a data rate of 640 bit/s was achieved.

3.1-4 Leaks induced by a system internal components

The main goal of air-gapped computers is to prevent information leaks. These computers are isolated from conventional networks such as the Internet. This, however, does not prevent potential threats due to breaches in network isolation (for example, the computer may be infected by malware if users plug in their phones to recharge them). In order to prevent any leak, the number of devices that are connectable to an air-gapped computer must be minimized

and limited to a screen and wired mouse keyboard.

However, Guri et al. [Gur15] showed that preventing data exfiltration is insufficient. They demonstrated the potential for using RAM to generate and modulate leaks, by invoking specific memory-related instructions to modulate and transmit electromagnetic signals at cellular frequencies. The leaks occur in the 600-1100 MHz range depending on the generation of Double Data Rate Synchronous Dynamic RAM (DDR-RAM) used. The modulation scheme used was a modified Binary Amplitude-Shift Keying (B-ASK). The '0' is obtained with a near zero amplitude (which is in fact the normal leakage level when nothing special is done). The '1' is obtained by invoking a memory instruction in order to leak a strong signal. This communication channel has a range of about 30 m at a 1 kbit/s data rate with a dedicated SDR receiver.

Funtenna [Cui15] is an open-source software payload that intentionally makes its host hardware act as an improvised RF transmitter (although not initially designed for electromagnetic communication). Mainly intended for embedded systems rather than computers, this software uses system standard General Purpose Input Output (GPIO), Pulse Width Modulation (PWM) or Universal Asynchronous Receiver Transmitter (UART) output to generate FSK or ASK modulation below 100 MHz.

In 2018, Guri et al. [Gur18d] presented a type of malware that can exfiltrate data via low frequency (less than 50 Hz) magnetic signals induced by computer CPU cores. By controlling the workload of the CPU cores with multiple sub-carrier modulations (one carrier per core), a bit rate of 40 bit/s was achieved at 1 m.

Matyunin et al. [Mat16] suggested using a hard disk drive magnetic head to generate magnetic emissions by invoking basic read/write operations on the disk. They used ASK modulations but due to such devices low power consumption, the practical range is lower than with a CPU core at about 15 cm with a data rate of 2 bit/s.

Hardened cryptography systems aim to prevent data retrieval resulting from side-channel attacks by giving the cryptography system a specific architectural design. Although circuits can be protected against intrusive attacks and power analysis, it is more difficult to secure them against radiation attacks since these are capable of targeting very precise points of a chip with near-field probes while remaining perfectly undetectable. These attacks are known as Electro-Magnetic Attacks (EMA) [Qui01] and all the processing and methods applicable to SPA [Man02] and DPA [Koc99] are possible with EMA [Le08]. For template attacks [Hey12] the threat mainly lies in the ability to recover a key with few traces. The attacker uses a

device identical to the target in order to thoroughly understand its functioning. A great deal of pre-processing can then be performed offline, which fastens the attack.

It should, however, be emphasized that other methods exist, such as using Short Time Fourier Transform (STFT) or AM demodulation to produce a narrower band signal to decrease the effect of noise [Mey11].

Aboulkassimi et al. [Abo11] demonstrated the possibility of extracting the key from AES encryption performed on a mobile phone using a near-field electromagnetic probe. [Agr03] found cryptographic related emanations using near-field probes as well as far-field probes (with log-periodic antenna). Goller et al. [Gol15] continued this study of using far-field probes to recover encryption keys. They successfully recovered them at a distance of 80 cm using a commercially available SDR (middle-end type). They also carried out a comparison between the use of a good quality SDR and a low-end SDR (e.g. a DVB-T stick). The difference of signal-to-noise ratio between the middle-end and low-end system was only about 2 dB, but with reduced received power for the low-end system, which, in practice dramatically reduces the recovery range. Consequently, compared to a maximum distance of 80 cm for the middle-end system, the low-cost system is unable to receive a usable signal at a distance greater than 10 cm.

Timing attacks target the execution time of cryptographic instructions. To prevent these attacks, constant-time security is used where every security-critical operation is modified so that it lasts exactly the same time regardless of the input data. Alam et al. [Ala18] utilized security features present in RSA implementation (openSSl) to speed up the extraction of the key. In fact, they only needed one trace to recover the RSA private key using a near-field probe (over 95% successful recovery). They detected the ciphering process through a specific pattern induced by the constant-time countermeasure (protects against timing attack).

3.1-5 Leaks induced by cross-talk

Cross-talk is a radiation side-channel where the radiation originates from a legacy channel, as illustrated by Fig. 2-9. The source signal (b) propagates through an adjacent conductor (a) due to a radiation phenomenon (c), resulting in a leak in the adjacent conductor (d). While this does not change the content carried by the leak, it does change the way it is propagated. Cross-talk phenomena were among the first threats highlighted by TEMPEST [NSA82]. The NSA created a special guide to reduce the incidence of this type of leak [NSA]. In 1967, Ware [War67] suggested that it was conceivable that sensitive information was being eavesdropped by cross-talk but

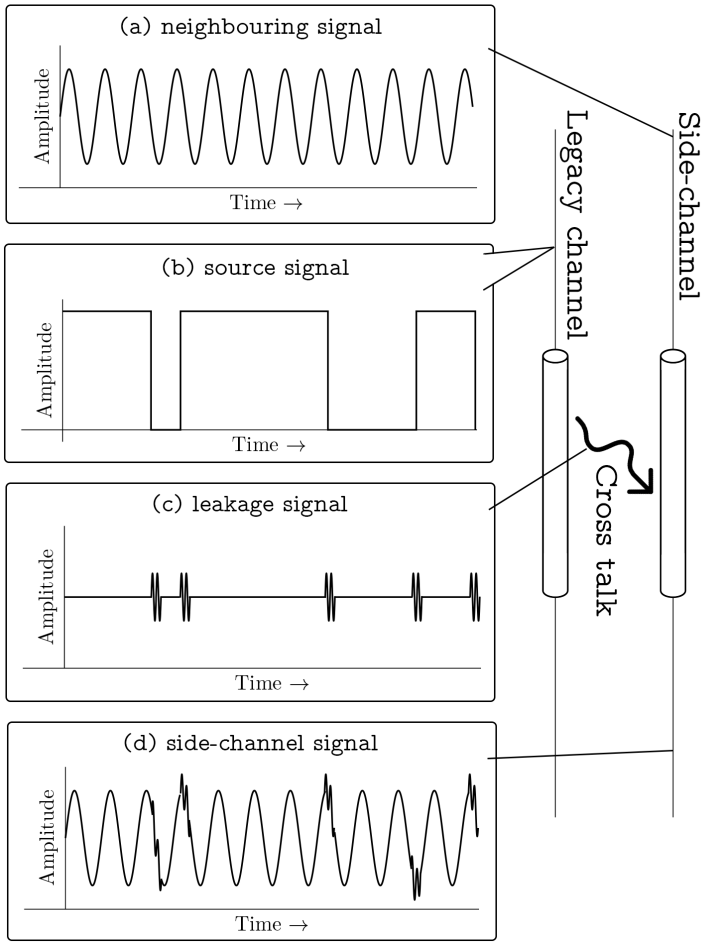


Figure 2-9 – Cross-talk model.

provided no evidence of it.

In a USB, the only security feature lies in the fact that all devices cannot monitor total bus traffic. This security is performed by active components (placed within the bus line) that redirect data flows to the right device. But, in fact, all downstream data (host to devices) are broadcast to all devices while only the upstream is unicast and redirected by the active component. This implementation has major security problems since all USB devices can listen plainly to all the information sent by the host even if they are not the rightful recipients. Su et al. [Su17] highlighted a cross-talk phenomenon within USB hubs that provide packet redirection. Most of the time, USB hubs consist of a single-chip solution, and isolation between lines is present. Nevertheless, the small size of the chip and the high data rate can still produce a cross-talk phenomenon between adjacent USB lines. The researchers successfully demonstrated

the interception of USB fingerprint scanner data (device to host stream) from an infected USB gadgets such as a USB light connected to the same USB hub as the scanner [Su17].

In [Yua17], an in-depth study of cross-talk phenomena is conducted with particular attention to the reconstruction of digital signals transmitted along differential connections. This paper addresses theoretical aspects and seeks to determine the conditions under which eavesdropping of information is possible. Reconstitution accuracy is dependent on the speed of the signal being listened to (risk of overlapping bits), and the quality of the cable where the red signal is leaking is also of significance. Although several wires can be used to enhance reconstruction quality, algorithmic complexity drastically increases and becomes prohibitive for real-time constraints.

3.2 Forced broadcast side-channels

Another kind of electromagnetic side-channels is the forced-broadcast, represented in Fig. 2-10), where the main carrier (a) is issued from outside the target and reflects while being altered by the target (b). This reflection thus leads to a leak of information (c). Forced broadcast is composed of two subcategories, namely illumination and mixing, considered respectively as active and passive attacks.

Radio illumination attacks force a system (screen, keyboard, microphone, etc.) to leak by mixing them with an external carrier and by studying the reflected wave altered by the targeted system. Most of the time, the mixing process results in an AM or FM modulation. These attacks are extremely powerful due to their ability to accurately choose a target (directive antenna are generally used) and their long-range (the attacker has plain control on the carrier transmission power and frequency), but they also present a significant disadvantage. The carrier used is generated by the attacker and is not a signal that normally exists in the conventional radio spectrum. As such, the target may notice this abnormal carrier and detect the attack.

A mixing leak is similar to the previous scenario except that the carrier modulated by the system originates from a source other than the attacker (WiFi access point, Digital Enhanced Cordless Telecommunications (DECT) phone, embedded radio transmitters, etc.). The hazards of such attacks are greater because the external signal comes from an authorized source and is therefore not categorized as an abnormal channel source, unlike illumination attacks. In these scenarios, since the attacker becomes totally passive, its detection is impossible [Wei15]. On the downside, weaker signal quality is assumed.

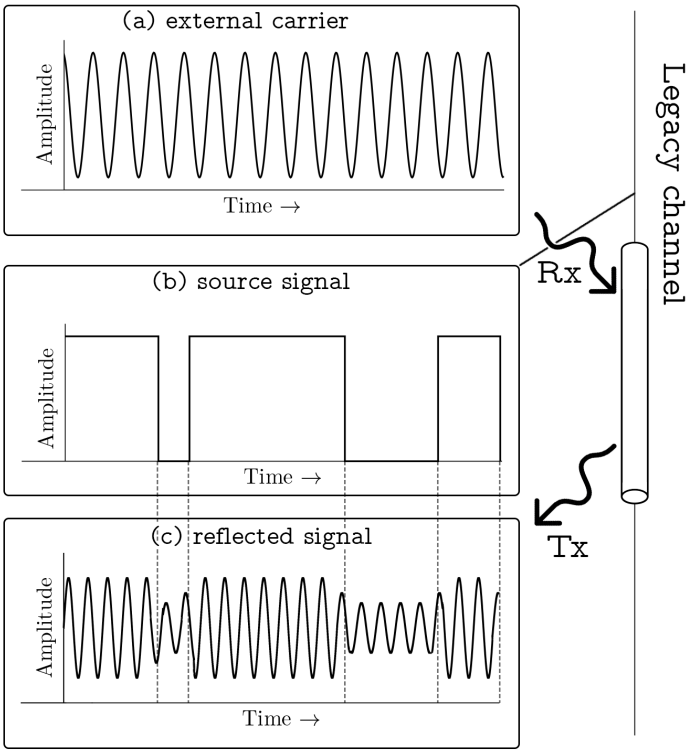


Figure 2-10 – Example of forced broadcast side-channel.

3.2-1 Illumination attack

The Thing, also known as The Great Seal Bug [Uni] (1945), was an unpowered covert listening device developed in the Soviet Union. The device was designed to recover speech. To ensure maximum concealment and eliminate maintenance requirements, the "bug" was totally passive and had to be remotely powered by a radio beam in order to be operated. It consisted of a resonant cavity microphone (see Fig. 2-11), which is basically a tuned circuit (T), composed of an inductance and a capacitance. Part of the capacitance is variable as it is effectively a microphone (M). An antenna (A) is coupled (C) to the tuned circuit. Any sound causes the membrane of the capacitance to vibrate, which decreases and increases the resonance frequency of the tuned circuit. As a result, the device produces a combination of AM and FM signals when illuminated by a beam. In practice, only the AM component was used. The device was planted in 1945 but only discovered in 1952. In 1951, a radio operator accidentally intercepted emissions returned by the bug, but could not discover their origin. Several detections of this type were made without pinpointing their origin. It was only one year later, after specific equipment

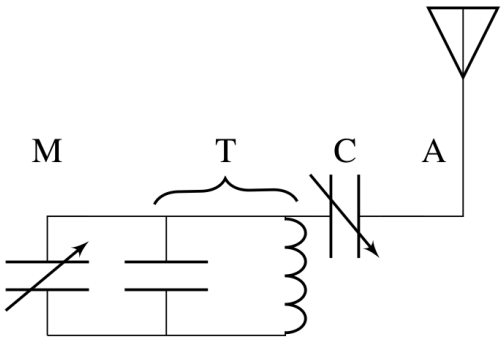


Figure 2-11 – Schematic of The Thing: (M) Microphone, (T) Tuned circuit, (C) Coupler, (A) Antenna.

was deployed and the radio spectrum was monitored continuously over several days, that the device was discovered. In order to detect it, the device had to propagate back a signal and therefore had to be illuminated. It is this ability to be deactivated remotely and not to radiate when turned off that makes this type of attack extremely discreet and thus dangerous, since a degree of chance is required to properly detect the attack.

The NSA used the same principle of a device powered by external sources in their VAGRANT program [NSA08]. Although information regarding this program is supposedly classified, some information has become public knowledge due to the Snowden leaks. The VAGRANT program aims to spy on a user computer interface by installing bugs in the equipment being monitored. Two types of bugs are used. An extremely basic version, which is useful for high-speed side-channels such as video cables, consists of a simple radio reflector placed on the cable. When a continuous carrier radio stream illuminates it, the data is modulated and redirected to the transmitter. A more complex version uses logic components powered by the radio beam to reflect the signal with FSK modulation (between the data lines it taps and an internal clock). The maximal range achieved by such a system was about 12 km in excellent conditions, which is significantly longer than for passive radio eavesdropping.

Due to the nature of the VAGRANT program, no scientific papers have been published on this type of attack, although some explanations may be found in [GBP14] and [Oss14]. In particular, no real expressions of the distances, means of recovering red signals and manufacture of bugs are explained. However, in 2018, Wakabayashi et al. [Wak18] addressed this issue. Rather than recreating the exact same version as the NSA, they used the principle of illumination attacks to create their own version: the bug used here is a field-effect transistor (FET) where

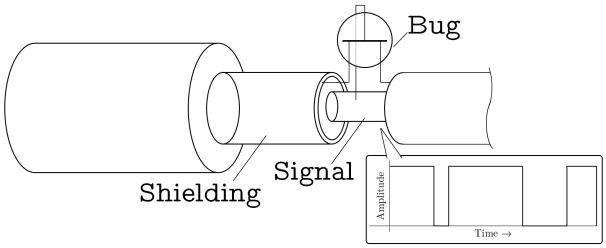


Figure 2-12 – Schematic of an FET bug placed on a data wire.

the gate of the FET is attached to a wire (see Fig. 2-12 to installation schematic). The wire shield is then used as a dipole antenna, which is long and perfectly camouflaged. In such attacks, the key issue is to determine the circuit resonance frequency since the sent beam is reflected at this particular frequency. The study showed that contrary to the general assumption, it is not only the length of the antenna that determines the frequency but also its shape and the bus to which it is connected. Not every FET transistor is usable for this purpose, and the one used exhibited high dynamic range with low noise (specific to RF applications). An interception data rate of approximately 10 Mbit/s was achieved within 10 m using an SDR radio. However, this limitation is only due to the performance of the SDR used and can be overcome with higher-end equipment (the setup used was estimated at USD 5,000, which is affordable to the general public).

Kinugawa et al. [Kin19] extended the scope of the aforementioned attack by targeting PCB in addition to data cables. They were able to record the power supply of a cryptographic circuit, allowing the interception of a secret RSA key up to 5 m. Although some components have been added on the PCB, only a qualified person will be able to differentiate them from the legitimate components. Other interesting targets are smart speakers as they constantly listen to their surroundings in order to detect wake-up words. Kinugawa et al. were able to demonstrate the reception of the inner microphone up to 5 m with a bug placed on an internal wire.

3.2-2 Mixing with external carrier attack

With the recent advent of SoC, processors, memories and radio communication circuits can be combined on the same circuit to create smaller and cheaper devices. Although these systems are mainly made up of digital components, an analog part still remains for power supply and communication. While modern radio protocols are digital and most of their layers are implemented in the digital domain, the generation, amplification, and transmission of radio

frequency signals are by nature analog operations. The recovery of keys through an analysis of various emanations of a processor or power supply has been widely studied [Mey11] [Gol15]. However, all these methods use probes that must be placed very close to the component being studied, which limits the criticality of the threat. In a system where a SoC used encryption, a shared power supply between the processor and the radio circuit, on the one hand, and physical proximity, on the other hand, may both result in the presence of processor activity on the transmitted radio signals. If, during a radio communication phase, the processor performs encryption (which is very common as secure radio exchange packets are encrypted on the fly), processor activity may then be present in the radio frequency signal. Therefore, it is possible to recover the encryption key remotely by listening to the radio communication [Cho20]. In [Cam18] [Cam20], Camurati et al. presented a successful attack of this kind. The target was a SoC nRF52832 with an internal Bluetooth link from which it was possible to retrieve the RSA encryption key at a distance of 10 m with methods very similar to EMA [Qui01].

Speakers are widely used in conference systems and may be, by their nature, carriers of confidential information. Soundproof walls are generally used as a countermeasure against sound leaks (see Section 2-2). However, it is still possible to recover sound from outside such secured rooms, not by using sound generated by the loudspeaker, but the small disturbances it causes to radio frequency signals in its environment [Wei15]. The advantage of this method using radio frequency waves rather than sound waves is that it does not require a perfect line of sight with the loudspeaker or physical access to the secured room. As with other leaks, AM modulation is performed between the sound and the external radio frequency signal. The signals disturbed by the loudspeaker may come from different sources. First, they may originate from the attacker. In such a scenario, the attacker is able to fully control the interception parameters and choose a frequency that allows the best signal recovery. In addition, it is not necessary to transmit continuously: signal bursts can be used since the voice can be sampled at a lower frequency (compared to a conventional radio frequency signal) and still be perfectly intelligible. This attack, although allowing excellent audio quality, requires a signal to be to actively transmitted. A different approach to radio illumination is to use radio frequency signals that are already present to intercept the sound signal. For example, this is the case of WiFi waves [Wei15] that are often present in conference rooms to allow Internet access. The threat may also be increased by the presence of a WiFi hot spot, as these are placed to maximize radio coverage. This naturally increases the range of compromising radio frequency waves, allowing potential interception even at a great distance from the loudspeaker.

The prominence of radio communication and the omnipresence of energy in radio bands make it possible either to conceal a radio illumination attack or use radio waves that are already present. In [Zha18], a WiFi router was used to create cheap radar devices. This leveraged the fact that WiFi frequencies travel through walls and are reflected on the human body, with the purpose of seeking the presence of a person as well as his movements. Like all radar images, a high signal processing workload is required to obtain usable data. Thus, in [Zha18], a deep neural network is used to detect joint movement. Moreover, this works even in the presence of a wall and in the absence of light sources, unlike the camera-based method. The radio-based system has been proved almost as accurate as the conventional camera-based system up to the maximum range of the router used (which was 12 m).

WiFi may also result in a higher resolution when, instead of targeting a whole body, only a part is monitored. Ali et al. [Ali17] applied gesture recognition to user finger and hand movements above a keyboard in order to recognize keystrokes. They used the fluctuations in the instantaneous channel state information caused by small hand movements. High key accuracy of 89.7% was obtained with k-nearest neighbor classifiers and a multi-antenna setup with off-the-shelf WiFi devices. A similar approach was adopted [Che15] with the use of an SDR-based system specially designed for this purpose. This increased correct key estimation to 91.8%.

Finally, application can be further extended by targeting a person face. The subtle movements involved require major signal processing as well as beamforming to reduce the noise. However, Wang et al. [Wan16b] show the feasibility of such an approach. Their system detects nine vowels and consonant patterns and uses context-based error correction techniques to correctly estimate the word pronounced. An efficiency of 91% was observed in a direct line of sight, this ratio drops to 18% if a wall is interposed.

3.3 Discussion and remarks

The previous section highlighted the main characteristics of side-channel emanations. We pointed out that the main disrupting element lies in the medium used. Indeed, the goal is invariably to recover red data: data originating or ending on the user side (screen, speaker, keyboard, etc.), data bus information, or the cryptography key, etc. Even if the methods used to recover data may be independent from the medium, they mostly exploit the medium specificities to improve their performance. Consequently, albeit targeting the same red signal, the medium used will have a significant influence on the ability to retrieve information and therefore on

the level of the threat.

Side-channel attacks based on electromagnetic signals are particularly threatening to security and privacy due to the diversity of targets as well as the different possible paths. The low cost of some methods and the low knowledge requirements for carrying them out making them even more cause for concern. Some improvement can be made with method such as multiple-input and multiple-output (MIMO) which allows using multiple antennas to exploit multipath propagation and therefore greatly improve the noise performance as well as ease to target devices.

On top of that, several indicators tend to exacerbate this issue. First, the evolution of current technology tends to increase the frequency of system clocks and the number of clock regions. This has two consequences: i) the number of side-channels may increase, and ii) these side-channels may appear on a wide carrier frequency range (and reach values more favorable to their propagation). Secondly, for a defender, potential threats are often identified based initially on channel sounding. Thus, finding a potential compromising signal is more difficult due to the increasing use of radio signals nowadays. Finally, this section has shown that the cost of interception is decreasing, with smaller yet effective equipment available so that powerful techniques such as machine learning [Het19] can even be envisioned.

The Table A-1 of the appendix is a summary of the various side-channel emanations discussed in this chapter including a classification by target, side-channel, activeness, intentionality and the practical range achieved. This allows having a synthesized view of the several potential attacks with the goal of improving the apprehension of their threat level.

Our subject of interest lies in the EM side-channels and in particular those which are difficult to detect by conventional means and not really covered in the literature, this includes the forced broadcast but also all the leaks affecting the transmitter which will then broadcast the leak in their messages.

Chapter **THREE**

INTERCEPTION METHODS OF FREQUENCY HOPPING SIGNALS

Contents

Frequency Hopping (FH) is a method of transmitting radio signals by changing the carrier frequency among many distinct frequencies occupying a large spectral band. The various frequencies shift a baseband signal thereby creating different channels. The change or *hop* sequence of the FH channels is defined by a sequence known by both the transmitter and receiver in advance and is usually generated with a pseudo random generator. An example can be seen in Fig. 1-2 that depicts a time-frequency grid of Bluetooth signal.

This chapter will firstly start by introducing FH in detail in Section 3-1, its origin, the

evolution of its use over time and, of course, why our study is about intercepting these kinds of signals. Section 3-2 will focus on the state of the art on frequency hopping intercepting systems. Finally, Section 3-3 lists our use case of side-channel interception and will also present our interception architecture.

1 Frequency Hopping context

1.1 Historical background

There is no exact year of invention for the FH, this concept has existed since the use of radio waves for telecommunication because it allows making a simple multi-user management (with different channel allocation). However, several patent applications on concepts more or less similar to the FH have been filed. The first practical application of FH was during the WWI, were German operators manually switched between specified frequencies to prevent eavesdropping from their adversary [Wil15]. Indeed, an eavesdropper not being aware of the frequency pattern to be used could not effectively listen to the communication because it was almost impossible for him to determine the current frequency on which the signal was transmitted. However, this method is not automatic and requires human intervention at both transmission and reception sides.

The first appearance of an automatic system is found in the patent [Bro29] in 1929, in which the hop sequence is cyclic and can take less than 10 possible values. A more elaborated solution, which is the one chosen as the first frequency hopping system, is the patent *"Secret Communications System"* [Lam41] in 1941. It uses a piano-roll to select among up to 88 frequencies, which must therefore be physically delivered before any transmission. In this way, a more complex sequence than the above patent can be used, the original purpose of this system was to improve the stealth and jamming resilience of torpedoes.

1.2 Notations

Frequency hopping signals can be represented along with a grid, composed of two axes:

- an axis for time graduated in time slots T_s,
- an axis for carrier frequencies graduated in channels N.

To give some examples, Bluetooth technology uses FH for transmission and has a time slot of 625μs over 80 channels, 40 in its low energy standard (BLE) with 80 MHz bandwidth in the 2.4 GHz ISM band. For obvious reasons, the military also uses this type of transmission, we can mention American *HAVE QUICK I* and *HAVE QUICK II* but little information can be found about them except they use the $225 - 400$ MHz UHF band. However, we do have some information about *SINCGARS* [cry] which has 2320 channels between 30 and 88 MHz and hops at 111 hop/sec. It is to be noted that these protocols are still available but that the radios using them are now SDR, which shows the great flexibility of this type of technology.

Fig. 3-1 depicts this grid (the blue area represents a signal), we consider that the time slots have all the same duration, and channels have the same bandwidth. A frequential and temporal sporadicity can be seen:

— Frequential, it corresponds to the fact that generally two successive transmissions do not happen in the same channel, and therefore the receiver must know the channel sequence which can be either in a pseudo-random sequence or one of the direct neighbor channels, as can be seen in the top left quarter of the figure. In the latter case, the knowledge of the hop spacing makes the job easier for an FH detector because it does not need to search in the whole spectrum but only the adjacent channels.

— Temporal, it corresponds to the fact that there is not necessarily a signal in each time slot. This can be seen in the top of the image, where on the left there is no temporal sporadicity while on the right there is.

Two phenomena of grid desynchronization are visible in the lower right quarter of the figure, due to a mis-synchronization of the receiver, *CFO* and *time unsync*, respectively a frequency and temporal error. They can lead to a wrong information retrieval from the legitimate receiver as well as an additional error from a possible eavesdropper who must already deal with the inherent difficulties of side-channel listening.

We consider a baseband message b (e.g *TRANSEC* or *Bluetooth*) of bandwidth F_b. This message carries no usable sensitive data and is called a black signal, but through a side-channel, a red signal can be mixed within the black signal (see Section 3-1 for a side-channel mixing model). The notion of red/black signal is not discussed in this part because only the detection of active channel is considered in the interception of FH signal literature.

This baseband message is then modulated following the FH sequence, resulting in a F_s

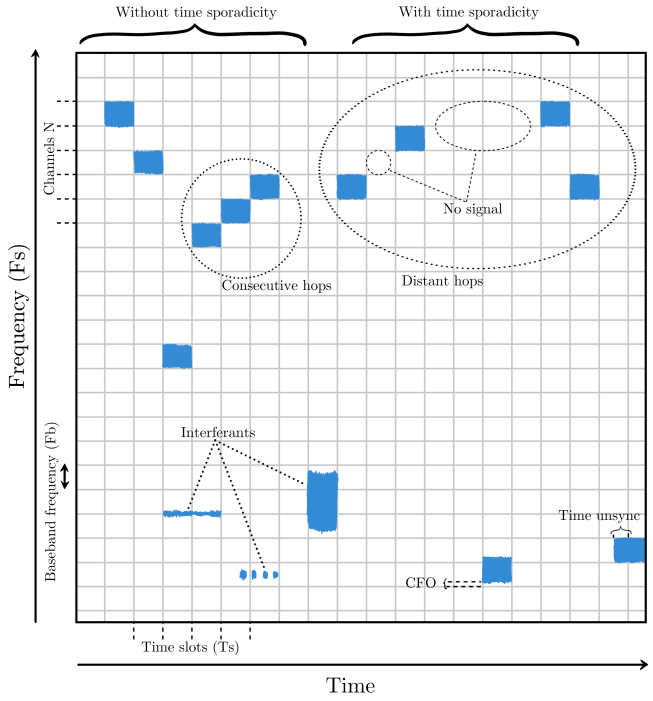


Figure 3-1 – Time Frequency plot of an FH model.

bandwidth transmitted signal:

$$x(t) = b(t)e^{2j\pi f_p(t)t}\alpha_{T_s}(t), p \in [0, \ldots N-1] \tag{3-1}$$

with $f_p(t)$ the associated channel frequency to transmitted symbols of $b(t)$, p the channel index, α_{T_s} the sporadicity factor expressing whether transmission occurs during a time slot:

$$\alpha_{T_s}(t) = \begin{cases} 0 \text{ if sporadicity,} \\ 1 \text{ otherwise,} \end{cases} \tag{3-2}$$

Assuming that N evenly distributed channels can be used, then

$$f_p(t) = p(t) \times F_b = \frac{p(t) \times F_s}{N},\ p(t) \in [0; N-1] \forall t. \tag{3-3}$$

The same FH index p is maintained constant for several consecutive time slots which define the so-called slot duration T_s, and $\lambda = \frac{T_s}{N}$ is the repetition factor, this value will be important for the Chapter 4 and is used to define the number of windows of width N contained in a time slot T_s. In the following, the received signal $d(t)$ is the transmitted signal only affected by a delay τ and impaired by an Additive White Gaussian Noise (AWGN) $w(t)$ of variance σ_w^2 and is therefore expressed as

$$d(t) = x(t - \tau) + w(t). \tag{3-4}$$

The baseband signal bandwidth F_b is lower than the FH signal bandwidth F_s, which adds a phenomenon of frequency spreading so that the FH signals are also called Frequency-Hopping Spread Spectrum (FHSS).

Frequency hopping has several advantages:

— An easier sharing between several users of the same spectral band with minimal mutual interference [Tor15].

— An increased robustness against interference, whether the interference is due to natural fading of the propagation medium (in which case changing the frequency avoids the interference), or due to an intentional jamming.

— A higher frequency diversity, when the carrier frequency changes in middle of time slot i.e., the hopping rate is greater than data rate [Tor15], which is called fast frequency hopping.

— An increased resistance towards interception because the eavesdropper is not aware of the hop sequence and has to observe a large spectrum continuously to catch the transmission or being extremely lucky to stumble upon the right channel frequency.

1.3 Why eavesdropping such signals ?

Listening to FH signals can have several uses, which are not only about snooping. The radio spectrum is wide but only a small part of it is usable by the public, who must then resort to the so-called industrial, Scientific and Medical (ISM) radio bands. Some of these bands, especially the 2.4 GHz, are extremely used, by WiFI, Bluetooth, Zigbee, Digital enhanced cordless telecommunications (DECT), … Some portions of the ISM band are heavily used while some others are rarely used. Not sharing the radio spectrum among users can result in the creation of unwanted denial of service events. Being able to detect unused space with an FH detector helps improve bandwidth efficiency. This is for instance how BT avoids channels with interference [Gol].

An FH detector can also be used to ensure the proper functioning of a radio transmitter that uses FH precisely without knowing the hop sequence. In the absence of a suitable detector, it would be very complex to check that the data being sent is correct. Moreover, monitoring the radio frequency of a device allows to analyze it in a non-intrusive way, which does not alter its operation but also renders the detection of this monitoring more difficult.

FH detectors can, of course, be used for security purpose, to eavesdrop an FH signal to check its content and may be employed to detect any side-channel affecting an FH radio, with the aim of discovering its cause to estimate its dangerousness and to suppress it. This procedure can also be used to detect a radio transmission that does not comply with the radio standard, e.g., a device that has been tampered with so that it transmits confidential information in addition to its normal operation. In this case only an FH detector will be able to observe this phenomenon. It is precisely to detect these cases that the present thesis has been carried out.

2 State of the art of FH interception methods

Most of the FH detections are based on periodic analysis of the spectrum and observation of an energy change on a given bandwidth. The frequential and temporal accuracy of the algorithm and the associated hardware define the ability of FH detection. The literature is not very extensive in the specific area of FH interception, but a similar problem exists in Cognitive Radio (CR).

A CR [Mit02] is a radio that can be dynamically configured and programmed. Allowing to select the best wireless channels in its vicinity to avoid interference or congestion due to

another user. Therefore, spectrum sensing is a critical issue for cognitive radio. Indeed, it is an effective way to detect spectrum holes in radio network [Bag11]. The spectrum scanning processing can be characterized in two groups, depending on how many separate channels can be detected at a time:

— Narrowband detection: this detection can only detect one channel in the whole sampled band. This represents the first detection method, and its limitation is due to the limited bandwidth capacity of radio. Therefore, if you want to detect N distinct channels at the same time, a wideband detector has to be used.

— Wideband detection: this detection is more modern and has been introduced thanks to the emergence of SDR that can acquire and process a large bandwidth and is easy to program compared to hardware radios (see chapter 5). It allows covering several bands by using a single receiver. It combines split-signal processing and narrowband detectors.

2.1 Narrowband detection processing

The purpose of narrowband detectors is to determine whether or not the band is occupied by a signal. The whole received band is therefore considered as a single channel. We assume the statistical test $\mathcal{H}_0$: the channel is not used, i.e., only noise is received and $\mathcal{H}_1$: the channel is used, i.e., a signal has been detected.

The model of the received band under both assumptions $\mathcal{H}_0$ and $\mathcal{H}_1$, can be expressed as:

$$\begin{cases} \mathcal{H}_0 : d[k] = w[k], \\ \mathcal{H}_1 : d[k] = x[k] + w[k], \end{cases} \tag{3-5}$$

where $d[k]$ represents the received signal, $x[k]$ is the transmitted signal, $w[k]$ represents an Additive White Gaussian Noise (AWGN) of variance σ_k^2, and k denotes the sampling index at channel rate F_s.

The sampling is assumed to be ideal such that $x[k] = x(kT_b)$ with T_b the sampling time (inverse of the narrowband channel bandwidth).

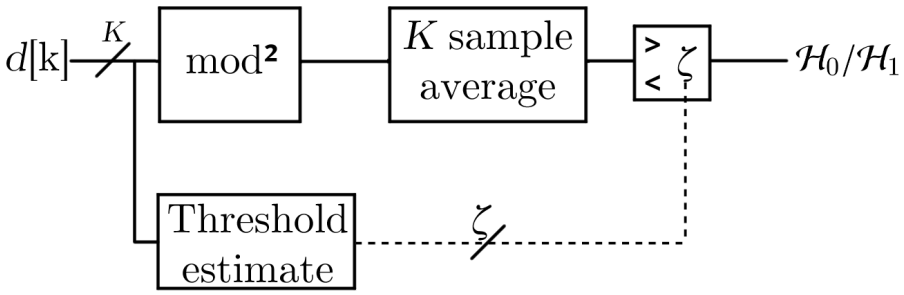


Figure 3-2 – Energy based detection algorithm.

The probability processes that will be described below define the probability of detection (P_d) and false alarm (P_f). These two probabilities can be defined as:

$$P_d = P(\mathcal{H}_1 \mid \mathcal{H}_1), \tag{3-6}$$

and:

$$P_f = P(\mathcal{H}_1 \mid \mathcal{H}_0). \tag{3-7}$$

2.1-1 Energy based detection

The energy detector is based on a very straightforward principle (a schematic sketch can be seen in Fig. 3-2), for each K samples batch the received energy υ is estimated and compared to a threshold ζ. This threshold is set in such a way that if the energy is lower there is no signal and higher a signal is received. Energy detection [Urk67] computes the averaged energy with

$$\upsilon[k] = \frac{1}{K} \sum_{h=1}^{K} |d[k-h]|^2, \tag{3-8}$$

with K the averaging factor and sample processing length, while $d[k]$ denotes the kth received sample.

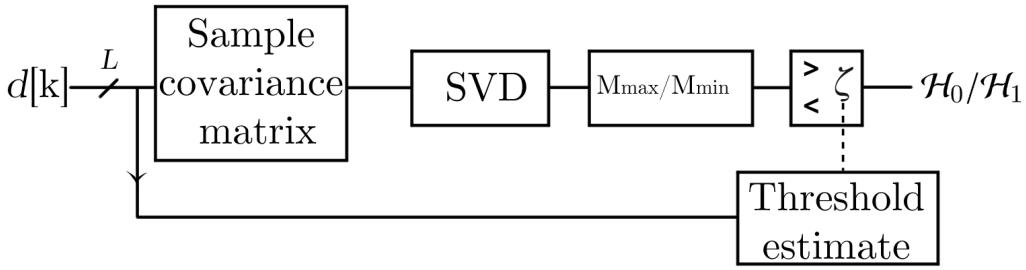


Figure 3-3 – Covariance based detection algorithm.

If the energy is higher than the threshold ζ then $\mathcal{H}_1$ is assumed. The threshold is dependent of the noise variance [Zen08], but the performance can be greatly improved in the case where the noise is not stationary by using a dynamic threshold. In [Zhu08], this kind of approach is addressed by using a double thresholding. It adds an unsure area, corresponding to the area around a static threshold to deal with uncertainty. If the energy of the samples is smaller than the area limit, then the band is unoccupied, but if the energy of samples is higher than the second limit, then the band is occupied. If the energy is inside the area, then the decision is not taken by the process but by a supervisor who analyzes the results of another detector monitoring the same band at another location, exploiting the spatial diversity. While this dual-threshold algorithm reduces P_f, the P_d is higher than with techniques that have one threshold only.

[Mur15] shows an adaptive threshold detection method based on an image binarization technique which takes in account the previous decisions and a linear function of the received signal mean and standard deviation.

Energy detection is a relatively low-complexity technique that does not require any prior knowledge of the signal characteristics. However, it uses only an energy level-based criterion and therefore cannot distinguish between a signal and a high noise level, which makes it subject to high uncertainty. It has a low detection performance at low SNR [Tan05]. In addition, energy detectors alone do not work efficiently for detecting spread spectrum signals [Cab04].

2.1-2 Covariance based detection

The covariance-based detection techniques compute the covariance matrix of the received signal and apply Singular Value Decomposition (SVD) to detect the presence of signals [Ber00]. The SVD is used to study the structure of the covariance matrix of the received signal. Because

the matrix values depend on whether a signal or noise is processed, a telecom signal can therefore be differentiated from noise (assumed to be white and Gaussian) because it is correlated. A schematic sketch can be seen in Fig. 3-3.

The auto-correlation samples of the received signal are expressed as:

$$r_l[k] = \frac{1}{A}\sum_{a=0}^{A-1} d[k+a]d^*[k+a-l], l = 0, 1, ..., L-1, \tag{3-9}$$

where d is the received samples, L the number of samples used and A the averaging factor.

The statistical covariance matrix (LxL) is then expressed as:

$$\hat{R}_d[k] = \begin{bmatrix} r_0[k] & r_1[k] & \dots & r_{L-1}[k] \\ r_1^*[k] & r_0[k] & \dots & r_{L-2}[k] \\ \vdots & \vdots & \ddots & \vdots \\ r_{L-1}^*[k] & r_{L-1}^*[k] & \dots & r_0[k] \end{bmatrix} \tag{3-10}$$

Then, SVD is applied on the matrix $\hat{R}_d$, in order to have the maximum M_{max} and minimum M_{min} eigenvalues of $\hat{R}_d$. Finally, the sensing decision is made by comparing the ratio of maximum to the minimum eigenvalue: $\frac{M_{max}}{M_{min}}$ to a threshold [Zay09].

To decrease the computation complexity of such a method, the authors of [Kum13] implemented a sample covariance matrix instead of a statistical covariance matrix.

2.1-3 Cyclostationary features based detection

The detectors based on the cyclostationary features use the periodicity in the signal or in its statistics (e.g. mean or autocorrelation) [Gar91], a schematic sketch can be seen in Fig. 3-4. A signal with nothing but noise is Wide-Sense Stationary (WSS) meaning with no correlation, while a modulated signal is cyclostationary, meaning with spectral correlation due to the redundancy of signal periodicities [Yaw16].

The received signal $d[k]$ is called cyclostationary if the mean and the autocorrelation of the signal are periodic:

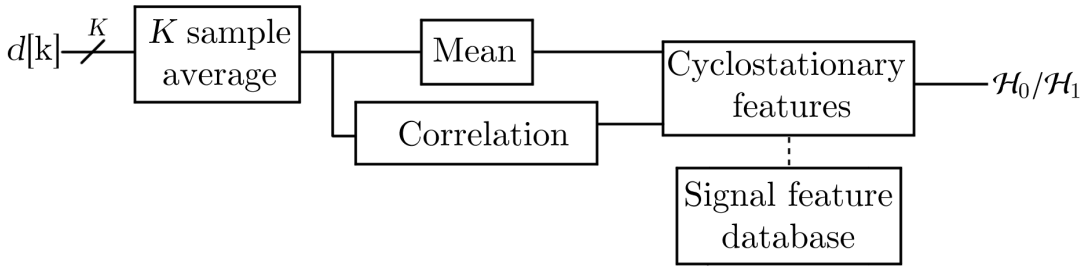


Figure 3-4 – Cyclostationary features based detection algorithm.

$$m_d[k] = m_d[k + K_0] = E[y[k]], \tag{3-11}$$

and:

$$R_d[k, \tau] = R_d[k + K_0, \tau], \tag{3-12}$$

where K_0 is the period of the signal $d[k]$, E denotes the expectation operator, R_d autocorrelation function as:

$$R_d[k, \tau] = E\left[d(k + \tau)d^*(k - \tau)e^{j2\pi\alpha k}\right], \tag{3-13}$$

and α denotes a cyclic frequency assumed to be known [One04].

The performance at low SNR can be improved by using correlation-based Euclidean distance as proposed in [Dam15].

Cyclostationary feature detection can detect signals with low SNR and above all to distinguish between a signal and noise. It needs some prior knowledge of the signal (*Signal feature database* of Fig. 3-4), such as symbol rates, modulation parameters. Moreover, it is usable for a single type of signal only, for another signal it will be necessary to readapt the parameters.

As for energy detection, cyclostationary detection performance can be further enhanced by increasing the number of averaged samples which reduces the capacity to detect short FH bursts. Moreover, it is at the cost of an increase in processing time and complexity as the

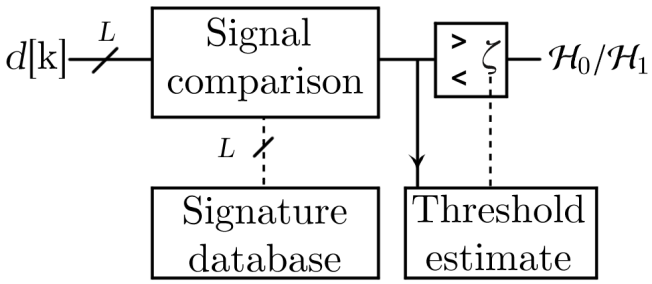


Figure 3-5 – Signature based detection algorithm.

number of signals to process increases. A trade-off between low SNR performance and small time slot detection must be taken for FH signal detection.

2.1-4 Signature based detection

The signature feature algorithm, also called matched filter [Zha14a] is even more dependent on *a priori* knowledge of the signal being listened to than the former algorithm. It compares the received signal with a signature, this signature is a part of the received signal, known in advance, generally, the preamble, pilot or synchronization samples of a network stack. These samples are compared to the received samples using a correlation operation (3-14), which are then compared to a threshold (ζ) to determine the sensing decision. The threshold is dependent on the noise level of the received signal [Lv15], a schematic sketch can be seen in Fig. 3-5.

$$\zeta = \frac{1}{L}\sum_{k=1}^{L} d[k]x_p^*[k], \tag{3-14}$$

with d the received vector samples of length L, x_p are the signature samples and L the number of samples used.

The use of a static threshold can lead to less accurate results in low SNR situation, as for the previously mentioned techniques a dynamically selected threshold can enhance the detection performance [Sal15].

As for the cyclostationary (Section 3-2.1-3), a signature detector can only work for one type of network standard. To overcome this issue, the authors of [Gho11] have designed a reconfigurable signature database, which can adapt to several different protocols. Hence, it requires perfect knowledge of the potential incoming radio protocols such as bandwidth, carrier

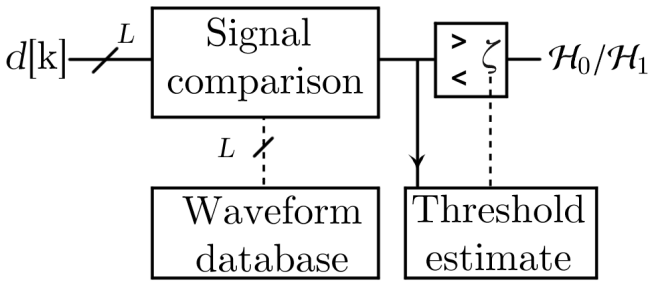


Figure 3-6 – Waveform based detection algorithm.

frequency, modulation type, pulse shaping, and frame formatting. The complexity increases significantly when more protocols are supported [Cab04]. Another disadvantage of signature detector is their poor energy efficiency. If only the signature part of the received signal must be compared, computations will however be made on the whole received signal, inducing an energy waste.

It should be noted that this algorithm works only if a known signature is present which, for secure signals, is not automatically the case.

2.1-5 Waveform based detection

This method is similar to signature detection but uses the signal waveform as a signature instead of a part of the signal, which basically enables it to be used on signals that do not have a preamble or synchronization signals, as illustrated by Fig. 3-6. Obviously, this method is only applicable to systems with known signal waveform, and is also called coherent sensing. The detection of a busy band is performed by comparing the received signal to the known waveform (3-15) and then a threshold is used to take a decision.

$$E_w = \Re\left(\sum_{k=1}^{L} d[k]{W_f}^*[k]\right), \tag{3-15}$$

where d is the received samples vector of length L, W_f the waveform model.

The waveform-based detection offers better performance than energy detectors and outperforms its convergence time which is an important point to detect fast FH signals [Tan05]. Furthermore, the P_d increases as the time slots are larger, but synchronization errors can

Algorithm	Advantages	Drawbacks	References
Energy detection	• Easy to implement • Low complexity • No *a priori* knowledge	• Poor performance at low SNR • Sensitive to noise uncertainty • No distinction between noise and signal	[Urk67] [Zen08] [Zhu08][Mur15] [Tan05][Cab04]
Covariance detection	• No *a priori* knowledge	• Mathematically intensive	[Ber00] [Zay09] [Kum13]
Cyclostationary detection	• Good at low SNR	• Requires *a priori* knowledge • Slow convergence time • Highly complex • Bad performance against fading • Bad performance against unsync	[Gar91] [Yaw16] [One04] [Dam15] [Tka07] [Sut07]
Signature detection	• Good at low SNR • Fast convergence time	• Requires extensive *a priori* knowledge • *A priori* knowledge is not always available • Highly complex	[Zha14a] [Lv15] [Sal15] [Gho11] [Cab04]
Waveform detection	• Good at low SNR • Fast convergence time	• Requires extensive *a priori* knowledge • Bad performance against unsync	[Tan05] [Cab06] [Mis07]

Table 3-1 – Advantages and drawbacks of narrowband detection methods.

cause a huge loss in performance [Cab06]. Such an algorithm is currently used in Worldwide Inter-operability for Microwave Access (WiMAX) signals for detecting uplink packet preambles [Mis07].

2.1-6 Comparison of the narrowband detection algorithms

Table 3-1 compares the aforementioned narrowband detection algorithms and the Fig. 3-7 compare their complexity in accordance with their accuracy.

The energy detection does not require any *a priori* knowledge about the signal, which makes it simple and versatile because it can work on any signal. Its drawbacks are inability to distinguish signals from noise, and its poor performance in noisy conditions.

Improved performance can be achieved by using covariance-based approaches, which share the qualities of the energy technique except that the computational complexity is much higher. Indeed, computations of the sample covariance matrix and its SVD are computationally intensive, and make them less suitable for real-time applications.

Cyclostationary detection can achieve good performance against noise, and can distinguish a signal from the noise, but at the expense of a low convergence speed, an increased complexity for long signals, vulnerability to *time unsync* [Tka07] and channel fading [Sut07] and most importantly, the need to have some knowledge of the signal that it is able to identify. This

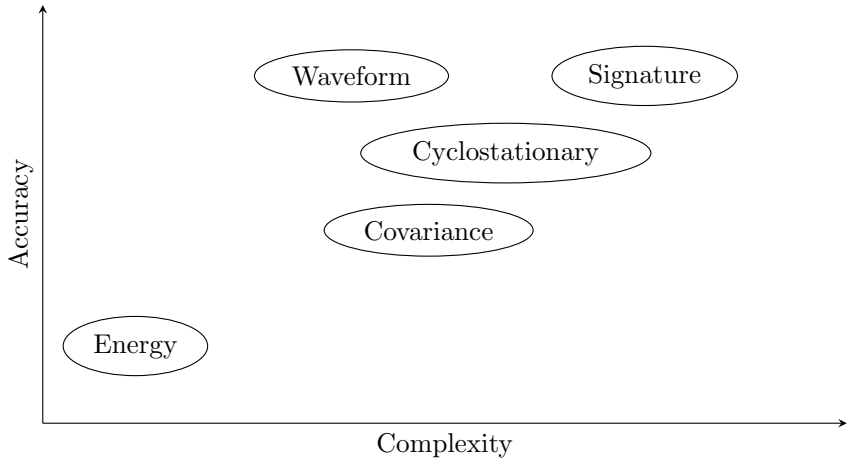


Figure 3-7 – Comparison of narrowband detection methods in terms of their sensing accuracies and complexities.

makes it *de facto* non-universal.

Signature detector needs only a few samples to achieve a good detection performance, even in low SNR condition. Its main drawback, however, is that *a priori* knowledge of the signal is required, and based on the signal received, this knowledge can simply not exist making it unfit for use.

Waveform-based detection is the most accurate method because of the coherent processing [Tan05]. It is fast, offers good performance against noise and is not so complex but like the others, knowledge of the signals is necessary.

In all cases, the use of a fixed threshold leads to poor performance when the noise is not stationary and its variance is unknown [Cab06].

For the acquisition of FH signals, our use case imposes the signals being unknown, therefore the use of methods requiring *a priori* knowledge on the signals is not an option, and a fast processing is also necessary, which means only the energy detection is a suitable choice. Among all the studied detectors, it is the one that offers the lowest performance, which is quite normal because unlike the others, it is not optimized to detect specific signals, but is more versatile. Our use case deals with the presence of multiple channels and not just one as proposed by narrow band detectors.

2.2 Wideband detection processing

A narrowband detector cannot be applied to a signal consisting of several channels. As a matter of fact, narrowband detectors make a single binary decision for the whole spectrum and therefore is not able to differentiate between channels that lie within the wideband spectrum. The evolution of radio protocols and radio hardware allows a greater availability of high communication rates over a large bandwidth. Therefore, it is more feasible to monitor a wide band of the radio spectrum with a single piece of equipment instead of an aggregation as required by narrowband detector. The majority of wideband detectors consists of dividing the sensed spectrum into several sub-bands and performing narrowband detection on each of them.

Wideband detectors (excluding the narrow band included inside) do not need to have any knowledge of the characteristics signal intercepted, but they all need to know the number of channels they can pick up. If this number is wrongly considered or if the channels are not correctly aligned, overlapping phenomena may occur, thus decreasing the detection performance.

In order to detect an FH signal efficiently over a wide band, a wideband detector is needed, some of the detectors presented hereafter will be used in thesis in order to perform benchmarks of FH detectors (see chapter 4). However, will only be selected those suitable for the constraints of our context, namely a low complexity (in order to be real-time) and a detection very close to blind detection of the FH signals (requiring the minimal information on the signals to be intercepted). The selected methods are those described in Section 3-2.2-2 to Section 3-2.2-4 namely: Multi-band joint detection, Filter bank detection and Wavelet detection.

It must be noted that the detectors which will be presented hereafter will comply with the Nyquist rate sampling, and the particular case of sub Nyquist sampling will be discussed in Section 3-2.2-6.

2.2-1 Sweep tune detection

One naive approach easily applicable is to *sweep* across a frequency range of interest where we want to find an FH burst. This method can be seen as consecutive windows obtained by changing the central frequency of the receiver. This method can easily be used in conjunction with a superheterodyne receiver, which does not require a large bandwidth. It is for instance

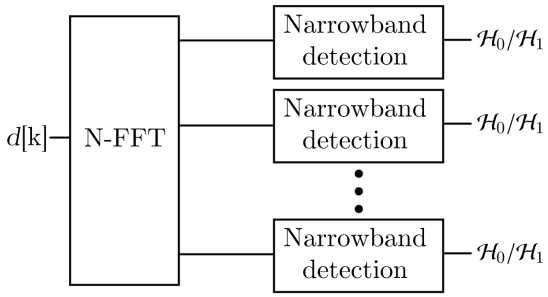


Figure 3-8 – Multi-band joint based detection algorithm.

used in spectrum analyzers as [roh], certified for TEMPEST security evaluation. However, such sequential-sensing approaches are ineffective for fast FH signals because they require longer times to sweep across the spectrum. Since they do not sample their whole detection band (unlike the method described hereafter), signal can be emitted just outside their current sampling windows.

2.2-2 Multi-band joint detection

Multi-band joint detection is after the sweeping detection the easiest and simplest form of detection and is described in Fig. 3-8. The subband splitting and decimation is done in one step with an FFT [Qua09], which requires a number of channels equal to a power of 2 for a low complexity. Non-aligned FFT to the power of 2 are possible as long as all the prime divisors of the FFT length are small but the complexity of $\frac{N}{2}log_2(N_{FFT})$ is not maintained.

The implementation of this method is detailed into the dedicated Section 3-3 as this is the method our real-time interception system is based on.

2.2-3 Filter bank detection

The filter detection (Fig. 3-9) uses a filter bank in order to separate the wideband signal into a fixed number of narrower bands [F-B08]. The filter bank is made by shifting a low-pass prototype filter as many times as the total number of channels. The bank is then composed of filters with different shifted central frequencies thus creating a PolyPhase Network (PPN). In each narrower band, the corresponding part of the wideband signal is down-converted to baseband, then low-pass filtered and finally sent to an Inverse Discrete Fourier Transform (IDFT). Each signal is then provided to narrowband detectors which operate at low sampling rate (i.e., at baseband signal rate F_b and not at wideband rate F_s).

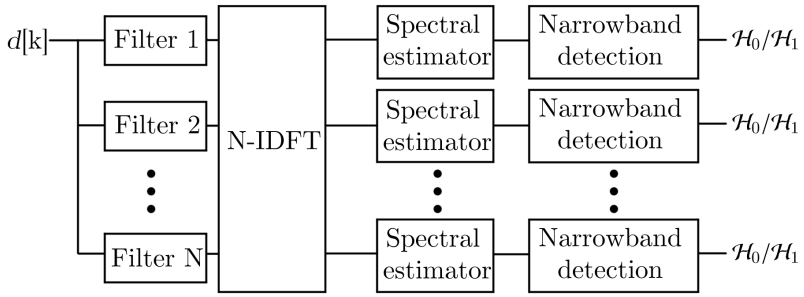


Figure 3-9 – Filter bank based detection algorithm.

The version that was used for our benchmarks use instead of a FFT a Hopping Discrete Fourier Transform (HDFT) as used in [Al-21] in their Filter Bank Multi-Carrier for Frequency Spreading (FBMC-FS) receivers. It performs the filtering in the oversampled frequency domain by defining the analysis filter in the frequency domain on a very limited number of points which improves computation speed, and a Phydyas filtering [Dor17] was used to reduce noise. The functional sketch then becomes Fig. 3-10, with K_{PPN} the temporal overlapping (linked to noise reduction), and W the length of the Phydyas filtering (impacts on the detectability of small hops). Fig. 3-11 further illustrates the despreading filter step with its weighted sliding filtering. Moreover the narrow band detectors are not independent but joint together to compare the values of the several bands to select the most likely containing an FH signal, thus reducing the impact of the thresholding choice which is then only static.

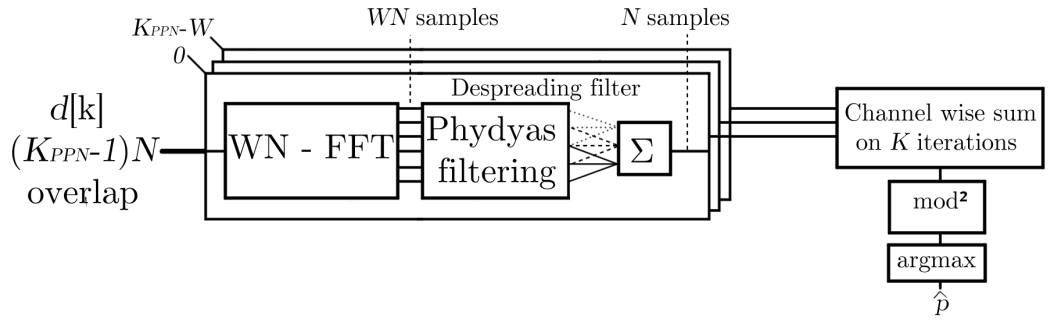


Figure 3-10 – HDFT filter bank based detection algorithm.

2.2-4 Wavelet detection

The basic principle of using wavelets is based on subband separation using a Power Spectral Density (PSD), a wavelet transform is then applied to detect an occupied subband as shown in Fig. 3-12.

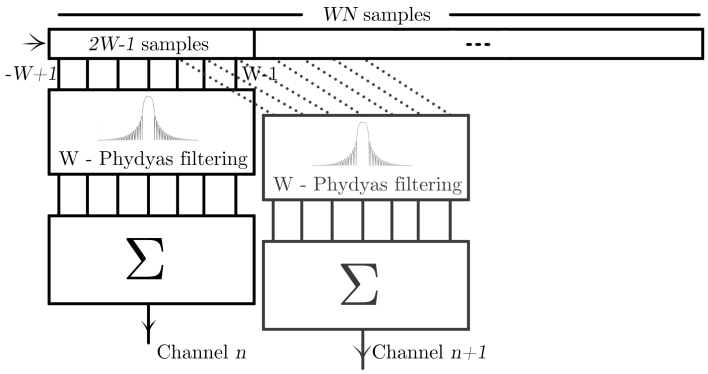


Figure 3-11 – Focus on despreading filter.

Figure 3-12 – Wavelet based detection algorithm.

The wavelet can be used in two different manners, namely as an AWGN signal detector or an edge detector. The first method is based on the assumption that a wavelet transform applied to a AWGN signal will greatly reduce its magnitude, whereas applied to a non-pure AWGN signal it will reduce less its magnitude. This method is depicted in Fig. 3-13, with K the temporal overlapping. A discrete wavelet transform [Zha14b] and a Savitzky–Golay filtering [Cap16] have been added to improve respectively computationally and noise resistance. We can see that for each channel, an energy estimation is performed before and after the transformation, and the $\mathcal{H}_0$ or $\mathcal{H}_1$ decision is given by observing the energy difference. This method has been used throughout the different benchmarks of Chapter 4.

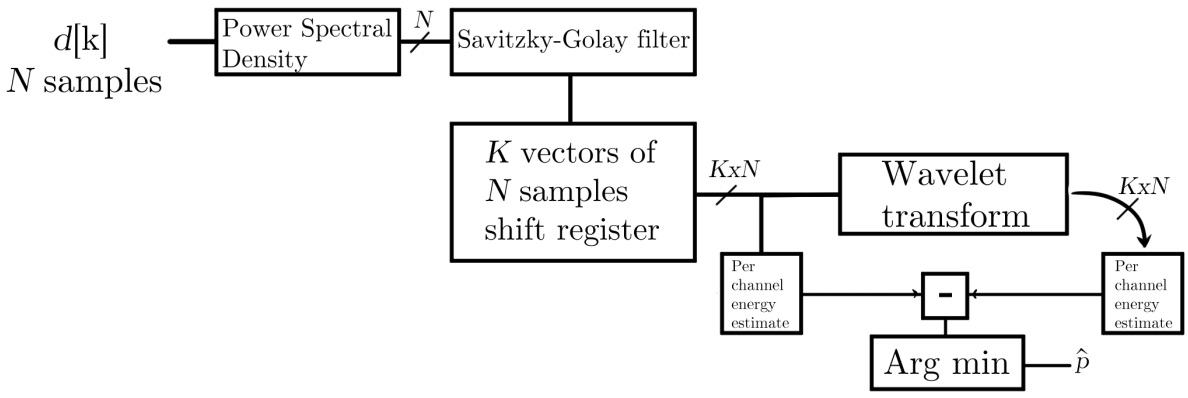


Figure 3-13 – Wavelet based detection algorithm based on [Zha14b] [Cap16].

The other use of the wavelet is an edge detection algorithm, i.e., using wavelets to detect sharp variations in an image, where the related image corresponds to the time-frequency representation from successive PSD. A sudden change in the PSD indicates either an impulse noise or a received radio burst, thus indicating that an FH channel is being used [Kum16]. Detecting the edges allows characterizing the time and frequency position of an FH burst. However, this kind of technique requires a lot of computation, much more than the previous presented method, making it impractical for real-time use.

2.2-5 Machine learning detection

Machine Learning (ML) has an interest that grows year after year as the possibilities offered are so numerous and is becoming more and more popular in signal processing. Their aim is to identify the presence of signals in a channel by formulating the process as a classification problem in which the classifier can be supervised [Wan16a] [Din13] [Nie17] [Mik14] [Par07] or unsupervised [Cla11] [Che17] [Li16] [Bka11] [Li10], and has to decide between $\mathcal{H}_0$ or $\mathcal{H}_1$ for each frequency channel.

Supervised ML build a mathematical model from a set of data called training set, which contains both the inputs and the desired outputs associated with each input set. In this type of learning, classes are defined from the start. Several learning classifiers can be used as K-nearest neighbors [Mik14] , Support Vector Machine (SVM) [Din13] [Mik14], Bayesian [Nie17] [Mik14], random forest decision tree [Wan16a] [Mik14] or hidden Markov sequences [Par07].

Unsupervised ML has no *a priori* data of the classes and will build them by itself as the entries are provided. Non-parametric approaches can be used like K-means algorithm [Cla11], self-organizing maps [Che17] or naïve Bayes [Li16]. However, other learning methods can be used such as game theory-based learning [Li10] or reinforcement learning [Bka11].

The subject of ML is extremely vast, and the applicable methods are so numerous that it could be the subject of a dedicated chapter. However, these methods have in common a high complexity and the need to have a strong computational power at disposal. Moreover, it is necessary to have representative signal data in order to have an accurate training which is impossible if we try to intercept unknown FH signals.

2.2-6 Sub Nyquist rate detection

Albeit not being the sampling technique used in this thesis, several detectors use a signal sampled at a rate lower than that required by the Shannon-Nyquist theorem. These detectors use compressed sensing (or sparse sampling) to sample a signal faster than the Nyquist rate. Through kernel optimization, the sparsity of a signal can be exploited to reconstruct it from fewer samples than required by the Nyquist–Shannon sampling theorem. This recovery is carried out by multiplying the input samples by a specific kernel, which if correctly chosen allows an error-free reconstruction in terms of Mean Square Error (MSE), but this is done under two requirements [Don06]:

- The input samples must be sparse (the spectrum is not filled), this is usually the case when there is one FH signal in the spectrum being monitored, but if multiple signals and interferers are present then this condition is not necessarily met.
- The kernel must satisfy restricted isometry property or have a small mutual coherence to guarantee recovery with the smallest error [Don06].

Although this method allows to reduce the number of samples to be processed, thus making the calculations faster (because it is possible to reduce the bandwidth needed to observe a signal), several major disadvantages exist. First, the sparsity level of wideband signal varies over time, the kernel must be corrected accordingly and so the number of measurements to make. This is a problem for an optimized architecture because it requires either an adaptative design or to pessimistically choose the number of measurements. It also requires to have *a priori* knowledge about the signals contained in the spectrum. Since the sampling frequency is lower, an error will have an impact on more samples after the recovery, which ultimately reduces the performances at low SNR. Finally, the performance is sensitive to design imperfections.

3 Multi-band joint based detection algorithms for fast FH

Our objective is the continuous observation, detection and extraction of FH signals over a wide band (> 100 MHz) and the analysis of the presence of an internal side-channel. The final goal is to have a portable system, which would be placed near a Device Under Test (DUT) or in a sensitive area. Then, the complete (or only a part of) spectrum will be analyzed and any

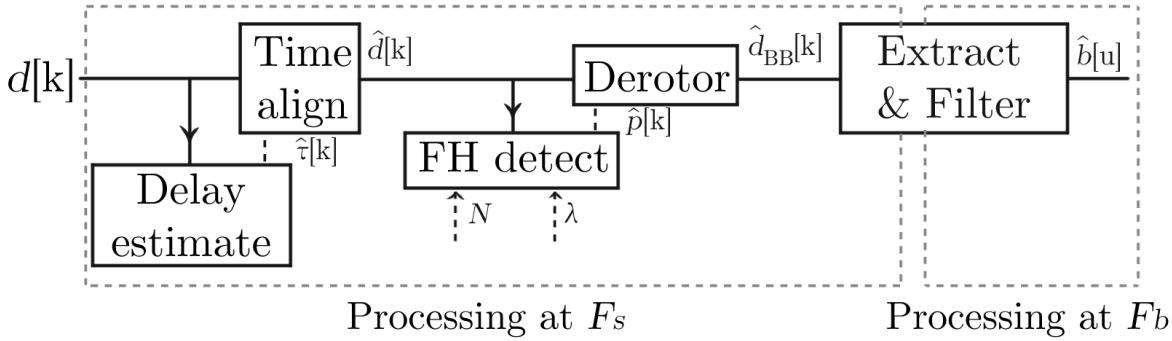


Figure 3-14 – Initial DFT multi-band joint based detection algorithm.

abnormal FH signals (not authorized or showing a side-channel) will be reported. Due to the bandwidth and time constraints (the detection of an abnormal signal must be reported quickly and not after several hours of calculations), a real-time detector must be used. The algorithm must therefore be fast and not too complex to be used leveraging hardware accelerators. The nature and parameters of the FH signal being unknown, the method to be used should not resort to *a priori* data.

An exception is made on the number of channels which is considered as known, but it is rather to define the number of detectable frequency bands. Several FH channels can be present within the same detection band, as long as only one FH signal is active at a time within the same detection band, the detection and extraction of red signals will not be impacted.

For all these reasons, a multi-band joint detection algorithm with an energy detector has been chosen as our detection algorithm. In particular, two versions have been developed based on the same principle but with 2 different processing hardware (General Purpose Processor (GPP) and FPGA/GPP) which will be developed in the next sections.

3.1 Algorithm flowgraph

Our algorithm, in addition to detecting frequency hopping channels, must also extract the information within the identified channel. Its operating principle is composed of standard telecommunication operations (linear flowgraph) as seen in Fig. 3-14, and is composed of three parts:

— A time synchronization part, its purpose is to realign the time-frequency grid in time, in order to improve the performance of the detector (because of averaging, the performance

decreases with desynchronization). This step will be more developed in 3-3.2. The output of this block is a buffer $\hat{d}$ aligned in the T/F grid on which is applied the channel detector.

— An FH channel detector part estimates which channel is used. The detector is based on the instantaneous periodogram, averaged by the repetition factor λ ($\lambda = \frac{T_s}{N}$). This can be efficiently implemented by means of Fast Fourier Transform (FFT) if N is a power of 2. Considering the input $\hat{d}[k]$,

the associated estimated channel index $\hat{p}$ is expressed as:

$$\hat{p} = \underset{p\in[0;N-1]}{\arg\max} \left\{ \sum_{l=0}^{\lambda-1} \left| \sum_{m=0}^{N-1} \hat{d}[lN+m] e^{\frac{-j2\pi mp}{N}} \right|^2 \right\}. \tag{3-16}$$

— A channel extraction part, given the chosen index (linked to the central frequency by $\hat{f}_p = \frac{\hat{p}F_s}{N}$) is used in the derotor that shifts the modulated channel into baseband:

$$\hat{d}_{\mathrm{BB}}[k] = \hat{d}[k] e^{\frac{-j2\pi k \hat{f}_p}{F_s}}. \tag{3-17}$$

The filtering stage lowers the rate of the detected signal from F_s to F_b. If the correct FH channel has been chosen, and assuming an ideal filtering operation, the resulting signal is baseband message b.

The number of detectable channels is defined by N, which also defines the bandwidth assigned to each channel by F_s/N, the λ averaging allows improving the performance at low SNR, however, this is done at the expense of short FH burst detection. Compared to the standard definition of a multi-band joint detection, the energy detectors placed at the output of each channel are not independent but linked and there can be only one channel detected at a time.

A joint detector decreases the false alarm P_f, since in our use case, the detection system is either placed next to a device to be tested, or placed in a secure area where no radio signals should be allowed. The highest energy received is assumed to be from the listened device or an illegal signal.

3.2 Time align

The purpose of this section is to describe the synchronization method. As shown in Fig. 3-14, this algorithm is placed upstream the FH detector and is only used at the beginning of the interception process as the synchronization delay τ is assumed to remain constant. It takes the complex baseband samples as inputs and provides the estimation delay $\hat{\tau}$. To achieve a fast synchronization algorithm (running at F_s), a limited complexity is required and, thus, methods using maximum likelihood [Ko05], wavelet transform [Sir10] or machine learning [Zha16] cannot be used. The proposed algorithm relies on Fourier transform to detect the signal hop with a high time resolution.

3.2-1 Synchronization overview

To keep a simple reception architecture one can want to use the FFT [Sor88] as used by the FH detector. Its size is equal to the number of channels N (with eventually zero padding to achieve a required power-of-2). However, the time accuracy is quite low as the resolution is directly linked to the FFT size, and the detector will not be able to detect a hop with accuracy smaller than N samples (in the best case, without noise). This time resolution can be problematic when the burst duration is very short (i.e., small values of λ are common in TRANSEC scenario). In order to increase the time accuracy, the proposed approach relies on a sliding Discrete Fourier Transform (sDFT). It computes a new spectral transform for each incoming sample as expressed in (3-18). To lower the complexity overhead of the sliding approach, a sDFT is used instead of a sliding FFT (sFFT) as it reuses the previous outputs (while a sFFT recalculates the full transform for each sample) in a recursive manner. Moreover, the recursive calculation does not require N to be a power-of-2 as with FFT radix-based implementation.

$$DFT[k+1] = e^{2j\pi k/N}\Big[DFT[k] - d[k-N] + d[k+1]\Big], \tag{3-18}$$

with d the received samples and N the DFT width.

From the raw output of the sDFT, additional post-processing is required to extract a delay estimation. The full algorithm is described in the next part. The synchronization step has a higher complexity than the FH detector. However, the detector has to be used for the whole interception process (in a real-time manner) while the delay estimation block only has to be

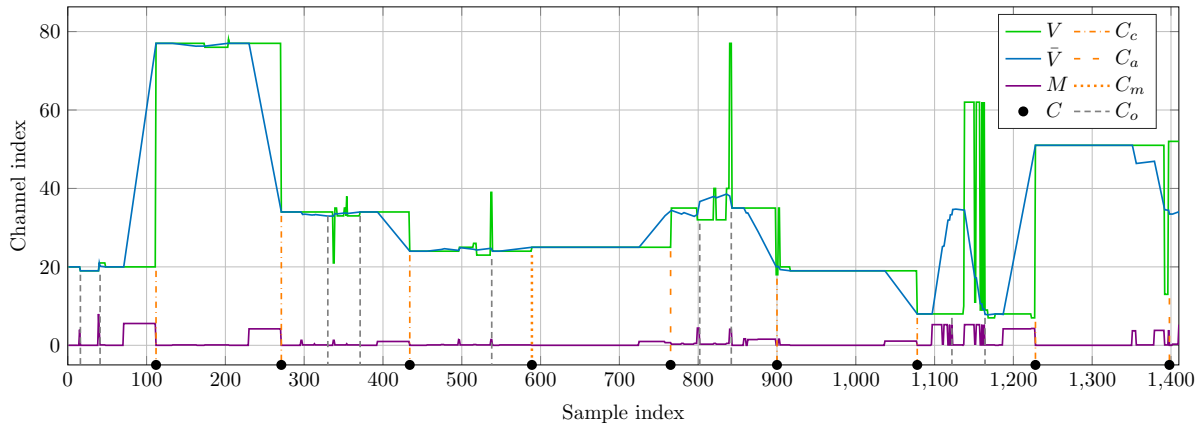


Figure 3-15 – Chronogram of the various delay estimation steps.

used once, at the start of the interception.

3.2-2 Algorithm description

Algorithm 1: Delay estimation

—— Evaluate the bin index of current channel used for each sample
for $k = [1 : \lambda N \times L]$ **do**
  $S \leftarrow |sDFT(d[k])|^2$
  $V[s] \leftarrow \arg\max(S)$
end for
$\bar{V}[k] = \frac{1}{\alpha} \times \sum_{l=0}^{\alpha-1} V[k+l]$
—— Evaluate the shift in the bin index, generate the associated pool and fix detected outliers
$M[k] = |\bar{V}[k] - \bar{V}[k-1]|$
$C_0 \leftarrow$ Find each end of non-zero area in M;
C = Correct outliers (C_0);
—— Evaluate synchronization time
$C \leftarrow$ sort(C, ascend);
$\hat{\mathcal{T}} \leftarrow$ linear regression(C);

Based on the expression of the received signal in (3-4), the purpose is to find $\hat{\tau}$. As the received signal $d[k]$ is sampled at F_s (with k the sample index), we want to find the sample delay index $\mathcal{T}$ defined by:

$$\mathcal{T} = \lfloor \tau \times F_s \rfloor . \tag{3-19}$$

The delay estimation step will process $L \times \lambda N$ samples of the signal $d[k]$ where L is the number of slots used for the synchronization. Estimating the synchronization delay $\hat{\tau}$ (or a synchronization index $\hat{\mathcal{T}} = \lfloor \hat{\tau} \times F_s \rfloor$) can be done by detecting each hop. Indeed, a hop occurs when k verifies:

$$k = l \times (\lambda N) + \mathcal{T}, \ \forall l \in [0; L-1], \tag{3-20}$$

and $d[k]$ is expected to face at most L hops. The proposed algorithm estimates the used channel with a high time accuracy and deduces the synchronization delay $\hat{\tau}$. This value will then be used by the time alignment block.

The algorithm is sequential and is divided into three main stages described in Algorithm 1. The first stage intends to extract the used channel for each incoming sample. The second stage estimates the hop locations and removes noise artifacts. Finally, the last stage gathers all the hop indexes and estimates the fine time delay estimation $\hat{\mathcal{T}}$. An illustrative chronogram of the different algorithm variables is depicted in Fig. 3-15. In this simulation, the parameters are $N = 80$ and $\lambda = 2$ (leading to a hop duration of $\lambda N = 160$ samples) and a SNR of 0 dB. The sample delay index is set to $k = 110$ and $L = 9$ bursts are used for the delay estimation.

As discussed in Section 3-3.2, the first step applies the sDFT and associates each sample with the channel with maximum energy in $V[k]$ (i.e., for each sample, the position of the maximum of the sDFT) depicted in green in Fig. 3-15. As this row estimation can strongly be affected by the additive noise, $\bar{V}[k]$, the sliding average of $V[k]$ with a depth α, is computed (blue curve). The alpha parameter can be adjusted according to the impulse noise contained in the signal, a large alpha will decrease the noise but will increase the convergence time and therefore decrease the performance. However, too much noise will also decrease performance. A trade-off must be done according to the received signal (in our case $\alpha = \frac{\lambda N}{4}$).

A hop is defined as a variation in the channel index and can be primarily obtained from the instantaneous derivative $M[k] = |\bar{V}[k] - \bar{V}[k-1]|$ (violet curve). If the channel index remains constant, the derivative is equal to zero. Thus, a hop can be defined as the end of a non-zero area of $M[k]$. It should be pointed out that we have areas (several consecutive terms of $M[k]$ different from zero) instead of peaks as the derivative is computed from $\bar{V}$, the sliding average of V. The obtained hop locations are then gathered in a pool C_0. There are, however, some cases where hops are not detected properly, and some additional processing (outliers correction) has to be done.

The next step is thus to correct outliers data to transform the initial pool C_0 into the new

pool C. The basic principle of this algorithm is based on the regular occurrence of hops (the time slots are all the same size). The algorithm will try to find the most probable hop grid, to do so it performs several successive analysis of the signal to deal with the different particular cases. Data contained in the pool can be of two types, correct C_c or outliers. Moreover, there can be three types of outliers: i) C_a denotes the presence of too close hops (with respect to the expected duration expressed in (3-20)). These hops will then be aggregated to form a single hop in the pool (orange dashed lines). ii) C_m denotes the absence of a hop which is either due to the successive use of two nearby channels drown in noise or to the re-use of the same channel. In this case, the missing end hop is generated in the pool (orange dotted lines). iii) C_o denotes the presence of a hop out of the hop grid: this can be due to impulsive noise or reception of a third-party signal. In both cases these hops will be discarded from the pool (black dashed lines).

Fig. 3-15 clearly shows that both the sliding average (transition from green to blue curves which removes most unwanted artifacts as for instance around sample 350) and the outlier correction improve the detection. The outliers C_a and C_o have been successfully detected and extracted and the missing hops C_m have been generated. At the end, the hop position can be extracted in the pool $C = C_c \cup C_a \cup C_m$ (black circles).

The final step consists in arranging the hop positions in C in chronological (i.e., ascend) order. As the elements in C should follow (3-20), a linear regression is carried out to obtain the delay estimation $\hat{\mathcal{T}}$ as the Y-intercept. Finally, this value is used by the time alignment block to obtain the signal used in the FH detector in (3-16) with:

$$\hat{d}[k] = d[k - \hat{\mathcal{T}}]. \tag{3-21}$$

The performance of the synchronization will be evaluated in Chapter 4 Section 4-2.1.

3.3 Algorithm refinements

The algorithm presented here above is functional but it has room for improvement. In fact, it follows a classical flowgraph sequence composed of independent blocks performing a specific function: channel detector, derotor, filtering, delay estimation, etc. The first improvement is based on joint estimation and detection and is reflected in the DFT-Bin Extraction (DFT-BE) algorithm which merges some blocks in order to gain in speed (Section 3-3.3-1), then the FFT-Bin Extraction method (FFT-BE) will be presented (Section 3-3.3-2). The latter one is not

an improvement of the DFT-BE version but rather the adaptation of our basic algorithm to the FPGA programming paradigm. In the following, the various simulations and experiments will be performed using these two versions and not the original proposition.

3.3-1 DFT-Bin Extraction method

The DFT method (which estimates the FH channel) requires that some processing is carried out at a high rate (at F_s). An optimization can be done (the Fig. 3-16 depicts this change) at the expense of some functionality loss, placed in the derotor and decimation steps. If the FH detect block instead of issuing the channel index $\hat{p}$ uses its energy (the value of the spectrum bin at the channel index $\hat{p}$) decimation and filtering are implicitly performed and so the derotor, filtering F_s to F_b filtering and decimation can be removed. However, the output channel can only be $F_s/N\ Hz$ wide, while the previous method could adjust the bandwidth as desired. Furthermore, if there is a shift in frequency of the time-frequency grid, this correction can no longer be done in the derotor stage but must be done either by a new block or by directly changing the central frequency of the radio. In spite of these apparent flaws, this method has the huge advantage of being able to handle high throughputs. Moreover, it does not imply any additional cost in complexity but a simple adjustment at startup which can easily be done within an SDR.

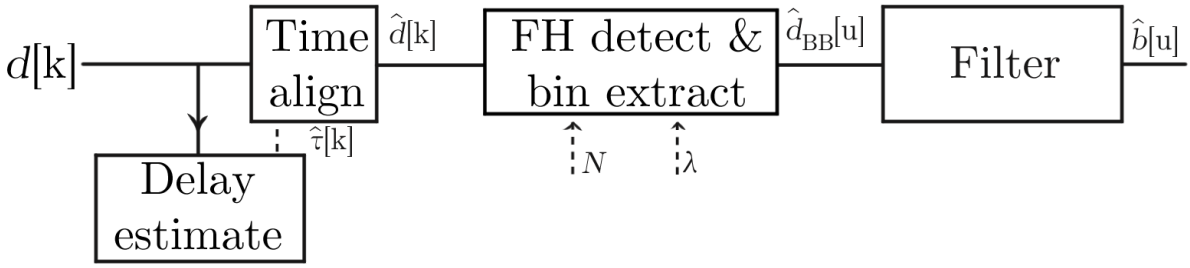


Figure 3-16 – DFT-BE detection algorithm.

3.3-2 FFT-Bin Extraction method

Our system is constrained by real-time use. For example, considering 16-bit IQ samples, a standard 1 gigE could only withstand 25 MS/s, a 1 line PCI-Express 50 MS/s. It is necessary to use a 10 gigE link to reach 200 MS/s which is the capacity of the radio we will use. At this speed, the data rate reaches 800 MB/s. For these throughput requirements, an FPGA-based acceleration is used, whose principle and design is described in the Chapter 5. A DFT allows choosing precisely the number of channels but it is difficult to implement efficiently on an

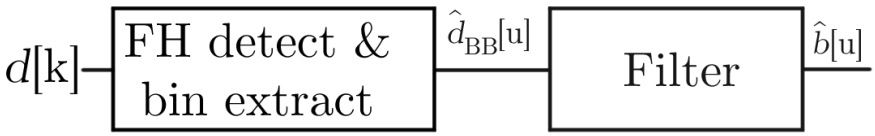


Figure 3-17 – FFT-BE, FPGA accelerated detection algorithm.

FPGA, for this reason a fixed size FFT will be used (more channels can be detected as $N < N_{FFT}$, in our case $N_{FFT} = 1024$). An aggregation of bin can be done on host side to get back to N channels.

In addition, the λ averaging has been removed in order to be able to detect fast hops. The averaging removal makes the system less sensitive to desynchronization so that the synchronization stage can be removed without significantly affecting the detection performance. However, the eventual spurious noise will not be filtered anymore.

The system is then greatly simplified as shown in Fig. 3-17. As well as the detector which can be expressed as:

$$\hat{p} = \underset{p\in[0;N_{FFT}-1]}{\arg\max} \left\{ \left| \sum_{m=0}^{N_{FFT}-1} \hat{d}[N_{FFT}+m] e^{\frac{-j2\pi mp}{N_{FFT}}} \right|^2 \right\}. \tag{3-22}$$

It should be noted that the number of channels of the FH signal and the one provided by the detector are not forcefully identical, as well as their respective bandwidth. However, this does not prevent the proper detection of the FH signal as long as the alignment and channel width are adequate.

3.4 Discussion on FH interception methods

	No time sync requirement	Very short burst detection	Low SNR capability	Robust to spurious	Computation requirement
DFT-BE	✗	✗	✓	✓	✓
FFT-BE	✓	✓	✗	✗	✓✓
Filter bank	✗	✗✗	✓	✓	✗
Wavelet	✗	✗	✗	✓	✗✗

Table 3-2 – Comparison between the benchmarked detectors.

As discussed in the previous section, there are several ways to detect frequency hopping signals, each with its own detection performance, complexity and data requirements. However, we have a specific use case that imposes to design a specific algorithm. Our system must be able to operate in a real-time system with a large bandwidth, it must be able to differentiate several channels, and this without knowing in advance the signal to detect and finally the FH signal is supposed to be of higher energy than the noise and possible interferes. In the following, four algorithms (Wavelet, HDFT filter bank, DFT-BE, FFT-BE) capable of wideband detection without prior knowledge of the signal will be studied, a short comparison is made in Table 3-2 taking into account the requirements of synchronization, detection sensitivity, noise performance and complexity. An in-depth study with benchmark will be conducted within the Chapter 4, with the goal of characterizing their performance under various situations, in order to determine which one works the best.

Chapter **FOUR**

EVALUATION OF THE PROPOSED APPROACHES IN THE TEMPEST CONTEXT

Contents

This chapter aims to characterize the performance of the three algorithms chosen earlier. Several cases will be evaluated considering two types of FH signals (as defined in the Table 4-1) corresponding to a straightforward case that can be easily encountered with commercially

available BT devices and a more secure case corresponding to a TRANSEC context. The performance will be evaluated on two aspects, firstly the ability to detect and follow an FH transmission and secondly the ability to extract hidden data within this transmission.

First of all, a context of side-channel will be given by specifying the scope of study related to the present work. Next the performance of the synchronization system will be evaluated in Section 4-2.1, then the ability to detect channels in Section 4-2.2 and finally the retrieval of red signals in Section 4-3. The performance will be evaluated in simulations using the Julia language [Bez17b], in order to control all parameters, such as noise level, time-frequency sporadicity, waveforms, red signal and its associated mixing scheme.

1 Side-channel model

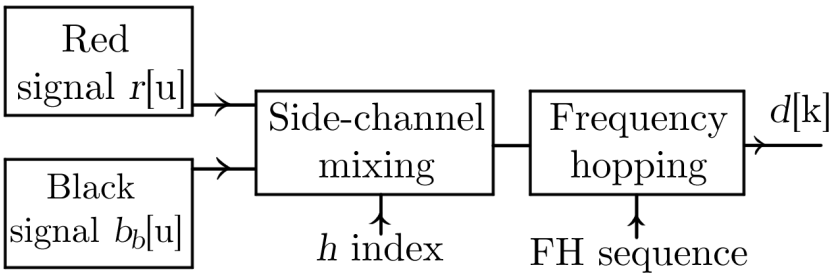


Figure 4-1 – Typical TEMPEST side-channel.

A side-channel signal is the mixture of a regular data (black signal) with a confidential illegitimate data (red signal) and can be modeled in several ways depending on its source (see Fig.4-1). In this thesis, we consider that the black signal carrier frequency follows a frequency hopping pattern and the red signal is combined with the black signal (baseband) in an additive way. This model corresponds to a large majority of observed TEMPEST side-channels. The mixing with the black signal is proportional with the factor h:

$$b[u] = b_b[u] + h \times r[u], \tag{4-1}$$

with $b_b[u]$ the baseband signal, $r[u]$ the red signal and u the sample index. The signal $b[u]$ has a normalized energy, which ensures an equivalent energy level between the different scenarios.

Moreover, the red signal studied in this thesis, when injected intentionally (in simulations

and in part of the real target tests) takes the form of an audio test pattern, similar to the one used in security audits.

A side-channel is characterized by a slight presence of the red signal. Indeed, they are mainly due to a hardware design flaws and therefore appear at low power levels (the modulation factor h is typically lower than 0.01). Also in a logical way, if the leak was important then it would have strongly impacted the normal operation of the device where it occurs and therefore would have been corrected at the device design stage or the device would not be functional.

The systems for recovering a red signal must be accurate (in the sense of avoiding rounding off during the computation of the slight presence of the red signal). They also must have enough bandwidth to capture all possible channels, and be able to process the necessary bandwidth.

Our proposed system has two stages, namely channel detection and red data recovery. The following sections focus on the evaluation of these two stages by considering first only the channel detection and then in a second phase the extraction of the red signal after having undergone the channel detection.

2 Detector evaluation

The detector evaluation can be characterized in two ways, the ability to perform proper time synchronization and the actual detection of a used FH channel. These two aspects will be discussed respectively in the Section 4-2.1 and Section 4-2.2. For the purpose of the simulations (and only them), the parameters of the signal to be intercepted will be assumed to be known to precisely estimate the impact of the various parameters.

2.1 Influence of timing synchronization mismatch

The synchronization algorithm defined in Algorithm 1 (Chapter 3) aims to detect the hop events (but without knowing on which channels they are located) in order to compensate the delay τ. It is to be noted that only the algorithms which carry out an average over λ require a synchronization, because their purpose is to prevent the error due to the fact that a hop occurs inside the averaging window. Therefore, they will be applied to the DFT-BE but not to the FFT-BE. Indeed, the DFT-BE detector uses an averaging of width defined by λ and N and thus operates on a window of $N\lambda$ consecutives samples. If a hop occurs, i.e., the samples fed to the

detector are located over two distinct channels, this would obviously lead to an error as only one channel is chosen over the whole duration of $N\lambda$ samples (equal to T_s).

The performance of the detector is evaluated in terms of Channel Error Rate (CER), defined as the probability that the estimated channel index $\hat{p}$ differs from the effective channel index p. The error is estimated for each sample, so an error is applied for each window $\hat{p}$. By estimating the error for each sample and not at each window it is possible to compare fairly detectors with different window sizes. Fig. 4-2 shows the CER versus the Signal-to-Noise Ratio (SNR) defined from (3-4) as σ_x^2/σ_n^2 with σ_x^2 the variance of the mixed signal defined in (3-1). This CER is computed for various values of λ and different synchronization errors Δ_τ. The time synchronization error is defined as the delay error normalized with the slot duration, i.e.:

$$\Delta_\tau = \frac{|\tau - \hat{\tau}|}{T_s}. \tag{4-2}$$

The simulation parameters are $N = 80$, $F_s = 80$ MHz and $F_b = 1$ MHz. The value of λ is associated to slot duration from 2 μs (i.e., very short, $\lambda = 2$ in purple) to 200 μs ($\lambda = 200$ in blue). We have also depicted the case $\lambda = 625$ (in green) which corresponds to the Bluetooth case. It shows that the DFT-BE detector is more resistant to the noise for high values of λ, as the noise is averaged in (3-16). Fig. 4-2 shows that the required fine synchronization is linked to the slot duration. In the meantime, one can say that a synchronization error lower than 15% of the slot duration leads to a small performance penalty (less than 3dB). This value is useful to know the minimum synchronization accuracy required by the synchronization stage.

The performance of the algorithm 1 is then evaluated by simulation on a TRANSEC-like signal with short slot duration. The simulation parameters are: $N = 80$, $F_s = 80$ MHz, $F_b = 1$ MHz and the slot duration are equal to $T_s = 2\mu s$ ($\lambda = 2$). The performance is assessed in terms of synchronization error defined above. Fig. 4-3 shows the synchronization performance according to B, the number of slots used to estimate the time delay. Performance is given for different SNR.

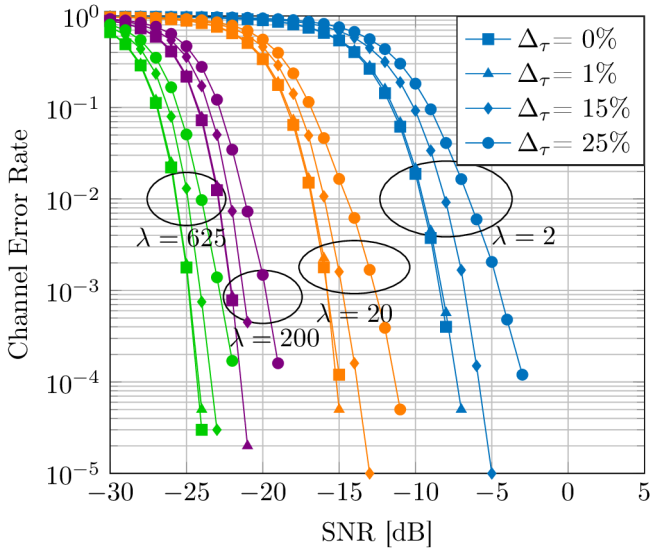


Figure 4-2 – Channel Error Rate vs SNR for various λ and delay errors Δ_τ with DFT-BE.

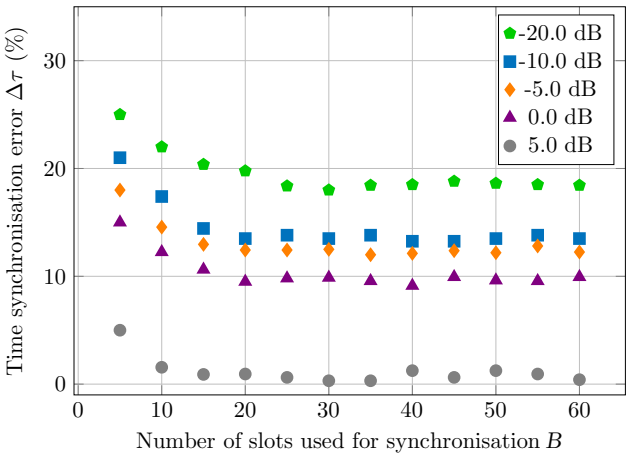


Figure 4-3 – Timing error of delay detection for various SNR levels.

Results show that synchronization is possible with only 5 hops. Besides, the accuracy is improved until 20 hops where a floor is observed. As the complexity of this processing increases with the number of slots used, we point out that the value of 20 used slots is a good trade-off between synchronization accuracy and computation time. The synchronization error is at this stage below the target of 15% (see Fig. 4-2). It should also be noted that these results can be extended to higher values of λ with comparable results.

	Number of channels N	λ	Number of samples per burst	T_s duration
Bluetooth LE	40	1 250	50 000	625 μs
TRANSEC	512	2	1 024	12.8 μs

Table 4-1 – Simulation parameters.

2.2 CER simulation benchmark

This section objective is to characterize the performance of the channel detectors (DFT-BE, FFT-BE, PPN and wavelet) in terms of CER under different situations:

— sample delay (residual synchronization error),

— various black signal modulations,

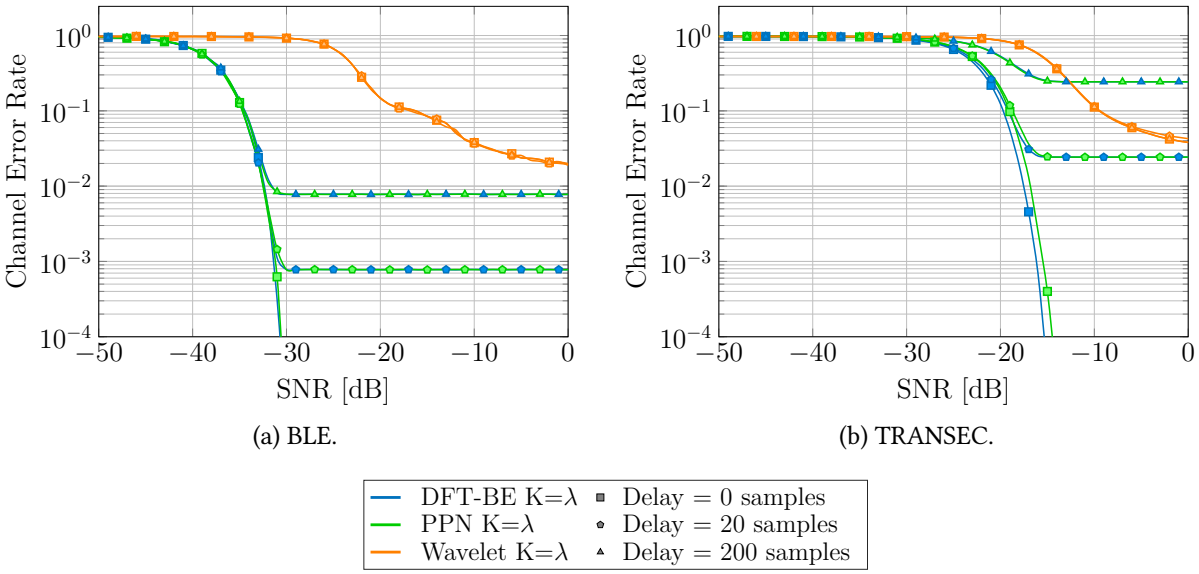
— various time slots and detectors averaging.

Two use cases have been defined; first a simple case corresponding to the Bluetooth Low Energy (but with various black signal modulations) and a second more complex one corresponding to a TRANSEC transmission case (this type of transmission is not standardized and is corresponding in our case to a fast signal with many channels) whose parameters are given in the Table 4-1. The FFT-BE will not be simulated (unless in Section 4-2.5) because the operating principle is identical to DFT-BE and the performance of the FFT-BE can be approximated with a DFT-BE (with $K = 1$). A CER below the threshold of 10^{-2} will be considered sufficient for an interception in the case of an adequacy between the number of channels of the detector and the generated signal

For all simulations unless specified, the signal is in perfect synchronization with time-frequency grid (i.e., no CFO, no delay, the time slot T_s, baseband frequency F_b and number of channels N are known), the time slot multiple λ equal the averaging factor K, the baseband signal b is a sine wave with no red signal injected. The scales of figures axes for each section are identical to facilitate comparison. For the following figures, some curves are pooled and the legend is then common for all the combined graphs.

2.3 Impact of delay

The delay represents a shift of the time-frequency grid in time and basically means that the detectors can receive samples from two different channels during the same processing



(a) BLE.

(b) TRANSEC.

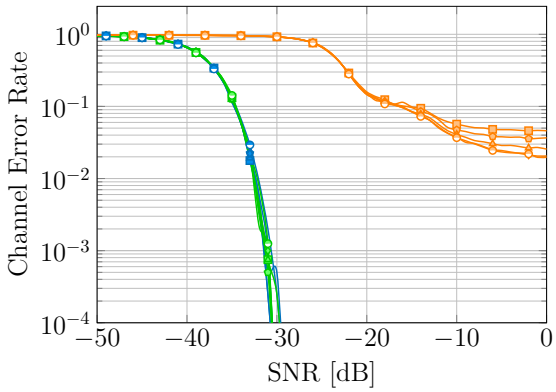
Figure 4-4 – Channel Error Rate vs SNR for various delays.

windows. So, at each hop there will be samples that are labeled with the wrong channel $\hat{p}$. This leads to the creation of high SNR error plateaus which can be seen in Fig. 4-4. The CER plateaus value can be estimated to $\alpha\frac{delay}{\lambda}$ where α is $min(\frac{N}{\lambda}, \frac{\lambda}{N})$.

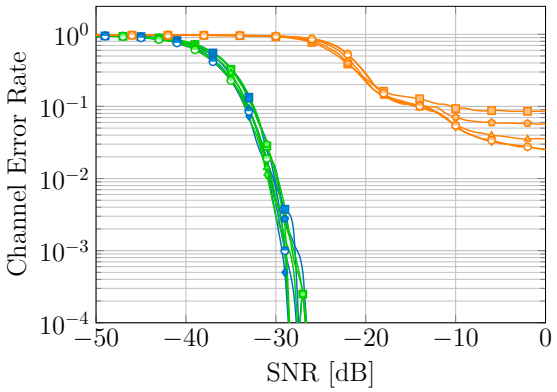
This benchmark shows that the PPN and DFT-BE algorithms have very similar performance while the wavelet has a much lower performance. The difference between TRANSEC and BLE is a 16 dB loss in performance. It should be noted that the plateau phenomenon is only due to averaging, lowering the λ will decrease this phenomenon, and cancel it in the case of the FFT-BE which does not use it (at the price of CER penalty due to the absence of averaging).

2.4 Impact of black and red signal modulation

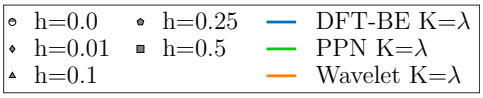
The modulation of the black signal as well as the addition of the red signal affects the overall waveform of the FH signal and can therefore alter the detection performance. In fact, the detectors observe rapid changes in frequency energy, if the waveform smooths out the channel boundaries then it will take longer for the algorithm to detect a significant change. An exploration of the waveform parameters is performed considering three black signal modulations: GFSK, BPSK and without modulation (which corresponds to a pure sine). For the first

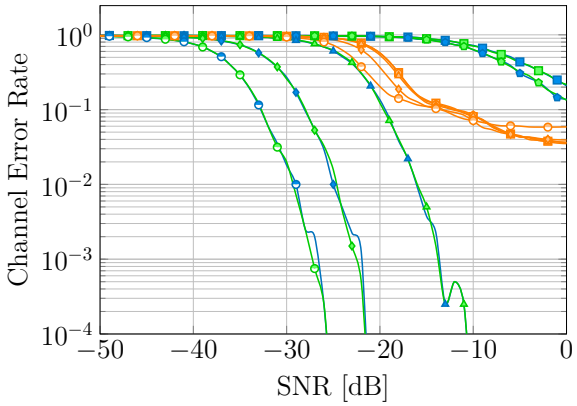


(a) No black modulation.



(b) BPSK modulation.





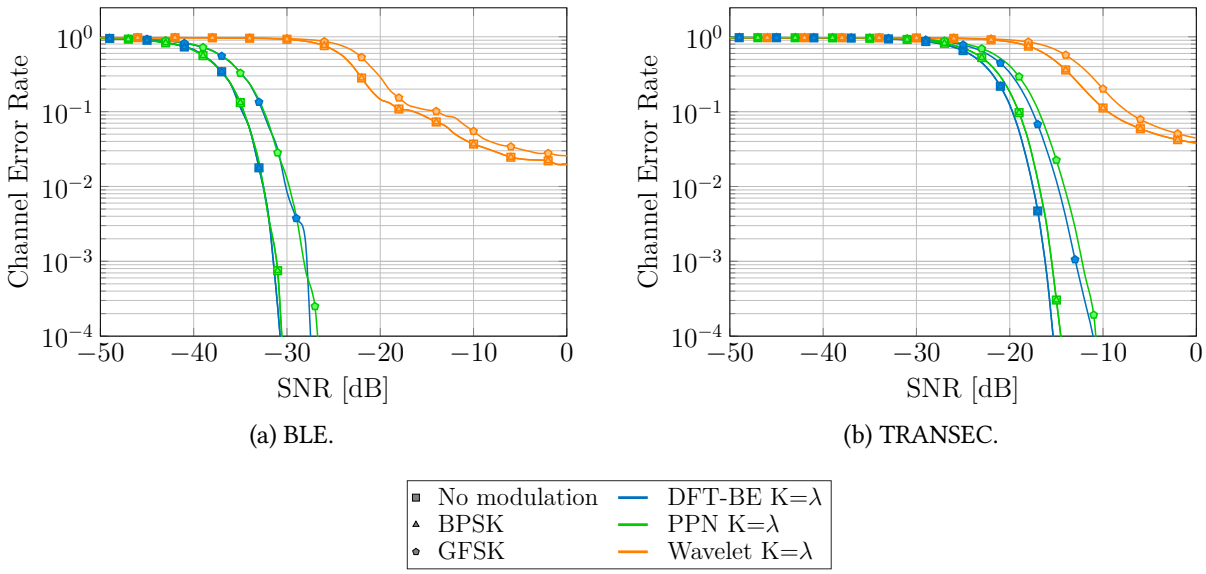
(c) GFSK modulation.

Figure 4-5 – Channel Error Rate vs SNR for various waveforms with BLE case.

two, the baseband message is a random binary sequence. The signal $b[k]$ has a normalized energy, which ensures an equivalent energy level between the different scenarios. An addition of red signal is made as described in Section 3-1 with various modulation indexes.

Fig. 4-5 shows the effects of exploring these parameters with a BLE scheme (a parallel between BLE and TRANSEC will be made in the next pool of figures), we see that once again the PPN and DFT-BE are very close and the wavelet has lower performance. For both BPSK and no modulation, there is little difference between the performance when changing h. However, cases where h index is greater than 0.1 have been plotted to check that interfering with the

baseband signal has consequences on the detector even if it would not correspond to a realistic side-channel.



(a) BLE.

(b) TRANSEC.

Figure 4-6 – Channel Error Rate vs SNR for various waveform with $h = 0.01$.

Fig. 4-6 shows a comparison between the two FH signal scenarios through the three waveforms, the h index has been fixed at 0.01 in order to have a realistic value. It is quite obvious that the more complex TRANSEC scenario has a much lower performance than the simple case, the results are very close between PPN and DFT-BE with the exception of the complex scenario where DFT-BE does better. As seen for the BLE, the GFSK gets lower detection performance in TRANSEC.

The PPN and DFT-BE are very close in performance and the wavelet has lower performance. In addition, the performance decreases as h increases in particular with a GFSK modulation. Since the GFSK has a slower time variation than the others, its detection takes a longer time and therefore more errors are due to the longer convergence time.

2.5 Impact of time slot duration and averaging factor

Fig. 4-8 shows the impact of the averaging on the performance. The generated FH signal is without red and black modulations and has several T_s expressed as a function of λ. The cases

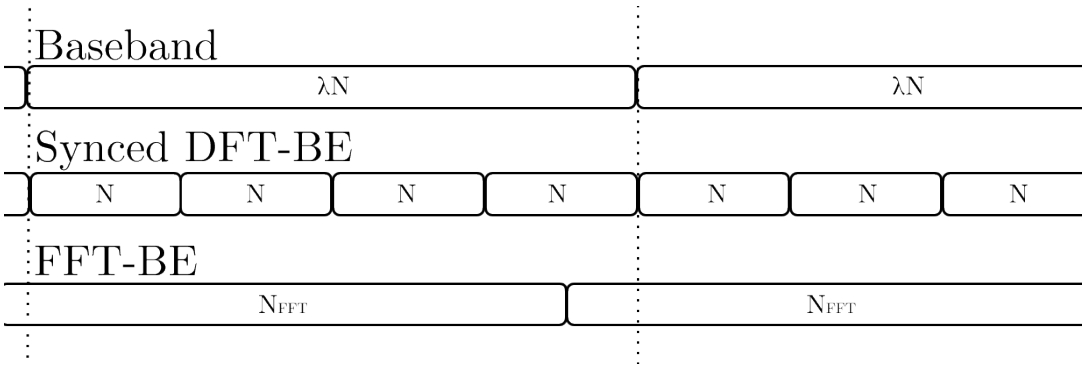
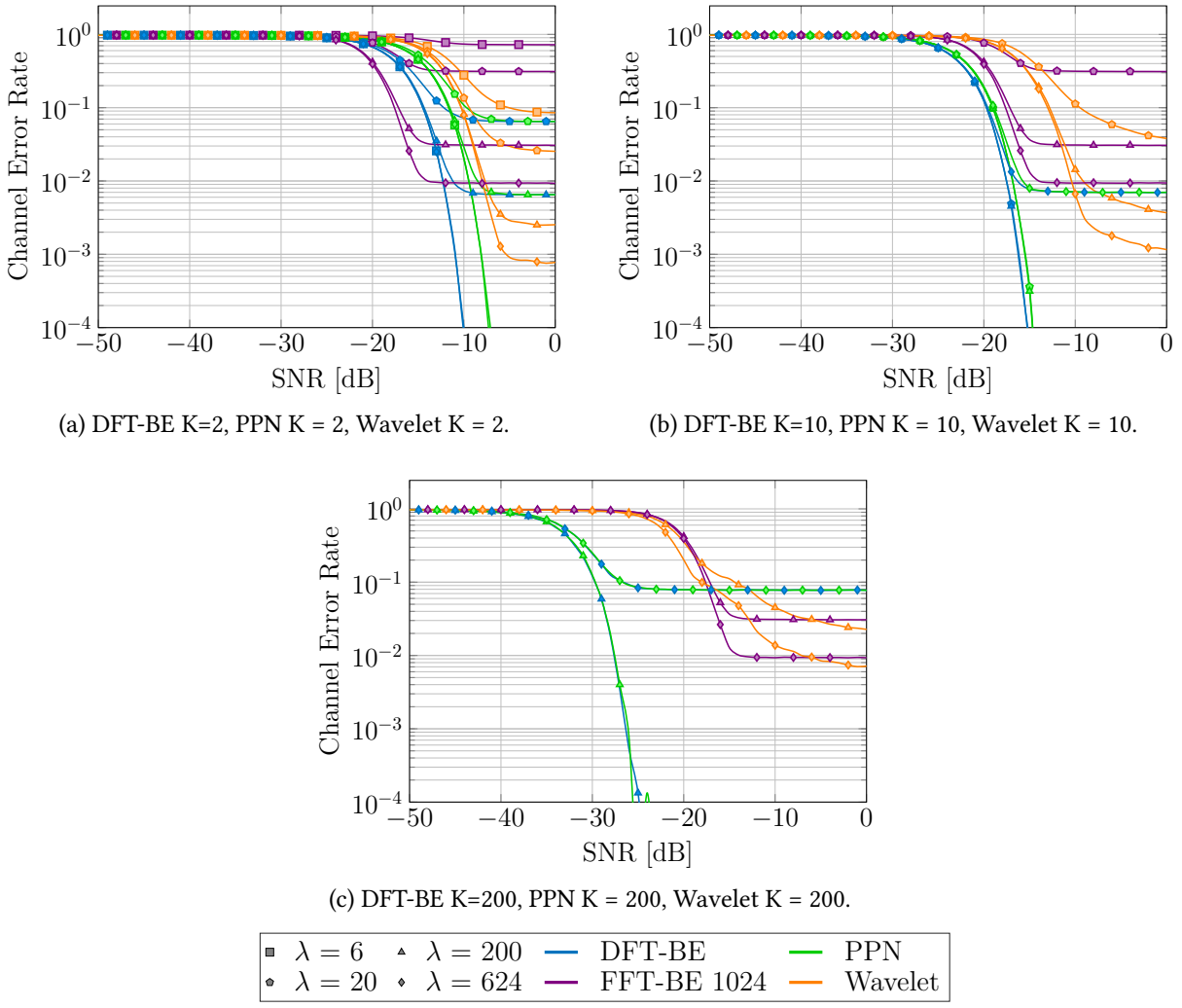


Figure 4-7 – FFT-BE and DFT-BE detection window.

where the averaging window is longer than the FH slot (i.e., $\lambda\ K > \lambda$) have been removed as the algorithm would work in an inaccurate range. The closer the averaging to the slot duration, the better the performance for the PPN and DFT-BE and ideally we should have $\lambda = K$. It is less the case for the wavelet, while the FFT-BE having no internal averaging there is no change for this algorithm, and it is for this very reason that these performances are not as high as for the DFT-BE. When λ is high, the signal hops less often and consequently there are fewer errors due to convergence time. Moreover, the performance of the FFT-BE being only dependent on λ, the performance can be qualified by $CER = \frac{N_{FFT}}{10 \times \lambda \times N^2}$ (with N_{FFT} the number of FFT bin) when the errors are only placed at the transitions (the 10 factor is due to the convergence time as a multiple of λ of the FFT-BE). The case of the FFT-BE is particular because the sample window is not synchronized with T_s as it is shown in the Fig. 4-7. For recall, an estimate of $\hat{p}$ is given per window (so each sample of the window has the same $\hat{p}$). A hop (the dotted lines in the figure) can happen in the middle of the processing window. The estimation of the CER being done for each sample, there will be a large number of errors that will be simply due to the fact that the processing window and the baseband λN window do not match. This problem is not present with synchronized algorithms like DFT-BE.

The evaluation of the different algorithms shows that the wavelet based detector achieves poor performance and is therefore not a viable choice for a real-time use (due to its high computation requirement), so this algorithm will be dropped for the next benchmarks. The DFT-BE and PPN algorithms have a similar and acceptable performance. However, the computational complexity of the DFT-BE is much lower than the PPN, which makes the DFT-BE a better choice for our use case.



(a) DFT-BE K=2, PPN K = 2, Wavelet K = 2.

(b) DFT-BE K=10, PPN K = 10, Wavelet K = 10.

(c) DFT-BE K=200, PPN K = 200, Wavelet K = 200.

Figure 4-8 – Channel Error Rate vs SNR for various time slot duration and averaging factors.

3 Red signal recovery simulation benchmarks

The final goal of our algorithm is the detection of red signals, which requires the detection of channels beforehand. This section is dedicated to the evaluation of the ability to extract a red signal. Since the signal extraction is placed after the channel detector, a bad estimation of $\hat{p}$ will lead to a loss of samples containing red data, the final results will be influenced by both the red extraction performance and the FH channel detector. The red signal is the same as in Section 3-2.4, specifically, an audio test pattern consisting of a succession of pure tones of 28 ms at frequencies 250 Hz, 450 Hz and 750 Hz. In this section, three algorithms will be evaluated,

the PPN and DFT-BE as seen before but also FFT-BE described in the Section 3-3.3-2.

3.1 Estimators of audio red signal quality

In order to estimate the performance, we need to have an estimator of the quality of the received red signals. As the audio test signal is made of several single tones, several methods are available: an ear test, as used in some security audit, spectrum analysis or degradation estimation methods as used in audio processing algorithms. The goal is to have, by knowing the original audio signal, an estimate of the degradation undergone and to have a global quality estimator.

3.1-1 Human ears

Using the human ear is the simplest and quickest form of audio quality estimation but lists several disadvantages [You10]. First it is a subjective test depending on the listener and his/her concentration level, second it is not an automated version, third it requires a large number of listeners. Of course, the precision is not high and the quality of the estimation is quite coarse. Generally, the human ears estimation is a discrete value between 1 and 4.

3.1-2 Direct frequency analysis

The direct frequency analysis corresponds to the estimation of the red signal level by observing the recovered signal spectrum at a precise frequency, depending on the red signal that we want to detect. The estimated level is either an SNR or a Total Harmonic Distortion (THD)[Shm05]. These methods are fast and simple; however, the signal can slightly shift in frequency in the recovery process (or due to carrier frequency offset of the radio front end for instance) and therefore a realignment mechanism is needed in order to compensate the shift. It works well at high SNR but at low SNR, this realignment can miss the signal and match with a noise peak and thus completely skew the results.

3.1-3 Perceptual Evaluation of Audio Quality

The Perceptual Evaluation of Audio Quality (PEAQ) is a ITU-R standard (BS.1387 of 1998) widely used in audio processing quality evaluation. It is based on the perception of the human ear to evaluate the degradation between two signals (one is the reference signal without any

degradation and the other is the degraded signal). Its main processing steps are defined in Fig. 4-9.

First a weighted spectrum of 2048 points is computed, then a normalization respecting the frequency sensitivity of the ear is carried out. The signal is then convolved with the standard frequency response of the external and middle ear, before the internal ear model is added through the addition of noise. The 2048 samples are then partitioned into 109 channels with a Bark filter [Smi99], allowing for frequency spreading. For each channel, a time spreading is performed, then a parameter estimation and Model Output Variable (MOV) are calculated (this corresponds to some feature extraction suitable for audio degradation), and their combination thanks to a neural network allows for the estimation of the Objective Difference Grade (ODG). This final value is between 1 and 4 and defines the degradation, 4 for an undegraded signal and 1 for a severely degraded signal.

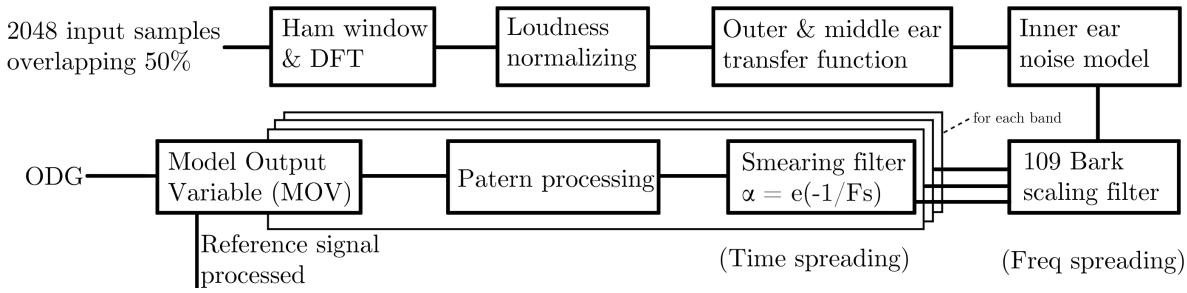


Figure 4-9 – Perceptual Evaluation of Audio Quality overview.

The PEAQ is functional and accurate in the case of low degradation, which corresponds to high quality audio signal impaired by lossy compression associated with audio codecs. However, it fails in presence of significant impairment, and among other things can estimate a high ODG when the tested signal is white noise. Furthermore, it fails drastically with impairments that are caused by sample losses [Kha17], and unfortunately this error will be frequent with each channel detection error in our application. Thus, this would not invalidate the side-channel and prevent audio content from being recovered.

The PEAQ should not be mistaken for the ITU Perceptual Evaluation of Speech Quality (PESQ) that works only for voice, our audio test pattern does not follow the characteristics of a human voice and the results will not be correct with this method.

A final argument against PEAQ is that it is not open source and requires a license to use, however a Matlab version exists [Kab02] but this one is not from ITU but by a researcher who

replicated the concept and shows similar ODG values compared to the licensed version.

3.1-4 Virtual Speech Quality Objective Listener

The Virtual Speech Quality Objective Listener (ViSQOL) is an open source alternative to PEAQ [Chi20]. It is intended to be used in the same conditions and also uses the specificities of the human hearing to operate. This estimator has been chosen to estimate the red signal quality recovered through the qualities that will be described below. For the use on FH extracted signals, the last steps of ViSQOL have been modified and its principle is given in Fig. 4-10. This modification only takes into account specific frequency bands related to the test pattern used and has been renamed Selected Bands Opinion Score (SBOS) to avoid confusion with the original version.

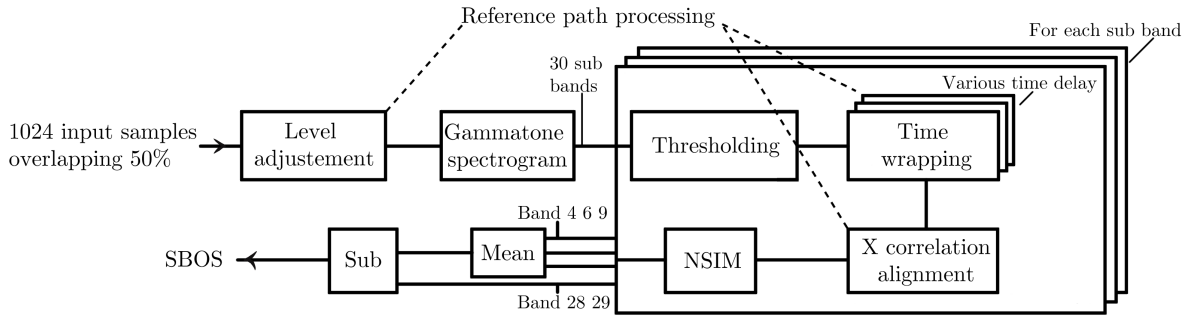


Figure 4-10 – Modded Virtual Speech Quality Objective Listener overview.

The algorithm starts by adjusting the power level between the two signals to be tested against, extract a spectrum but only generates 30 bands. On each of these bands, a temporal and frequential adjustment is made in order to match the two signals to be compared, then an Neurogram Similarity Index Measure (NSIM) [Hin12] is made. This step corresponds to use a computational model of the auditory periphery (like PEAQ) to extract similarities between the signals. However, unlike the PEAQ which only outputs a degradation index for the entire spectrum, the ViSQOL does this for each of the 30 bands. In the original version, a Support Vector Regression (SVR) Mapping using a pre-generated model is used to obtain an ODG. The modified version will select the three bands corresponding to our test signal, average them and remove the estimated noise level based on empty bands (they are empty only in our specific case).

	Automatic detection	Sparse signal tolerance	Low quality audio tolerance	High noise tolerance
Human ear	✗	✓	✓	✓
Bin energy	✓	✓	✗	✗
PEAQ	✓	✗	✗	✗
ViSQOL	✓	✓	✓	✗
SBOS	✓	✓	✓	✓

Table 4-2 – Comparison of red signal quality estimators.

SBOS value	Signal quality
< 0.1	Inaudible
$[0.1 : 0.3[$	Very noisy
$[0.3; 0.5[$	Noisy
> 0.5	Good

Table 4-3 – Chart of SBOS quality estimator.

3.1-5 Comparaison of signal quality estimators

To summarize, several methods to detect the red signal quality are available and are compared in Table. 4-2. However, as we have seen, the SBOS method is the most suitable for our application and will be used in subsequent work. It should be noted that obviously if the nature of the red signal changes, then the estimators will have to be changed, only the direct frequency analysis could be used for other signals than audio ones.

As the SBOS scale is not standard nor known, the quality readings can be analyzed as shown in the Table 4-3. The important element of these values is the threshold of 0.1 where the signal is considered audible and side-channel can be exploited.

In order to allow a finer comparison between the algorithms, mechanisms allowing improving the quality of the extracted audio signal is not used. The signal is, on the other hand, filtered and decimated (by a factor 41 for the BLE and 3 for the TRANSEC, but this last one having narrower FH channel bandwidth decreases the extracted noise). Adding an audio reconstruction system would add the performance of the reconstruction process to our benchmark and therefore will not be adequate to estimate the performance of the interception alone.

The remaining section objective is to characterize the performance of the red signal recovery

in terms of SBOS under different situations:

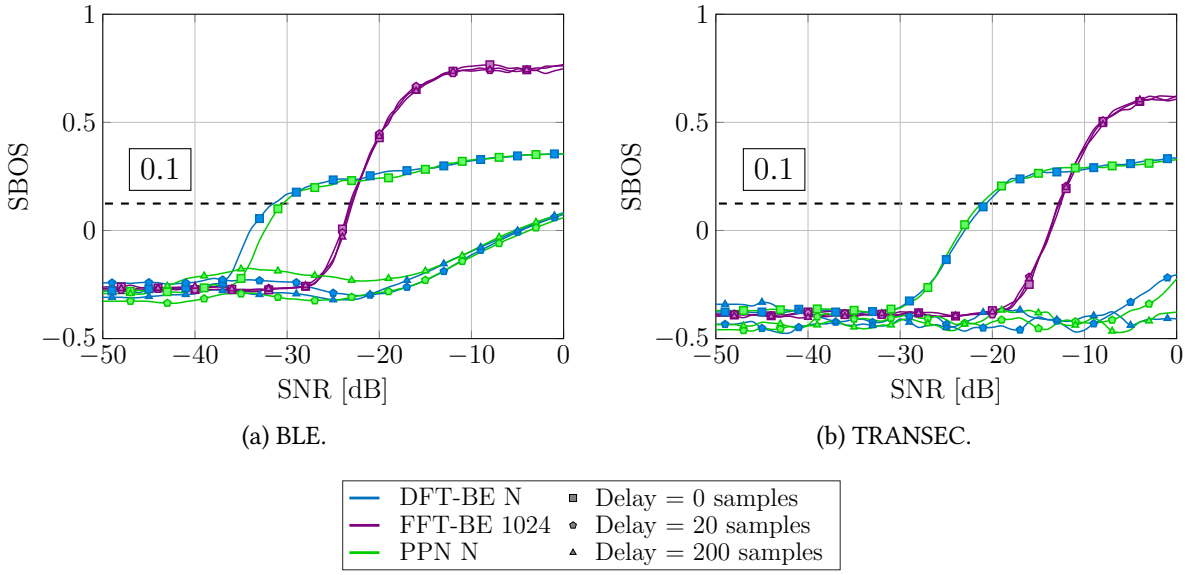
— sample delay (residual synchronization error),

— various waveform,

— various time slot and detectors averaging,

— various time sporadicity.

3.2 Impact of delay



(a) BLE.

(b) TRANSEC.

Figure 4-11 – SBOS vs SNR for various delays.

The impact of the delay (see Fig. 4-11) is similar to that observed with the CER analysis, i.e., a delay leads to a loss of channels and therefore of red data. However, a slight difference can be observed in the BLE case where the time slot is quite long. Indeed, the losses are more limited and there is no significant difference between a delay of 20 and 200 samples. As opposed to the TRANSEC case where a 200 samples delay does not allow to cross the threshold of audibility of 0.1 (to put in perspective a Bluetooth signal has $50\ 000$ samples per T_s while the TRANSEC has only $1\ 024$).

The most interesting information that emerges from these curves is the resilience of the FFT-BE method to the time delay. It even obtains better performance than the DFT-BE method in some cases although its CER performances are inferior. The method presented here does not use a synchronization mechanism, so averaging over K does improve CER performance but does not improve audio reconstruction performance, which confirms the earlier statement in Section 3-3.3-2 to not include synchronization on the FFT-BE version embedded in an FPGA. One explanation of this behavior is that the FFT-BE gives a better estimate of $\hat{p}$ per T_s and while in general there is more errors, these errors are short and of smaller impact. The filtering and decimation stage used to extract the red signal are consequently able to reconstitute the signal more reliably.

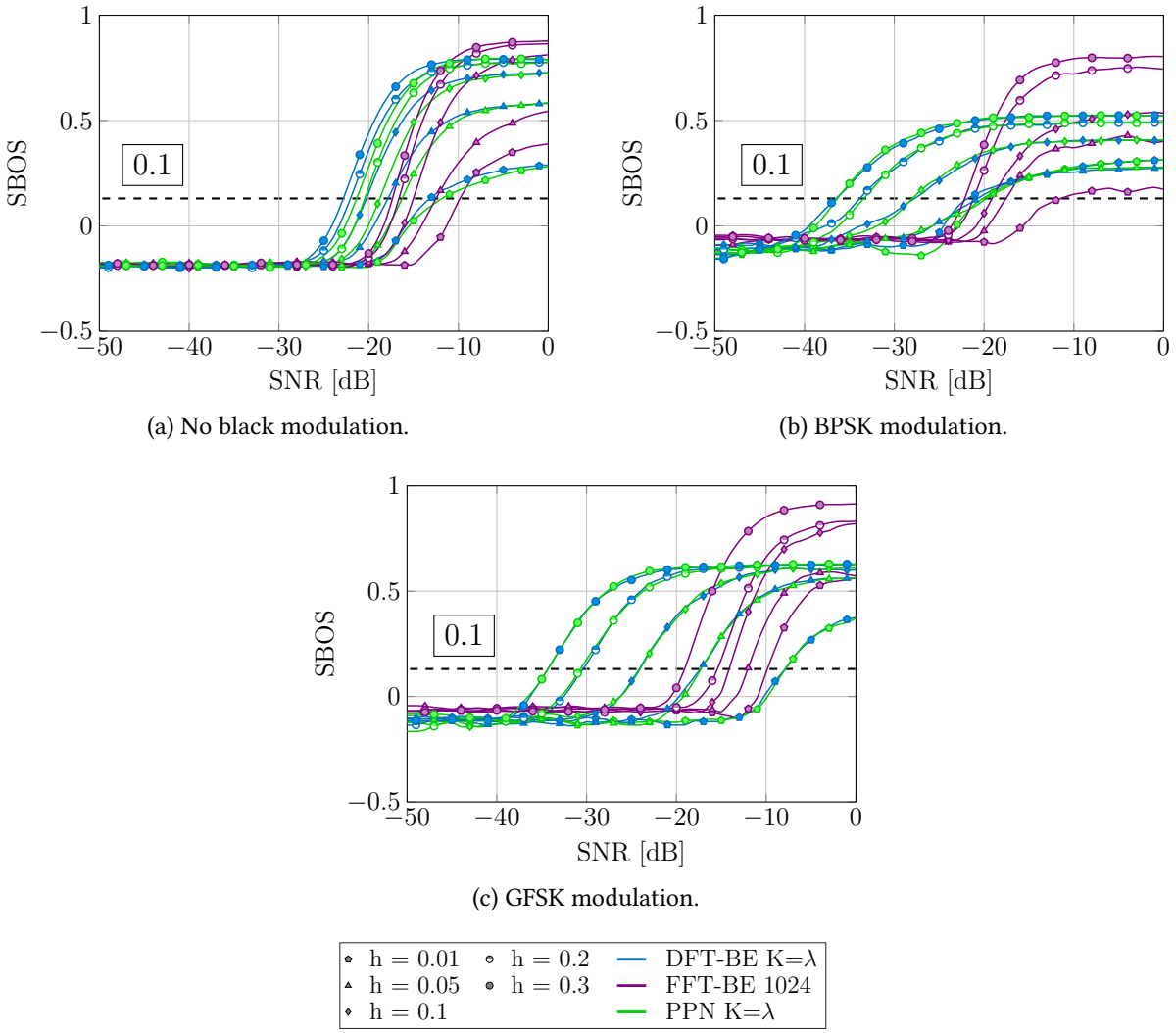
3.3 Impact of black signal modulation

The study of the waveform in Fig. 4-12 reveals a multitude of information, first of all the higher the mixing index h the better the red signal is recovered. This is normal as the index does not affect the CER performance significantly for the BPSK and without modulation. Additionally, increasing the power of the red signal leads to better reconstruction. The BPSK only suffers a loss of 3dB compared to no modulated signal with CER. In the case of SBOS there is a big difference in terms of performance. BPSK performs better in audio recovery while its performance in CER is inferior compared to no modulation.

A higher mixing index h increase the red quality at the end of the recovery step. The GFSK obtains very good performance compared to the other black modulation, while its performance in CER is very inferior. This strengthens the statement that good performance in CER does not necessarily means good performance in red recovery. The results shown here involve only the TRANSEC scheme however similar results are obtained for the BLE case.

3.4 Impact of time slot duration and averaging factor

Fig. 4-13 shows the impact of the averaging on the performances. The generated FH signal is without black modulation (only sinus wave) and the red mixing index is of 0.1, and has several T_s expressed as a function of λ. The cases where $K > \lambda$ have been removed as the algorithm would work in an inaccurate range.



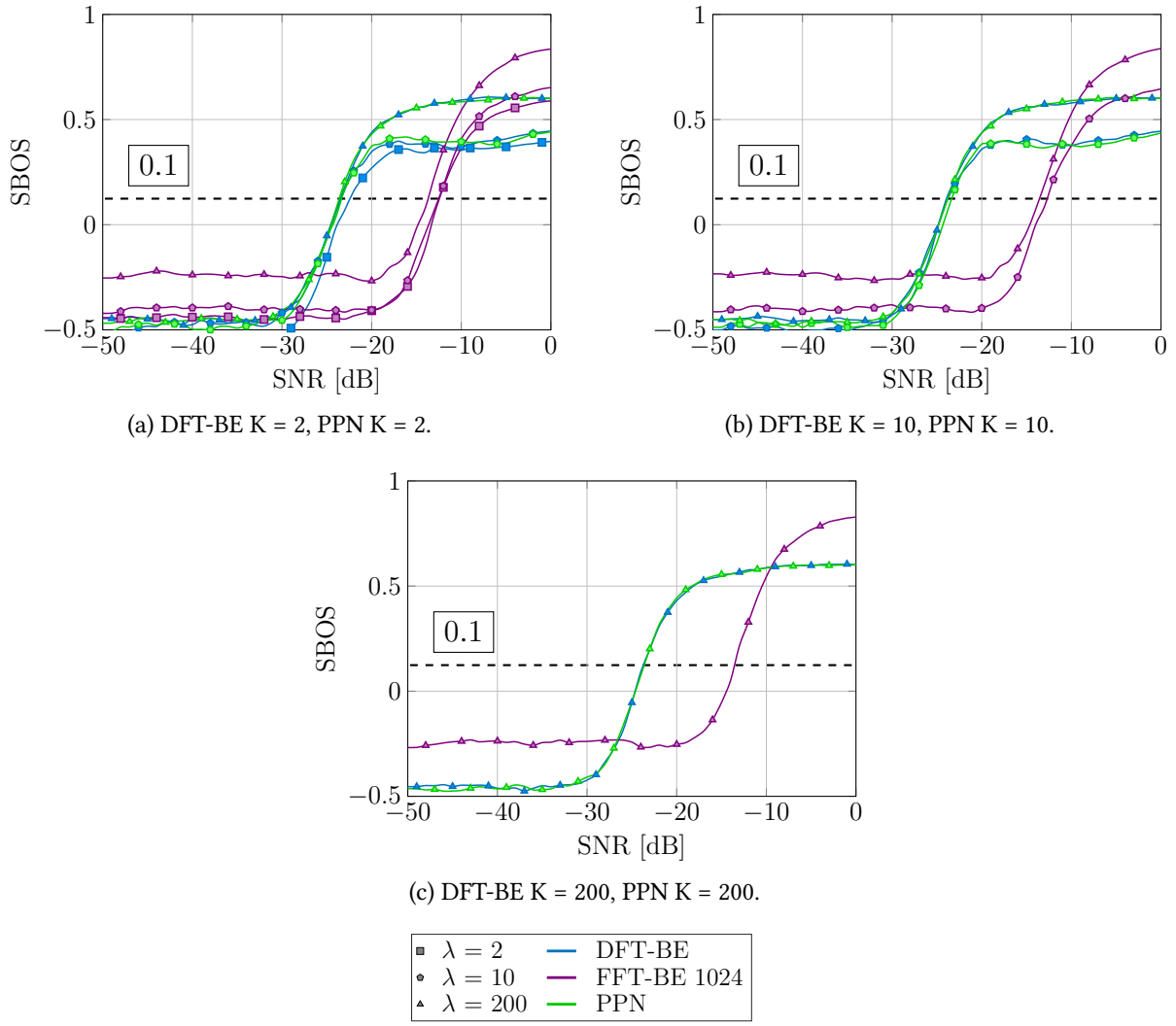
(a) No black modulation.

(b) BPSK modulation.

(c) GFSK modulation.

Figure 4-12 – SBOS vs SNR for various waveform for TRANSEC case.

An averaging does not bring a strong gain of performance, the duration of a time slot has, on the other hand an impact on the performance, in fact the less there are hops the less there is misdetection (due to the convergence time of the algorithms). DFT-BE and PPN are again more efficient at low SNR thanks to their averaging, while the FFT-BE method has better performance at high SNR.



(a) DFT-BE K = 2, PPN K = 2.

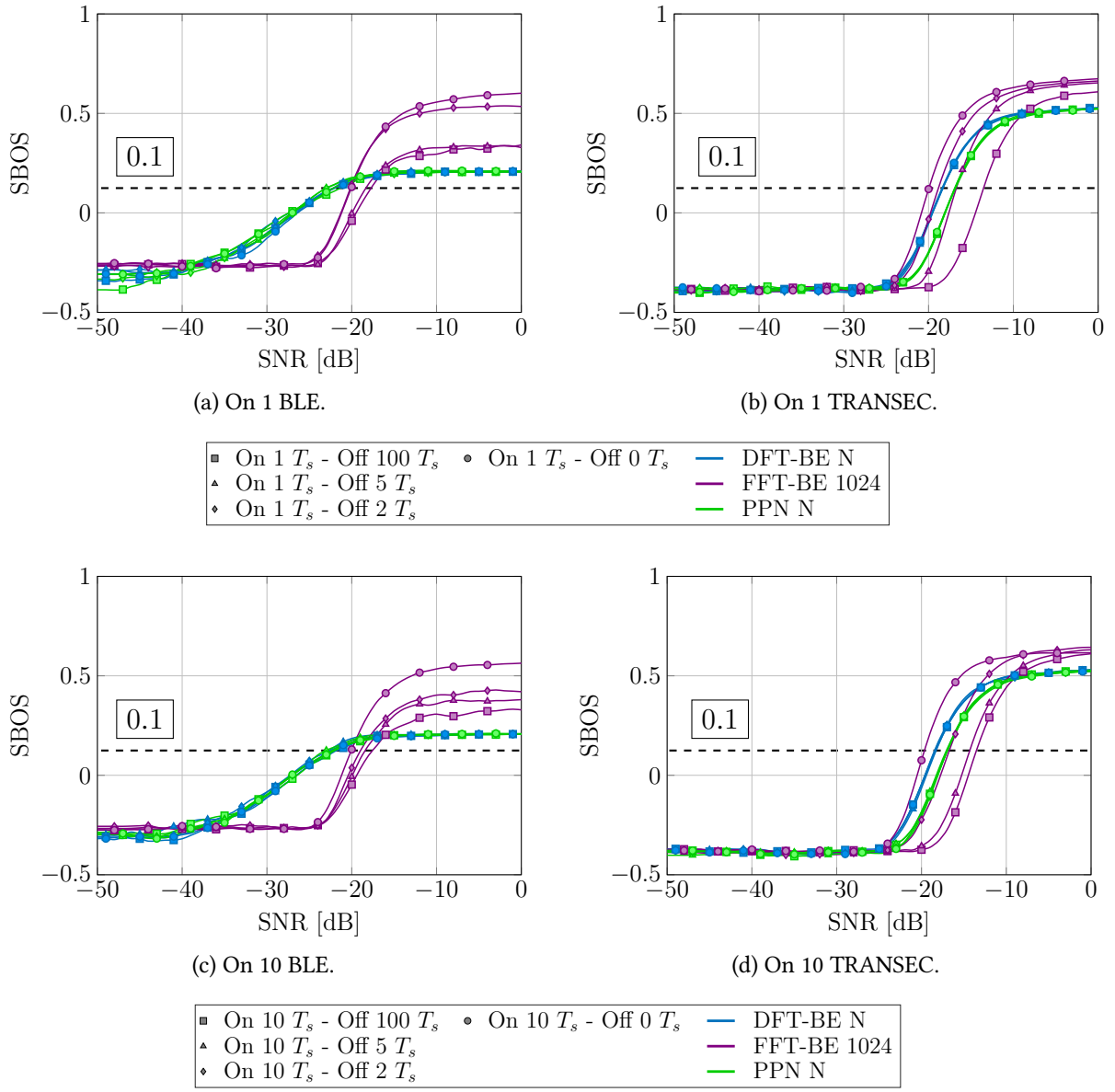
(b) DFT-BE K = 10, PPN K = 10.

(c) DFT-BE K = 200, PPN K = 200.

Figure 4-13 – SBOS vs SNR for various time slot duration and averaging factor.

3.5 Impact of time sporadicity

Finally, the last benchmark deals with temporal sporadicity. It represents the fact that the FH signal may not emit anything during some time slot. This point was not addressed for the CER because it did not make sense to estimate the detection performance when there is no signal to detect. Time sporadicity has been modeled in our case as a duty cycle where the FH signal is present for a given number of T_s (On T_s in Fig. 4-14) and then disappears for a fixed period (Off T_s in Fig. 4-14). Obviously, increasing the duration between two FH transmissions decreases the performance because the red signal cannot be recovered, however, increasing the



(a) On 1 BLE.

(b) On 1 TRANSEC.

(c) On 10 BLE.

(d) On 10 TRANSEC.

Figure 4-14 – SBOS vs SNR for various time sporadicity.

duration when the signal is active does not significantly increase the performance and a single T_s regularly is sufficient to recover a signal.

The DFT-BE and PPN get much better performance at low SNR with the BLE but their performance decreases, even reaching the FFT-BE with TRANSEC. The latter has no major change in performance between the various scenarios when the duty cycle is high, and a slight decrease in the other cases.

4 Conclusion on benchmarks

The different simulations have been useful to characterize the different algorithms. Firstly, the performance of the time synchronization is known, notably the number of samples required to have a sufficient time synchronization in order to make the DFT-BE working in acceptable condition. The various CER benchmark showed that the wavelet implementation is not very efficient.

The filter bank PPN and DFT-BE algorithms have very similar performance even if the DFT-BE is slightly more accurate in both CER and SBOS. In terms of number of computations, the filter bank is, however more demanding than its DFT-BE counterpart.

The FFT-BE revision which is designed for real-time use and therefore loses some features compared to the DFT-BE such as averaging, has relatively good performance in SBOS, and this despite the fact that its performance in CER is not outstanding. The cases for which the DFT-BE is better than the FFT-BE are however in special cases and not very realistic (for example a synchronization to the exact sample).

The various benchmarks showed that the performance of CER alone is not sufficient to estimate the performance in the TEMPEST context, because it does not give any insight on the ability to reconstruct information. SBOS is difficult to compute but gives a level of threat (associated to red data extraction) that CER does not. Moreover, a better performance in CER does not necessarily give a better performance in SBOS (see the performance compared of the DFT-BE and FFT-BE in Fig. 4-8). It is therefore important to evaluate the parameters, another kind of red data will lead to the use of a different estimator than SBOS but the CER will still be

valid.

To conclude, the FFT-BE offer a very good compromise between performance, noise and desynchronization resistance. This confirms our choice to use it in practice in our interception system which will be presented in detail in the following chapter, with its features, implementation constraints and, of course, the developed solution.

Chapter **FIVE**

ON THE USE OF SOFTWARE DEFINED RADIO FOR TEMPEST

Contents

This chapter aims at highlighting the potential of SDR use in TEMPEST scenario, especially in an FH context. By utilizing the SDR processing capabilities and the various processing acceleration capabilities, this chapter will illustrate the possibility of acquiring a 200 MHz bandwidth and processing it in real-time using the FFT-BE algorithm described in Section 5-3.3-2. To begin, an introduction to SDR is made in Section 5-1, and a special attention will be paid to

the notion of high-throughput data processing and to the means to achieve it. We introduce the two-language problem which can be addressed with the Julia language. The question on how we can program such device is discussed in Section 5-2. Finally, the implementation choices of the FFT-BE algorithm will be presented in the Section 5-3.

It should be noted that our radio has a different bandwidth between the radio front end (160 MHz) and the associated DAC (200 MHz). The processing being carried out through the samples provided by the latter, we will refer to a bandwidth of 200 MHz because it is actually the rate of the processed samples, although only 160 MHz contain useful signal.

1 Software defined radio

1.1 SDR definition

Radio telecommunication systems were originally built on the basis of ASICs, i.e., specialized circuits that could only handle a single application [Chi02]. These circuits have the advantage of being very efficient in energy and in silicon surface because each etched component is necessary and none is unnecessary, unlike a GPP that has some functionality that will potentially never be used by an application. However, there are several major disadvantages: first, the design of such a system is very expensive in terms of both human development costs and factory costs, which require the manufacturing of a lot of ASIC in order to become cost-effective. The design is fixed during the design process and the slightest mistake in the design requires the replacement of some of the photolithographic masks [Sch76]. Finally, the system is also non-evolutive and each modification (for example to handle a new waveform) will require redesign and new masks.

In order to overcome these problems, several researchers had the idea to use the new capacities of the computer science to create reconfigurable radio systems: the software-defined radio. The first reference of this type of radio is from 1985 with a team of Texas Division of E-Systems Inc. with a digital baseband receiver that has been published in their E-Team company newsletter [Joh85]. The receiver was equipped with an array of processors that performed adaptive filtering for interference cancellation and demodulation of broadband signals. This system was a receiver only and therefore not a complete radio transceiver.

It is in 1993 with Joseph Mitola that the concept of SDR is given [Mit93]. SDR is the generic terminology that depicts a flexible digital signal processing architecture with very high

reconfiguration capabilities, by using as few analog components as possible and by locating the digital/analog conversion stage as close as possible to the antennas. A system of this nature offers several advantages:

- An SDR can be used for different applications
- A design error can be quickly corrected
- Prototyping can be easily done
- It is easy to upgrade the functionality
- Functionality through IP blocks can be easily exchanged between various projects reducing development time
- The systems created can be flexible and adapt themselves to their environment (this is also called cognitive radio [Mit02])

The primary goal of SDR is to shift as many blocks as possible into the digital domain (mixers, filters, modulators/demodulators, detectors or error correction ...) in order to use programmable devices. An ideal SDR structure can be seen in Fig. 5-1. In this representation, the signal is directly converted at the transmitter and receiver antenna by respectively a Digital to Analog Converter (DAC) and an Analog to Digital Converter (ADC). Such architecture delegates all the processing requirements to programmable digital components (such as GPP). However, the digitization of the signal on the antennas requires ADC and DAC technologies with a high sampling rate (for instance minimum 5 GS/s if you want to operate with WiFi), it is also required to have components that can handle the incoming data flow at a high rate. In addition, GPP-based SDR implementations exhibit some limitations compared to ASIC counterparts. Indeed, ASIC designs are highly specialized and are therefore able to achieve higher throughput while consuming several orders of magnitude less power compared to GPP.

In order to ease the pressure on the digitization stage, the SDR only monitors a part of the radio spectrum. For this purpose, a filter and a mixer connected to a Local Oscillator (LO) are used to lower the frequency of the signal [Rob10]. Two architectures are possible to obtain the baseband IQ samples in the desired band: Zero Intermediate Frequency (ZIF) and Low Intermediate Frequency (LIF). In ZIF receiver, the analog signal is mixed to baseband and two converters directly sample the I and Q data of the signal centered on DC (a parasitic noise is present at 0 Hz, i.e., in the middle of the band). A problem that can happen is that since the mixer LO tone lands at exactly DC, any DC leaks of the LO will bleed-through the mixer and becomes DC in the digitalized signal (the so-called DC Offset).

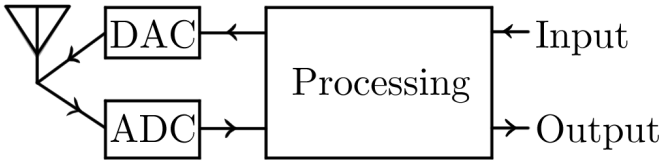


Figure 5-1 – General structure of an ideal Software-Defined Radio.

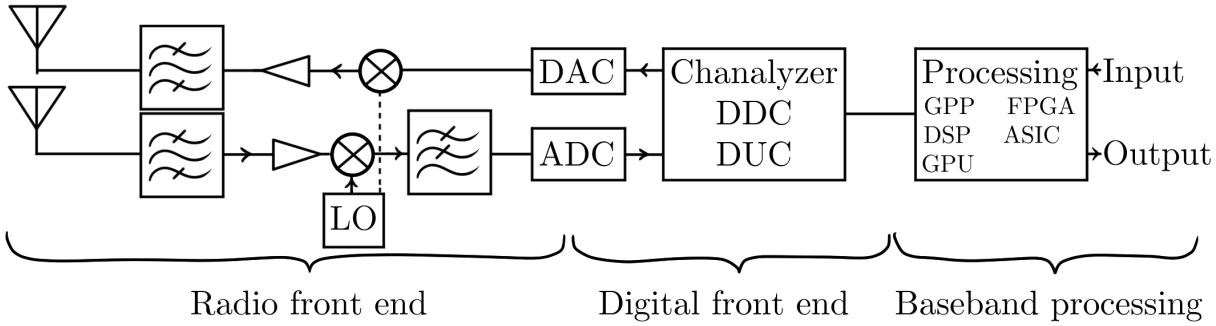


Figure 5-2 – General structure of an LIF Software-Defined Radio.

The LIF receiver brings the signal to a so-called intermediate frequency before it is lowered again (see Fig. 5-2). This step reduces the design constraints of the first stage frequency descent (which allows to realize it in a less accurate way) because the signal will be reprocessed thereafter. In the same way the DC spike of the LO can be erased at the 2nd descent in frequency which does not produce DC spike if it is realized in digital within an SDR. The LIF structure leads to image frequencies issue, which requires the synthesis of high-performance intermediate filters to cancel them [Val01].

The first stage of frequency decrease is usually applied in "Radio front end" and the second stage is included in the same hardware part or can be in the "Digital front end" with a Digital Down Converter (DDC)/ Digital Up Converter (DUC). A difficulty of the radio front end part is that it has to work over a wide bandwidth and be completely uniform, i.e., having the same sensitivity and noise rejection over the whole applicable spectrum. This specificity makes SDR at a given frequency noiser than a pure ASIC radio because it is complicated to have a uniform response. To compensate that, some radios have more precise response for specific frequencies in order to have better performance on widely used frequency bands (e.g. ISM bands, telephone bands).

1.2 Hardware acceleration

The high computational demands of data processing in an SDR, as well as the energy efficiency issue have led to supplementing GPP with various accelerators in order to unload them from part of the treatments. These accelerators are less universal than the GPP, but they can be used in many different ways, making them usable in a variety of applications [Dar12]. Several hardware accelerations can be used and some of them blur the boundary between SDR and radio ASIC:

- Multi GPP: This solution is not really a dedicated device but rather the addition of several GPP linked together. Instead of a single GPP, several GPP can indeed be used in parallel, where each one takes care of a predefined task. Multi-core systems can also be used instead of several individual GPP but all these systems must respect the principles of distributed programming [Glo02].
- Digital Signal Processor (DSP): are derived from GPP where some signal processing operations have been accelerated, for example operations like a multiplication followed by an addition are performed in one clock cycle instead of several as in a conventional GPP. DSPs commonly use custom memory architectures that can retrieve multiple data or instructions at the same time. In addition, DSP programs are generally more finely optimized (because they are used for highly optimized tasks) and have hand-optimized assembly-code libraries to maximize the use of the hardware instead of relying on the compiler to handle essential algorithms parts [Lee04]. Programs created for a GPP will work on a DSP but the other way is not always true due to hardware optimizations.
- Graphics Processing Unit (GPU): as their name implies, these processors were initially created for graphics processing but can also be used in similar tasks to image processing, i.e., matrix processing, parallel equalization or filtering. These processors are efficient for highly parallel tasks that require high memory throughput but their efficiency and usefulness is greatly reduced for other types of operations [Kim10]. In general, GPU are 5 times faster than GPP and 3 times faster than DSP [Gra13].
- ASIC: for some tasks requiring high performance, an ASIC accelerator can be used. These accelerators are highly specialized and are generally reserved for the complex tasks often encountered in designs as DDC/ DUC, standard protocol decoder (which will never be modified), Ethernet management used to connect the SDR with the outside [Che07].
- FPGA: these circuits have the quality of computational performance and parallelism of an

ASIC (while being less efficient in power consumption and speed) but having the ability to be rewired. Their programming is more extensive than for a GPP and it is possible to perform exactly the same tasks [Rod11]. It is therefore possible to avoid having a GPP in an SDR but only an FPGA. FPGA are different from GPP in the execution of their tasks. A GPP executes instructions linearly from memory and can easily switch from one function to another without slowing down the processor and does not require additional resources. In an FPGA, all functionalities are mapped to dedicated circuits and each new function requires more FPGA resources and can in some cases require a downgrade of the operating frequency.

As we have seen, there is no device that is better than the others, but each one has its advantages and weaknesses. A method to get the best out of it is to associate several of them and distribute the tasks to the different accelerators according to their affinity and the performance required for the application. This approach is adopted in the ZU28DR RF System on Chip (RFSoC), which combines a large FPGA with integrated ASIC hardware accelerators, a four-core ARM as the DSP accelerator and a dual-core ARM as the core GPP. This architecture is for example used in the SDR X410 [labc].

1.3 Programming paradigm

Programming SDR requires skills in embedded system programming because the SDR are generally compact with limited resources but also High Performance Computing (HPC) in order to sustain heavy computation load. SDR have some particularities, their programs are usually real-time, the data throughput can be important and the latency of the processing must be low.

1.3-1 Dataflow model of computation

Programming an SDR is generally done through a flowgraph, the samples arrive in a processing block, are processed and then sent to the next block, which is similar to a pipeline architecture as seen in a CPU. Each block performs a single task e.g. filtering, decoding, symbol remapping…Some control data link the block signals indicating that a block is ready to receive data or ready to send data. In practice, for architectures with independent processing blocks (e.g. ASIC, FPGA) or when a block is implemented in a separate device an actual link connects the blocks to each other but for GPP or GPU the blocks are executed sequentially. FIFO are

sometimes inserted between blocks to act as a buffer in case a block had a delay.

The different blocks can also be implemented with a centralized architecture that replaces the point-to-point connections between blocks with a shared point. This point can be a data bus (bus-centric), a processor (processor-centric), or a device (device-centric). This approach allows for greater flexibility and control (which only needs to be implemented in the central part), it is possible to have several identical processing blocks and the data will be routed to the block with the greatest availability, a rapid change in data flow is possible by routing the packets to new blocks, whereas in the case of the point-to-point version, this requires a link to be foreseen in advance.

However, this kind of architecture has an important problem that all data have to go through the central point which can lead to bottleneck and slow down the whole dataflow. Direct Memory Access (DMA) engine can be used in some cases to reduce the load of the central point. DMA is a feature that allows some hardware subsystems to access the memory of another device without going through its associated processing unit (CPU, GPU ...).

1.3-2 Constraints

Latency constraints: The radio works in real-time, which means that it processes the data faster than it arrives (i.e., *sampling rate* < *processing rate*). Increasing latency (due to jitter) can lead to cache overflow or data overwriting due to lack of memory capacity. For bandwidths below 1 MHz, GPP are commonly used but for higher bandwidth accelerators or FPGA based systems are to be preferred, not because GPP cannot handle the data fast enough but rather because they will have a too high energy consumption to be used [Gra13].

It should be noted that while latency is a constant delay which is therefore perfectly deterministic, jitter is a non-deterministic delay (but which can be bounded) which depends on events external to the task for example the execution of tasks in parallel on the same GPP, drop in network traffic overload).

Because of real-time operation required by a telecommunication system, response time (due to latency and jitter) is a major concern in SDR, the processing performance can be increased by operating with blocks of data, rather than on one sample at a time which can lead to an increasing latency. In cases where an SDR is built with a GPP at its heart, an operating system is used which can entail a jitter when handling interrupts and context switching for different threads. The jitter can exceed 10 ms on a Windows system but real-time operating systems such

as VxWorks can achieve jitter below 1 ms [Gra13]. Another latency to take into account is the data link between the different components, the radio part and the processing part are not in the same chip and need a link, as well as between the different accelerators. This latency is of $240\,\mu$s with USB [Sch07], under $10\,\mu$s for 10 GbE [Fen05], under $100\,\mu$s for 1 GbE [Hug05], up to $150\,\mu$s for General Purpose Memory Controller (GPMC) (used between a FPGA and a GPP in some SDR) [Tru13] or $7\,\mu$s with Peripheral-Component-Interconnect-Express-Bus(PCIe) [Li20].

Strictly speaking, latency is not a problem for a real-time system because the data will be processed but with a delay. However, it is a problem for a communication system because it entails a delay between the reception of a message and its reply. In the case of communication protocol, this delay is quite short for example it is $10\ \mu$s for WiMax, and within this period the data must be digitized, sent to the appropriate devices, processed and returned to the transmitter. Because of this issue, it was not possible to use an old SDR for WiFi communication because the link between the radio and the GPP was too slow.

Throughput constraints: The data throughput is also to be taken into account, it becomes important as the bandwidth increases. For example, considering a 16-bit transceiver (8 bits per IQ channel) the raw data rate, i.e., without the different link overheads and metadata (timestamp, error, performance report, RSSI ...) can be seen in the Table 5-1.

To give a rough idea, a USB 2 link only holds 8 MHz, 61 MHz for its USB 3.0 version, a standard 1 GbE Ethernet link in all modern devices can only support a bandwidth of 25 MHz, for larger widths it requires a 10 GbE Ethernet or multiple PCIe lane. We can also see that with a high bandwidth, the storage of data alone is complex because it is necessary to find a storage medium capable of withstanding the throughput (the best as of 2021 is Seagate FireCuda 520 Gen4 with 4.3 GB/s [Sea] which is not sufficient to register a 5G FR2 data link) and that a real-time processing is to be preferred.

One difficulty in programming SDRs is the multitude of languages. In the case of a GPP-based system everything can be done with a single language, but in the case where accelerators are used, a new language per accelerator category is typically the case. Also, each language has its own programming logic, an FPGA is not used like a GPU. This requires either a person with expertise in several languages or a team of several people. There are also languages dedicated to rapid prototyping and algorithmic development for example Matlab which allows to produce a working system quickly but not optimized and which is later translated into a more efficient language.

Radio bandwidth	Raw data throughput
15 kHz (standard radio music channel)	234 kB/s
100 kHz	1.5 MB/s
1 MHz (Bluetooth channel)	15 MB/s
22 MHz (Wi-Fi channel)	336 MB/s
100 MHz (whole ISM 2.4 GHz band)	1.5 GB/s
200 MHz (5G 3.4 GHz band)	3 GB/s
640 MHz (LTE-Advanced Pro)	9.5 GB/s
800 MHz (5G frequency range 2)	11.9 GB/s

Table 5-1 – Raw IQ data weight for standard radio bandwidth.

There is then a *two-language problem*, namely developing in a high-level language, fast to write and easy to debug, and then transcribing what has been developed in a low level language, harder to write but more efficient. This requires more development time to achieve the program, but this problem can be solved by using a language that is fast to write and easily optimizable, and this is the subject of the following section.

2 Towards a language for efficient use of SDR

The *two-language problem* expressed in the above statement can be solved with the Julia language. The purpose here is not to present the language itself, and the interested readers can have a look at [Bez17a] or [Bez18], but to rather give insights on why it is suitable for SDR prototyping. This section aims to present this language and to compare its performance with other languages applied to signal processing.

2.1 Language for efficient SDR prototyping

Julia is a programming language whose syntax is really close to scripting languages (for instance Python or Matlab) but that offers performance similar to compiled languages as it is compiled through LLVM [Lat04] and so works for various platforms. The Just In Time (JIT) compilation and the Multiple Dispatch (MD) feature bridge the gap between on one side interactivity and code concision (required for easy exploration and prototyping) and on the other side efficient compiled code (for good runtime performance). In this thesis, an open-source tool dedicated to SDR prototyping in Julia language has been developed: *AbstractSDRs.jl*.

2.1-1 Multiple dispatch

What it is: Julia has been built around several key ideas. The main one is to propose a compiled language with multiple dispatch, meaning that a function can be dynamically dispatched based on the runtime type of its argument, and not statically at compile time nor considering all possible argument cases. Each function will be optimized based on these arguments and the optimization will be kept in memory for the future calls of the function with the same type of arguments.

Why it is important: For performance. MD is the baseline of Julia language and it has been demonstrated that Julia uses it more than other languages [Bez17a]. It allows to have high performance while keeping the code execution paths tight and minimal. In our scope, it will also ensure the support of many different SDR binding through a common Application Program Interface (API) while being sure that appropriate function call is done, and this, at runtime. The second key advantage is the possibility to efficiently use fixed point processing [Bez17a] which is an often-used output ADC format.

2.1-2 Dataflow type inference

What it is: the typing of code is determined by the flow of data through it. It means that the code is applied to types and not to the associated values. Note that the types can be concrete, and, as it is stated in [Bez17a], ability to concretely type code is closely related to being capable to properly execute performance-critical code.

Why it is important: For performance. One can write efficient loops, that will be automatically unrolled and vectorized at compile time. It particularly eases the prototyping as efficient code does not require intensive optimization. This is particularly an advantage for SDR as we will often have to cope with buffers (i.e., data signals). With the use of this kind of efficient loops, effortless porting of computationally intensive processing can be achieved.

2.1-3 Call to low-level languages

What it is: In Julia, C and Fortran functions can be directly called without any additional glue code. It means that calling low-level C function requires no code generation nor additional compilation.

Why it is important: For portability. It has two key properties. First, it will ease the use of heavily optimized functions (already written in those languages) as bindings. Well known and established libraries can thus be easily ported to Julia. This is for instance the case of the FFTW library associated to FFT [Fri98]. Secondly, as most of the SDR drivers are written in C language, it will also pave the way for the integration of SDR bindings in Julia without any performance penalty. Note that, albeit not being incorporated in the base of Julia, external packages exist to exploit other languages such as C++ and Python, extending the scope of potential available libraries.

2.1-4 Multi-architecture and co-processors support

What it is: Julia supports different architectures. Regarding GPP, it allows to use Julia on both x86 and ARM. Due to the use of LLVM compilation engine, it is also possible to compute Julia code on GPU leveraging the use of co-processors [Bes18].

Why it is important: For scalability. SDRs have many different architectures and the same Julia code can be run on these different devices. For instance, some devices are based on ARM Cortex A9 with 32 bits architecture on which the long-term support of Julia works. In addition, as Julia is also very effective on parallelization (with the same code enhanced through macros), one can envision to use the same code as for high-level simulations on computation grid as real-time processing on embedded SDR.

2.1-5 Plotting tools and user interfaces

What it is: Julia language offers easy integration with a strong ecosystem of plots (with various backends such as ones from Python or GR) and the possibility to create Web applications with custom parameters (such as sliders, database exploration…). It means that a custom web app can be easily deployed based on the core calculation (which is computationally effective) and a limited number of external packages dedicated to web app porting (namely Interact.jl and Blink.jl).

Why it is important: For interactivity. Albeit not being as powerful and flexible as Gnuradio with the Gnuradio companion initiative, this kind of integration is a precious tool for prototyping. Indeed, it allows to have efficient and rapid tools for monitoring, exploring and debugging data. On the other hand, it also paves the way for simple yet functional demonstrators for dissemination or pedagogical purposes.

2.2 Introducing AbstractSDRs.jl package

We introduce here a common API to monitor several different SDR architectures. Each radio type is managed by its own sub-package and a master package (*AbstractSDRs.jl*) is dedicated to the gathering in a common interface. This kind of nested package architecture guarantee both flexibility (each sub-package can be independently modified) and extendibility (other packages can be added afterwards). The package is fully open source and the current version can be found here [Jul20]. In particular, the proposed approach is capable of monitoring, configure and transmit/receive samples:

— with Universal Software Radio Peripheral (USRP) from Ettus Research. These SDRs are immensely popular and several radios with various architecture (FPGA-based, ARM-based) can be monitored through the use of a special API: USRP Hardware Driver (UHD). In the proposed approach, the sub-package *UHDBindings* wraps the C API in order to use all functions defined in the low level interface. In *AbstractSDRs.jl* we propose two different binding options. The first one *UHDBindings.jl* wraps the C level API proposed by Ettus. It allows a simple yet powerful interface with Julia but it cannot leverage use of custom FPGA IP through RFNoC (see Appendix B). To address this, we have also incorporated the *RFNoc.jl* package that directly wraps the C++ code to be able to use instantiate crossbar graph [Bra16] and thus use custom FGPA blocks.

— with Analog Device Active Learning Module Pluto (ADALM-Pluto), an SDR proposed by Analog Device that uses a cortex A9 and the well-known AD9361 as the transceiver. The proposed sub-package *AdalmPluto.jl* uses some low-level C bindings of the driver supplied by Analog device (IIO library).

— with data exchange between a host computer and a remote computer (sub-package *SDROverNetwork.jl*). This interface requires a Julia session with *AbstractSDRs.jl* and is done through the use of ZeroMQ sockets in order to configure the radio from the host PC and send/receive the samples. This has two key advantages: i) allowing efficient use of SoC-based SDR with x86 or ARM (for instance the case the embed series of the USRP e.g. USRP e310/e320) ii) enforcing the scalability of the proposed approach with tree-based SDR network topology.

— with RTL-SDR dongles. Contrary to the other proposed packages, the *RTLSDR.jl* sub-package has not been proposed by the authors but has been incorporated into the master package.

All these different use-cases are encapsulated in a common API in *AbstractSDRs.jl* which can pave the way for an easy switch on different radios for prototyping. It is also to note that extension on other SDR support is quite straightforward thanks to the module encapsulation and MD.

2.2-1 AbstractSDRs syntax example

To better stress the prototyping through code concision, we present here a simple example written in Julia code that opens a radio, configure it and get samples. We also compute the square modulus of the FFT as the processing unit. The code example is depicted below:

```
using AbstractSDRs  # SDR integration
using FFTW          # FFT support
function main();
    # --- Radio parameter
    sdr = :uhd;     # Targeting USRP
    fc  = 2400e6;   # Carrier Frequency  [Hz]
    bw  = 16e6      # Bandwidth [Hz]
    g   = 30;       # Gain [dB]
    N   = 1024;     # Packet size [Samples]
    # --- Opening radio
    radio = openSDR(sdr,fc,bw,g
                    ;args="addr=192.168.10.13");
    # --- Loop on getting samples and processing
    try
        while(true)
            y = recv(radio,N);  # Get samples
            z = abs2.(fft(y));  # Computation
        end
    catch exception
        # Waiting for <ctrl-c> from user
        @info "Getting interruption";
    end
    # --- Close the radio
    close(radio);
end
```

Several important remarks can be made. The specific SDR targeting is done through the symbol `:sdr`. Then, when instantiating the SDR, `;` is used to add special keyword arguments.

These arguments are optional and can be used for specific parameters associated to the SDR (e.g FPGA bitstream path, radio IP address,...). Third, processing part leads to extreme code readability while ensuring very good runtime performance (see Section 5-2.3). Finally, note that the code proposed in the example is the more readable but not the more efficient as there are allocations at every loop (in buffer and FFT).

Several optimizations can be done when calculating the square modulus as described in next Section. It is preferable to pre-allocate processing array such as the input radio buffer and uses `recv!(buffer,radio)` with FFT pre-allocation (using FFT plans [Fri05]). It means that writing high-level code in Julia may not directly lead to high performance. However, code optimization (i.e., to increase throughput) can be directly done in Julia language (through some well-known performance tricks such as pre-allocation, unrolling loops and SIMD use [Mit13] with Julia macros), making transition from prototyping to high performance fast and straightforward.

Finally, the proposed ecosystem has been extended with application-oriented packages, related to spectral analysis (with *AbstractSDRsSpectrum.jl*) or FM radio receiver (with *AbstractSDRsFMReceiver.jl*).

2.3 Benchmarks and performance assessment

In this section we propose some performance evaluation using the proposed approach and compare it to other classical approaches used in the literature.

2.3-1 Benchmark properties

For this we choose to compare the performance offered with the Julia-based *AbstractSDRs.jl* package against C++ and Python. For a fair comparison, the following important statements have to be done:

— Performance is compared in terms of output rate after processing. For every language, the number of samples processed in a given amount of time is counted to deduce the rate (or throughput). This obtained rate is a function of the incoming rate from the SDR and in an ideal case one should obtain exactly the same rate at the output of the SDR and the output of the processing block.

— The proposed approach has been tested with the use of an SDR X310 from Ettus Research as it allows the maximal instantaneous bandwidth of 200 MHz. This SDR offers at max

200 MS/s which is high enough to discriminate the different approaches. Indeed, with the use of narrowband SDR (such as RTL-SDR) the performance difference can only be seen with extensive computational overhead which is not the case of the proposed micro-benchmarks.

- Performance is compared with C++ (low-level high-performance language) and Python (high-level language with concise semantic). Note that we have not added the Gnuradio approach as the processing blocks should be written in C++ (using SWIG) nor the Cython approach as they fall into the *two-language* problem we have stated beforehand.
- All the codes have been evaluated using one thread for both sample acquisition and processing, and the `-O3` flag for compilation (C++ and Julia). We also have used different optimizations for C++, Python and Julia described afterwards.
- Rate performance is achieved by Monte Carlo simulations with 20 independent runs of 10 seconds (more runs would not affect the result). All the code used for the benchmark (in all languages) and versions associated to the used modules can be found here [Lav20c].
- Sliding average of the square modulus of the FFT is considered as the processing. The sliding window is rectangular with 16 samples. For the three languages, the FFT is computed with the use of FFTW with the same compilation flags (and same output rate). Difference between the languages lies in the implementation of the square absolute, how the sliding mean is implemented and the memory management (where each run has to process roughly 30 GB of IQ stream due to the large bandwidth benchmarked).

2.3-2 Code versions definition

We define four versions of the code, the initial one L0 and three optimization levels (namely L1, L2 and L3). These three optimizations levels are independent and will be sequentially applied:

- L0 corresponds to the rapid prototyping (e.g the minimal code version that allows a proper processing chain). Note that it approximately represents 100 lines in Julia, 200 in Python and 380 in C++.
- L1 corresponds to L0 with algorithmic optimizations namely by removing boundary checks in `for` loops, using buffer pre-allocations and expliciting `for` loops for square absolute and sliding average.
- L2 corresponds to L1 with optimization on memory side and corresponds to use of low-level containers for buffers (i.e., continuous indexing arrays).

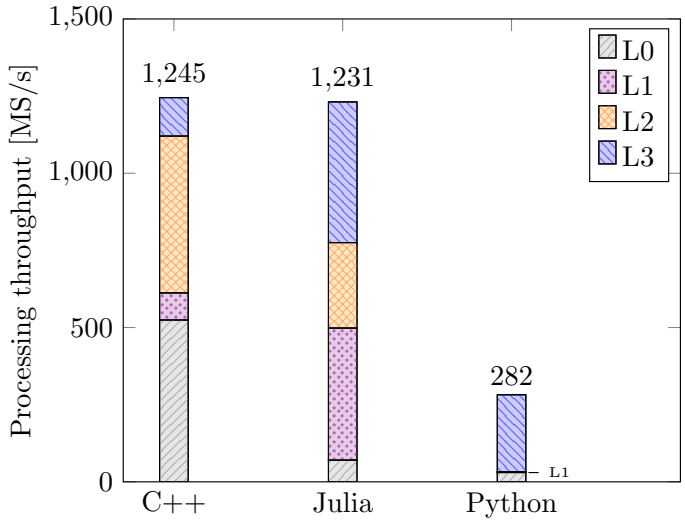


Figure 5-3 – Benchmark of the throughput of various optimization levels and languages.

— L3 corresponds to L2 with optimization on processor instructions and corresponds to the use of SSE and AVX vectorial instructions [Kar11] on the two `for` loops (absolute value and sliding average).

As additional notes, L2 cannot be fully applied on Python and L3 vectorization has been incorporated through the use of JIT via Numba.

2.3-3 Comparison between versions

We evaluate the benefits of the proposed versions in Fig. 5-3. In this figure, no radio has been used in order to point out the maximal achievable rate.

Regarding C++, the main gain has been achieved with the use of static arrays (i.e., L2). The L3 vectorization can help to reach the maximal rate of 1.245 GS/s. It is also clear that the main benefit obtained with Python is based on the JIT engine (i.e., L3). Regarding Julia, all the optimization tricks equally help to increase the throughput. The optimized rate in Julia (1.231 GS/s) is comparable to C++ but with less development time to L0 and easier application of optimizations (i.e., from L0 to L3).

2.3-4 Benchmark with X310

We now include the SDR and we depict in Fig. 5-4 the performance obtained with the use of the X310. The output rate (i.e., after processing) in samples per second is evaluated versus

the rate provided by the SDR which explains the lower rates encountered compared to Fig. 5-3. Worst performance is obtained by Python L0 code as expected. Julia without any optimization performs better than Python (with a very similar syntax) but is far from C++. Without any optimization, the three languages do not reach the 200 MS/s limit. When they come to the highest optimization level, both C++ and Julia achieves the maximal rate. It is not the case of Python (albeit being JIT compiled).

Note again that higher performance can be achieved in Python using double language (Swig or Cython) but it is not the purpose of the experiment done here and would largely increase the time to development.

As a side note also, concerning specifically the Julia way, it means that there is no visible penalty induced by the use of our bindings as predicted.

Finally, albeit writing high-level code in Julia may not directly lead to high performance, code optimization (i.e., to increase throughput) can be directly done in Julia language with the use of macros. It makes rapid and straightforward the transition from prototyping to high performance allowing the language to address real-time high bandwidth processing.

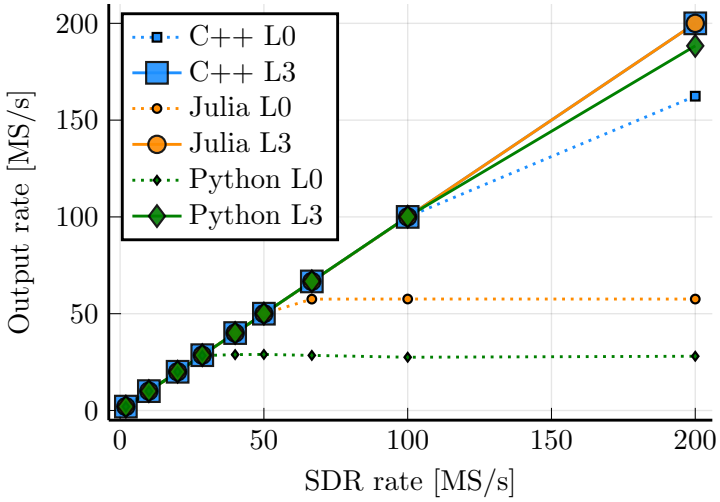


Figure 5-4 – Benchmark for initial code and highest optimization level code using X310 device.

3 FFT-BE algorithm implementation

The FFT-BE is a version of our interception algorithm applied to real-time processing and designed to run at a high throughput while using realistic hardware resources (i.e., a standard computer and its hardware accelerator). The purpose of the end of this chapter is to highlight the constraints encountered during the development and how they were overcome in order to reach the actual design.

3.1 Limitation of full GPP implementation

The choice of language has a major impact on the performance but no matter what the language is, it will necessarily be limited by the hardware on which it is executed. Although the Julia program can run the core (the most computationally intensive part) of the FH detection process at more than 1 GS/s, as seen in Fig. 5-3, the maximum throughput is limited to 200 MS/s due to the maximum sampling rate of the radio DAC.

However, the performance can be altered by other components in the flow and creates what is called a bottleneck (i.e., the speed of the chain is restricted by its slowest element). It must therefore be considered a series of additional components that must support the throughput such as: Ethernet devices, operating systems, Ethernet stack or data integrity checks performed between exchanges…

Although it is possible to achieve the processing throughput of the radio using a soft-only chain with Julia (the reception of IQ and their packaging in an Ethernet stream is done by an FPGA), performance drops may occur unexpectedly due to other programs running on the same host which may prevent the processing of some samples. It also uses large part of the CPU capacities which *de facto* reduces the capacities to derive intensive complementary red signal decoding (processing of the recovered red information).

From a security point of view, it has already been observed that an attacker has infiltrated a GPP based system, to tamper its operation. Succeeding in infiltrating an SDR based communication network would allow an attacker to get all the information that passes through it. Tampering an FPGA based system is all the more complicated because firstly, people capable of developing on this kind of hardware are less frequent than those capable of developing on a GPP. Furthermore, it requires to redesign the FPGA by adding new functionalities (data exfiltration, metadata sending …) without altering the primary functions, which is extremely

complex [Mat20]. Therefore it makes sense to use an FPGA for security audit material because it is more complicated to tamper with its functioning.

For all these reasons, it was chosen to use a mixed design, namely to use an FPGA embedded in an SDR to perform the core processing (FH detection) and let a GPP handle the analysis of red data (less demanding in resources and throughput). Moreover, the detection of FH channels is independent of the hidden red data, while the red data can be of different types (and so the processing of red data). There is a need for flexibility only on the red data analysis part, the FH channel interception can be fixed in a FPGA. This is supported by an FPGA-based detection and a GPP-based analysis architecture.

3.2 Choice of hardware SDR

During this thesis two SDRs were studied in detail, the E310 and X310 from Ettus, both of which work with UHD and a FPGA for a high throughput. A table listing their main characteristics is given in Table 5-2. The first SDR studied is the E310, it is an SDR embedding an FPGA and a dual core ARM processor for processing. It has the particularity to be very compact, hand-sized and uses little power (typically 2-6 watts). However, its small size is a reflection of its processing capacity, which can be limited for some applications. The detection of FH signals requires the largest possible bandwidth and therefore requires a lot of resources. The components of the radio are organized in a linear way: the radio front end is connected to the FPGA which is connected to the GPP which is connected to the external world through an Ethernet link. The E310 also has an internal bandwidth limitation, indeed the radio can acquire 56 MHz of bandwidth but cannot stream this bandwidth in its full extent to the GPP because the maximum throughput between the two devices is 6 MS/s. This makes it mandatory to process the data in the FPGA or to reduce the bandwidth in order to perform processing only on the GPP. The development of applications is then more complex with this bottleneck because a classic design path for SDR is to perform all processing in soft (GPP or DSP) and if the performance is not satisfactory, the critical parts are shifted into accelerators.

The final choice of radio is the X310, mainly due to its higher bandwidth. This radio embeds a bigger FPGA but no GPP, this last one is realized by using the processor of the host computer (which is connected to the SDR through Ethernet). The radio is capable of acquiring twice 160 MHz bandwidth (160 MHz per radio front end, but sampled at 160 MHz due to DAC) and sending them to the host. Due to the lack of equipment capable of receiving two 10 GbE links, the different tests will only be carried out with one radio front end and therefore 160 MHz

	E310	X310
Processing device	Zynq 7020 (FPGA + GPP)	Kintex7-410T (FPGA)
DAC/ADC	AD 9361	UBX 160
Frequency range	70 MHz - 6 GHz	10 MHz - 6 GHz
Internal IQ bandwidth	56 MHz	$4x200$ MHz
Exiting radio sample rate	6 MS/s	400 MS/s
Exiting radio link	1 GbE	2 * 10 GbE PciE 4 lines
Rx/Tx interface	2x2	2x2
Size	Hand size	Rack size

Table 5-2 – Comparison between used radios.

band (which is 26 times more than with an E310). This band is sufficient to listen to Bluetooth signals, which represent an easy-to-acquire FH system that can have internal compromises (presence of red data) due to their low-cost manufacturing. Moreover, the FPGA is large enough to contain our implementation of the FFT-BE which will be described more precisely in the Section 5-3.4.

3.3 Note on Ettus radio ecosystem

The Ettus radio comes with UHD [labb] and RF Network-on-Chip (RFNoC) [laba] which provide respectively an abstraction layer at software and FPGA levels. UHD is a hardware driver library used in order to address several SDR families with different configurations using the same function, through an API. It is possible to use UHD in standalone (in C++) or with other applications such as GNU Radio,, Labview or Simulink and even Julia (with AbstractSDRs). It provides an abstraction layer between the host and the radio by interfacing directly with its integrated FPGA (or GPP if present like on the E310) to control the gain, amplitude, center frequency, sample rate as well as establishing uplink and downlink stream to exchange data. UHD also supports control and management messages such as loss of samples due to slow processing by SDR or host, IQ timestamp and integrity control.

RFNoC is a network-distributed heterogeneous processing tool, it enables simplified management of processing within an FPGA, as well as a layer for communication with internal DACs and ADCs and a communication layer to move data in and out of the FPGA. It allows the exchange of data streams between multiple FPGAs and GPPs and host computers in a

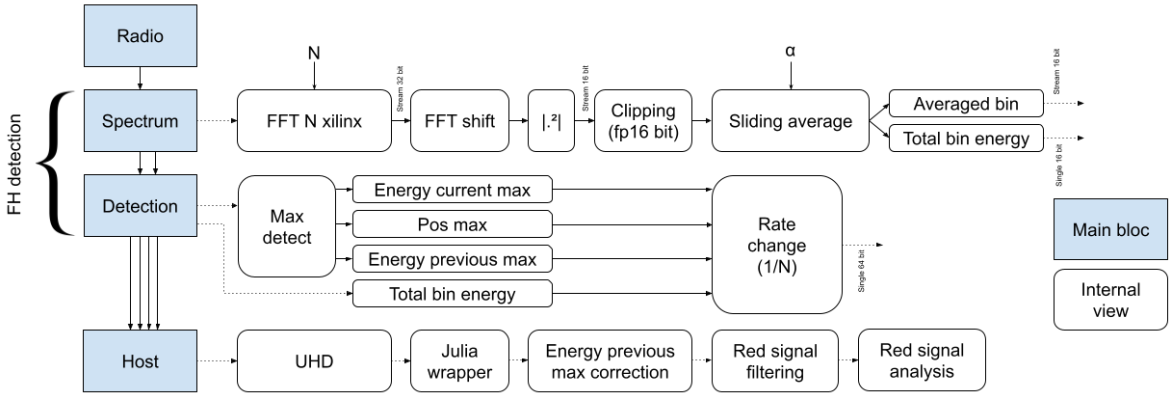


Figure 5-5 – FFT-BE FPGA processing blocks.

transparent way, i.e., without the need to code a communication layer. It establishes in an FPGA a bus-centric architecture between several processing blocks that it can connect on the fly (without requiring resynthesis of the FPGA). More explanation on RFNoC is shown in Appendix B.

The purpose of UHD and RFNoC is to decrease the design time for SDR by defining a workflow where developers only have to add their specific processing blocks, all other flow management and configuration is already done. For example, an FPGA processing block can be controlled directly from the widely used GNU Radio [Val08] without the user needing to do any further development apart from connecting his processing block to RFNoC.

3.4 Hard/soft partitioning

The operation of the FFT-BE is similar to the one presented in Section 3-3.3-2 but has only some additional functionalities and is designed to run partly on FPGA and partly on a GPP using the SDR host computer. Indeed, the channel detection and extraction parts are realized on the FPGA but the red data extraction part is realized on the GPP. As this last one does not require important resources (the bandwidth has been greatly reduced) and it is easier to listen to the result of an audio signal extraction on a device with an audio output without to synthesize an audio interface.

The different blocks can be separated into four areas: *Radio*, *Spectum*, *Detection*, *Host*, and their details are shown in Fig. 5-5 and will be further developed hereafter.

3.4-1 Radio

This part corresponds to the management of the ADC and is managed by RFNoC. The IQs coming out of this block are under 16 bits (8 bits per channel I and Q) and are grouped by packets of 1024. This is the maximum packet size authorized by the AXI bus (see appendix B) with a power of 2. All the processing will therefore be carried out using 1024 samples at a time.

3.4-2 Spectrum

This part corresponds to the spectrum estimation (with fixed point processing) and its shaping for the channel detection. The N-FFT is performed with the Xilinx IP FFT with rectangular window in double precision 16 bits. The two parts of the spectrum are rearranged to have contiguous frequency bins, then the square modulus is performed followed by a clipping to have single precision data. A slight (<10) sliding averaging over the spectrum bins is performed to remove possible spurious and digital noise. The output of *Spectrum* is composed of the N averaged points and their sum. This sum is not really used for FH detection, but is used to evaluate the spectrum utilization. If many signals are present, the sum of energies will be high which is an indicator of potential interferer.

3.4-3 Detection

The detection of the signal is done exactly as defined in Section 3-3.3-2, by taking the max argument of the averaged bin and returning the bin index. It also returns the energy of the bin associated to the previous detected channel (of the current processing samples). This can be used in post processing (on GPP) in case of detection error (due to a noise burst or channel hop).

This corrects a *posteriori* errors which can be of two kinds as described in Fig. 5-6 where:

— It can repair an impulse error, as it can be see at S_1 where the detector has locally mistaken of channel (outputting 11 instead of 10) but the system also returning the index detected at the previous iteration (S_{1-1}), at the instant S_1, the content of bin 11 and 10 will be returned. The channel correction (i.e., choosing to take the index 10 instead of 11, is done by seeing that the channel index suddenly changes and then returns to the original quickly, but can also be done by seeing that there is no red data detected (if of course there is any).

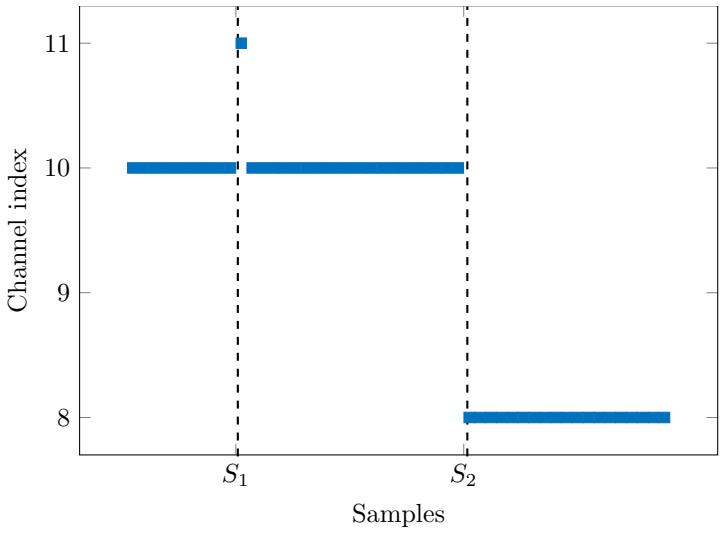


Figure 5-6 – Double output error correction scenario.

— The double output also allows not to lose information during a hop when the latter happens during an observation window (a batch of N_{FFT} samples). As it can be seen at S_2, the new channel 8 is detected but as a historic of 1 channel is kept, we also have the energy of 10 in output, the red signal reconstruction will then merge the data extracted from 8 and 10 in order not to lose data. It is considered that the signal extracted from a non-active channel is assimilated to a white noise and thus has little impact on a good audio reconstruction.

A rate change block which concatenates the four values to send them back to the host is placed at the end. This block does not modify the data, it is for the flow management with RFNoC in particular to inform the transition from a block of N samples to 2 samples of 32 bits (each returned value is in simple precision under 16 bits). The block of *Spectrum* and *Detection* being placed in the same IP block, the verification of the flow is done only at the output.

3.4-4 Host

The final step is to send back the data to host for red data processing. The delivery of data is carried out by UHD. The data of the received channels are at baseband rate. In our application case, the red signal is an audio signal, its extraction involves a straightforward decimation with filtering. A *posteriori* corrections are also performed to improve the extraction performance using the extracted channel "Energy previous max". A final analysis is carried out which varies according to the case: a SBOS can be carried out if the original red signal is

known or a more traditional analysis with the spectrum analysis and its audio playback can be used instead.

3.4-5 FPGA silicium utilization

This last section deals with the use of the FPGA resources of our interception system, and especially the resources needed for each of the elements constituting our design. A representation of the elements with their location on the FPGA grid can be seen in Fig. 5-7. Several elements are visible and are detailed in the following list:

- *Control & Radio 1 & Radio 2*: Radio front end
- *FH detect*: *Detection* block of Fig. 5-5, the red *AXI Wrapper* inside represents the handling part to connect to the bus-centric interface
- *SFPP*: 1 GbE and 10 GbE management interface
- *FFT & FFT Extra*: *Spectrum* block of Fig. 5-5, the red *AXI Wrapper* inside represents the handling part to connect to the bus-centric interface
- *Xbar*: Bus-centric interface
- *Ram*: Various memory used by the design blocks
- *DDC*: Digital Down Converter
- *DMA FIFO*: Fast buffer management for interconnection with the XBar
- *UHD*: UHD and RFNoC management

It should be noted that in our design the *Radio 1* and *DDC* zones are not used. We can see in this figure that the processing for the interception of FH signals represents only a small part of the resources (and that a quarter of these resources are used for interconnection). The radio front end and the management of the bus-centric RNoC take up each one a quarter of the design. We can also note that there is still space to insert a second batch of *Spectrum* and *Detection* blocks, so it is feasible to use the second radio front end to intercept another part of the radio spectrum at the same time.

A more detailed view of the resources used by the FH processing can be seen in Table 5-3. The *Total FPGA* expresses the available resources in the FPGA, *Total design* the used resources by the whole design, *Percentage in FPGA* the design utilization in percentage of the available resources, *Detection* and *Spectrum* the resources used by these two processings and *Percentage in design* expresses the block utilization as percentage of the resources used by the whole design.

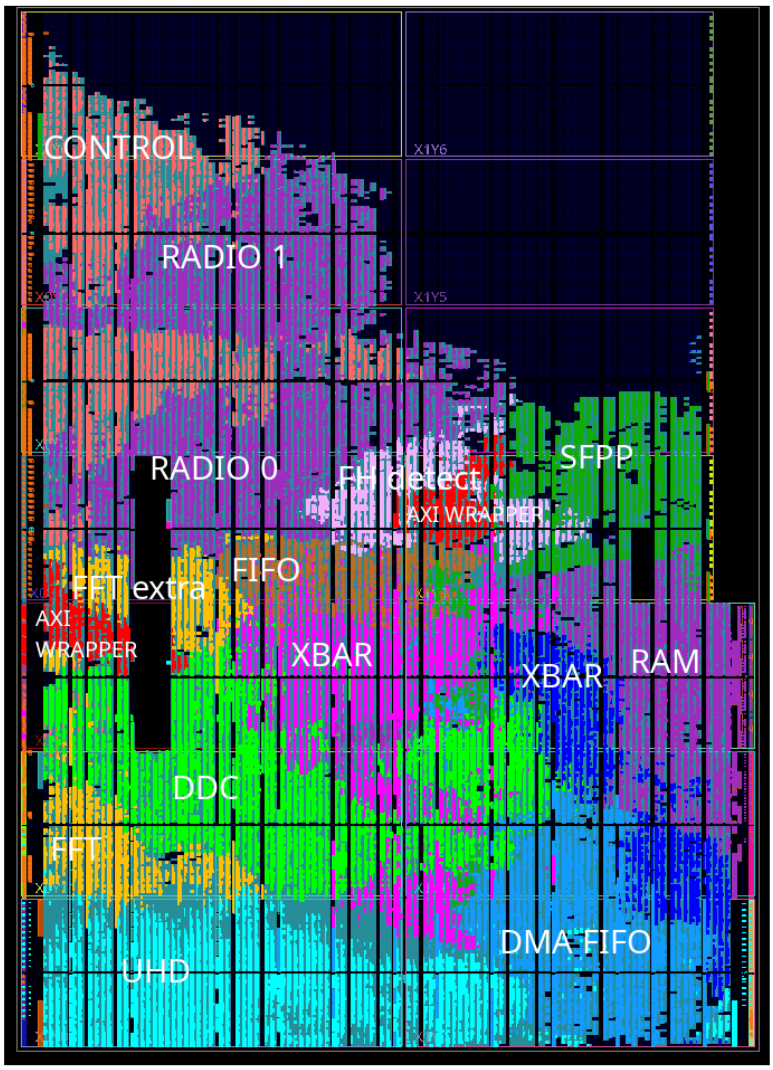


Figure 5-7 – FFT-BE FPGA utilization.

It can be seen that although being the core processing part of the FPGA design, our two processing blocks do not represent the largest part of the design, and that the overlay provided by Ettus with UHD and RFNoC as well as the interconnects take up the majority of the design.

	Total LUT	Logic LUT	Memory LUT	FLOP LATCH	RAMB36	RAMB18	DSP48 Blocks	Logic cell
Total FPGA	254,200	254,200	90,600	508,400	795	1,590	1,540	406,720
Total Design	111,582	95,249	16,333	164,952	314	55	245	242,610
Percentage in FPGA	43.90%	37.47%	18.03%	32.45%	42.96%	3.46%	15.91%	59.65%
Detection	7,732	7,327	405	22,871	8	0	0	35,766
Percentage in design	6.93%	7.69%	2.48%	13.87%	2.55%	0%	0%	14.74%
Spectrum	5,823	4,253	1,570	10,726	12	11	52	21,047
Percentage in design	5.22%	4.47%	9.61%	6.50%	3.82%	20%	21.22%	8.68%

Table 5-3 – Resource utilization.

It is therefore important to take into account during the design phase that not all resources will be available for user applications. It is also worth to replace these imposed blocks by customized versions, reducing the available functionality to limit the resource use. For some very high-performance applications that require a lot of resources it is worthwhile to override UHD and RFNoC with custom systems, but this is not the case in this thesis.

4 Conclusions

SDR open up the possibility of creating new radio systems faster and at a lower cost, and offers the possibility of upgrading existing systems. It also provides new opportunities for intelligent radio systems able to adapt to their environment. The processing carried out within an SDR must respond to flow constraints that require a preliminary study.

A presentation of a new methodology for efficient SDR prototyping based on Julia language has been done. Julia offers key properties that make this language appealing for SDR, namely high runtime performance, easy portability, strong scalability and convenient interactivity. These strong assets pave the way for efficient prototyping with SDR, addressing the so called *two-language problem* (rewriting high-level code in low-level language for performance). The benefits of the approach (language and ecosystem) have been proven by use of benchmarks that compare the rate performance with the ones of C++ and Python. It means that rapid prototyping of very large bandwidth and extensive computational processing can be envisioned in real-time and through concise code even in embedded SDR thus bringing down the classic *two-language* barrier encountered in prototyping.

An ideal SDR will only rely on the use of a GPP, however these are not the most powerful in terms of throughput. Hardware accelerators can be used to overcome this shortcoming. This is illustrated by the implementation of the FFT-BE on a GPP which limits the throughput to 85 MS/S while a FPGA solution can go higher. The interception system being functional and validated in simulation, the next step is to validate the design and then perform real target listening.

Chapter **SIX**

TOWARD REAL EAVESDROPPING

Contents

This final chapter aims to concretize the FH signals interception system and the red signal recovery. The purpose is twofold: validate the hardware architecture described in Chapter 5 and show how BLE devices can lead to information compromising. These systems have indeed several characteristics which make their interception valuable:

— They are usually built with a SoC which, according to the literature in Chapter 2, carries a high risk of side-channel due to physical proximity between components.

— They feature an FH transceiver which does not emit continuously allowing exploring both time and frequency sporadicity.

— They are very widespread and several different models are available on the market.

The tests that will be performed are divided into three categories, i) the use of synthesized signals generated with specialized hardwares, aim to assess the proper functioning of the interception system in real-time and to confirm the simulation results of Chapter 4 by using the same input signals. ii) Altered devices are real systems where a leak has been forced by external means. These experiments aim to characterize the ability to detect a side-channel in a more realistic scenario. Special wired devices will be used to focus the study on signal going out of the transceivers only. iii) Unaltered devices are real systems where a leak occurs without being forced, corresponding to a real exploit where the device generates a side-channel of its own.

1 Side-channel leak metrics

As FH transmissions are considered, the correct channel must be extracted. CER is an appopriate metric to estimate the performance of our detection system. However this can only be done with *a priori* knowledge of the FH sequence, which will be unknown for BT devices. In the latter case, other metrics will be exploited.

In these experiments, audio signals are considered as red signals embedded in the side-channel. This way, the values can be complied with some measurements made in security audits, paving the way for appealing proof-of-concept. To objectively evaluate recovered audio signals, the used metric is the SBOS as in Section 4-3.1). A side-channel will be considered as active if the SBOS has a value greater than 0.1 as described in the Table 4-3. It should be noted that is it mandatory to have the original red signal to be able to estimate the SBOS.

The original red signal is accessible when forcing the leak, for instance by tampering the power supply with a specific pattern or by manually coupling the BT devices with an audio speaker that plays a known audio sequence.

In the case where the red signal is not fully accessible, the spectrum of the recovered signal will be used and if a peak is visible at the right frequency compromise will be considered. This estimation has a significant weakness in the case where the red signal is multi-tones i.e., a succession of tones at different frequencies. Because each tone will be visible only a fraction of the time and therefore the energy of the red signal will be distributed to several frequencies of the spectrum.

The equipment that is used in this Chapter is listed in Appendix C.

2 Level 1: Validation of hardware architecture

This section aims to validate the FPGA and host architecture of the FFT-BE (Section 3-3.3-2) and to compare its performance with simulations of Chapter 4 for identical scenarios.

2.1 Signal synthesized with a vector signal generator

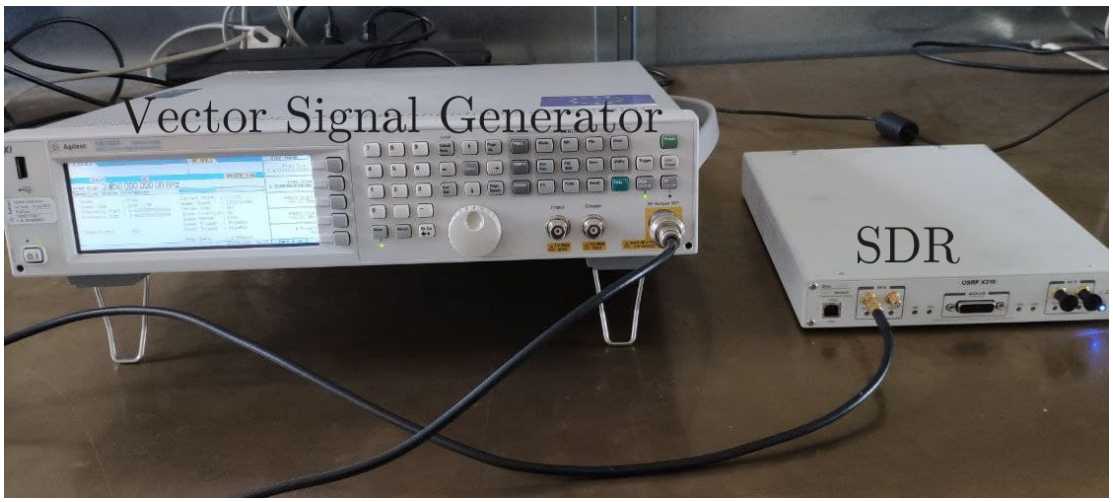


Figure 6-1 – Validation setup with VSG.

To ensure the proper functioning of the different parts of the interception system, a signal generator is used. A mixture model similar to the one exposed in Section 3-1 is used. For this proof of concept, signals are transmitted through cables to the SDR (which embeds the FFT-BE algorithm on the FPGA) from a Vector Signal Generator (VSG) which allows generating common signals and to apply modulation on them (the setup can be seen in Fig. 6-1). The signal generated follows a frequency sweep with a hop duration of 1 ms and channels width of 1 MHz. No additional noise is added, the VSG output power has been set to -20 dBm. On the radio side a 30 dB attenuator is used and the radio Rx gain has been set to 6 dB. The baseband message is a sine wave with an AM modulation ($h = 0.01$) of a one tone red signal at 1 kHz.

The results can be seen in Fig. 6-2 where a dotted line is marked for each hop. The four figures correspond to the four outputs of the algorithm:

— a) *Pos max* is the channel index observed which is located between 1 and N_{FFT} ($N_{FFT} = 1024$ in our design). In this figure the frequency steps are clearly visible.

— b) *Total bin energy* corresponds to the power of three received signals computed in the frequency domain. Since the generated signal has only one active channel at each moment, the total energy is very similar to the energy of the extracted channel. However, this signal cannot be used to perform red signal extraction it will be greatly impaired by interferers and noise in a non-controlled scenario.

— c) *Energy current max* represents the extracted channel i.e., the baseband signal at the current channel index. The evolution of this signal in time domain exhibits the red signal as no baseband modulation is present in this test.

— d) *Energy previous max* represents the baseband signal at the previous estimated channel index, its observation allows compensating the sync error and we can see that such an error occurs twice (at samples $2\,500$ and $6\,000$). Moreover, if this extracted channel is often non-empty, it is an indicator that the input signal is strongly noisy.

The *Energy current max* and *Energy previous max* can be post-processed together at the host side to reduce the loss of information introduced by a channel detection error and consequently improve the performance of red data recovery. This output is called the *corrected pos max output* (CPM) in the remaining of this chapter. In this case, the correction is negligible because our test signal is quite simple, and the curve of CPM is identical to Fig. 6-2-c).

The Fig. 6-3 presents the spectrum of the extracted red signal. The red signal is filtered at 5 kHz with an 8-FIR low pass filter to avoid aliasing. There is a strong energy at 1 kHz which indicates that the signal has been correctly intercepted and that no distortion has altered the red signal.

— These curves demonstrate that the interception system is functional and provides real-time processing of an FH signal at 200 MS/s with the recovery of a red signal.

— No flow integrity errors were detected, which confirms that all data are successfully processed at this rate.

— 200 MS/s is the highest speed achievable with the SDR used, the system could go at a higher speed on the same FPGA but we have no ADC available to test a higher speed.

Albeit being useful to validate the architecture and to be sure that the FH hops are correctly detected, using a VSG has several limitations on the channel detection part. Indeed, we are

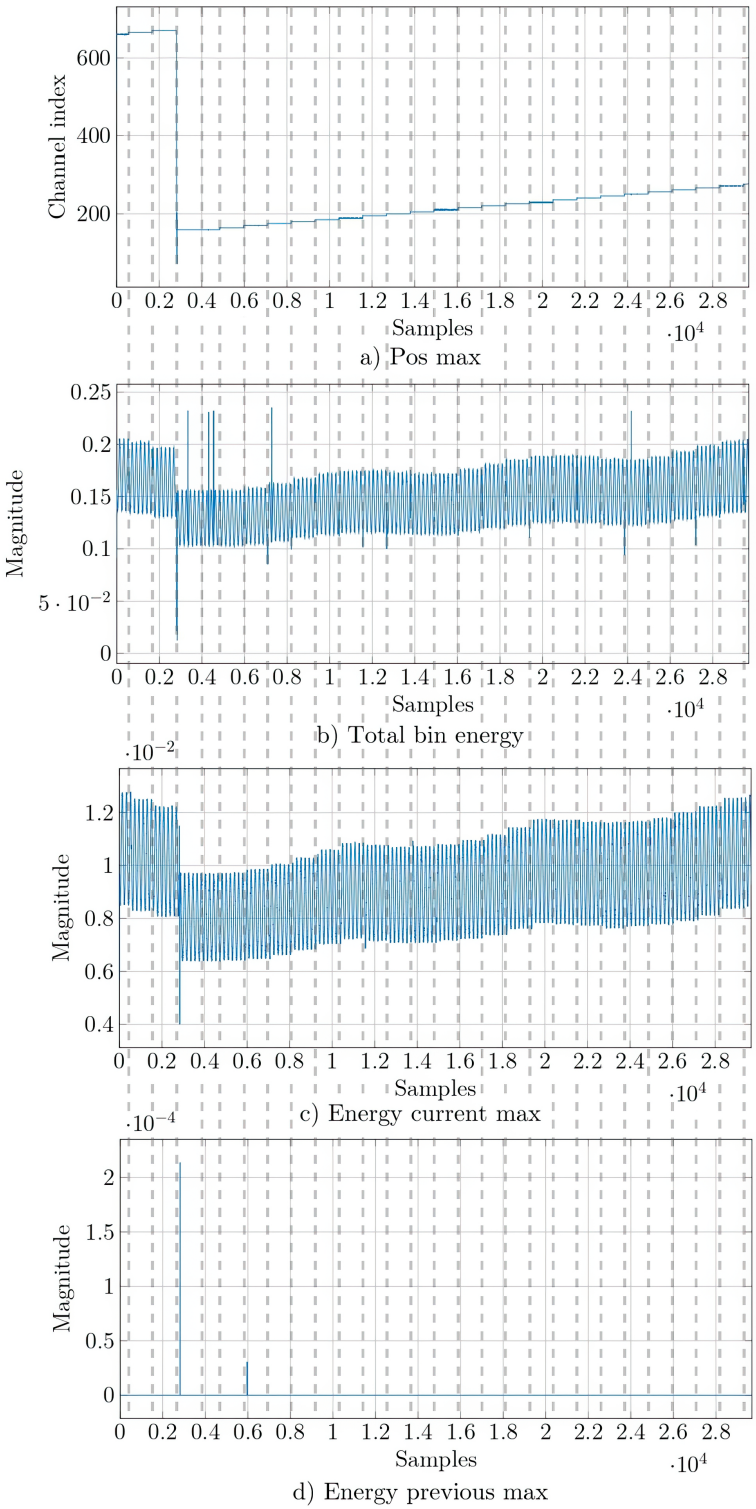


Figure 6-2 – FFT-BE CPM output for a frequency sweep.

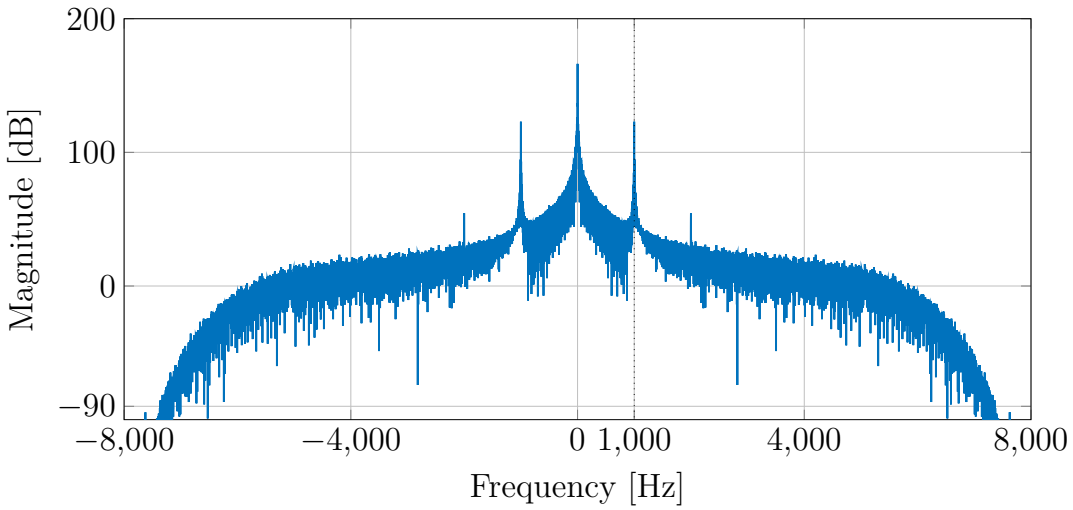


Figure 6-3 – FFT-BE CPM outputs of a 1 tone at 1 kHz.

not in presence of a random FH pattern but rather a system realizing a frequency sweep, meaning the following channel is therefore routinely the adjacent channel, this feature has a consequence on the convergence time of the algorithm and is not representative of a true FH signal (especially TRANSEC signals). Moreover, our synthesizer is not able to hop faster than 100 μs which is a limitation for TRANSEC use cases. It can also not perform a black signal modulation other than a sine wave. For these reasons, the experiments will be rather done in the next part by generating the IQ with a computer thanks to Julia and then will be transmitted by the VSG.

2.2 Toward more real like FH signals with Julia

Instead of sending to the SDR the IQ generated by the VSG, it is possible to create the IQ stream through Julia language in the same way as they were generated in simulation in Chapter 4. As the hop sequence and the red signal are known when creating the signals, it is also possible to use the metrics from Chapter 4 to compare the performance of the real system with those of the simulations.

There is no direct synchronization between the SDR receiving the signal and the VSG emitting it (unlike the simulations where they are synchronized with the exact sample), in order to use the CER and SBOS metrics, it is necessary to resynchronize the signals a posteriori. This synchronization is realized thanks to a correlation between the signals (the original and the received ones), the argmax index of this correlation estimates the delay to be applied to the

signals to be resynchronized. This method is efficient because the original signal is perfectly known and no degradation is applied to the signal. It should be noted that the following measurements use this synchronization and that errors induced by the synchronization may occur and lower the performance in addition to that of the interception system. In particular at low SNR, the synchronization will be less efficient and therefore will decrease the performance of the detection system.

The first test consists in ensuring that a correct channel is detected, for this we use the CER metric. The results of Fig. 6-4 are compared to the simulation of Fig. 4-8. The SNR variation has been realized by changing the VSG Tx gain. The signal has 40 channels that randomly hop. Due to the hardware limitation of the VSG, the bandwidth of the signal is reduced to 40 MHz, and the same λ values have been used as in simulation (and thus not the same T_s). The performance in the real experiments is very close to the simulations, although the different plateaus observed in experimentation are at higher CER value than in simulation. This difference may be due to the synchronization mechanism for CER computation which has an impact on the performance.

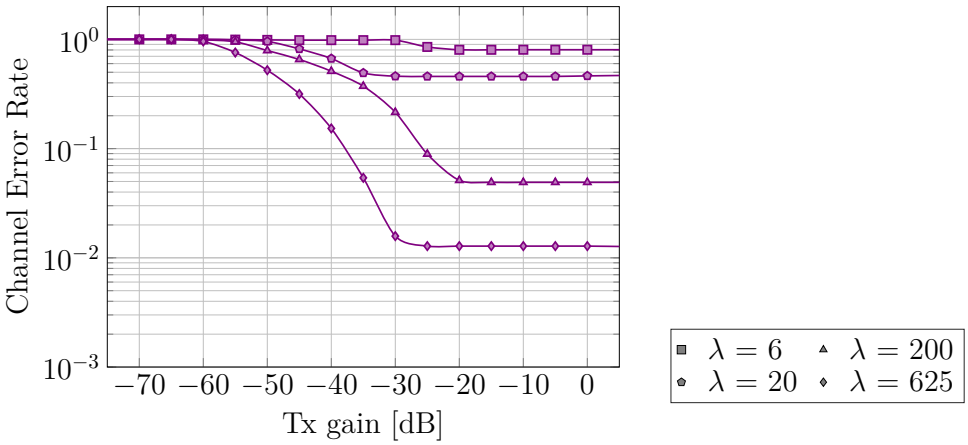


Figure 6-4 – CER vs Tx gain for various time slot duration.

The next experiments focus on the recovery of red signals, the latter is as for the simulation composed of three successive sine waves. The use case of the Fig. 6-5 and 6-6 is the TRANSEC as described in the Table 4-1 but reduced to a smaller bandwidth (same channel width and same λ). Fig. 6-5 (which can be put in comparison with its simulated counterpart Fig. 4-12) shows the SBOS for various modulation index h, and demonstrates identical results for both experiments and simulations. It should be noted that for this test, the modulation index h goes to lower values than in simulations.

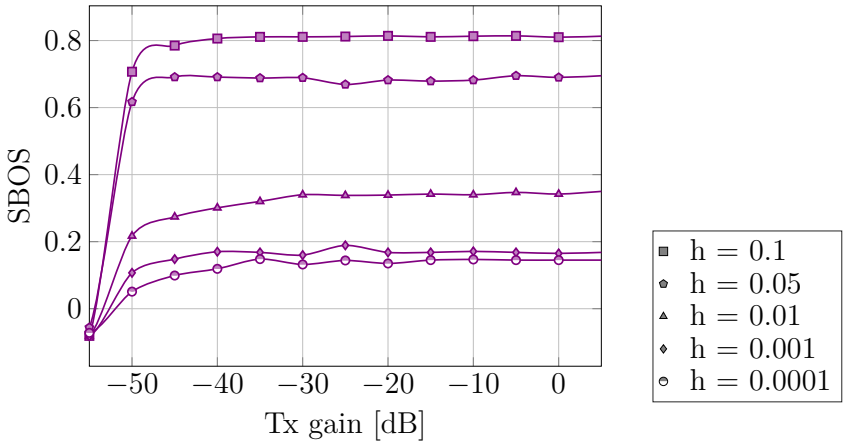


Figure 6-5 – SBOS with various h on TRANSEC case.

Unlike the two previous tests, when time sporadicity occurs the SBOS shows a result inferior by half to the simulation results (see Fig. 6-6 and Fig. 4-14), but this loss must be weighed against the synchronization used to compare the SBOS , which is not perfect and may cause errors.

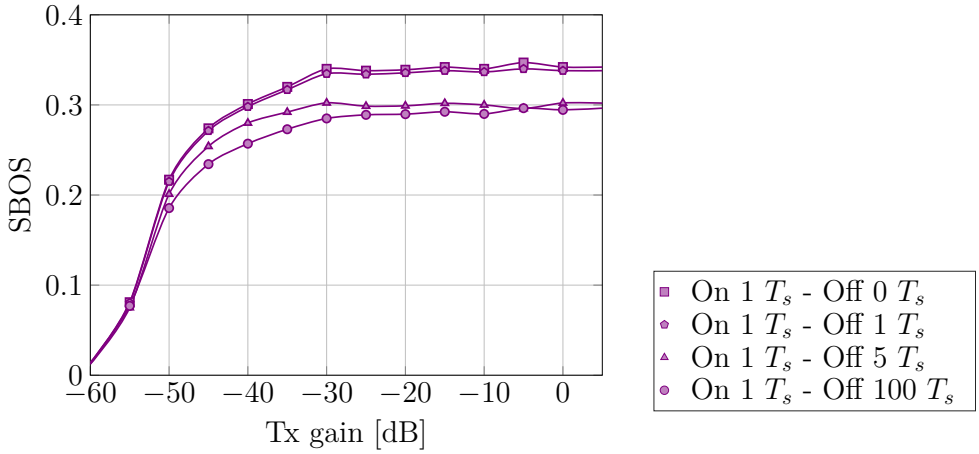


Figure 6-6 – SBOS with time sporadicity on TRANSEC case with $h = 0.1$.

The last test aims to check if our system is able to intercept a sporadic signal issued from an existing device. This device is a BLE dongle, working in beacon mode, which means that it will periodically emit a frame. The latter always has a duration of 1.7 ms and is repeated every 23 ms (*BLE beacon 1* case). A case where the sporadicity is more important is also tested, since the signal of 1.7 ms is emitted every 44.5 and 66 ms for *BLE beacon 2* and *BLE beacon 3* respectively. The samples complying with these timings are generated with Julia, a GFSK

modulation with random symbols was also used (the same as in BLE device). The results of this interception can be seen Fig. 6-7.

We find plateaus close to a SBOS of 0.8, which is the maximum ceiling of our red signal recovery constated in simulations. This result can be due to two phenomena, whether the fact that the red signal is emitted for a longer time (but with a longer time without signals) makes the decimation and filtering succeed in filling the holes generated by sporadicity, or that the synchronization mechanism works better with this type of sporadicity and that therefore a "good" signal is sent to the SBOS estimator contrary to the previous case.

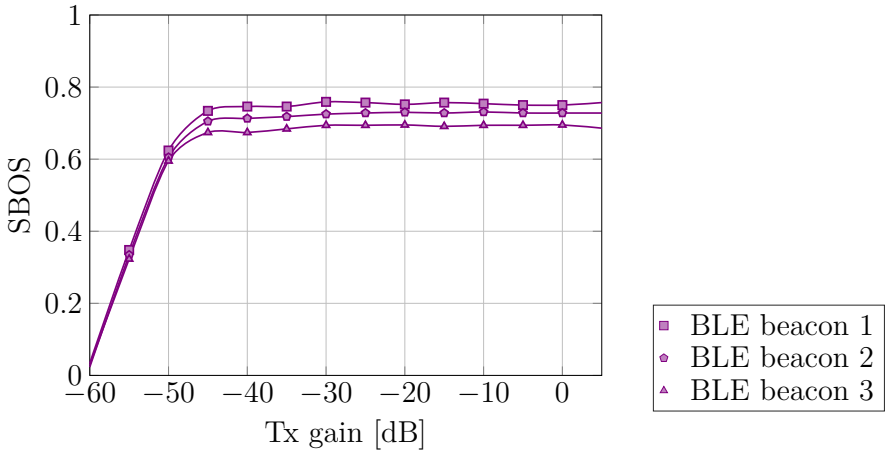


Figure 6-7 – SBOS with time sporadicity on BLE beacon case with $h = 0.1$.

- These tests prove that our interception system runs in real-time in an FPGA, the recovered values are close to the simulated data, which implies that no factor due to the implementation deteriorates the performance
- If a red audio signal is present in a BLE signal, our system is able to detect it. As a result, we can move to the next experimentation level which is not to generate the signals but to use real FH transceivers (BLE).

3 Level 2: Altered devices

The goal of the experiments in this section is to force leaks at different places in a SoC and to observe to what extent it is conceivable to induce a side-channel and if our interception system is able to recover it. The injection points have been chosen considering the different

Model	Payload duration	Silent duration	Periodicity
Nordic PCA10059	1.7 ms	21.3 ms	43 Hz
ESP32	1.7 ms	47.6 ms	22.3 Hz
Nordic nRF52832	1.7 ms	98.3 ms	10 Hz

Table 6-1 – BLE beacon mode timing.

side-channels observed in the Chapter 2. Two valuable locations have been chosen as described in Fig. 6-8: the radio transceiver (violet arrow) and the power supply (blue arrow).

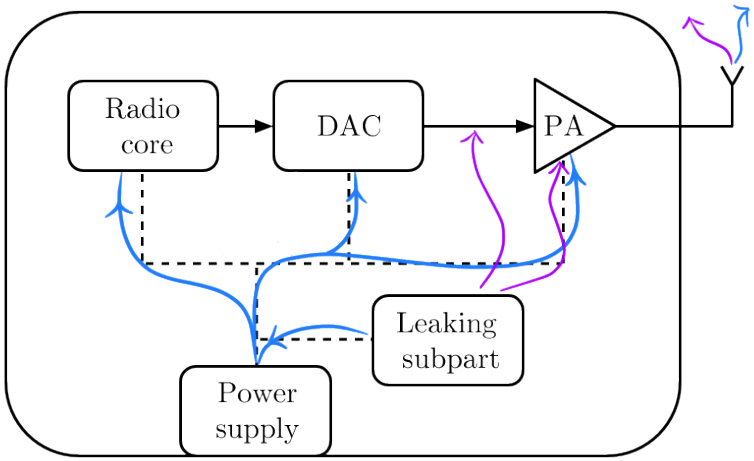


Figure 6-8 – Forced side-channel emplacement of level 2 in a BLE SoC. The arrows correspond to the tested compromise scenarios: the radio transceiver (violet arrow) and the power supply (blue arrow).

In this section we will use commercially available FH sources, namely BLE dongles, which will be set in BLE beacon mode, i.e., they will regularly emit a signal as described in Table 6-1. Some of them allow connecting to the SDR in a wired way (ESP32 and nRF52832) and others only in wireless (PCA10059). In this last case, the measurements will be realized in an anechoic chamber in order to only listen to the signal of the target. Using BLE dongle, it is no longer possible to know the hop sequence and thus no CER is available. It is still possible to know the red signal and thus a SBOS is available with a synchronization as described in the previous section.

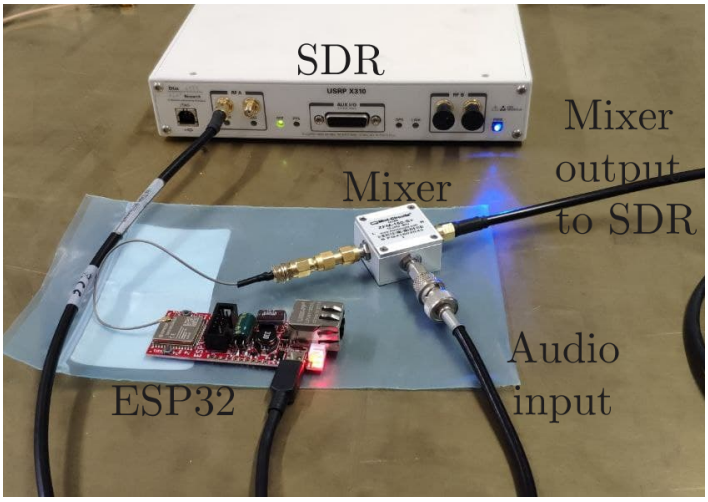


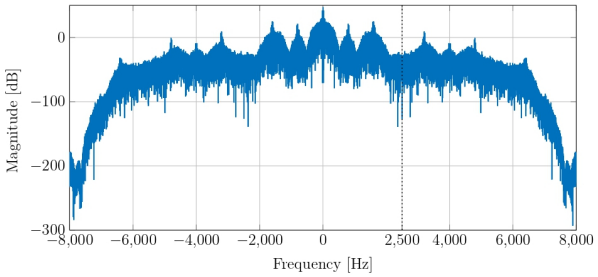
Figure 6-9 – Forced side-channels with mixers.

3.1 Analog mixer

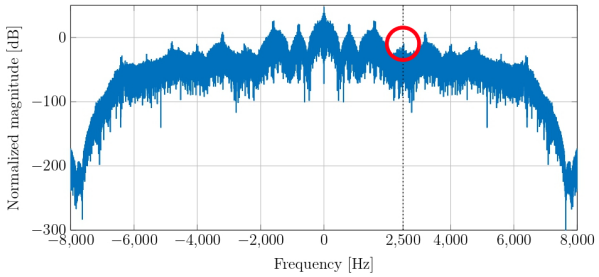
A side-channel in the transceiver implies that all the signals coming out of the transceiver are mixed with the red signal. This phenomenon (violet arrow of Fig. 6-8) can happen when a cross talk is made between the signal to be emitted (before the power amplification) and an internal signal (red signal) or when a sub-part of the SoC consumes enough energy to affect the global supply of the SoC. Since transceivers consume a lot of energy, the filtering of their power supply is limited in order to minimize the power losses and costs, so imperfections on the power supply will be immediately reflected on the emitted signal.

In this scenario the leak is forced by adding a mixer at the output of the transmitter (with a 1:1 mixing ratio). A red signal is then connected (audio comes from a smartphone DAC) at the second input of the mixer and the output is connected to the SDR. The setup can be seen in Fig. 6-9 for the ESP32 dongle (the wiring is identical for nRF52832). The injection of the audio signal is first a pure wave sine sent at a different frequency for each dongle. The choice of the frequency is to ease the leak observation by selecting frequency in areas where the spectrum is flat.

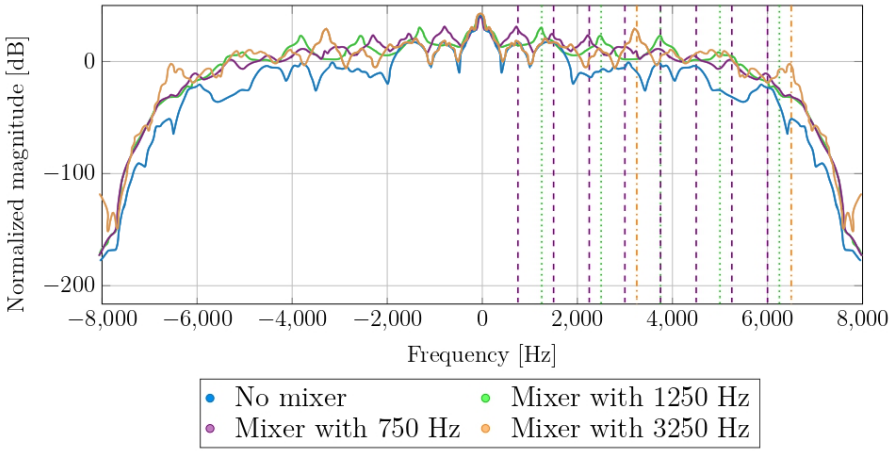
For both models of wired dongle, a capture of the low frequencies of the spectrum (0 to 8 kHz) is performed with and without audio signal injected in the mixer. Fig. 6-10 shows for both dongles, we observe the fundamental frequency of the audio signal injected and visible with a peak of 25 dB. However a different behavior is observed between the devices. Indeed, the nRF52832 capture contains harmonics of the audio signal while the ESP32 does not have



(a) ESP32 FFT-BE CPM output with no audio.



(b) ESP32 FFT-BE CPM output with red audio at 2.5 kHz.



(c) nRF52832 FFT-BE CPM output envelope with various red audio frequencies.

Figure 6-10 – FFT-BE CPM output on BLE dongle with mixer at transceiver output.

any. The output of the ESP32 is an output meant to be connected to an antenna whereas the nRF52832 output is specially designed for observation probes. This difference can be responsible for the different behavior.

In the low frequency spectrum of the ESP32 (shown in Fig. 6-11), harmonics of 22.3 Hz are visible and correspond to the periodicity of the transmissions made by the dongle, a similar observation is visible on the nRF52832 at 10 Hz.

In a second step, a multi-tone audio signal is used for SBOS evaluation. The results can be

Model	SBOS
ESP32	0.32
Nordic nRF52832	0.26

Table 6-2 – SBOS with mixer at transceiver output.

seen in Table 6-2. The values are much lower than those shown in Fig. 6-7 (the ESP32 case corresponds to *BLE beacon 2* case). This difference can be explained by the low amplitude of the injected red signal. Since the latter comes from a low power audio DAC, its amplitude is relatively low and not adapted to an impedance of $50\ \Omega$ like the telecom material. However, for both dongles the SBOS value is greater than 0.1 and therefore we can consider that there is a side-channel.

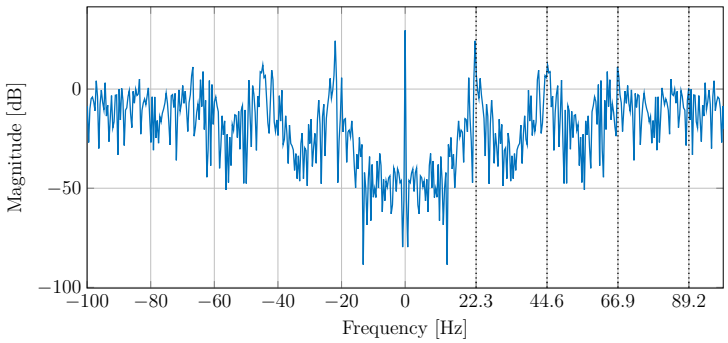


Figure 6-11 – Zoom on low frequencies of ESP32 baseband received spectrum.

Two conclusions can be drawn from those tests. i) we are able to force a side-channel and able to detect it if it affects the output of an audio transceiver. ii) with the same setup, both devices have a different behavior (presence or not of harmonics of the audio signal).

3.2 Power supply

In this test, the leak is not present at the output of the emitter but within its power supply. In other words, the power supply of the whole Bluetooth dongle is modulated by a red signal. It is a scenario that corresponds to the fact that in the SoC, the power supply is common to several subparts and that the different power spikes of a subpart will impact the power supply of all the other subparts of the chip. It corresponds to the blue arrow in Fig. 6-8.

The first idea to force such a side-channel was to power the dongle through a low frequency function generator whose output is a sine with an offset, which then allows injecting a compromise in the dongle power supply. However the used dongles are low power and their power supply filtering is efficient enough to erase the sine part of the power signal. The ripple of the input power was not visible on the power supply of the board after the filtering stage.

Therefore, a second approach to force a leak has been tested and corresponds to generate

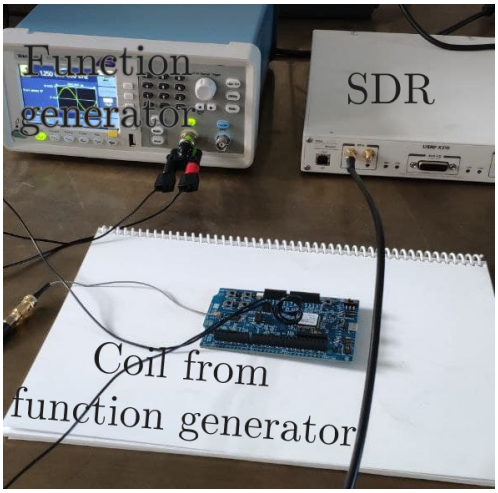


Figure 6-12 – Forced side-channel with a coil.

a leak as a subpart would do. For this purpose, a coil with an alternative signal (20Vpp at a frequency of 1250 Hz issued from a low frequency function generator with adapted impedance) has been placed above the board power supply to generate an electromagnetic disturbance that will affect the board. This setup can be seen in Fig. 6-12. The result of this test is given in Fig. 6-13a with no coil and in Fig. 6-13b with the coil radiating a red signal. We can see that the leakage is clearly present at 1250 Hz and has several harmonics.

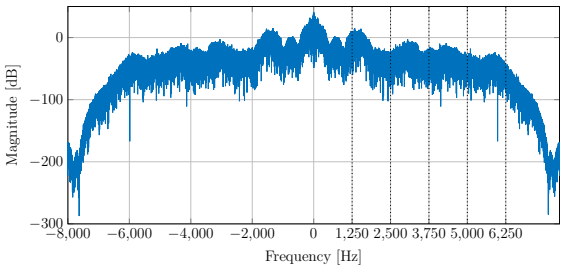
However, the reading of a periodogram on a large time acquisition can be quite cumbersome if the red signal is of lesser amplitude because it can be obscured in the noise. The spectrum previously issued is made from direct output of the FFT-BE and thus used all samples received (including those without red information). Considering the temporal sporadicity, a large part of the used samples corresponds to the time when there is no received signal, and therefore strong noise component is added to the measurements, and estimating the peak height is quite difficult and not accurate.

In order to improve the reading and the detection of the red signal, Algorithm 2 has been used. It is based on the instantaneous Welch spectral density estimation but using only the observation windows in which an FH signal is detected. In order to further improve the comparison when the leakage is intentionally generated, a spectrum difference is realized, i.e., the algorithm is executed on a signal where a compromise is supposed to be present (or at least forced) and on another one where the studied compromise is not supposed to exist. The two signal acquisitions are executed one after the other, with the same material in the same positions to ensure that only the difference due to the leakage appears on the figure and not

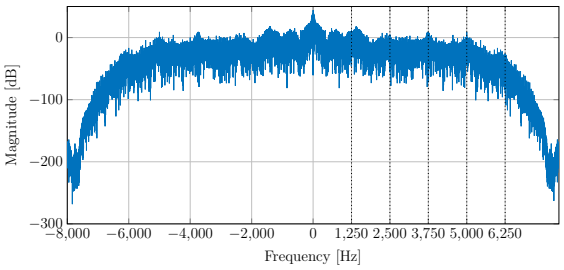
the influence of the environment or how the signal is observed.

Algorithm 2: Partial spectrogram in Julia language

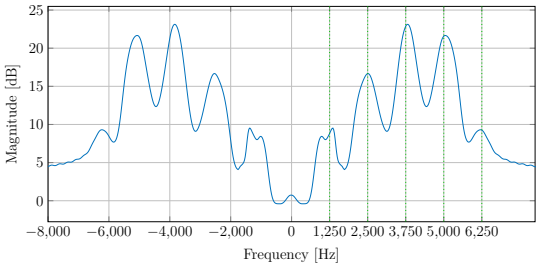
```
  ---- signal is the input data, S is the final spectrum
  ---- Init value, storage and apodisation window
1 N = 2048
2 S = zeros(N)
3 blackman = getBlackmanWindow(N)
  ---- Detect start index of all FH bursts
4 indexes = getBurstStarts(signal)
  for index ∈ indexes do
    ---- Isolate burst samples in a vector of N samples
5   burst = getBurst(index, signal)
6   bPadded = zeroPadding(burst,N)
    ---- Compute square modulus of FFT with a Blackman window
7   x = bPadded .* blackman
8   y = x |> fft |> fftshift .|> abs2 |> 1/N
    ---- Spectrum accumulation
9   S .= S .+ y
  end for
```



(a) nRF52832 CPM output with coil power off.



(b) nRF52832 CPM output with coil at 1.25 kHz.



(c) Difference between the coil power on and off with Algorithm 2.

Figure 6-13 – FFT-BE CPM output on nRF52832 dongle with a coil.

The result of this spectrum difference can be seen in Fig. 6-13c, it shows the difference

between a spectrum of a signal without forced side-channels and with forced side-channels, 0dB means that there is no difference between the signals, and above 0 there is energy added with the compromised signal. The height of a spike (from the 0) is therefore directly the amplitude of the leaks. In this case, the amplitude of the first spike is 8 dB and the one of the second spike is 16.5 dB. The compromising information would be more accessible to an attacker by using the first harmonic rather than the main frequency.

This second test gives insight into a new way of injecting a side-channel in a BLE dongle in a scenario more likely to happen in real life than with the mixer. The coil used could actually be a device placed next to a BLE dongle emitting an EM field similar to the coil, or the coil could simply be a nearby cable that runs very close to the dongle. The amplitude of the compromise was relatively high which leaves open the possibility that an accidental (and with therefore a lower amplitude) leakage could occur and still be detected with the Algorithm 2.
The tests of this section show that red signals can be recovered despite the presence of a real FH which is supposed to be there to prevent interception attempts, and that the red signal can be available at several harmonics which multiplies the possibility for an attacker to recover information. Indeed, it is possible to merge the different harmonics together in order to maximize the recovered information.

4 Level 3: Unaltered devices

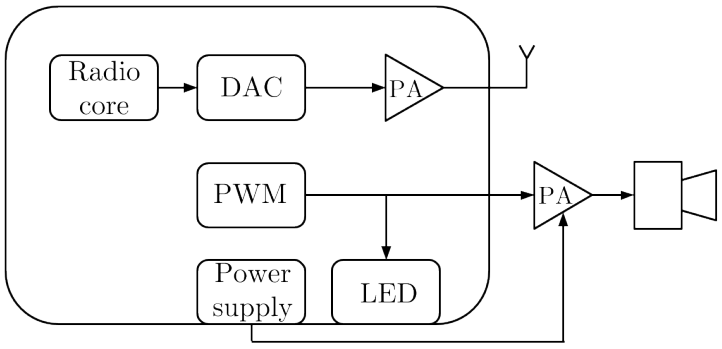
This last level focuses on BLE devices that are not modified and whose compromise has not been forced by an external mean. The leaks will come from the devices themselves due to their operating behavior. In order to detect a compromise, we have programmed the dongle to perform (in addition to their BLE beacon task) common tasks such as reading an audio stream, flashing LED or acquiring an analog signal and the outgoing BLE signal through our interception system has been observed. We will look if the characteristic frequency (red signal) of the tasks is present in the signal. For this we will make sure that these characteristic frequencies are 1000 or 1250 Hz. By using these two frequencies, we can ensure that the behavior is identical or not depending on the frequency but also if a frequency does not fall into a particular case that would influence its amplitude.

4.1 Speaker impact

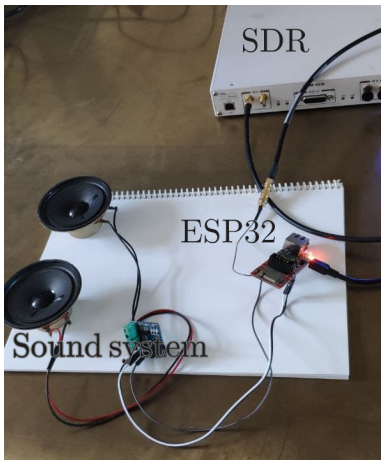
This experiment involves using an audio system (a) with a Bluetooth connection. Generally, in such a system, an external device (b) sends the audio stream to be played via Bluetooth. The system (a) then returns acknowledgements to confirm the correct reception of the audio data. The Bluetooth signal is sent periodically because the audio stream has a constant rate.

This acknowledgement is equivalent to our BLE device in beacon mode but no external device will be sending data to the BLE device. Thus, the audio data will be contained in the BLE dongle to study the compromises of a single device. The setup of this test is shown in Fig. 6-14. The dongle generates a PWM at 1250 Hz (for ESP32) and 1000 Hz (for nRF52832) to a class D audio amplifier, this last one is then connected to two 3W speakers. The nRF52832 has its PWM output sent to both the audio amplifier and the LED to provide an indicator of operation.

In this scenario several side-channels can occur, i) the PMW signal can leak and impact the transceiver, ii) the amplifier can create power supply disturbances that can impact the transceiver or iii) the loudspeaker coils create an electromagnetic field that can disturb the internal signals.



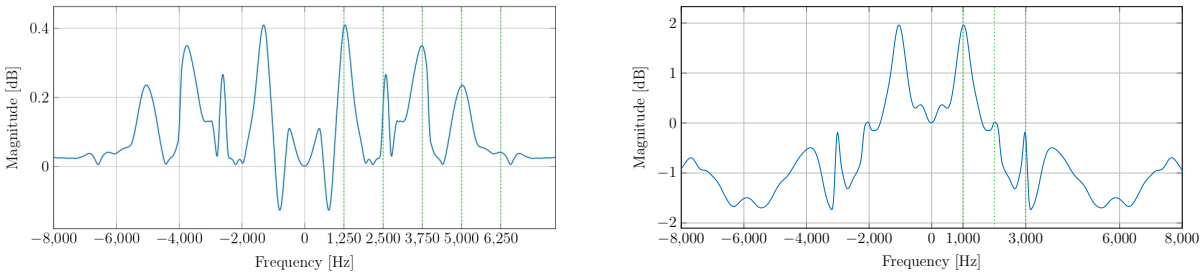
(a) Schematics of nRF52832 setup.



(b) Setup of ESP32 with a sound system subpart.

Figure 6-14 – Setup for side-channel checking issued from a sound system subpart.

The results of this test are given for the ESP32 and the nRF52832 in Fig. 6-15. The two models have once again a different behavior : the amplitude of the red signals are different:



(a) ESP32.

(b) nRF52832.

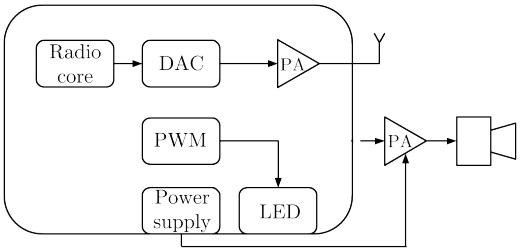
Figure 6-15 – Difference of FFT-BE CPM output between with and without PWM.

0.4 dB and 2 dB respectively for the ESP32 and nRF52832 (while in the case of the mixer the main frequency had the same amplitude). Unlike other tests, the amplitude of the red signals is very weak and is not visible using the conventional spectrogram but is clearly visible with Algorithm 2.

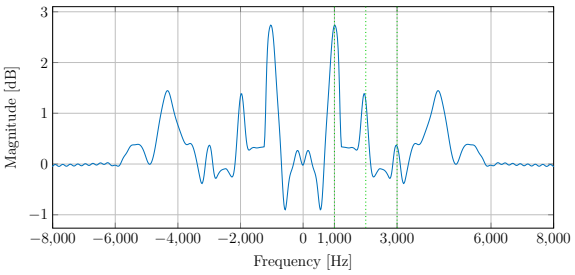
During the tests on the nRF52832, it turned out that the audio stream was audible in the speakers although the wire between the amplifier input and the PWM output was disconnected, performing then the setup of Fig. 6-16a. The Fig. 6-16b shows that the red signal is present in the FH signal with a peak of 2.7 dB. Since the speakers are still playing the signal, it means that the audio signal is present in the power supply of the board. The leak is not only due to the consumption of the amplifier but also to the BLE dongle itself. The amplitude of the spike being higher when the cable is disconnected, we can assume that the input of the audio amplifier acts as an attenuator on the side-channel. This behavior has been observed on two different nRF52832 boards but not on the ESP32 board. This difference may be due to the design of the two boards. The nRF52832 is a development board while the ESP32 is an industrial version, with a higher-grade power supply.

> The system can detect a weak leak in an FH system but also that a leak can be present naturally, i.e. without forcing it, an audio stream connected to amplifiers can leak the audio content on an FH transmission. Therefore, the leaks have a large range and pass-through soundproof walls due to the radio carrier, which is a threat to privacy. The components used are not modified, so this configuration can exist in reality. Moreover, several components can be the root of a leakage, so the side-channel removal can be more intricate.

The origin of the leakage can be attributed to two components of the board, either the one



(a) Schematics of nRF52832 setup with audio wire disconnected.



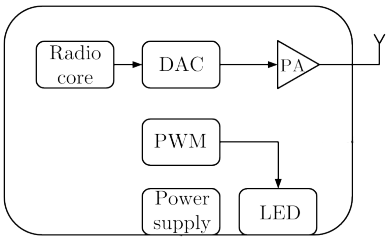
(b) Difference of FFT-BE CPM output between with and PWM without generated.

Figure 6-16 – Side-channel checking with audio wire disconnected (PWM at 1 kHz).

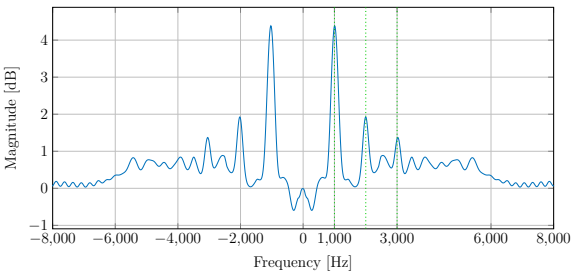
realizing the PWM or the LED, the next two sections will focus on each of the two possible sources.

4.2 LED BLINKING IMPACTS

This section focuses on determining to what extent LED and the signal that drives them can generate a compromise on the radio transceiver. To do this, a PWM at 1000 Hz is generated on two boards, at first the nRF52832 and later on with the PCA10059, using the following setup Fig. 6-17a. To begin we observe the signal of the nRF52832 using a wire connected to the output of the transceiver, the result is visible in Fig. 6-17b. We can see that not only a leakage is clearly noticeable at the appropriate frequency, but that the leakage is even higher than in the previous test section with a spike of 4.2 dB. The audio amplifier that was previously connected includes large capacitors in order to compensate noise generated in the power supply and thus probably reduces the amplitude of the side-channel.



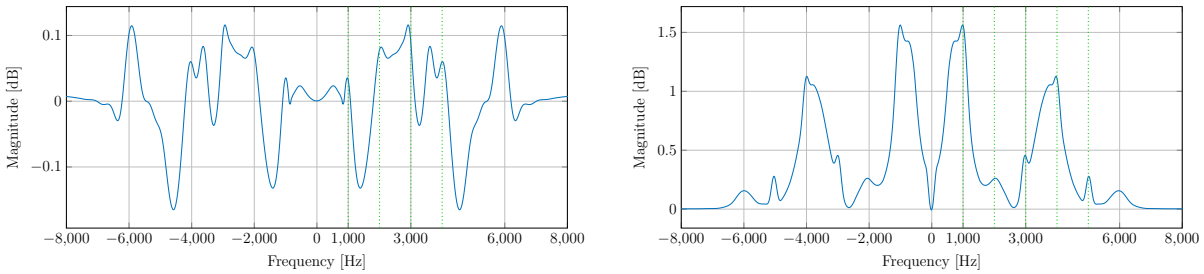
(a) Schematics of nRF52832 setup.



(b) Difference of FFT-BE CPM output between with and PWM without generated on nRF52832.

Figure 6-17 – Side-channel checking against LED with wired SDR.

The signals studied above were captured by connecting the output of a radio transceiver to a cable in order to eliminate possible interferers, but this case does not take into account other EM leakage that could be generated on the dongle before the antenna output. Moreover, this does not correspond to a realistic attack scenario, were if an attacker listens to the EM emissions of a dongle, it means that he does not have physical access to it. To overcome these limitations, the BLE dongle nRF52832 and PCA10059 have been placed in an anechoic chamber in order to investigate their radiation only. A 2.4 GHz antenna is placed 10 cm next to the dongle and its cable does not overlap the dongle to avoid interference (the nRF52832 is not used with a cable but with its integrated antenna in this case). The results of these measurements are shown in Fig. 6-18. We can observe that the red signal is still present on both dongles but with a different amplitude (1.5 dB for the nRF52832 and less than 0.1 dB for the PCA10059). Moreover, the signal received is much noisier than with a cable in the case of the nRF52832 (in addition its amplitude is halved). The compromise of the PCA10059 is extremely low and can be easily blended in the noise to become unnoticeable.



(a) PCA10059.

(b) nRF52832.

Figure 6-18 – Side-channel checking against LED with wireless SDR.

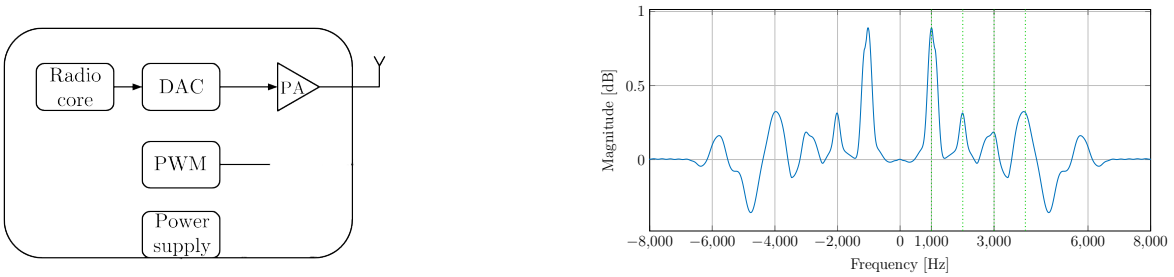
— This section demonstrates that a leak can be produced by a blinking LED and that this phenomenon is not limited to a particular model of dongle but that several models have the same flaws.

— The interception system is able to extract them remotely as an attacker would do.

4.3 Internal system impact: PWM generator

To further investigate the origin of a leak on the nRF52832 board, it has been observed that the combination of LED and PWM generated a leak, to find out where the leak comes from.

This section will investigate if a PWM signal sent to a GPIO output connected to nothing can still generate a leak, as given in Fig. 6-19a. The corresponding recovered signal is in Fig. 6-19b. The study of this figure shows that a leakage is still present at the same position as with the LED of the previous Section, but its amplitude has been greatly reduced to 0.8 dB thus indicating that the LED was indeed leaking but is not the only one causing leaks.



(a) Schematics of nRF52832 setup.

(b) Difference of FFT-BE output between with and PWM without generated.

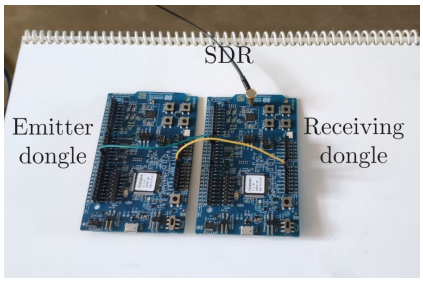
Figure 6-19 – Side-channel checking against PWM leak (at 1 kHz).

The experiments of this section demonstrate that there is a combination of factors that can generate a leakage and that a small leak can be amplified internally. The PWM alone generates a leak but is amplified by LED. Such a leak is a major security threat because it allows in an air gap bridging context to leak information on a radio transceiver without directly modifying its baseband signal but using a side-channel instead. It is therefore more difficult to detect that such attack is used because as it is common for microcontroller to have unused outputs.

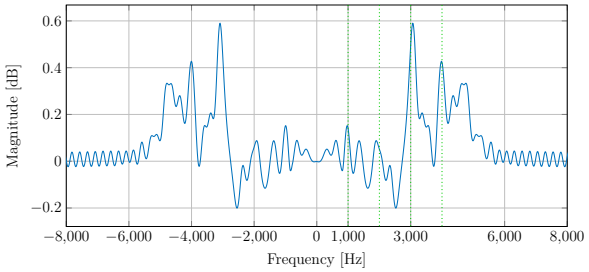
4.4 Internal system impact: onboard wire

A leak exists with a PWM alone when it is connected to a GPIO, so the leak can be the part of the dongle that generates the PWM or it is the routing of the PWM signal within the board that generates this leak. To discern these two origins, we will stop generating a PWM within the observed device but send it to one of its GPIO (other than that used by the LED in the previous tests) coming from another board, and observe if a compromise is present on the intercepted FH signal. The setup is then shown in Fig. 6-20a. A dongle (left) is used to make a PWM which is sent to the listened dongle (right), a cable is also placed between the two dongles to connect the grounds. The associated result is shown in Fig. 6-20b. The leakage is of low amplitude and until its harmonic at 3000 Hz it is not really possible to say that a leakage is

occurring. The maximum leak amplitude is 0.6 dB which is the same order of magnitude as in the previous section.



(a) Photo of routage checking setup.



(b) Difference of FFT-BE CPM output between with and PWM without generated.

Figure 6-20 – Side-channel checking against PWM leak (at 1 kHz).

This test demonstrates that the routing alone of a signal in a board can generate a leak which means that potentially all interactions using an output including data buses can leak and end up on the FH signal. Moreover, a high frequency signal (e.g. a data bus) will tend to generate a higher radiation and consequently a higher leakage.

5 Conclusion

This chapter is the final embodiment of what has been studied in this thesis, starting with the study of the side-channels to know their origin, to verify if it is possible to create them and to study their presence. Different methods to intercept an FH signal have been considered, allowing the creation of a real-time interception system. The measurements of this chapter have been realized according to three levels:

— A first level is used to validate the correct operation of the interception system in real-time. Thus it valides the FPGA design and makes a link with the simulated performance of the Chapter 4. It also checks if our system was able to correctly intercept leaks generated by devices following the BLE protocol, namely a frequency hopping system (with channels of 2 MHz), having a high temporal sporadicity (the signal is active less than 10% of the time) and using a black signal GFSK modulation.

— The second level aims at generating a leak in a BLE dongle and detecting them on the recovered FH signal. For this purpose a mixer was used on the antenna output and a

coil placed near the power supply in order to emulate compromises coming from sub parts of a system. In both cases our system is not only able to intercept the BLE signals and extract seamlessly the baseband signal, but also to see that the various red signals injected were present on the radio signals. Variations in the leaks of different dongle models have been observed, making it possible to differentiate them by simply observing their trace.

— The last level consists in the observation of the BLE dongle executing several tasks corresponding to real use cases: generation of sound signals, LED control, signal generators and signal acquisition. In all cases considered, leakage on the FH signal has been noticed. These leaks are directly correlated with the tasks performed by the dongle because the characteristic frequencies have been observed. Moreover, the switch off of the tasks leads to leakage disappearance. This last level corresponds to the realization of exploit which allows to confirm that it is possible to recover sensitive data internal to a device only by listening to its FH transmissions.

Chapter **SEVEN**

CONCLUSION & PERSPECTIVES

Contents

This chapter aims to recall the global context of the work performed during this thesis. A summary of the research and contributions will be presented. The short- and long-term perspectives opened by this thesis will also be given.

1 Conclusion

Information processing systems are an important and essential part of our life, for applications ranging from industrial to individual contexts. A large part of these systems will carry out processing and will store, move and process sensitive information that one would like to limit unwanted circulation. The study and protection against the information leakage are designed by the so-called *Information Security*. It concerns, among other things, how to mitigate information risks and limit unauthorized access to data. Most of the time information security addresses how to protect against cyber attacks, and involves digital measures such as encryption, firewall, electronic passwords. The applications and threats to security concern areas wider than just digital means and involve for example social engineering or the data scraping.

Among the areas often ignored is the side-channel analysis. A side-channel is the unwanted deviation of information from its legitimate channel to another channel called side-channel. An attack based on a side-channel is, for example, looking for information through indirect means. It looks at how a task is performed, rather than looking at the outcome of the task in order to extract sensitive information.

This thesis considers side-channel emanating from a device. These side-channels involve physical medium such as light, vibrations, power supply disturbances or even electromagnetic radiation. These side-channels are a result of the devices operation and for the most of them are difficult to erase. A study of the different side-channel attacks of the literature is made considering the type of medium and the sensitive information that passes through it. A characterization according to their dangerousness is carried out which is mainly related to their range and the possible throughput of information.

Among the different existing side-channels, the one affecting the radio transceiver stands out from the others, mainly because of its greater range as it is propagated by the radio transmitter. Its study requires knowledge on radio frequencies, the ability to operate a reception system adapted to the device you are listening to, but also knowledge on the internal functioning of the device to identify where the leak comes from. However, intercepting a radio signal can become complex when it constantly changes its carrier frequency through what is called frequency hopping.

One of the main objectives of this thesis is the interception of FH signals to analyze its baseband hidden information. Several constraints on the FH interception system are due to its prospective use in security audits:

— The interception system must be done in practice, i.e., is concretized by a physical system and not only realized in the simulation.

— The interception must be done blindly, i.e., without knowing the precise characteristics of the FH signal, in particular its hop sequence is unknown.

— The interception must be done on a wide band because the FH channels are not known but supposed to be contained within a specific frequency range.

— The interception must be done in real-time due to bandwidth constraints which would require significant storage if the data were processed afterwards.

— The interception system must be transportable in order to be used in security audits

in sensitive areas, therefore it must not resort to cloud computing for processing that should be done locally.

The methods to intercept FH signals are diverse. However, many of them depend on *a priori* information about the FH signal or require complex computations. The number of methods that can be used in our study is thus reduced to three due to the constraints: DFT multi-band, PPN and wavelet methods. These methods have been analyzed in simulations in order to choose the one with the best performance considering both the ability to detect the channel of an FH signal but also the ability to extract the sensitive signal from the baseband.

The use of a DFT-based method has proven to be the most relevant in terms of performance and computational complexity and is therefore the method chosen for implementation in this thesis. The interception system is based on an SDR allowing fast system design, adaptability to several frequency bands, and fast customization. However, the high bandwidth requires several implementation precautions. The computations must be done efficiently, for this reason a hardware accelerator based on FPGA has been added to the design. This last one handles all the processing requiring high throughput. The GPP is then only used for processing on the extracted baseband. Thus, our solution mixes speed of FPGA and flexibility of GPP. A deployment of the RFNoC stack through the Julia language allowed a proof-of-concept design as well as an optimized design, and performance evaluation testbench using the same programming language.

Finally, the interception system has been tested against various targets. A step-by-step approach was considered, starting with listening to the signals synthesized with dedicated radio equipment to ascertain the real-time performance and the likeness of the actual and simulated performance (in both channel detection and red signal recovery). Then exploring the available side-channels thanks to the state of the art, compromises have been enforced within commercial frequency hopping systems. These various tests have allowed highlighting not only the good functioning of the interception system that has succeeded in extracting sensitive data but also that it is possible to inject a compromise into a system following the template used in *air gap bridging*. The final step is the listening to targets performing various tasks common to radio systems such as music broadcasting, LED control or signal digitization. Thanks to our system, leaks were detected due to the previously mentioned tasks and therefore a recovery of the associated red data was performed. This confirms that it is indeed possible to intercept side-channels when they affect radio transceivers secured by frequency hopping. Moreover,

different behaviors have been highlighted between various FH devices, paving the way for future countermeasures by studying the designs that minimize leakage.

Although the interception system is functional, some improvements can be included to efficiently address new targets and extend its area of application. These improvements are detailed in the next section as the perspectives of this work.

2 Perspectives

The work carried out offers several perspectives of study that can be classified in two different categories, the short and long term. The first one considers the use as-is of the interception system by applying it to new targets and by extending its fields of application. The long-term perspectives involve altering the interception system to improve its interception performance.

2.1 Short-term perspectives

Although several targets have already been tested in order to ascertain the presence of side-channels, our study can be extended to other devices such as Bluetooth headsets, computers or micro system like a Raspberry Pi. These last two cases involve more data processing and therefore constitute a prime target for a potential attacker. Inspecting wider bandwidth devices such as 5G transceivers would also expand the possible uses of the interception system. Verifying that no leakage is present in this type of device would be a significant advantage in security audits.

The real performance has been evaluated either by connecting the devices by cable or by placing the antenna relatively close to the BT device to be tested. An investigation could evaluate the performance as a function of the distance between the antenna and the BT device, but also under various radio environments like a house or an office.

The interception system also provides a new perspective other than the detection of confidential information: the ability to recognize devices by using the recurrent peculiarity of their radio spectrum. This new feature would allow the interception system to not only detect a red signal but also to identify which device is transmitting it. This is useful in the case of a radio environment where several devices are transmitting FH signals on the same bands.

A proof of concept has been performed using the 3 available BLE dongle models (3 dongles are of the same model and two others of two different models). First a characteristic signature of the nRF52832 was performed by merging several signals obtained through Algorithm 2. Then the signature is compared against several listening of various dongles, this result is shown in Fig. 7-1.

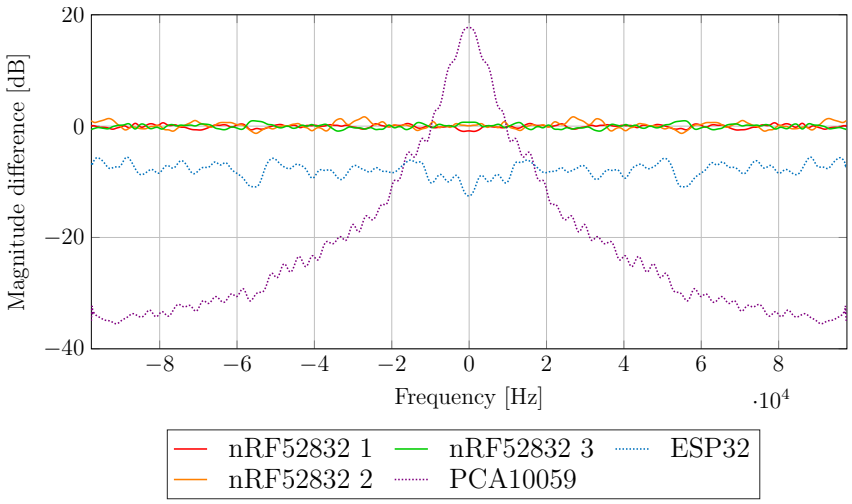


Figure 7-1 – Proof of concept of radio fingerprinting.

The three nRF52832 curves are very close to the 0 indicating that they are effectively issued from nRF52832 dongle, while the two other dongles are far away and have a different shape. It is therefore possible to recognize the device used by looking at the output of the partial spectrum algorithm.

The work carried out highlighted that there were various leaks in the device and that these leaks can be propagated to several places and impact the radio signals emitted by transceivers. Analysis of the data in the leak can help tracing it back to its origin, which can help discover the cause and path of the leak. These steps are essential in order to either suppress a leak or if this is not possible to establish countermeasures preventing an attacker from recovering confidential information. These measures can be the implementation of power supply filtering, as we have seen between Fig. 6-16 and Fig. 6-17 where removing the capacitors of the audio amplifier has increased or rather not lowered the leakage. Side-channel jamming can be performed by leaking misleading signal for example by using free GPIO, as it is possible to do in Section 6-4.4.

The commonly accepted recommendations for safety as defined in [NSA82] involve physical distancing around the device. However, in our case it is impractical since the range of the radio

signals is too long. This makes leaks impacting radio transceivers an important threat because harder to contain.

2.2 Long-term perspectives

The interception system can only listen to a device when its signals are stronger than the interference, therefore the antenna must be placed close to the device to be investigated in order to minimize strong interference. However, this is not always possible and therefore the leak detection will not be accurate. To improve the performance it is necessary to isolate more finely the device under test, this can be done by using directional antennas, e.g. the one in Fig. 7-2 tuned to intercept radiation from a video cable. Diversity can also be used, by merging the signals from several spatially separated antennas.

Figure 7-2 – 3D printed Yagi antenna at 770 MHz.

Currently the detector has some limitations on the width and number of detectable channels. The number of channels depends on the size of the FFT which goes from 16 to 1024 by power of two, the received bandwidth is 160 MHz maximum, and the size of the channels depends directly on the ratio between the received bandwidth and the FFT size. It is possible to change these limits without changing the hardware.

— The radio has two front ends, each of them able to receive 160 Ms/s, but only one is used at a time. It is possible to use both front ends at the same time to obtain a bandwidth of 320 MHz.

— The FFT size limit is not directly due to the FFT IP but to the way the data transport is managed. Indeed 1024 is the maximum value (aligned with a power of two) of samples that can be sent in a single frame via the X310 crossbar. An FFT is performed frame by frame, which explains the 1024 limit. But if one wants to detect more channels (some

military protocols can use $10,000$ channels for example), a buffer must be inserted in the FFT IP in order to increase the number of samples provided to the FFT.

— At the output of the FPGA, two channels are issued, the detected channel and the channel of the previous index. It is however possible to change which channel is returned without significant modification, for example by outputting two adjacent channels to double the recovered bandwidth.

— To finish with possible improvements, the detector has difficulty to detect small time slot T_s with many channels (minimum 6 μs with 1024 channels in the current configuration). To circumvent this problem, given that the detector does not take much space in the FPGA, it is possible to place several of them in parallel but with shifted reading windows. This will create a temporal overlapping which will improve the temporal accuracy but will also improve the performance against noise as it will be possible to perform an averaging on the output of the detector.

List of Figures

List of Tables

TABLE OF ACRONYMS

ADC Analog to Digital Converter
AES Advanced Encryption Standard
ALU Arithmetic-Logic Unit
AM Amplitude Modulation
API Application Program Interface
ASIC Application Specific Integrated Circuit
AWGN Additive White Gaussian Noise
B-ASK Binary Amplitude-Shift Keying
B-FSK Binary FSK
BT Bluetooth
BLE Bluetooth Low Energy
CER Channel Error Rate
CPA Correlation Power Analysis
CPU Central Processing Unit
CR Cognitive Radio
DAC Digital to Analog Converter
DDC Digital Down Converter
DDF Detector Derotor Filtering
DDR-RAM Data Rate Synchronous Dynamic RAM
DECT Digital Enhanced Cordless Telecommunications
DFA Differential Fault Analysis
DFG Data Flow Graph
DFT-BE DFT - Bin Extraction
DMA Direct Memory Access
DPA Differential Power Analysis
DSP Digital Signal Processor
DUC Digital Up Converter
DUT Device Under Test
EMC Electromagnetic Compatibility
FBMC-FS Filter Bank Multi-Carrier for Frequency Spreading
FFT-BE FFT - Bin Extraction
FFT Fast Fourier Transform
FH Frequency Hopping
FHSS Frequency Hopping Spread Spectrum
FPGA Field Programmable Gate Array
FPU Floating Point Unit
GbE Gigabit Ethernet
GPIO General Purpose Input Output
GPMC General Purpose Memory Controller
GPP General Purpose Processor
GPU Graphics Processing Unit
HDFT Hopping Discrete Fourier Transform
HDL Hardware Description Language
HLS High Level Synthesis
HPC High Performance Computing
IDFT Inverse Discrete Fourier Transform
ISM industrial, scientific and medical
JIT Just In Time
LIF Low Intermediate Frequency
LLVM Low Level Virtual Machine
MD Multiple Dispatch
ML Machine Learning
MSE Mean Square Error
NoC Network on Chip
NRZI Non-Return to Zero Inverted
NSA National Security Agency
ODG Objective Difference Grade
PCB Printed Circuit Board
PCIe Peripheral Component Interconnect Express Bus
PEAQ Perceptual Evaluation of Audio Quality
PESQ Perceptual Evaluation of Speech Quality
PPN PolyPhase Network
PSD Power Spectral Density
PWM Pulse Width Modulation

RFNoC RF Network-on-Chip
RF Radio frequency
SBOS Selected Bands Opinion Score
sDFT sliding Discrete Fourier Transform
SDR Software-Defined Radio
sFFT sliding Fast Fourier Transform
SNR Signal-to-Noise Ratio
SoC System on Chip
SPA Simple Power Analysis
STFT Short Time Fourier Transform
SVD Singular Value Decomposition
SVM Support Vector Machine
SVR Support Vector Regression
THD Total Harmonic Distortion
UHD USRP Hardware Driver
USB Universal Serial Bus
USRP Universal Software Radio Peripheral
ViSQOL Virtual Speech Quality Objective Listener
VSG Vector Signal Generator
ZIF Zero Intermediate Frequency

TABLE OF SYMBOLS

A	averaging factor of covariance based detector
α_p	time sporadicity factor expressing whether transmission occurs during a time slot
b	baseband message
C	detected hop positions of the synchronization algorithm
C_0	initial hop pool of the synchronization algorithm
C_a	pool of too close hops of the synchronization algorithm
C_m	pool of missing hops of the synchronization algorithm
C_o	pool of hops out of the hop grid of the synchronization algorithm
d	received FH signal
E_w	internal value of waveform based detector
F_b	baseband bandwidth
$f_p(t)$	associated channel frequency of p
F_s	transmitted signal bandwidth
$\hat{d}$	d aligned with synchronization algorithm
$\hat{d}_{\mathrm{BB}}[k]$	aligned baseband message at F_s
$\hat{p}$	estimated channel index
$\hat{R}_d[k]$	statistical covariance matrix (LxL)
K	internal averaging factor of detectors, counterpart of λ at Rx side, act as a temporal overlapping of the detectors

k	sample index at F_s
K_{PPN}	temporal overlapping of HDFT filter bank based detector
L	number of samples used in the detectors for one estimation of $\hat{p}$
λ	repetition factor, as multiple of N
$\mathcal{H}_0$	statistical test of detectors: the channel is not used
$\mathcal{H}_1$	statistical test of detectors: the channel is used
$M[k]$	instantaneous derivative of the synchronization algorithm
M_{max}	maximum eigenvalues of $\hat{R}_d$
M_{min}	minimum eigenvalues of $\hat{R}_d$
N	number of channels of FH
N_{FFT}	number of bins of the FFT-BE method
p	channel index
P_d	probability of detection of the detectors
P_f	probability of false alarm of the detectors
$R_d[k,\tau]$	autocorrelation
$r_l[k]$	samples autocorrelation
τ	Rx delay associated to d
T_s	duration of a time slot
υ	received energy of energy based detector
u	sample index at F_b
$V[k]$	arg max DFT of the synchronization algorithm
$\bar{V}[k]$	sliding average of $V[k]$ with a depth α of the synchronization algorithm
W	width of Phydyas filtering in HDFT filter bank based detector
W_f	waveform model of waveform detector
$w(t)$	Additive White Gaussian Noise (AWGN) of variance σ_w^2
$x[k]$	transmitted signal
ζ	threshold of the detectors

Appendices

Appendix A

SIDE-CHANNEL ATTACK LITERATURE CLASSIFICATION

The Table A-1 is an overview of the different side-channels mentioned in Chapter 2. It provides a better understanding of the wide range of side-channels and includes their classification according to their target, their medium, as well as their activeness and their intentionality.

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Novel method to detect and recover the keystrokes of PS/2 keyboard	[Du13]	Du	2013	Power line	External Component	Passive	Fortuitous	0 m
A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion	[Man02]	Mangard	2002	Power line	Crypto system	Passive	Fortuitous	0 m
Differential Power Analysis	[Koc99]	Kocher	1999	Power line	Crypto system	Passive	Fortuitous	0 m
Correlation Power Analysis with a Leakage Model	[Bri04]	Brier	2004	Power line	Crypto system	Passive	Fortuitous	0 m
Differential fault analysis of secret key cryptosystems	[Bih97]	Biham	1997	Power line	Crypto system	Passive	Fortuitous	0 m
Localized Electromagnetic Analysis of Cryptographic Implementations	[Hey12]	Heyszl	2012	Power line	Crypto system	Passive	Fortuitous	0 m
Information Recovery Using Electromagnetic Emanations From Display Devices Under Realistic Environment	[Le08]	Le	2008	Power line	Crypto system	Passive	Fortuitous	0 m
No Free Charge Theorem: A Covert Channel via USB Charging Cable on Mobile Devices	[Spo17]	Spolaor	2017	Power line	Internal Component	Passive	Intentional	0 m
Information leakage from optical emanations	[Lou02]	Loughry	2002	Light	Status LED	Passive	Fortuitous	38 m
xLED: Covert Data Exfiltration fromAir-Gapped Networks via Router LEDs	[Gur17b]	Guri	2017	Light	Status LED	Active	Intentional	Line of sight
LED-it-GO: Leaking (A Lot of) Datafrom Air-Gapped Computers via the (Small) Hard Drive LED	[Gur17c]	Guri	2017	Light	Status LED	Active	Intentional	Line of sight
Exfiltration of Information from Air-Gapped Machines Using Monitor LED Indicator	[Sep14]	Sepetnitsky	2014	Light	Status LED	Active	Intentional	Line of sight
Optical Time-Domain Eavesdropping Risks of CRT Displays	[Kuh02]	Kuhn	2002	Light	Screen	Passive	Fortuitous	80 m
Compromising Reflections-or-How to Read LCD Monitors around the Corner	[Bac08]	Backes	2008	Light	Screen	Passive	Fortuitous	30 m

Table A-1 continued from previous page

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Tempest in a Teapot: Compromising Reflections Revisited	[Bac09]	Backes	2009	Light	Screen	Passive	Fortuitous	30 m
An optical covert-channel to leak data through an air-gap	[Gur16a]	Guri	2016	Light	Screen	Passive	Intentional	30 m
Side Channel Attacks Using Optical Sampling Of Mechanical Energy And Power Line Leakag	[Bar09]	Barisani	2009	Light	Sound system	Passive	Fortuitous	Line of sight
Light Commands: Laser-Based Audio Injection on Voice-Controllable Systems	[Sug20]	Sugawara	2019	Light	Sound system	Active	Intentional	Line of sight
PILOT: Password and PIN Information Leakage from Obfuscated Typing Videos	[Bal19]	Balagani	2019	Light	External Component	Passive	Fortuitous	Line of sight
EyeTell: Video-Assisted Touchscreen Keystroke Inference from Eye Movements	[Che18]	Chen	2018	Light	External Component	Passive	Fortuitous	Line of sight
On Covert Acoustical Mesh Networks in Air	[Han13]	Hanspach	2013	Sound	Sound system	Active	Intentional	19.7 m
MOSQUITO: Covert Ultrasonic Transmissions between Two Air-Gapped Computers using Speaker-to-Speaker Communication	[Gur18c]	Guri	2018	Sound	Sound system	Active	Intentional	8 m
Acoustic Data Exfiltration from Speakerless Air-Gapped Computers via Covert Hard-Drive Noise ('DiskFiltration')	[Gur17a]	Guri	2017	Sound	Sound system	Active	Intentional	2 m
Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers	[Gur16c]	Guri	2016	Sound	Sound system	Active	Intentional	8 m
Acoustic cryptanalysis: on nosy people and noisy machines	[Tro04]	Tromer	2004	Sound	Internal Component	Passive	Fortuitous	2 m
RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis	[Gen14]	Genkin	2014	Sound	Internal Component	Passive	Fortuitous	4 m
Acoustic Side-Channel Attacks on Printers	[Bac10]	Backes	2010	Sound	External Component	Passive	Fortuitous	4 m
Keyboard acoustic emanations	[Aso04]	Asonov	2004	Sound	External Component	Passive	Fortuitous	15 m

Table A-1 continued from previous page

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Don't Skype & Type!	[Com17]	Compagno	2017	Sound	External Component	Passive	Fortuitous	Internet
Keyboard Emanations in Remote Voice Calls	[Ana18]	Anand	2018	Sound	External Component	Passive	Fortuitous	Internet
Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels	[Gen18]	Genkin	2018	Sound	External Component	Passive	Fortuitous	10 m
Screaming Channels: When Electromagnetic Side-Channels Meet Radio Transceivers	[Cam18]	Camurati	2018	EM - forced	Internal Component	Active	Fortuitous	10 m
Acoustic Eavesdropping through Wireless Vibrometry	[Wei15]	Wei	2015	EM - forced	Sound system	Active	Fortuitous	5 m
Through-Wall Human Pose Estimation Using Radio Signals	[Zha18]	Zhao	2018	EM - forced	External Component	Active	Intentional	12 m
Recognizing Keystrokes Using WiFi Devices	[Ali17]	Ali	2017	EM - forced	External Component	Active	Fortuitous	4 m
Tracking Keystrokes Using Wireless Signals	[Che15]	Chen	2015	EM - forced	External Component	Active	Fortuitous	5 m
We Can Hear You with Wi-Fi!	[Wan16b]	Wang	2016	EM - forced	Sound system	Active	Fortuitous	2 m
NSA catalog pages	[NSA08]	NSA	2008	EM - forced	Device cable	Active	Intentional	12 km
A Feasibility Study of Radio-frequency Retroreflector Attack	[Wak18]	Wakabayashi	2018	EM - forced	Device cable	Active	Intentional	10 m
Electromagnetic Information Extortion from Electronic Devices Using Interceptor and Its Countermeasure	[Kin19]	Kinugawa	2019	EM - forced	Internal Component and device cable	Active	Intentional	5 m
The thing	[Uni]	Soviet Union	1945	EM - forced	Sound system	Active	Intentional	> 30 m
Electromagnetic radiation from video display units: An eavesdropping risk?	[Eck85]	Van Eck	1985	EM - radiation	Screen	Passive	Fortuitous	1 km
Electromagnetic Eavesdropping Risks of Flat-panel Displays	[Kuh04]	Kuhn	2004	EM - radiation	Screen	Passive	Fortuitous	3 m
Eavesdropping attacks on computer displays	[Kuh06]	Kuhn	2006	EM - radiation	Screen	Passive	Fortuitous	10 m

Table A-1 continued from previous page

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Realistic eavesdropping attacks on computer displays with low-cost and mobile receiver system	[Eli12]	Elibol	2012	EM - radiation	Screen	Passive	Fortuitous	46 m
Investigation of Unintentional Video Emanations From a VGA Connector in the Desktop Computers	[Zha17b]	Zhang	2017	EM - radiation	Screen	Passive	Fortuitous	~ 10 cm
A novel method for computer video leaking signal detection	[Shi14]	Shi	2014	EM - radiation	Screen	Passive	Fortuitous	n.a.
Remote video eavesdropping using a software-defined radio platform	[Mar14]	Marinov	2014	EM - radiation	Screen	Passive	Fortuitous	7 m
A Threat for Tablet PCs in Public Space	[Hay14]	Hayashi	2014	EM - radiation	Screen	Passive	Fortuitous	10 m
Information Recovery Using Electromagnetic Emanations From Display Devices Under Realistic Environment	[Lee19]	Lee	2019	EM - radiation	Screen	Passive	Fortuitous	20 m
Video information recovery from EM leakage of computers based on storage oscilloscope	[Yan10]	Yang	2010	EM - radiation	Screen	Passive	Fortuitous	~ 2 m
Accuracy Enhancement of Electromagnetic Side-Channel Attacks on Computer Monitors	[Say18]	Sayakkara	2018	EM - radiation	Screen	Passive	Fortuitous	~ 2 m
Tempest for Eliza	[Thi01]	Erikyyy	2001	EM - radiation	Screen	Passive	Intentional	100 m
AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies	[Gur14]	Guri	2014	EM - radiation	Screen	Passive	Fortuitous	7 m
TEMPEST: A Signal Problem	[NSA07]	NSA	2007	EM - radiation	External Component	Passive	Fortuitous	25 m
Keyboard acoustic emanations	[Aso04]	Asonov	2004	EM - radiation	External Component	Passive	Fortuitous	20 m
Compromising Electromagnetic Emanations of Wired and Wireless Keyboards	[Vua09]	Vuagnoux	2009	EM - radiation	External Component	Passive	Fortuitous	20 m
The threat of information theft by reception of electromagnetic radiation from RS-232 cables	[Smu90]	Smulders	1990	EM - radiation	Device cable	Passive	Fortuitous	9 m

Table A-1 continued from previous page

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Trust The Wire, They Always Told Me!	[Sch16]	Schulz	2016	EM - radiation	Device cable	Passive	Fortuitous	2 cm
Measurement and analysis of the compromising electromagnetic emanations from USB keyboard	[Sim16]	Sim	2016	EM - radiation	Device cable	Passive	Fortuitous	~ 20 cm
Reconstruction of leaked signal from USB keyboards	[Cho16]	Choi	2016	EM - radiation	Device cable	Passive	Fortuitous	15 cm
TEMPEST in USB	[Zha17a]	Zhang	2017	EM - radiation	Device cable	Passive	Fortuitous	0 m
USBee: Air-gap covert-channel via electromagnetic emission from USB	[Gur16b]	Guri	2016	EM - radiation	Device cable	Passive	Intentional	9 m
TEMPEST: A Signal Problem	[NSA07]	NSA	2007	EM - radiation	Internal Component	Passive	Fortuitous	25 m
Meet “badBIOS,” the mysterious MacPC malware that jumps airgaps	[Goo13]	Goodin	2013	EM - radiation	Internal Component	Passive	Intentional	~ 10 m
GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies	[Gur15]	Guri	2015	EM - radiation	Internal Component	Passive	Intentional	30 m
Funtenna	[Cui15]	Cui	2015	EM - radiation	Internal Component	Passive	Intentional	> 5 m
ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields	[Gur18d]	Guri	2018	EM - radiation	Internal Component	Passive	Intentional	1 m
Covert channels using mobile device’s magnetic field sensors	[Mat16]	Matyunin	2016	EM - radiation	Internal Component	Passive	Intentional	15 cm
ElectroMagnetic Analysis EMA: Measures and Counter-Measures for Smart Cards	[Qui01]	Quisquater	2001	EM - radiation	Internal Component	Passive	Fortuitous	2 cm
Side Channel Attacks on Smartphones and Embedded Devices Using Standard Radio Equipment	[Gol15]	Goller	2015	EM - radiation	Internal Component	Passive	Fortuitous	80 cm
One&Done: A Single-Decryption EM-Based Attack on OpenSSL Constant-Time Blinded RSA	[Ala18]	Alam	2018	EM - radiation	Internal Component	Passive	Fortuitous	2 cm

Table A-1 continued from previous page

Name	Ref	First Author	Year	Medium	Target	Activeness	Intentionality	Max range
Enhancement of simple electromagnetic attacks by pre-characterization in frequency domain and demodulation techniques	[Mey11]	Meynard	2011	EM - radiation	Internal Component	Passive	Fortuitous	~ 4 cm
ElectroMagnetic analysis (EMA) of software AES on Java mobile phones	[Abo11]	Aboulkassimi	2011	EM - radiation	Internal Component	Passive	Fortuitous	1 cm
The EM Side-Channel(s)	[Agr03]	Agrawal	2003	EM - radiation	Internal Component	Passive	Fortuitous	12 m
Security and privacy in computer systems	[War67]	Ware	1967	EM - radiation	Device cable	Passive	Fortuitous	0 m
USB Snooping Made Easy: Crosstalk Leakage Attacks on USB Hubs	[Su17]	Su	2017	EM - radiation	Device cable	Passive	Fortuitous	0 m
Crosstalk-Sensitive Loops and Reconstruction Algorithms to Eavesdrop Digital Signals Transmitted Along Differential Interconnects	[Yua17]	Yuan	2017	EM - radiation	Device cable	Passive	Fortuitous	2 cm

Table A-1 – Classification of the papers on side-channel attacks (EM stands for ElectroMagnetic).

Appendix B

RFNoC

RFNoC [laba] is a network-distributed heterogeneous processing tool which enables faster development of applications based on USRP devices with an internal FPGA. It allows notably a seamless use of both host-based and FPGA-based processing in an application. It provides a way to leverage SDR processing capabilities with FPGA acceleration across multiple FPGAs and devices across a network. In association with UHD, it provides useful and convenient tools for SDR-based application development and provides access to high-level languages such as Julia, Python, Matlab, LabVIEW.

1 RFNoC stack

Radio Frequency Network-on-Chip (RFNoC) works in combination with UHD in order to provide a solution for data exchange, monitoring and control in real time without the burden of coding the different parts. For this RFNoC is implemented in an FPGA and provides templates for the development of processing blocks. Thanks to these templates, the developed blocks can run on several models and generations of SDR. UHD, on the other hand, is installed on a computer called host in the following. It ensures communication with the SDR on the host side, and also provides a control interface for configuring the FPGA through RFNoC. The use of RFNoC in order to develop an SDR application relies on several layers that will be described hereafter.

1.1 FPGA integration

The signal processing FPGA algorithms are contained in modules known as "Computation Engines" or simply "Blocks" as seen in Fig. B-1. Each block has several specific parts that will be further detailed in Section 3. But it is noteworthy that in each block there is an interface wrapper provided in order to encapsulate IP processing to use within RFNoC. This allows importing generic FPGA IP or even Xilinx CoreGen IP blocks and using them without having to rewrite their internal functioning. Each RFNoC block processing is independent from other blocks, and can be designed with any tool that supports AXI stream interfaces, including VHDL, Verilog, or Vivado HLS (it should be noted that as for now, every USRP device with embeds an FPGA use a Xilinx FPGA).

1.2 RFNoC stack

UHD integration: UHD provides an abstraction layer to dialogue with many different SDR while keeping the same instruction. It offers notably the following features:

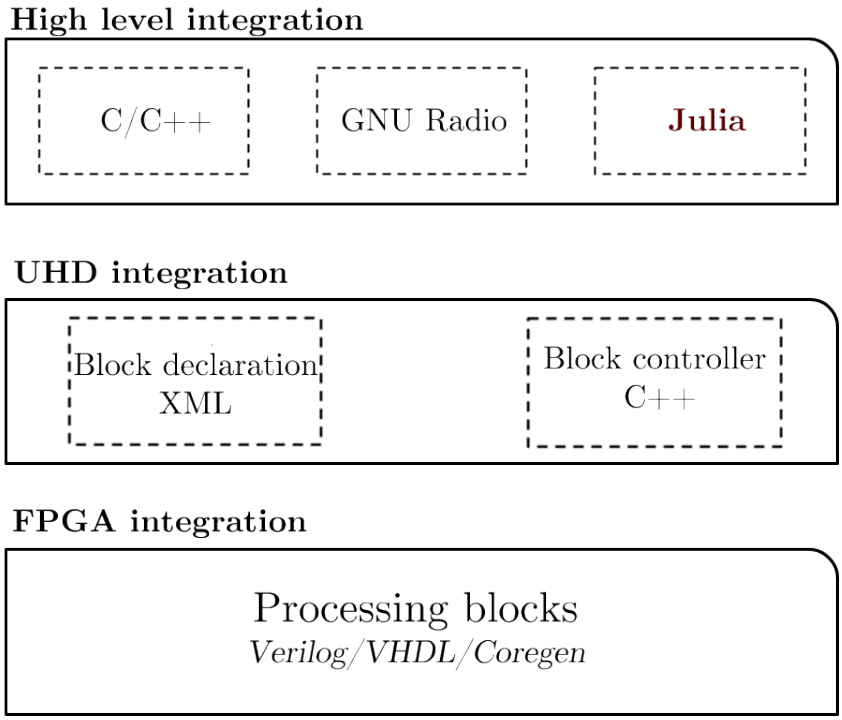


Figure B-1 – RFNoC stack.

— Configure connections between blocks, setup flow control and configure stream IDs
— Configure block-specific settings
— Initiate streaming of the ingress egress interface (Ethernet, PCIe, USB, AXI)
— Provide API calls for block-specific command and direct access to FPGA internal register
— Ensure data type matching and checking of flow error

To operate UHD needs two components specific to each RFNoC block: a block declaration and a block controller.

Block declaration: a block declaration is an xml file containing a description of the FPGA blocks:

— Its unique NoC-ID to distinguish it from other blocks
— Input- and output ports description along with the associated data types, vector length, packet size
— Readback registers to provide asynchronous monitoring link from FPGA to host
— Settings registers to provide asynchronous command link from host to FPGA

Block controller: It allows having a user interface for each block to simplify its use. A default controller configurable through XML is present and can address a vast majority of block use case. But in some specific cases, when complex operations are required in the controller that is easier expressed in C++ code than XML, a custom controller can be used instead. However, writing custom block controllers requires recompilation of UHD, whereas default controller is interpreted at runtime via XML files.

At each startup, UHD entails a lookup process. It will first query the NoC-ID of each block. It will then look up the block declaration associated with each NoC-ID and then will associate each RFNoC block its corresponding block controller.

1.3 High-level integration

The final layer corresponds to the application code, and consists in using, sending and receiving data to the SDR using UHD calls (the latter performing all the transport and configuration layer automatically.)

It is possible to design an application in several different languages as long as there is a gateway between UHD (in C++) and the language used. It is straightforward to use a C/C++

code to handle the SDR but as indicated in the Section 5-2, there is a *two-language problem*. It is also an option to use the well-known GNU Radio Companion through the use of additional configuration files. For this thesis, the language chosen is Julia and does not require additional development for any new application because a gateway with UHD is already implemented through *AbstractSDRs.jl*.

2 Network-on-Chip aspect

As its name indicates, RFNoC proposes a Network-on-Chip architecture, that uses a bus-centric paradigm (Section 5-1.3-1). This bus is called crossbar in the RFNoC stack, and its operation is described in Fig. B-2.

A crossbar is a link between blocks, it is set up through UHD directives. Up to 256 unique crossbars can exist in an FPGA design, and with up to 16 ports per crossbar can be configured. This way, it is possible to stream between two or more blocks, for example, to realize the sum of block outputs. It is up to the crossbars to link the different blocks to the output interface of the SDR and control data. To ensure that all the exchanges are carried out correctly, each block must use the same protocol (Compressed HeaDeR (CHDR)), and as seen before a wrapper (provided by Ettus and universal) is used for this purpose: the NoC Shell.

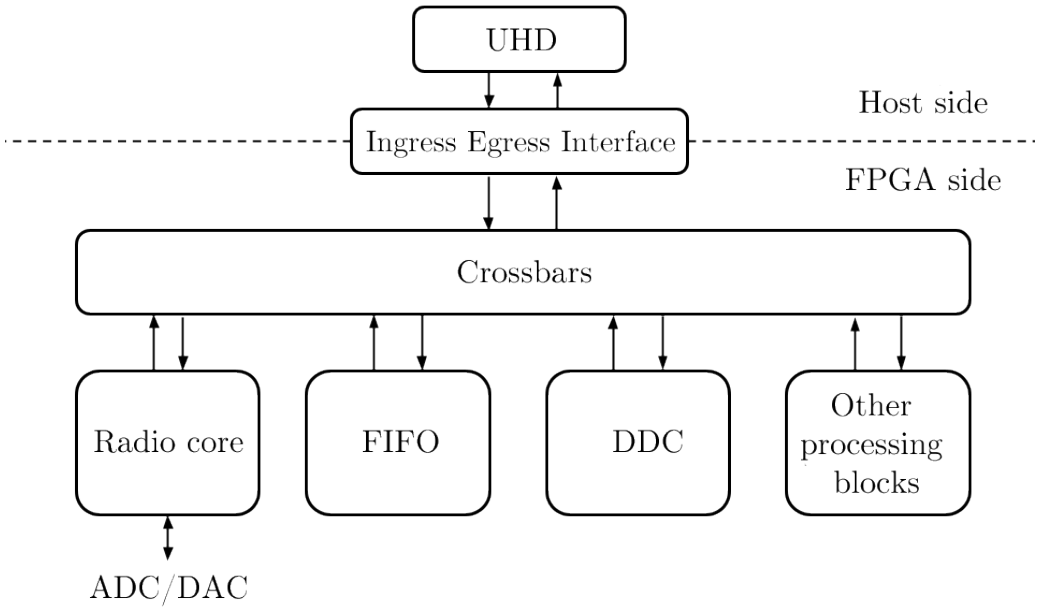


Figure B-2 – Crossbar fuctionning example.

3 NoC shell

The last contribution brought by RFNoC is the NoC Shell which allows interfacing the different IP processing with the crossbar (see Fig. B-3). It allows converting AXI streams (from user IP) into CHDR while performing flow control and integrity check operations. It is possible to send and receive special asynchronous messages via registers in the FPGA from the host. Thanks to the NoC Shell, any IP having as input/output interface an AXI bus can be implemented quickly within an SDR. This NoC Shell does not require any particular modification and this even if the block used has a custom block controller.

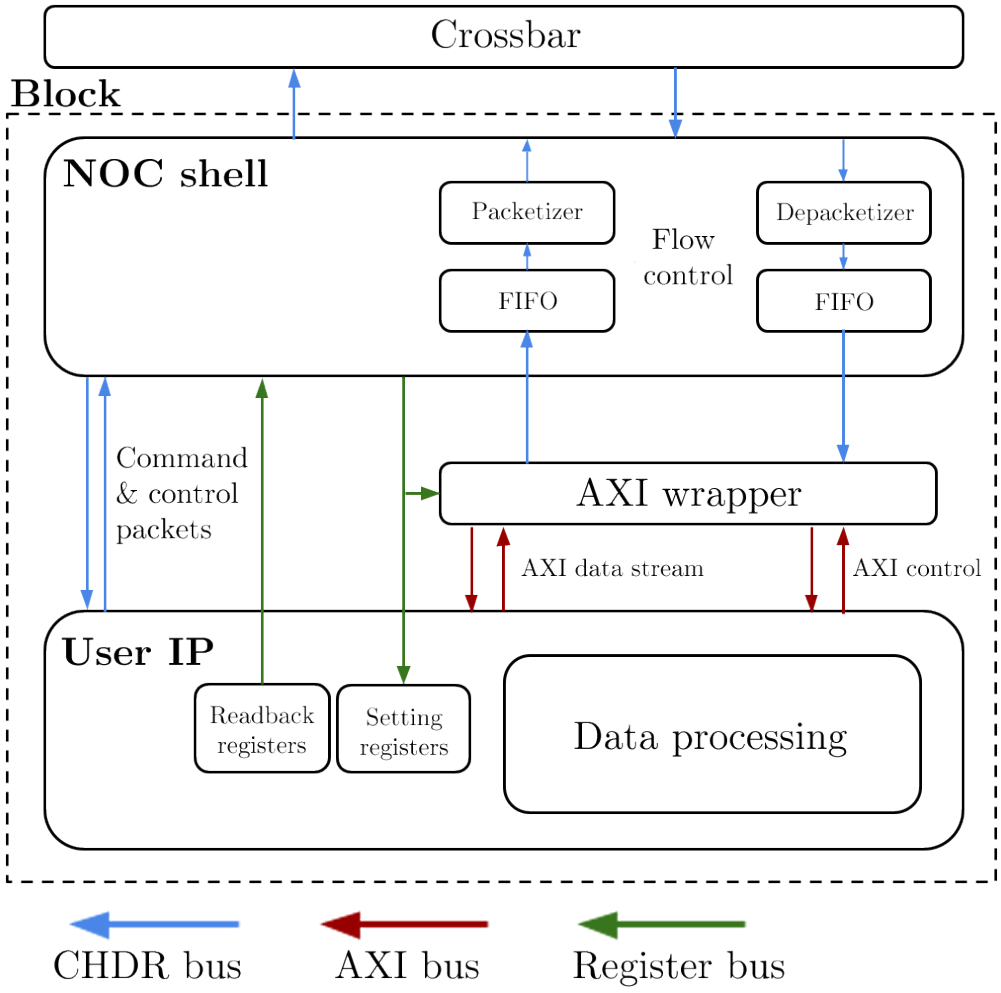


Figure B-3 – NoC Shell innerview.

Appendix C

EXPERIMENTAL SETUP

The equipment that is used in this Chapter 6 is listed below:

CPU	intel i7-8850H
RAM	32 Go
OS	Ubuntu 20.04
Vectorization instruction	SSE4.2, AVX2
Julia version	2.5

Table C-1 – Host computer.

Model	USRP X310
FPGA	KINTEX7-410T
Sampling rate	200 MS/s per channel (16-bit)
Radio frontend	2 UBX 160, 2 Rx, 2 Tx
Frequency range	10-6000 MHz
SDR to host link	SFP 10 GbE

Table C-2 – SDR.

Model	Agilent N5182A
Bandwidth	125 MHz
Frequency range	100 kHz to 6 GHz
Frequency sweep	100 μs per step

Table C-3 – Vector Signal Generator.

Models with antennas	Nordic PCA10059 Nordic nRF52832
Wired models	ESP32 Nordic nRF52832

Table C-4 – BLE devices.

PUBLICATIONS

[Lav20a] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Interception of Frequency-Hopping Signals for TEMPEST Attacks », *in Proc. Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, Virtuelle, France, Dec. 2020.

[Lav20b] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Towards Real Time Interception of Frequency Hopping Signals », *in Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, Sept. 2020, pp. 1–6.

[Lav21a] Lavaud, C., Gerzaguet, R., Gautier, M., and Berder, O., « AbstractSDRs: Bring down the two-language barrier with Julia Language for efficient SDR prototyping », *in IEEE Embedded Systems Letters* (2021).

[Lav21b] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Whispering Devices: A Survey on How Side-channels Lead to Compromised Information », *in Journal of Hardware and Systems Security (HASS)*, Mar. 2021.

BIBLIOGRAPHY

[Abo11] Aboulkassimi, D., Agoyan, M., Freund, L., Fournier, J., Robisson, B., and Tria, A., « ElectroMagnetic analysis (EMA) of software AES on Java mobile phones », *in Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov. 2011 (cit. on pp. 40, 48, 183).

[Agr03] Agrawal, D., Archambeault, B., Josyula., R., and Rohatgi, P., « The EM Side-Channel(s) », *in Proc. Cryptographic Hardware and Embedded Systems (CHES)*, Feb. 2003, pp. 29–45 (cit. on pp. 31, 40, 48, 183).

[Aic15] Aichinger, B., « DDR memory errors caused by Row Hammer », *in Proc. IEEE High Performance Extreme Computing Conference (HPEC)*, Sept. 2015 (cit. on p. 26).

[Al-21] Al-amaireh, H. and Kollár, Z., « Optimization of Hopping DFT for FS-FBMC Receivers », *in Signal Processing* 182 (May 2021), p. 107983 (cit. on p. 74).

[Ala18] Alam, M., Khan, H. A., Dey, M., Sinha, N., Callan, R., Zajic, A., and Prvulovic, M., « One&Done: A Single-Decryption EM-Based Attack on OpenSSL's Constant-Time Blinded RSA », *in USENIX Security Symposium*, Aug. 2018, pp. 585–602 (cit. on pp. 40, 48, 182).

[Ali17] Ali, K., Liu, A. X., Wang, W., and Shahzad, M., « Recognizing Keystrokes Using WiFi Devices », *in IEEE Journal on Selected Areas in Communications* 35.5 (May 2017), pp. 1175–1190 (cit. on pp. 40, 55, 180).

[Ana18] Anand, S. A. and Saxena, N., « Keyboard Emanations in Remote Voice Calls », *in Proc. ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2018 (cit. on pp. 32, 35, 37, 180).

[Aso04] Asonov, D. and Agrawal, R., « Keyboard acoustic emanations », *in Proc. IEEE Symposium on Security and Privacy (SP)*, June 2004 (cit. on pp. 32, 35, 37, 40, 45, 179, 181).

[Bac08] Backes, M. and Unruh, D., « Compromising Reflections-or-How to Read LCD Monitors around the Corner », *in Proc. IEEE Symposium on Security and Privacy (SP)*, May 2008 (cit. on pp. 32, 37, 178).

[Bac09] Backes, M., Chen, T., Duermuth, M., Lensch, H. P., and Welk, M., « Tempest in a Teapot: Compromising Reflections Revisited », *in Proc. IEEE Symposium on Security and Privacy (SP)*, May 2009 (cit. on pp. 32, 37, 179).

[Bac10] Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., and Sporleder, C., « Acoustic Side-Channel Attacks on Printers », *in USENIX Security Symposium*, Sept. 2010, pp. 307–322 (cit. on pp. 32, 34, 179).

[Bag11] Bagayoko, A., Fijalkow, I., and Tortelier, P., « Power Control of Spectrum-Sharing in Fading Environment With Partial Channel State Information », *in*59.*5* (May 2011), pp. 2244–2256 (cit. on p. 63).

[Bal19] Balagani, K. et al., « PILOT: Password and PIN Information Leakage from Obfuscated Typing Videos », *in*(Mar. 2019), arXiv: `http://arxiv.org/abs/1904.00188v2` (cit. on pp. 32, 38, 179).

[Bar09] Barisani, A. and Bianco, D., « Side Channel Attacks Using Optical Sampling Of Mechanical Energy And Power Line Leakage », *in defcon 17*, 2009 (cit. on pp. 32, 37, 179).

[Bej12] Bejarano, J. M. R., Yun, A., and Cuesta, B. D. L., « Security in IP satellite networks: COMSEC and TRANSEC integration aspects », *in Proc. Advanced Satellite Multimedia Systems Conference (ASMS) and Signal Processing for Space Communications Workshop (SPSC)*, Sept. 2012 (cit. on p. 18).

[Ber00] Berder, O., Bouder, C., and Burel, G., « Identification of frequency hopping communications », *in Problems in Modern Applied Mathematics* (2000) (cit. on pp. 8, 65, 70).

[Bes18] Besard, T., Foket, C., and Sutter, B. D., « Effective extensible programming: unleashing Julia on GPUs », *in IEEE Transactions on Parallel and Distributed Systems* 30.*4* (2018), pp. 827–841 (cit. on p. 119).

[Bez17a] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., « Julia: A fresh approach to numerical computing », *in SIAM review* 59.*1* (2017), pp. 65–98 (cit. on pp. 11, 117, 118).

[Bez17b] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., « Julia: A fresh approach to numerical computing », *in SIAM review* 59.*1* (2017), pp. 65–98 (cit. on p. 88).

[Bez18] Bezanson, J., Chen, J., Chung, B., Karpinski, S., Shah, V. B., Vitek, J., and Zoubritzky, L., « Julia: Dynamism and performance reconciled by design », *in Proc. of the ACM on Programming Languages* 2 (2018), pp. 1–23 (cit. on p. 117).

[Bih97] Biham, E. and Shamir, A., « Differential fault analysis of secret key cryptosystems », *in Proc. Advances in Cryptology (CRYPTO)*, Aug. 1997, pp. 513–525 (cit. on pp. 32, 34, 178).

[Bka11] Bkassiny, M., Jayaweera, S. K., and Avery, K. A., « Distributed Reinforcement Learning based MAC protocols for autonomous cognitive secondary users », *in Proc. Annual Wireless and Optical Communications Conference (WOCC)*, Apr. 2011 (cit. on p. 76).

[Bra16] Braun, M., Pendlum, J., and Ettus, M., « RFNoC: RF network-on-chip », *in Proc. GNU Radio Conference*, vol. 1, 1, 2016 (cit. on p. 120).

[Bre16] Brent, C. and Carlisle, A., « A Survey and Taxonomy Aimed at the Detection and Measurement of Covert Channels », *in Proc. 4th ACM Workshop on Information Hiding and Multimedia Security (IH MMSec)*, June 2016 (cit. on p. 17).

[Bri04] Brier, E., Clavier, C., and Olivier, F., « Correlation Power Analysis with a Leakage Model », *in Proc. Cryptographic Hardware and Embedded Systems (CHES)*, Jan. 2004, pp. 16–29 (cit. on pp. 32, 178).

[Bro29] Broertjes, W., « Method of maintaining secrecy in the transmission of wireless telegraphic messages », US1869659A, 1929 (cit. on p. 58).

[Cab04] Cabric, D., Mishra, S. M., and Brodersen, R. W., « Implementation issues in spectrum sensing for cognitive radios », *in Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, (ACSSC)*, 2004 (cit. on pp. 65, 69, 70).

[Cab06] Cabric, D., Tkachenko, A., and Brodersen, R., « Spectrum Sensing Measurements of Pilot, Energy, and Collaborative Detection », *in Proc. IEEE Military Communications conference MILCOM*, Oct. 2006 (cit. on pp. 70, 71).

[Cam18] Camurati, G., Poeplau, S., Muench, M., T.Hayes, and Francillon, A., « Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers », *in Proc. ACM conference on Computer and communications security (CCS)*, Oct. 2018 (cit. on pp. 40, 54, 180).

[Cam20] Camurati, G., Francillon, A., and Standaert, F.-X., « Understanding Screaming Channels: From a Detailed Analysis to Improved Attacks », *inIACR Transactions on Cryptographic Hardware and Embedded Systems* (June 2020), pp. 358–401 (cit. on p. 54).

[Cap16] Capriglione, D., Cerro, G., Ferrigno, L., and Miele, G., « Analysis and implementation of a wavelet based spectrum sensing method for low SNR scenarios », *in Proc. IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016 (cit. on p. 75).

[Che07] Chen, S., Zhang, T., and Xin, Y., « Relaxed K-Best MIMO Signal Detector Design and VLSI Implementation », *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 15.*3* (Mar. 2007), pp. 328–337 (cit. on p. 113).

[Che15] Chen, B., Yenamandra, V., and Srinivasan, K., « Tracking Keystrokes Using Wireless Signals », *in Proc. Annual International Conference on Mobile Systems, Applications, and Services - (MobiSys)*, Sept. 2015 (cit. on pp. 40, 55, 180).

[Che17] Cheng, Z., Song, T., Zhang, J., Hu, J., Hu, Y., Shen, L., Li, X., and Wu, J., « Self-organizing map-based scheme against probabilistic SSDF attack in cognitive radio networks », *in International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2017 (cit. on p. 76).

[Che18] Chen, Y., Li, T., Zhang, R., Zhang, Y., and Hedgpeth, T., « EyeTell: Video-Assisted Touchscreen Keystroke Inference from Eye Movements », *in IEEE Symposium on Security and Privacy (SP)*, May 2018 (cit. on pp. 32, 38, 179).

[Chi02] Chinnery, D. and Keutzer, K., « Closing the gap between ASIC and custom: tools and techniques for high-performance ASIC design », *in*2002 (cit. on p. 110).

[Chi20] Chinen, M., Lim, F. S. C., Skoglund, J., Gureev, N., O'Gorman, F., and Hines, A., « ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric », *in Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2020 (cit. on pp. 10, 100).

[Cho16] Choi, H.-J., Lee, H. S., Sim, D., Yook, J.-G., and Sim, K., « Reconstruction of leaked signal from USB keyboards », *in Proc. IEEE Asia-Pacific Radio Science Conference (RASC)*, Aug. 2016 (cit. on pp. 40, 46, 182).

[Cho20] Choi, J., Yang, H.-Y., and Cho, D.-H., « TEMPEST Comeback: A Realistic Audio Eavesdropping Threat on Mixed-signal SoCs », *inProc of the ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2020 (cit. on p. 54).

[Cla11] Clancy, T. C., Khawar, A., and Newman, T. R., « Robust Signal Classification Using Unsupervised Learning », *in IEEE Transactions on Wireless Communications* 10.*4* (Apr. 2011), pp. 1289–1299 (cit. on p. 76).

[Com17] Compagno, A., Conti, M., Lain, D., and Tsudik, G., « Don't Skype & Type! », *in Proc. ACM Asia Conference on Computer and Communications Security (ASIACCS)*, Apr. 2017 (cit. on pp. 32, 35, 37, 180).

[cry] cryptomuseum, *SINCGARS*, URL: `https://www.cryptomuseum.com/radio/sincgars/index.htm` (cit. on p. 59).

[Cui15] Cui, A., *Funtenna*, 2015, URL: `www.funtenna.org` (cit. on pp. 40, 47, 182).

[Dam15] Damavandi, M.-A. and Nader-Esfahani, S., « Compressive Wideband Spectrum Sensing in Cognitive Radio Systems Based on Cyclostationary Feature Detection », *in Proc. International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, Sept. 2015 (cit. on pp. 67, 70).

[Dar12] Dardaillon, M., Marquet, K., Risset, T., and Scherrer, A., « Software defined radio architecture survey for cognitive testbeds », *in Proc. 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug. 2012 (cit. on p. 113).

[Din13] Ding, G., Wu, Q., Yao, Y.-D., Wang, J., and Chen, Y., « Kernel-Based Learning for Statistical Signal Processing in Cognitive Radio Networks: Theoretical Foundations, Example Applications, and Future Directions », *in IEEE Signal Processing Magazine* 30.*4* (July 2013), pp. 126–136 (cit. on p. 76).

[Don06] Donoho, D. L., « For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution », *in Communications on Pure and Applied Mathematics* 59.*6* (2006), pp. 797–829 (cit. on p. 77).

[Dor17] Doré, J.-B., Gerzaguet, R., Cassiau, N., and Ktenas, D., « Waveform contenders for 5G: Description, analysis and comparison », *in Physical Communication* 24 (Sept. 2017), pp. 46–61 (cit. on p. 74).

[Du13] Du, Y. L., Lu, Y.-H., and Zhang, J.-L., « Novel Method to Detect and Recover the Keystrokes of PS/2 Keyboard », *in Progress In Electromagnetics Research C* 41 (June 2013), pp. 151–161 (cit. on pp. 31, 32, 178).

[Eck85] Eck, W. V., « Electromagnetic radiation from video display units: An eavesdropping risk? », *in Computers & Security* 4.4 (Dec. 1985), pp. 269–286 (cit. on pp. 40, 42, 180).

[Eli12] Elibol, F., Sarac, U., and Erer, I., « Realistic eavesdropping attacks on computer displays with low-cost and mobile receiver system », *in Proc. European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 1767–1771 (cit. on pp. 40, 43, 181).

[F-B08] F.-Boroujeny, B., « Filter Bank Spectrum Sensing for Cognitive Radios », *in IEEE Transactions on Signal Processing* 56.5 (May 2008), pp. 1801–1811 (cit. on pp. 9, 73).

[Fen05] Feng, W., Balaji, P., Baron, C., Bhuyan, L., and Panda, D., « Performance Characterization of a 10-Gigabit Ethernet TOE », *in Proc. Symposium on High Performance Interconnects (HOTI)*, Aug. 17, 2005 (cit. on p. 116).

[Fri05] Frigo, M. and Johnson, S. G., « The Design and Implementation of FFTW3 », *in Proceedings of the IEEE* 93.2 (2005), pp. 216–231 (cit. on p. 122).

[Fri98] Frigo, M. and Johnson, S. G., « FFTW: An adaptive software architecture for the FFT », *in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 1998, pp. 1381–1384 (cit. on p. 119).

[Gar91] Gardner, W. A., « Exploitation of spectral redundancy in cyclostationary signals », *in Signal Processing Magazine* 8.2 (Apr. 1991), pp. 14–36 (cit. on pp. 8, 66, 70).

[GBP14] GBPPR, *TAWDRYYARD Experiments*, 2014, URL: https://www.qsl.net/n9zia/vision/index.html (cit. on p. 52).

[Ge16] Ge, Q., Yarom, Y., Li, F., and Heiser, G., « Your Processor Leaks Information - and There's Nothing You Can Do About It », *in*(Dec. 14, 2016), arXiv: abs/1612.04474v (cit. on p. 26).

[Gen14] Genkin, D., Shamir, A., and Tromer, E., « RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis », *in Proc. Advances in Cryptology (CRYPTO)*, Oct. 2014, pp. 444–461 (cit. on pp. 32, 35, 179).

[Gen18] Genkin, D., Pattani, M., Schuster, R., and Tromer, E., « Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels », *in*(Aug. 2018), arXiv: abs/1809.02629 (cit. on pp. 32, 35, 180).

[Gho11] Gholamipour, A. H., G., A., Celebi, H., Toreyin, B. U., Saghir, M. A. R., Kurdahi, F., and Eltawil, A., « Reconfigurable filter implementation of a matched-filter based spectrum sensor for Cognitive Radio systems », *in Proc. International Symposium of Circuits and Systems (ISCAS)*, May 2011 (cit. on pp. 68, 70).

[Glo02] Glossner, J., Raja, T., Hokenek, E., and Moudgill, M., « A multithreaded processor architecture for SDR », *in Information and Communications Magazine* 19.*11* (2002), pp. 70–84 (cit. on p. 113).

[Gol] Golmie, N., Rebala, O., and Chevrollier, N., « Bluetooth adaptive frequency hopping and scheduling », *in Proc. IEEE Military Communications Conference (MILCOM)* (cit. on p. 62).

[Gol15] Goller, G. and Sigl, G., « Side Channel Attacks on Smartphones and Embedded Devices Using Standard Radio Equipment », *in Proc. International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, July 2015, pp. 255–270 (cit. on pp. 40, 48, 54, 182).

[Goo13] Goodin, D., *Meet "badBIOS," the mysterious MacPC malware that jumps airgaps*, 2013, URL: `http://arstechnica.com/security/2013/10/meet-badbios-the-mysterious-mac-and-pc-malware-that-jumps-airgaps/` (cit. on pp. 40, 182).

[Gra13] Grayver, E., *Implementing Software Defined Radio*, Springer New York, 2013 (cit. on pp. 113, 115, 116).

[Gur14] Guri, M., Kedma, G., Kachlon, A., and Elovici, Y., « AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies », *in Proc. International Conference on Malicious and Unwanted Software (MALWARE)*, Oct. 2014 (cit. on pp. 40, 44, 181).

[Gur15] Guri, M., Kachlon, A., Hasson, O., Kedma, G., Mirsky, Y., and Elovici, Y., « GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies », *in USENIX Security Symposium*, Aug. 2015 (cit. on pp. 40, 47, 182).

[Gur16a] Guri, M., Hasson, O., Kedma, G., and Elovici, Y., « An optical covert-channel to leak data through an air-gap », *in Proc. IEEE Annual Conference on Privacy, Security and Trust (PST)*, Dec. 2016 (cit. on pp. 32, 37, 179).

[Gur16b] Guri, M., Monitz, M., and Elovici, Y., « USBee: Air-gap covert-channel via electromagnetic emission from USB », *in Proc. Annual Conference on Privacy, Security and Trust (PST)*, Dec. 2016 (cit. on pp. 40, 46, 182).

[Gur16c] Guri, M., Solewicz, Y. A., Daidakulov, A., and Elovici, Y., « Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers », *in*(June 2016), arXiv: abs/1606.05915 (cit. on pp. 32, 36, 179).

[Gur17a] Guri, M., Solewicz, Y., Zadov, B., Daidakulov, A., and Elovici, Y., « Acoustic Data Exfiltration from Speakerless Air-Gapped Computers via Covert Hard-Drive Noise ('DiskFiltration') », *in Proc. Computer Security (ESORICS)*, Aug. 2017, pp. 98–115 (cit. on pp. 32, 36, 179).

[Gur17b] Guri, M., Zadov, B., Daidakulov, A., and Elovici, Y., « xLED: Covert Data Exfiltration from Air-Gapped Networks via Router LEDs », *in*(June 2017), arXiv: abs/1706.01140 (cit. on pp. 32, 37, 178).

[Gur17c] Guri, M., Zadov, B., and Y., « LED-it-GO: Leaking (A Lot of) Data from Air-Gapped Computers via the (Small) Hard Drive LED », *in Detection of Intrusions and Malware, and Vulnerability Assessment* (Feb. 2017), pp. 161–184 (cit. on pp. 32, 37, 178).

[Gur18a] Guri, M., « BeatCoin: Leaking Private Keys from Air-Gapped Cryptocurrency Wallets », *in*(Apr. 2018), arXiv: abs/1804.08714 (cit. on p. 29).

[Gur18b] Guri, M. and Elovici, Y., « Bridgeware », *in Communications of the ACM* 61.4 (Mar. 2018), pp. 74–82 (cit. on pp. 17, 18, 31).

[Gur18c] Guri, M., Solewicz, Y., Zadov, B., Daidakulov, A., and Elovici, Y., « MOSQUITO: Covert Ultrasonic Transmissions between Two Air-Gapped Computers using Speaker-to-Speaker Communication », *in*(Mar. 2018), arXiv: abs/1803.03422 (cit. on pp. 32, 36, 179).

[Gur18d] Guri, M., Zadov, B., Daidakulov, A., and Elovici, Y., « ODINI : Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields », *in*(Feb. 2018), arXiv: abs/1802.02700 (cit. on pp. 40, 47, 182).

[Han13] Hanspach, M. and Goetz, M., « On Covert Acoustical Mesh Networks in Air », *in Journal of Communications* 8.11 (Nov. 2013), pp. 758–767 (cit. on pp. 32, 36, 179).

[Hay14] Hayashi, Y., Homma, N., Miura, M., T., and Sone, H., « A Threat for Tablet PCs in Public Space », *in Proc. ACM Conference on Computer and Communications Security (SIGSAC)*, Nov. 2014 (cit. on pp. 40, 43, 44, 181).

[Hay17] Hayashi, Y., Homma, N., Toriumi, Y., Takaya, K., and Aoki, T., « Remote Visualization of Screen Images Using a Pseudo-Antenna That Blends Into the Mobile Environment », *in IEEE Transactions on Electromagnetic Compatibility* 59.*1* (Aug. 2017), pp. 24–33 (cit. on pp. 32, 40, 43).

[Het19] Hettwer, B., Gehrer, S., and Güneysu, T., « Applications of machine learning techniques in side-channel attacks: a survey », *in Journal of Cryptographic Engineering* (Apr. 2019) (cit. on pp. 40, 56).

[Hey12] Heyszl, J., Mangard, S., Heinz, B., Stumpf, F., and Sigl, G., « Localized Electromagnetic Analysis of Cryptographic Implementations », *in Lecture Notes in Computer Science*, 2012, pp. 231–244 (cit. on pp. 40, 47, 178).

[Hin12] Hines, A. and Harte, N., « Speech intelligibility prediction using a Neurogram Similarity Index Measure », *in Speech Communication* 54.*2* (Feb. 2012), pp. 306–320 (cit. on p. 100).

[Hug05] Hughes-Jones, R., Clarke, P., and Dallison, S., « Performance of 1 and 10 Gigabit Ethernet cards with server quality motherboards », *in Future Generation Computer Systems* 21.*4* (Apr. 2005), pp. 469–488 (cit. on p. 116).

[Jie13] Jiemin, Z. and Yongmei, L., « The study of the standards architecture and the standards attributes based on EMC standards and TEMPEST standards in computer system », *in Proc. International Conference on Computer Science & Education (ICCSE)*, Apr. 2013 (cit. on p. 39).

[Joh85] Johnson, P., « New Research Lab Leads to Unique Radio Receiver », *in E-Systems Team* 5.*4* (May 1985), pp. 6–7 (cit. on p. 110).

[Jul20] Julia Telecom, *AbstractSDRs - Common API for Software Defined Radio*, https://github.com/JuliaTelecom/AbstractSDRs.jl, 2020 (cit. on p. 120).

[Kab02] Kabal, P., *An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality*, tech. rep., TSP Lab Technical Report, Dept. Electrical and Computer Engineering, McGill University, 2002, pp. 1–89 (cit. on p. 99).

[Kar11] Karrenberg, R. and Hack, S., « Whole-function vectorization », *in Proc. International Symposium on Code Generation and Optimization (CGO)*, 2011, pp. 141–150 (cit. on p. 124).

[Kha17] Khalifeh, A. F., Al-Tamimi, A.-K., and Darabkh, K. A., « Perceptual evaluation of audio quality under lossy networks », *in Proc. International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017 (cit. on p. 99).

[Kim10] Kim, J., Hyeon, S., and Choi, S., « Implementation of an SDR system using graphics processing unit », *in IEEE Communications Magazine* 48.*3* (Mar. 2010), pp. 156–162 (cit. on p. 113).

[Kin19] Kinugawa, M., Fujimoto, D., and Hayashi, Y., « Electromagnetic Information Extortion from Electronic Devices Using Interceptor and Its Countermeasure », *in Proc. IACR Cryptographic Hardware and Embedded Systems (CHES)* (2019) (cit. on pp. 53, 180).

[Ko05] Ko, C. C., W. Zhi, and Chin, F., « ML-based frequency estimation and synchronization of frequency hopping signals », *in IEEE Transactions on Signal Processing* 53.*2* (2005), pp. 403–410 (cit. on p. 80).

[Koc18] Kocher, P., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., and Yarom, Y., « Spectre Attacks: Exploiting Speculative Execution », *in*(Jan. 3, 2018), arXiv: `abs/1801.01203v1` (cit. on pp. 6, 26).

[Koc99] Kocher, P., Jaffeand, J., and Jun, B., « Differential Power Analysis », *in Proc. Advances in Cryptology (CRYPTO)*, Dec. 1999, pp. 388–397 (cit. on pp. 32, 34, 47, 178).

[Kuh02] Kuhn, M. G., « Optical Time-Domain Eavesdropping Risks of CRT Displays », *in Proc. IEEE Symposium on Security and Privacy (SP)*, May 2002 (cit. on pp. 32, 36, 178).

[Kuh04] Kuhn, M. G., « Electromagnetic Eavesdropping Risks of Flat-panel Displays », *in Proc. International Conference on Privacy Enhancing Technologies (PET)*, May 2004, pp. 88–107 (cit. on pp. 7, 40, 42, 180).

[Kuh06] Kuhn, M. G., « Eavesdropping attacks on computer displays », *in Proc. Information Security Summit (ISS)*, May 2006, pp. 24–25 (cit. on pp. 40, 43, 44, 180).

[Kum13] Kumar, K., Saravanan, R., and Muthaiah, R., « Cognitive radio spectrum sensing algorithms based on eigenvalue and covariance methods », *in*5 (Jan. 2013), pp. 594–601 (cit. on pp. 66, 70).

[Kum16] Kumar, A., Saha, S., and Bhattacharya, R., « Proc. Improved wavelet transform based edge detection for wide band spectrum sensing in Cognitive Radio », *in USNC-URSI Radio Science Meeting (NRSM)*, June 2016 (cit. on p. 76).

[laba] lab, ettus, *RFNoC*, URL: `https://kb.ettus.com/RFNoC` (cit. on pp. 128, 184).

[labb] lab, ettus, *Universal Hardware Driver wiki*, URL: `https://files.ettus.com/manual/` (cit. on p. 128).

[labc] lab, ettus, ed., *X410 SDR*, URL: `https://www.ettus.com/all-products/usrp-x410/` (cit. on p. 114).

[Lam41] Lamarr, H. and Antheil, G., « Secret communication system », US2292387A, 1941 (cit. on p. 58).

[Lat04] Lattner, C. and Adve, V., « LLVM: A compilation framework for lifelong program analysis & transformation », *in Proc. International Symposium on Code Generation and Optimization (CGO)*, 2004, pp. 75–86 (cit. on p. 117).

[Lav20a] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Interception of Frequency-Hopping Signals for TEMPEST Attacks », *in Proc. Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information*, Virtuelle, France, Dec. 2020 (cit. on p. 22).

[Lav20b] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Towards Real Time Interception of Frequency Hopping Signals », *in Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, Sept. 2020, pp. 1–6 (cit. on p. 22).

[Lav20c] Lavaud, C., GerzaguetR, R., Gautier, M., and Berder, O., *AbstractSDRsBenchmark repository*, https://github.com/RGerzaguet/AbstractSDRsBenchmark, 2020 (cit. on p. 123).

[Lav21a] Lavaud, C., Gerzaguet, R., Gautier, M., and Berder, O., « AbstractSDRs: Bring down the two-language barrier with Julia Language for efficient SDR prototyping », *in IEEE Embedded Systems Letters* (2021) (cit. on pp. 11, 23).

[Lav21b] Lavaud, C., Gerzaguet, R., Gautier, M., Berder, O., Nogues, E., and Molton, S., « Whispering Devices: A Survey on How Side-channels Lead to Compromised Information », *in Journal of Hardware and Systems Security (HASS)*, Mar. 2021 (cit. on p. 23).

[Le08] Le, T.-H., Canovas, C., and Clédière, J., « An overview of side channel analysis attacks », *in Proc. ACM symposium on Information, computer and communications security (ASIACCS)*, 2008 (cit. on pp. 32, 34, 47, 178).

[Lee04] Lee, J.-S., Park, J.-H., Kim, S.-W., Li, Y., and Ryu, H.-G., « Implementation of DSP-based digital receiver for the SDR application », *in Proc. Asia-Pacific Conference on Communications (APCC)*, Sept. 1, 2004 (cit. on p. 113).

[Lee19] Lee, H. S., Choi, D. H., Sim, K., and Yook, J.-G., « Information Recovery Using Electromagnetic Emanations From Display Devices Under Realistic Environment », *in Transactions on Electromagnetic Compatibility* 61.*4* (Aug. 2019), pp. 1098–1106 (cit. on pp. 40, 44, 181).

[Li10] Li, H., Liu, Y., and Zhang, D., « Dynamic spectrum access for cognitive radio systems with repeated games », *in Proc. IEEE International Conference on Wireless Communications, Networking and Information Security (WCINS)*, June 2010 (cit. on p. 76).

[Li16] Li, Y. and Peng, Q., « Achieving secure spectrum sensing in presence of malicious attacks utilizing unsupervised machine learning », *in IEEE Military Communications Conference (MILCOM)*, Nov. 2016 (cit. on p. 76).

[Li20] Li, A., Song, S. L., Chen, J., Li, J., Liu, X., Tallent, N. R., and Barker, K. J., « Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect », *in Transactions on Parallel and Distributed Systems* 31.*1* (Jan. 2020), pp. 94–110 (cit. on p. 116).

[Lip20] Lipp, M. et al., « Meltdown: Reading Kernel Memory from User Space », *in Communications of the ACM* 63.*6* (May 2020), pp. 46–56 (cit. on p. 6).

[Lou02] Loughry, J. and Umphress, D. A., « Information leakage from optical emanations », *in ACM Transactions on Information and System Security* 5.*3* (Aug. 2002), pp. 262–289 (cit. on pp. 32, 37, 178).

[Lv15] Lv, Q. and Gao, F., « Matched filter based spectrum sensing and power level recognition with multiple antennas », *in Proc. China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, July 2015 (cit. on pp. 68, 70).

[Man02] Mangard, S., « A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion », *in Proc. International Conference on Information Security and Cryptology (ICISC)*, Apr. 2002, pp. 343–358 (cit. on pp. 32, 33, 47, 178).

[Mar14] Marinov, M., « Remote video eavesdropping using a software-defined radio platform », MA thesis, St Edmund's College, 2014 (cit. on pp. 40, 43, 181).

[Mat16] Matyunin, N., Szefer, J., Biedermann, S., and Katzenbeisser, S., « Covert channels using mobile device's magnetic field sensors », *in Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2016 (cit. on pp. 40, 47, 182).

[Mat20] Matas, K., La, T., Grunchevski, N., Pham, K., and Koch, D., « Invited Tutorial », *in Proc. International Symposium on Field-Programmable Gate Arrays (ISFPGA)*, Feb. 2020 (cit. on p. 127).

[Mey11] Meynard, O., Réal, D., Flament, F., Guilley, S., Homma, N., and Danger, J., « Enhancement of simple electro-magnetic attacks by pre-characterization in frequency domain and demodulation techniques », *in Proc. Design, Automation Test in Europe (DATE)*, 2011, pp. 1–6 (cit. on pp. 40, 48, 54, 183).

[Mik14] Mikaeil, A. M., Guo, B., and Wang, Z., « Machine Learning to Data Fusion Approach for Cooperative Spectrum Sensing », *in International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Oct. 2014 (cit. on p. 76).

[Mis07] Mishra, S. M., S.Brink, Mahadevappa, R., and Brodersen, R. W., « Cognitive Technology for Ultra-Wideband/WiMax Coexistence », *in Proc. International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Apr. 2007 (cit. on p. 70).

[Mit02] Mitola, J., « Cognitive radio. An integrated agent architecture for software defined radio. », *in Ph.D. Dissertation, KTH* (2002) (cit. on pp. 8, 19, 62, 111).

[Mit13] Mitra, G., Johnston, B., Rendell, A. P., McCreath, E., and Zhou, J., « Use of SIMD Vector Operations to Accelerate Application Code Performance on Low-Powered ARM and Intel Platforms », *in Proc. IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, 2013, pp. 1107–1116 (cit. on p. 122).

[Mit92] Mitola, J., « Software radios-survey, critical evaluation and future directions », *in Proc. National Telesystems Conference (NTC)*, Aug. 1992 (cit. on pp. 11, 19).

[Mit93] Mitola, J., « Software radios: Survey, critical evaluation and future directions », *in IEEE Aerospace and Electronic Systems Magazine* 8.4 (Apr. 1993), pp. 25–36 (cit. on p. 110).

[Mol03] Moll, F., Roca, M., and Isern, E., « Analysis of dissipation energy of switching digital CMOS gates with coupled outputs », *in Microelectronics Journal* 34.*9* (Sept. 2003), pp. 833–842 (cit. on p. 45).

[Mur15] Muralidharan, A., Venkateswaran, P., Ajay, S. G., Prakash, D. A., Arora, M., and Kirthiga, S., « An adaptive threshold method for energy based spectrum sensing in Cognitive Radio Networks », *in Proc. International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Dec. 2015 (cit. on pp. 65, 70).

[Nie17] Nie, G., Ding, G., Zhang, L., and Wu, Q., « Byzantine Defense in Collaborative Spectrum Sensing via Bayesian Learning », *in IEEE Access* 5 (2017), pp. 20089–20098 (cit. on p. 76).

[NSA] NSA, *National Security Telecommunications and Information Systems Security Advisory Memorandum NSTISSAM TEMPEST/2-95: RED/BLACK Installation Guidance*, transcript: http://cryptome.org/tempest-2-95.htm (cit. on pp. 26, 40, 48).

[NSA07] NSA, *TEMPEST: A Signal Problem*, transcript: https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/cryptologic-spectrum/tempest.pdf, Sept. 2007 (cit. on pp. 40, 45, 181, 182).

[NSA08] NSA, *VAGRANT program*, Catalogue extract: http://cryptome.org/2013/12/nsa-catalog-appelbaum.pdf, July 2008 (cit. on pp. 28, 40, 52, 180).

[NSA75] NSA, *National COMSEC/EMSEC Information Memorandum NACSEM-5112: NONSTOP Evaluation Techniques*, Partially declassified transcript: http://cryptome.org/nacsem-5112.htm, Apr. 1975 (cit. on p. 40).

[NSA82] NSA, *NACSIM 5000: Tempest Fundamentals. National Security Agency*, Partially declassified transcript: http://cryptome.org/jya/nacsim-5000/nacsim-5000.htm, Feb. 1982 (cit. on pp. 26, 39, 40, 48, 163).

[NSA92] NSA, *National Security Telecommunications and Information Systems Security Advisory Memorandum NSTISSAM TEMPEST/1-92: Compromising Emanations Lab- oratory Test Requirements*, Partially declassified transcript: http://cryptome.org/nsa-tempest.htm, Dec. 1992 (cit. on pp. 39, 40).

[NSA93] NSA, *National Security Telecommunications and Information Systems Security Instruction NSTISSI No. 7000: TEMPEST Countermeasures for Facilities*, Partially declassified transcript: http://cryptome.org/nstissi-7000.htm, Nov. 1993 (cit. on p. 40).

[NSA94] NSA, *Specification NSA No. 94-106: Specification for Shielded Enclosures*, Transcript: http://cryptome.org/nsa-94-106.htm, Aug. 1994 (cit. on p. 40).

[One04] Oner, M. and Jondral, F., « Cyclostationarity based air interface recognition for software radio systems », *in Proc. Radio and Wireless Conference (RAWCON)*, 2004 (cit. on pp. 67, 70).

[Oss14] Ossmann, M., « The NSA Playset: RF Retroreflectors », *in DEF CON 22*, Aug. 2014 (cit. on p. 52).

[Par07] Park, C.-H., Kim, S.-W., Lim, S.-M., and Song, M.-S., « HMM Based Channel Status Predictor for Cognitive Radio », *in Asia-Pacific Microwave Conference (APMC)*, Dec. 2007 (cit. on p. 76).

[Pfl08] Pfleeger, S. L. and Rue, R., « Cybersecurity Economic Issues: Clearing the Path to Good Practice », *in IEEE Software* 25.*1* (Jan. 2008), pp. 35–42 (cit. on p. 25).

[Qua09] Quan, Z., Cui, S., Sayed, A., and Poor, H., « Optimal Multiband Joint Detection for Spectrum Sensing in Cognitive Radio Networks », *in IEEE Transactions on Signal Processing* 57.*3* (Mar. 2009), pp. 1128–1140 (cit. on p. 73).

[Qui01] Quisquater, J.-J. and Samyde, D., « ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards », *in Proc. International Conference on Research in Smart Cards: Smart Card Programming and Security (E-SMART)*, Sept. 2001, pp. 200–210 (cit. on pp. 40, 47, 54, 182).

[Rob10] Robert, F., Diet, A., Villegas, M., Epifano, F., Cathelin, P., Triaire, P., and Baudoin, G., « Architecture and filtering requirements for fully digital multi-radio transmitters », *in Proc. IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, Sept. 2010 (cit. on p. 111).

[Rod11] Rodriguez, A. S., Mensinger, M. C., Ahn, I. S., and Lu, Y., « Model-based software-defined radio(SDR) design using FPGA », *in Proc. International Conference on Electro-Information Technology (IET)*, May 2011 (cit. on p. 114).

[roh] rohde-schwarz, *R&S FSWT Test Receiver*, URL: `https://www.rohde-schwarz.com/fr/produit/fswt-page-de-demarrage-produits_63493-310144.html` (cit. on p. 73).

[Sal15] Salahdine, F., Ghazi, H. E., Kaabouch, N., and Fihri, W. F., « Matched filter detection with dynamic threshold for cognitive radio networks », *in Proc. International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Oct. 2015 (cit. on pp. 68, 70).

[Say18] Sayakkara, A., Le-Khac, N., and Scanlon, M., « Accuracy Enhancement of Electromagnetic Side-Channel Attacks on Computer Monitors », *in Proc. International Conference on Availability, Reliability and Security (ARES)*, Aug. 2018 (cit. on pp. 40, 44, 181).

[Sch07] Schmid, T., Sekkat, O., and Srivastava, M. B., « An experimental study of network performance impact of increased latency in software defined radios », *in Proc. international workshop on Wireless network testbeds, experimental evaluation and characterization - WinTECH*, 2007 (cit. on p. 116).

[Sch16] Schulz, M., Klapper, P., Hollick, M., Tews, E., and Katzenbeisser, S., « Trust The Wire, They Always Told Me! », *in Proc. ACM Conference on Security & Privacy in Wireless and Mobile Networks - (WiSec)*, 2016 (cit. on pp. 40, 46, 182).

[Sch76] Schwartz, B. and Robbins, H., « Chemical etching of silicon: IV. Etching technology », *in Journal of the electrochemical society* 123.*12* (1976), p. 1903 (cit. on p. 110).

[Sea] Seagate, *Seagate Firecuda*, URL: `https://www.seagate.com/fr/fr/internal-hard-drives/ssd/firecuda-ssd/` (cit. on p. 116).

[Sep14] Sepetnitsky, V., Guri, M., and Elovici, Y., « Exfiltration of Information from Air-Gapped Machines Using Monitor's LED Indicator », *in Proc. IEEE Joint Intelligence and Security Informatics Conference (JISIC)*, Sept. 2014 (cit. on pp. 32, 37, 178).

[Shi14] Shi, J., Huang, W.-q., Wei, D., and Sun, D.-g., « A novel method for computer video leaking signal detection », *in Proc. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug. 2014 (cit. on pp. 40, 43, 181).

[Shm05] Shmilovitz, D., « On the definition of total harmonic distortion and its effect on measurement interpretation », *in IEEE Transactions on Power Delivery* 20.*1* (Jan. 2005), pp. 526–528 (cit. on p. 98).

[Sim16] Sim, D.-J., Lee, H. S., Yook, J.-G., and Sim, K., « Measurement and analysis of the compromising electromagnetic emanations from USB keyboard », *in Proc. IEEE Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC)*, May 2016 (cit. on pp. 40, 46, 182).

[Sir10] Sirotiya, M. and Banerjee, A., « Detection and estimation of frequency hopping signals using wavelet transform », *in UK-India-IDRC International Workshop on Cognitive Wireless Systems (UKIWCWS)*, 2010, pp. 1–5 (cit. on p. 80).

[Smi99] Smith, J. and Abel, J., « Bark and ERB bilinear transforms », *in IEEE Transactions on Speech and Audio Processing* 7.6 (1999), pp. 697–708 (cit. on p. 99).

[Smu90] Smulders, P., « The threat of information theft by reception of electromagnetic radiation from RS-232 cables », *in Computers & Security* 9.1 (Feb. 1990), pp. 53–58 (cit. on pp. 40, 45, 181).

[Sor88] Sorensen, H. and Burrus, C., « Efficient computation of the short-time fast Fourier transform », *in International Conference on Acoustics, Speech, and Signal Processing*, 1988, 1894–1897 vol.3 (cit. on p. 80).

[Spo17] Spolaor, R., Abudahi, L., Moonsamy, V., Conti, M., and Poovendran, R., « No Free Charge Theorem: A Covert Channel via USB Charging Cable on Mobile Devices », *in Applied Cryptography and Network Security*, 2017, pp. 83–102 (cit. on pp. 31, 32, 178).

[Su17] Su, Y., Genkin, D., Ranasinghe, D., and Yarom, Y., « USB Snooping Made Easy: Crosstalk Leakage Attacks on USB Hubs », *in USENIX Security Symposium*, Aug. 2017, pp. 1145–1161 (cit. on pp. 40, 49, 50, 183).

[Sug20] Sugawara, T., Cyr, B., Rampazzi, S., Genkin, D., and Fu, K., « Light Commands: Laser-Based Audio Injection on Voice-Controllable Systems », *in 29th USENIX Security Symposium*, 2020 (cit. on pp. 32, 38, 179).

[Sul15] Sullivan, D. T., *Survey of Malware Threats and Recommendations to Improve Cybersecurity for Industrial Control Systems*, tech. rep., US Army Research Laboratory, 2015 (cit. on p. 25).

[Sut07] Sutton, P. D., Lotze, J., Nolan, K. E., and Doyle, L. E., « Cyclostationary Signature Detection in Multipath Rayleigh Fading Environments », *in Proc. International Conference on Cognitive Radio Oriented Wireless Networks and Communications (crowncom)*, Aug. 2007 (cit. on p. 70).

[Sze18] Szefer, J., « Survey of Microarchitectural Side and Covert Channels, Attacks, and Defenses », *in Journal of Hardware and Systems Security* 3.3 (Sept. 2018), pp. 219–234 (cit. on p. 26).

[Tan05] Tang, H., « Some physical layer issues of wide-band cognitive radio systems », *in Proc. International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2005 (cit. on pp. 8, 65, 69–71).

[Thi01] Thiele, E., *Tempest for Eliza*, 2001 (cit. on pp. 40, 44, 181).

[Tka07] Tkachenko, A., Cabric, D., and Brodersen, R. W., « Cyclostationary Feature Detector Experiments Using Reconfigurable BEE2 », *in Proc. International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Apr. 2007 (cit. on p. 70).

[Tor15] Torrieri, D., « Chapter 3 Frequency-Hopping Systems », *in Principles of Spread-Spectrum Communication Systems*, 2015, pp. 147–201 (cit. on p. 61).

[Tro04] Tromer, E., « Acoustic cryptanalysis: on nosy people and noisy machines », *in Eurocrypt Rump Session* (May 2004) (cit. on pp. 32, 35, 179).

[Tru13] Truong, N. B., Suh, Y.-J., and Yu, C., « Latency Analysis in GNU Radio/USRP-Based Software Radio Platforms », *in Proc. Military Communications Conference (MILCOM)*, Nov. 2013 (cit. on p. 116).

[Uni] Union, S., *The Thing*, URL: `cryptomuseum.com/covert/bugs/thing/index.htm#ref` (cit. on pp. 40, 51, 180).

[Urk67] Urkowitz, H., « Energy detection of unknown deterministic signals », *in Proc. of the IEEE* 55.4 (1967), pp. 523–531 (cit. on pp. 8, 64, 70).

[Val01] Valkama, M., « Advanced I/Q signal processing for wideband receivers models and algorithms », PhD thesis, Tampere University of Technology, 2001 (cit. on p. 112).

[Val08] Valerio, D., « Open source software-defined radio: A survey on gnuradio and its applications », *in Forschungszentrum Telekommunikation Wien* (2008) (cit. on p. 129).

[Vua09] Vuagnoux, M. and Pasini, S., « Compromising Electromagnetic Emanations of Wired and Wireless Keyboards », *in USENIX Security Symposium*, Aug. 2009, pp. 1–16 (cit. on pp. 40, 45, 181).

[Wak18] Wakabayashi, S., Maruyama, S., Mori, T., Goto, S., Kinugawa, M., Hayashi, Y.-I., and Smith, M., « A Feasibility Study of Radio-frequency Retroreflector Attack », *in USENIX Workshop on Offensive Technologies*, Aug. 2018 (cit. on pp. 40, 52, 180).

[Wan16a] Wang, D. and Yang, Z., « An novel spectrum sensing scheme combined with machine learning », *in International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2016 (cit. on p. 76).

[Wan16b] Wang, G., Zou, Y., Zhou, Z., Wu, K., and Ni, L. M., « We Can Hear You with Wi-Fi! », *in IEEE Transactions on Mobile Computing* 15.*11* (Nov. 2016), pp. 2907–2920 (cit. on pp. 40, 55, 180).

[War67] Ware, W. H., « Security and privacy in computer systems », *in Proc. Spring joint computer conference (SJCC)*, Apr. 1967 (cit. on pp. 40, 48, 183).

[Wei15] Wei, T., Wang, S., Zhou, A., and Zhang, X., « Acoustic Eavesdropping through Wireless Vibrometry », *in Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Sept. 2015 (cit. on pp. 40, 50, 54, 180).

[Wil15] Wilhelm, Z., *Wireless telegraphy*, 1915 (cit. on p. 58).

[Yan10] Yang, W., Lu, Y., and Xu, J., « Video information recovery from EM leakage of computers based on storage oscilloscope », *in Frontiers of Electrical and Electronic Engineering in China* 5.*2* (Apr. 2010), pp. 143–146 (cit. on pp. 40, 44, 181).

[Yaw16] Yawada, P. S. and Wei, A. J., « Cyclostationary Detection Based on Non-cooperative spectrum sensing in cognitive radio network », *in Proc. International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, June 2016 (cit. on pp. 66, 70).

[You10] You, J., Reiter, U., Hannuksela, M. M., Gabbouj, M., and Perkis, A., « Perceptual-based quality assessment for audio–visual services: A survey », *in Proc. Signal Processing: Image Communication (SP)*, vol. 25, 7, Aug. 2010, pp. 482–501 (cit. on p. 98).

[Yua17] Yuan, K., Grassi, F., Spadacini, G., and Pignari, S. A., « Crosstalk-Sensitive Loops and Reconstruction Algorithms to Eavesdrop Digital Signals Transmitted Along Differential Interconnects », *in IEEE Transactions on Electromagnetic Compatibility* 59.*1* (Feb. 2017), pp. 256–265 (cit. on pp. 40, 50, 183).

[Zay09] Zayen, B., Hayar, A., and Kansanen, K., « Blind Spectrum Sensing for Cognitive Radio Based on Signal Space Dimension Estimation », *in Proc. International Conference on Communications (ICC)*, June 2009 (cit. on pp. 66, 70).

[Zen08] Zeng, Y., Liang, Y. C., and Zhang, R., « Blindly Combined Energy Detection for Spectrum Sensing in Cognitive Radio », *in IEEE Signal Processing Letters* 15 (2008), pp. 649–652 (cit. on pp. 65, 70).

[Zha14a] Zhang, X., Chai, R., and Gao, F., « Matched filter based spectrum sensing and power level detection for cognitive radio network », *in Proc. Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2014 (cit. on pp. 8, 68, 70).

[Zha14b] Zhao, Y., Wu, Y., Wang, J., Zhong, X., and Mei, L., « Wavelet transform for spectrum sensing in Cognitive Radio networks », *in International Conference on Audio, Language and Image Processing*, July 2014 (cit. on pp. 9, 75).

[Zha16] Zhang, Y., Yang, W., and Cheng, Y., « Sparsity analysis of FH-BPSK signals via K-SVD dictionary learning », *in IEEE International Conference on Computer Communication and the Internet (ICCCI)*, 2016, pp. 310–315 (cit. on p. 80).

[Zha17a] Zhang, C., Zhang, H., Luo, J., and Du, Y., « TEMPEST in USB », *in Proc. IEEE International Symposium on Electromagnetic Compatibility (EMC)*, Oct. 2017 (cit. on pp. 40, 46, 182).

[Zha17b] Zhang, N., Lu, Y., Cui, Q., and Wang, Y., « Investigation of Unintentional Video Emanations From a VGA Connector in the Desktop Computers », *in IEEE Transactions on Electromagnetic Compatibility* 59.6 (Dec. 2017), pp. 1826–1834 (cit. on pp. 40, 43, 181).

[Zha18] Zhao, M., Li, T., Alsheikh, M. A., Tian, Y., Zhao, H., Torralba, A., and Katabi, D., « Through-Wall Human Pose Estimation Using Radio Signals », *in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018 (cit. on pp. 40, 55, 180).

[Zhu08] Zhu, J., Xu, Z., Wang, F., Huang, B., and Zhang, B., « Double Threshold Energy Detection of Cooperative Spectrum Sensing in Cognitive Radio », *in Proc. International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, May 2008 (cit. on pp. 65, 70).

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Systèmes reconfigurables pour l'interception de signaux sporadiques compromettants

Mot clés : Canaux auxiliaires, TEMPEST, Radio logicielle, Traitement temps réel

Résumé : Les canaux auxiliaires correspondent à la déviation d'une information de son chemin légitime par l'intermédiaire d'un ou plusieurs phénomènes physiques. Ces fuites sont capables de contourner certaines sécurités et d'impacter les systèmes de communication sécurisés.
Dans cette thèse, nous étudions les différents canaux auxiliaires existants et analysons leurs principales caractéristiques. Les fuites d'information sous forme d'ondes électromagnétiques apparaissent clairement comme les plus difficiles à percevoir et nous nous intéressons ici plus particulièrement aux canaux auxiliaires impactant les communications à saut de fréquence. Cette étude mène à la création d'un système d'interception de ces signaux en temps réel et large bande. L'utilisation de radio logicielle combinée au langage Julia a permis la réalisation et la validation du système d'interception qui par la suite a été utilisé pour confirmer la présence de canaux auxiliaires télécom sur des transmissions Bluetooth. Des compromissions ont ainsi été mises en évidence sur des composants sur puce à base de microcontrôleurs.

Title: Reconfigurable systems for the interception of compromising sporadic signals

Keywords: Side-channel, TEMPEST, Software defined radio, Real-time processing

Abstract: The side-channels are defined as the deviation of information from its legitimate path due to one or several physical phenomena. These leaks are able to bypass some security systems and to affect secure communication systems.
In this thesis, we investigate the various existing side-channels and discuss their main properties. Among the various leakages, electromagnetic waves appear to be the most difficult to perceive and we particularly focus here in the side-channel impacting frequency hopping signals. A real-time wideband interception system is proposed and used on frequency hopping transceivers. The proposed interception system is based on software defined radio and leverages hardware resources and a new software methodology based on Julia language. Finally, the existence of telecom side-channels in Bluetooth communications has been demonstrated on several microcontrollers chips, even in normal process functioning.