Périphériques sans fil NRF24 : écoute des communications

Alain Schneider
alain.schneider@cogiceo.com / alain.schneider@ozwald.fr

COGICEO

Résumé La famille « NRF24 » du constructeur norvégien « Nordic Semiconductor » regroupe des puces de communications utilisant la bande des 2.4GHz. Dans cette famille, le chipset NRF24L01+ connaît un succès important. Depuis 2010, ce composant intéresse les hackers, entre autre parce qu'il est embarqué dans la plupart des claviers et souris sans fils qui inondent le marché grand public. À l'heure actuelle, on dénombre trois méthodes d'écoute des communications de ce chipset. Nous proposons de reproduire ces attaques et de tester leurs limites.

1 Présentation du chipset

1.1 Le chipset NRF24L01+

Le chipset NRF24L01+ permet d'offrir à un montage électronique une couche de communication radio accessible via une interface SPI standard. La même puce permet de recevoir ou d'émettre, peut fonctionner sur plus d'une centaine de canaux, et consomme très peu d'électricité.

1.2 Utilisations du chipset

De par ses caractéristiques, ce chipset se retrouve dans de nombreuses applications embarquées. Il équipe des capteurs d'équipements de sports (cardiofréquencemètres...), des jouets radio-guidés, des télécommandes multimédia (lecteurs blu-ray...), des claviers et souris sans fils, etc.

Le cas qui nous intéresse est celui des claviers sans fil puisque la confidentialité des frappes est critique. Ce sujet a d'ailleurs déjà fait l'objet de recherches variées, les techniques présentées allant de l'analyse des fuites électromagnétiques des câbles PS/2 jusqu'à la mesure par laser des vibrations causées par les frappes clavier d'ordinateur portable [2]. Il est donc normal que l'étude des NRF24L01+ vienne compléter cette panoplie.

1.3 Format des paquets

Les puces NRF24L01+ modulent en GFSK sur des canaux démarrant à 2.4GHz et s'étalant sur 1MHz ou 2MHz. Sur ces canaux, deux protocoles sont utilisables : « ShockBurst™ » et « Enhanced ShockBurst™ ».

Protocole « ShockBurst™ »

Les paquets « ShockBurst™ » contiennent un préambule radio, l'adresse de destination, la charge utile, et un éventuel CRC.

Préambule	Adresse	Charge utile	CRC
1 octet	3-5 octets	0-32 octet(s)	0-2 octet(s)

Le préambule radio est une alternance de 0 et 1, le dernier bit étant l'inverse du premier bit de l'adresse. L'adresse est composée de 3 à 5 octets quelconques. La charge utile est d'une longueur fixe comprise entre 0 et 32 octets. Le CRC, facultatif, peut faire un ou deux octets.

Protocole « Enhanced ShockBurst™ »

Les chipset modernes de chez Nordic, dont le NRF24L01+, supportent également « Enhanced Shockburst™ ». Ces trames contiennent un préambule radio, l'adresse de destination, un en-tête, la charge utile, et un CRC facultatif. La charge utile peut-être d'une longueur fixe ou variable (auquel cas sa taille est renseignée dans 6 des 9 bits de l'en-tête).

Préambule	Adresse	En-tête	Charge utile	CRC
1 octet	3-5 octets	9 bits	0-32 octet(s)	0-2 octet(s)

2 Méthode d'écoute : Keykeriki !

2.1 Difficulté de l'écoute

En connaissant la configuration des équipements ciblés (canal utilisé, adresse de destination, taille du CRC, taille des paquets) il suffit de configurer un NRF24L01+ avec ces paramètres pour recevoir les paquets en même temps que le récepteur légitime. Il est donc important de savoir s'il est possible de deviner ces paramètres.

Une étude de 2010 [6] s'attaque au problème et propose d'utiliser une puce Amiccom A7125 générant un flux de bits continu via démodulation GFSK sur les même canaux que les NRF24L01+ puis de chercher, dans ce flux à 2Mbps, des séquences ressemblant à des paquets valides. Faute de puissance de calcul suffisante pour tester toutes les configurations possibles à chaque bit reçu, les auteurs cherchent juste une alternance typique d'un préambule radio puis une taille de paquet « bien connue » (0, 8, 16, ou 32 octets) là où serait l'en-tête Enhanced ShockBurst™. A chaque séquence valide, ils notent l'adresse du périphérique à qui serait destiné ce paquet.

Si une adresse apparait plusieurs fois ils considèrent que l'influence du bruit est exclue et que les paquets étaient bien valides.

La puissance de calcul requise pour un traitement à 2Mbps reste cependant importante pour un contexte embarqué, et en utilisant une « taille bien connue » comme empreinte les auteurs s'empèchent d'identifier les paquets de taille différente ou n'utilisant pas la fonctionnalité de taille variable (ces paquets ne renseignant pas leur taille dans l'en-tête).

2.2 Écoute des claviers et souris sans fil

Les concepteurs de Keykeriki ont constaté que :
- Les charges utiles de nombreux périphériques Microsoft comportent un en-tête, un paquet USBHID et une somme de contrôle.
- Les paquets USBHID des souris ne sont pas chiffrés, ceux des claviers sont chiffrés par un XOR trivialement réversible.

Ils sont ainsi parvenus à écouter et déchiffrer les communications des périphériques Microsoft dont ils disposaient en 2010. Le périphérique Microsoft que nous avons analysé (en 2014) n'utilise pas ce format et nous ne parvenons pas actuellement à déchiffrer ses communications. Les périphériques d'autres marques (Logitech, Rainbow, etc.) sont en cours d'analyse.

2.3 Extensions possibles

La piste des FPGA a été abordée par les concepteurs de Keykeriki pour remplacer l'ARM traitant le flux à 2Mbps mais aucune implémentation n'a été présentée. D'après nos tests, une telle implémentation ne serait pertinente que dans un contexte embarqué où le profil des paquets recherchés est déjà connu. Étant donné le poids des équipements [1] ce système peut être embarqué sur un drone de taille modeste et, ainsi, s'approcher d'étages élevés de tours de bureau ou de zones physiquement protégées.

Une autre piste que nous explorons consiste à remplacer l'ARM par un analyseur logique bas de gamme [2] envoyant le flux sur PC via USB. Cette configuration permet d'utiliser un système où beaucoup de puissance peut être obtenue à faible prix en contrepartie d'une perte en mobilité.

3 Méthode de Travis Goodspeed

L'année suivante, Travis Goodspeed publie une méthode d'écoute [7] sans Amiccom et n'utilisant qu'un NRF24L01+.

1. Notre système composé d'un Amiccom A7125 et d'un CycloneII pèse 28 grammes.
2. Par exemple, un clone des analyseurs salaea, basé sur une puce CY7C68013A.

3.1 Beaucoup de bruit pour deux astuces

La méthode repose sur deux astuces. La première consiste à renseigner une valeur interdite dans le registre qui définit la taille d'adresse. Ce registre fait 2 bits et seules les valeurs 1, 2, et 3 sont autorisées d'après la documentation (l'adresse faisant alors 2 octets plus la valeur du registre). En fixant ce registre à 0 le NRF24L01+ se comporte pourtant de façon normale et utilise une adresse de 2 octets.

03	SETUP_AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' – 5 bytes LSByte is used if address width is below 5 bytes

Figure 1. Extrait de la documentation Nordic

La seconde astuce consiste à utiliser le bruit ambiant. D'après les analyses de Travis, le bruit contient beaucoup de suites de 0 et d'alternances 0/1. Il espère donc que le bruit ambiant ressemble souvent à un préambule radio. La puce NRF24L01+ est donc configurée en Shockburst™, avec une adresse de deux octets 0x0055 (ou 0x00AA). Avec cette configuration, le bruit ambiant précédant un vrai paquet peut passer pour un préambule, le véritable préambule est alors interprété comme l'adresse de deux octets, et les 32 octets suivant ce préambule/adresse sont mis à disposition par la puce comme s'il s'agissait de la charge utile d'un paquet Shockburst™. Les 32 octets remontés contiennent en fait le paquet Enhanced Shockburst™ original dans son intégralité (adresse, en-tête, charge, éventuel CRC). La figure 2 résume la double interprétation qui peut être faite de la même suite de bits et que Travis exploite.

Interprétation légitime	bruit			préambule	adresse					Suite du paquet « Enhanced shockburst » : en-tête, charge utile, CRC	bruit		
Flux à 2Mbps	xx	55	00	55	01	02	03	04	05	YY...YY	xx	xx	xx
Interprétation de Travis	bruit	préambule	adresse		Charge utile de taille maximale d'un paquet Shockburst								

Figure 2. Interprétation à la Goodspeed

Cette méthode permet l'écoute de paquets Enhanced Shockburst™ sans connaitre leur adresse au préalable. Bien entendu, beaucoup de bruit est également relevé. Pour identifier les adresses valides le problème est similaire à celui posé par la méthode Keykeriki. Une solution proche fonctionne : seules les adresses qui se répètent sont considérées valides.

3.2 Reproduction de l'attaque

Après ré-implémentation de l'attaque de Travis sur un matériel différent (arduino à la place d'un goodfet [1]), nous avons été en mesure d'obtenir des résultats similaires, permettant d'identifier les adresses utilisées :

```
...
01 02 03 02 01 CF 05 1C 03 00 80 00 00 00 00 00 00 00 00 00 00 46 1C D3 ...
...
01 02 03 02 01 CF 05 1C 03 00 80 00 00 00 00 00 00 00 00 00 00 46 1F CF ...
...
```

Listing 1. Interception en méthode Goodspeed sur Arduino

3.3 Exploitation de l'attaque

Nous avons visé une souris Microsoft avec cette attaque et avons constaté que, comme notre cible de test, cette souris ne fait pas usage de la fonctionnalité de taille variable des paquets. En effet, les 6 bits d'en-tête valent 0b110011 dans tous les paquets interceptés. Cette constatation suggère une amélioration simple de la méthode d'identification de paquets dans Keykeriki : rajouter aux tailles « usuelles » de paquets (0, 8, 16 et 32) la valeur « 51 » (0b110011) qui semble indiquer « longueur fixe ».

4 Repousser les frontières du RTLSDR

4.1 Petit historique du RTLSDR

Historiquement, les étapes de modulation et de démodulation radio sont réalisées par du matériel dédié, comme les puces NRF24L01+. Il est néanmoins possible, sous réserve d'avoir un convertisseur Analogique/Numérique rapide, de réaliser ces étapes de façon logicielle. L'avantage de cette approche est l'adaptabilité à plusieurs types de modulations et de protocoles, l'inconvénient est le prix généralement très élevé du matériel.

En 2012, Antti Palosaari envoie un courriel [3] à une liste de diffusion expliquant qu'un récepteur TV USB bon marché peut être utilisé comme

radio logicielle. Bien que cette fonctionnalité était déjà connue [5], ce n'est qu'à partir de ce courriel que le RTL2382U devient célèbre comme radio logicielle USB à moins de 20$. Les applications les plus populaires de ces radio logicielle sont à ce jour les écoutes : des communication ADS-B, des bandes basses du GSM2, des satellites météo, etc.

4.2 RTLSDR et NRF24L01+

Les communications des NRF24L01+ sont nativement à l'abri puisqu'à 2.4GHz elles sont au delà du maximum atteint en RTLSDR (2.2GHz). Une méthode d'écoute de ces puces avec RTLSDR existe pourtant [4].

En effet, certains pays utilisent le système de réception de TV satellite « MMDS » où des modules « LNB » convertissent les signaux à plusieurs GHz en provenance des satellites en signaux de fréquence plus faible transportables sur des câbles classiques. L'article explique qu'en utilisant un LNB (coûtant moins de 20$) il est possible de capter la bande des 2.4GHz. Un code pour déchiffrer les paquets NRF24L01+ est disponible.

L'intérêt que nous voyons à cette approche (encore en test lors de la rédaction de cet article), par rapport à l'Amiccom, réside principalement dans les interfaces d'antenne. En effet, l'Amiccom est une puce conçue pour l'embarqué qui se trouve surtout dans des PCB basiques ; en revanche, les LNB sont conçus pour servir de récepteur TV et sont donc interfaçable avec des antennes de qualité. Nos tests de portée seront mis à disposition dès que des résultats quantifiées seront disponibles.

Références

1. Alain Schneider. NRF24L01+ sniffing for Arduino. `https://github.com/cogiceo`, 2014.
2. Andrea Barisani and Daniele Bianco. Sniffing Keystrokes With Lasers/Voltmeters. Black Hat USA, 2009.
3. Antti Palosaari. SDR FM demodulation. `http://comments.gmane.org/gmane.linux.drivers.video-input-infrastructure/44461`, 2012.
4. Cyber Explorer. Sniffing and decoding NRF24L01+ and Bluetooth LE packets for under $30. `http://blog.cyberexplorer.me/2014/01/sniffing-and-decoding-nrf24l01-and.html`, 2014.
5. rtlsdr.org wiki. History and discovery of rtlsdr. `http://rtlsdr.org/#history_and_discovery_of_rtlsdr`.
6. Thorsten Schroeder and Max Moser. Practical Exploitation of Modern Wireless Devices. CanSecWest, 2010.
7. Travis Goodspeed. Promiscuity is the nRF24L01+'s Duty. `http://travisgoodspeed.blogspot.fr/2011/02/promiscuity-is-nrf24l01s-duty.html`, 2011.