Analyse de sécurité des modems mobiles

Benoit Michau
benoit.michau@ssi.gouv.fr

ANSSI

Résumé Les modems mobiles sont présents partout, et de plus en plus utilisés à travers le monde. A ce jour, la plupart des analyses de sécurité publiques ont porté sur leurs interfaces GSM, et GPRS en moindre mesure. Cependant, la plupart d'entre eux supportent la 3G, et de plus en plus le LTE. Ces nouvelles technologies ont un niveau de sécurité bien supérieur à la 2G, mais induisent également une complexité accrue au sein des logiciels utilisés dans les modems. Il devient possible depuis quelques années, grâce au développement des équipements radio et de la radio logicielle, de mettre en place des moyens de test 3G et LTE à moindre coût. Ceci permet l'analyse du comportement des modems, et de leur sécurité, sur ces technologies.

Introduction

Cette présentation a pour but d'exposer des informations techniques visant à comprendre le fonctionnement des modems et terminaux mobiles, et plus largement des réseaux mobiles 2G, 3G et LTE. Nous souhaitons faciliter une meilleure compréhension de ces technologies en décrivant le mode de fonctionnement standard entre terminaux et réseaux mobiles ainsi que les outils et techniques employés pour évaluer la sécurité des interfaces de communications radio-mobiles.

La conduite de tests et d'analyses sur les modems constitue un élément important permettant de garantir le bon fonctionnement et la bonne sécurisation de ceux-ci ; cette démarche doit permettre de mettre au jour certains problèmes qui peuvent alors être corrigés par les fabricants et donner lieu à des mises à jour. La sécurité des communications de dizaines de millions d'utilisateurs et d'abonnés aux services mobiles peut ainsi être améliorée.

Dans un premier temps, les grandes lignes des normes et principes de communications mobiles sont présentées. Une annexe copieuse détaille également certains aspects des protocoles de signalisation et donne des exemples de messages échangés entre terminaux et réseaux. Ensuite, les différents constructeurs de modems cellulaires sont passés en revue, ainsi que les architectures matérielles sous-jacentes. Quelques méthodes d'analyses

statiques sont expliquées, afin d'aider à la compréhension des architectures matérielles et logicielles d'aujourd'hui. Puis, différents outils réseaux sont présentés : ceux existant sur Internet et plus particulièrement ceux disponibles en source ouverte, ainsi que certains développés par nos soins.

Enfin, quelques exemples de résultats issus de nos campagnes de test sont exposés : ceux-ci vont de la simple constatation du non-respect de la norme, pouvant dans certains cas avoir des conséquences graves sur la sécurité des communications, aux bogues d'implantation logicielle permettant une corruption mémoire.

1 Technologies mobiles

1.1 Principe général des communications mobiles

Les réseaux mobiles sont constitués de deux parties distinctes : le réseau d'accès radio (RAN : Radio Access Network) et le cœur de réseau (CN : Core Network). Le réseau d'accès radio prend en charge la connexion radio avec les terminaux, alors que le cœur de réseau permet l'établissement des services quelque soit la localisation et la mobilité du terminal au sein du réseau d'accès radio.

Le schéma 1 présente les principaux équipements de réseau d'accès 2G (GERAN, avec les BTS et BSC/PCU) et 3G (UTRAN, avec les NodeB et RNC) ainsi que les équipements de cœur de réseau en circuit (CS pour circuit switched, avec les MSC/VLR pour commuter les circuits voix, les SMS-C et HLR/AuC) et paquet (PS pour packet switched, avec les SGSN et GGSN pour router le trafic IP, et le HLR/AuC).

De manière générale, les terminaux ne sont pas connectés en permanence avec le réseau. Ils reçoivent des informations de diffusion émises par les antennes-relais sur un canal descendant, dit de broadcast : celles-ci permettent aux terminaux de connaître en permanence l'identification et la configuration des cellules des réseaux mobiles environnants. Lorsque les terminaux doivent se connecter à un réseau, ils suivent une procédure pour obtenir un canal duplex (montant et descendant) dédié avec l'antenne-relais la plus adaptée. Ils commencent par émettre une demande d'accès sur le canal montant, dit RACH ; si le réseau d'accès radio reçoit correctement cette demande, il renvoie au terminal les caractéristiques du canal dédié qui va pouvoir être utilisé pour transporter le service demandé par le terminal, cet envoi est fait sur le canal descendant dit de paging. Une fois un canal duplex dédié établi entre le terminal et le réseau d'accès radio, celui-ci peut communiquer avec le cœur de réseau pour réaliser le service

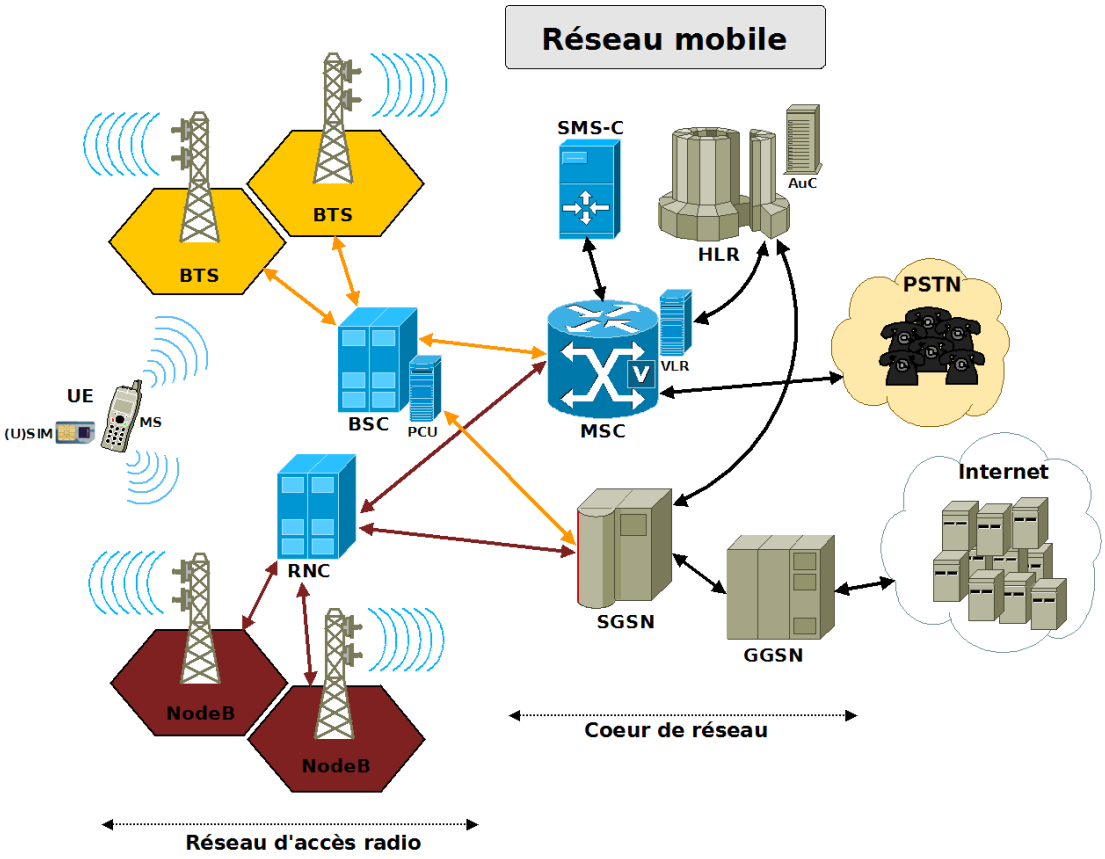

Figure 1. Architecture de base d'un réseau mobile 2G / 3G

adéquat (établissement d'appel voix, établissement d'une connexion de données, transfert d'un message court, *etc...*).

C'est lors de l'établissement de la communication entre un terminal et le cœur de réseau qu'ont lieu les procédures de sécurité : authentification, dérivation de clés et allocation d'identités temporaires. À la suite de l'authentification et de la dérivation de clés, la sécurisation du canal radio peut être mise en œuvre (chiffrement, contrôle d'intégrité cryptographique, *etc...*). Lorsqu'un terminal doit être joint (par exemple, lorsqu'il reçoit un appel ou un SMS), celui-ci est averti via un message contenant son identité temporaire sur le canal de paging. Lorsqu'il reçoit une telle notification, il peut alors suivre la procédure décrite plus haut pour se connecter au réseau d'accès radio puis au réseau cœur.

L'ensemble de ces communications entre terminal et RAN, et terminal et CN, suivent des protocoles standardisés, dits de signalisation. Ils permettent aux terminaux de se configurer correctement pour fonctionner avec les réseaux mobiles et de rendre les services demandés.

L'organisme principal aujourd'hui responsable de la maintenance des normes mobiles existantes et du développement des nouvelles est le 3GPP, mis en place lors du développement de la téléphonie mobile de 3ème génération. L'ensemble des normes mobiles (à l'exception des algorithmes de chiffrement GSM et GPRS) est publique et consultable sur le site Internet du 3GPP. Un tableau de synthèse permet de consulter les normes par domaine d'application [50].

Les paragraphes qui suivent donnent quelques explications sur l'historique et les différences qui existent entre les multiples technologies de réseaux mobiles, au-delà de la simple architecture d'un réseau 2G-3G tel qu'il peut exister aujourd'hui.

1.2 GSM

La première version de la norme GSM a été établie dès 1988, développée au sein de quelques états européens. La norme a été prise en charge par l'ETSI (European Telecommunications Standards Institute), et est aujourd'hui maintenue par le 3GPP. Le GSM est déployé à travers le monde entier, dans plus de 200 pays par plus de 800 opérateurs. La GSMA (Association des opérateurs GSM) a recensé plus de 3 milliards d'abonnements GSM dès 2008. Le GSM sera encore présent pour de nombreuses années à venir.

Une modulation numérique simple est utilisée pour la transmission radio : GMSK (Gaussian Minimum Shift Keying). Elle utilise une largeur de bande de 200 kHz, ce qui limite le débit d'information maximum d'un canal numérique GSM. Les utilisateurs multiples sont ordonnancés temporellement : c'est-à-dire que chaque canal radio de 200 kHz de bande passante est subdivisé en blocs de ressource temporelle (appelés également slot), chaque bloc pouvant être attribué à un terminal différent.

Le cœur de réseau GSM est composé de commutateurs de circuits numériques (MSC : Mobile Switching Center) associé à un gestionnaire de localisation (VLR : Visitor Location Register) afin de localiser et connecter les appels entre terminaux ; d'un SMS-Center, en charge de transmettre les messages courts émis par les mobiles attaché au réseau ; et d'un gestionnaire de localisation central (HLR pour Home Location Register, associé à un AuC, Authentication Center).

Les terminaux sont équipés d'une carte à puce SIM (Subscriber Identity Module) pour l'identification et l'authentification des terminaux au sein des réseaux mobiles. La carte SIM contient un IMSI (International Mobile Subscriber Identity) qui est un identifiant unique au monde pour chaque abonné mobile. Elle est capable de s'authentifier au réseau mobile grâce

au partage d'une clé symétrique avec le HLR-AuC et l'utilisation d'un algorithme de dérivation de clés commun avec cet équipement.

Chaque réseau cœur d'opérateur mobile est connecté à d'autres réseaux téléphoniques via des passerelles : des réseaux téléphoniques commutés fixes et / ou d'autres réseaux téléphoniques mobiles. Chaque opérateur mobile interconnecte ainsi ses équipements de cœur de réseau avec de nombreux autres opérateurs nationaux et internationaux pour transférer le trafic des abonnés, mais également pour échanger de la signalisation, dite SS7.

La technologie GSM a été développée afin de permettre les appels entre terminaux mobiles ; en Europe, la norme GSM utilise des canaux fréquentiels sur les bandes GSM-900 (autour de 900 MHz) et DCS-1800 (autour de 1.8 GHz). Le SMS a été développé en 1992, il utilise les canaux de signalisation afin de transférer des messages « utilisateurs ». Le transfert de données et de fax sur circuits GSM a été défini en 1994 : le débit de données s'élève alors à 9.6 kb/s !

1.3 GPRS

La conception du GPRS a pris plusieurs années : la première version de la norme GPRS est apparue en 1998. Elle est aujourd'hui classiquement associée au GSM, car elle réutilise une partie de ses concepts et de ses équipements ; elle est maintenue par le 3GPP.

Le GPRS (General Packet Radio Service) consiste à définir un nouveau mode, dit PS (Packet Switching) en opposition au mode CS (Circuit Switching) qui est le mode initialement développé pour le GSM. Le mode PS permet le transfert de données par paquet de manière asynchrone : bien adapté à une navigation Internet (ou WAP, dans les premiers temps !). Mais pour cela, des évolutions importantes sont réalisées par rapport au GSM.

Au niveau de l'interface radio, une nouvelle manière d'allouer les blocs de ressources temporelles, et d'ordonnancer les utilisateurs entre eux, est définie : ceci permet d'optimiser l'occupation du canal radio par de multiples utilisateurs, qui effectuent des transferts de paquets sans nécessiter un débit de données constant dans le temps.

Par ailleurs, de nouveaux équipements sont introduits dans le cœur de réseau : le SGSN (Serving GPRS Support Node), qui prend en charge la signalisation cœur de réseau vis à vis du terminal, gère sa mobilité, et fait transiter son trafic dans un tunnel GTP (GPRS Tunneling Protocol) ; et le GGSN (Gateway GPRS Support Node), qui termine les tunnels GTP et route le trafic des utilisateurs vers des réseaux de services (typiquement,

Internet) via des APN (Access Point Name). Le domaine PS introduit par le GPRS fonctionne, au niveau de la gestion de la mobilité et des procédures de sécurité, de manière parallèle et très similaire au domaine CS du GSM : ainsi les terminaux s'authentifient de manière indépendante pour le domaine CS et pour le domaine PS ; ils disposent également d'identités temporaires et de clés de chiffrement de l'interface radio différentes entre les interfaces GSM et GPRS.

Des interconnexions sont réalisées entre opérateurs mobiles au niveau national et international, afin de pouvoir router les données du domaine PS des abonnés mobiles, quel que soit le réseau mobile sur lequel ils sont attachés. La plupart des interconnexions entre opérateurs à l'international sont faites au niveau de zones dites GRX / IPX (GPRS eXchange / IP eXchange).

1.4 EDGE

EDGE (Enhanced Data rate for GSM Evolution) est essentiellement une évolution de l'interface radio GPRS, principalement au niveau de la modulation et des mécanismes de codage et de redondance des transmissions. La première norme EDGE a été développée en 2000, et est aujourd'hui maintenue par le 3GPP.

L'EDGE permet entre autre l'utilisation d'une modulation en phase à 8 états, dite 8-PSK ; ainsi, au lieu de transmettre 1 bit d'information par période du signal d'information (on parle également de « symbole ») avec une modulation GMSK, 3 bits d'information sont transmis, permettant ainsi de tripler le débit d'information par rapport à une interface radio GPRS classique. Cette modulation permet d'atteindre, pour un canal fréquentiel de 200 kHz, des débits théoriques ponctuels de 473 kb/s, alors que le GPRS ne délivre que 171 kb/s.

1.5 UMTS

Parallèlement à l'évolution des technologies GPRS et EDGE, le 3GPP (regroupement de multiples organismes de normalisation européen, américain et asiatique) développe dès 1999 la norme de téléphonie de 3ème génération. Il s'agit d'une refonte complète de l'interface radio, et d'une évolution du cœur de réseau GSM / GPRS.

Une interface radio complètement nouvelle est développée, afin de supporter les deux modes CS et PS de manière uniforme. Elle fonctionne sur des fréquences différentes du GSM : en Europe, la 3G utilise des fréquences porteuses autour de 2.1 GHz. Par ailleurs, le type de modulation

et de codage sont censés être optimisés pour le support de communication en mode paquet. Le fait est que de très nombreux brevets, de Qualcomm entre autre, s'appliquent à ces techniques de modulation et codage.

L'interface radio s'appuie sur une modulation, *a priori* simple, en décalage de phase de type QPSK sur une bande passante de 5MHz. Cependant, le principe de l'étalement spectral est très largement utilisé avec des mécanismes de convolution des signaux par codes orthogonaux et de scrambling par séquence spécifique, permettant le multiplexage des utilisateurs et la distinction entre les cellules, d'où la dénomination de CDMA (Code Division Multiple Access). Ainsi en 3G, l'ensemble des utilisateurs d'une cellule radio émettent et reçoivent des signaux sur la même fréquence et au même instant, chacun étalant ses données par un code propre. Avec cette modulation, le débit théorique du canal radio pour un utilisateur s'élève à 384 kb/s.

Le cœur de réseau, développé initialement pour le GSM pour la partie circuit CS et pour le GPRS pour la partie paquet PS, évolue afin de supporter des nouveaux codecs et de nouvelles procédures de mobilité. La procédure d'authentification évolue de manière importante avec l'introduction de cartes USIM permettant une authentification mutuelle entre le terminal et le réseau mobile.

Les interconnexions existantes pour les réseaux GSM et GPRS au niveau national et international sont réutilisées, afin de permettre les situations de roaming pour les abonnés 3G.

1.6 HSPA

Depuis 2003, des optimisations sont réalisées de manière incrémentale sur la norme UMTS. Ainsi le 3GPP a normalisé le HSDPA (High-Speed Downlink Packet Access) qui permet l'augmentation du débit de données sur les canaux radio descendants dédiés, le HSUPA (High-Speed Uplink Packet Access) qui permet de même l'augmentation du débit sur les canaux radio montants dédiés. Ces deux évolutions consistent entre autre à utiliser une modulation des signaux radio plus denses, dites 16-QAM (Quadrature Amplitude Modulation à 16 états, permettant de transmettre 4 bits par symbole).

Enfin le HSPA+ est une nouvelle amélioration sur les canaux radio permettant d'atteindre des débits encore supérieurs en introduisant l'utilisation de techniques de transmission MIMO (Multiple Input / Multiple Output), et en densifiant encore la modulation radio avec l'utilisation du 64-QAM.

Ces évolutions permettent à un utilisateur d'atteindre des débits descendants jusqu'à 42 Mb/s sur le canal radio. Cependant, en parallèle, le 3GPP a commencé à standardiser la norme destinée à devenir la 4G : le LTE.

1.7 LTE

La première version de la norme LTE (Long Term Evolution) est apparue en 2009, développée par le 3GPP. Pour cette nouvelle norme, tous les concepts du réseau mobile sont renouvelés : le réseau d'accès radio comme le cœur de réseau. Le mode circuit CS est abandonné : le réseau LTE est ainsi un réseau fonctionnant uniquement en mode paquet. Pour le support des communications vocales, une infrastructure VoIP (par exemple l'IMS qui s'appuie sur les protocoles SIP et RTP) doit être adjointe au réseau LTE.

La norme LTE est faite pour fonctionner sur un très grand nombre de fréquences porteuses : à partir de 400 MHz et jusqu'au-delà de 3 GHz. Cela permet d'utiliser la technologie LTE pour des types de couvertures différents : pour des cellules et des transmissions longue-distance (plusieurs dizaines de kilomètres) aux fréquences les plus basses, ou des cellules à grosse capacité en environnement urbain aux fréquences plus élevées, voire des petites cellules à l'intérieur des bâtiments aux fréquences les plus hautes.

L'architecture du réseau d'accès radio LTE est assez différente de l'existant en téléphonie mobile 2G et 3G : il n'y a plus de contrôleurs radio intermédiaires entre les antennes-relais et le cœur de réseau. Les antennes-relais LTE peuvent être interconnectées entre elles (via des interfaces dites X2) et se rattachent toutes directement au cœur de réseau LTE (via les interfaces dites S1). L'interface radio avec les terminaux utilise aussi des concepts différents par rapport aux technologies antérieures : la modulation radio et le multiplexage s'appuient sur l'OFDM (Orthogonal Frequency Division Multiplexing), qui consiste à « découper » le canal radio large bande en sous-canaux en bande étroite. Ces sous-canaux sont alors assignés dynamiquement et permettent le multiplexage des utilisateurs au sein d'une cellule. La bande passante d'un canal LTE large bande peut être de 5MHz, 10 MHz ou 20 MHz.

Au niveau du cœur de réseau, il y a une séparation des équipements selon leurs fonctions : le MME (Mobility Management Entity) traite uniquement de la signalisation (signalisation avec les antennes-relais, et avec les terminaux), alors que les S-GW (Serving-Gateway) et P-GW

(Packet-Gateway) agrègent et font transiter les données utilisateurs, toujours encapsulées dans des tunnels GTP. Le HLR hérite d'une nouvelle dénomination : HSS (Home Subscribers Server), mais conserve des fonctionnalités équivalentes. L'authentification au réseau LTE doit obligatoirement se faire avec une carte USIM (introduite avec les réseaux mobiles de 3ème génération). Le schéma 2 présente les différents équipements et interfaces au sein d'un réseau LTE : le terminal, le réseau d'accès radio constitué uniquement d'eNodeB, et le cœur de réseau.

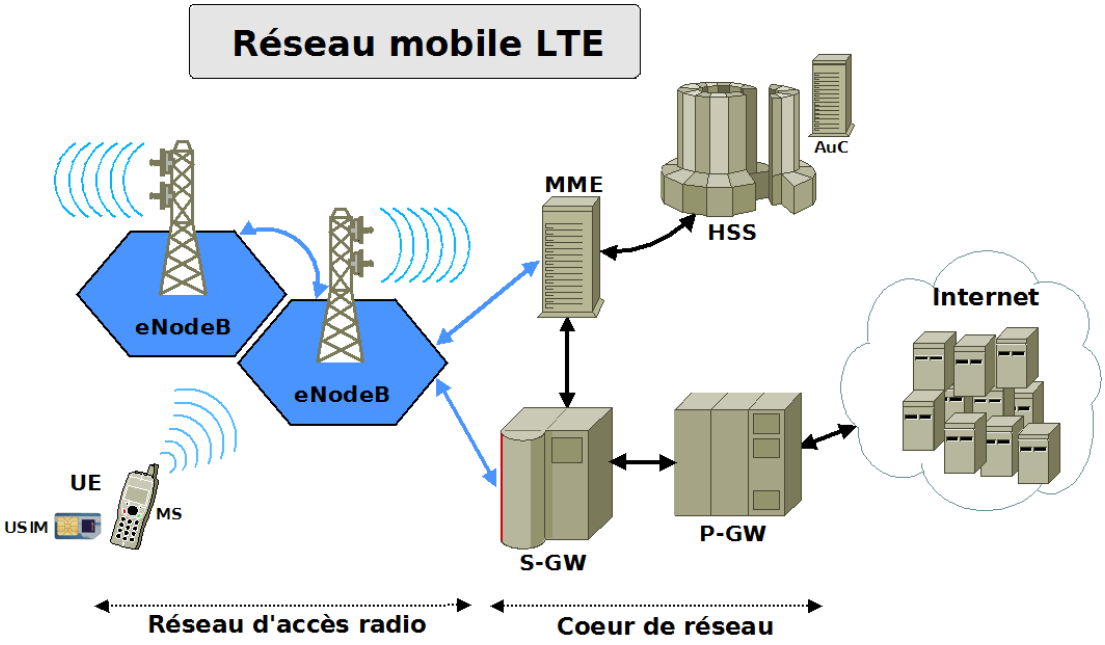

Figure 2. Architecture de base d'un réseau mobile LTE

Les interconnexions existantes pour les réseaux GPRS et 3G (éventuellement via les zones GRX / IPX) au niveau national et international sont réutilisées et évoluent, afin de permettre les situations de roaming pour les abonnés LTE.

1.8 Protocoles de signalisation

Comme indiqué à la fin du paragraphe 1.1, il y a deux types d'échanges au sein d'un réseau mobile : la signalisation, et les données utilisateurs. Ces deux types de données ne sont jamais mélangés, contrairement à certains protocoles Internet, comme par exemple HTTP qui embarque les données utilisateurs (couramment, les données HTML) directement sur les en-têtes HTTP. Étant donné l'historique et l'évolution des réseaux mobiles au cours

du temps, on constate une multiplication des protocoles de signalisation, ainsi que des niveaux de complexité parfois très élevés pour certains d'entre eux.

Des éditeurs de solutions de tests publient des posters permettant d'apprécier la complexité des piles protocolaires et des interfaces au sein des réseaux mobiles. Parmi ceux-ci, les constructeurs NetHawk, Agilent et Rohde&Schwarz sont assez connus pour publier des informations synthétiques.

Dans le cas de cette étude, l'interface entre le terminal mobile et le réseau est celle qui nous intéresse directement, puisque la plus exposée. L'interface radio est supportée par un type de modulation et de codage spécifique à la technologie utilisée (GSM / GPRS / EDGE / UMTS / HSPA / LTE). De plus, des mécanismes d'accès aux canaux, dits MAC (Media Access Control), et de gestion de liaison, dits RLC (Radio Link Control), sont mis en œuvre afin de pouvoir transférer de manière fiable des messages de signalisation entre terminaux et réseaux mobiles. Enfin, lesdits messages de signalisation sont structurés selon différents principes :

- ils contiennent (presque) tous un en-tête fixe de 2 octets référençant le type de protocole et le type de message ;
- ils contiennent une suite (normalement) ordonnée d'éléments d'information, spécifiques à chaque type de message ;
- les éléments d'information optionnels ou conditionnels sont préfixés par un champ de type « tag » de 8 bits ;
- les éléments d'information qui ont une longueur variable sont préfixés par un champ de type « length » de 8 ou 16 bits.

À l'exception de certains types de message de signalisation RRC, qui utilisent des notations CSN.1 ou ASN.1 (voir en annexe 7.1), l'ensemble des messages de signalisation mobile respectent de telles structures. Ces principes sont décrits principalement dans la norme 3GPP TS 24.007. Les protocoles de signalisation permettant la gestion des services circuits sont : MM (Mobility Management), CC (Call Control) et SS (Supplementary Services) ; ceux permettant la gestion des services de données en 2G et 3G sont : GMM (GPRS Mobility Management) et SM (Session Management) ; ceux permettant la gestion des services LTE sont : EMM (EPS Mobility Management) et ESM (EPS Session Management). S'y ajoute le protocole SMS (Short Message Service), ainsi que les protocoles de gestion de la localisation : LCS (LoCation Services), RRLP (Radio Ressource Location services Protocol) en 2G et 3G et LPP (LTE Positionning Protocol) en LTE, et enfin SUPL (Secure User-Plane Location) permettant de travailler sur une connexion de données.

Le schéma 3 illustre ces protocoles de signalisation tels qu'ils sont mis en œuvre entre les terminaux et les réseaux mobiles. Des informations supplémentaires sur ces protocoles sont données en annexe 7.1.

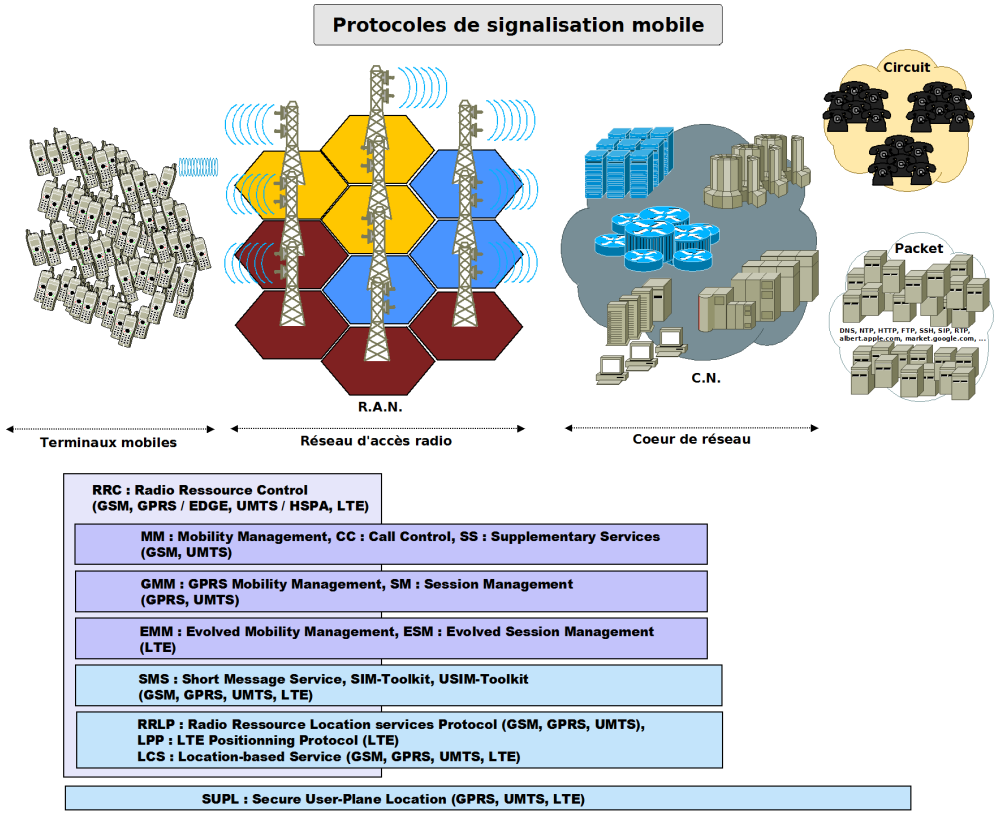

Figure 3. Description des principaux protocoles de signalisation de l'interface radio

2 Architecture matérielle et logicielle des modems

2.1 Principes mis en œuvre

Le modem est souvent un système autonome dans les mobiles récents. A l'époque des réseaux GSM/GPRS, et des premiers lancements 3G, ce système était le seul présent dans le terminal, et délivrait souvent une API Java ME pour l'exécution d'applications simples (gestion SMS, calendrier, réveil matin, *etc...*). De nombreux terminaux d'entrée de gamme continuent d'utiliser une telle architecture aujourd'hui. Qualcomm propose par exemple l'API Brew (ou Brew MP plus récemment) qui permet d'exécuter des applications spécifiques, développées en C, C++ ou Java, directement sur l'OS du modem [4].

D'un autre côté dans les mobiles évolués et les smartphones, le modem mobile est appelé et utilisé par un second système embarqué hébergeant l'OS applicatif (Android, Windows Mobile, iPhone OS, Blackberry OS, Symbian, Bada, Firefox OS, *etc...*), à travers une interface de communication. Cette interface de communication se divise en une interface de commande et une interface de transfert de données à haut-débit. Le 3GPP standardise dans la norme TS 27.007 l'interface de commande AT qui permet de contrôler un modem mobile. Cette interface est très majoritairement utilisée pour contrôler les modems mobiles sur clé USB ; par contre, certains fabricants proposent d'autres interfaces de contrôle pour leurs modems, par exemple Qualcomm propose un protocole propriétaire dénommé QMI pour ses modems LTE. L'interface de transfert de données (qui va permettre le transfert du flux audio de la voix et des paquets IP) est, quant à elle, relativement propriétaire et dépendante de l'architecture du téléphone : classiquement, il s'agit soit d'une interface sérielle à haut débit (par exemple de type USB), soit d'un partage d'une zone mémoire commune entre le système du modem et celui applicatif.

Le modem mobile s'appuie systématiquement sur un système d'exploitation simple, temps réel (RTOS), et avec peu, voire pas de mécanismes de cloisonnement ou de virtualisation de la mémoire : ainsi dans un modem, toutes les tâches s'exécutent dans le même espace d'adressage. Lorsqu'un hyperviseur contrôle le modem, des restrictions d'accès à certaines zones mémoires peuvent être imposées. Les multiples mécanismes logiques s'exécutent au sein de tâches distinctes du RTOS : la gestion des multiples protocoles de signalisation et de leurs états respectifs pour l'ensemble des technologies supportées, la gestion des liaisons radio (avec ou sans acquittement, avec ou sans ré-ordonnancement, *etc...*) mais aussi la prise en charge des interfaces de communication (de la carte à puce, des composants radio numériques et analogiques, des interfaces de contrôle et de transfert avec le système applicatif) et des autres composants matériels. A l'exception de l'introduction récente de l'architecture Hexagon par Qualcomm, la quasi-totalité des modems mobiles s'appuient sur une architecture et un cœur ARM. Selon les cas, le modem s'exécute sur un processeur ARM dédié, appelé alors baseband processor (BP), ou colocalisé avec l'OS applicatif au sein d'un SoC disposant de plusieurs cœurs ARM.

Par ailleurs, de nombreuses tâches spécifiques nécessitant des ressources de calculs importantes sont prises en charge par des composants spécifiques (DSP et/ou ASIC) : parmi ceux-ci, les mécanismes cryptographiques (par exemple le chiffrement du trafic utilisateur, pouvant attendre 150 Mb/s en LTE), et les mécanismes d'encodage / décodage nécessaires aux trans-

missions radio (par exemple pour l'émission d'une trame radio : calcul d'intégrité de la trame, fragmentation, scrambling, convolution, modulation numérique en bande de base...). De plus, les mécanismes d'émission et réception radio sur différentes bandes de fréquences nécessitent systématiquement des circuits analogiques supplémentaires.

D'un point de vue théorique, et en lien avec le chapitre 1, il est possible de voir un modem mobile comme présenté dans la figure 4. Pour chaque technologie radio (2G, 3G et LTE), il y a un frontend radio spécifique, correspondant à des bandes de fréquences porteuses distinctes, ainsi que des gestionnaires bas niveau pouvant faire appel à des routines matérielles spécifiques. Plus on remonte à des aspects logiques, avec la prises en charge des protocoles de signalisation, plus cela correspond à des traitements purement logiciels ; enfin, les deux principaux domaines de services (CS avec les appels voix, PS avec le transfert de données) sont exposés à travers des interfaces avec l'OS applicatif.

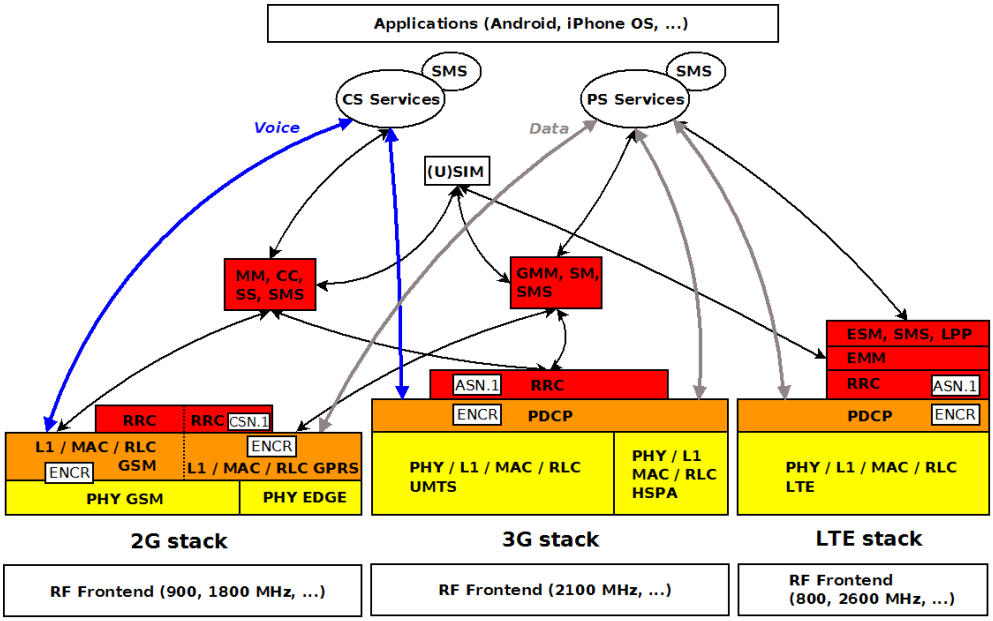

Figure 4. Représentation des fonctions standards au sein d'un modem mobile multi-mode

2.2 Fabricants et éditeurs

Le monde des modems mobiles évolue rapidement, avec le développement des multiples technologies de communications. On peut d'une part distinguer les fabricants de chipsets et de SoC (System on Chip) qui

réalisent des puces incluant des cœurs ARM mais aussi d'autres composants nécessaires aux terminaux et modems, et les éditeurs de piles logicielles radio-mobiles. Dans de nombreux cas, le fabricant du matériel est également éditeur du logiciel du modem mobile, mais il arrive de rencontrer parfois des situations surprenantes...

- Qualcomm est actuellement le principal fournisseur de chipsets et SoC 2G-3G-LTE (sans doute plus de 90% de parts de marché en LTE), mais aussi un des principaux fournisseurs de chipsets 2G et 3G depuis plusieurs années. Qualcomm propose aujourd'hui des « system on chip » multicoeurs incluant plusieurs composants : cœurs DSP, blocs d'accélération matérielle (chiffrement mobile, GPU, codecs audio / vidéo). Qualcomm a récemment remplacé l'emploi de cœurs ARM par sa propre architecture de calcul, Hexagon : le code des modems LTE s'exécute désormais sur cette architecture. En parallèle à ses puces, Qualcomm édite une suite logicielle pour exploiter chipsets et SoC : précédemment à l'introduction d'Hexagon, le système AMSS était associé à un hyperviseur L4 et une couche temps-réel historique (REX). QuRT (aussi appelé BLAST) a remplacé REX/AMSS dans les SoC LTE les plus récents.
- Intel (ex-Infineon) : fournisseur de modems discrets, légèrement en retard par rapport à Qualcomm sur le LTE, mais bien présent avec ses modems 2G/3G depuis de nombreuses années (il équipe notamment – hors LTE – les téléphones Galaxy S/S2/S3 et de nombreux iPhone) ; il s'appuie sur un RTOS commercial : ThreadX. Les principales solutions Intel actuels sont les chipsets XMM, incluant le composant modem X-Gold.
- ST-Ericsson était jusqu'à sa disparition un fabricant de puces intégrant un modem et un processeur applicatif, dénommé NovaThor. Le modèle U8500 a connu un succès certain : 25 millions de téléphones l'intègrent, en particulier. plusieurs téléphones de Sony ainsi que le Galaxy S3 Mini. La partie logicielle du modem de la puce U8500 est fourni par Renesas.
- Renesas Mobile : branche de Renesas dédiée à la fabrication de processeurs applicatifs et de modems LTE et multi-modes, partiellement rachetée en 2013 par Broadcom. Renesas Mobile a intégré le logiciel radio-mobile de Nokia dès 2009, avant d'acheter la branche R&D de Nokia qui en fait le développement. En conséquence, le logiciel du modem mobile de Renesas est commun à celui de très nombreux mobiles Nokia jusqu'en 2010 (Nokia ayant par la suite

basculé sur des SoC Qualcomm avec l'introduction des terminaux sous Windows Phone 7).
- Broadcom : fabricant de nombreux processeurs et chipsets, dont des chipsets mobiles 2G et 3G (comme le BCM21553 équipant le Samsung Galaxy Y). Suite à l'acquisition de Renesas, Broadcom développe une nouvelle ligne de puces LTE.
- Hisilicon : filiale de Huawei, fabricant de processeurs et chipsets mobiles, principalement intégrés dans des téléphones et clés USB Huawei ; les chipsets radio-mobiles sont dénommés Balong.
- Parmi les fabricants de modems multi-modes moins connus : NVIDIA (ex-Icera) qui équipe certains téléphones ZTE, Mediatek (très présent sur le marché chinois), Spreadtrum. Texas Instruments qui a apparemment quitté en 2008 le marché des modems.
- Il existe aussi d'autres fabricants de chipsets mobiles LTE qui éditent éventuellement eux-mêmes leur logiciel : Samsung fabrique par exemple un chipset LTE équipé de son propre logiciel dénommé kalmia ; Sequans est une société française spécialisée dans les modems OFDM (WiMAX, LTE) et fabrique son propre modem LTE ; Altair est une société israélienne qui fabrique également son propre modem LTE.

Depuis l'avènement de la 3G, puis les premiers lancements LTE, Qualcomm a pris une position dominante sur une grande partie des terminaux avec ses chipsets intégrés Snapdragon, en particulier sur les terminaux haut-de-gamme. Tous ses composants sont développés par et sous license Qualcomm. L'intégration du Wi-Fi et du Bluetooth, et partiellement du système de géolocalisation, a été effectuée grâce au rachat de l'entreprise Atheros. De ce fait, Qualcomm fournit une suite logicielle importante afin que les intégrateurs et fabricants de téléphones mobiles puissent exploiter au mieux leur matériel.

Afin d'avoir une idée plus précise sur le design matériel des téléphones mobiles, des sites Internet réalisent désassemblages et identifications des composants sur des téléphones connus. On constate l'omniprésence de Qualcomm sur le segment des ordiphones haut de gamme :

- Nokia N900 [19]
- HTC One [15]
- Sony Xperia Z [47]
- Google Nexus 5 [14]
- Apple iPhone 5S [5]

2.3 Etat de l'art de l'analyse des modems mobiles

Historiquement, les modems de téléphones portables ont été largement étudiés par la communauté du « désimlockage » dans un but lucratif. La compétition forte entre les différents groupes impliqués n'a pas contribué à diffuser publiquement les techniques et méthodes employées. La situation a évolué et d'autre chercheurs partagent maintenant leur savoir sur Internet.

On peut tout d'abord citer le projet « MADos » de g3gg0 et Wumpus [30] qui ont analysé de façon détaillée le modem du Nokia 3310, et publié le résultat de leur analyse sur Internet [31]. Ces travaux ont été réutilisés par des individus revendant des téléphones 3310 dits espions et reflashés avec une version modifiée du code de Nokia pour décrocher automatiquement, sans avertissement visuel ou sonore. De manière plus honorable, les travaux de g3gg0 ont servi à identifier et activer les fonctions de traces réseau du 3310, permettant alors la capture des trames GSM émises et reçues. Cela fut mis en pratique par Duncan Salerno dans le projet DCT3-GSMTAP [23].

En 2007, George Hotz (geohot) implémente en moins d'un mois une attaque matérielle sur le modem S-Gold de l'iPhone 2G, avec l'aide de gray, iProof, dinopio et lazyc0der. Il explique de manière très détaillée sa démarche sur son blog (actuellement fermé, mais l'historique peut être consulté [10]). Grâce à l'obtention de la datasheet du modem et d'un iPhone sacrifiable, George Hotz a dessoudé le modem pour exposer les contacts et tracer ainsi l'emplacement du JTAG vers un header de 64 pins. De cette manière, il a été possible d'accéder au contenu de la mémoire morte de démarrage (« bootrom »). Une faille permettant de contourner la vérification de la signature du code en mémoire flash a alors été identifiée.

A contrario, le désimlockage des versions ultérieures de l'iPhone a reposé sur l'exploitation de vulnérabilités dans le code du modem, typiquement dans le traitement des commandes AT. En 2011, Luis Miras présente à ekoparty l'instrumentation du modem de l'iPhone 3G via l'exploit « yellowsn0w » [41] (dépassement de mémoire tampon dans le traitement de la commande AT+STKPROG). En 2012 est publié « iOS Hacker's Handbook » dont le chapitre 11 a été écrit par Ralf-Philipp Weinmann avec l'aide de MuscleNerd et planetbeing [17]. Ils détaillent le code d'exploitation « ultrasn0w » qui enlève le blocage SIM sur certaines versions du modem de l'iPhone 4 et antérieur. En particulier sont exposés comment l'exploit, toujours par une commande AT vulnérable, contourne par une chaîne de POR (programmation orientée retour) la non exécution des zones de mémoire inscriptibles. De même en 2012, lors de la conférence Hack In The Box Amsterdam, MuscleNerd et planetbeing

décrivent comment les contre-mesures d'Apple ont rendu progressivement le blocage SIM de l'iPhone plus difficile à contourner [12].

Parallèlement aux travaux de la « Chronic Dev (team) » sur l'iPhone [49], Ralf-Philipp Weinmann détaille lors des conférences hack.lu 2010 [39] et USENIX 2012 [40], ses travaux relatifs à l'instrumentation d'OpenBTS pour identifier des problèmes de sécurité dans différentes modèles de téléphones. Il montre ainsi comment un jeton d'authentification mobile AUTN trop long (normalement limité à 16 octets) donne lieu à un dépassement de mémoire tampon dans la pile d'un modem Qualcomm ; il présente aussi comment un identifiant temporaire TMSI de 128 octets au lieu de 4 provoque le crash d'un modem Infineon présent sur les iPhone 4 et inférieurs. En 2011, Guillaume Delugré présente et publie un debugger destiné aux modems Qualcomm dans les clés USB Option ICON 225 [8]. Un site également intéressant est celui de Tim Newsham qui présente une analyse des modems Qualcomm, qui s'exécutent sur un hyperviseur dérivé de L4 [16].

Plus récemment, Willem Hengeveld a publié, via le github de la société allemande GSMK, un module de processeur IDA pour l'architecture Hexagon [20]. Ralf-Philipp Weinmann jette en parallèle les bases de l'analyse sécurité de cette nouvelle architecture [1]. On peut également citer deux analyseurs de logs publiés en 2013, l'un pour les modems LTE de Samsung développé par Ramtin Amin [2], l'autre pour les modems XGold qui équipent les mobiles Samsung développé par Tobias Engel [3].

Il faut mentionner aussi le forum xda-developers et notamment le contributeur *E :V :A*, qui anime plusieurs discussions sur l'analyse de modems Qualcomm et Intel. Enfin, un des principaux efforts dans le monde de l'open-source est le projet OsmocomBB [35]. Harald Welte, Sylvain Munaut, ainsi que quelques autres développeurs, publient ce projet, qui est une implémentation open-source d'un logiciel baseband GSM destiné aux téléphones GSM Motorola C1XY. Ces téléphones, reposant sur la plate-forme Calypso de Texas Instruments, furent choisis de par l'absence de chaîne de confiance lors de la phase de démarrage et en raison de la disponibilité sur Internet des datasheets relatives aux différents éléments constitutifs du système.

2.4 Techniques d'analyse

Différents moyens d'analyse existent afin d'étudier le fonctionnement des modems mobiles. Chacun a ses avantages et ses inconvénients, mais dans tous les cas, il ne faut pas escompter disposer de quelconques codes source pour les modems récents (sauf à signer un NDA avec le fabricant).

Analyse statique de l'image binaire du modem

Cette méthode nécessite le moins de moyens matériels. Il suffit de pouvoir récupérer l'image binaire qui est chargée puis exécutée par le processeur du modem. Pour cela, on peut récupérer des images de mises à jour complètes des téléphones sur les sites des constructeurs de mobiles (Sony, LG, Samsung, Huawei, ZTE, HTC, *etc...* ou sur des sites tiers tels que freeflashfile.com ou dc-files.com). Il est alors possible de décompresser l'image pour récupérer spécifiquement l'exécutable du modem, qui peut être sous une forme qui n'est pas directement celle chargeable par le processeur radio-mobile. Il est aussi généralement possible de récupérer les images des modems directement sur les systèmes de fichiers des téléphones mobiles.

En règle générale, une image exécutable de modem mobile, selon le nombre de technologies prises en charge (sa complexité) et la manière dont elle a été compilée (sa « verbosité » entre autre), pèse de quelques méga à quelques dizaines de mégaoctets. Pour certains éditeurs, l'image est en fait divisée en plusieurs fichiers qui vont être chargés à des positions différentes en mémoire, avant que l'exécution nominale ne soit lancée.

Une fois l'image exécutable correctement extraite, il faut identifier les structures utiles au RTOS pour réaliser l'exécution : il s'agit la plupart du temps de la liste des tâches ou fonctions principales avec pour chacune, leur point d'entrée dans l'image exécutable (l'adresse initiale assignée au pointeur d'exécution lorsqu'on entre dans la tâche), modulo l'adresse de chargement initiale de l'image par le processeur (ou les adresses, lorsque l'image est divisée en plusieurs fichiers). Souvent, une chaîne de caractère descriptive est associée au point d'entrée de la tâche ou de la fonction, permettant d'identifier facilement son rôle au sein du modem.

A partir de ces points d'entrée identifiés, il est possible de désassembler assez proprement le code exécutable du modem via un outil commercial (IDA Pro) ou gratuit (radare ou objdump ARM / Hexagon).

Notons que Ralf-Philipp Weinmann a publié un script [42] permettant de récupérer sur une image de modem Intel-Infineon X-Gold utilisant ThreadX, la liste des tâches à exécuter avec leurs principales caractéristiques (nom, point d'entrée, adresse de la stack, *etc...*). Le projet open-source docl4amss [16] propose également une bonne documentation et des scripts d'analyse pour AMSS de Qualcomm.

Si l'on exclut le ou les composants en charge du traitement du signal (DSP), le jeu d'instructions du processeur exécutant les fonctions de traitement de plus haut niveau, comme par exemple les gestionnaires de sig-

nalisation, est généralement connu. ARM dans ses différentes déclinaisons est très majoritairement employé, mis à part :

- Qualcomm avec l'introduction d'Hexagon dans ses puces LTE. Cette architecture est heureusement documentée de manière détaillée [9], et des chaînes de compilation basées sur gcc et llvm ainsi que des patchs pour le noyau Linux sont également fournis [11] ;
- NVIDIA (ex-Icera) qui a apparemment développé une architecture non documentée, mais qui paraît, au vu des brevets déposés [6] proche dans l'esprit de celle de Qualcomm : VLIW (paquets d'instructions en blocs de 64 bits) avec des extensions pour le calcul parallèle (SIMD) et couplé à des blocs d'accélération reconfigurables dits « DXP$^{\text{TM}}$ » (Deep Execution Processor) [38]. De même que pour Hexagon, l'ensemble des traitements, de la radio jusqu'aux protocoles de haut niveau, devrait avoir lieu sur cette même architecture.

Analyse statique d'empreintes mémoire du modem en cours d'exécution

La prise d'empreinte du système en cours d'exécution, ou dump mémoire, peut se faire par différents moyens, selon les possibilités offertes par le modem, l'OS applicatif et le matériel. Certains modems offrent, via leur interface de contrôle, la possibilité le lire et / ou écrire leur mémoire. Ceci peut permettre de récupérer une copie complète de la mémoire du modem alors même que celui-ci s'exécute : cette méthode peut être utilisée directement via l'interface USB pour les modems sur clé USB, ou via l'OS applicatif (par exemple, certains mobiles sous Android le permettent). Elle est employée par Guillaume Delugré pour injecter le débogueur qcombbdbg dans la mémoire d'une clef 3G à base de modem MSM6280 de Qualcomm.

Avec certains téléphones, il est également possible de récupérer un « crash-dump », une copie mémoire lors de l'arrêt inopiné de l'exécution du modem. La suite logicielle propriétaire (et sous licence) QPST de Qualcomm permet ainsi la récupération d'une image mémoire ; les modems d'Intel disposent également de cette fonctionnalité, déclenchable par un menu de diagnostic spécifique (sur les téléphones Samsung avec l'APK SamsungServiceMode, composer le *#197328640#*). Cependant, le crash-dump n'est généralement pas une copie directe de la mémoire, mais contenue dans un format propriétaire qu'il est alors difficile d'analyser sans connaissances préalable du format.

Une alternative plus générique est d'employer l'interface JTAG du modem lorsqu'elle est disponible. C'est généralement le cas sur de nombreux

modèles de clefs 3G à base de puce Qualcomm ; la principale difficulté étant l'identification des pins correspondants sur le PCB. Il arrive que leur emplacement soit documenté sur Internet ou par un logiciel spécialisé (comme par exemple la RIFF BOX). Alternativement, le projet JTAGulator de Joe Grand [26] donne la possibilité de tester un grand nombre de combinaisons pour retrouver le pinout complet.

Une fois l'interface JTAG identifiée, les mécanismes JTAG offrent la lecture et l'écriture en mémoire : il est donc possible d'effectuer une copie de celle-ci via cette interface. L'outil open-source OpenOCD permet alors de récupérer le contenu de la mémoire ; de poser des points d'arrêts et de tracer l'exécution du code.

Cette méthode ne fonctionne plus avec l'introduction de puces LTE et d'Hexagon par Qualcomm, et donc l'abandon de l'architecture ARM (pour laquelle les commandes JTAG variaient très peu entre constructeurs). En l'absence de documentation, le seul fournisseur tiers de produits de déboguage JTAG pour Hexagon reste Lauterbach [7].

Analyse dynamique de l'exécution du modem

L'analyse dynamique de l'exécution du modem, par JTAG, est *a priori* assez délicate ; en particulier l'arrêt prolongé du processeur peut donner lieu au déclenchement inopiné de « watchdog » et au redémarrage du processeur. Le JTAG reste cependant un outil majeur en raison de son indépendance par rapport à un constructeur spécifique.

Certains systèmes d'exploitation de modems prévoient des fonctionnalités d'analyse et de reporting via des outils propriétaires. Qualcomm fournit par exemple une suite d'outils (en particulier QPST et QXDM) destinés principalement aux fabricants de terminaux et aux opérateurs mobiles : ces outils permettent de récupérer de nombreuses informations à propos des tâches en cours d'exécution dans le système, mais aussi de nombreux paramètres et statistiques concernant la configuration radio et la connectivité du modem.

Analyse du comportement du modem sur les réseaux mobiles

Une dernière méthode d'analyse du modem est de regarder son fonctionnement détaillé à partir de traces faites côté réseau mobile : ceci est présenté dans le chapitre 3.

Il existe également des outils qui collectent les logs générés par les modems et en permettent l'analyse par exemple avec wireshark (tels que ceux publié par Tobias Engel ou Ramtin Amin). Globalement, peu de

moyens documentés existent pour réaliser l'analyse du comportement des modems 3G et LTE sur les réseaux mobiles.

2.5 Exemples d'analyses

Ce chapitre a pour but de décrire quelques analyses succinctes qui ont été faites sur des images binaires de modems commerciaux, afin d'obtenir une meilleure compréhension des logiques qui y sont implantées.

Un des buts principaux de l'analyse est de pouvoir identifier précisément des morceaux de code machine responsable de fonctions critiques d'un point de vue de la sécurité du modem et des communications prises en charge. Par exemple, les fonctions réalisant l'authentification entre le terminal et le réseau mobile (voir la norme TS 24,008, section 4.3.2, « Authentication procedure », et section 4.7.7 « Authentication and ciphering procedure »), ou la sélection des algorithmes de chiffrement et de contrôle d'intégrité, lors de l'établissement de canaux de communications dédiés (procédures appelées « Security mode control », et réalisées au niveau du protocole RRC pour le GSM, la 3G et le LTE, mais également au niveau du protocole EMM pour le LTE). On peut également essayer d'identifier les morceaux de code responsables des copies en mémoire des messages de signalisation lorsque ceux-ci sont remontés par les couches basses (MAC, RLC) vers les gestionnaires de signalisation de plus haut niveau (RRC, MM / GMM, EMM, gestionnaire de carte SIM).

Modem Intel-Infineon de la série XMM

Dans ce premier exemple, nous analysons un téléphone équipé d'un modem Intel XMM6260 qui fonctionne en 2G et 3G. Nous avons pu prélever l'image exécutable du modem lorsque celle-ci est chargée par Android : le fichier en question s'appelle *HWT6260.fls.fls*. Les logs Android mentionnent l'offset à partir duquel le fichier est lu ainsi que sa longueur, et l'adresse en mémoire à laquelle il est copié. Il peut ainsi être facilement chargé dans IDA pour analyse.

Il est possible également de récupérer des ROM contenant les images complètes du mobile, pour différentes régions du monde. Un script Perl spécifiquement développé pour désarchiver ces images (split_updata.pl) circule sur xda-developpers et permet d'extraire des partitions contenant *a priori* les images modems des téléphones. Il doit être possible de convertir cette image fournie par le script Perl vers le fichier .fls qui est finalement chargé par le modem, mais cela n'a pas été investigué.

Une fois l'image exécutable extraite du fichier .fls, on peut vérifier la présence des chaînes 1.

```
 4106f a    XMM6260_V2_SMPH_FLASHLESS_USB-HSIC_REV_2.7 2013-May-27
     20:34:30
33ce37 aCopyright (c) 1996-2009 Express Logic Inc. * ThreadX ARM11/
    RVDS Version G5.3.1.5.2 SN: Infineon_SVC_SH_Version *
```

Listing 1. Chaînes intéressantes pour l'identification du logiciel radio-mobile XMM

Grâce à l'analyse déjà réalisée par Ralf-Philipp Weinmann sur ce type de modem Intel, on peut suivre la démarche proposée dans son script de chargement IDA flsloader, particulièrement dans le script *make_tasktable.py* qui permet d'identifier la liste des tâches principales lancées par le système ThreadX. On peut ainsi récupérer une liste de 60 tâches, avec leur nom, leur point d'entrée, et l'adresse initiale de leur stack : se référer à l'annexe 7.2 pour la liste complète.

La connaissance des points d'entrée de chaque tâche permet de désassembler une bonne partie du code machine du modem. Attention à mettre à 0 le bit de poids faible des points d'entrée (celui-ci indique le passage du processeur ARM en mode Thumb). Le nom des tâches, le nombre très important de blocs logiques constituant certaines d'entre elles ainsi que les chaînes référencées, permettent de faire un rapprochement possible entre le code machine et les fonctions logiques suivantes :

- *umacul*, *umacdl*, *umacc* : UTRAN MAC (UpLink / DownLink) ;
- *urlcul*, *urlcdl*, *urlcc* : UTRAN RLC (UpLink / DownLink) ;
- *urrcbp*, *urrcdc*, *urrcm* : UTRAN RRC, gestionnaire de ressources radio 3G ;
- *urabmupdcp* : UTRAN Radio Access Bearer MUltiplexer PDCP, gestionnaire de la couche PDCP 3G ;
- *mac*, *rlc*, *llc* : GSM / GPRS MAC, RLC et LLC, gestionnaires des liaisons radio 2G ;
- *rrc* et *grr* : gestionnaires de ressources radio 2G (protocoles RRC) ;
- *mmc*, *mme*, *mmr* et *gmm* : gestionnaires de la mobilité (protocoles MM et GMM) ;
- *sim* : gestionnaire de la carte SIM ;
- *mng*, *mni*, *mnm*, *mnp* et *mns* : gestionnaires de contrôle des services du modem (appelés probablement via le gestionnaire de commandes AT).

De nombreuses chaînes sont cependant positionnées dans des tableaux sans être directement appelées par le code machine ; elles sont pointées par

des tableaux de références et / ou d'indexation. En conséquence, beaucoup de travail reste à faire si l'on souhaite pouvoir identifier précisément certaines fonctions du modem réalisant des traitements critiques pour la sécurité des communications mobiles.

Modem Broadcom de la série BRCM21552

Dans cet exemple, nous récupérons une archive contenant une ROM pour un téléphone Samsung Galaxy Y S5360, qui fonctionne en 2G et 3G. On peut trouver le manuel de service Samsung sur Internet, ainsi que les sources du kernel Android Gingerbread, qui inclut le module de prise en charge du modem Broadcom [48].

Les composants sur le PCB peuvent être facilement identifiés sur la figure 5.

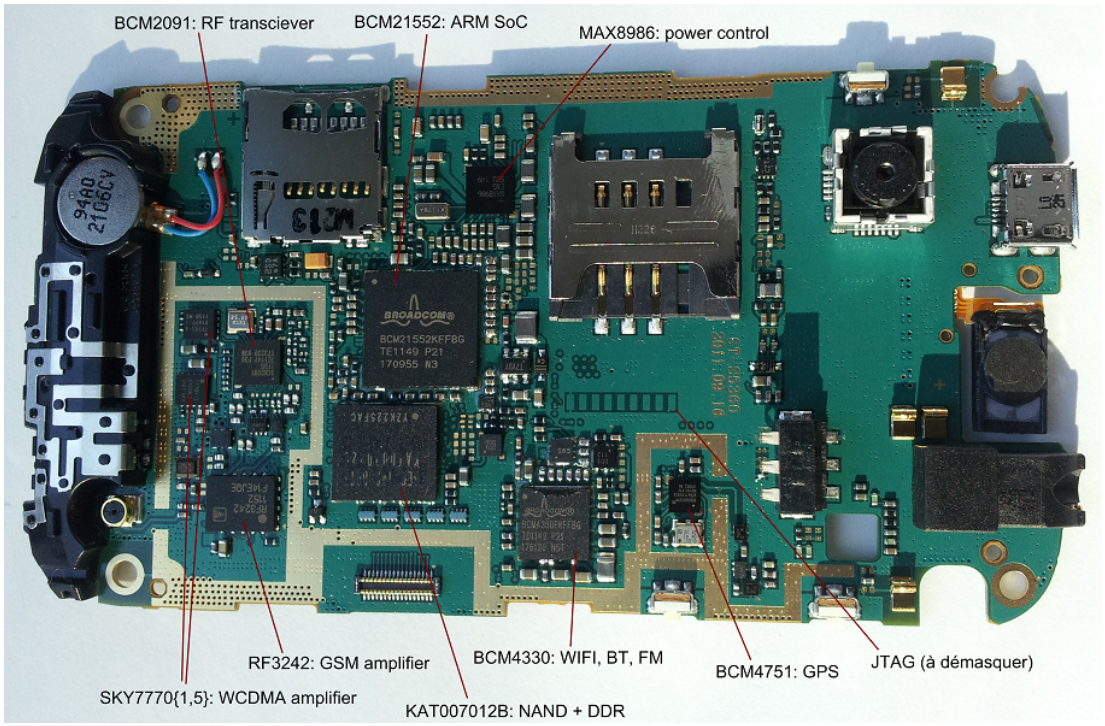

Figure 5. PCB d'un Samsung Galaxy Y (S5360)

On constate qu'il n'y a qu'une seule puce pour supporter l'OS applicatif Android ainsi que le modem. D'après le site Broadcom, il s'agit d'un puce intégrée HSPA, qui contient un processeur graphique ainsi qu'un processeur ARM11 supportant Windows Mobile et Android : « 3G phone-on-a-chip ». En conséquence, on peut penser qu'il s'agit d'un SoC disposant de deux

unités de calcul (une pour le modem, une pour l'OS applicatif). Le port JTAG, une fois démasqué, permet d'accéder au processeur du modem.

Il est possible de récupérer des images du modem sur le site xda-developers. Pour cet exemple, nous allons regarder le fichier *modem-XXLH2.img*. On peut tout d'abord vérifier la présence des chaînes dans le listing 2.

```
2618e0 @(#)Broadcom GL SUPL ver. 2 106146, 2011/Aug/30, 02:24:44
2b82c8 @(#)Broadcom GL RRLP ver. 1 106146, 2011/Aug/30, 02:24:44
2ef9c0 @(#)Broadcom GL RRC ver. 1 106146, 2011/Aug/30, 02:24:44
750fe4 BCM21553_Modem_SI1220.2_V2.4
8540f0 Copyright (c) 1996-2010 Express Logic Inc. * ThreadX ARM11/
    RVDS Version G5.4.5.3 SN: 3058-099-1301 *
8570c4 @(#)Broadcom GLL ver. 2.18.8 106146, 2011/Aug/30, 02:24:44,
    //depot/client/core/qa_line
```

Listing 2. Chaînes intéressantes pour l'identification du logiciel radio-mobile BCM

Le copyright d'Express Logic sur ThreadX donne un indice, en rapport avec l'exemple précédent. En parcourant les chaînes contenues dans l'image, on retrouve une liste de chaînes qui ne nous est pas inconnue : df2:0, mma:0, mac:0, umacc:0, urlcc:0, mncc:0, urrcdc:0, *etc...*

Il existe une similarité avec le modem Intel-Infineon présenté au chapitre précédent. Une recherche sur Internet permet de découvrir la société Comneon, fondée en 1991 et filiale de Infineon Technologies AG, ayant été rachetée par Intel en 2010 (au même titre que Infineon). La spécialité de Comneon est le développement de logiciels pour les communications mobiles (GSM, GPRS, EDGE, WCDMA/FDD). Il semble que la pile logicielle radio-mobile de Comneon ait été utilisée non seulement par Infineon et Broadcom par le passé, mais aussi par NXP, NEC, NTT-Docomo... Aujourd'hui, Broadcom indique utiliser une pile logicielle entièrement développée en interne.

Il est possible de procéder d'une manière similaire à l'exemple précédent pour retrouver les structures de description des tâches exécutées par le modem, qui référencent les chaînes identifiées précédemment. On retrouve en effet une liste de pointeurs vers ces noms, espacés de 28 octets, et modulo l'offset de chargement de l'image exécutable. La structure de 28 octets semble se composer comme indiqué dans le listing 3.

```
struct bcm_task {
  uint32 task_name ;
  uint32 task_entry ;
  uint32 null ;
  uint32 flag1 ;
```

```
  uint32 flag2 ;
  uint32 task_id ;
  uint32 unknown_addr ; } ;
```

Listing 3. Structure d'un descripteur de tâche pour le démarrage du modem Broadcom

On récupère ainsi le nom et l'identifiant des tâches, ainsi que leur point d'entrée : se référer à l'annexe 7.2. pour la liste complète. Ce faisant, on constate une parfaite correspondance avec le fichier *proc_ id.h* disponible sur le github contenant les sources Android mises à disposition par Broadcom. Il devient alors aisé de lancer une analyse plus fine avec IDA en indiquant les points d'entrée de chaque tâche pour désassemblage, après avoir chargé le fichier à l'offset adéquat.

Après chargement dans IDA, on constate malheureusement qu'une partie du code n'est pas désassemblé ; de plus, certaines tâches qui sont dans le modem Intel volumineuses en code, semblent ici être relativement vides. Par exemple, le graphe d'appel de la tâche grr (attention : plusieurs milliers de nœuds) présente de fortes similarités entre ce modem Broadcom et celui d'Intel, alors que celui de la tâche gmm est ici quasiment vide et celui d'Intel présente un gros volume de logique.

A première vue pour ce Samsung Galaxy Y, il est possible que Broadcom n'ait conservé que les gestionnaires de canaux physiques et logiques 2G, ainsi que SIM, et ait abandonné la prise en charge des canaux 3G ainsi que des gestionnaires de mobilité fournie par la pile Comneon pour la réimplanter avec ses propres routines. Malheureusement à ce jour, nous n'avons pas identifié clairement le code machine produit par Broadcom et responsable de ces fonctions. Une piste prometteuse consisterait à analyser le tableau de 46 chaînes au tout début du fichier *modem-XXLH2.img*, et leurs possibles référencements par des structures au sein de l'image exécutable.

Modem ST-Ericsson Novathor

ST-Ericsson a créé une joint-venture le temps de développer un SoC intégré supportant un OS applicatif ainsi qu'un modem mobile. Ce système, dénommé Novathor, a été utilisé dans de nombreux mobiles 2G / 3G de Samsung et Sony-Ericsson entre 2012 et 2013. Malheureusement, la joint-venture a été dissoute mi-2013 ; le SoC Novathor est désormais un produit distribué par STMicroelectronics, qui ne semble plus développé.

Nous avons récupéré une image du modem sur un mobile Samsung Galaxy S3 mini : il s'agit d'un fichier de près de 12 MO, qui contient certaines chaînes indiquées dans le listing 4.

```
    5aa #The following line must not be modified, it is required by
        Nokia Bridge
 12cb1e (c) Renesas Mobile
 3b5942 SS Server (c) Renesas Mobile Corporation.
 3d5996 MCE server, (c) Nokia
 4258ca (c) Nokia Perm Server
 4d5946 (c) Nokia NVD Server
 6a38da Net server #279, (c) Nokia
 6a3d62 GSS Server (c) Nokia
 6d1682 (c) Renesas Mobile
 7d1e4f armlink --dll --override_visibility --map --no_remove [...]
 b2e3a0 armcc --c99 --thumb --bss_threshold=0 --debug -c --asm [...]
 b2ede3 armcc --c99 --thumb --bss_threshold=0 --debug -c --asm [...]
```

Listing 4. Chaînes intéressantes pour l'identification du logiciel Novathor

Il est surprenant de noter autant de mentions à Nokia et Renesas dans l'image du modem ST-Ericsson. Comme évoqué au chapitre 2.2, Renesas a racheté la branche de Nokia responsable du développement de la partie modem en 2010. Il semble ici que ST-Ericsson ait intégré du logiciel venant de Renesas (et donc développé initialement par Nokia) pour la réalisation de son modem 2G / 3G.

En ouvrant le fichier dans un éditeur hexadecimal, on constate dès l'entête la présence d'une table descriptive des composants du logiciel. Chaque structure de 32 octets contient une chaîne, un offset et une longueur. La lecture de cette table donne la liste des fichiers indiqués dans le listing 5.

```
modem    env: (length: 961)
default  env: (length: 57)
product  env: (length: 245)
l1       out: (length: 1291384)
RFHAL    do : (length: 2221624)
l23      out: (length: 3614976)
IPC      do : (length: 78800)
SECUR    do : (length: 124252)
INTC     do : (length: 33608)
I2CHAL   do : (length: 67184)
I2CHAL_Tdo : (length: 16268)
SIMHAL   do : (length: 259556)
SIMHAL_Tdo : (length: 93160)
IPCLPBK do : (length: 30968)
VLTFRQ   do : (length: 29992)
CPULD_L1do : (length: 16908)
CPULD_L2do : (length: 16436)
EVDET    do : (length: 25360)
TSE      do : (length: 39680)
TRACTRL do : (length: 14860)
```

```
PMCONF  do : (length: 12936)
SCLKD   do : (length: 24352)
SCLKMGT do : (length: 32008)
hwinit  out: (length: 6980)
INTEG   do : (length: 10360)
OEM     do : (length: 2091636)
PIF     do : (length: 1402256)
```

Listing 5. Liste des fichiers contenu dans l'image du modem NovaThor

Ceci permet de segmenter le fichier du modem en sous-fichiers :

- les fichiers « env » sont des fichiers de texte contenant des informations sur l'initialisation du système et l'environnement matériel (on y trouve par exemple le segment de mémoire qui doit être partagé avec le processeur applicatif exécutant Android).
- Les fichiers « out » et « do » contiennent tous un en-tête ELF ainsi que du code machine. On peut supposer que les « out » sont les principaux exécutables, et qu'ils s'appuient sur des fonctions présentes dans les « do » (Dynamic Object ?) qui jouent alors le rôle de bibliothèques.

L'étude des en-têtes ELF nous apprend plusieurs éléments :

- L'ABI semble propriétaire de ST-Ericsson (valeur EI_OSABI à 64), cela pourrait aussi correspondre à l'utilisation de la toolchain ARM (armcc, armlink, *etc...*).
- L'exécutable principal contenu dans le fichier est décrit par le 1er « program header », qui est chargé à une adresse virtuelle égale à l'adresse physique.
- Certains drapeaux des « program header », ainsi que certains types des « section header » correspondent à des attributs qui semblent propriétaires de ST-Ericsson.

Malgré ces quelques spécificités, il est possible de charger chaque fichier indépendamment dans IDA pro. Ceci permet de désassembler une bonne partie du code exécutable contenu dans chaque fichier, les symboles exportés par les bibliothèques sont alors correctement identifiés. Cependant, les éventuelles références directes aux objets dans les bibliothèques (.do) et entre exécutables (.out) ne sont pas résolues. Pour cela, il faudra réaliser un script IDA qui, à partir de chaque en-tête ELF de chaque fichier, va charger et désassembler les sections appropriées dans un espace d'adressage unique au sein d'une image RAM complète.

Modem Qualcomm Snapdragon sur architecture Hexagon

Pour ce dernier exemple, nous avons examiné une image de téléphone Google Nexus 5. Ce téléphone est produit par LG, c'est un des premiers terminaux à utiliser le SoC Qualcomm Snapdragon 800. Ce SoC intègre une unité de calcul Hexagon, qui prend en charge l'intégralité de l'exécution du logiciel radio-mobile 2G / 3G / LTE. Ce modem s'appuie *a priori* sur un micro-noyau spécifiquement développé par Qualcomm, dénommé QuRT.

Il est possible de récupérer des images du modem sur Internet. Dans notre cas, nous allons travailler sur l'image récupérée dans le fichier *LGD820_KRT16M_RADIO.zip* de 19 MO. Un fichier modem.img de 64 MO se trouve dans le zip. Il s'agit en fait d'un système de fichiers, qui peut être monté sous Linux ou simplement ouvert avec 7z. On en extrait deux fichiers préfixés « mba » dont un nommé mba.mdt et 21 fichiers préfixés « modem » dont un nommé modem.mdt. Ces deux fichiers *.mdt* sont en fait des en-têtes ELF.

On constate que l'en-tête ELF dans le fichier modem.mdt décrit 28 « program header ». En les regardant individuellement, on constate que les longueurs (champ *p_filesz*) correspondent exactement à l'ensemble des fichiers modem.b00 à modem.b26 extraits du système de fichiers modem.img. L'ABI réferencée dans l'en-tête ELF principal semble standard. Le listing 6 montre une partie cet en-tête ELF.

```
### [Elf32_Ehdr] ###
 ### [e_ident] ###
  <[EI_MAG] : '\x7fELF'>
  <[EI_CLASS] : '1 : ELFCLASS32: 32-bit objects'>
  <[EI_DATA] : '1 : ELFDATA2LSB: Little endian'>
  <[EI_VERSION] : '1 : EV_CURRENT: Current'>
  <[EI_OSABI] : '0 : ELFOSABI_SYSV: UNIX System V ABI'>
  <[EI_OSABIVERSION] : 0>
  <[EI_PAD] : 0x000000000000>
  <[EI_NIDENT] : 0>
 <[e_type] : '2 : ET_EXEC: Executable file'>
 <[e_machine] : '164 : Qualcomm Hexagon'>
 <[e_version] : '1 : EV_CURRENT: Current'>
 <[e_entry] : 138412032>
 <[e_phoff] : 52>
 <[e_shoff] : 0>
 <[e_flags] : 0x04000000>
 <[e_ehsize] : 52>
 <[e_phentsize] : 32>
 <[e_phnum] : 28>
 <[e_shentsize] : 40>
 <[e_shnum] : 0>
 <[e_shstrndx] : 0>
[...]
    ### [Elf32_Phdr] ###
     <[p_type] : '1 : PT_LOAD: Loadable segment'>
```

```
     <[p_offset] : 16338944>
     <[p_vaddr] : 154857472>
     <[p_paddr] : 154857472>
     <[p_filesz] : 13680656>
     <[p_memsz] : 13680656>
     <[p_flags] : '5 : PF_X+R'>
     <[p_align] : 4096>
    [...]
```

Listing 6. Début de l'en-tête ELF du fichier modem.mdt, et program header correspondant au fichier modem.b17

Dans un premier temps, il faut reconstruire un fichier ELF unique, en concaténant les fichiers modem.* tout en respectant les valeurs *p_offset* des program header (en ajoutant du padding dans le fichier final si nécessaire). Ceci permet d'obtenir un fichier ELF complet de 41584 ko. Avant de charger l'image dans IDA, il convient de récupérer le module de processeur IDA pour l'architecture Hexagon mis à disposition sur github. Le chargement du fichier ELF dans IDA entraîne un désassemblage très complet de l'exécutable du modem, simplement à partir du point d'entrée donné par l'en-tête ELF : voir la figure 6.

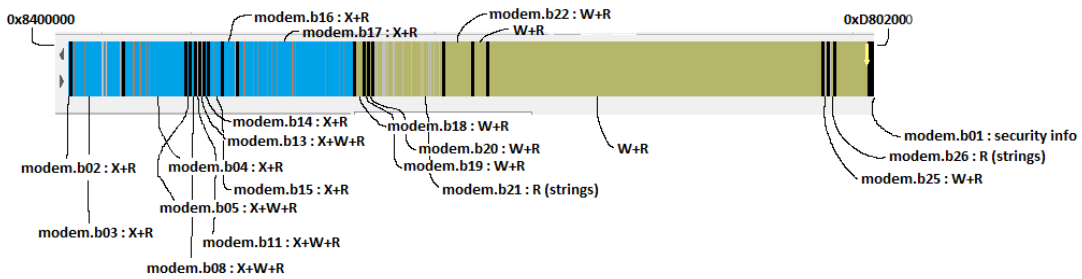

Figure 6. Répartition de la mémoire suite au chargement dans IDA du fichier ELF reconstitué

Il est possible de faire de même en concaténant les fichiers mba.mdt et mba.b00 afin d'obtenir un désassemblé de la routine d'initialisation du micro-noyau QuRT de Qualcomm. Il est fort probable que les 2 exécutables cohabitent dans le même espace d'adressage (mba est chargé à l'offset 0xD800000, modem est chargé à 0x8400000) et que modem puisse rappeler certaines fonctions d'initialisation de mba.

Par la suite dans l'image du modem, on trouve un tableau de chaînes qui s'apparente à la liste des tâches lancées classiquement par les modems Qualcomm (on peut prendre comme référence les informations données sur le googlecode du projet docl4amss [16]). On dénombre ainsi 265 chaînes,

toutes référencées par une liste de structures de 28 octets chacune, selon le format indiqué dans le listing 7.

```
struct qurt_amss_task {
  uint32 task_name ;
  uint32 flag1 ;
  uint32 task_entry ;
  uint32 flag2 ;
  uint32 flag3 ;
  uint32 flag4 ;
  uint32 stack_addr ; } ;
```

Listing 7. Structure des descripteurs de tâches lancées lors de l'initialisation du système QuRT

Ceci permet de mettre un nom sur des blocs de code machine, et facilite la compréhension et l'analyse du modem. Attention à la visualisation des blocs logiques dans IDA, les instructions « jump » et « call » désassemblées avec le module de processeur Hexagon ne donnent pas toujours lieu à la poursuite du graphe d'appel dans la visualisation graphique ; seule la visualisation « proximity browser » permet de visualiser les graphes d'appel de certaine tâche dans leur ensemble.

Une recherche sur Internet permet de tomber sur un site interessant [21], qui est un clone d'un ancien répertoire du site codeaurora.org (utilisé par Qualcomm pour publier ses logiciels en source ouverte) : le fichier *tmc.c* est le lanceur de tâches d'un modem Qualcomm. Il permet, grâce à un grand nombre de définitions explicites, de corréler les tâches avec leurs fonctions. On peut ainsi faire, par exemple, les rapprochements suivants :

- *bam_drv* : « Bus Access Manager driver », il semble qu'il s'agisse du système de communication interne au SoC Snapdragon ;
- *dog* : RTOS watchdog ;
- *nv**, *fs*, *fs_** : gestionnaires de mémoires non-volatiles (y compris du système de fichiers EFS propriétaire de Qualcomm) ;
- *smd**, *smem*, *smp2p*, *smsm* : « Shared Memory » (daemon, peer-to-peer, state machine), il s'agit probablement du système de communication avec l'OS applicatif, via partage de mémoire ;
- *diag*, *diag_f3** : gestionnaire du protocole DIAG pour le debug du modem ;
- *qmi_** : « Qualcomm MSM Interface », gestionnaire de contrôle du modem depuis une interface externe (Qualcomm n'utilise pas forcément les commandes AT pour le contrôle de ses modems, se réferer à la bilbiothèque en source ouverte libqmi) ;
- *gps_fs*, *gpsfft*, *nf*, *gnss*, *gnss_** : gestionnaires de géo-positionnement (GPS, GLONASS) ;

- *locotdoa** : gestionnaire de géo-positionnement par rapport aux antennes du réseau mobile (OTDOA : « Observed Time Difference of Arrival ») ;
- *lm*, *loc_middleware* : gestionnaires de géolocalisation ;
- *gsm_** : gestionnaires des liaisons et ressources radio GSM ;
- *cb* : gestionnaire des messages de diffusion par les cellules GSM ;
- *hdrsrch*, *hdrrx*, *hdrtx*, *hdrmc*, *hdrdec* : « High Data Rate », gestionnaires des liaisons radio CDMA2000 (norme de radio-communication américaine) ;
- *mc*, *rxtx*, *srch*, *rx*, *tx* : « Qualcomm TMC 1X », gestionnaires des liaisons radio EV-DO (norme de radio-communication américaine) ;
- *tds_** : gestionnaires des liaisons radio TD-SCDMA (variante de la 3G européenne, principalement utilisée en Asie) ;
- *wcdma_**, *dswcsd_**, *rrc* : gestionnaires des liaisons et ressources radio WCDMA ;
- *rabm* : « Radio Access Bearer Management », gestionnaire global des canaux radio ;
- *mm*, *reg*, *mn_cnm*, *cm*, *ps*, *sm* : gestionnaires de signalisation de haut niveau (mobilité, enregistrement et connectivité aux réseaux, appels, données, sessions) ;
- *wms** : « Qualcomm Wireless Messaging Service », gestionnaires des SMS ;
- *pdcommtcp*, *pdcommwms* : gestionnaires de services de liaisons de données au sein du modem ;
- *auth*, *uim** : « User Identity Module », gestionnaire de module d'authentification embarqué (certaines normes de radio-communications américaines ne nécessitent pas de carte SIM pour l'authentification, cette fonction est alors embarquée directement dans le modem) ;
- *gsdi*, *gstk* : « Qualcomm Generic SIM Driver Interface » et « Qualcomm Generic SIM Toolkit », gestionnaires de cartes SIM ;
- *ims** : gestionnaire IMS (protocoles SIP et RTP, à confirmer).

Cependant, un grand absent demeure : le gestionnaire de liaison et de ressources radio LTE. Les tâches recensées semblent correspondre à l'existant provenant des modems Qualcomm AMSS sur architecture ARM. Il est possible que le gestionnaire LTE, dont l'introduction correspond à celle de l'architecture Hexagon, soit pris en charge de manière distincte par rapport à la partie logicielle historique des modems AMSS.

On peut malgré tout identifier via les chaînes préfixées *lte_* de larges portions de code exécutable ; on note également la référence « lte_rrc_osys_asn1util » qui semble indiquer que Qualcomm utilise

du logiciel fourni par Objective SYSTEMS pour traiter la syntaxe ASN.1 issue de la norme RRC LTE.

2.6 Conclusion sur l'analyse de code

Même si certains éditeurs et constructeurs peuvent réutiliser certaines parties de logiciels communes, la diversité du code d'un constructeur à l'autre reste importante. Par ailleurs, l'analyse de ces systèmes embarqués, même s'ils ne présentent pas d'obfuscation ni de pièges particuliers, reste délicate et austère. De plus, l'introduction d'architectures nouvelles (comme Hexagon introduite par Qualcomm) complexifie la compréhension du fonctionnement de base du système. De ce fait, l'analyse du comportement des modems vus de l'extérieur (depuis l'interface radio-mobile, et donc vu du réseau) reste une technique de choix afin d'évaluer leur sécurité.

3 Réseaux mobiles de test

3.1 Précautions vis-à-vis de l'émission radio

Le spectre radio utilisé pour la téléphonie mobile est soumis à licence. Ces licences sont achetées par les opérateurs mobiles au prix fort auprès de l'Etat, et ils en ont l'usage exclusif pour un nombre d'années donné. En contrepartie, ils doivent assurer un certain niveau de service (couverture de la population nationale et du territoire, gestion des appels d'urgence -112-, interfaces d'interception légale mise à disposition des services judiciaires du pays). En conséquence, il n'est pas autorisé d'émettre sur les bandes GSM, UMTS, LTE pour toute entité qui n'est pas un opérateur mobile.

L'ensemble de nos tests ont été réalisés dans des conditions très strictes, notamment au regard de l'article 226 du code pénal. Les tests menés sur des terminaux doivent être réalisés en cage de Faraday. Il est par ailleurs utile d'isoler le réseau de test logiquement des réseaux nationaux, en employant des codes réseau distincts de tout opérateur existant en France et à l'international (par exemple le code PLMN classique pour les tests mobiles 001/01). Une mesure de précaution supplémentaire est la configuration de listes blanches d'IMSI au sein des réseaux de test, accompagnée de codes de rejet adaptés pour éjecter proprement les mobiles non identifiés au préalable.

Le but des procédures de tests mises en œuvre ainsi que des explications des vulnérabilités données au chapitre 4 est d'améliorer la sécurité des communications mobiles dans le respect de la loi.

3.2 Etat de l'art des matériels et logiciels publics

La principale difficulté pour réaliser des réseaux mobiles de tests est le support de l'interface radio de bas niveau (essentiellement le support de la modulation et du codage des canaux physiques). De ce fait, deux tendances existent :

- La première consiste à réutiliser des équipements radio existants, tels que des stations de base commerciales, 2G, 3G ou LTE. Même si leur coût semble important, certains matériels obsolètes (BTS GSM des années 90 par exemple) peuvent être acquis d'occasion pour de faibles sommes. Par ailleurs, l'industrie développe de plus en plus des systèmes compacts et à faible capacité (picocells et femtocells), qui ont de fait un coût, une complexité et un encombrement moindre que les équipements radio-mobiles classiques. Il est alors possible d'utiliser certains de ces équipements radio commerciaux avec des logiciels spécifiquement développés pour la réalisation de tests mobiles.
- La deuxième consiste à utiliser des systèmes radio programmables (également appelé SDR pour « Software Defined Radio »). Il s'agit d'un domaine assez nouveau, où des petites entreprises proposent des équipements radio suffisamment performants pour des coûts assez faibles. Les plus connus sont Ettus Research avec leurs USRP [43], Fairwaves avec l'UmTRX [44], Nuand avec le bladeRF [46], MyriadRF qui propose des cartes utilisant des composants radio de Lime Microsystems [45]. Dans ce cas, il est nécessaire que le logiciel de test mobile prenne en charge tous les traitements radio.

Depuis plusieurs années, des logiciels disponibles en source ouverte, permettent de réaliser un réseau GSM de test :

- OpenBTS [33] : il s'agit d'une application UNIX qui s'appuie sur un transmetteur radio USRP. Le logiciel réalise tous les traitements radio d'une interface GSM, et intègre des gestionnaires de signalisation 2G qui fonctionnent de concert avec un IPBX SIP (asterisk ou yate) pour supporter les appels voix et les SMS. Récemment, le support du GPRS a été intégré à l'application. Deux nouveaux projets de BTS logicielle commencent à voir le jour : osmoBTS et yateBTS.
- OpenBSC [32] : il s'agit d'une application UNIX qui réalise de nombreuses fonctions de cœur de réseaux mobiles 2G, et s'interface avec des BTS commerciales (Siemens, IP.Access) ou open-source (osmoBTS).

Il s'agit des deux projets qui ont été le plus utilisés pour faire des tests de sécurité sur des modems mobiles GSM :

- Harald Welte présente un mécanisme de fuzzing via OpenBSC dès 2009 [13] ;
- Ralf-Philipp Weinmann détaille lors des conférences hack.lu 2010 [39] et USENIX 2012 [40] ses travaux relatifs à l'instrumentation d'OpenBTS. Il met à jour des vulnérabilités de type dépassement de mémoire tampon dans certains modems, et développe même un code d'exploitation pour déclencher des commandes AT arbitraires dans certains mobiles.
- Nico Golde et Collin Mulliner présentent en 2011 à CanSecWest [36] un certain nombre de bogues et de vulnérabilités dans les gestionnaires SMS de nombreux terminaux mobiles, principalement d'entrée de gamme ; ce travail est réalisé grâce à une instrumentation d'OpenBSC.
- Sébastien Dudek et Guillaume Delugré présentent en 2012 à hack.lu [37] une architecture de fuzzing basée sur OpenBTS et d'instrumentation d'Android afin de superviser les modems fuzzés.

On peut également cité les travaux de Sylvain Munaut qui a porté le transmetteur d'OpenBTS pour fonctionner avec des terminaux Motorola d'OsmocomBB, ainsi que les nombreuses présentations de Karsten Nohl présentant des attaques contre les algorithmes de chiffrement GSM (A5/1 et A5/2) et GPRS (GEA1 et GEA2).

On constate cependant que toutes ces recherches et résultats ne concernent que la téléphonie mobile de 2nde génération (GSM et plus légèrement GPRS). A ce jour, peu d'outils permettent de travailler sur les protocoles radio-mobiles 3G et LTE. On peut tout de même citer certains efforts de développement :

- L'unique projet open-source concernant la 3G et le WCDMA est un projet EURECOM dénommé "Wireless3G4Free". Le développement de celui-ci a été abandonné depuis quelques années au profit du projet "openairinterface" orienté vers le LTE [24]. On peut noter que l'EURECOM a développé des réseaux 3G et LTE complets, qui s'appuient sur des cartes radio spécifiquement développées pour ceux-ci.
- Des projets d'analyse de sécurité de femtocells ont conduit à l'éventuelle réutilisation de ces dernières pour potentiellement tester des mobiles 3G : la femtocell SFR d'ancienne génération, ainsi que la femtocell de l'opérateur américain AT&T ont été ainsi détournées.
- Quelques projets en source ouverte sur le LTE sont en train de voir le jour ; parmi les plus aboutis (ceux-ci restent malgré tout à un

stade très expérimental), on peut citer OpenLTE [34], GNU Radio LTE Receiver [25] et libLTE [27].
- Le projet logiciel le plus abouti reste celui de Fabrice Bellard : LTEENB [29], qui est aujourd'hui commercialisé par Amarisoft. Le logiciel permet, associé à un USRP, d'émuler une antenne LTE ainsi qu'un cœur de réseau LTE de manière fiable.

3.3 Réseau GSM de test

Depuis 2008, la disponibilité d'OpenBTS pour l'émulation d'un réseau GSM de base permet le passage d'appels voix et l'envoi / réception de SMS. Depuis 2013, le support du GPRS a été ajouté. Etant donnée la maturité du projet, son caractère purement logiciel associé au support des cartes de radio logicielle USRP, et la documentation des travaux de sécurité déjà réalisés sur ce système, il se prête bien à la réalisation de tests de sécurité sur les terminaux mobiles 2G à moindre coût.

Le fait d'être une antenne purement logicielle permet de modifier certains aspects du fonctionnement GSM de manière assez radicale :
- il est ainsi possible d'influer sur l'ordonnancement et le contenu des canaux logiques, tels que les canaux de broadcast RRC GSM ;
- il est également possible d'influer sur les canaux de communication et de signalisation dédiés, et de travailler sur la signalisation RRC, MM, CC et SMS.

Aujourd'hui cependant, il est admis que la technologie GSM, de par sa conception, n'est que peu sécurisée. De plus, les nombreux travaux déjà publiés sur le sujet ont mis en lumière certains problèmes existants dans les implémentations des modems mobiles. En conséquence, nos travaux de recherches ont été plus orientés vers les tests sur les technologies 3G et LTE.

3.4 Réseau 3G de test

Il n'existe malheureusement pas à l'heure actuelle de logiciel ou système en source ouverte fonctionnel et disponible pour travailler sur la 3G. Les techniques de modulation par étalement fréquentiel utilisées dans le WCDMA rendent un développement en radio logicielle pure assez difficile. Nous avons ainsi pris la décision de travailler avec un opérateur français et de réutiliser du matériel radio 3G existant. De cette manière, toute la gestion de l'interface radio (modulation, codage canal, liaison et signalisation RRC) est prise en charge par ce matériel commercial. Il

est alors possible de s'interfacer avec celui-ci et de prendre en charge les protocoles et sous-systèmes correspondant à un cœur de réseau mobile. Ceci permet d'exposer et de tester ces sous-systèmes au sein des modems mobiles.

Un cœur de réseau spécifique et minimal a été développé, afin de pouvoir prendre en charge un type spécifique de matériel radio 3G, spécialement configuré pour l'occasion. De cette manière, il nous est possible de travailler sur les protocoles suivants avec les terminaux mobiles :

- MM, avec les procédures d'attachement et de détachement du réseau, d'authentification et de renouvellement d'identité temporaire dans le domaine CS ;
- SMS, avec l'envoi et la réception de tout type de messages courts ;
- GMM, avec les procédures d'attachement et de détachement du réseau, d'authentification et de renouvellement d'identité temporaire dans le domaine PS ; SM, avec les procédures d'attribution de contexte PDP (pour la connectivité IP, via un APN) et d'établissement de tunnels GTP. L'interface IP du mobile est entièrement exposée et accessible à travers le protocole GTP qui encapsule les paquets de données qui sont transmis à travers l'interface radio ;
- Il est possible de plus de sélectionner l'algorithme de chiffrement (UEA0 / UEA1) mis en place via le protocole RRC lors de l'établissement de communications radio entre un mobile et l'équipement radio.

Les mécanismes d'établissement d'appels (CC), de mobilité entre plusieurs cellules (MM et GMM), de services supplémentaires (SS) et de géolocalisation n'ont jusqu'ici pas été testés. Par ailleurs, les tests menés avec cette solution n'ont pas toujours été satisfaisants : le logiciel du matériel radio utilisé réalise un contrôle et un éventuel reformatage des messages de signalisation ; ceci nous a menés à tester certaines adaptations et modifications sur ce matériel commercial, afin de pouvoir améliorer quelques peu nos capacités de tests.

L'ensemble des fonctionnalités et des scénarios de tests a été développé en Python, à l'aide de la bibliothèque libmich [28], qui facilite la construction et le décodage de nombreux messages de signalisation mobile.

3.5 Réseau LTE de test

La réalisation d'un réseau LTE de test a été rendu possible grâce à l'acquisition du logiciel LTE100 d'Amarisoft. Ce dernier est constitué de deux binaires Linux : un réalisant les fonctions de radio logicielle pour

l'antenne LTE (eNodeB), un autre assurant les fonctions d'un cœur de réseau LTE (MME, SGW-PGW et HSS). Ces deux binaires communiquent entre eux via le protocole de signalisation standard entre les antennes et le cœur de réseau LTE : S1-AP.

Ceci a rendu possible, de même que pour les tests sur les terminaux 3G, le développement d'un cœur de réseau LTE spécifique. Les fonctionnalités suivantes ont été implantées afin de pouvoir être efficacement testées :

- un gestionnaire simple S1AP a été développé afin de communiquer avec le binaire réalisant la fonction d'antenne LTE eNodeB. Ceci permet de se passer du binaire Amarisoft qui réalise les fonctions de cœur de réseau ;
- un gestionnaire NAS, avec les mécanismes de sécurité associés. En effet, la signalisation NAS entre un terminal mobile et un MME bénéficie d'une sécurité de bout en bout avec chiffrement (algorithmes EEA0/1/2/3) et contrôle d'intégrité cryptographique (EIA0/1/2/3) ;
- un gestionnaire EMM qui gère l'attachement et le détachement au réseau, l'authentification et le renouvellement d'identité temporaire ;
- un gestionnaire ESM simpliste qui gère l'attribution du contexte PDP par défaut (pour la connectivité IP, via un APN) et d'établissement de tunnels GTP. De même que pour le réseau 3G de test, l'interface IP du mobile est entièrement exposée ;
- il est possible par ailleurs de configurer finement le fonctionnement de l'eNodeB via ses fichiers de configuration. Ceci inclut les algorithmes à utiliser pour le chiffrement des communications radio (EEA0/1/2) et le contrôle d'intégrité cryptographique (EIA0/1/2) appliqué à la signalisation RRC.

Cette réimplémentation des fonctionnalités du cœur de réseau LTE (MME, gestionnaires d'authentification HSS et de tunnels GTP) permet ainsi de réaliser différents scénarios de test affinés sur les modems et terminaux LTE.

L'ensemble des fonctionnalités et des scénarios de tests a également été développé en Python, toujours à l'aide de la bibliothèque libmich [28], mais aussi du gestionnaire CryptoMobile [22] qui prend en charge les différents algorithmes de chiffrement mobile 3G et LTE.

4 Premiers résultats des tests réseaux

4.1 Périmètres des tests

Nos tests ont surtout à ce stade été menés sur les modems 3G (une dizaine de clés USB 3G, et une vingtaine de terminaux 2G / 3G ou 2G

/ 3G / LTE). De nombreux terminaux mobiles de type smartphone ont été prêtés personnellement par de valeureux personnels de l'ANSSI. Nous tenons à préciser qu'aucun mobile n'a été détérioré lors de cette étude. Des scénarios de tests à la limite, voire en dehors des spécifications 3GPP, ont été développés via des adaptations des scripts Python réalisant les fonctions décrites dans les chapitres 3.4 et 3.5.

Certains résultats qui sont présentés ci-dessous ont été anonymisés, afin de ne pas exposer des éditeurs ou constructeurs spécifiques. Le but de cette publication n'est pas ailleurs pas de montrer du doigt un quelconque fabricant, mais plutôt de sensibiliser l'ensemble des intervenants sur les systèmes mobiles et les problèmes de sécurité qui y sont associés.

4.2 Absence d'indicateur de sécurité de l'interface radio

Le premier fait simple constaté lors de nos tests 3G et LTE est l'absence systématique d'information, au niveau de l'interface graphique des mobiles, lorsque la connexion radio-mobile est en clair. Il est ainsi quasiment impossible pour un utilisateur de mobile de savoir si ses communications transmises par voie radio sont correctement chiffrées, ou tout simplement en clair, et donc écoutables par quiconque est muni d'un récepteur et d'une antenne adéquate.

La norme 3GPP TS 22.101, qui décrit les services de base assurés par les technologies mobiles, définit la nécessité de présenter un indicateur graphique à l'utilisateur d'un téléphone lorsque la communication radio n'est pas chiffrée (fonctionnalité dite de « ciphering indicator »). On constate malheureusement qu'aucun smartphone d'aujourd'hui (du moins, aucun de ceux testés) n'implante une telle fonctionnalité, pourtant essentielle à une perception correcte de la sécurité par les abonnés aux services mobiles de nouvelles générations.

En rapport avec ce sujet, le site Internet osmocom publie une liste de terminaux qui fournissent cet indicateur [18], on remarque en définitive que seuls quelques vieux mobiles GSM supportent cette fonctionnalité.

4.3 Etat général du support cryptographique dans les modems

Petit historique des algorithmes cryptographiques dans les réseaux mobiles

Quelle que soit la technologie mobile (2G, 3G, LTE), les mécanismes cryptographiques y sont omniprésents (cela ne veut pas dire qu'ils sont toujours efficaces) :

- le GSM permet le chiffrement des canaux de voix (TCH) et de signalisation (SDCCH et SACCH/FACCH) entre le terminal et la BTS avec les algorithmes suivants : A5/1, A5/2, A5/3 ou A5/4 ;
- le GPRS permet le chiffrement des liaisons logiques (LLC) entre le terminal et le SGSN avec les algorithmes suivants : GEA1, GEA2, GEA3 ou GEA4 ;
- la 3G permet le chiffrement des paquets logiques (PDCP) entre le terminal et le RNC avec les algorithmes suivants : UEA1 ou UEA2. Un contrôle d'intégrité cryptographique est de plus appliqué à la signalisation RRC (qui encapsule elle-même la signalisation échangée entre le modem et le cœur de réseau) avec les algorithmes suivants : UIA1 ou UIA2 ;
- le LTE permet le chiffrement des paquets logiques (PDCP) entre le terminal et l'eNodeB avec les algorithmes suivants : EEA1, EEA2 ou EEA3. La signalisation RRC est de plus contrôlée en intégrité avec EIA1, EIA2 ou EIA3. Enfin la signalisation NAS est également chiffrée et contrôlée en intégrité de bout en bout entre terminal et MME, grâce aux mêmes algorithmes.

Les algorithmes A5/1, A5/2, GEA1 et GEA2 ont été développés lors de l'introduction du GSM puis du GPRS. Leurs spécifications ne sont pas publiques, il s'agit dans l'absolu d'algorithmes de chiffrement à flux utilisant des clés de 64 bits.

Lors du développement de la 3G, l'algorithme de chiffrement par bloc Kasumi a été choisi afin d'être utilisé pour UEA1 (en mode compteur) et pour UIA1 (en mode MAC). UEA1 et UIA1 utilisent des clés de 128 bits. Cet algorithme a également été adapté au GSM et au GPRS, afin de pouvoir remplacer à terme les algorithmes originaux. Ainsi ont été développés A5/3 et GEA3 (utilisant des clés de 64 bits) et A5/4 et GEA4 (utilisant des clés de 128 bits). Puis à partir de 2007, l'algorithme SNOW 3G a été introduit pour la 3G, comme algorithme de substitution, au cas où des attaques contre Kasumi viendraient à apparaître. Ceci a donné lieu à l'introduction de UEA2 et UIA2, utilisant des clés de 128 bits.

Pour le LTE, SNOW 3G a été réutilisé sous les dénominations EEA1 et EIA1, utilisant des clés de 128 bits. Dès la première version de la norme LTE, le support d'AES a également été introduit selon les dénominations EEA2 et EIA2, utilisant des clés de 128 bits. Enfin, l'algorithme ZUC a été introduit en 2011 selon les dénominations EEA3 et EIA3, utilisant également des clés de 128 bits.

Exemple d'annonce de support cryptographique fallacieux

Lorsque les modems mobiles accèdent à un réseau mobile, ceux-ci annoncent les algorithmes cryptographiques qu'ils supportent. Cela permet au réseau, en fonction de sa propre configuration, de choisir l'algorithme le plus adapté pour protéger les communications avec le mobile.

Lors de nos tests, nous avons pu constater qu'un modem USB 2G / 3G annonçait des capacités cryptographiques pour le moins extravagantes concernant la sécurité GSM : voir le listing 8.

```
### [MSRACap] ###
 <[AccessTechnoType] : '0 : GSM P'>
 ### [MSRAAccessCap] ###
  <[Length] : 73>
  ### [MSRAContent] ###
   <[RFPowerCap] : 0b100>
   ### [MSRAA5bits] ###
    <[A51] : 0b1>
    <[A52] : 0b0>
    <[A53] : 0b1>
    <[A54] : 0b1>
    <[A55] : 0b1>
    <[A56] : 0b1>
    <[A57] : 0b1>
   <[ESInd] : 0b1>
   <[PS] : 0b1>
   <[VGCS] : 0b0>
   <[VBS] : 0b0>
   ### [MultislotCap] ###
[...]
```

Listing 8. Extrait du « MSRadioAccessCapability » d'un modem mobile, qui annonce le support d'algorithmes cryptographiques qui n'existent pas

Ceci est probablement le fait du constructeur du modem USB et non du constructeur du chipset radio-mobile. Le fabricant du modem a dû modifier ce paramètre explicitement. Ce type d'annonce extravagante peut poser problème lorsque le modem se connecte à un réseau qui utilise A5/3 ou mieux, A5/4, et que le modem dit le supporter mais échoue lors de son utilisation. Les abonnés qui disposent d'un tel modem risquent alors d'être mécontents de leur opérateur, qui malheureusement n'y est pour rien !

Retrait du support d'A5/3 dans des terminaux récents

Nos tests ont également permis de constater l'absence de support de l'algorithme A5/3 dans un grand nombre de smartphones, y compris ceux proposés récemment par un des principaux fabricants ; ceci alors que ces terminaux supportent GEA3 et UEA1 / UIA1 (Kasumi) correctement,

et alors que quasiment tous ses concurrents supportent A5/3 de manière systématique.

Dans ce cas, il semble que le fabricant désactive spécifiquement la prise en charge de cet algorithme A5/3 dans les chipsets radio-mobiles qu'il intègre dans ses terminaux. Ceci est très dommageable sachant que l'algorithme GSM qui demeure (A5/1) est très faillible, d'autant plus avec les opérateurs GSM qui commencent à déployer l'algorithme A5/3 sur leurs parcs d'antennes GSM.

Absence systématique du support de UEA2 / UIA2 dans les modems 2G / 3G / LTE

Nous avons également eu une légère déception en constatant l'absence de support de UEA2 / UIA2 pour la partie 3G de tous les mobiles 2G / 3G / LTE testés, alors que l'algorithme SNOW 3G est systématiquement implanté et supporté pour la partie LTE, en tant que EEA1 / EIA1. Les conventions d'appels entre UEA2 et EEA1, UIA2 et EIA1, étant très similaires, on aurait pu penser que l'intégration de SNOW 3G dans les modems avec support LTE bénéficie également à la partie 3G du modem. Ce n'est malheureusement pas le cas.

4.4 Corruption mémoire pré-authentification sur un réseau de données

Une vulnérabilité de type écrasement mémoire pré-authentification a été découverte sur un modem 2G / 3G. Ce modem équipe de nombreux téléphones chez de multiples constructeurs. Cette vulnérabilité est déclenchée en envoyant une requête d'authentification 3G pour le domaine PS contenant un paramètre AUTN de longueur supérieure à 16 octets. Lors de la copie de ce paramètre entre le gestionnaire de mobilité GMM et le gestionnaire de carte USIM, il semble qu'une zone mémoire de 16 octets soit réservée (c'est la longueur normale donnée dans la spécification 3GPP), mais que la copie de la donnée AUTN se fasse ensuite itérativement selon le préfixe de longueur de ce paramètre. En conséquence, une corruption d'une structure mémoire du RTOS du modem à lieu. Le tableau 9 présente la requête d'authentification qui déclenche la corruption mémoire.

```
### [AUTHENTICATION_CIPHERING_REQUEST] ###
 <Skip Indicator [SI] : 0b0000>
 <Protocol Discriminator [PD] : '8 : GPRS mobility management
     messages'>
 <[Type] : '18 : GPRS - Authentication and ciphering request'>
```

```
<IMEISV requested [IMEISV] : 1>
<Ciphering algorithm [CiphAlg] : '0 : ciphering not used'>
<A&C reference number [ACRef] : 0x0>
<Force to standby [ForceStdby] : '0 : Force to standby not
    indicated'>
### [RAND] ###
 <[T] : 33>
 <[V] : 0x2397ec4a6eae6d130ca1fe9d3fb15a3c>
### [CKSN] ###
 <[T] : 8>
 <[V] : 2>
### [AUTN] ###
 <[T] : 40>
 <[L] : 166>
 <[V] : 0x36208a7438910000c2f995a318c1b7b9414141414141
     [...]414141414141>
```

Listing 9. Requête d'authentification 3G dans le domaine PS déclenchant la corruption mémoire

Dans le cas du mobile étudié lors de nos tests, il en résulte un vidage de la mémoire du modem sur la mémoire partagée avec le système d'exploitation applicatif, et l'arrêt de la connexion mobile pendant quelques dizaines de secondes à quelques minutes (le temps de production du vidage mémoire et de redémarrage du modem).

Le fabricant du modem a répondu très rapidement après avoir été sollicité, et des versions patchées du code du modem ont finalement été distribuées pour les terminaux reconnus vulnérables au bout de quelques mois. L'étude de la différence entre le modem patché et celui vulnérable a permis d'identifier précisément la zone mémoire écrasée, qui se situe au niveau des variables globales initialisées du RTOS. L'écrasement de la zone mémoire de 16 octets réservée au paramètre AUTN semble corrompre d'autres variables globales du système, ce qui entraîne son redémarrage.

4.5 Contournement du contrôle d'intégrité cryptographique sur un réseau LTE

Dans les réseaux 3G et LTE, un contrôle d'intégrité cryptographique est systématiquement appliqué sur les messages de signalisation échangés entre le terminal et le réseau : ceci permet de prolonger le caractère authentifié du canal de signalisation, et garantit qu'il n'y ait pas de détournement de la communication une fois l'authentification mutuelle effectuée.

Or il a été reconnu dès 2012, au sein des entreprises et opérateurs télécoms, que les modems LTE d'un grand constructeur avaient une permissivité trop grande concernant ce contrôle d'intégrité sur les connexions

LTE : les terminaux équipés de ces modems permissifs acceptaient l'établissement de sessions de données sans activer proprement ce contrôle d'intégrité (mode dit « EIA0 »). Il s'agit d'un mode hors norme, mis en place apparemment au lancement des premiers réseaux LTE pour faciliter la compatibilité entre modems et réseaux, et plus généralement en accélérer le déploiement. Pour rappel, les premiers déploiements LTE ont débuté dès la fin 2009 en Suède (avec Telia-Sonera), et courant 2010 aux Etats-Unis (avec Verizon). Ce mode EIA0 a malheureusement été conservé jusqu'en 2012, année durant laquelle le fabricant de modem a modifié sa base de code afin d'interdire l'usage d'EIA0.

Ce mode introduit malheureusement une vulnérabilité qui permet à un attaquant de facilement réaliser une interception active LTE sur ces modems permissifs : il est ainsi possible de faire une attaque par fausse station de base LTE, fluide et difficilement détectable du point de vue de la personne piégée. Aujourd'hui, tous les modems LTE de ce fabricant interdisent l'usage d'EIA0 ; il peut cependant rester des modems USB et terminaux mobiles LTE, acquis avant la fin 2012, pour lesquels cette vulnérabilité demeure (soit que le fabricant du terminal n'ait pas intégré et distribué une image non vulnérable pour son modem, soit que l'opérateur ou l'abonné n'ait pas lui-même réalisé une mise à jour du terminal alors que celle-ci était disponible).

Nous avons pu tester cette attaque en laboratoire, et confirmer que les modems LTE récents de ce fabricant, sortis après 2012, ne contiennent plus cette permissivité destructrice pour la sécurité des connexions LTE.

4.6 Déni de service ciblé sur un réseau LTE

Lors de la définition de scénarios de tests pour les modems LTE, nous avons identifié un cas spécifique de déni de service à l'encontre de modems multimode répandus. Cela permet à une fausse station de base LTE, sans établissement de contexte de sécurité préalable, d'engendrer un détachement durable d'un terminal ciblé, sans que celui-ci ne tente de se rattacher à son réseau légitime. Le modem du terminal doit alors être redémarré électriquement afin de pouvoir se rattacher à son réseau légitime et rétablir les communications.

Ce problème, qui n'induit pas de risques d'interception ni d'écoute sur les communications mobiles, vient d'être corrigé chez le fabricant.

4.7 Comportements suspicieux et petits travers des OS applicatifs et des applications

Le fait de disposer d'un réseau mobile de test, qui supporte les connexions de données à haut débit, ainsi que l'échange de SMS, permet de visualiser facilement les connexions IP ou les envois de SMS initiés par les mobiles, sans action (ni même intention) de l'abonné, mais également les services et démons réseaux potentiellement en écoute au niveau de l'OS applicatif. Les constats fait dans ce chapitre ne nécessitent pas forcément d'avoir un réseau mobile de test, et peuvent dans la plupart des cas être évalués via un réseau Wi-Fi.

Android, le cloud et la publicité

On peut constater que de très nombreux mobiles Android (on ne citera aucun constructeur, mais la plupart intègre par défaut des applications « peu respectueuses »), alors même qu'ils sortent fraîchement de leur emballage, accèdent dès leur première connexion IP non seulement au cloud google, mais également souvent au cloud du fabricant de mobile, et parfois aussi à plusieurs régies publicitaires.

Ces connexions ont lieu simplement par principe, et sans aucun consentement de l'acheteur du mobile qui vient tout juste d'allumer son téléphone pour la première fois.

Services réseaux en écoute

Même si cela reste rare, il a été constaté sur quelques mobiles la présence de services réseaux en écoute :

- sur un Samsung Galaxy S3 mini, en Android 4.1.1, les ports TCP/2001, 2002, 2003 et 2005 ;
- sur un iPhone 4S, le port TCP/62078 (port *a priori* top-listé par nmap) ;
- sur un terminal Blackberry Bold 9700, un service en écoute sur le port UDP/19780 essaie d'établir une connexion vers une IP en dure (résolue dans le domaine de rim.net) vers le port UDP/19781.

Les services en écoute sur ces ports, et leur sécurité intrinsèque, n'ont pas été évalués. Dans le cas du Samsung, cela pourrait correspondre au service Kies ; pour l'iPhone 4S, au système de synchronisation avec iTunes ; pour le Blackberry, au système d'enregistrement sur les serveurs RIM.

Apple iPhone, ou l'épine dans le pied des opérateurs (et des utilisateurs)

De nombreuses restrictions ont pu être observées sur des terminaux iPhone 5, en 3G comme en LTE. Tout d'abord, même pour les iPhones 5 qui ne sont pas simlockés, il existe une restriction imposée par Apple, qui fait qu'on ne peut pas connecter un iPhone sur un réseau mobile qui n'est pas recensé et autorisé par Apple.

Il semble que les iPhone obtiennent une liste blanche de réseaux mobiles autorisés depuis l'un des serveurs Apple (dans certains cas, il semble que ce soit *albert.apple.com*). Dans les premières versions d'iPhone 5 (iOS 6), l'iPhone se connectait éventuellement au réseau mobile non autorisé par Apple, avant de télécharger la liste puis de se déconnecter. Dans les versions les plus récentes (iOS 7), l'iPhone oblige l'utilisateur à passer par une connexion Wi-Fi pour s'activer et accéder aux serveurs d'activation. Une fois ceci effectué, si vous tentez d'attacher l'iPhone à un réseau mobile non référencé par Apple, vous obtenez le type de message tel que présenté dans l'image 7.

Figure 7. iPhone 5 mécontent

Il a également été constaté sur un iPhone 5S (avec iOS 7), un envoi systématique de SMS lors de l'activation avec une nouvelle carte USIM. Tout d'abord l'iPhone tente l'envoi d'un SMS vers un premier numéro

correspondant à son opérateur en France et contenant la chaîne « STATE ». Si cet envoi de SMS échoue à plusieurs reprises, l'iPhone s'énerve et tente alors d'envoyer un SMS vers un numéro situé au Royaume-Uni ! Le contenu du SMS est présenté dans le listing 10.

```
### [TP_Destination_Address] ###
 <length of digits [Length] : 12>
 <Extension [Ext] : 1>
 <Type of number [Type] : '1 : international number'>
 <Numbering plan identification [NumPlan] : '1 : ISDN / telephony
     numbering plan (E.164 / E.163)'>
 <[Num] : 447786205094>
### [TP_PID] ###
 <[Format] : '0 : telematic indication'>
 <[Telematic] : '0 : no telematic interworking, but SME-to-SME
     protocol'>
 <[Protocol] : '0 : Short Message Type 0'>
### [TP_DCS] ###
 <[Group] : '0 : General Data Coding'>
 <[GroupExt] : '0 : uncompressed - no class meaning'>
 <[Charset] : '0 : GSM 7 bit default alphabet'>
 <[Class] : 0>
<[TP_VP_rel] : 30 minutes>
<User Data Length (in character) [TP_UDL] : 89>
<[TP_UD] : REG-REQ?v=3;t=77[...]D5;r=5a[...]>
```

Listing 10. SMS d'un iPhone 5C vers le Royaume-Uni

Au vu du nombre de complaintes visibles sur les forums Apple, principalement suite à des surcoûts dûs à de multiples envois de SMS à l'étranger (il suffit de taper le numéro de téléphone dans un moteur de recherche), il semblerait que ce soit la procédure d'activation d'iMessage et de FaceTime qui soit ici coupable de cet envoi SMS indélicat.

5 Conclusion

Les modems mobiles s'appuient tous sur un système d'exploitation minimaliste, mais reste néanmoins des exécutables complexes. Leur analyse d'un point de vue sécurité est délicate du fait de l'hétérogénéité des systèmes existants, du peu de documentation et d'information publique (le développement de modems mobiles est un « business » très réservé et très lucratif) mais aussi de l'accès difficile au composant qui l'exécute comme aux protocoles de communications qu'il prend en charge. Il existe malgré tout des solutions peu coûteuses, à la portée de petits laboratoires ou de petites entreprises, qui permettent de les tester assez efficacement. Certaines approches et possibilités sont présentées dans les chapitres 2 et 3.

Même si les technologies de réseaux mobiles de deuxième génération demeurent très vulnérables aux écoutes passives et interceptions actives, des efforts considérables ont été fait lors du développement des normes mobiles 3G et LTE, afin de renforcer la sécurité des échanges et éviter toute intrusion au niveau des interfaces radio. Cependant, comme on peut le voir suite à la lecture du chapitre 4, les problèmes de sécurité, bogues et vulnérabilités demeurent bien présents dans les modems les plus récents, quels que soient les constructeurs de modems comme de téléphones, éditeurs logiciels et autres intégrateurs. Certaines de ces vulnérabilités peuvent permettre le contournement des mécanismes de sécurité de la 3G et du LTE, et faciliter ainsi l'interception des communications radio.

Il en ressort qu'une attention particulière est nécessaire de la part de tous les acteurs de la chaîne mobile (fabricants de modems, constructeurs et intégrateurs de téléphones mobiles, opérateurs de réseaux mobiles et autres distributeurs accrédités, et enfin, utilisateurs et abonnés aux services mobiles) pour veiller à la bonne sécurisation de ces systèmes, utilisés par des dizaines de millions de personnes à travers le monde. Les fabricants de modems sont le plus souvent ouverts à la remontée d'information, et prêts à patcher leur logiciel rapidement ; de même, de nombreux fabricants de terminaux intègrent et distribuent rapidement les mises à jour pour leur modem. La correction des bogues et vulnérabilités dans les modems, ainsi que leur mise à jour régulière, doivent être faites au même titre que pour les systèmes d'exploitation grand public.

6 Remerciements

Je tiens à remercier :
- l'Agence Nationale de la Sécurité des Systèmes d'Information, qui me donne l'opportunité de réaliser ce travail et de le publier ;
- Christophe Devine, qui participe activement à ces recherches et a notamment contribué à cet article ;
- José Lopes Esteves et José Araujo, pour leur relecture attentive.

7 Annexes

7.1 Détail des protocoles de signalisation

RRC : Radio Ressources Configuration

Il s'agit d'une appellation commune pour l'ensemble des protocoles de signalisation radio 2G / 3G / 4G permettant la sélection et la renégociation des caractéristiques de l'interface radio, et donc l'établissement et le maintien des canaux de communication radio duplex. Cette signalisation est utilisée exclusivement entre les terminaux et les réseaux d'accès radio (BTS-BSC en 2G, NodeB-RNC en 3G, eNodeB en 4G). On peut distinguer deux grands types de signalisation RRC : celle diffusée par les antennes sur les canaux descendants ou celle sur les canaux montants non dédiés (cela inclut les canaux de broadcast, de paging sur le sens descendant, et le canal d'accès aléatoire sur le sens montant), et celle utilisée entre un terminal spécifique et le réseau d'accès radio sur un canal duplex dédié.

Par ailleurs, en 3G et 4G, la signalisation RRC sert également à encapsuler la signalisation utilisée par le cœur de réseau. Ceci permet aux réseaux d'accès radio 3G et 4G de ne pas avoir à interpréter les contenus en provenance ou à destination du cœur de réseau, et de simplement les intégrer ou les extraire des messages RRC.

RRC GSM :

Lors de la réalisation de la norme GSM, les descriptions restent encore simples : les messages de signalisation sont tous décrits dans des tableaux, indiquant les structures des éléments d'information (Value, Length-Value, Tag, Tag-Value, ou Tag-Length-Value), que ce soit pour les canaux en diffusion comme pour les canaux dédiés. La norme 3GPP TS 04.08 (jusqu'en version 5 pour le seul GSM) décrit dans le chapitre 9.1 toutes les structures de messages de signalisation RRC pour le GSM. Un exemple est donné dans le listing 11.

```
IEI  Information element        Type / Reference       Presence  Format  length

     L2 Pseudo Length           L2 Pseudo Length          M        V       1
     10.5.2.19

     RR management              Protocol Discriminator    M        V       1/2
     Protocol Discriminator     10.2

     Skip Indicator             Skip Indicator            M        V       1/2
     10.3.1

     Immediate Assignment       Message Type              M        V       1
     Message Type               10.4

     Page Mode                  Page Mode                 M        V       1/2
     10.5.2.26
```

```
    Spare Half Octet          Spare Half Octet         M       V      1/2
    10.5.1.8

    Channel Description       Channel Description      M       V      3
    10.5.2.5

    Request Reference         Request Reference        M       V      3
    10.5.2.30

    Timing Advance            Timing Advance           M       V      1
    10.5.2.40

    Mobile Allocation         Mobile Allocation        M       LV     19
    10.5.2.21

7C  Starting Time             Starting Time            O       TV     3
    10.5.2.38

    IA Rest Octets (frequency IA Rest Octets           M       V      011
    parameters, before time)  10.5.2.16
```

Listing 11. Structure du message RRC Immediate Assignment, émis sur le canal de paging pour l'assignation d'un canal dédié à un terminal (TS 04.08, section 9.1.18)

RRC GPRS :

Lors du développement des normes GPRS, puis EDGE, un nouveau type de représentation des messages a été utilisé : le CSN.1 (Concrete Syntax Notation One). Il s'agit d'un pseudo-langage de description de structures bit à bit. Il permet une plus grande compacité des messages, mais est plus délicat à interpréter. Lorsque le réseau GPRS réutilise une infrastructure GSM existante, les canaux de diffusion des antennes-relais GSM sont réutilisés pour diffuser les informations GPRS. Ainsi, la description d'un réseau GPRS est partiellement insérée dans le padding des messages de signalisation GSM. Les normes 3GPP TS 04.08 et TS 04.18 (à partir de la version 6 pour le GPRS), TS 04.60, puis TS 44.18 et TS 44.60 contiennent les descriptions de tous les messages de signalisation RRC pour le GPRS et l'EDGE.

On peut prendre pour exemple la description du champ SI4 rest octets (TS 04.08, section 10.5.2.35) qui définit le contenu des bits de padding de la balise GSM System Information 4 diffusée par les antennes-relais. Le listing 12 illustre le type de syntaxe de la notation CSN.1.

```
<SI4 Rest Octets> ::=   <SI4 Rest Octets_O>
            {L<Break indicator>| H<SI Rest Octets_S}
            <spare padding >;


<SI4 Rest Octets_O> ::= <Optional selection parameters>
            <Optional Power offset>
            {L | H < GPRS Indicator  >};


<SI4 Rest Octets_S> ::= {L | H <LSA Parameters>}
            {L | H <Cell  Identity: bit(16)>}
            {L | H <LSA ID information  >};
```

```
<Break Indicator >::=   L |  H ;
[...]
```

Listing 12. Structure du début de l'élément d'information RRC SI4 Rest Octets, émis sur le canal de broadcast dans le padding du message System Information 4 (TS 04.08 v.7, section 9.1.18)

Par la suite et lors du développement des réseaux UMTS puis LTE, cette notation CSN.1 a de nouveau été employée pour ajouter des descriptions des cellules 3G et LTE voisines dans les trames balises GSM : ceci a donné lieu à des complexifications considérables, comme le montre la description du message System Information 2 Quater (voir TS 44.18, section 10.5.2.33b) : une description CSN.1 de 300 lignes décrit une structure de 20 octets. On y trouve en plus des structures itératives telles que celle présentée dans le listing 13.

```
< UTRAN FDD Description struct  >::=
    { 0 | 1 < Bandwidth_FDD : bit (3) }
    { 1 < Repeated UTRAN FDD Neighbour Cells : Repeated UTRAN FDD Neighbour Cells struct
         >> } ** 0 ;

< Repeated UTRAN FDD Neighbour Cells struct > ::=
    0 < FDD-ARFCN : bit (14) >
        -- The value ‘’1 was used in an earlier version of the protocol and shall not be
            used.
    < FDD_Indic0 : bit >
    <  NR_OF_FDD_CELLS: bit (5) >
    < FDD_CELL_INFORMATION  Field: bit(p(NR_OF_FDD_CELLS)) > ; -- p(x) defined in  table9
         .1.54.1
```

Listing 13. Exemple de structure CSN.1 itérative pour la description de cellules 3G voisines dans l'élément SI2Quater Rest Octets

La norme TS 44.060 définit un grand nombre d'éléments d'information utilisés dans la signalisation GPRS. La section 12.10a décrit l'élément permettant de transmettre au mobile les paramètres du canal GPRS dédié qui lui est assigné. On y trouve des structures CSN.1 récursives, telles que celle présentée dans le listing 14.

```
< GPRS Mobile Allocation IE > ::=
    < HSN : bit (6) >
    { 0 | 1 < RFL number list : < RFL number list struct > > }
    { 0 < MA_LENGTH : bit (6) >
        < MA_BITMAP : bit (val(MA_LENGTH) + 1) >
    | 1 { 0 | 1 < ARFCN index list : < ARFCN index list struct > > } } ;
< RFL number list struct > ::=
    < RFL_NUMBER : bit (4) >
    { 0 | 1 < RFL number list struct > } ;
< ARFCN index list struct > ::=
    < ARFCN_INDEX : bit (6) >
    { 0 | 1 < ARFCN index list struct > } ;
```

Listing 14. Exemple de structure CSN.1 récursive pour l'assignation d'un canal GPRS dédié à un mobile

Il faut noter que tous les mobiles compatibles GPRS, sans exception, doivent contenir les encodeurs et décodeurs CSN.1 adéquats afin de pouvoir interpréter ce type d'information transmis par les réseaux GSM / GPRS / EDGE.

RRC UMTS :

Lors du développement de la norme UMTS et de son évolution, le 3GPP a opté pour l'usage d'une syntaxe ASN.1 pour la description du protocole RRC. L'encodage BASIC-PER unaligned (PER pour « Packed Encoded Rules », et non aligné sur l'octet) est utilisé pour le transfert des messages entre terminal et RAN : ceci procure une compression importante des messages. De même qu'un réseau GSM peut diffuser des informations sur les cellules UMTS ou LTE voisines, le réseau UMTS peut diffuser des informations à propos des cellules GSM / GPRS ou LTE voisines. La (colossale) norme 3GPP TS 25.331 décrit le protocole RRC de l'UMTS.

L'avantage de l'usage ASN.1 est que le langage est bien mieux défini que le CSN.1 (des compilateurs ASN.1 existent, qui peuvent « compiler » les spécifications 3GPP en langage C ou autre). L'encodage PER unaligned permet également de minimiser l'utilisation de la bande passante. Malgré tout, la norme RRC de l'UMTS est d'une complexité très importante étant donné l'étendue des fonctionnalités proposées sur les interfaces radio UMTS et HSPA. Le listing 15 donne un exemple de syntaxe ASN.1 pour un message de signalisation RRC.

```
-- **************************************************
--
-- RRC CONNECTION SETUP
--
-- **************************************************

RRCConnectionSetup ::= CHOICE {
    r3                              SEQUENCE {
        rrcConnectionSetup-r3           RRCConnectionSetup-r3-IEs,
        laterNonCriticalExtensions          SEQUENCE {
            -- Container for additional R99 extensions
            rrcConnectionSetup-r3-add-ext       BIT STRING      OPTIONAL,
            nonCriticalExtensions               SEQUENCE {}     OPTIONAL
        }   OPTIONAL
    },
    later-than-r3                   SEQUENCE {
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
        criticalExtensions              SEQUENCE {}
    }
}

RRCConnectionSetup-r3-IEs ::= SEQUENCE {
    -- TABULAR: Integrity protection shall not be performed on this message.
    -- User equipment IEs
        initialUE-Identity              InitialUE-Identity,
        rrc-TransactionIdentifier       RRC-TransactionIdentifier,
        activationTime                  ActivationTime                      OPTIONAL,
        new-U-RNTI                      U-RNTI,
        new-c-RNTI                      C-RNTI                              OPTIONAL,
        rrc-StateIndicator              RRC-StateIndicator,
        utran-DRX-CycleLengthCoeff      UTRAN-DRX-CycleLengthCoefficient,
        -- TABULAR: If capabilityUpdateRequirement is not present, the default value
        -- defined in 10.3.3.2 shall be used.
```

```
        capabilityUpdateRequirement     CapabilityUpdateRequirement         OPTIONAL,
    -- Radio bearer IEs
        srb-InformationSetupList        SRB-InformationSetupList2,
    -- Transport channel IEs
        ul-CommonTransChInfo            UL-CommonTransChInfo                OPTIONAL,
        -- NOTE: ul-AddReconfTransChInfoList should be optional in later versions
        -- of this message
        ul-AddReconfTransChInfoList     UL-AddReconfTransChInfoList,
        dl-CommonTransChInfo            DL-CommonTransChInfo                OPTIONAL,
        -- NOTE: dl-AddReconfTransChInfoList should be optional in later versions
        -- of this message
        dl-AddReconfTransChInfoList     DL-AddReconfTransChInfoList,
    -- Physical channel IEs
        frequencyInfo                   FrequencyInfo                       OPTIONAL,
        maxAllowedUL-TX-Power           MaxAllowedUL-TX-Power               OPTIONAL,
        ul-ChannelRequirement           UL-ChannelRequirement               OPTIONAL,
        dl-CommonInformation            DL-CommonInformation                OPTIONAL,
        dl-InformationPerRL-List        DL-InformationPerRL-List            OPTIONAL
```

Listing 15. Exemple de description ASN.1 du message RRC UMTS Connection Setup (TS 25.331, section 11.2) pour l'établissement d'un canal de signalisation dédié avec un mobile

RRC LTE :
Le développement de la norme LTE a suivi les mêmes principes que l'UMTS. La syntaxe ASN.1 BASIC-PER unaligned est réutilisée. Le protocole RRC et l'ensemble des messages de signalisation LTE sont décrits dans la norme 3GPP TS 36.331.

MM / CC : Mobility Management et Call Control

Les protocoles de gestion de la mobilité et du contrôle d'appel ont été développés dès la première norme GSM, ils sont spécifiques au domaine circuit CS des réseaux mobiles. Ils décrivent les procédures nécessaires afin qu'un terminal :

- s'enregistre auprès ou se détache du réseau mobile ;
- s'authentifie auprès de celui-ci ;
- récupère une identité temporaire qui sera ensuite renouvelée régulièrement ;
- reporte ses changements de localisation au sein du RAN auprès du cœur de réseau ;
- reçoive et émette des appels.

Même si ces différentes étapes semblent triviales, le fait de dépendre de l'interface radio et de la carte SIM pour pouvoir échanger de tels messages, ainsi que quelques subtilités supplémentaires rendent le protocole relativement complexe en terme de fonctionnement : la figure 8, extraite de la norme 3GPP TS 24.008, présente le cycle de transitions du seul protocole MM pour un mobile.

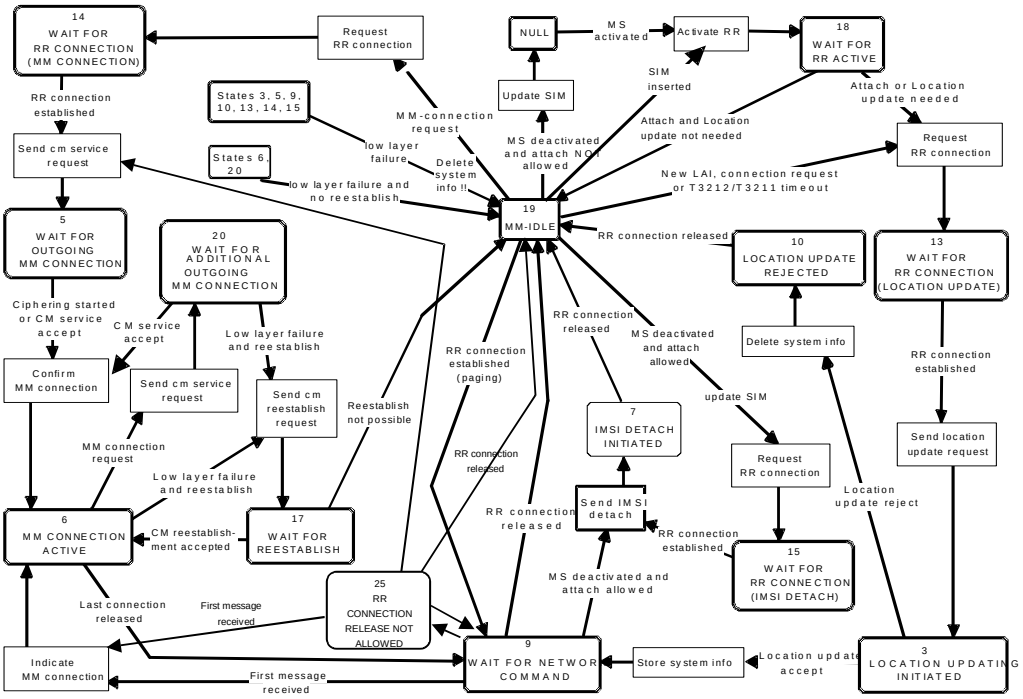

Figure 8. Machine à état pour la gestion du protocole MM par le terminal mobile

Ces deux protocoles MM et CC ont été modifiés au fur et à mesure des évolutions du GSM et du développement de la 3G (qui intègre par exemple une nouvelle méthode d'authentification). La norme qui les décrit est donc commune à ces technologies, il s'agit de la norme 3GPP TS 24.008.

SS : Supplementary Services

Les services supplémentaires sont une suite de services du domaine circuit CS, supportés selon les cas par de nombreux réseaux et un grand nombre de terminaux mobiles. Ces services sont normalisés dans de multiples spécifications du 3GPP : TS 24.010 pour les aspects généraux, puis TS 24.072 jusqu'à TS 24.096. On retrouve par exemple les services suivants :

- le transfert d'appel (vers une boîte vocale ou un numéro tiers) ;
- la suspension d'appel ;
- l'interdiction d'appels entrants ou sortants ;
- l'identification du numéro appelant ;
- l'USSD (Unstructured Supplementary Service Data).

La plupart de ces services sont utilisés pour le traitement des appels voix. Cependant, l'USSD est différent puisqu'il s'agit d'un service de

données très simpliste (norme 3GPP TS 24.090) : il permet l'accès à des services interactifs de l'opérateur via l'émission et la réception de messages très proches de ceux du SMS, et la navigation dans des menus simples pour l'utilisateur via une sélection de codes numériques. Il est intéressant de noter que dans certains cas d'usage et de mobilité d'un abonné, le service USSD accédé par le terminal peut être hébergé directement sur le HLR de l'opérateur de l'abonné.

Par exemple en France, l'opérateur Orange propose un service USSD pour consulter le solde de son abonnement (#123#), mais aussi d'autres services USSD, tel que le « chat SMS » (#102#) qui peut ressembler à un irc sur SMS.

GMM / SM : GPRS Mobility Management et Session Management

Les protocoles de gestion de la mobilité GPRS et d'établissement de sessions paquet ont été développés dès la première norme GPRS, ils sont spécifiques au domaine PS des réseaux mobiles. Ils décrivent les procédures nécessaires afin qu'un terminal :

- s'enregistre auprès ou se détache du réseau mobile (cela peut être implicite selon les procédures du protocole MM) ;
- s'authentifie auprès de celui-ci ;
- établisse et mette en œuvre le chiffrement du canal radio (uniquement dans le cas du GPRS et de l'EDGE, ou le chiffrement du canal est effectif entre terminal et SGSN, et non dans le cas de la 3G) ;
- récupère une identité temporaire qui sera ensuite renouvelée régulièrement ;
- reporte ses changements de localisation au sein du RAN auprès du cœur de réseau ;
- établisse des sessions de transfert de données par paquets (avec gestion de l'APN et de l'adressage IP).

Ces deux protocoles ont évolué au fur et à mesure des évolutions du GPRS et du développement de la 3G. Ils sont également décrits dans la norme 3GPP TS 24.008. L'authentification avec le domaine PS est systématiquement distincte de celle avec le domaine CS, de même que l'identité temporaire attribuée au terminal ; de plus la localisation du terminal dans le domaine PS intègre un paramètre RAC (Routing Area Code), supplémentaire au paramètre LAC (Location Area Code) de localisation dans le domaine CS.

EMM / ESM : Evolved Mobility Management et Evolved Session Management

Puisque le développement de la norme LTE a entraîné une refonte complète du réseau mobile, tant au niveau de l'accès radio que du cœur de réseau, des nouveaux protocoles de gestion de la mobilité et de la session ont été développés. Ils conservent des similitudes avec les protocoles GMM et SM du GPRS.

La principale particularité de la signalisation de cœur de réseau LTE (aussi appelée signalisation NAS), échangée entre terminaux et MME, est qu'elle comporte des mécanismes de sécurité propres (chiffrement et contrôle d'intégrité), en sus de la sécurisation du canal radio entre terminal et eNodeB. Ces deux protocoles EMM et ESM sont définis dans la norme 3GPP TS 24.301.

SMS : Short Message Service

Le protocole SMS est particulier : il s'agit de données utilisateurs (les messages courts) transportées dans des canaux de signalisation. Ce service a été développé très rapidement après le lancement de la technologie GSM, et demeure aujourd'hui présent dans les réseaux GPRS, 3G et LTE, sous la même forme. Les normes qui décrivent le fonctionnement de ce service sont les documents 3GPP TS 24.011 concernant le format des messages de signalisation et les procédures de transport associées, TS 24.040 pour le format des messages courts eux-mêmes et les procédures applicatives, et TS 23.038 pour l'encodage des caractères constituant le message court.

Malgré son apparente simplicité, le SMS propose un très grand nombre de possibilités, telles que :

- l'échange de messages courts encodés selon différents alphabets (européen, asiatiques, ...), utilisant des caractères sur 7 bits, 8 bits ou 16 bits ;
- l'échange de données utilisateurs selon des protocoles particuliers (teletex, telefax, ERMES, X.400, ...), et même d'emails de type SMTP via des passerelles de transcodage ;
- l'échange de messages de service destinés à reconfigurer le terminal ou la carte (U)SIM (via des instructions de SIM-Toolkit), sans visibilité pour l'utilisateur.

Il existe également un mécanisme pour transmettre des SMS génériques sur les canaux de diffusion des antennes-relais GSM (appelé SMS Cell Broadcast). Il n'est pas tellement utilisé en Europe mais on peut le trouver dans certains pays d'Asie ou d'Afrique. Les mobiles doivent alors être

configurés spécifiquement pour accepter et lire ces messages diffusés par les cellules.

Services de géolocalisation

Les mécanismes de géolocalisation au sein des réseaux mobiles ont été introduits assez rapidement après les débuts de la technologie GSM. Ils ont été développés principalement pour que les mobiles puissent transmettre leur localisation précise aux plate-formes de gestion des appels d'urgence. En parallèle, des systèmes applicatifs pouvant employer ces données de positionnement ont été introduits afin de donner la possibilité aux opérateurs de fournir des services supplémentaires.

Ils mettent en œuvre différents protocoles :

- RRLP : Radio Ressource LCS (Location Services) Protocol, qui permet de calculer le positionnement d'un mobile et d'échanger les informations nécessaires entre les terminaux à positionner et les réseaux 2G et 3G. Ce protocole décrit les différentes méthodes de calcul de position possibles pour un terminal et est encapsulé dans la signalisation RRC. La norme 3GPP TS 44.031 décrit le protocole pour les réseaux 2G, tandis que la norme du protocole RRC 3G (TS 25.331) intègre directement ces mécanismes de détermination de la position ;
- LPP : LTE Positioning Protocol, qui permet de calculer le positionnement d'un terminal, de la même manière que pour RRLP. La différence étant qu'en LTE il n'y a pas de contrôleurs radio, les messages LPP sont échangés entre le terminal et le cœur de réseau LTE encapsulés dans des messages de signalisation EMM. La norme TS 36.355 définit le protocole LPP ;
- LCS : Location Services, Supplementary Services, défini dans la norme 3GPP TS 24.030 pour une utilisation « applicative » des données de positionnement par le cœur de réseau 2G / 3G, et dans la norme TS 24.171 pour une utilisation par le cœur de réseau LTE.

La position d'un terminal peut être calculée directement par celui-ci, à partir des données du réseau mobile et de systèmes de géolocalisation satellite (GPS, GLONASS) intégrés au terminal. Une assistance peut être fournie par le réseau pour que le terminal calcule lui-même sa position, ou à l'inverse le terminal peut fournir ses relevés bruts au réseau qui calculera alors la position du terminal. Enfin, si aucune information n'est fournie par le terminal, le réseau peut calculer une position approximative du terminal à partir des derniers accès actifs sur ses cellules radio.

Avec l'avènement des services de données et de la 3G, un nouveau protocole permettant le calcul de positionnement a été défini : SUPL (Secure User-Plane Location protocol). Il permet à un mobile de contacter un serveur, à travers un lien IP / UDP, et d'échanger avec lui des données similaires à ce qui est fait en RRLP ou LPP. Le protocole SUPL est normé par l'organisme OMA.

7.2 Informations techniques récupérées lors de l'analyse statique d'images exécutables de modems

Liste des tâches lancées au démarrage d'un modem Intel

```
# task_name
l1u:1
umacul:1
umacdl:1
umacc:1
urlcul:1
urlcdl:1
urlcc:1
urrcbp:1
urrcdc:1
urrcm:1
ubmc:1
urabmupdcp:1
l1g:1
dll:1
dll:2
llc:1
mac:1
rlc:1
rrc:1
grr:1
rrl:1
atc:1
dch:1
df2:1
drl:1
dtn:1
dtt:1
gmm:1
gmr:1
itx:1
mmc:1
mma:1
mme:1
mmr:1
mnc:1
mng:1
mni:1
mnm:1
mnp:1
```

```
mns:1
oms:1
pch:1
snp:1
sim:1
smr:1
mmi:1
mdh:1
pbh:1
xdr:1
gps:1
mtc:1
biph:1
mon
ata
ipr_rx1
ipr_rx2
ipr_rx3
mux
io_evt
```

Listing 16. Liste des noms des tâches telles que lancées par ThreadX au démarrage du modem Intel XMM

Liste des tâches lancées au démarrage d'un modem Broadcom

```
# task_id  task_name
0  l1u:0
1  ubmc:0
2  umacc:0
3  umacdl:0
4  umacul:0
5  urabm:0
6  urlcc:0
7  urlcdl:0
8  urlcul:0
9  urrcbp:0
10  urrcdc:0
11  urrcm:0
12  dll:0
13  dll:1
14  gmm:0
16  gmr:0
17  grr:0
19  llc:0
20  mac:0
22  mma:0
24  mmc:0
26  mme:0
28  mmr:0
30  rlc:0
32  rrc:0
34  smr:0
```

```
35  snp:0
36  gl1s:0
37  mmreg:0
39  mncc:0
41  mns:0
43  mnss:0
45  oms:0
47  dch:0
48  dtt:0
49  dtn:0
50  drl:0
51  df2:0
52  agps4u:0
53  rrl:0
54  sim:0
56  hucm:0
57  vmc:0
```

Listing 17. Liste des identifiants et noms des tâches telles que lancées par ThreadX au démarrage du modem Broadcom

Références

1. Analyse de la sécurité de l'architecture Hexagon . https://rpw.io/slides/rpw-30c3-hexagon.pdf.
2. Analyseur radio pour les modems Samsung Kalmia . http://labs.p1sec.com/author/ramtin.
3. Analyseur radio pour les terminaux Samsung équipés de modems Intel . https://github.com/2b-as/xgoldmon.
4. API Brew Qualcomm. https://developer.brewmp.com/resources/tech-guides/media-technology-guide/high-level-architecture.
5. Apple iPhone 5S désassemblé. http://www.ifixit.com/Teardown/iPhone+5s+Teardown/17383.
6. Brevet NVidia sur l'architecture de processeur. http://patentimages.storage.googleapis.com/pdfs/US20050223196.pdf.
7. Debugger Hexagon de Lauterbauch . http://www2.lauterbach.com/pdf/debugger_hexagon.pdf.
8. Debugger pour le modem Qualcomm des clés USB ICON 225 . https://code.google.com/p/qcombbdbg/.
9. Documentation de réference du SDK Hexagon (nécessite un enregistrement sur le site Qualcomm) . https://developer.qualcomm.com/download/hexagon-sdk-programmers-reference.zip.
10. Déverrouillage de l'iPhone 2G par geohot . http://www.tayloredge.com/museum/museum/IPhone.pdf.
11. Environnements de développement pour Hexagon . https://www.codeaurora.org/patches/quic/hexagon/.

12. Evolution des mécanismes de déverrouillage iPhone. http://www.slideshare.net/gadgetsdna/musclenerd-evolution-of-iphone-baseband-and-unlocks.
13. Fuzzing GSM avec OpenBSC . http://events.ccc.de/congress/2009/Fahrplan/attachments/1503_openbsc_gsm_fuzzing.pdf.
14. Google Nexus 5 désassemblé . http://www.ifixit.com/Teardown/Nexus+5+Teardown/19016.
15. HTC One désassemblé . http://electronics360.globalspec.com/article/3447/htc-one-99htt223-00-mobile-handset-teardown.
16. Informations sur l'architecture Qualcomm L4 et AMSS . https://code.google.com/p/docl4amss/.
17. iOS Hacker's Handbook, chapitre 11. http://books.google.fr/books?id=M7DVvMx0b6kC&pg=PA327&lpg=PA327.
18. Liste de terminaux supportant ou non l'indicateur de non-chiffrement radio . http://security.osmocom.org/trac/wiki/WillMyPhoneShowAnUnencryptetConnection.
19. Nokia N900 désassemblé. http://electronics360.globalspec.com/article/2213/nokia-n900-mobile-phone-teardown.
20. Plug-in IDA pour la prise en charge de l'architecture Hexagon . https://github.com/gsmk/hexagon.
21. Projet contenant un miroir codeaurora, avec fichiers sources Qualcomm . https://github.com/dpavlin/huawei/.
22. Projet CryptoMobile . https://github.com/mitshell/CryptoMobile.
23. Projet DCT3-GSMTAP . http://bb.osmocom.org/trac/wiki/dct3-gsmtap.
24. Projet EURECOM openairinterface.org . http://www.openairinterface.org.
25. Projet GNU Radio LTE receiver . https://github.com/kit-cel/gr-lte.
26. Projet JTAGulator . http://www.grandideastudio.com/portfolio/jtagulator/.
27. Projet libLTE . https://github.com/ismagom/libLTE.
28. Projet libmich . https://github.com/mitshell/libmich.
29. Projet LTEENB de Fabrice Bellard . http://bellard.org/lte/.
30. Projet MADos . http://www.g3gg0.de/wordpress/projects/mados/.
31. Projet Nokix . http://nokix.sourceforge.net/.
32. Projet OpenBSC . http://openbsc.osmocom.org/trac/wiki/OpenBSC.
33. Projet OpenBTS . http://openbts.org.
34. Projet OpenLTE . http://openlte.sourceforge.net.
35. Projet Osmocom-bb . http://bb.osmocom.org/trac.
36. Présentation de Colin Mulliner et Nico Golde sur le fuzzing SMS . http://ngolde.de/sms/smsodeath_mulliner_golde_cansecwest2011.pdf.
37. Présentation de Guillaume Delugré et Sebastien Dudek sur le fuzzing GSM. http://archive.hack.lu/2012/Fuzzing_The_GSM_Protocol_Stack_-_Sebastien_Dudek_Guillaume_Delugre.pdf.
38. Présentation de l'architecture spécifique pour les modems NVidia-Icera . http://www.iet-cambridge.org.uk/arc/seminar05/slides/SimonKnowles.pdf).

39. Présentation de Ralf-Philipp Weinmann à hack.lu 2010 . http://archive.hack.lu/2010/Weinmann-All-Your-Baseband-Are-Belong-To-Us-slides.pdf.
40. Présentation de Ralf-Philipp Weinmann à USENIX 2012 . https://www.usenix.org/system/files/conference/woot12/woot12-final24.pdf.
41. Présentation par Luis Mira de méthodes de déverrouillage d'iPhone. http://reverse.put.as/wp-content/uploads/2011/06/basebandplayground-luismiras.pdf.
42. Scripts de désassemblage pour modems Intel . https://github.com/rpw/flsloader.
43. Site du fabricant de cartes radio Ettus . http://home.ettus.com.
44. Site du fabricant de cartes radio Fairwaves . https://fairwaves.co.
45. Site du fabricant de cartes radio MyriadRF . http://myriadrf.org.
46. Site du fabricant de cartes radio Nuand . http://nuand.com.
47. Sony Xperia Z désassemblé . http://electronics360.globalspec.com/article/3563/sony-xperia-z-c6603-mobile-handset-teardown.
48. Sources du noyau pour Samsung Galaxy Y . https://github.com/rajamalw/galaxy-s5360/.
49. WIKI iPhone de la Chronic Dev team . http://theiphonewiki.com/wiki/Main_Page.
50. 3GPP specification numbering. http://www.3gpp.org/specifications/specification-numbering.