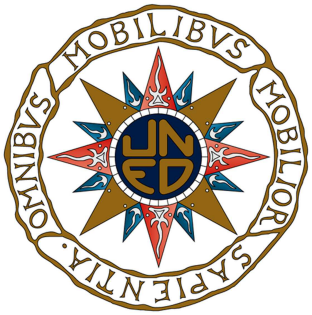
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingenieros Informáticos
Departamento de Informática y Automática



EUROPEAN COMMISSION
Joint Research Centre
Institute for the Protection and Security of the Citizen

SEGURIDAD Y PRIVACIDAD EN LAS COMUNICACIONES INALÁMBRICAS PERSONALES

TESIS DOCTORAL

José Ignacio Sánchez Martín
Ingeniero Informático
2014

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Escuela Técnica Superior de Ingenieros Informáticos
Departamento de Informática y Automática

SEGURIDAD Y PRIVACIDAD EN LAS COMUNICACIONES INALÁMBRICAS PERSONALES

D. José Ignacio Sánchez Martín
Ingeniero Informático por la Universidad de Deusto

Directores:

Dr. D. Sebastián Dormido Canto

Profesor Titular de Universidad del Departamento de Informática y Automática de la Universidad Nacional de Educación a Distancia

Dr. D. José Sánchez Moreno

Profesor Titular de Universidad del Departamento de Informática y Automática de la Universidad Nacional de Educación a Distancia

A Ainhoa e Irati

Agradecimientos

Dirijo mis primeros agradecimientos a mis pequeñas Irati y Ainhoa por cederme su tiempo para la elaboración de esta tesis y ser mi fuente de inspiración. A ellas va dedicada esta tesis.

Agradezco a mi esposa Ainara López su comprensión y apoyo incondicional durante el largo período de la elaboración de la tesis. A mis padres les agradezco su apoyo y sabios consejos que me han guiado durante mi vida académica y profesional.

Un agradecimiento especial a los directores de mi tesis, Dr. Sebastían Dormido Canto y Dr. José Sánchez Moreno, por sus valiosos comentarios, su guía y su firme apoyo durante el proceso de la elaboración de la tesis, sin los cuales ésta no habría sido posible.

A Jean Pierre Nordvik, jefe de la *Digital Citizen Security Unit* del *Institute for the Protection and Security of the Citizen* del *Joint Research Centre* de la Comisión Europea, así como a todos los compañeros y compañeras de unidad, verdaderos genios con los que tengo el privilegio de trabajar. Un agradecimiento en particular para los coautores de mis trabajos y compañeros de investigación, Gianmarco Baldini, Laurent Beslay, Iwen Coisel, Igor Nai Fovino, David Shaw y Riccardo Satta, entre otros, por su pasión y compromiso incondicional con la ciencia y la seguridad de la información.

Por supuesto, un profundo agradecimiento a toda la comunidad investigadora en la rama de la seguridad de la información y en especial a los autores de los trabajos citados en la bibliografía, gigantes sobre cuyos hombros se apoya esta tesis.

Resumen

El objetivo de la presente tesis es determinar si el conjunto de las principales tecnologías de transmisión inalámbrica habitualmente utilizadas en comunicaciones personales ofrece actualmente una protección efectiva de la seguridad y privacidad de las mismas.

La tesis se centra en la investigación de la seguridad de las tecnologías DECT, GSM y WiFi, protocolos de referencia en los respectivos escenarios de utilización de medios inalámbricos para comunicaciones de carácter personal. Dichas tecnologías, a diferencia de las conexiones tradicionales efectuadas mediante cable, conllevan una serie de desafíos de privacidad específicos a la naturaleza inherente del medio compartido de propagación, que son objeto de estudio en la presente tesis.

La seguridad de dichos protocolos es investigada en profundidad, siguiendo un enfoque tanto teórico como experimental, con el objetivo de determinar su capacidad efectiva para la protección de la privacidad de las comunicaciones personales, dados los recientes avances en técnicas de criptoanálisis, la creciente capacidad de cómputo de los ordenadores personales y la aparición de una nueva generación de dispositivos de Radio Definida por Software (SDR) de bajo coste.

Con el objetivo de establecer la capacidad que dichos dispositivos SDR poseen para la interceptación de comunicaciones digitales, se desarrolla una novedosa implementación software así como una serie de experimentos que, aprovechándose de vulnerabilidades existentes en los protocolos objeto de estudio, demuestran su efectividad en la interceptación práctica de comunicaciones DECT y GSM.

En lo referente a los protocolos criptográficos previstos en DECT, GSM y WiFi, se explora su resistencia ante diferentes técnicas de criptoanálisis, demostrándose experimentalmente para todos ellos la viabilidad de interceptación de comunicaciones cifradas en escenarios reales de utilización. Así mismo, se describen y desarrollan nuevas técnicas de criptoanálisis así como de ataque a las limitaciones y debilidades en los protocolos criptográficos, que

demuestran experimentalmente ser efectivas para el compromiso de la privacidad de las comunicaciones cifradas.

De los resultados experimentales obtenidos en la presente tesis se desprende que el conjunto de protocolos DECT, GSM y WiFi sufre actualmente serias limitaciones en su capacidad para proteger las comunicaciones personales efectuadas mediante medios inalámbricos, incluso ante atacantes casuales con escasos recursos a su alcance.

Como conclusión de la tesis, se proponen una serie de contramedidas aplicables a los estándares e implementaciones actuales, dirigidas a minimizar el riesgo identificado para la seguridad y privacidad de las comunicaciones personales, así como un conjunto de recomendaciones dirigidas a las futuras generaciones de estándares, formuladas en base a los resultados obtenidos en la investigación realizada.

Abstract

The aim of this thesis is to determine whether the dominant set of wireless technologies used for the transmission of personal communications currently offer sufficient protection against security and privacy threats.

The thesis concentrates on the DECT, GSM and WiFi technologies, which represent the reference protocols in their respective use case scenarios of wireless transmission of personal communications. These technologies, unlike traditional wired communications, bring a set of specific privacy challenges, inherent to the nature of their shared propagation media, that is subject of study in this thesis.

The security of these reference protocols is analytically researched, following both a theoretical and an experimental approach. The aim is to determine their effectiveness for the protection of the privacy of personal communications, given the recent advances in cryptanalysis, increasing computing power of personal computers and the emerging generation of low-cost Software Defined Radios (SDR).

With the purpose of establishing the capacity that these SDR devices possess for the interception of digital personal communications, a novel software implementation is developed and a set of experiments are carried out, taking advantage of existing vulnerabilities in the protocols, that prove to be effective in the practical interception of DECT and GSM communications.

With regards to the DECT, GSM and Wi-Fi cryptographic protocols, their resistance to several cryptanalysis techniques is explored. The experimental validation follows for all three to show the practicability of the interception of encrypted communications in realistic scenarios. New cryptanalysis techniques and attacks against the extant weaknesses and limitations in these ubiquitous cryptographic protocols are also described and developed, demonstrating experimentally their effectiveness in compromising the privacy of the encrypted communications.

From experimental results achieved it can be concluded that the reference set of DECT, GSM and Wi-Fi protocols currently suffer serious limitations in their capacity to safeguard personal communications that take place wirelessly, even in the case of casual attackers with limited resources.

In conclusion of the thesis, on the basis of the results obtained from the research carried out, is formulated a set of countermeasures applicable to the current protocols and implementations, in order to mitigate the identified privacy and security risk, as well as a set of recommendations addressed to the next generation of standards.

Índice general

Índice de tablas

Índice de figuras

Lista de abreviaturas

AGCH	*Access Grant Channel*
ARFCN	*Absolute Radio-Frequency Channel Number*
ARP	*Address Resolution Protocol*
ASIC	*Application Specific Integrated Circuit*
AuC	*Authentication Centre*
BCCH	*Broadcast Control Channel*
BCH	*Broadcast Channel*
BSC	*Base Station Controller*
BSS	*Base Station Subsystem*
BTS	*Base Transceiver Station*
CCCH	*Common Control Channel*
CI	*Common Interface*
CKSN	*Cipher Key Sequence Number*
CRC	*Cyclic Redundancy Check*
DCCH	*Dedicated Control Channel*
DECT	*Digital Enhanced Cordless Telecommunications*
DECT-ULE	*DECT Ultra Low Energy*
DHCP	*Dynamic Host Configuration Protocol*
DSAA	*DECT Standard Authentication Algorithm*
DSC	*DECT Standard Cipher*
DSP	*Digital Signal Processing*
EER	*Equal Error Rate*
EIR	*Equipment Identity Register*
EMC	*Equipment Manufacturers Code*
ETSI	*European Telecommunications Standards Institute*
FAC	*Final Assembly Code*
FACCH	*Fast Associated Control Channel*
FCCH	*Frequency Correction Channel*
FCS	*Frame Check Sequence*
FDMA	*Frequency Division Multiple Access*

FIR *Finite Impulse Response*

FP *Fixed Part*

FPGA *Field-Programmable Gate Array*

FPN *Fixed Part Number*

FS *Forward Secrecy*

GAP *Generic Access Profile*

GFSK *Gaussian Frequency-Shift Keying*

GMSC *Gateway Mobile Switching Center*

GMSK *Gaussian Minimum Shift Keying*

HLR *Home Location Register*

HMM *Hidden Markov Model*

HSM *Hardware Security Module*

HTML *HyperText Markup Language*

ICCID *Integrated Circuit Card Identifier*

ICV *Integrity Check Value*

IMEI *International Mobile Equipment Identity*

IMSI *InternationalMobile Subscriber Identity*

IoT *Internet of Things*

IPUI *International Portable User Identification*

ITU *International Telecommunication Union*

IV *Initialization Vector*

JVM *Java Virtual Machine*

kNN *k-Nearest Neighbors*

KSA *Key Scheduling Algorithm*

KSS *KeyStream Segment*

LNA *Low Noise Amplifier*

MCC *Mobile Country Code*

ME *Mobile Equipment*

Mhz *Megahertz*

MiTM *Man-in-The-Middle*

MNC *Mobile Network Code*

MS *Mobile Station*

MSC *Mobile Switching Center*

MSIN *Mobile Subscriber Identification Number*

MTU *Maximum Transmission Unit*

NG-DECT *New Generation DECT*

NSS *Network Switching Subsystem*

OTP *One Time Pad*

PCAP *Packet CAPture format*

PCH *Paging Channel*

PCMCIA *Personal Computer Memory Card International Association*

PFS *Perfect Forward Secrecy*

PIN *Personal Identification Number*

POTS *Plain Old Telephony Service*

PP *Portable Part*

PRGA *Pseudo-Random Generation Algorithm*

PRNG *PseudoRandom Number Generator*

PSN *Portable Equipment Serial Number*

PSTN *Public Switched Telephone Network*

PUK *Personal Unlock Key*

RACH *Random Access Channel*

RFPI *Radio Fixed Part Identifier*

RPE-LTP *Regular-Pulse Excitation Long-Term Predictor*

RPN *Radio fixed Part Number*

RTB *Red de Telefonía Básica*

SACCH *Slow Associated Control Channel*

SCH *Synchronization Channel*

SCK *Static Cipher Key*

SDCCH *Standalone Dedicated Control Channel*

SDR *Software Defined Radio*

SFH *Slow Frequency Hopping*

SIM *Subscriber Identity Module*

SNAP	*Subnetwork Access Protocol*
SNR	*Serial Number*
SSH	*Secure Shell*
SVM	*Support Vector Machine*
TAC	*Type Allocation Code*
TCH	*Traffic Channel*
TCH/F	*Full-Rate Traffic Channel*
TCH/H	*Half-Rate Traffic Channel*
TCP	*Transmission Control Protocol*
TMSI	*Temporary Mobile Subscriber Identity*
TMTO	*Trade-Memory-TradeOff*
TOR	*The Onion Router*
UAK	*User Authentication Key*
VLR	*Visitor Location Register*
WEP	*Wireless Equivalent Privacy*
WPA	*Wi-Fi Protected Access*
WPA-PSK	*Wi-Fi Protected Access - Pre-Shared Key*
WPS	*Wi-Fi Protected Setup*

Capítulo 1

Introducción

En los últimos años se ha producido una explosión sin precedentes en el despliegue global de tecnologías de comunicación de naturaleza inalámbrica, las cuales posibilitan actualmente una conexión ubicua y permanente, tanto a Internet como a la red de telefonía mundial. Este nuevo paradigma ha cambiado para siempre la forma en que los ciudadanos europeos nos comunicamos, transmitimos e intercambiamos información personal.

En entornos residenciales, los teléfonos inalámbricos y los puntos de acceso WiFi ofrecen un acceso flexible a la línea telefónica fija e Internet respectivamente. DECT es un protocolo de acceso inalámbrico a la red de telefonía fija usado por la práctica totalidad de los teléfonos inalámbricos existentes en el mercado, con un total estimado de cerca de mil millones de terminales vendidos y un ratio de crecimiento de cien millones anuales, según datos publicados por ETSI (*European Telecommunications Standards Institute*) [57]. Por su parte, el protocolo IEEE 802.11 de redes inalámbricas de acceso local, comúnmente conocido como WiFi, goza actualmente de un carácter ubicuo con miles de millones de dispositivos inteligentes vendidos anualmente en el mercado[1], desde teléfonos inteligentes hasta ordenadores personales.

[1]Según el informe de prensa publicado por Gartner el 13 de Febrero de 2014, el número total de teléfonos inteligentes vendidos en 2013 es estimado en unos 1000 millones, superando por primera vez al número total de ordenadores personales. `http://www.gartner.com/newsroom/id/2665715`

En este contexto, ambas tecnologías de comunicación inalámbrica se han posicionado como los estándares de elección del mercado para la intercomunicación de datos personales dentro del ecosistema de la Casa Inteligente, dando paso a una nueva generación de Sistemas de Comunicación Unificada que los integra actuando como concentradores de comunicaciones entre los diferentes dispositivos inteligentes del hogar.

Fuera de la vivienda, la sociedad vive permanentemente conectada a la red mundial de telefonía e Internet mediante el uso de teléfonos móviles conectados a las diversas redes GSM pertenecientes a los operadores móviles existentes en el mercado. Entre los diferentes estándares disponibles, GSM es aquel que posee una mayor presencia alcanzando actualmente el 95 % de la población mundial [81].

A diferencia de las tradicionales conexiones de voz y datos mediante el uso de cable, la utilización de tecnologías inalámbricas conlleva una serie de desafíos de privacidad específicos a la naturaleza inherente del medio compartido de propagación. Tradicionalmente, el procesamiento de la señal de radio y el protocolo subyacente, por parte de adversarios determinados a comprometer la seguridad y la privacidad de las comunicaciones efectuadas por terceros, requería de la utilización de considerables recursos, tanto en términos de hardware especializado como capacidad de cómputo.

Sin embargo, en tiempos recientes ha emergido una nueva generación de dispositivos de Radio Definida por Software de muy bajo coste que, en combinación con un ordenador personal estándar, posibilita el procesamiento de señales digitales en un amplio espectro de frecuencias que incluye los protocolos de comunicación inalámbrica mencionados anteriormente. Este hecho, unido a los signos de debilidad mostrados por los mecanismos de seguridad provistos por estos estándares, principalmente cifrado y autenticación, sugieren que las tecnologías inalámbricas utilizadas diariamente por cientos de millones de ciudadanos europeos pueden no ofrecer suficientes garantías respecto a la seguridad y privacidad de las comunicaciones personales frente a eventuales atacantes.

La presente tesis se centra en la investigación del grado de seguridad y privacidad efectivo ofrecido por los estándares más utilizados de comunicación inalámbrica y sus implementaciones, para comunicaciones en entornos domésticos y de uso personal. A tal efecto, se investigan las tecnologías DECT, GSM y WiFi, desde un punto de vista teórico y experimental, identificando y demostrando el alcance de vulnerabilidades en las mismas y explorando el impacto que la capacidad de cómputo de los ordenadores personales actuales, los avances en técnicas de criptoanálisis y la disponibilidad de los nuevos dispositivos de Radio Definida por Software de bajo coste presentan para la seguridad y privacidad de las comunicaciones personales efectuadas por medios inalámbricos.

1.1. Hipótesis y objetivos de la tesis

El objetivo de la presente tesis es determinar si el conjunto de las principales tecnologías de transmisión inalámbrica utilizadas en comunicaciones personales ofrece actualmente una protección adecuada de la seguridad y privacidad de las mismas.

Se parte de la hipótesis de que, en el contexto actual, las principales tecnologías de transmisión inalámbrica, DECT, GSM y WiFi, no son efectivas en la protección de la privacidad de las comunicaciones personales.

A fin de validar a esta hipótesis y evaluar el impacto real en la privacidad, se analiza y evalúa la seguridad de los protocolos de transmisión y los mecanismos de seguridad previstos en los mismos, no solo desde el punto de vista teórico sino también experimental, en contextos similares a los escenarios de despliegue reales.

Con el propósito de alcanzar el objetivo principal de la tesis se presentan los siguientes objetivos secundarios.

- Determinar la capacidad que los nuevos dispositivos de Radio Definida por Software de bajo coste poseen para posibilitar ataques efectivos

contra la seguridad y privacidad de las comunicaciones personales efectuadas sobre los protocolos inalámbricos DECT y GSM.

- Determinar la viabilidad de atacar de forma efectiva con modestos recursos los mecanismos de seguridad previstos en los estándares DECT, GSM y WiFi con el objeto de comprometer la privacidad de las comunicaciones personales que se encuentran protegidas por estos últimos.

- Demostrar experimentalmente la viabilidad de comprometer la seguridad y privacidad de las comunicaciones personales efectuadas sobre los protocolos DECT, GSM y WiFi, en escenarios de despliegue reales, interceptando comunicaciones de voz y datos.

1.2. Metodología y Materiales

A fin de alcanzar los objetivos de la tesis, descritos en la sección 1.1, se plantea la siguiente metodología.

1. Estudiar la viabilidad de monitorización remota de transmisiones efectuadas mediante las tecnologías inalámbricas objeto de la tesis, utilizando para ello dispositivos de radios definidas por software de bajo coste y ordenadores personales estándar.

2. Demostrar experimentalmente la viabilidad de interceptación remota de comunicaciones inalámbricas de terceros, desarrollando implementaciones software según proceda, en caso de ausencia de herramientas, para llevar a cabo las demostraciones experimentales.

3. Estudiar las vulnerabilidades presentes en el conjunto de mecanismos criptográficos soportados por las diferentes tecnologías inalámbricas objeto de estudio de la presente tesis.

4. Explorar la viabilidad de la aplicación práctica de ataques criptográficos, desarrollando si procede nuevos ataques más efectivos, con el objetivo

de demostrar la ineficacia de los mecanismos criptográficos actuales en la protección de la seguridad y privacidad de las comunicaciones.

5. Realizar pruebas experimentales en escenarios representativos de situaciones reales, desarrollando implementaciones software si fuera preciso, con el objetivo de demostrar la viabilidad real de atacar la seguridad y privacidad de comunicaciones de terceros en los casos en los que el cifrado se encuentre activo.

6. Identificar posibles contramedidas para mitigar el riesgo en base a los resultados obtenidos y elaborar recomendaciones para futuros protocolos.

La metodología descrita aborda los objetivos de la tesis desde una doble vertiente teórica y experimental. Siguiendo los objetivos de la tesis, la investigación cubrirá las tecnologías DECT, GSM y IEEE 802.11 también conocida como WiFi, excluyendo la nueva generación de protocolos, aún en expansión, tales como DECT ULE, UMTS y LTE.

La parte experimental vendrá comprendida por simulaciones soportadas sobre desarrollos realizados en *Matlab*, *C* y *Python*, así como pruebas reales en laboratorio, más representativas de escenarios de uso real de las tecnologías.

Para estas últimas, y en especial en lo referente a la investigación de las capacidades de los nuevos dispositivos SDR para la ejecución de ataques contra los protocolos de transmisión inalámbrica analizados en la presente tesis, se utilizan los siguientes materiales hardware.

1. Dispositivo RTL-SDR basado en Realtek RTL2832U con sintonizador Elonics E4000. El RTL-SDR es un SDR de bajo coste (con un valor que oscila entre 10 a 30 euros en el mercado) que en el caso del modelo mencionado es capaz de operar con frecuencias de muestreo de hasta 2.56 Msps en un rango de frecuencias de 52 Mhz a 2200 MHz con un rango ciego típico de 1100 MHz a 1250 MHz [123].

2. *Universal Software Radio Peripherals* versión N210. La gama USRP[2] ofrece

[2]Los detalles completos sobre la gama de productos USRP y el modelo N210 se pueden encontrar en `http://www.ettus.com/`

un conjunto de SDRs de bajo coste de gama alta, simples y flexibles que ofrecen grandes prestaciones tanto para recepción como para transmisión. El modelo N210 utilizado en la presente tesis, se encuentra basado en los siguientes componentes: una FPGA Xilinx Spartan-3A DSP 3400, ancho de banda dual de 100 MS/s, conversor analógico-digital de 14-bit, entradas externas para señales de 10 MHz y 1 PPS (SMA) y conexión opcional para GPS. El modelo N210 también viene equipado con funcionalidad MIMO, aunque dicha capacidad no haya sido utilizada para la elaboración de esta tesis.

3. Alpha AWUS036H [3] con antena 5dB. La Alpha AWUS036H es una tarjeta de red WiFi USB 2.0 que cuenta con 1 vatio de potencia de transmisión y soporta los protocolos 802.11b y 802.11g, gozando de un excelente soporte en GNU/Linux.

4. Cámara de Faraday, con los siguientes niveles de atenuación:

   - DECT: 84dB de atenuación para el rango de frecuencias de 1.9 Ghz.
   - GSM: 79dB de atenuación para la banda GSM-900 y 84dB para GSM-1800.
   - WiFi: 85dB de atenuación para la banda de 2.4Ghz y 87dB para 5 Ghz.

Los recursos software utilizados para el desarrollo de las implementaciones necesarias para llevar a cabo las pruebas experimentales, incluyen GNU Radio [19], Wireshark [36], OpenBTS [26], Asterisk [4], Matlab [5], Python 2 [6] y Gcc [7].

[3] http://www.alfa.com.tw/products_show.php?pc=34&ps=92
[4] http://www.asterisk.org/
[5] http://www.mathworks.it/products/matlab/
[6] https://www.python.org/
[7] https://gcc.gnu.org/

1.3. Contribución sobre el estado del conocimiento

La presente tesis ha aportado a lo largo de su desarrollo una serie de contribuciones relevantes al estado de conocimiento que han sido publicadas en revistas y conferencias internacionales de factor de impacto y revisadas por pares.

En esta sección se detallan los artículos más relevantes que se han publicados en el desarrollo de la tesis.

- Experimental passive eavesdropping of Digital Enhanced Cordless Telecommunication voice communications through low-cost software-defined radios. Security and Communication Networks, 2014 [137].

- Practical interception of DECT encrypted voice communications in Unified Communications environments. IEEE Joint Intelligence and Security Informatics Conference (IEEE JISIC), 2014 [34].

- Privacy leakages in Smart-Home Wireless Technologies. IEEE International Carnahan Conference on Security Technology (IEEE ICCST), 2014 [138].

- Improved cryptanalysis of the DECT Standard Cipher. Cryptographic Hardware and Embedded Systems (CHES), 2015 [35].

Adicionalmente a dichas publicaciones, parte del código fuente desarrollado durante la elaboración de la investigación de la tesis ha sido publicado en [136], de tal manera que otros investigadores puedan revisar la metodología aplicada y replicar los resultados. Actualmente el código fuente se encuentra incluido en diversas distribuciones GNU/Linux, incluyendo RedHat (Fedora)[8] y Ubuntu [9].

[8] `https://admin.fedoraproject.org/pkgdb/package/weplab/`

[9] `http://manpages.ubuntu.com/manpages/precise/man1/weplab.1.html`

1.4. Organización de la tesis

En el capítulo 1 se realiza una introducción a la presente tesis, describiendo su motivación, detallando sus objetivos y metodología, y resumiendo la contribución realizada al estado del conocimiento.

El capítulo 2 se centra en la investigación de la seguridad del protocolo DECT de comunicación de dispositivos inalámbricos de telefonía fija. En este capítulo se analiza en detalle el funcionamiento del protocolo DECT y sus características de seguridad, así como el estado del arte en lo referente a sus vulnerabilidades y ataques conocidos.

Primeramente se explora la amenaza que los nuevos dispositivos de Radio Definida por Software de bajo coste suponen para las comunicaciones inalámbricas DECT. Para ello, se desarrolla una implementación completa de un receptor DECT y se analiza su uso para la interceptación pasiva de comunicaciones DECT cuando el cifrado no se encuentra activo[10].

Posteriormente, se analiza la seguridad de las comunicaciones cifradas y se describe un nuevo ataque al intercambio de claves criptográfico de DECT que, en combinación con la implementación realizada, posibilita la interceptación pasiva de conversaciones cifradas en aquellos casos en los que el proceso de emparejamiento de los dispositivos DECT tenga lugar durante la ejecución del ataque.

Finalmente, se analiza en detalle el algoritmo de cifrado DSC previsto en el estándar DECT y se describe un nuevo criptoanálisis que demuestra experimentalmente ser cuatro veces más eficaz en el ataque a comunicaciones cifradas DECT que el mejor ataque conocido.

El capítulo 3 analiza en detalle el protocolo GSM en lo referente a su funcionamiento y características de seguridad. Tras analizar en detalle las vulnerabilidades y ataques existentes en el estado del arte, se demuestra experimentalmente la viabilidad de realizar ataques contra la privacidad de las identidades

[10]Como se describe en detalle en el capítulo 2, los resultados experimentales demuestran que gran porcentaje de los teléfonos DECT de segunda mano analizados no incorporan cifrado en sus comunicaciones

de usuarios GSM utilizando para ello Radios Definidas por Software.

A continuación, se analizan en detalle los mecanismos de seguridad utilizados en GSM para la protección de la confidencialidad de las comunicaciones y se demuestra experimentalmente un ataque activo de suplantación de identidad de una célula GSM legítima, capaz de interceptar y recuperar una llamada de voz realizada por un usuario, así como todo su tráfico de datos con Internet.

Finalmente, se analizan las vulnerabilidades del protocolo de cifrado A5/1, utilizado por la casi totalidad de los operadores GSM en Europa para el cifrado de las comunicaciones de los usuarios, y se demuestra experimentalmente la viabilidad de la recuperación de la clave de cifrado así como la interceptación pasiva de comunicaciones GSM cifradas.

El capítulo 4 es dedicado al análisis del protocolo 802.11, comúnmente conocido como WiFi. Se analiza el estado del arte y vulnerabilidades existentes, para posteriormente describir un criptoanálisis del protocolo WEP cuya implementación realizada por el autor de esta tesis, denominada como *Weplab*, demuestra la viabilidad de romper una clave WEP desde GNU/Linux en escenarios reales.

Posteriormente, en el contexto de la utilización del protocolo WiFi como nexo de unión entre los diferentes dispositivos inteligentes del ecosistema de la Casa Inteligente, se analizan las limitaciones del protocolo 802.11 ante ataques de canal lateral basados en análisis de tráfico. Se demuestra en un escenario experimental la viabilidad del análisis de tráfico para la detección de los sitios web visitados por un dispositivo de la red, aún cuando ésta se encuentre cifrada y la clave no se vea comprometida.

El capítulo 5 es dedicado al análisis del conjunto de lecciones aprendidas tras la investigación realizada en los capítulos anteriores sobre DECT, GSM y WiFi. Se sintetiza una serie de recomendaciones para el diseño de las medidas de seguridad de futuros protocolos y se realiza una serie de sugerencias para el despliegue de contramedidas en las implementaciones actuales, a fin de reducir el riesgo que las vulnerabilidades existentes suponen para la seguridad y privacidad de las comunicaciones personales de los ciudadanos.

Finalmente, en el capítulo 6, se realiza un análisis de las principales conclusiones obtenidas en la presente tesis en lo referente a la validación de la hipótesis formulada y los objetivos perseguidos, señalando futuras líneas de investigación.

Capítulo 2

DECT

En el presente capítulo de la tesis se investiga el grado efectivo de seguridad y privacidad de las comunicaciones personales que tienen lugar sobre el protocolo inalámbrico DECT.

Los objetivos perseguidos y la metodología seguida en la investigación realizada en este capítulo, serán aquellos definidos de forma general para la tesis, descritos respectivamente en las secciones 1.1 y 1.2 del capítulo 1.

Primeramente, tras analizar en detalle el estado del arte, se procede a diseñar y desarrollar una herramienta capaz de monitorizar pasivamente comunicaciones inalámbricas DECT, utilizando para ello dispositivos de Radio Definida por Software (SDR) de bajo coste. Dicha implementación cumple con un triple objetivo.

1. Analizar en detalle el protocolo DECT en lo referente a su capacidad para la protección de la seguridad y privacidad de las comunicaciones.

2. Analizar y demostrar experimentalmente la capacidad que la nueva generación de dispositivos SDR de bajo coste posee para llevar a cabo ataques a la seguridad y privacidad de las comunicaciones personales que son efectuadas sobre el estándar DECT.

3. Desarrollar una herramienta que permita la experimentación de ataques contra los protocolos de cifrado previstos en DECT, con el objetivo de

evaluar el grado de seguridad ofrecidos por éstos y la dificultad de su abuso por parte de atacantes con escasos recursos.

A continuación, se procede a analizar y evaluar el grado efectivo de protección de la privacidad de las comunicaciones para aquellas que utilicen cifrado. Para ello, se analiza primeramente la seguridad del protocolo de emparejamiento de dispositivos DECT y se evalúa experimentalmente la viabilidad de atacarlo, utilizando a tal efecto la implementación desarrollada anteriormente para la monitorización de comunicaciones DECT mediante el uso de dispositivos SDR de bajo coste.

Finalmente, se analiza en detalle la seguridad ofrecida por el protocolo de cifrado DSC y se describe un nuevo ataque criptográfico capaz de derivar la clave de cifrado utilizada para la protección de la privacidad de las comunicaciones de voz. A fin de establecer el impacto práctico que dicho ataque puede causar en la seguridad y privacidad de las comunicaciones DECT, aún cuando éstas utilicen cifrado, se evalúa experimentalmente su rendimiento en comparación con el único ataque de criptoanálisis contra DSC descrito en la literatura.

El presente capítulo realiza una importante contribución a la validación de la hipótesis de la tesis y consecución de los objetivos propuestos, en lo referente al estándar DECT. El resultado de la investigación realizada demuestra que el estándar DECT, actualmente implementado en un total estimado de 1000 millones de dispositivos [57], no ofrece garantías suficientes para la seguridad y privacidad de las comunicaciones ante atacantes casuales, dada la amplia disponibilidad de dispositivos SDR de bajo coste en el mercado.

Así mismo, se demuestra la viabilidad de atacar conversaciones cifradas, derivando la clave mediante el criptoanálisis del algoritmo de cifrado DSC o el ataque al protocolo de emparejamiento criptográfico de dispositivos DECT.

2.1. El estándar DECT

DECT, acrónimo de *Digital Enhanced Cordless Telecommunications*, es un protocolo de comunicación digital por radiofrecuencia, estandarizado por ETSI como tecnología de acceso inalámbrica para sistemas de telefonía fija. A pesar de que DECT fuera inicialmente concebido como un estándar europeo, su popularidad creció rápidamente fuera de las fronteras de la Unión Europea convirtiéndose en un estándar global.

Actualmente, DECT es el protocolo de acceso utilizado por la gran mayoría de instalaciones de telefonía fija inalámbrica a nivel mundial y la casi totalidad a nivel nacional y europeo. Se estima que su base de instalación ronda entre los 800 y 1000 millones de instalaciones a nivel global. Según ETSI [57], DECT es actualmente el estándar de ETSI más popular, tras GSM, con un crecimiento estimado en 100 millones de instalaciones nuevas cada año.

A pesar de que DECT sea principalmente utilizado para la transmisión de voz, su diseño flexible permite también su utilización para la transmisión de datos, por lo que es posible encontrar instalaciones DECT en escenarios diversos tales como terminales de pago móviles, entornos SCADA, sistemas de control de tráfico y contadores eléctricos inteligentes.

Sin embargo, DECT, en la gran mayoría de casos, es utilizado en entornos profesionales y residenciales para la transmisión inalámbrica de voz en instalaciones de telefonía fija. La interoperabilidad entre dispositivos de diferentes fabricantes viene garantizada por el perfil de interoperabilidad GAP, que forma parte integral del estándar DECT y es implementado por prácticamente la práctica totalidad de los teléfonos inalámbricos DECT disponibles en el mercado.

El estándar DECT de ETSI se extiende por más de mil páginas distribuidas en varios documentos. Para los propósitos de esta tesis se distinguen los siguientes 8 documentos principales correspondientes a la Interfaz Común (CI) de DECT.

- *CI Part 1: Overview* [48]. En este documento se realiza una introducción

al estándar DECT, describiendo su estructura técnica de forma general e introduciendo las diversas partes que conforman el documento de la interfaz común, así como los documentos que conforman las evoluciones de DECT de nueva generación (DECT-NG) y bajo consumo (DECT-ULE).

- *CI Part 2: Physical Layer* (PHL) [49]: En este documento dedicado a la capa física, se describe como opera el protocolo DECT a nivel de radiofrecuencia y se determinan los parámetros de funcionamiento, tales como las frecuencias, la modulación, la división temporal y la estructura de las tramas.

- *CI Part 3: Medium Access Control (MAC) layer* [50]: En este tercer documento describe el funcionamiento del Medio de Control de Acceso.

- *CI Part 4: Data Link Control (DLC) layer* [51]: En el documento referido a la capa de Control de Enlace de Datos de DECT, se especifican los enlaces lógicos de datos en los campos de control (*C-plane*) y usuario (*U-plane*).

- *CI Part 5: Network (NWK) layer* [52]: El documento relativo a la capa de red especifica las funciones de control de enlace, control de llamada, gestión de movilidad y funcionalidades adicionales.

- *CI Part 6: Identities and addressing* [53]: En este documento se especifican los diferentes identificativos utilizados en los equipos y las relaciones entre ellos.

- *CI Part 7: Security features* [54]: El séptimo documento de la interfaz común especifica los mecanismos de seguridad en DECT y su funcionamiento en el protocolo. De especial relevancia son los mecanismos de autenticación y cifrado, que serán analizados en detalle en secciones posteriores.

- *CI Part 8: Speech and audio coding and transmission* [55]: En este documento se establecen los requisitos para la codificación y transmisión de audio en tiempo real en comunicaciones DECT.

Dentro del estándar DECT, el perfil GAP [56] ofrece interoperabilidad entre dispositivos y modelos de varios fabricantes, permitiendo por ejemplo la utilización de un terminal móvil de un fabricante en la estación base DECT de otro.

De forma complementaria a DECT, NG-DECT es una evolución del estándar introducida en 2007, donde se extienden algunas funcionalidades en lo referente a la transmisión de datos y se introducen nuevos códecs de audio de alta resolución. El estándar NG-DECT se encuentra en continua evolución y se compone de las siguientes partes.

- *NG-DECT Part 1: Wideband speech* [58]
- *NG-DECT Part 2: Support of transparent IP packet data* [46]
- *NG-DECT Part 3: Extended wideband speech services* [59]
- *NG-DECT Part 4: Light Data Services* [47]
- *NG-DECT Part 5: Additional feature set nr. 1 for extended wideband speech services* [60]

Por otra parte, DECT-ULE (*DECT Ultra Low Energy*) es un nuevo estándar para las comunicaciones de datos en el entorno de IoT diseñado para operar con un consumo muy bajo de energía. DECT-ULE se encuentra actualmente en pleno desarrollo.

Dada la popularidad de los teléfonos inalámbricos DECT en los entornos residenciales de los ciudadanos europeos, en este capítulo se explora la seguridad y privacidad de dichas comunicaciones inalámbricas de voz. Dentro del estándar DECT, se pone especial énfasis en el análisis del perfil de interoperabilidad GAP y en las partes comunes del estándar a dicho perfil, dado que la casi totalidad de teléfonos inalámbricos existentes en el mercado lo implementan, a fin de garantizar la interoperabilidad entre dispositivos de diferentes fabricantes.

Una instalación DECT típica, como la que se puede encontrar en un entorno residencial para acceso a telefonía fija, se compone de una base DECT,

también llamada Parte Fija o FP, y uno o más teléfonos inalámbricos asociados, también llamados Parte Portátil o PP. La base DECT (FP) se encuentra típicamente conectada por cable a la red analógica de telefonía básica, también llamada RTB o POTS, mediante una conexión RJ-11. En esencia, los terminales inalámbricos DECT se comportan de forma similar a los clásicos teléfonos analógicos tradicionalmente conectados mediante cable.

2.2. El estado del arte

A pesar de la ubicuidad de DECT en las comunicaciones de telefonía inalámbrica fija, tanto en entornos profesionales como personales, no existe mucha literatura, sobre todo en comparación con otros protocolos como GSM, en lo referente a los riesgos para la privacidad y la seguridad de sus comunicaciones.

2.2.1. Escucha remota de comunicaciones no cifradas

El primer ataque de seguridad documentado contra la confidencialidad de las comunicaciones de voz bajo el protocolo DECT era presentado por el equipo de *dedected.org*[1] en [140]. Los autores demostraban cómo una tarjeta PCMCIA con un chip determinado, disponible entonces en el mercado, podía ser modificada por un atacante para realizar una escucha remota de las comunicaciones DECT cuando éstas no utilizaban cifrado.

Para ello, los autores realizaron un proceso de ingeniería inversa al hardware de la tarjeta, y al controlador software para Microsoft Windows que la acompañaba, y desarrollaron un controlador para GNU/Linux. Dicho controlador fue programado para ofrecer un control a bajo nivel de la tarjeta PCMCIA permitiendo capturar y extraer los paquetes DECT pertenecientes a las comunicaciones de otras redes. En combinación con algunas herramientas adicionales,

[1] La página `http://dedected.org` ya no se encuentra activa en el momento de escribir estas líneas. Sin embargo parte del software liberado por los autores puede aún encontrarse en repositorios de terceros tales como `https://github.com/Sitwon/dedected/tree/master/com-on-air_cs-linux`

los autores demostraron cómo era posible extraer la carga de voz posibilitando de esta manera un ataque contra la confidencialidad de las comunicaciones, cuando la red DECT no utilizaba cifrado.

El equipo de *dedected.org* analizó diversos teléfonos DECT disponibles en el mercado [102] y encontró que en numerosas ocasiones el cifrado no se encontraba activo.

Del análisis de diversas implementaciones DECT del mercado se desprende que, en numerosas ocasiones, el cifrado, si bien se encontraba activo para la transmisión de voz, no protegía la trasmisión de datos de control tales como los números de teléfono llamados y llamantes. En dichos casos un atacante podía recuperar remotamente los números de teléfono a los que una tercera persona llamaba, o desde los que era llamado.

Sin embargo, la baja disponibilidad en el mercado de la tarjeta DECT PCMCIA específica utilizada, dificultaba la implementación práctica del ataque, por lo que el riesgo sobre la privacidad de las comunicaciones de voz DECT era considerado limitado.

2.2.2. Ataques contra el algoritmo de autenticación DSAA

El algoritmo DSAA tiene una finalidad doble. Por un lado, es el encargado de ofrecer autenticación mutua entre la Parte Fija (PF), o estación base, y la Parte Portátil (PP), o terminal móvil, utilizando los procedimientos de Autenticación de FP y Autenticación de PP respectivamente. Dichos procesos de autenticación serán los encargados de prevenir ataques del tipo *Man-in-The-Middle* (MiTM) y evitar el fraude, garantizando que únicamente los terminales móviles autorizados podrán conectarse a la estación base para la recepción y ejecución de llamadas.

Por otra parte, en caso de que el cifrado se encuentre activo, el procedimiento de Autenticación de PP será el encargado de derivar una clave criptográfica de sesión aleatoria, acordada y conocida por ambas partes, que será la utilizada para cifrar la comunicación. El algoritmo DSAA también se encuentra involucrado en el mecanismo de emparejamiento criptográfico en DECT, el

cual será detallado en la sección 2.5 del presente capítulo.

En el contexto del perfil de interoperabilidad GAP, donde diferentes modelos de estaciones base y terminales de diversos fabricantes pueden interoperar juntas, los servicios de autenticación ofrecidos mediante el algoritmo DSAA son de especial relevancia. En efecto, la autenticación del terminal móvil por parte de la estación base DECT, previene que un tercero no autorizado pueda conectarse con un terminal móvil GAP y realice llamadas telefónicas a cargo del usuario de la línea.

A pesar de que el algoritmo DSAA forme parte del estándar DECT, en concreto de la parte 7 de mecanismos de seguridad [54], sus detalles e implementación se encuentran disponibles únicamente bajo acuerdos de confidencialidad. Sin embargo, en [140] y [95] los autores realizaron un procedimiento de ingeniería inversa a dispositivos hardware que implementaban dicho algoritmo y publicaron sus detalles internos junto con una implementación software de referencia. En el mismo artículo, los autores describieron el que sería el primer, y último, criptoanálisis del algoritmo DSAA y enumeraron las diferentes vulnerabilidades detectadas en el mismo.

Básicamente, DSAA es un algoritmo de cifrado en bloque que recibe 2 entradas, de 128 bits y 64 bits de longitud respectivamente, y produce una salida de 128 bits. El modo de operación del algoritmo, en sus diferentes variantes, se detalla en la sección 2.5.2, donde se describe un ataque práctico contra el mecanismo de emparejamiento criptográfico de DECT para la interceptación de comunicaciones cifradas.

La principal vulnerabilidad detectada en el diseño del algoritmo consiste en que, a pesar de la utilización de una clave secreta de 128 bits, la fortaleza del algoritmo se reduce a 2^{64}, dado que los 64-bits centrales de los 128-bits de salida dependen únicamente los 64 bits centrales de la clave.

Analizando diversas implementaciones en teléfonos DECT disponibles en el mercado, los autores descubrieron que la generación de números pseudoaleatorios (PRNG) contenía muy poca entropía, con lo que se posibilitaban cierto tipo de ataques contra los procesos de autenticación donde DSAA se

encontraba involucrado.

Otra vulnerabilidad extremadamente común en las diversas implementaciones analizadas por los autores, era la falta de autenticación mutua entre la Parte Fija (FP) y la Parte Portátil (PP). Si bien la Parte Fija autenticaba correctamente Parte Portátil, con el fin de evitar el fraude telefónico, ésta no hacía lo propio con la Parte Fija.

Esta falta de autenticación mutua entre ambas partes posibilitaba la ejecución de ataques tipo *Man-in-The-Middle*, donde un atacante es capaz de suplantar la identidad de la Parte Fija, de tal manera que el teléfono DECT conectara con dicha estación base del atacante en lugar de con la legítima. En dicho escenario, el atacante redirigiría el tráfico de voz a su destino final, de tal manera que la interceptación fuese transparente para la víctima. Sin embargo, dicho ataque solo posibilitaría la interceptación de comunicaciones de voz iniciadas por la víctima y no las recibidas por ésta, como sería el caso de una llamada entrante.

Dicho ataque fue implementado en [95] mediante la modificación del controlador binario de una tarjeta PCMCIA DECT suministrado por el fabricante para Microsoft Windows, de tal forma que el RFPI de la tarjeta fuese modificado de forma arbitraria para suplantar la identidad de la estación base legítima. De dicha manera, en un momento dado habría 2 estaciones con el mismo RFPI de forma simultánea. Mediante una interferencia selectiva al canal físico o lógico utilizado por la estación legítima, se conseguiría que el teléfono se sincronizara con la estación base del atacante.

2.2.3. Ataques contra el algoritmo de cifrado DSC

El algoritmo de cifrado DSC, al igual que en el caso del DSAA, no se encuentra disponible al público como parte del estándar. En [117] los autores publicaron sus detalles por primera vez tras derivarlo de una implementación hardware mediante un procedimiento de ingeniería inversa. En dicho trabajo, los autores publican un criptoanálisis del DSC inspirado en los ataques de correlación que habían sido aplicados de forma efectiva contra el algoritmo de

cifrado A5/1 [99].

Posteriormente, en [100], los autores presentan un nuevo ataque activo capaz de recuperar los *keystreams* utilizados en una conversación cifrada previa y demuestran la viabilidad de descifrarla mediante el uso de esta técnica.

En la sección 2.6, se presentan los detalles completos del algoritmo DSC, así como el funcionamiento del ataque NTW. Sus resultados se compararan con un criptoanálisis mejorado que es presentado en la sección 2.6.

2.3. *DECT-Eye*, una plataforma para el análisis de seguridad del protocolo DECT

En esta sección se describe la investigación realizada referente a las capacidades que la nueva generación de dispositivos SDR de bajo coste posee, para llevar a cabo monitorización remota de comunicaciones DECT.

A ese respecto, se describe y detalla la implementación software realizada para la monitorización de comunicaciones DECT, bautizada como *DECT-Eye*, así como una serie de experimentos de laboratorio orientados a demostrar la efectividad de dicha implementación.

2.3.1. Análisis del protocolo de comunicación DECT

El rango de frecuencias utilizado por DECT varía en función del área geográfica, siendo de 1800 Mhz - 1900 Mhz para Europa, 920-1930 Mhz para Estados Unidos y Canadá, 1910-1920 Mhz para Brasil, 1893-1906 Mhz para Japón y 1910-1930 Mhz para América Latina. En lo referente a la potencia máxima de transmisión permitida, ésta se sitúa en 250 mW para Europa y 100 mW para Estados Unidos.

El estándar DECT de ETSI se divide en 5 capas principales. La capa física [48] se encarga de la transmisión de las tramas individuales a nivel físico tratando modulaciones y frecuencias. La capa de control de acceso al medio (MAC) [49] establece canales lógicos de comunicación entre dispositivos que

operan sobre el medio físico. La capa de control de enlace de datos [50] ofrece servicios tanto orientados como no orientados a la conexión para los mensajes de control (plano C) y carga de datos de voz (plano U). La capa de red [51] se encarga del control de llamadas y los servicios de gestión. Finalmente, la codificación de audio [55] determina como el audio se codifica y decodifica para su transmisión digital.

El protocolo DECT se basa en Acceso Múltiple por División de Frecuencia (FDMA), Acceso Múltiple por División de Tiempo (TDMA) y Transmisión en dos sentidos por División de Tiempo (TDD). El rango de frecuencias asignado a DECT por el estándar se divide en varios canales, donde cada uno ellos posee un ancho de banda de 1728 Mhz. El número total de canales varía en función de la localización geográfica, siendo un total de 10 para Europa, comenzando en los 1880 Mhz, y 5 para Estados Unidos, comenzando en los 1920 Mhz. Cada uno de los canales físicos se divide en varios canales lógicos bidireccionales mediante el uso de TDD/TDMA, a fin de permitir la transmisión simultánea de varios dispositivos mediante la división en ranuras temporales. El funcionamiento de FDMA/TDMA/TDD en DECT se esquematiza en la figura 2.1.

Cada 10 milisegundos se transmite una trama DECT en un canal determinado. Dicha trama se divide en 24 ranuras de tiempo con una duración individual de 0.41 ms cada una. Las primeras 12 ranuras son reservadas para transmisiones originadas por la Parte Fija o estación base DECT ($FP \rightarrow PP$) y las otras 12 ranuras para las transmisiones de las Partes Portátiles o terminales inalámbricos móviles ($PP \rightarrow FP$). Cada ranura de tiempo contiene un total de 420 bits.

En el estándar DECT existen cuatro tipo de paquetes físicos. El paquete físico corto (P00) tiene una longitud de 96 bits y se utiliza principalmente para transmisiones cortas de datos no orientadas a la conexión. El paquete físico básico (P32), también llamado de ranura completa, es el tipo más común con una longitud de 424 bits. El paquete físico de baja capacidad (P08j) tiene un tamaño de 184 bits y ocupa la mitad de una ranura de tiempo. Finalmente, el paquete físico de alta capacidad (P80) contiene 904 bits y ocupa una doble

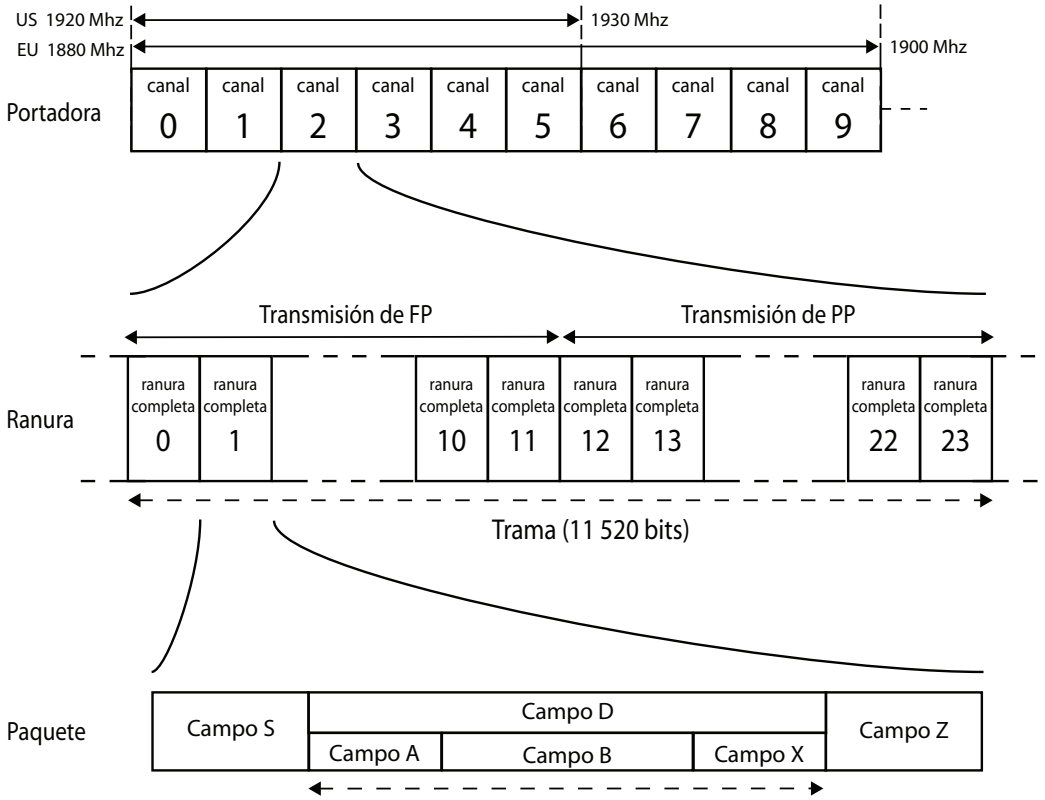


Figura 2.1: Estructura FDMA/TDMA de DECT

ranura de tiempo.

Típicamente cada ranura de tiempo contiene un único paquete físico básico (P32), a pesar de que en ciertos casos pueda contener un paquete físico corto (P00), dos paquetes físicos de baja capacidad (P08j) o medio paquete físico de alta capacidad (P80). En la mayoría de las implementaciones de telefonía inalámbrica solo se utilizan paquetes físicos de ranura completa (P32).

Las tramas DECT compuestas por 12 $FP \rightarrow PP$ ranuras temporales junto con otras 12 $PP \rightarrow FP$, se agrupan en una multitrama de un tamaño de 16 tramas y una duración total de 160 ms (10 $ms/trama \times 16\ tramas/multitrama$), tal y como se describe en la figura 2.2. La primera trama de la multitrama, enviada por la Parte Fija (FP), contiene habitualmente información de sincronización. A su vez, las multitramas se agrupan formando hipertramas de 25 multitramas cada una.

La utilización de Acceso Múltiple por División de Tiempo (TDMA) junto con Transmisión en dos sentidos por División de Tiempo (TDD) crea 12

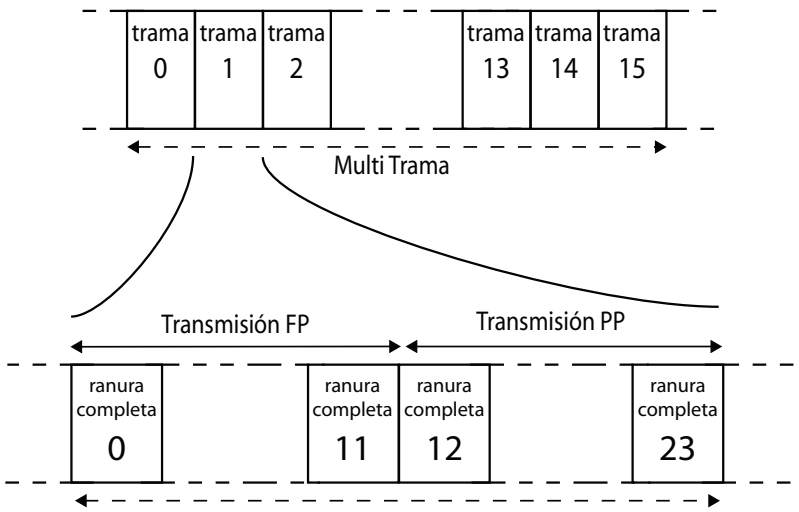


Figura 2.2: Estructura de tramas y multitramas en DECT

sub-canales lógicos con ranuras de tiempo independientes para los canales de bajada ($FP \rightarrow PP$) y subida ($PP \rightarrow FP$), tal y como se describe en la figura 2.3.

La identidad de una estación DECT se determina por su Identificador de Parte Fija de Radio (RFPI) que actúa como un identificador único para dicha instalación DECT. El RFPI se compone de 40 bits divididos en 3 partes principales: el Código del Fabricante del Equipo (EMC), reservado por ETSI o un proveedor autorizado para el fabricante del dispositivo, el Número de la Parte Fija (FPN) reservado por el fabricante como identificador único para cada EMC, y el Número de Parte fija de Radio (RPN) reservado por el fabricante o instalador con el objetivo de permitir escenarios de despliegue con varias Partes Fijas (FP).

Por otra parte, la Parte Portátil (PP) es identificada por la Identificación Internacional de Usuario Portátil (IPUI), que al igual que el RFPI para la Parte Fija (FP), se comporta como un identificador único de dispositivo para la Parte Portátil (PP). El IPUI también se compone de 40-bits y contiene un EMC junto con un Número de Serie de equipo Portátil (PSN).

En lo referente a su función, tanto el RFPI como el IPUI muestran similitudes con una dirección MAC *Ethernet*, la cual también identifica de forma única la tarjeta de red a la que se encuentra asignada. De hecho, el campo EMC per-

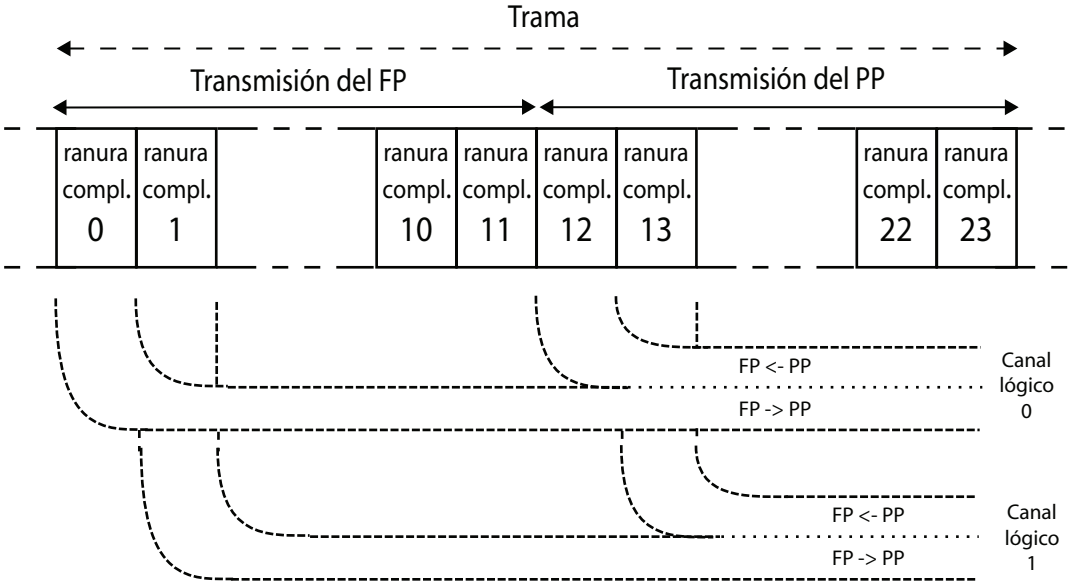


Figura 2.3: Estructura de canales y sub-canales lógicos en DECT

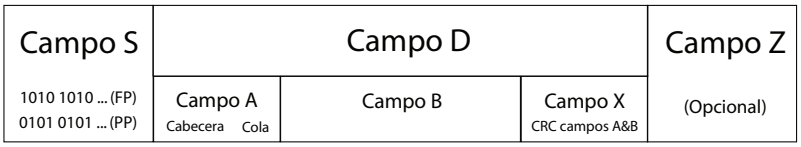


Figura 2.4: Estructura del paquete DECT

mite realizar una identificación remota sencilla del fabricante de una Parte Fija o Parte Portátil determinada, mediante la observación de los paquetes DECT emitidos por dicho dispositivo. Esta capacidad para determinar el fabricante de forma remota puede ser de gran valía para un adversario, ya que ciertos tipos de ataques de seguridad son específicos a determinados equipos de ciertos fabricantes. De la misma manera, existen ciertas implicaciones en la privacidad del usuario que serán objeto de análisis en secciones posteriores.

A nivel de la capa física todos los bits de los paquetes DECT son modulados mediante GFSK siguiendo la configuración *1a* dictada en el estándar ETSI, la cual es la utilizada en las instalaciones residenciales de DECT para acceso a telefonía fija. El estándar DECT también acomoda otras modulaciones basadas en DBPSK y QAM para otros usos.

El paquete DECT, cuya estructura se describe en la figura 2.4, se compone de 3 campos principales, el campo S, el campo D y el campo Z.

El campo S tiene una longitud de 32 bits y su objetivo es ayudar al receptor a sincronizar con la señal. Los primeros 16 bits son un preámbulo y los últimos 16 bits representan el tipo de paquete DECT.

El campo D es el contenedor principal de la información de control y datos. En el caso del paquete de ranura completo, dicho campo tiene una longitud de 388 bits, dividida en los sub-campos A, B y X. El campo A tiene una longitud de 64 bits y contiene información de control tal como el RFPI o el IPUI. Se divide en una cabecera de 8 bits y una cola que varía en base a la modulación utilizada. Los 16 bits restantes se utilizan para un control de redundancia cíclica (CRC), protegiendo de errores al contenido del campo A.

El campo B tiene una longitud de 320 bits y, en el caso de comunicación digital de telefonía inalámbrica, contiene la voz digitalizada y codificada típicamente con el códec G726. Finalmente, el campo X representa los 16-bits de CRC del campo A junto con los 4 bits del CRC del campo B. El campo Z restante es opcional y tendría como objetivo ser utilizado por el receptor para la detección de interferencia no sincronizada.

El estándar DECT permite autenticación y cifrado mediante el uso de los algoritmos DSAA y DSC. Dichos algoritmos fueron mantenidos inicialmente en secreto encontrándose disponible únicamente para los fabricantes bajo un estricto acuerdo de confidencialidad. Sin embargo, en [95] y [117] los respectivos autores derivaron ambos algoritmos por procedimientos de ingeniería inversa partiendo de implementaciones existentes en software y hardware. Actualmente, tanto DSAA como DSC se consideran de dominio público dadas dichas publicaciones. Dichos algoritmos son analizados en detalle en la sección 2.6.

2.3.2. Plataforma de monitorización DECT basada en SDR

En Europa, el protocolo DECT opera en el rango de frecuencias 1880 Mhz a 1930 Mhz donde se utilizan un total de 10 canales con 1.728 Mhz de ancho de banda cada uno. Los datos son transmitidos a un ratio de 1.152 Mbps. Por lo tanto, la velocidad de muestreo requerida para mantener intacto el contenido de información de la señal, deberá ser al menos de 2.304 Msps (el doble del

ratio de transmisión de la información modulada).

La implementación que se se ha desarrollado en este capítulo se encuentra preparada para funcionar con cualquier Radio Definida por Software (SDR) capaz de operar en el rango de frecuencias 1880 Mhz - 1930 Mhz y tomar muestras a una velocidad de al menos 2.304 Msps. Actualmente, prácticamente cualquier dispositivo SDR, incluyendo aquellos que bajo coste, cumple con dichas especificaciones. Para el desarrollo de la plataforma y su posterior utilización en esta investigación, se han utilizado dos dispositivos representativos de ambos extremos del mercado de SDR de bajo coste, el USRP n210 y el RTL2832U con sintonizador Elonics 4000. La especificación completa de ambos dispositivos puede consultarse en la sección 1.2 del capítulo 1.

Tal y como se describe en la figura 2.6, la implementación de la plataforma de monitorización de comunicaciones DECT se divide en 2 partes principales, un diagrama de flujo de GNU Radio dirigido por Python y un motor DECT programado en Python. El flujo de GNU Radio se encarga de controlar el SDR y realizar la demodulación de la señal, agrupando y enviando los bits decodificados al motor DECT.

A pesar de que técnicamente sea posible la escucha simultánea de los 10 canales utilizados por DECT, la implementación de dicha aproximación requeriría del uso de un SDR con un ancho de banda, de al menos 24 Msps, así como una gran cantidad de recursos para el proceso de la señal de cada canal. Por lo tanto, la implementación realizada ha sido diseñada para escuchar un único canal cada vez, con el objetivo de permitir el uso de dispositivos SDR con un ancho de banda reducido, como el RTL-SDR, y su utilización en sistemas con modestos recursos de procesador.

El motor DECT realiza la correlación y des-encapsulado de los paquetes DECT, junto con la gestión del resto de las capas hasta llegar a la decodificación de la voz digital transmitida. El flujo demodulado de bits es recibido desde el flujo GNU Radio mediante una conexión TCP. La arquitectura propuesta permite la distribución de ambos componentes en sistemas separados, y permite acoplar otros frontales DSP alternativos.

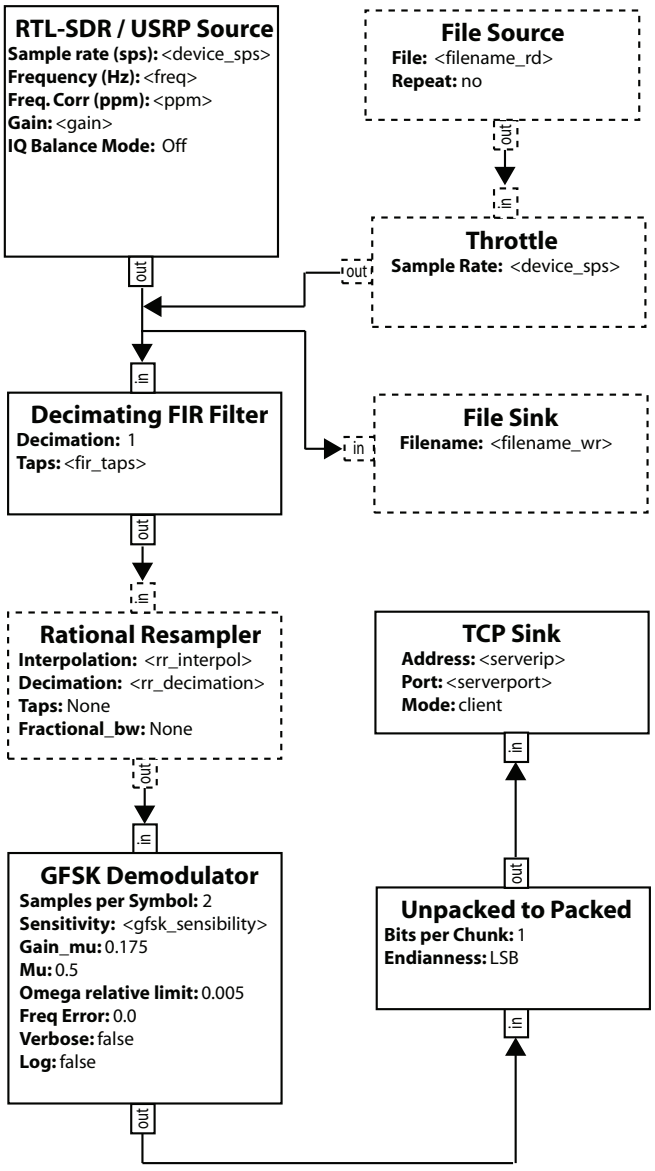


Figura 2.5: Diagrama de flujo GNU Radio

Parámetro	RTL-SDR	USRP N210	Descripción
device_sps	2,304*e*6	$\frac{100e6}{10}$	Frecuencia de muestreo del SDR
rr_interpol	n/a	144	Ratio de interpolación de la frecuencia de muestreo
rr_decimation	n/a	625	Ratio de decimado de la frecuencia de muestreo

Tabla 2.1: Parámetros del diagrama de flujo GNU Radio dependientes del SDR

Parámetro	Valor	Descripción
freq	$freq\_ch\_0 - (channel \times ch\_freq\_jmp)$	Cálculo frecuencia de un canal DECT determinado
fir_taps	low_pass (fir_gain, device_sps, dect_occ_bw, dect_ch_bw, passband_ripple, stopband_att)	Cálculo taps para el filtro FIR

Tabla 2.2: Cálculo de los parámetros del flujo GNU Radio independientes

El flujo GNU Radio completo junto con todos sus componentes y parámetros se presenta en la figura 2.5. Las líneas punteadas representan módulos que no son utilizados bajo todos los modos de operación de la implementación.

La frecuencia de muestreo de entrada el bloque de demodulación de la señal deberá de ser 2.304 Msps. El RTL2832U E4000 soporta cualquier velocidad de muestreo dentro de sus especificaciones, sin embargo para la familia de USRP se debe utilizar un decimado sobre la frecuencia de muestreo nativa.

En la implementación realizada se utiliza un bloque GNU Radio llamado *Rational Replampler*, con el objetivo de normalizar la frecuencia muestreo de entrada al demodulador en 2.304 Msps. Dicho módulo, mostrado en la figura 2.5, solo es utilizado para los SDR de la familia USRP. Los parámetros requeridos por el módulo de remuestreado dependen, por tanto, de la velocidad de muestreo nativa del dispositivo SDR específico que este siendo utilizado con la implementación. En la tabla 2.1 se presentan los valores respectivos para el

Nombre parámetro	Valor	Descripción
freq_ch_0 (EU)	1897.344e6	Frecuencia DECT canal 0 en Europa
ch_freq_jmp	1.728e6	Ancho de banda del canal DECT
fir_gain	2.0	Ganancia filtro FIR
stopband_att	60	Atenuación Stopband del filtro FIR
passband_ripple	1.0	Propagación pasobanda filtro FIR
dect_ch_bw	$\frac{ch\_freq\_jmp}{2}$	Ancho de banda del canal DECT
gap_bitrate	$\frac{2{,}304e6}{2}$	Ratio de transmisión de los bits en el perfil GAP
dect_occ_bw	$\frac{gap\_bitrate \times 1{,}03}{2}$	Ocupación de ancho de banda

Tabla 2.3: Parámetros relevantes utilizados en el receptor DECT basado en SDR

RTL2832U E4000 y el USRP N210, que serán posteriormente utilizados para la realización del resto de experimentos. En el caso del USRP, el parámetro *device_sps* se calcula como un ratio de la frecuencia de muestreo nativa (10^6) y el factor de división de 10.

El filtro de canal utilizado en la implementación es del tipo FIR (*Finite Impulse Response*) con 20 *taps* y un factor de decimado de 1. Los parámetros utilizados para el cálculo de los *taps* se resumen en las tablas 2.1 y 2.2. El cálculo de la frecuencia adecuada para la recepción de una canal DECT determinado se calcula con la fórmula $freq\_ch_0 - (channel \times ch\_freq\_jmp)$.

Por ejemplo, para el cálculo de la frecuencia correspondiente al canal 6 en Europa, se utiliza la siguiente fórmula: $1897{,}344 - (6 \times 1{,}728) = 1886{,}976 Mhz$

El demodulador GFSK estándar utilizado en la implementación es un detector diferencial de 1 bit, que utiliza demodulación F_m no coherente para extraer la señal de banda base. En teoría, tanto la demodulación coherente como no coherente serían posibles, pero esta última es la preferida en el caso de DECT, ya que, tal y somo se describe en [130], se podría observar una desviación de hasta 30 % respecto a la frecuencia instantánea. El demodulador estima la señal de reloj utilizada para remuestrear la señal demodulada. Se

utilizan decisiones absolutas con límites fijados en los instantes de las muestras para recuperar los bits modulados. La entrada al demodulador es la señal compleja en banda base, mientra que la salida es un flujo de bits agrupados en 1 bit de información por byte (el menos significativo). Posteriormente, el bloque *unpacked_to_packed* de la figura 2.5 reagrupa los bits menos significativos y los empaquetan 8 bits de información por byte.

Tal y como se describe en [94], el pulso de respuesta puede ser afectado por todos los símbolos pasados y futuros, aunque en la práctica solo se consideran típicamente los 2 vecinos más cercanos, dado que solo se utilizan 3 bits consecutivos cada vez. En la implementación del demodulador GFSK de GNU Radio, es posible controlar el impacto de los vecinos más próximos mediante un parámetro denominado *sensibilidad*. Este parámetro es equivalente al ancho de banda normalizado de 3 dB en [94], donde se señala como un parámetro común en el diseño de DECT. En la sección 2.3.3 se realizan experimentos con varios valores para dicho parámetro y se analizan los rendimientos obtenidos. El diagrama de flujo de la figura 2.5 junto con las tablas 2.1, 2.2 y 2.3 contienen toda la información relevante sobre la implementación de la parte de GNU Radio.

Una vez los bits son demodulados y empaquetados, éstos son enviados al motor DECT mediante una conexión TCP, tal y como se muestra en la figura 2.6.

El motor DECT Python contiene un hilo de ejecución que recibe continuamente los bits desde la red y los almacena en un *buffer* de memoria. Otro hilo de ejecución será el encargado de procesar dicho *buffer* de memoria para encontrar los paquetes DECT capturados.

El módulo correlacionador localiza los paquetes DECT identificando, dentro del flujo de bits demodulados, el preámbulo del campo S, en base a una tolerancia configurable mediante el límite definido en la distancia de hamming aceptada. Dicho valor de distancia de hamming podría dar lugar a una carga excesiva del extractor de paquetes y pérdida de tramas debido a una sensibilidad excesiva a interferencia entre canales y estaciones DECT. A pesar de que

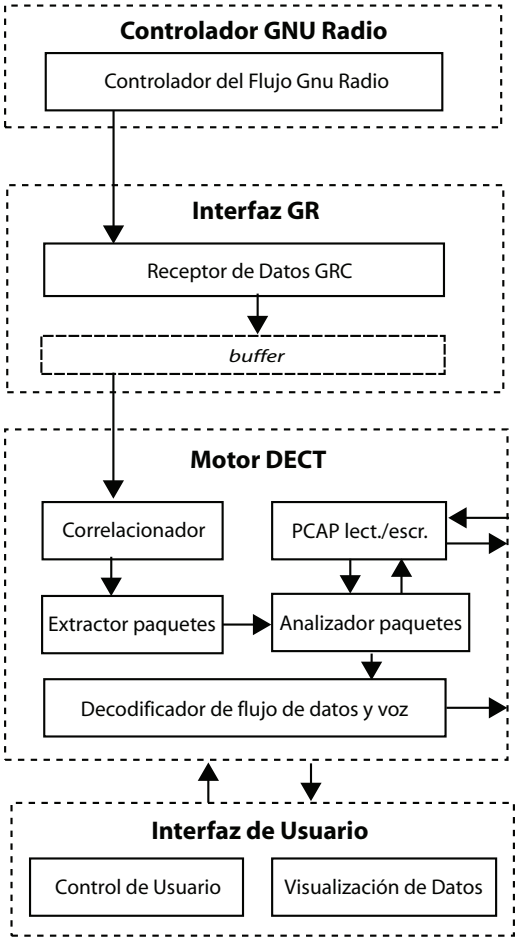


Figura 2.6: Arquitectura plataforma de monitorización DECT

esto pueda ser beneficioso para monitorizar una mayor espectro de estaciones, es perjudicial en el caso de querer centrarse en la monitorización de un sistema DECT específico. En la sección 2.3.3 se analizan experimentalmente los efectos de la variación del parámetro de distancia de hamming.

En el momento en que el correlacionador encuentre un preámbulo correcto dentro de la distancia de hamming definida para un paquete, $FP \leftarrow PP$ o $PP \leftarrow FP$, se verifican los últimos 16 bits del campo S para obtener el tipo de paquete DECT capturado. Si se encuentra un tipo conocido, el extractor de paquetes identificará la longitud y lo extraerá. Una vez extraído se verifican los errores CRC en los campos A y B, si lo hubiera, y en caso de que todo este correcto se copia a un *buffer* de memoria local.

Una vez en el *buffer* de memoria, se realiza un análisis completo del paquete identificando su tipo (RFP o PP), el RFPI asociado, presencia de campo

B, así como el resto de campos y banderas relevantes. Con el fin de seguir la transmisión TDD/TDMA, se han realizado dos implementaciones diferentes.

En una primera implementación se ha utilizado el orden de los paquetes DECT recibidos y su posición relativa respecto al resto de los paquetes. El paquete de inicio de una multitrama DECT es detectado utilizando los paquetes de sincronización enviados periódicamente por las estaciones DECT. La posición del paquete actual respecto al inicio de la multitrama determinará su número de trama, ranura y asociación RFPI.

A pesar de que, como se demuestra en la sección 2.3.3, esta aproximación funciona bajo condiciones ideales, posee una gran desventaja en lo referente a su sensibilidad para tramas perdidas. Además de un posible fallo en la asociación RFPI para paquetes PP, si una de las 16 tramas de la multitrama se pierde, tanto para FP como PP, el número de trama para el resto de las tramas de dicha mutitrama será erróneo. Un número de trama erróneo asignado a un paquete DECT, derivará en una decodificación errónea de la carga de voz digital que contenga.

En efecto, siguiendo el estándar DECT, el campo B se encuentra codificado con un *scrambler* [50], el cual genera una secuencia que se añade mediante la operación lógica XOR bit a bit al contenido a codificar. Este mecanismo pretende evitar secuencias largas de ceros o unos que pueda confundir al demodulador del lado del receptor. Dicha secuencia generada por el *scrambler* es dependiente del número de trama, por lo que un paquete DECT asignado a un número de trama incorrecto será decodificado con una secuencia errónea del *scrambler*, lo cual arruinará el contenido de la trama que quedará inservible.

Debido a esta gran limitación de la implementación de TDD/TDMA basada en el orden de los paquetes, se realiza una segunda implementación basada en el tiempo de recepción de cada paquete. El tiempo de recepción se deriva del desplazamiento del primer bit demodulado del preámbulo del paquete con respecto al inicio del flujo de paquetes demodulados.

El ratio de transmisión de bits de GAP en DECT se calcula como $\frac{2{,}304e6}{2}$. Cada intervalo de 10ms se puede calcular como $gap\_bitrate \times (10/1000)$.

Este cálculo permite la determinación exacta del número de trama y su asociación RFPI independientemente del número de tramas perdidas en la multitrama. En la sección 2.4.2 se presentan los resultados experimentales de la capacidad de ambas implementaciones TDD/TDMA en lo referente a su capacidad de interceptar pasivamente conversaciones DECT.

Tanto el flujo de GNU Radio como el motor DECT, son gestionados por el mismo programa Python que además se encarga de la interacción con el usuario mediante una interfaz de texto y permite la configuración de una amplia variedad de parámetros de funcionamiento mediante parámetros aceptados por línea de comando.

Con el objetivo de potenciar la plataforma creada para su utilización como herramienta de monitorización de sistemas DECT, necesaria para llevar a cabo el resto de la investigación realizada en este capítulo, se le dota con la capacidad de guardar y recuperar lecturas de radiofrecuencia en formato IQ, así como guardar y recuperar colecciones de paquetes en formato PCAP. Esta última característica permite la interacción con *Wireshark* [36] para el análisis de los paquetes DECT capturados con el objetivo de permitir la investigación realizada sobre DECT, objeto del presente capítulo.

2.3.3. Resultados

La figura 2.7 muestra las ráfagas generadas por una estación DECT (FP) sin ninguna conversación de voz activa, tal y como son capturadas remotamente por la plataforma de monitorización desarrollada en 2.3.2. Efectivamente, una FP DECT transmitirá 100 tramas DECT por segundo independientemente de que exista un uso por parte del usuario o no. En la figura 2.7 se puede observar que el intervalo entre ráfagas es de 10 milisegundos. Esto indica claramente que en el canal DECT monitorizado existe una única estación DECT donde las únicas transmisiones existentes provienen de la estación base DECT (FP), sin que exista ninguna comunicación proveniente de una Parte Portátil (PP).

La figura 2.8a muestra una vista ampliada de una trama DECT enviada por la FP. La figura 2.8b muestra en detalle el arranque de la transmisión DECT,

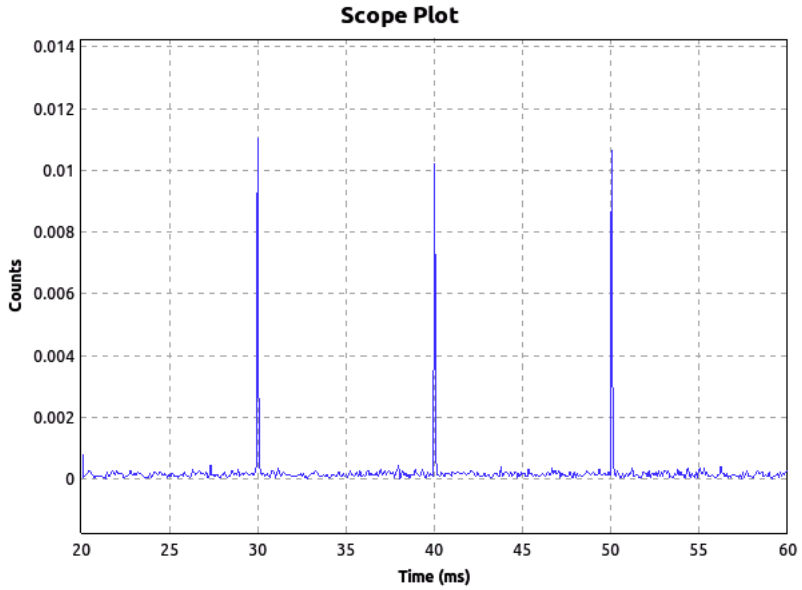


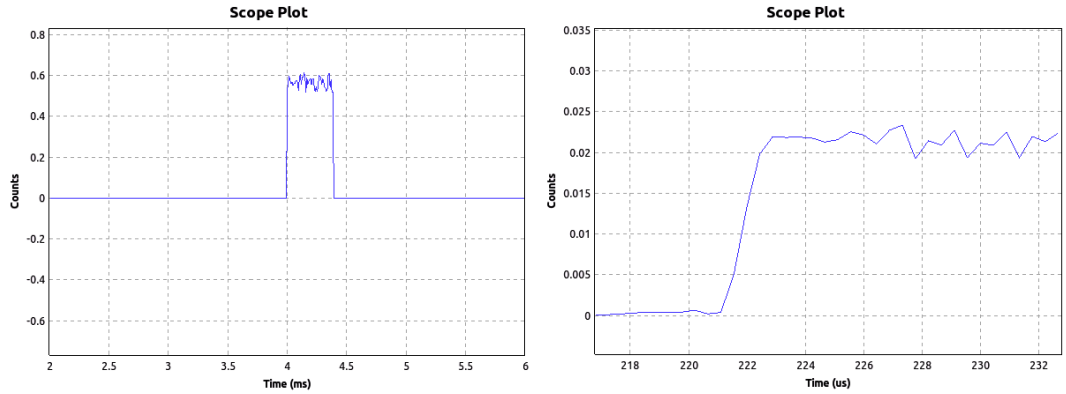
Figura 2.7: Ráfagas DECT

una característica que en numerosas ocasiones es representativa de los componentes hardware con los que se ha implementado la radio del transmisor.

Tal y como se apuntada en la sección 2.3.1, el parámetro de sensibilidad del demodulador GFSK juega un papel muy importante que redundará en un mayor o menor rendimiento del sistema. Con el objetivo de maximizar la capacidad de la plataforma de monitorización DECT desarrollada para capturar la voz digital transmitida en el escenario de una interceptación pasiva de comunicaciones DECT, se ha evaluado el rendimiento de la implementación para diferentes valores de sensibilidad GFSK, tanto para el RTL-SDR como para el USRP N210. La figura 2.9a muestra una gráfica de la evolución de los parámetros de calidad de recepción de la señal para diferentes valores de sensibilidad del demodulador GFSK en la plataforma USRP N210. La figura 2.9b muestra también dichos resultados para la plataforma RTL-SDR.

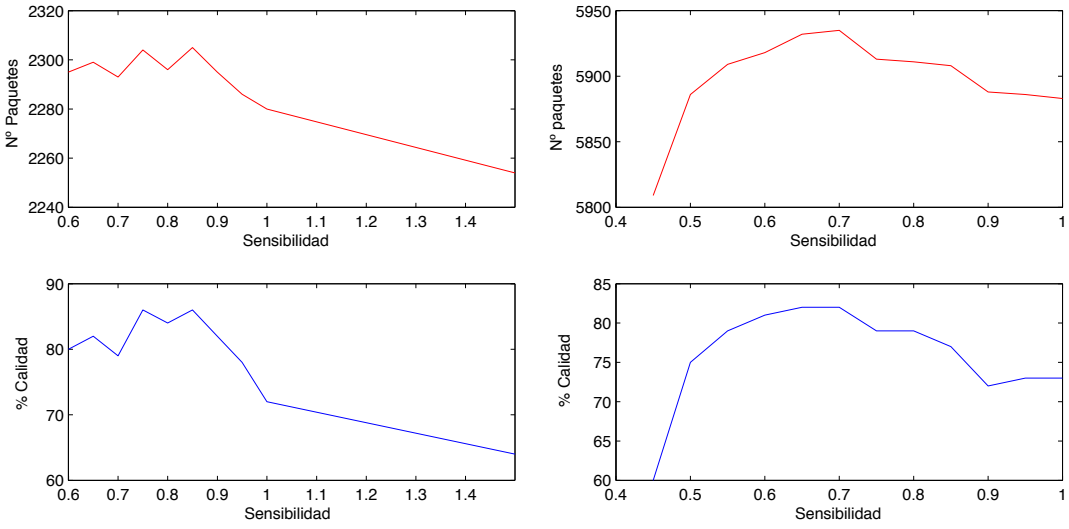
Valores muy alejados al óptimo resultan en un dramático decremento de la calidad de recepción dando lugar a un gran porcentaje de pérdida de paquetes debido a la Interferencia Entre Símbolos y errores de CRC.

En términos de errores de CRC, la figura 2.10a muestra la evolución de éstos para la plataforma USRP N210 con respecto a diferentes valores de dis-



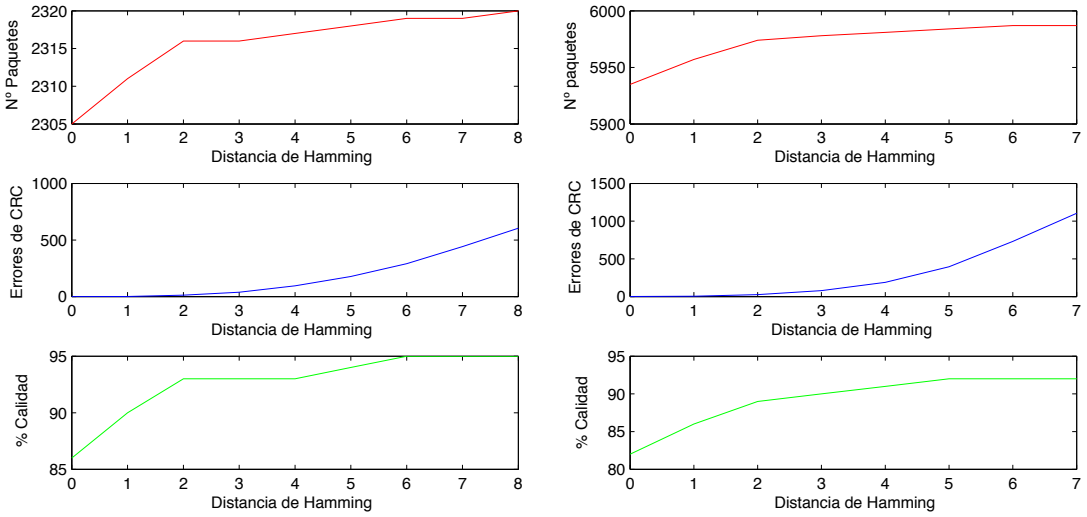
(a) Ráfaga DECT capturada (b) Rampa inicial de una ráfaga DECT

Figura 2.8: Ráfagas DECT capturadas por la plataforma de monitorización DECT desarrollada



(a) USRP N210 (b) RTL-SDR

Figura 2.9: Parámetros de calidad de recepción DECT en base a la sensibilidad del demodulador GFSK



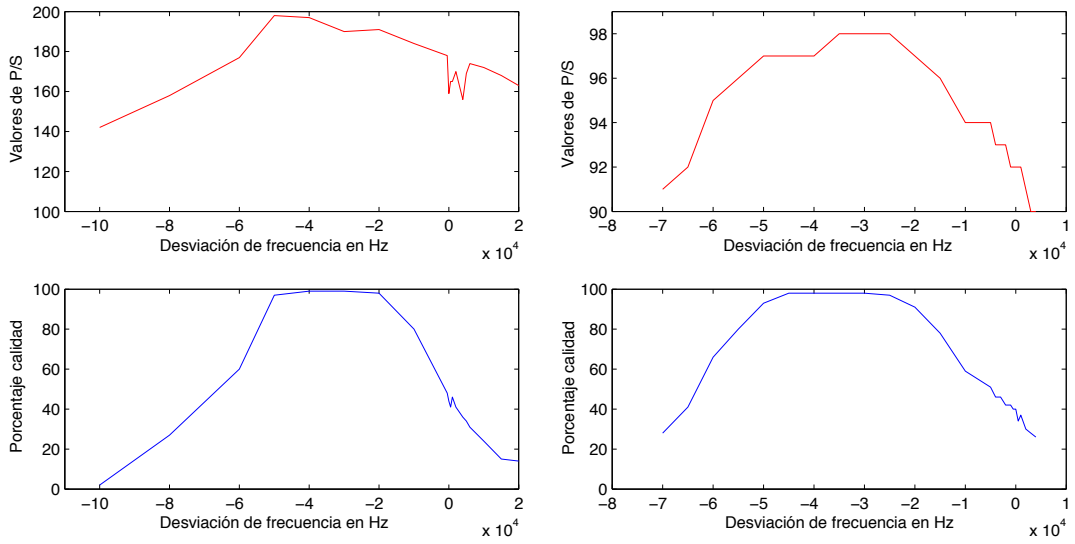
(a) USRP N210 (b) RTL-SDR

Figura 2.10: Número de paquetes con CRC erróneo en base a diferentes valores de distancia de Hamming

tancia de Hamming. Como es de esperar, a mayor distancia de hamming, mayor el porcentaje de errores de CRC detectados en las pruebas experimentales. La figura 2.10b muestra el mismo resultado experimental para la plataforma RTL-SDR.

En el análisis de los resultados se puede observar que existe un valor óptimo de sensibilidad del demodulador GFSK y la distancia de Hamming, situado en 0.85 y 2 respectivamente. Por lo tanto, ambos valores serán los utilizados en el experimento sobre interceptación pasiva de tráfico DECT basado en SDR cuyos resultados se presentan en la sección 2.4.2.

Otro aspecto muy importante que debía ser verificado experimentalmente es la desviación de frecuencia debida a oscilaciones de temperatura del reloj interno de los SDR utilizados. La figuras 2.11a y 2.11b muestran la curvatura de evolución de la calidad de recepción de una señal DECT respecto a diferentes valores de corrección de frecuencia expresados en hercios, para las plataformas USRP N210 y RTL-SDR respectivamente. Ambas lecturas han sido tomadas con la sensibilidad del demodulador GFSK fijada a 0.85 y la distancia de Hamming a 0.



(a) USPR N210 (b) RTL-SDR

Figura 2.11: Parámetros de calidad de recepción de paquetes DECT en relación al desplazamiento de frecuencia

Mediante la funcionalidad de exploración del entorno provista por la plataforma de monitorización DECT desarrollada, se explora la capacidad que un adversario tendría para reconocimiento del entorno analizando todo el espectro de canales DECT reservados en Europa buscando transmisiones de estaciones DECT (FP). Las transmisiones periódicas de tramas DECT por parte de una FP revelan inmediatamente la existencia de estaciones vecinas.

Como se ha mencionado anteriormente, en ausencia de comunicaciones de voz cada estación DECT transmite con una periodicidad de 100 tramas por segundo (una trama FP cada 10 ms), mientras que en presencia de una conversación de voz en curso la cantidad de tramas por segundo se situará en 200 (100 transmitidas por la FP y otras 100 por la PP). Por lo tanto, la cantidad de paquetes DECT por segundo asociado a un RFPI determinado resultado un indicador fiable de la calidad de recepción de dicha estación DECT concreta.

En la implementación de TDMA, basada en el orden de los paquetes, la extracción correcta de los datos de voz depende de que no haya pérdida de tramas en las multitramas, que puedan dar lugar a asignaciones erróneas en

los números de tramas de los paquetes. Por lo tanto, el porcentaje de multitramas sin pérdida de tramas (es decir, con 16 tramas correctamente recibidas) resulta un indicador secundario útil para la estimación de la calidad de recepción.

2.4. Interceptación de comunicaciones de voz DECT

Con el objetivo de demostrar la capacidad de los dispositivos SDR de bajo coste para llevar a cabo ataques contra comunicaciones personales sobre protocolo DECT, en esta sección se realizan una serie de experimentos, donde se recrean en laboratorio escenarios reales de utilización por parte de usuarios y se explora la capacidad que la implementación previamente desarrollada posee para la interceptación pasiva de comunicaciones.

En los experimentos se utilizan, a modo comparativo, el USRP N210 y el RTL-SDR con sintonizador Elonics E4000, cuyas especificaciones se detallan en la sección 1.2 del capítulo 1.

El propósito de los experimentos será la demostración de la interceptación de comunicaciones no cifradas. La interceptación de comunicaciones cifradas será objeto de secciones posteriores.

2.4.1. Reconocimiento de identidades de usuario DECT

DECT es un protocolo que opera mediante radio frecuencia sobre un medio compartido y, como tal, sus comunicaciones son susceptibles de ser monitorizadas e interceptadas por terceros.

Cada FP transmite de forma constante 100 tramas DECT por segundo independientemente de la presencia o ausencia de llamadas en curso. Todas las tramas recibidas son analizadas y su integridad es verificada por los CRCs existentes para los campos A y B si lo hubiera.

El RFPI contenido en la trama permite obtener la identificación del FP asociado, el cual, tal y como se ha detallado en la sección 2.3.1, es único para dicha estación base y permite determinar el fabricante del dispositivo. El identifica-

dor RFPI se encuentra dentro del campo A y nunca va cifrado, por lo que cualquier tercero será capaz de obtenerlo si se encuentra en el rango de recepción de la estación DECT.

Al contrario que GSM, DECT no prevé la utilización de identificadores temporales con lo que el RFPI puede ser directamente utilizado para identificar y localizar geográficamente al usuario de dicha estación DECT. Dado que DECT tiende a ser utilizado como tecnología de acceso a redes de telefonía fija, las implicaciones de privacidad no son tan extensas como en el caso de GSM. Sin embargo, la capacidad de geolocalizar de forma única al usuario puede ser de cierta relevancia en determinadas situaciones y en combinación con otros ataques a DECT como interceptación de llamadas o análisis de tráfico.

En ausencia de llamada en curso, las tramas DECT enviadas por la FP únicamente poseerán campo A y no existirán normalmente tramas enviadas por partes móviles (PPs). En el momento en que comienza una comunicación, por ejemplo una llamada, la transferencia bidireccional de voz comienza inmediatamente produciéndose tráfico PP y FP, con campo B en todas las tramas. Efectivamente, el campo B contiene la voz transmitida codificada habitualmente con el códec G726, por lo que la presencia de tramas $FP \rightarrow PP$ y $PP \rightarrow FP$ con campo B revelará la existencia de una llamada, independientemente de si la comunicación se encuentre cifrada o no.

Por lo tanto, en DECT, tanto en redes protegidas con DSC como en aquellas que no lo están, el análisis del tráfico permitirá determinar la existencia, duración y horario de las llamadas producidas en la línea fija, facilitando la generación de perfiles geolocalizados de los usuarios de dichas redes.

A modo de ejemplo, una banda criminal podría utilizar una implementación de monitorización de DECT basada en SDRs de bajo coste, como la que se ha desarrollado para la presente tesis, para realizar perfiles geolocalizados de usuarios residenciales, de tal manera que puedan fácilmente inferir cuando un usuario se encuentra ausente de su vivienda para robar en ella.

Cuando un usuario recibe o efectúa una llamada telefónica mediante un sistema DECT, el número de teléfono llamante o llamado es transferido en el

campo C de las tramas DECT. En un sistema DECT adecuadamente protegido mediante el uso de DSC, dicha información viajará cifrada, por lo que un atacante deberá primero romper la clave DSC para obtenerla. En redes no cifradas, dicha información puede ser obtenida directamente del campo C de las tramas DECT, permitiendo la elaboración de perfiles más avanzados de usuarios en base al análisis de sus llamadas y la interrelación entre ellos.

De la misma manera, en una llamada telefónica en curso, un atacante podrá obtener remotamente toda la conversación, en perfecta calidad digital, en sistemas DECT que no implementen cifrado, o en aquellos que implementen cifrado DSC en el caso en que el atacante logre comprometer la clave.

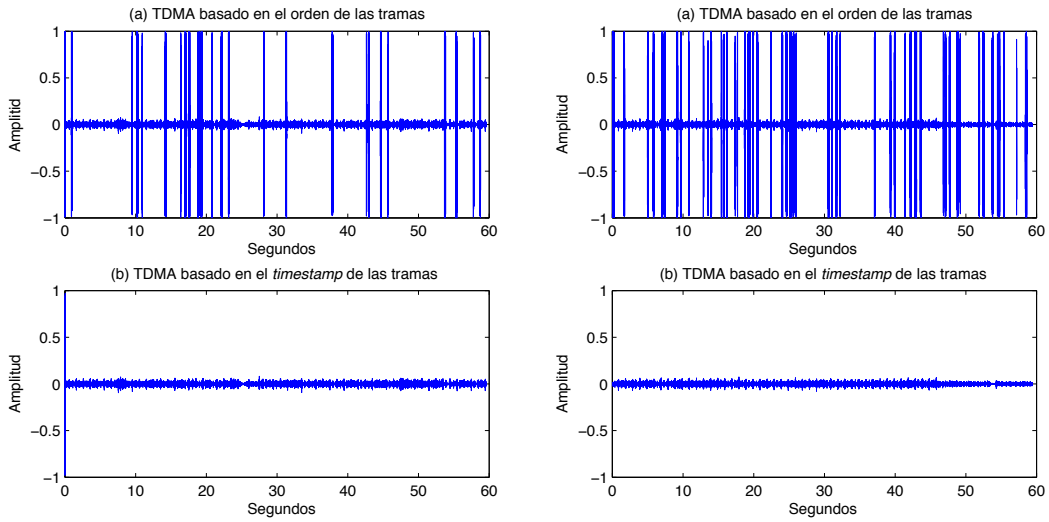
2.4.2. Interceptación pasiva de comunicaciones mediante *DECT-Eye*

En la presente sección de la tesis se exploran las capacidades que los SDR de bajo coste ofrecen para atacar la seguridad y privacidad de las comunicaciones DECT en el contexto de su utilización para comunicaciones personales de voz en entornos residenciales.

Con el fin de recrear experimentalmente un ataque pasivo a la privacidad de DECT, se utiliza la plataforma de monitorización basada en SDR desarrollada en la sección 2.3.2 junto con los dispositivos SDR detallados en la sección 1.2 del capítulo 1, el Realtek RTL2832U con sintonizador Elonics E4000 y el USRP N210 de Ettus Research.

Ambos SDR, representativos de los extremos opuestos del rango de Radios Definidas por Software de bajo coste, son analizados y comparados entre sí en el escenario de su utilización para atacar la privacidad de las comunicaciones DECT.

Con el objetivo de analizar la viabilidad de realizar una interceptación pasiva de comunicaciones de voz DECT con Radios Definidas por Software, se recrea en laboratorio una situación real utilizando un kit DECT marca Phillips modelo 515, cuya base (FP) y terminal móvil (PP) son situados primeramente a 3 metros y posteriormente a 15 metros del atacante simulado.



(a) USRP N210 con ganancia de 0 (b) RTL-SDR con ganancia de 15

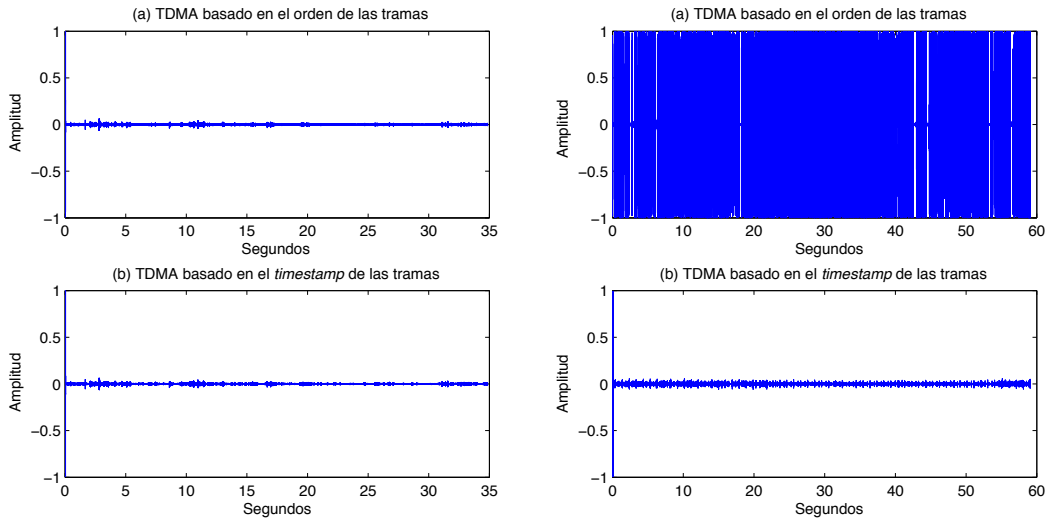
Figura 2.12: Resultados de captura de audio en transmisión DECT interceptada a 3 metros de distancia

En todos los casos, para realizar el ataque se utiliza una antena omnidireccional de baja ganancia (1dBi) y no se usa ningún tipo de amplificador externo. Para la simulación de la conversación personal del escenario del ataque, se reproduce de forma cíclica un fichero musical en el auricular del teléfono.

Se realiza el experimento, tanto con el USRP N210 como con el RTL-SDR, utilizando los valores óptimos de 0.85 para la sensibilidad del demodulador GFSK y una distancia Hamming de 2.

Con el objetivo de proporcionar una comparativa entre ambas implementaciones de TDMA desarrolladas (basada en orden de paquetes y basada en tiempo), se presentan para cada uno de los 4 escenarios, la visualización de la forma de onda del sonido digital extraído de la conversación interceptada.

Las figuras 2.12a 2.13a (para el USRP N210) 2.12b y 2.13b (para el RTL-SDR), muestran la visualización del audio extraído de la conversación DECT, para las distancias de 3 metros y 15 metros. En cada caso se muestra el resultado de la extracción de audio para ambas implementaciones de TDMA, la basada en el orden de los paquetes y la basada en tiempo de recepción. El resultado del audio no ha sido post-procesado con ninguna técnica de mejora



(a) USRP N210 con ganancia de 15 (b) RTL-SDR con ganancia de 15

Figura 2.13: Resultados de captura de audio en transmisión DECT interceptada a 15 metros de distancia

o reducción de ruido.

Los picos presentes en el audio extraído con la implementación de TDMA basada en el orden de los paquetes (parte -a- de las figuras 2.12a 2.13a, 2.12b y 2.13b), corresponden con datos de audio extraídos incorrectamente derivados de un proceso de *descrambling* inapropiado, debido a errores en la asignación del número de trama al paquete. Efectivamente, tal y como se ha explicado previamente, en dicha implementación de TDMA, una pérdida en una trama de una multitrama provocará una asignación errónea del número de trama al resto de los paquetes DECT de la trama actual. Dado que la decodificación del audio es dependiente del número de trama, dicho error de asignación derivará en la invalidación de la mayoría del contenido de datos de voz en la mayoría de las tramas de la multitrama.

Sin embargo, en el caso de la implementación TDMA mejorada basada en el tiempo de recepción de los paquetes DECT, el cálculo del número de trama ya no es dependiente del orden del resto de tramas, por lo que una pérdida de tramas en la multitrama ya no arruinará los datos de voz contenidos en el resto de la multitrama. En efecto, se puede observar, en la parte -b- de las figuras

Dispositivo	Distancia	Ganancia	% Tramas	% Multitramas
RTL-SDR	3 metros	15	95 %	84 %
RTL-SDR	15 metros	15	85 %	13 %
N210	3 metros	0	92 %	92 %
N210	15 metros	15	93 %	98 %

Tabla 2.4: Parámetros de calidad de señal para los diferentes escenarios experimentales

2.12a 2.13a, 2.12b y 2.13b, que la calidad del audio extraído, bajo las mismas condiciones, es muy superior en comparación con la implementación TDMA anteriormente presentada (en la parte -a- de las mismas figuras). De hecho, dado que la voz se encuentra codificada con el códec G726, éste ofrece tolerancia a la pérdida de paquetes, de tal manera que es capaz de reconstruir la información contenida en las tramas perdidas. El resultado es un audio virtualmente perfecto, incluso en el escenario de la figura 2.13b donde la implementación TDMA presentada previamente ofrecía un rendimiento bastante pobre.

La tabla 2.4 muestra una comparativa de los parámetros de calidad de la señal DECT monitorizada para los diferentes escenarios recreados en el experimento. La cuarta columna indica el porcentaje de tramas correctamente recibidas sobre el total enviadas por la FP y la PP (200 tramas por segundo). La quinta columna muestra el porcentaje de multitramas correctamente recibidas sobre el total de las enviadas por la FP y la PP.

Analizando en la tabla 2.4 el porcentaje de multitramas correctamente recibidas para los diferentes escenarios, junto con las figuras 2.13a y 2.13b, podemos comprobar el rendimiento superior que la plataforma USRP N210 ofrece respecto a la RTL-SDR, hecho no sorprendente dado la diferencia de coste entre ambas.

Sin embargo, a pesar de las diferencias, el resultado final de la simulación de interceptación de comunicación de voz inalámbrica DECT es similar en ambas plataformas para la implementación de TDMA basada en tiempo de recep-

ción de paquetes, dado el papel jugado por el códec G726.

Tal y como se ha descrito anteriormente, dichos resultados han sido obtenidos utilizando antenas omnidireccionales de baja ganancia (1dBi) y sin utilizar amplificadores externos. Utilizando antenas direccionales de mayor ganancia así como amplificadores, se podría ampliar substancialmente el rango de acción obteniendo rendimientos similares.

La expresión clásica de atenuación de señal en el espacio, en dB, es $FSPL(dB) = 20log_10(d) + 20log_10(f) - 147{,}55$, donde f es la frecuencia de la comunicación de radio en Hz y d es la distancia en metros desde el SDR y el origen de la transmisión DECT. Existen antenas direccionales para la frecuencia de DECT con una ganancia de 11dB por menos de 100 euros, las cuales podrían ser combinadas con Amplificadores de Bajo Ruido (LNA) disponibles a bajo coste ofreciendo 10dB adicionales. La ganancia adicional derivada de dicho escenario ((11dB-1db) + 10 dB = 20 dB) sería del orden de 10 veces la distancia de recepción. Por lo tanto, con un presupuesto menor de 100 euros es posible obtener los resultados mostrados en las figuras 2.13a y 2.13b, para una distancia de 150 metros.

Utilizando antenas y amplificadores más caros, dentro de un presupuesto de unos miles de euros, es posible obtener una ganancia de 40dB, lo cual permitiría obtener resultados comparables a las figuras 2.13a y 2.13b, para una distancia de 1.5 Km.

2.5. Interceptación de comunicaciones cifradas en DECT

2.5.1. Autenticación y cifrado en DECT

Con el objetivo de proteger la confidencialidad de las comunicaciones, DECT soporta el cifrado de datos mediante un algoritmo conocido como *DECT Standard Cipher* (DSC). Los detalles internos del algoritmo DSC así como los ataques existentes contra el mismo son descritos en detalle en la sección 2.6 donde

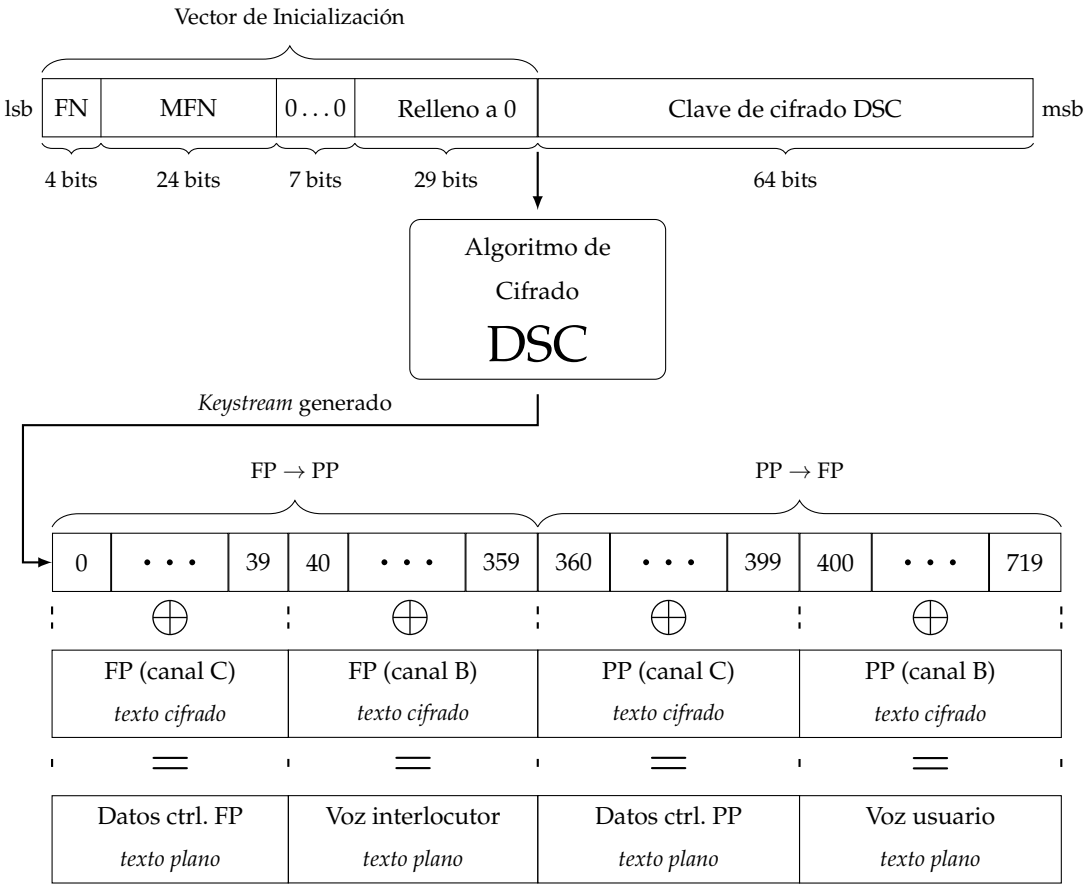


Figura 2.14: Protocolo operativo de cifrado y descifrado en DECT

se introduce un novedoso criptoanálisis contra dicho algoritmo.

DSC es un algoritmo de cifrado de flujo que toma como entradas un vector de inicialización de 35 bits junto con una clave secreta de 64 bits para producir un *keystream* de 720 bits de longitud. El proceso operativo de cifrado y descifrado que se describe a continuación, se encuentra esquematizado en la figura 2.14.

El vector de inicialización de 35 bits es específico para cada trama DECT a cifrar o descifrar, siendo determinado en base al número de trama y el número de multi-trama. A pesar de que la longitud del vector de inicialización del DSC sea de 35 bits, tan solo 28 de ellos son efectivos, ya que el número de trama es determinado con 4 bits y el número de multi-trama con 24. De los 35 bits del vector de inicialización, los 4 bits menos significativos son determinados por el número de trama, los siguientes 24 bits por el número de multi-trama y los

7 bits más significativos son fijados a ceros. Internamente, DSC extenderá con ceros dicho vector de inicialización hasta una longitud de 64 bits y lo concatenará con la clave secreta, componiendo un total de 128-bits, tal y como se representa en la parte superior de la figura 2.14.

La ejecución del algoritmo DSC por cada trama enviada o recibida, resulta en la generación de un *keystream* de 720 bits de longitud, específico para cada trama, que será el utilizado para el cifrado y descifrado de datos. Dicho *keystream* de 720 bits se divide en 2 segmentos de 360 bits cada uno, los cuales a su vez se dividen en 2 partes de 40 bits y 320 bits respectivamente.

El primer segmento de 360 bits se utiliza para el cifrado y descifrado de los datos enviados por la Parte Fija DECT (FP) o estación base, mientras que el segundo segmento se utiliza para los datos enviados por la Parte Portátil (PP). Si los datos transmitidos son pertenecientes al campo C del paquete DECT (datos de control), se utilizarán los primeros 40 bits del segmento. Los 320 bits restantes serán los utilizados para los datos pertenecientes al campo B.

En cualquiera de los casos, como es habitual con los algoritmos de cifrado de flujo, el cifrado y descifrado se realizan aplicando una operación lógica XOR, bit a bit, entre la parte correspondiente del *keystream* y los datos a cifrar o descifrar.

La figura 2.14 detalla todo el proceso de cifrado descrito previamente.

Dado que el número de trama y multi-trama DECT se calcula en base a información que viaja en texto plano, la seguridad del cifrado DSC en DECT se sustenta mediante la clave secreta de 64 bits. En efecto, para que un terminal DECT pueda realizar una comunicación de voz de forma cifrada con una estación base, ambos habrán de poseer necesariamente una clave secreta de 64 bits idéntica.

Básicamente existen 2 formas mediante las cuales dicha clave es establecida. La primera consiste en que sea generada e introducida en la Parte Fija y Parte Portátil por el fabricante del dispositivo DECT, como parte de su proceso de producción. En dicho escenario, la clave secreta del DSC se denomina SCK.

Sin embargo, la forma más habitual es aquella en la que la clave es nego-

ciada de forma aleatoria entre la Parte Fija y la Parte Portátil, al inicio de una conexión. En efecto, este es el caso para la casi totalidad de teléfonos DECT analizados en esta tesis y es un requisito establecido por el perfil de interoperatividad GAP del estándar DECT. En el caso de que FP y PP soporten la renovación de clave, la DCK podría ser renegociada en varias ocasiones durante una conexión en curso.

La negociación de la clave DCK ocurre durante el proceso de autenticación de la Parte Portátil utilizando el algoritmo de Autenticación Estándar de DECT denominado DSAA. El propósito de dicho proceso de autenticación de la Parte Portátil es doble. Por un lado tiene como objetivo confirmar que la Parte Portátil, que pretende establecer la conexión, se encuentra autorizada para hacerlo. Este paso es importante para evitar ataques tipo MiTM y de suplantación de identidad donde un tercero pueda utilizar una linea DECT sin autorización para efectuar llamadas. Por otro lado, dicho proceso de autenticación permitirá a ambos extremos, FP y PP, acordar una clave DCK aleatoria para cifrar la comunicación.

Tal y como se ha descrito anteriormente, el algoritmo DSAA, a pesar de formar parte integral del estándar DECT, solo se encontraba disponible bajo acuerdos de confidencialidad. Sin embargo, en [95] los autores determinaron el funcionamiento del algoritmo mediante un proceso de ingeniería inversa de hardware y publicaron sus detalles.

A modo de resumen, el algoritmo DSAA es un cifrado de bloque que recibe como entradas 2 bloques de 128 bits y 64 bits respectivamente para producir una salida de 128 bits. Existen 4 formas diferentes de usar el algoritmo de cifrado DSAA, denominadas A_{11}, A_{12}, A_{21} y A_{22}, las cuales difieren entre ellas en el significado semántico de sus entradas y salidas, siendo éstas procesadas también de forma diversa. La figura 2.15 presenta de forma esquemática las funciones A_{11}, A_{12}, A_{21} y A_{22} en relación al algoritmo de cifrado DSAA. Dichas funciones serán utilizadas por los procesos de autenticación y emparejado criptográfico de terminales DECT, tal y como será detallado a continuación.

El proceso de autenticación de la Parte Portátil se describe en la figura 2.15.

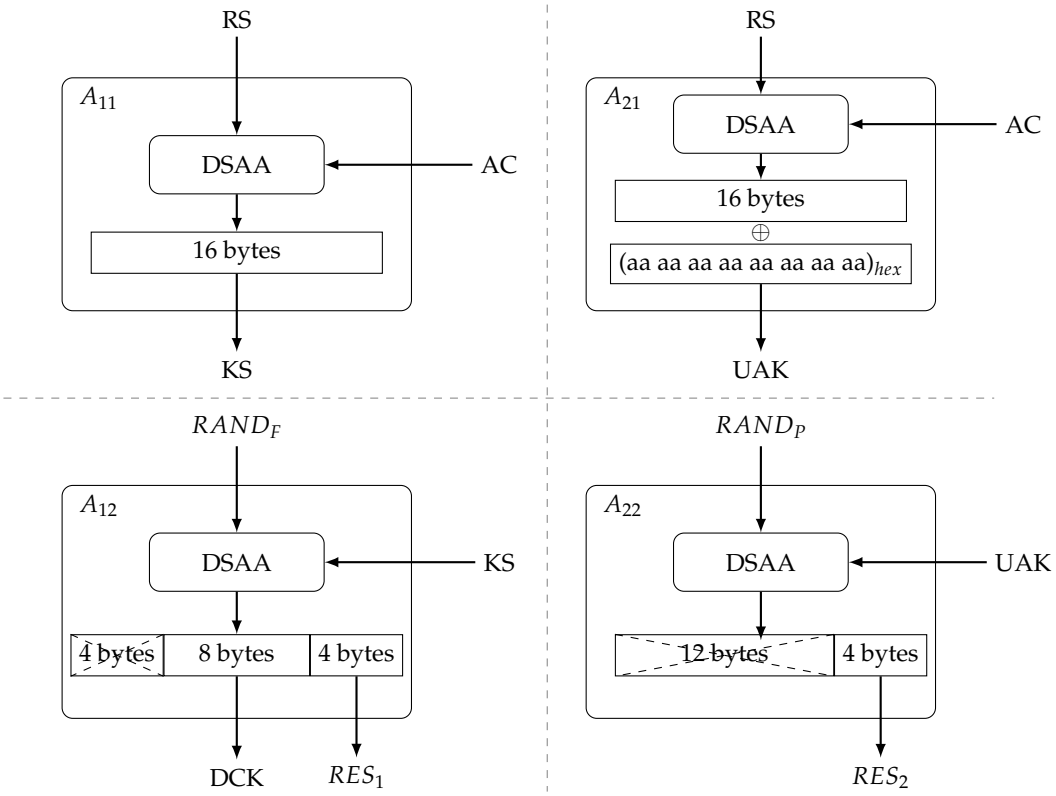


Figura 2.15: Las 4 formas de uso de DSAA

FP y PP deben previamente poseer una clave común de 128 bits de longitud, denominada UAK. El proceso de autenticación es iniciado por la Parte Fija (FP), la cual generará 2 números aleatorios de 64 bits denominados RS y $RAND_F$.

En un primer paso, ambos números RS y $RAND_F$ serán enviados a la Parte Portátil (PP) como parte de un mensaje denominado $AUTH\_REQUEST$. Una vez dicho mensaje es recibido por la PP, ambas partes calcularán los valores KS, DCK and $XRES_1$ de la siguiente manera:

$$\begin{aligned} KS &= A_{11}(RS, UAK) \\ (DCK, XRES_1) &= A_{12}(RAND_F, KS). \end{aligned}$$

Una vez DCK y $XRES_1$ hayan sido calculados ($XRES_1$ será denominado RES_1 al ser calculado por PP), PP enviará $XRES_1$ de vuelta a FP como parte de un mensaje denominado $AUTH\_REPLY$. FP verificará que el valor $XRES_1$ recibido de PP sea idéntico al calculado por él mismo. En caso afirmativo el pro-

ceso de autenticación habrá terminado exitosamente y ambos extremos habrán acordado una clave DCK que será utilizada para cifrar la comunicación dentro de la conexión establecida. La figura 2.16 detalla de forma gráfica todo el proceso.

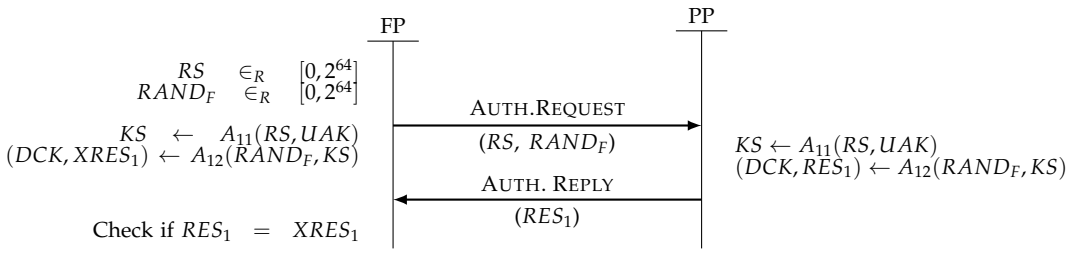


Figura 2.16: Autenticación de PP en DECT

Un atacante pasivo que hubiera monitorizado el proceso solo podría determinar la clave DCK si tuviera en su posesión la clave UAK de 128-bits. Dicha clave secreta UAK, compartida por FP y PP, puede ser generada básicamente de 2 formas. La primera sería que fuese introducida en FP y PP por el fabricante durante el proceso de fabricación del dispositivo DECT. En este caso, sin embargo, la interoperatividad entre dispositivos de diferentes fabricantes no sería posible ya que no compartirían una clave UAK idéntica.

Por lo tanto, para garantizar la interoperatividad, el perfil GAP del estándar DECT, define un proceso de emparejamiento criptográfico entre FP y PP que permite a ambos acordar una clave UAK aleatoria. Dicho proceso es implementado por la casi totalidad de dispositivos DECT del mercado. En términos prácticos, el proceso de emparejamiento criptográfico, permite registrar un terminal DECT (PP) de un fabricante en la estación base DECT (FP) de otro fabricante, introduciendo un código PIN conocido por ambos.

En los últimos tiempos, el emparejamiento criptográfico de DECT ha adquirido especial relevancia gracias a los sistemas de comunicación unificados (UC). Los UC permiten la integración de diferentes tipos de comunicaciones en tiempo real, tales como telefonía, videoconferencia, control de llamadas y mensajería instantánea. Las llamadas de voz pueden ser encaminadas tanto

por líneas de comunicación clásicas como PSTN y RDSI, como por protocolos seguros basados en VoIP sobre redes de datos como Internet.

Dichos UC, que tradicionalmente eran desplegados en entornos profesionales, se encuentran actualmente en plena expansión en entornos residenciales dentro del ecosistema de la casa inteligente. Dada la popularidad, fiabilidad y flexibilidad de los dispositivos DECT, existen cada vez más sistemas de comunicación unificados que permiten la integración de dispositivos DECT mediante el perfil de interoperatividad GAP.

En un contexto heterogéneo como éste, el emparejado criptográfico DECT permite registrar cualquier teléfono DECT con el sistema de comunicación unificado. En términos prácticos, el PIN será seleccionado en la interfaz de gestión del UC y, al igual que ocurre con cualquier FP y PP DECT, el usuario utilizará el menú del terminal para encontrar la estación base y registrarse tras insertar el código PIN correcto.

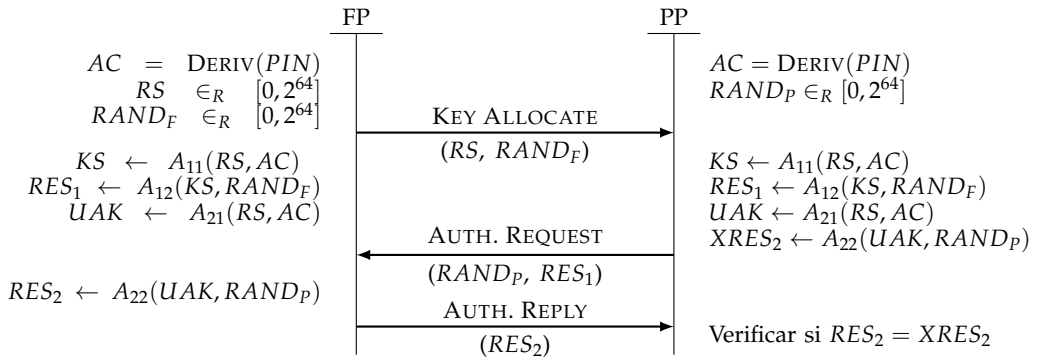


Figura 2.17: Emparejado criptográfico en DECT

El proceso se describe en la figura 2.17. Tras la petición de registro de PP, FP genera 2 números aleatorios de 64 bits denominados RS y $RAND_F$ y los envía a PP dentro de un mensaje denominado $KEY\_ALLOCATE$. Utilizando el código PIN que habrá de ser conocido tanto por FP como por PP, ambos calculan $KS = A_{11}(RS, AC)$. Una vez KS hay sido calculado, UAK será determinado calculando $A_{21}(RS, AC)$. A su vez, RES_1, que actuará como código de confirmación, se calculará como $A_{12}(KS, RAND_F)$.

Una vez RES_1 haya sido calculado por ambos extremos, PP generará un número aleatorio de 64 bits, denominado $RAND_P$ y lo enviará a FP junto con el valor RES_1 dentro de un mensaje del tipo $AUTH\_REPLY$. Finalmente, PP realizará el mismo cálculo con el objetivo de confirmar que el valor RES_1 recibido concuerda con el calculado. En caso afirmativo, el UAK negociado será confirmado y el emparejamiento criptográfico terminará exitosamente. FP y PP habrán acordado una clave UAK aleatoria de 128-bits que será utilizada desde ese momento para cualquier proceso de autenticación y cifrado. Todas las futuras claves DCK utilizadas para el cifrado de comunicaciones, derivarán de dicha clave UAK, tal y como se ha descrito previamente en esta sección.

2.5.2. Un ataque contra el sistema de emparejamiento criptográfico de DECT

En esta sección se presenta un ataque pasivo contra el mecanismo de emparejamiento criptográfico de DECT. El ataque propuesto es capaz de recuperar el código PIN y derivar la clave UAK negociada por FP y PP, monitorizando de forma pasiva las comunicaciones durante el proceso de emparejamiento.

Tal y como se ha mencionado anteriormente, el proceso de emparejamiento criptográfico DECT se compone de 3 pasos, donde una serie de valores son generados de forma aleatoria por las partes y son intercambiados por el aire.

En un primer paso, los valores RS y $RAND_F$ son generados de forma aleatoria por FP y son transmitidos en texto plano a PP. Tal y como se ha demostrado anteriormente en la sección 2.4.2, un atacante puede capturar todo el intercambio de mensajes entre FP y PP, potencialmente desde cientos de metros, utilizando SDRs de bajo coste. En este escenario, el mensaje $KEY\_ALLOCATE$ podrá ser interceptado, por lo que los valores RS y $RAND_F$ serán conocidos para un atacante.

Por lo tanto, para derivar la clave UAK negociada, un atacante tendría que determinar los siguientes valores:

$$
\begin{aligned}
AC &= \text{DERIV}(PIN) \\
UAK &= A_{21}(RS, AC).
\end{aligned}
$$

Dado que el código PIN sería desconocido para el atacante, existirían un total de 10^n posibles claves UAK, siendo n el número de dígitos del código PIN, típicamente 4.

Sin embargo, el atacante también podría interceptar el mensaje de respuesta $AUTH\_REQUEST$ que PP enviaría a FP tras la recepción del mensaje $KEY\_ALLOCATE$. En dicho mensaje, el PP comunica los valores $RAND_P$ y RES_1 calculados de la siguiente manera:

$$
\begin{aligned}
AC &= \text{DERIV}(PIN) \\
KS &= A_{11}(RS, AC) \\
RES_1 &= A_{12}(KS, RAND_F).
\end{aligned}
$$

Armado con esta información, el atacante podría utilizar el valor RES_1 interceptado para lanzar una búsqueda exhaustiva sobre todos los posibles valores del código PIN desconocido a fin de encontrar aquel que satisfaga la ecuación $RES_1 = A_{12}(A_{11}(RS, \text{DERIV}(PIN)), RAND_F)$.

En la figura 2.18 se presenta en detalle el algoritmo completo para la recuperación del código PIN desconocido y el derivado de la clave UAK negociada, en base a los valores $RAND_F$, KS y RES_1 interceptados.

En efecto, una vez el atacante se encuentre en posesión del PIN correcto, la clave UAK podrá ser calculada como $UAK = A_{21}(RS, \text{DERIV}(PIN))$.

En posesión de la clave UAK, el atacante podrá descifrar en tiempo real cualquier conversación cifrada futura, interceptando el intercambio de valores en el proceso de la Autenticación del PP, descrito en la sección anterior y detallado en la figura 2.16. Utilizando dichos valores podrá derivar la clave de sesión DCK de 64-bits y proceder al descifrado de la comunicación en curso.

En efecto, utilizando la clave UAK junto con los valores RS y $RAND_F$ intercambiados en el mensaje $AUTH\_REQUEST$ enviado por FP como parte del

Algoritmo 1: Algoritmo de ataque al código PIN

for $PIN = 0$ *to* 10^4 **do**

- $AC = \text{DERIV}(PIN)$;
- $KS \leftarrow A_{11}(RS, AC)$;
- $CRES_1 \leftarrow A_{12}(KS, RAND_F)$;
- **if** $RES_1 == CRES_1$ **then**
  - $UAK \leftarrow A_{21}(RS, AC)$;
  - **return** UAK;

Figura 2.18: Algoritmo de ataque al código PIN

proceso de Autenticación del PP, el atacante podrá derivar la clave DCK de la siguiente manera.

$$\begin{aligned} KS &= A_{11}(RS, UAK) \\ (DCK, RES_1) &= A_{12}(RAND_F, KS). \end{aligned}$$

La clave DCK derivada puede ser confirmada verificando que el valor RES_1 calculado se corresponde con el enviado por PP en el mensaje $AUTH\_REQUEST$.

Una vez la clave DCK de 64 bits haya sido determinada, toda la transmisión cifrada entre FP y PP podrá ser interceptada y descifrada por el atacante, siendo posible la extracción del contenido en texto plano, tanto del campo B como del canal C, siguiendo el proceso detallado previamente en la figura 2.14.

Cualquier nueva clave DCK resultante de una eventual negociación DCK, en caso de la aplicación del procedimiento de renegociación de claves del estándar, podrá ser derivada por el atacante de la misma forma, ya que éste se encuentra en posesión de la clave permanente UAK de 128 bits.

2.5.3. Resultados experimentales de la interceptación práctica de comunicaciones DECT cifradas

El ataque contra el emparejamiento criptográfico DECT descrito en la sección anterior ha sido implementado de forma práctica en entorno de laborato-

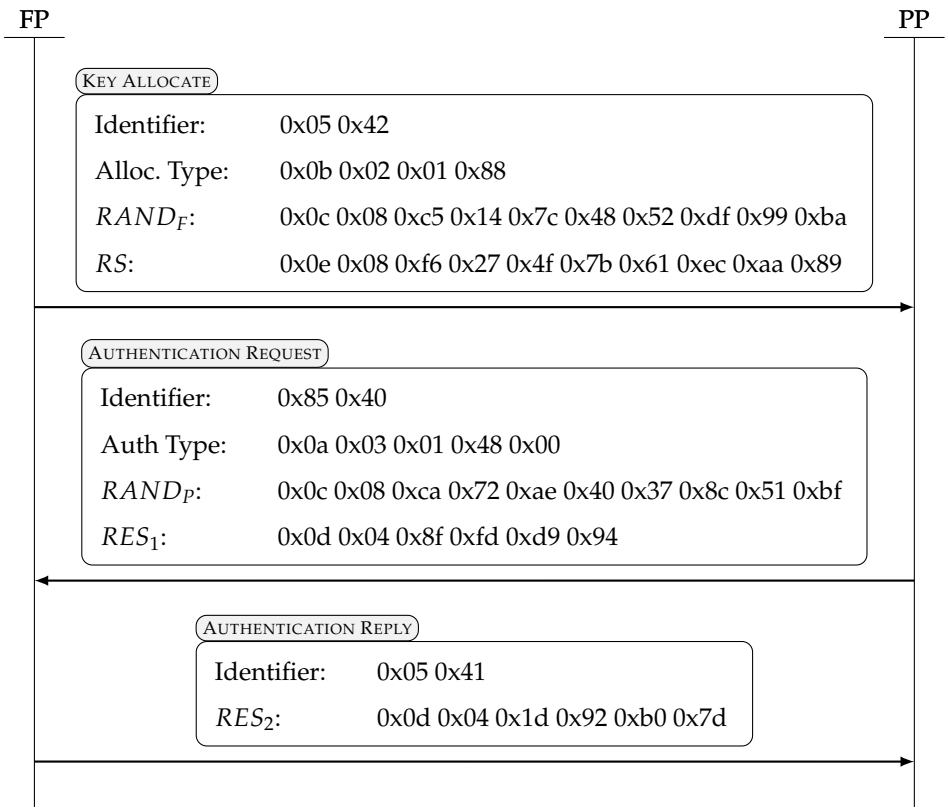


Figura 2.19: Captura experimental de emparejamiento criptográfico DECT

rio utilizando diferentes dispositivos DECT disponibles en el mercado.

En la presente sección se detalla la implementación práctica de interceptación de comunicaciones DECT cifradas y se muestran los resultados experimentales. Dado que la interceptación de comunicaciones no cifradas ha sido demostrada en detalle previamente en la sección 2.4.2, la presente sección se centrará en los detalles específicos relativos al empleo de cifrado.

La figura 2.19 muestra el intercambio de mensajes entre FP y PP durante un proceso de emparejado criptográfico DECT en uno de los teléfonos analizados. Se puede observar que el proceso es similar al descrito anteriormente en la figura 2.17 en la sección anterior. En la figura 2.19 se representa en hexadecimal el intercambio de datos entre ambos FP y PP. El flujo de datos se ha procesado para representar los mensajes $KEY\_ALLOCATE$, $AUTHENTICATION\_REQUEST$ y $AUTHENTICATION\_REPLY$. La tabla 2.5 muestra los valores de los diferentes parámetros intercambiados.

Nombre parámetro	Valor
$RAND_F$	ba99df52487c14c5
RS	89aaec617b4f27f6
$RAND_P$	bf518c3740ae72ca
RES_1	94d9fd8f

Tabla 2.5: Valores extraídos en el emparejamiento criptográfico DECT

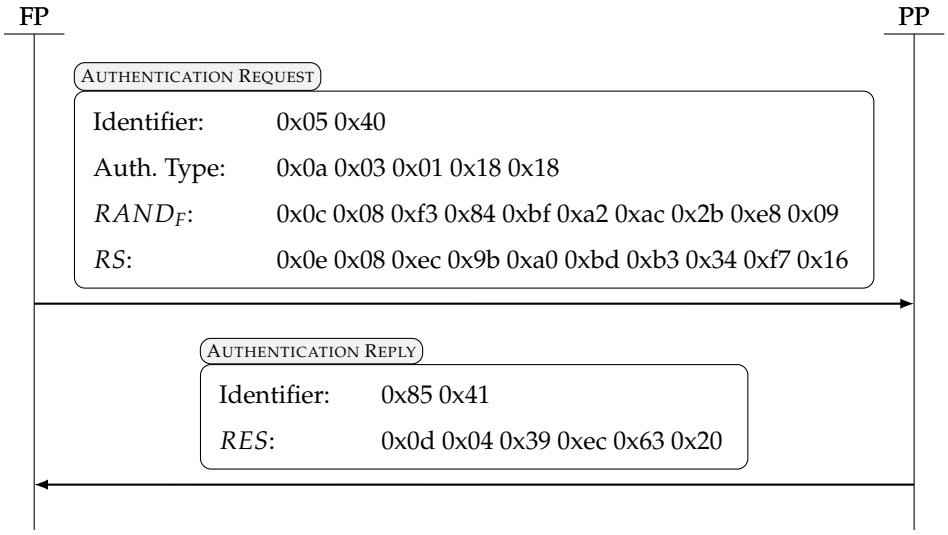


Figura 2.20: Resultados experimentales de la Autenticación de la Parte Portátil

Con el objetivo de implementar el ataque descrito en la sección 2.5.2, se implementa y ejecuta el algoritmo descrito en la figura 2.18 con los parámetros $RAND_F$, RS y RES_1 extraídos de la comunicación interceptada.

La implementación realizada del algoritmo es capaz de realizar la búsqueda exhaustiva sobre las 10^4 combinaciones para un PIN de 4 dígitos, en menos de 1 segundo. En el caso del experimento, el algoritmo determina rápidamente que el PIN que satisface la ecuación $RES_1 = A_{12}(A_{11}(RS, \text{DERIV}(PIN)), RAND_F)$, es 0000.

Una vez determinado el código PIN correcto, el algoritmo calcula la clave UAK de 128 bits como $UAK = A_{21}(RS, \text{DERIV}(PIN))$. En el escenario del experimento, la clave UAK tiene el valor de `d3fc924f771f5278c74f91fc6a79d0d2`.

Nombre parámetro	Valor
$RAND_F$	9e82baca2bf84f3
RS	16f734b3bda09bec
RES	2063ec39

Tabla 2.6: Valores intercambiados en el experimento de la Autenticación de la Parte Portátil

Una vez el teléfono ha sido registrado en la estación base DECT, se procede a realizar una llamada de voz a la vez que, simulando un ataque de interceptación pasivo, se monitoriza remotamente la comunicación DECT cifrada. La figura 2.20 muestra el intercambio de mensajes de control en el canal *C* al inicio de la comunicación de la llamada de voz. En la figura se muestran los 2 mensajes intercambiados ($AUTHENTICATION\_REQUEST$ y $AUTHENTICATION\_REPLY$) junto con su contenido representado en valores hexadecimales.

Este proceso corresponde con la Autenticación de la Parte Portátil, tal y como se ha descrito en la sección 2.5.2 y se detallaba en la figura 2.16.

Los valores intercambiados durante el experimento, correspondientes a la figura 2.20 se detallan en la tabla 2.6.

Utilizando la clave permanente de 128 bits UAK obtenida del proceso anterior, se calcula ahora DCK y RES mediante la siguiente ecuación.

$$(DCK, RES) = A_{12}(RAND_F, A_{11}(RS, UAK))$$

A modo de verificación adicional, se comprueba que el valor RES calculado concuerda con el observado en el mensaje AUTH_REPLY interceptado. En el experimento dicho valor es de hecho `2063ec39` y concuerda con el interceptado en el mensaje, por lo que se verifica que la clave UAK derivada anteriormente es correcta.

Por lo tanto, la clave DCK de 64 bit calculada en el proceso habrá necesariamente de ser aquella con la que se cifre el resto de la comunicación. El valor calculado para DCK en el experimento, es `1d8796079fca243a`.

En posesión de la clave DCK, se procede a descifrar la conversación telefónica siguiendo el procedimiento descrito en la sección 2.5.2 y esquematizado en la figura 2.14.

Por cada trama DECT interceptada, se determina el número de trama y de multi-trama. El número de trama es determinado siguiendo el proceso descrito en la sección 2.3.2, bien sea mediante el conteo de tramas respecto a la inicial de una multi-trama, o mediante el uso de un *timestamp* estimado para la recepción de la trama. El número de multi-trama se determina por un método similar junto con un reajuste automático en base al número de multi-trama que es anunciado periódicamente por la Parte Fija.

Junto con la clave DCK de 64 bits derivada anteriormente, el número de trama y multi-trama son utilizados por el algoritmo DSC para producir 720 bits de *keystream* específicos para el descifrado de dicha trama. El proceso de descifrado de la voz que ha sido implementado, realiza una operación lógica XOR de la porción correspondiente del *keystream* y los datos provenientes del campo *B* del paquete DECT contenido en el trama. En efecto, dichos datos cifrados en el campo *B* corresponden al audio codificado. La implementación del proceso de descifrado se detalla esquemáticamente en la figura 2.14.

Una vez los datos del campo *B* de los paquetes DECT han sido descifrados, se intercambian los *nibbles* y se decodifica la voz utilizando el códec G.726. El resultado es la voz transmitida, la cual, en nuestros experimentos donde la interceptación es realizada a pocos metros de distancia, puede ser escuchada con una calidad perfecta.

En la figura 2.21 se muestra un gráfico, de la amplitud sobre el tiempo, del audio extraído de la llamada de voz DECT cifrada.

A modo de contraste, en la figura 2.22 se muestra un gráfico similar de un intento de extraer el audio del mismo experimento, en esta ocasión sin descifrarlo previamente. El resultado es ruido uniforme sin ningún tipo de patrón reconocible. Cualquier intento de descifrar el audio con una clave incorrecta, lleva al mismo resultado, incluso aunque la clave DCK incorrecta difiera en un único bit de la correcta.

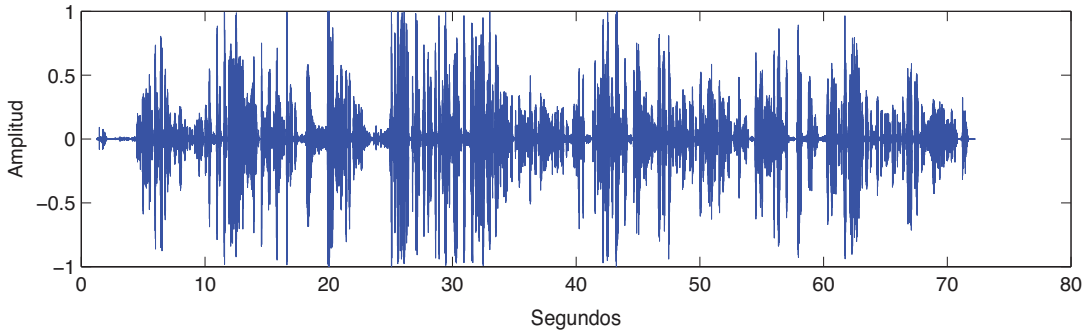


Figura 2.21: Audio de la voz descifrado con la clave DCK correcta

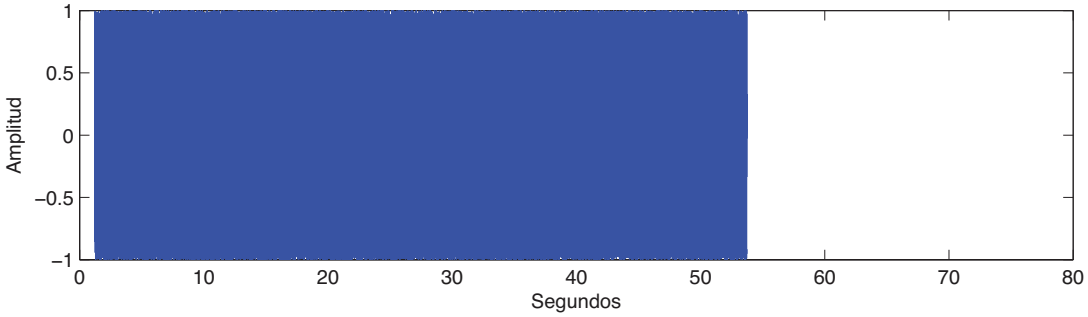


Figura 2.22: Intento de extracción de audio cifrado sin la clave DCK correcta

2.6. Criptoanálisis mejorado del algoritmo de cifrado DECT

2.6.1. El DSC en detalle

La metodología de utilización del protocolo de cifrado DSC se detalla en la parte 7 de la Interfaz Común del estándar DECT [54]. Sin embargo, al igual que ocurre con el DSAA, la implementación del DSC se ofrece únicamente bajo acuerdos de confidencialidad y no se encuentra disponible para su consulta en el estándar público.

Tal y como se ha descrito en la sección 2.2, en [117] los autores publicaron el algoritmo extraído de una implementación hardware por procedimientos de ingeniería inversa. Los autores, también presentaron en la misma publicación un criptoanálisis del DSC inspirado en los ataques de correlación existentes para el algoritmo A5/1 de GSM [99].

En esencia, DSC es un algoritmo de cifrado de flujo con clave de 64-bits ba-

sado en 4 registros de desplazamiento con retroalimentación lineal, $R_1, \ldots, R_4$, (LFSR) en configuración Galois con pulso de reloj irregular, junto con un combinador no lineal de salida, $\mathcal{O}$, dotado de un bit de memoria, z. Los tres primeros registros, llamados los registros principales, son los utilizados para producir el flujo de bits de la clave, denominado *keystream*. Dicho *keystream* será utilizado posteriormente como clave en un cifrado de relleno de un solo uso (OTP) que cifrará mediante XOR los datos en texto plano de un paquete DECT determinado. Por otro lado, el cuarto registro LFSR se utiliza, en combinación con los otros tres, para controlar el reloj irregular de los registros principales. Tal y como se detallará a continuación, el DSC se inicializa con un Vector de Inicialización (IV) de 35 bits y una clave simétrica de 64 bits, denominados IV y KEY respectivamente.

Cada pareja de IV y KEY es utilizada para producir 720 bits de *keystream*, divididos en 2 segmentos (KSS) de 360 bits cada uno. Tal y como se describe en la parte 7 del estándar [54], el primer segmento de *keystream* es utilizado para cifrar las tramas DECT enviadas por la estación base (FP), mientras que el segundo es utilizado para cifrar las tramas enviadas por el terminal móvil (PP). En ambos casos, los primeros 40 bits son utilizados para cifrar los datos pertenecientes al canal C dentro del campo A de la trama (si lo hubiera), mientras que el resto de bits son utilizados para cifrar el los datos pertenecientes al campo B.

En la figura 2.23 se muestra el algoritmo DSC al detalle. Dicha figura ha sido creada modificando la publicada en [117] a fin de arreglar un error que se ha encontrado en el polinomio utilizado por el segundo registro.

El estado interno del algoritmo DSC está compuesto por 81 bits divididos entre los 4 registros LFSRs y el bit de memoria del combinador de salida. Los tres registros principales, R1, R2 y R3, se utilizan para generar las entradas del combinador de salida. El último registro se utiliza exclusivamente para controlar el pulso de reloj irregular de los tres registros principales en cada ronda del algoritmo de cifrado. Los 4 registros aparecen detallados en la figura 2.23 y sus polinomios de retroalimentación se describen en la tabla 2.7.

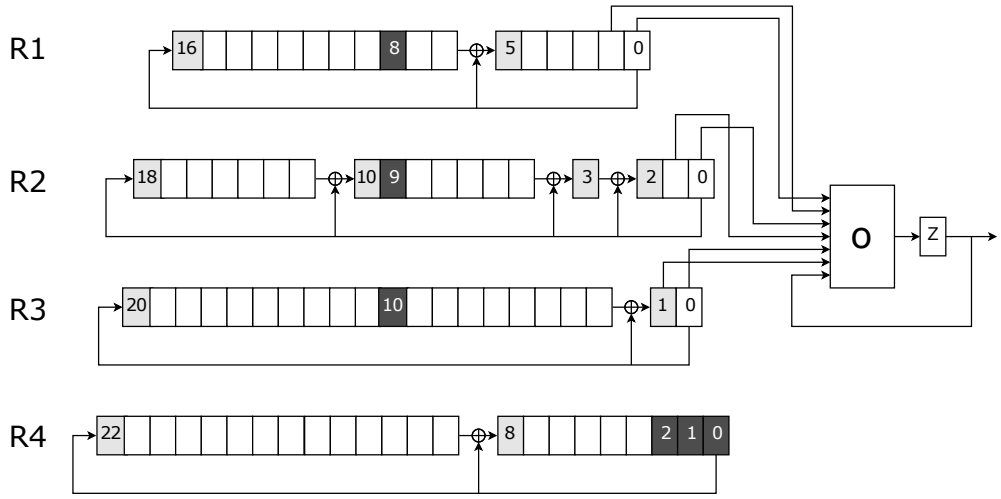


Figura 2.23: Esquema del cifrado DSC

LFSR	Longitud	Polinomio
R1	17 bits	$x^{17} + x^6 + 1$
R2	19 bits	$x^{19} + x^{11} + x^4 + x^3 + 1$
R3	21 bits	$x^{21} + x + 1$
R4	23 bits	$x^{23} + x^9 + 1$

Tabla 2.7: Descripción de los polinomios de los registros de desplazamiento con retroalimentación lineal utilizados en DSC

En el proceso de carga de la clave, se inicializa el estado interno del DSC cargando bit a bit los 64 bits del IV, seguidos de los 64 bits de la clave. Los 35 bits efectivos del IV son rellenados con ceros hasta llegar a 64 bits. Por lo tanto, la longitud de clave efectiva dentro de los 128 bits cargados, teniendo en cuenta el IV, es de 99 bits. Los 128 bits son introducidos bit a bit simultáneamente en R1, R2, R3 y R4 realizando una operación lógica XOR con el bit más significativo de cada registro y desplazando regularmente cada uno tras la introducción de cada bit. Por lo tanto, el proceso de carga de la clave tendrá una duración de 128 pulsos de reloj.

Una vez concluido, se ejecutan un total de 40 rondas donde los bits de *keystream* generados serán directamente descartados. Tras cada ronda, R4 recibirá regularmente 3 pulsos de reloj, mientras que los 3 registros principales sufrirán un desplazamiento irregular de 2 o 3 pulsos de reloj. Por cada registro, dicho número irregular de pulsos de reloj, ic_i, será determinado por ciertos bits específicos de los otros 3 registros incluyendo R4. Los bits de cada registro utilizados para el cálculo del desplazamiento irregular, se muestran con fondo gris en la figura 2.23.

Con el objetivo de definir formalmente los valores ic_i, se introduce la notación $x_{i,j}^{c_i}$ para referirse a j bits menos significativos, comenzando desde 0, del registro i una vez ha sido desplazando c_i pulsos de reloj. Mientras no exista ambigüedad se omitirán los valores c_i y se denominarán genéricamente desplazamientos de reloj o simplemente desplazamientos.

El número de veces que cada registro principal es desplazado en una ronda determinada, denotado ic_i, se define con las siguientes ecuaciones.

$$\begin{aligned} ic_1 &= 2 + (x_{4,0} \oplus x_{2,9} \oplus x_{3,10}) \\ ic_2 &= 2 + (x_{4,1} \oplus x_{1,8} \oplus x_{3,10}) \\ ic_3 &= 2 + (x_{4,2} \oplus x_{1,8} \oplus x_{2,9}) \end{aligned}$$

Una vez que el proceso de inicialización del DSC es completado, tras la ejecución de la carga de la clave y las posteriores 40 rondas, comienza la ronda número 0 del DSC, la cual dará lugar al primer bit del *keystream*, denominado

como z_0. En términos genéricos, el bit z_i del *keystream* será producido al final de la ronda i.

Todos los bits del *keystream* son producidos por el combinador de salida, $\mathcal{O}$, el cual consiste en una función cúbica no lineal que toma como entradas los 2 bits menos significativos de los 3 registros principales (R1, R2 y R3) junto con el contenido del bit de memoria que representa el bit generado en la ronda anterior. Los 6 bits de entrada tomados de los registros principales, son denominados el *estado* del DSC. Dicho estado depende del número de ronda de DSC al que se refiere así como del IV y la KEY.

En cada ronda del DSC, el combinador de salida genera un nuevo bit que será almacenado en el bit de memoria, denominado z en la Figura 2.23. En ese momento el bit que ocupa la memoria sale como nuevo bit del *keystream* y el nuevo bit generado por el combinador de salida ocupa su lugar. Dicho bit que ocupa ahora la memoria será utilizado como parte de las entradas del combinador de salida, en la nueva ronda del DSC.

La función del combinador de salida, donde $s = (x_{1,0}, x_{1,1}, x_{2,0}, x_{2,1}, x_{3,0}, x_{3,1})$ es el estado del DSC para la ronda actual y se define como

$$\begin{aligned}
\mathcal{O}(s,z) &= x_{1,1}x_{1,0}z \oplus x_{2,0}x_{1,1}x_{1,0} \oplus x_{1,1}z \\
&\oplus\ x_{2,1}x_{1,0}z \oplus x_{2,1}x_{2,0}x_{1,0} \oplus x_{3,0}z \\
&\oplus\ x_{3,0}x_{1,0}z \oplus x_{3,1} \oplus x_{3,0}x_{2,0}x_{1,0} \oplus x_{3,1}z \\
&\oplus\ x_{1,1}x_{1,0} \oplus x_{2,0}x_{1,1} \oplus x_{3,1}x_{1,0} \oplus x_{2,1}
\end{aligned}$$

El propósito principal de la función es romper la linealidad existente en los 3 registros principales. A simple vista, la función presenta una distribución uniforme en las probabilidades de salida, donde para 128 posibles combinaciones de entradas (6 bits del estado junto con el bit de memoria), la mitad de ellas producen un 0 y la otra mitad un 1.

Cualquier modificación del estado modificará el bit de salida de la función del combinador, un 50 % de las veces en media. Sin embargo, el bit de salida permanecerá invariante en un 56.25 % del tiempo en caso que los únicos bits del estado modificados sean pertenecientes a un único registro. Las probabili-

dades volverán a ser del 50 % en el momento en el que existan bits modificados en al menos 2 registros. Dicho hecho no fue constatado por el criptoanálisis propuesto por [117], por lo que su utilización permite la propuesta de un modelo teórico más preciso tal y como de detallará en las secciones posteriores.

Analizando experimentalmente el combinador de salida del DSC, se observa que la distribución de probabilidades en la modificación del bit de salida, es dependiente de los valores específicos de los 7 bits de entrada a la función.

Con el objetivo de facilitar la lectura del criptoanálisis que se describe en el resto de esta sección, se introduce la siguiente notación matemática.

$\mathcal{D}_{sc}(\text{KEY}, \text{IV})$, para $sc = (c_1, c_2, c_3)$, devuelve el estado del DSC, inicializado con KEY y IV, cuando los 3 registros principales reciban respectivamente c_1, c_2, y c_3 desplazamientos de reloj.

$\mathcal{D}_l(\text{KEY}, \text{IV})$ devuelve el estado del DSC, inicializado con KEY y IV, que ha sido utilizado para producir el bit de salida al final de la ronda l.

Se extiende también la operación *xor* para definir la operación lógica XOR entre 2 estados, donde la operación se aplique bit a bit entre ambos.

2.6.2. El ataque NTW y sus resultados

El ataque Nolh-Tews-Weinmann (NTW) es un ataque basado en texto plano conocido (*Known-Plaintext Attack*) capaz de recuperar la clave DSC de 64 bits de una forma más eficiente que un búsqueda exhaustiva sobre los 2^{64} posibles valores de clave. Como es el caso de los ataques basados en texto plano conocido, el ataque NTW asume que el atacante se encuentra en posesión de material de texto plano correspondiente a texto cifrado. En este caso concreto, el ataque requiere de la posesión de un conjunto de vectores de inicialización (IV) y sus correspondientes *keystream*. Como se ha explicado anteriormente, el texto cifrado en DSC es el resultado de realizar una operación lógica XOR entre el texto plano a cifrar y el *keystream* correspondiente a la ejecución del cifrado DSC sobre un IV y una clave determinados. Por lo tanto, conociendo el texto plano correspondiente a un texto cifrado, es posible realizar la operación inversa y obtener el *keystream* utilizado.

En la primera fase del ataque NTW, se determinan un conjunto de ecuaciones lineales afines que especifican relaciones entre un conjunto de bits de la clave. En una segunda parte del ataque los bits de la claves restantes (no derivados por las ecuaciones) son determinados siguiendo un procedimiento de búsqueda exhaustiva, con el objetivo de obtener la clave DSC de 64 bits correspondiente al texto cifrado y al IV.

Dada la linealidad de los 4 registros LFSR del DSC, cada uno de sus bits (80 en total) puede ser definido como una combinación lineal entre bits de la clave y bits del IV. El conocimiento de un bit $x_{i,j}^{c_i}$ determinado, permite obtener una ecuación lineal que se refiere al mismo. El objetivo del ataque NTW es obtener una porción del estado interno del DSC, para un rango determinado de desplazamientos de reloj de los 3 registros principales, de tal manera que se pueda obtener un número suficiente de ecuaciones lineales, n, como para poder recuperar la clave DSC de 64-bits siguiendo una búsqueda exhaustiva de los 2^{64-n} posibles valores de clave restantes.

Cada registro principal (R1, R2 y R3) del DSC tiene de media una probabilidad del 50 % de ser desplazado 2 veces en cada ronda (en lugar de 3). De hecho, el número de desplazamientos c_i para un registro determinado en la ronda l se define siguiendo una distribución binomial desplazada con modo $2{,}5l + 100$. Siguiendo esta fórmula es posible calcular el trío de desplazamientos de registro totales para los 3 registros principales, para una ronda determinada del DSC. Por ejemplo, en el caso de la primera ronda, el modo de la distribución sería 102.5, con lo que los tríos de desplazamientos de registro más probables para los 3 registros principales serían, cualquier combinación de 102 y 103, tal como 102-103-102.

Dada la linealidad exhibida por el algoritmo de cifrado DSC, la siguiente ecuación se verifica en todo momento para cualquier trío de desplazamientos de reloj, $sc = (c_1, c_2, c_3)$.

$$\mathcal{D}_{sc}(\text{KEY}, \text{IV}) = \mathcal{D}_{sc}(\text{KEY}, 0) \oplus \mathcal{D}_{sc}(0, \text{IV})$$

Dados los valores en sc correspondientes al número de desplazamientos de los registros principales tras la ronda l, la siguiente ecuación se verifica igual-

mente.

$$\mathcal{D}_{sc}(\text{KEY}, \text{IV}) = \mathcal{D}_{l}(\text{KEY}, \text{IV})$$

A pesar de que $\mathcal{D}_{sc}(\text{KEY}, \text{IV})$ no sea conocido, la siguiente ecuación también se verifica.

$$\mathcal{O}(\mathcal{D}_{sc}(\text{KEY}, \text{IV}), z_{l-1}) = z_l$$

Esta ecuación será denotada como $eqn(sc, l)$ en el resto del capítulo. Dado que el ataque NTW se encuentra basado en texto plano conocido, tanto los IVs como los bits del *keystream* (o cierta parte de ellos) son conocidos por el atacante.

Por lo tanto, es posible evaluar dicha ecuación $eqn(sc, l)$ con los 64 posibles valores del estado formados por los 6 bits procedentes de los 3 registros principales para una ronda DSC determinada, que da lugar a un bit conocido del *keystream*. El estado correcto habrá de estar en el subconjunto de estados $\mathcal{S}$ que satisface la ecuación $eqn(sc, l)$. Utilizando el IV correspondiente, es posible calcular $\tilde{s} = \mathcal{D}_{sc}(0, \text{IV})$. Por lo tanto, si sc es el estado correcto para dichos 6 bits de estado, $s = \mathcal{D}_{sc}(\text{KEY}, 0)$ habrá de encontrarse comprendido en el subconjunto $\tilde{\mathcal{S}} = \{\tilde{s} \oplus s^*; \forall\, s^* \in \mathcal{S}\}$.

Por otro lado, si sc no fuera el estado correcto de 6 bits correspondiente a la ronda l, el estado correcto aun tendría una posibilidad del 50 % de encontrarse en el subconjunto de estados que satisfacen la ecuación [2], según [117]. Por lo tanto, el estado correcto tendrá más de 50 % de probabilidad de encontrarse en dicho subconjunto, mientras que cualquier otro estado tendrá únicamente un 50 %.

Este experimento se puede entender como un proceso de Bernoulli donde el éxito es representado por la presencia del estado correcto dentro de la lista de estados candidatos. Si es repetido un número suficiente de veces, el estado más frecuente será por lo tanto el correspondiente a $\mathcal{D}_{sc}(\text{KEY}, 0)$.

[2]Este cálculo de probabilidades constituye una de las inexactitudes presentes en el ataque NTW, que será corregido dentro del criptoanálisis mejorado que se propone en esta tesis.

Siguiendo el procedimiento anterior, para un conjunto de IVs y *keystreams*, es posible determinar 6 bits de estado interno del DSC, el cual dará lugar a 6 ecuaciones lineares que se refieren a bits de la clave. En dicho escenario, dado que las ecuaciones son independientes, el segundo paso del ataque NTW requeriría realizar una búsqueda exhaustiva sobre 2^{58} posibles valores restantes, lo cual no sería factible en la práctica.

Con el objetivo de reducir el tiempo computacional requerido para este último paso de búsqueda exhaustiva, el ataque NTW extiende el principio anterior para cada combinación posible de desplazamientos de registros principales (35 en su artículo) teniendo en cuenta cierto rango de bits de *keystream* (hasta 19). Por cada trío de número de desplazamientos de registro, se genera una tabla de frecuencias para almacenar cada estado candidato potencial.

Una vez que todas las muestras (IV y *keystream*) son analizadas, el valor de cada variable (el bit de estado proveniente de un registro determinado para un número específico de desplazamientos) es determinado mediante las entradas de la tabla de frecuencias que involucran a ese bit específico. De esta manera es posible determinar 108 bits que potencialmente dan lugar a 108 ecuaciones. Un subconjunto de dichas ecuaciones es seleccionado en base al valor obtenido de acuerdo a un ranking y la viabilidad de resolución del sistema específico de ecuaciones (mas información en [117]). En efecto, ciertas ecuaciones son incompatibles dada la redundancia existente en los bits de estado solapados en desplazamientos consecutivos (e.g. $x_{20}^{i} = x_{21}^{i-1}$).

Una vez se han seleccionado suficientes ecuaciones (por ejemplo, 30), el resto de bits de la clave 2^{64-30} son determinados siguiendo una búsqueda exhaustiva.

Los autores de [117] han ejecutado el ataque descrito para diversa cantidad de texto plano (*keystreams*) tanto contra el canal *C* como el campo *B*. En sus resultados experimentales se han considerado diferente número total de ecuaciones, desde 10 hasta 40. A modo de comparación con el criptoanálisis mejorado que se propone a continuación, en las tablas 2.8 y 2.9 se presenta un resumen de los resultados del ataque NTW para el canal *C* y el campo *B* res-

Número ecuaciones	8192 *keystreams*	16384 *keystreams*	32768 *keystreams*
10 ecuaciones	2 %	30 %	96 %
20 ecuaciones	0 %	2 %	78 %
30 ecuaciones	0 %	1 %	48 %
40 ecuaciones	0 %	0 %	11 %

Tabla 2.8: Tasa de éxito del ataque sobre los datos del canal *C*

Número ecuaciones	16384 *keystreams*	32768 *keystreams*	65536 *keystreams*
10 ecuaciones	2 %	30 %	92 %
20 ecuaciones	0 %	2 %	65 %
30 ecuaciones	0 %	0 %	28 %
40 ecuaciones	0 %	0 %	4 %

Tabla 2.9: Tasa de éxito del ataque sobre los datos del campo *B*

pectivamente. Los detalles completos sobre los resultados obtenidos se pueden consultar en [117].

De los resultados se desprende que para alcanzar una probabilidad de éxito del 50 % contra el canal *C* de tal manera que se obtengan suficientes ecuaciones (digamos 30) como para permitir una búsqueda exhaustiva viable, es necesario utilizar al menos 32,768 *keystreams*.

En el caso del campo *B*, el ataque alcanza una probabilidad de éxito del 28 % para 30 ecuaciones tras utilizar aproximadamente 65,536 *keystreams*. En un artículo posterior [160] se presentan unos resultados ligeramente mejores utilizando un sistema alternativo de ranking. En el mejor de los casos, la mejora presentada eleva la probabilidad de éxito del 71 % al 90 % para 22 ecuaciones y 32,768 *keystreams*.

2.6.3. Modelo teórico para un criptoanálisis mejorado

El criptoanálisis mejorado que se propone se basa en el mismo principio que el ataque NTW utiliza para derivar el subconjunto probable de candidatos para un trío determinado de desplazamientos de registro en los registros principales. Sin embargo, en lugar de generar una tabla de frecuencia para cada ecuación $eqn(sc, l)$, se genera una tabla de frecuencias que contiene como entradas todas las posibles combinaciones de estado para un rango de desplazamientos de registro determinado de los registros principales. Cada número de desplazamientos para los 3 registros, se asocia a 6 bits del estado interno de DSC. Sin embargo, dado que desplazamientos consecutivos del mismo registro comparten uno de los bits, cada candidato de la tabla de frecuencias se encuentra compuesto por $eqn(sc, l)$ de estado interno de DSC. Todas las posibles combinaciones inválidas de estados, p.ej. $x_{20}^{i} \neq x_{21}^{i-1}$, son descartadas directamente en la generación inicial de estados posibles.

Por cada candidato de la tabla de frecuencias, correspondiente a un subconjunto del estado interno del DSC, se evalúan todas las ecuaciones $eqn(sc, l)$ relevantes para el rango de desplazamientos seleccionado. En la evaluación de una ecuación determinada, se incrementará la puntuación de aquellos estados de la tabla que validen dicha ecuación. El valor a incrementar se denomina peso y es determinado de forma proporcional a la probabilidad de que el estado correcto se encuentre en el subconjunto de estados que validen dicha ecuación, para un IV y *keystream* aleatorios. Una vez todas las muestras de texto plano disponible (parejas de IV y *keystream*) sean procesadas, los candidatos de la tabla se ordenan el base al valor obtenido.

Del candidato que recibe la puntuación más alta, se pueden derivar por tanto $3(len_c + 1)$ ecuaciones lineales. Utilizando dichas ecuaciones, es posible determinar ciertos bits de la clave DSC en base al valor de otros bits de la clave y del IV. En la última fase del ataque, el resto de bits, hasta 64, pueden ser determinados siguiendo una búsqueda exhaustiva.

La determinación de los pesos para cada ecuación $eqn(sc, l)$ del rango de desplazamientos de registro seleccionados, en la fase preliminar del ataque,

constituye un paso crucial. Únicamente las ecuaciones con un peso no nulo serán evaluadas en la siguiente fase del ataque.

A pesar de que dicha fase preliminar sea similar a la propuesta por el ataque NTW, existe una diferencia fundamental en el cálculo de los valores de los pesos. En los experimentos realizados, se ha constatado que los valores presentados por el ataque NTW no eran correctos, debido al comportamiento no homogéneo del combinador de salida del DSC, explicado anteriormente.

Con el objetivo de explicar matemáticamente dicho comportamiento y proponer un cálculo más preciso de los pesos, se propone la siguiente notación.

- Donde no exista ambigüedad no se mencionará KEY y IV de forma explícita;
- s_l se refiere a $\mathcal{D}_l(\text{KEY}, \text{IV})$;
- $s(tc_1, tc_2, tc_3)$ se refiere a $\mathcal{D}_{(tc_1,tc_2,tc_3)}(\text{KEY}, \text{IV})$;
- tc_i denota la hipótesis del número de desplazamientos reales del registro i, mientras que $c_{i,l}$ denota el número de desplazamientos reales del registro i en la ronda l;
- $\mathscr{P} = \{(IV, KS)\}$ denota el conjunto de muestras de texto plano, cada una de ellas compuesta de un *keystream* KS y su vector inicial IV asociado.

El peso de una ecuación determinada representa la probabilidad de que el combinador de salida produzca el mismo bit de *keystream* cuando utilice como entradas tanto s_l como $s(tc_1, tc_2, tc_3)$. En todo momento, la ecuación $eqn((tc_1, tc_2, tc_3), l)$ se considera correcta si:

$$\mathcal{O}(s_l, z_{l-1}) = \mathcal{O}\big(s(tc_1, tc_2, tc_3), z_{l-1}\big).$$

Dicha ecuación será correcta con una probabilidad del 100 % si los registros principales han sido desplazados tc_1, tc_2, y tc_3 pulsos de reloj respectivamente para la ronda l. En este caso, el estado correcto habrá necesariamente de estar en el subconjunto de estados candidatos que verifiquen la ecuación.

La probabilidad de que un registro i se desplace exactamente tc_i pulsos de reloj en la ronda l, es:

$$Pr\left[c_{i,l} = tc_i\right] = \binom{40+l}{tc_i - (80+2l)} 2^{-(40+l)}.$$

La probabilidad, denotada como p_1, de que los 3 registros principales se desplacen tc_1, tc_2, y tc_3 pulsos de reloj respectivamente para la ronda l, se define como:

$$p_1 = \prod_{i=1}^{3} Pr\left[c_{i,l} = tc_i\right]$$

En el caso en el que el número de desplazamientos de registro asumido para uno de los registros difiera del número de desplazamientos reales que dicho registro ha sufrido para una ronda determinada, cabría esperar, según el modelo teórico presentado por [117], que la probabilidad de que el bit de salida del combinador fuese el mismo sea del 50 %. En dicho caso, la probabilidad de que el estado correcto estuviese en el subconjunto de candidatos que validan la ecuación sería la siguiente:

$$p = p_1,1 + (1 - p_1)\frac{1}{2}.$$

Sin embargo, debido al comportamiento peculiar que exhibe el combinador de salida, descrito anteriormente en 2.6.1, en el caso en que el número de desplazamientos determinados para los registros principales difiera del real, la probabilidad de que la ecuación sea correcta para el bit de salida real, no es exactamente del 50 %.

En efecto, si la determinación del número de desplazamientos es correcta para 2 registros principales, e incorrecta para el otro (sean cuales sean), la ecuación será correcta con una probabilidad de un 56.25 %. Dicho hecho ha sido constatado experimentalmente en los experimentos realizados.

Por lo tanto, basado en este hecho, la probabilidad global para una ecuación determinada puede ser redefinida de la siguiente manera. Considerando 2 probabilidades, intermedias:

La probabilidad p_2 de la determinación del número de desplazamientos sea incorrecta únicamente para uno de los registros principales, se puede definir como:

$$p_2 = \sum_{i=1}^{3} \left(1 - Pr\left[c_{i,l} = tc_i\right]\right) \prod_{\substack{j \neq i; \\ j=1}}^{3} Pr\left[c_{j,l} = tc_j\right].$$

La probabilidad p_3 de que la determinación del número de desplazamientos sea incorrecta para dos registros principales, se puede definir como:

$$\begin{aligned} p_3 &= \sum_{i=1}^{3} Pr\left[c_{i,l} = tc_i\right] \prod_{\substack{j \neq i; \\ j=1}}^{3} \left(1 - Pr\left[c_{j,l} = tc_j\right]\right) \\ &+ \prod_{k=1}^{3} \left(1 - Pr\left[c_{k,l} = tc_k\right]\right). \end{aligned}$$

Por lo tanto, la probabilidad de que una ecuación determinada sea válida puede ser expresada de la siguiente manera:

$$Pr[eqn(sc, l)] = p_1 + 0,5625 * p_2 + 0,5 * p_3$$

Por ejemplo, la ecuación $((102, 102, 102), 1)$ correspondiente a la ronda 41 responsable del cálculo del bit z_1 del *keystream*, tiene una probabilidad de ser correcta del 50,338 %. Siguiendo la aproximación de [99], el peso w_{eqn} que se asociará a dicha ecuación es calculado como la función logarítmica de la probabilidad de la misma, de la siguiente manera:

$$w(eqn(sc, l)) = log\left(\frac{Pr[eqn]}{1 - Pr[eqn]}\right).$$

Siguiendo los cálculos y metodología descrita, se procede al cálculo de probabilidades y pesos para todas las ecuaciones posibles dentro del rango seleccionado de números de desplazamientos de registro y bits disponibles de *keystream*.

Una vez las ecuaciones y sus pesos han sido determinadas en la fase preliminar del ataque, se da paso a la siguiente fase, consistente en la determinación de los estados más probables de $\mathcal{D}_{sc}(\text{KEY}, 0)$ para el rango máximo posible de desplazamientos de reloj $\mathcal{R}$.

Para ello, se genera una tabla de frecuencias $\mathscr{T}$ que contiene todas las combinaciones posibles de bits de estado para el rango seleccionado de desplazamientos. Tal y como se ha descrito anteriormente, dicha tabla contiene $3(len_c + 1)$ estados diferentes resultantes de las diversas combinaciones. Cada entrada de la tabla almacena la puntuación que representa la probabilidad total acumulada para dicho candidato concreto, calculada mediante la suma de todos los pesos asociados a las ecuaciones que resultan válidas para el mismo.

Por cada trío de desplazamientos de reloj $sc \in \mathcal{R}^3$ y todas las rondas $l \in \mathcal{K}$ para las cuales el bit del *keystream* es conocido, se evalúan las ecuaciones $eqn(sc, l)$ contra todos los posibles estados candidatos almacenados en la tabla de frecuencias.

Los estados que verifiquen una ecuación determinada serán aquellos que potencialmente correspondan a $\tilde{s} = \mathcal{D}_{sc}(\text{KEY}, IV)$. La parte del Vector de Inicialización IV del estado puede ser eliminada debido a la linealidad del DSC, ya que $s^* = \mathcal{D}_{sc}(0, IV)$. Por lo tanto, los candidatos que reciban el paso de una ecuación serán aquellos que $s = \tilde{s} \oplus s^*$.

Esta primera fase del ataque se puede resumir en el pseudo-código 2.24.

Tal y como se describe en el algoritmo, las ecuaciones se evalúan para todos los estados candidatos y todos los bit de texto plano (*keystream*) disponibles. Al igual que ocurre en el ataque NTW, cuanto más número de *keystreams* se encuentren disponibles, mayores serán las probabilidades de que el estado correcto se encuentre en el subconjunto de aquellos con puntuaciones mayores tras el análisis.

Por cada estado candidato de la tabla $\mathscr{T}$ se puede derivar un sistema de $(3 \times len_c + 3)$ ecuaciones lineales con 64 incógnitas. Si las determinaciones sobre los bits de estado son correctas, la clave real DSC utilizada para generar todas las muestras analizadas, habrá de verificar dicho sistemas de ecuacio-

Algoritmo 2: Selección del mejor candidato en el criptoanálisis del cifrado DSC

for $sc \in \mathcal{R}^3$ **do**
  for $l \in \mathcal{K}$ **do**
    for $\tilde{s} \in \mathscr{T}$ **do**
      for $(IV, KS) \in \mathscr{P}$ **do**
        if $\mathcal{O}(\tilde{s}, KS[l-1]) == KS[l]$ **then**
          $s \leftarrow \tilde{s} \oplus \mathcal{D}_{sc}(0, IV)$;
          $\mathscr{T}[s] \leftarrow_{+} w(eqn(sc, l))$;

Figura 2.24: Selección del mejor candidato en el criptoanálisis del cifrado DSC

nes.

A modo de facilitar la utilización del sistema de ecuaciones en la fase final de búsqueda exhaustiva, se aplica la reducción Gausiana sobre el mismo.

Asumiendo ecuaciones independientes [3], $(3 \times len_c + 3)$ bits de la clave pueden ser determinados en base al valor del resto de bits explorados durante la búsqueda exhaustiva.

En la fase final del ataque, todas las combinaciones posibles para el resto de bits son exploradas siguiendo una búsqueda exhaustiva. Por cada combinación posible, se determinarán el resto del bits de la clave candidata utilizando el sistema de ecuaciones y se ejecutará el algoritmo DSC utilizando dicha clave junto con un Vector de Inicialización (IV) asociado a una de las muestras disponibles de texto plano (*keystream*). En el proceso de cifrado se generan 64-bits de *keystream* que son comparados con los 64-bits de *keystream* disponibles para dicho IV.

En el caso de que ambas secuencias de 64 bits sean idénticas, la clave DSC candidata será la correcta con una abrumadora probabilidad del $(1 - 2^{-64}) \approx 100\,\%$. Una verificación posterior de dicha clave sobre otros IVs y *keystreams*

[3] Las ecuaciones serán independientes siempre que el número máximo de desplazamientos utilizado en la primera fase no supere la longitud máximo del registro menor

disponibles, elevará dicha probabilidad al 100 %.

2.6.4. Implementación del criptoanálisis

La implementación práctica del ataque descrito en la sección anterior presenta cierta problemática en lo referente a la eficiencia del proceso. El parámetro que más influye en la eficiencia es la longitud del rango de desplazamientos de los registros, denotado como len_c.

En efecto, len_c determinará, tanto la cantidad de ecuaciones n_{eq} a ser evaluadas durante el proceso, como el tamaño de la tabla de candidatos $\mathscr{T}$. Ambos valores tienen una influencia cúbica sobre la eficiencia del proceso de criptoanálisis, según queda definido por la siguiente expresión.

$$\begin{aligned} n_{eq} &= len_c^3.len_k \\ |\mathscr{T}| &= 2^{3(len_c+1)} \end{aligned}$$

La eficiencia puede ser mejorada en cierta medida ejecutando en paralelo la evaluación de las ecuaciones para los candidatos de $\mathscr{T}$. Sin embargo, la gran cantidad de ecuaciones existentes para un rango grande, limita en gran medida la implementación práctica del ataque utilizando hardware estándar.

Por ejemplo, para un rango de desplazamientos de registros de 7 unidades, utilizando una implementación eficiente que se beneficie de un alto nivel de paralelismo en las operaciones, la primera fase del ataque sobre una clave puede durar varias horas. En dicho escenario, únicamente se podrían derivar 20 ecuaciones de la clave, existiendo la necesidad de realizar, en la segunda fase, una búsqueda exhaustiva sobre los bits restantes, lo cual duraría hasta varios días dependiendo del hardware utilizado.

A modo de posibilitar la implementación práctica del ataque en un tiempo razonable con el objetivo de obtener resultados experimentales fiables, se propone la división de la primera fase del criptoanálisis en una estructura de rangos de desplazamientos, donde éstos sean evaluados individualmente de forma jerárquica.

El rango de desplazamientos será dividido en varios subrangos y la primera fase del ataque será aplicada a cada uno de ellos de forma individual, utilizando para ello las ecuaciones relevantes para las combinaciones de desplazamientos pertenecientes a dicho rango. Como resultado se obtiene una tabla de candidatos probables para cada rango.

En una siguiente fase, anterior a la búsqueda exhaustiva, los candidatos más probables de cada rango serán combinados entre ellos, para volver a someterlos a evaluación utilizando las ecuaciones relevantes al nuevo rango. El proceso será repetido de forma iterativa. A pesar de que esta aproximación incrementa la eficiencia de la implementación, también disminuye la probabilidad de éxito, dada la menor cantidad de ecuaciones que son utilizadas para realizar la selección inicial de los candidatos más probables en cada tabla.

La utilización de solapamientos en la definición de los rangos de desplazamientos permite ajustar el balance entre la mejora de eficiencia y la pérdida de eficacia en el ataque. En efecto, mucho solapamiento entre rangos incrementará el número de rangos requerido, lo cual derivará en una mayor pérdida de eficacia. En ese sentido, la utilización de rangos consecutivos de desplazamientos de registros supone un buen compromiso entre eficacia e eficiencia en la implementación del ataque.

Por ejemplo, un rango global de desplazamientos de registros de 12 unidades (102-114) será dividido en 4 subrangos de 3 unidades cada uno. Tal y como se ha explicado anteriormente, en una fase inicial se aplica a cada rango el análisis detallado en la sección 2.6.3, con el objetivo de generar 4 tablas con la probabilidad de todos los candidatos.

A continuación se toman los candidatos más probables de las 2 primeras tablas (p.ej. los 200 primeros) para el primer y segundo rango respectivamente, y se genera una tabla combinada resultante del producto cartesiano de sus miembros. La probabilidad asignada a cada candidato se fija inicialmente con su valor anterior asociado. En la creación del producto cartesiano, se eliminan estados candidatos imposibles, fácilmente detectables dado el solapamiento de 3 bits en estados pertenecientes a números de desplazamientos de registros

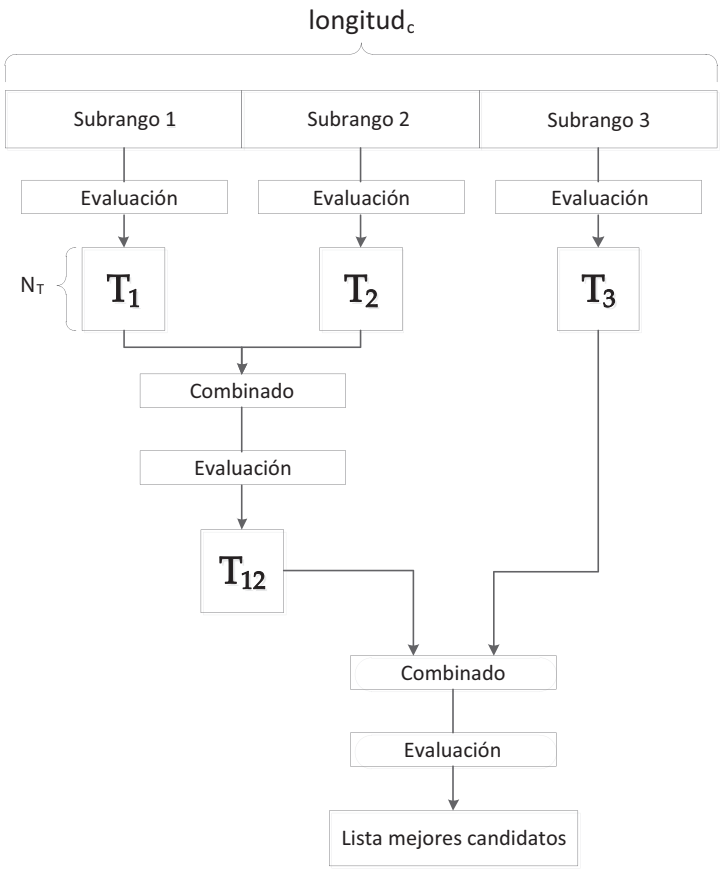


Figura 2.25: Descripción de la implementación del criptoanálisis

consecutivos.

Dado el tamaño N_T de la lista de candidatos más probables de 2 tablas, el proceso combinatorio de 2 tablas generará, de media, una tabla con $0{,}5 \times N_T^2$ candidatos. Este coeficiente representa la reducción de tamaño derivada del descarte de los candidatos imposibles de suceder dado el solapamiento de los bits de estado.

Una vez la tabla combinada de ambos rangos es generada, se procede a evaluar contra sus candidatos, todas aquellas ecuaciones pertenecientes al nuevo rango extendido, que no hayan sido evaluadas anteriormente. Efectivamente, no resulta necesario re-evaluar las ecuaciones ya utilizadas en la fase anterior, ya que su contribución al cálculo de la probabilidad de cada estado candidato, se encuentra ya comprendida en el valor inicial asignado a cada uno.

Una nueva tabla reducida N_R será generada con los candidatos más probables de la tabla combinada. El proceso es repetido iterativamente, tal y como muestra la figura 2.25 con los rangos restantes hasta obtener suficientes ecuaciones para garantizar la viabilidad de la fase final de búsqueda exhaustiva.

El pseudo-código de la figura 2.24 muestra de forma global el algoritmo de la implementación del ataque. Dicho algoritmo es capaz de incrementar la eficiencia global del ataque reduciendo el tiempo de cómputo requerido gracias al descarte temprano de los estados menos prometedores.

Los parámetros N_T y N_R tienen un impacto directo tanto en el rendimiento (eficiencia) como en la probabilidad final de éxito del ataque (eficacia). Su modificación permite seleccionar de antemano la cantidad de recursos computaciones y tiempo disponible, a fin de obtener la mayor eficacia posible dentro de dichas limitaciones. Si ambos son suficientemente grandes, se obtiene una eficacia similar a la que se obtendría sin utilizar división de rangos, tal y como el ataque era descrito en la sección anterior.

2.6.5. Análisis de resultados

Con el objetivo de obtener resultados experimentales, se ha desarrollado la implementación descrita en la sección 2.6.4, utilizando sub-rangos de 4 desplazamientos de longitud. La implementación ha sido programada en Python y C bajo GNU/Linux.

Dado que el ataque presentado es del tipo texto plano conocido, para todos los experimentos realizados a continuación se ha asumido conocimiento de 8 bits de *keystream* por parte del atacante.

En esta sección se muestran los resultados obtenidos tanto para el ataque contra el canal *C*, utilizando el rango de desplazamientos 102-114, como el campo *B*, utilizando el rango 202-214.

- Canal *C*: sub-rangos 102-104, 105-107, 108-110 y 111-113
- Campo *B*: sub-rangos 202-204, 205-207, 208-210 y 211-213

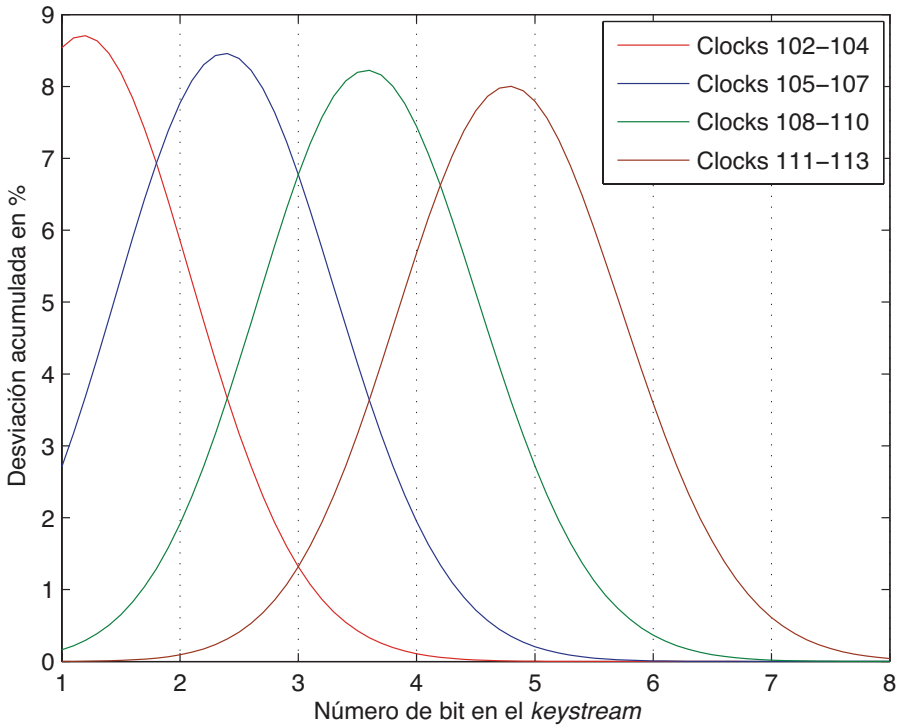


Figura 2.26: Sesgo acumulado para rangos de 3 pulsos de reloj en el canal C empezando en 102

Primeramente se han determinado, tanto de forma teórica como experimental, las ecuaciones relevantes para el rango seleccionado junto con sus pesos correspondientes.

Las figuras 2.26 y 2.27 muestran la distribución de la probabilidad acumulada para todas las ecuaciones relevantes para el estado correspondiente al número de desplazamientos de cada uno de los rangos de 3 unidades, para el canal *C* y campo *B* respectivamente.

Tal y como se ha descrito en la sección 2.6.3, el peso final de la ecuación es calculado de forma logarítmica en base a la probabilidad que la ecuación tiene de ser correcta para un IV y clave aleatorias.

Como se ha mencionado en la sección 2.6.4, el número N_T de candidatos más probables para las tablas de los rangos individuales, es un factor decisivo para grado de eficacia del ataque. Cuando más pequeño sea el tamaño mayor será la eficiencia del ataque, dado el menor número de candidatos existentes

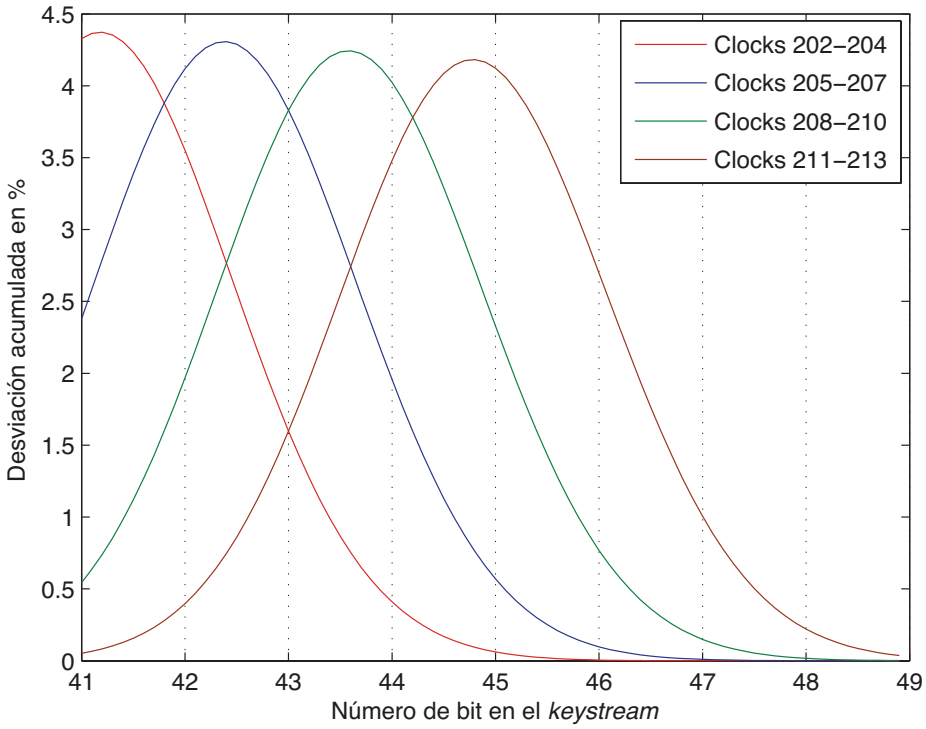


Figura 2.27: Sesgo acumulado para rangos de 3 pulsos de reloj en el campo B empezando en 202

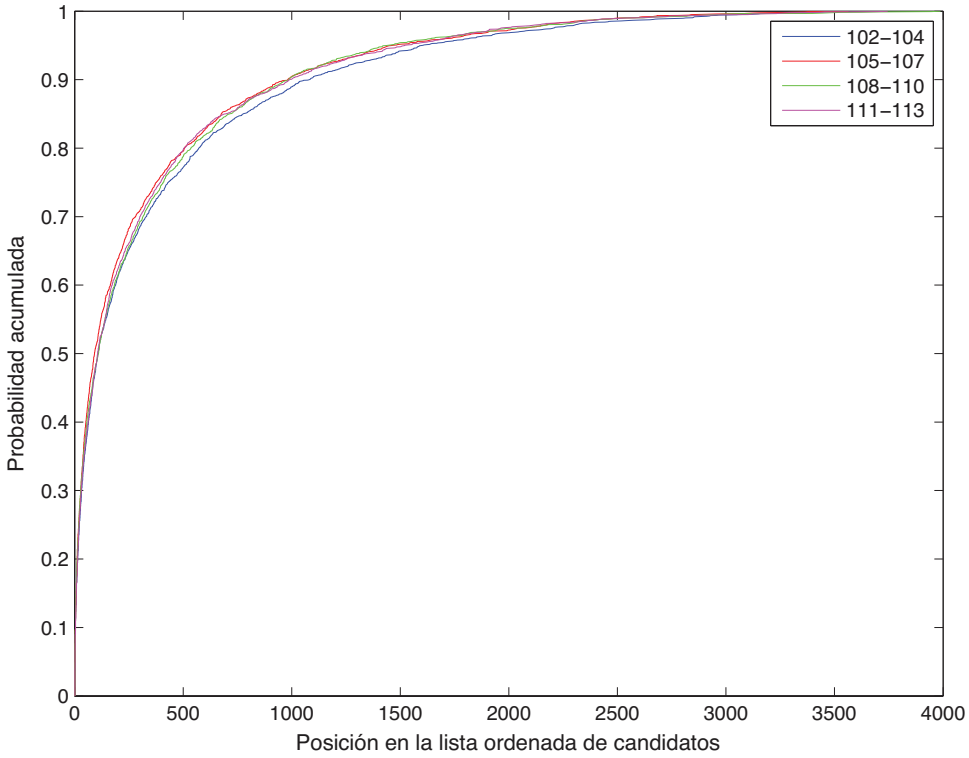


Figura 2.28: Distribución de la probabilidad de que el candidato correcto se encuentre en la tabla reducida para los diferentes subrangos en el rango de desplazamientos de reloj 102-113 para 4096 *keystreams*

en las tablas combinadas, pero menor será su eficacia.

En efecto, la probabilidad de que el estado correcto se encuentre en la lista de candidatos más probables, es dependiente del tamaño de la misma. Dicha probabilidad es, a su vez, dependiente de la cantidad de *keystreams* disponibles para el criptoanálisis, tal y como se explica en el modelo teórico presentado en la sección 2.6.3.

Se han llevado a cabo una serie de experimentos para evaluar la evolución de la probabilidad de éxito en el análisis de los sub-rangos, para diferentes cantidades de texto plano disponible y para los escenarios de ataque al canal *C* y campo *B*.

Los resultados experimentales para el canal *C* se muestran en las figuras 2.28, 2.29 y 2.30, para 4096, 8192 y 16384 *keystreams* respectivamente.

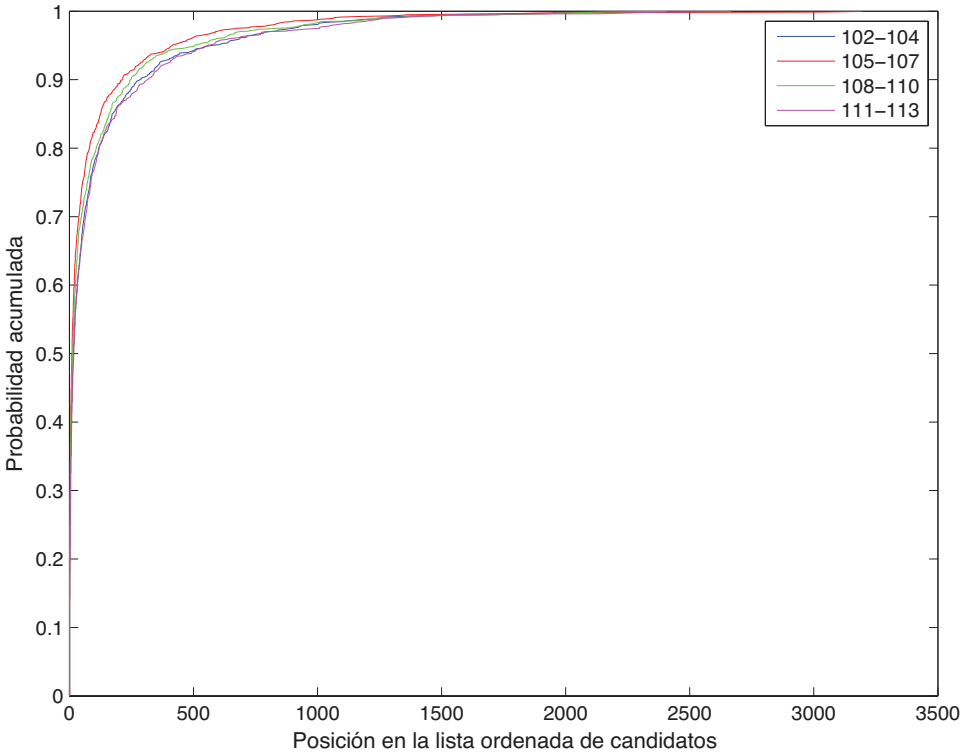


Figura 2.29: Distribución de la probabilidad de que el candidato correcto se encuentre en la tabla reducida para los diferentes subrangos en el rango de desplazamientos de reloj 102-113 para 8192 *keystreams*

La figura 2.31 presenta los resultados obtenidos atacando el campo B con 16384 *keystreams* disponibles.

Se puede observar en las figuras que un incremento en la cantidad de *keystreams* disponibles para el criptoanálisis redunda en un claro incremento en las probabilidades de que el estado correcto se encuentre entre los determinados como más probables. A pesar de que dicha cantidad de texto plano disponible tiene una influencia lineal negativa sobre la eficiencia del ataque, en los experimentos realizados la diferencia es de tan solo algunos minutos para una clave determinada.

Como se ha comentado anteriormente, el parámetro que tiene mayor influencia sobre la eficiencia, es el tamaño de la lista de candidatos más probables para ser utilizada en la creación de la tabla combinada de candidatos en

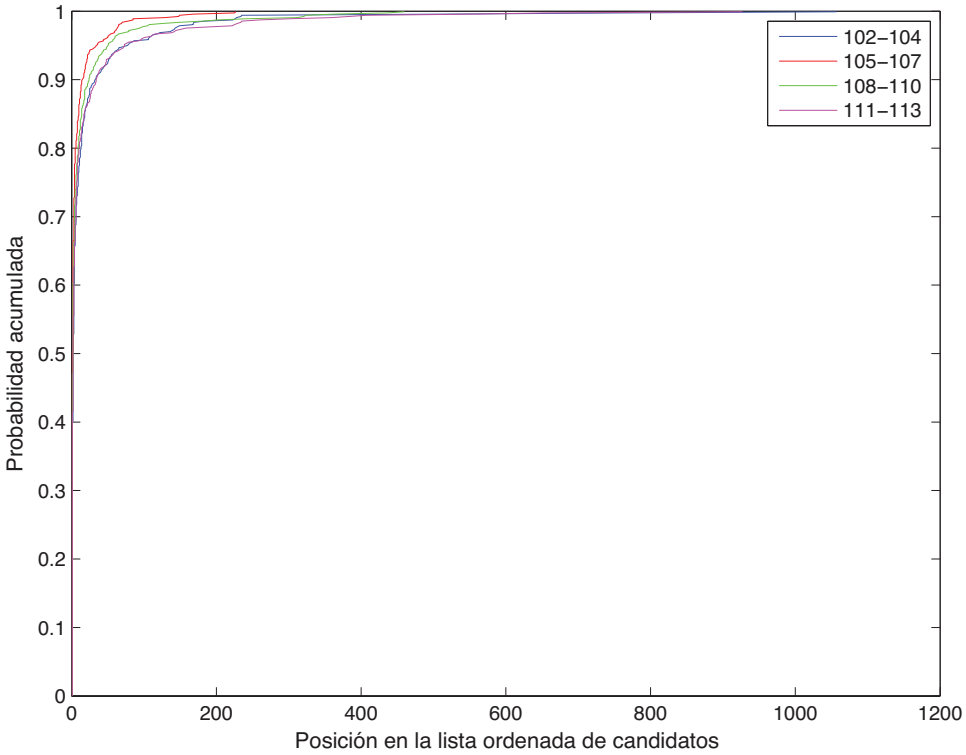


Figura 2.30: Distribución de la probabilidad de que el candidato correcto se encuentre en la tabla reducida para los diferentes subrangos en el rango de desplazamientos de reloj 102-113 para 16384 *keystreams*

el rango extendido. En los resultados presentados en las figuras se observa la evolución de la probabilidad de éxito sobre el tamaño de la lista de candidatos.

En las pruebas experimentales se confirma la importancia de las ecuaciones adicionales derivadas del rango extendido de desplazamientos en la evaluación de las tablas combinadas. En efecto, aunque el estado correcto no se encuentre entre los primeros dentro de la lista reducida de candidatos probables para una tabla individual, es muy probable que sea el candidato seleccionado, una vez el ataque haya concluido tras la evaluación de todas las tablas combinadas.

Por el contrario, si el candidato correcto no se encuentra en la lista de candidatos probables de alguno de los 4 rangos individuales, será imposible que el ataque sea exitoso, ya que dicho candidato no existirá en la tabla combinada

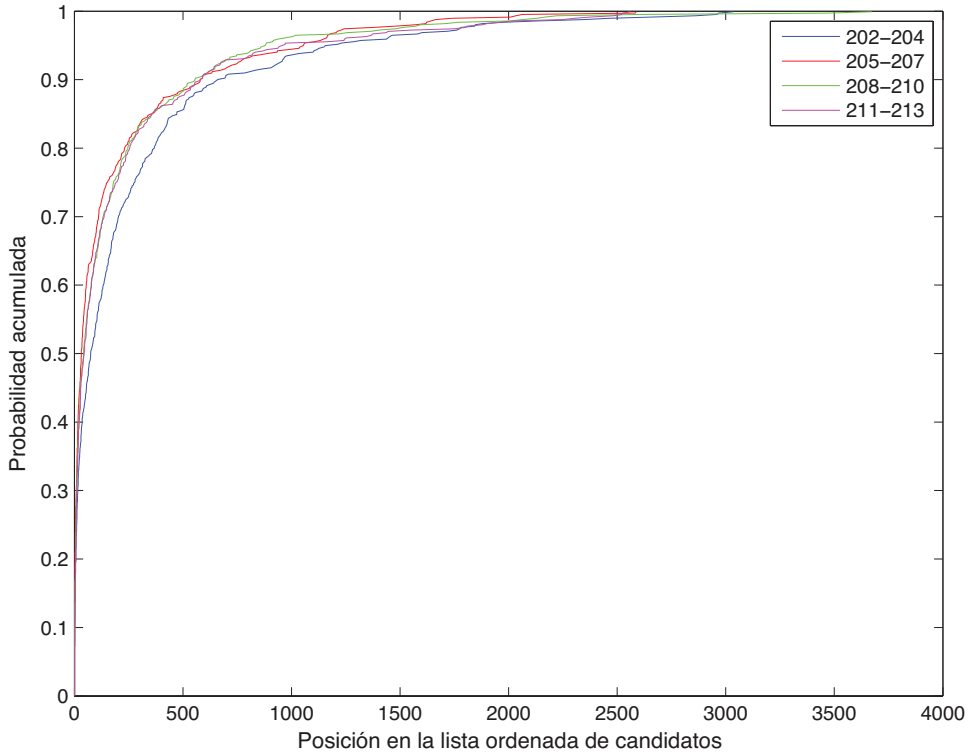


Figura 2.31: Distribución de la probabilidad de que el candidato correcto se encuentre en la tabla reducida para los diferentes subrangos en el rango de desplazamientos de reloj 202-213 para 16384 *keystreams*

correspondiente. Este hecho es útil para el cálculo previo de las probabilidades mínimas previstas de éxito en base a los parámetros N_T y N_R.

Asumiendo conocimiento sobre los primeros 8 bits de *keystream*, se ha llevado a cabo el ataque completo contra el canal C para diversas cantidades de *keystream*, utilizando el rango global de desplazamientos de registros 102-113 (incluido).

Se ha dividido dicho rango en 4 subrangos de combinaciones de longitud 3 y se han utilizado los valores 200 y 50 para los parámetros N_T y N_R. Dichos valores para el tamaño de la lista de candidatos más probables, han sido determinados experimentalmente y permiten realizar el ataque completo es cuestión de minutos.

Los resultados experimentales del ataque, respecto a la probabilidad de

Número ecuaciones	4096 *keystreams*	8192 *keystreams*	16384 *keystreams*
9 ecuaciones	35 %	85 %	98 %
21 ecuaciones	16 %	73 %	97 %
33 ecuaciones	6 %	**55 %**	95 %
39 ecuaciones	2 %	33 %	84 %

Tabla 2.10: Tasa de éxito del ataque contra los datos del canal *C* para el rango [102,113]

encontrar la clave correcta, se presentan en la tabla 2.10 para el rango de desplazamientos de registros 102-113 y diferentes cantidades de texto plano disponible.

Siguiendo la fórmula presentada en la sección 2.6.3, para el rango completo 102-113 el número de ecuaciones de la clave derivadas es de 39.

Durante la experimentación se ha observado que, cuando el ataque no es exitoso, a menudo resulta ser debido al fallo en la determinación de unos pocos bits del estado candidato. Dichos bits tienen a concentrarse en la periferia del rango de desplazamientos, en nuestro caso 3 bits de estado de 102 y 3 bits de 113.

Dicho hecho se explica por la menor cantidad de ecuaciones que contribuyen en dichos número de desplazamientos, tal y como se puede también observar en la figura 2.26.

Por lo tanto, se propone ignorar dichos bits, y utilizar los 33 bits restantes con los que se pueden derivar 33 ecuaciones que relacionan linealmente bits de la clave con bits del IV. En dicho escenario, la última fase de búsqueda exhaustiva de la clave necesita únicamente explorar 2^{31} combinaciones lo que lleva tan solo unos segundos en la implementación realizada en C para CPU y OpenCL para GPUs.

Considerando 33 ecuaciones, utilizando 8192 *keystreams* la probabilidad de éxito es ligeramente mayor del 50 %, tal y como se observa en la tabla 2.10. En este caso, la búsqueda exhaustiva de la clave explorando las 2^{31} posibles

Número ecuaciones	8192 *keystreams*	16384 *keystreams*	32768 *keystreams*
9 ecuaciones	19 %	69 %	100 %
21 ecuaciones	10 %	57 %	98 %
33 ecuaciones	3 %	**36 %**	88 %
39 ecuaciones	1 %	21 %	69 %

Tabla 2.11: Tasa de éxito del ataque contra los datos del campo *B* para el rango [202,213]

combinaciones es capaz de recuperar la clave de 64-bit correcta con una probabilidad del 100 % en unos 15 segundos.

Se han realizado experimentos similares atacando el campo *B*, asumiendo conocimiento por parte de un atacante sobre los primeros 8 bits del *keystream*, bits 41 a 49, utilizados para cifrarlo. El rango de desplazamientos utilizado, relevante para dichos bits, es de 202 a 213.

Tal y como se observa en la Figura 2.11, la distribución de probabilidad acumulada por las ecuaciones relevantes para dicho rango, es sustancialmente inferior que en el caso del ataque al canal *C*, cuya distribución se mostraba en la Figura 2.10.

Por lo tanto, se deberá utilizar un número mayor de *keystreams* para obtener resultados comparables a los presentados para el campo *C*. En los diversos experimentos realizados se han utilizado subrangos de combinaciones de longitud 3 junto con los valores 200 y 50 para los parámetros N_T y N_R.

La tabla 2.11 muestra los resultados experimentales para diversas cantidades de *keystream* disponibles, desde 8192 hasta 32384. Como se puede observar, utilizando 16384 *keystreams* se obtiene una probabilidad de 36 % de obtener la clave correcta tras el proceso del criptoanálisis.

2.6.6. Comparación de resultados con el ataque NTW

Una de las principales limitaciones para la implementación práctica del ataque NTW era la dificultad de obtención de suficiente material de texto plano,

en forma de *keystreams*, necesario para llevar a cabo el ataque son una probabilidad de éxito aceptable. En efecto, en el escenario de un ataque al campo *C* de la comunicación de voz DECT, el ataque NTW requería de al menos 2^{15} *keystreams* para alcanzar una probabilidad de éxito del 50 % tras una búsqueda exhaustiva sobre 2^{34} posibles valores de la clave (al haber derivado el ataque 30 ecuaciones lineales).

Así mismo, se ha confirmado en las pruebas de laboratorio la existencia, en muchas implementaciones, de un flujo constante de 5 tramas DECT con campos *C* con contenido predecible durante una conversación. En dicho escenario, el ataque NTW requeriría de 1 hora y 49 minutos de conversación para poder recopilar suficientes *keystreams* (2^{15}), como para alcanzar una probabilidad del 50 % de recuperar la clave tras el proceso del criptoanálisis.

En el mismo escenario, los resultados experimentales presentados en la sección 2.6.5 demuestran que el criptoanálisis presentado en esta tesis es capaz de recuperar la clave DSC tras únicamente 20 minutos de conversación y con un nivel de confianza similar del 50 %.

Los resultados experimentales atacando el campo *B* también presentados en la sección 2.6.5, son de la misma manera consistentes, presentando un nivel de eficacia similar al obtenido por el ataque NTW para el mismo escenario, pero requiriendo 4 veces menos cantidad de *keystreams* para llevarlo a cabo. En el mismo escenario descrito por los autores del ataque NTW [117], asumiendo que los primeros 8 bits del *keystream* utilizado para cifrar el campo *B* son predecibles, el método de criptoanálisis presentado en esta tesis sería capaz de recuperar la clave con una probabilidad de éxito del 30 %, tras únicamente 3 minutos. Con 6 minutos de conversación, la probabilidad de éxito se elevaría al 95 %.

2.7. Conclusiones

De la investigación teórica y experimental realizada en el presente capítulo sobre la seguridad y privacidad de las comunicaciones inalámbricas DECT,

se desprende que el estándar y las implementaciones existentes en el mercado no ofrecen suficiente garantía respecto a la seguridad y privacidad de las comunicaciones personales.

Queda patente que la disponibilidad de dispositivos SDR de bajo coste redunda en un incremento del riesgo a la seguridad y privacidad de las comunicaciones DECT, al haberse demostrado la capacidad de éstos para la interceptación de comunicaciones de voz efectuadas sobre el protocolo DECT. El hardware requerido para la interceptación pasiva de comunicaciones DECT ha dejado de ser una barrera de entrada, dada la viabilidad de llevar a cabo ataques incluso con aquellos dispositivos SDR de muy bajo coste, como el RTL-SDR con un valor de mercado de unos 20€. La investigación realizada y sus resultados han sido publicados en [137].

Incluso en el caso en que las comunicaciones DECT utilicen cifrado, se ha demostrado que existen vulnerabilidades importantes inherentes al diseño del algoritmo de cifrado y el protocolo de emparejamiento criptográfico, que pueden ser utilizadas para la ruptura de la clave en escenarios de utilización reales, posibilitando la interceptación pasiva de comunicaciones cifradas.

Se ha descrito teóricamente y demostrado experimentalmente cómo el protocolo de emparejamiento criptográfico de dispositivos es vulnerable ante ataques de interceptación pasivos, donde un atacante pueda derivar la clave permanente UAK para derivar la futuras claves DCK y descifrar comunicaciones cifradas. Existen varios escenarios posibles donde dicho ataque podría llevarse a cabo, como acceso físico al dispositivo o en el escenario de las Comunicaciones Unificadas en entornos domésticos. Los resultados de la investigación realizada en la sección 2.5 han sido publicados en [34].

En aquellos casos donde el usuario no utilice el emparejamiento de dispositivos DECT, o éste se realice de forma segura, por ejemplo realizando el emparejamiento dentro de una cámara de Fáraday, se ha demostrado la viabilidad de criptoanalizar el protocolo de cifrado para derivar la clave DCK utilizada para el cifrado de la comunicación DECT. En comparación con el mejor ataque descrito en la literatura, el criptoanálisis desarrollado en este capítulo es 4

veces más efectivo, posibilitando su utilización en escenarios reales para la interceptación de comunicaciones DECT cifradas. El criptoanálisis desarrollado y sus resultados se describen en [35].

Capítulo 3

GSM

El presente capítulo de la tesis tiene una naturaleza eminentemente experimental orientada a investigar el grado efectivo de protección de la seguridad de las comunicaciones existente actualmente en las comunicaciones GSM, siguiendo los objetivos y metodología general descrita en las secciones 1.1 y 1.2 del capítulo 1.

Primeramente se procede a analizar el funcionamiento del protocolo GSM, en lo que concierne a los objetivos de la presente tesis, así como el estado del arte en lo referente a las vulnerabilidades y ataques conocidos a la privacidad de las comunicaciones. Posteriormente, se realizan una serie de experimentos dirigidos a determinar el riesgo para la privacidad de las comunicaciones que suponen los nuevos dispositivos SDR de bajo coste en combinación con ataques prácticos contra los protocolos criptográficos utilizados en GSM.

A este respecto, se explora primero la capacidad que dichos dispositivos poseen para la interceptación pasiva y activa de comunicaciones GSM, realizando diversos experimentos que simulen situaciones reales de utilización por parte de usuarios. El objetivo de dichos experimentos es contribuir a la validación de la hipótesis de la tesis y los objetivos de la misma, evaluando el grado efectivo de protección de la privacidad que ofrece el protocolo GSM, mediante la evaluación de la viabilidad de llevar a cabo ataques efectivos con un nivel limitado de recursos.

Finalmente, se analiza en detalle el algoritmo de cifrado A5/1, utilizado

por la práctica totalidad de operadores europeos para el cifrado de las comunicaciones, y se demuestra experimentalmente la aplicación práctica de un ataque de compromiso-espacio-tiempo documentado en la literatura. Se describe y demuestra experimentalmente como dicho ataque puede ser utilizado en situaciones de despliegue reales para la ruptura de las claves de cifrado y la interceptación pasiva de comunicaciones GSM.

3.1. La arquitectura GSM

El protocolo GSM, acrónimo de *Global System for Mobile communication*, es un protocolo veterano, diseñado a finales de la década de los 80, que actualmente cuenta con un total estimado de 7000 millones subscriptores a nivel global, alcanzando el 95 % de la población mundial [81].

En comparación con su antecesora, la red GSM actual, conocida como de segunda generación, emplea comunicaciones completamente digitales para voz y datos, con el objetivo de realizar un uso más eficiente del espectro de radiofrecuencia y ofrecer una mayor protección de las comunicaciones personales mediante la utilización de cifrado.

Las nuevas redes de telefonía, conocidas como de tercera (3G) y cuarta generación (4G), ofrecen conexión de banda ancha móvil a Internet y actualmente coexisten con la red GSM, la cual ofrece una mayor cobertura. La casi totalidad de los terminales del mercado compatibles con la red 3G soportan también la red GSM, conmutando entre una y otra de forma transparente al usuario según la cobertura existente en cada momento.

En ocasiones, se utiliza la expresión *red celular* para referirse a la red GSM. En efecto, con el objetivo de ofrecer una gran cobertura geográfica, la red GSM divide el terreno en células, donde cada célula se compone por una torre GSM (BTS) capaz de cubrir un área de hasta 35 kilómetros. La arquitectura GSM se encuentra diseñada para coordinar dichas células y permitir una comunicación transparente entre los teléfonos móviles asociados a ellas, incluyendo la transición de una célula a otra.

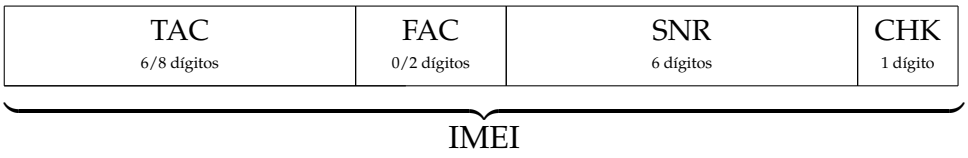


Figura 3.1: Estructura y composición del IMEI

El teléfono GSM, ya bien sea un teléfono clásico como un Nokia 3110 o uno inteligente como *iphone*, se denomina *Mobile Equipment*, o ME, y permite realizar y recibir llamadas en una red de telefonía GSM. A diferencia de la primera generación de telefonía móvil, en GSM el terminal ME no integra ninguna identidad de suscripción y es independiente del operador de telefonía móvil[1].

Cada ME contiene un identificador único de 15 dígitos denominado IMEI que identifica de forma única dicho terminal a nivel global. El IMEI se compone de los campos TAC, FAC, SNR y CHK, tal y como se detalla en la figura 3.2. El campo TAC indica el modelo concreto de terminal y tiene una longitud de 8 dígitos. En los IMEI anteriores a 2004, los últimos 2 dígitos del campo TAC eran denominados FAC e indicaban el código del certificado de conformidad nacional para dicho modelo. El TAC del IMEI permite identificar el país donde dicho dispositivo ha sido fabricado. El campo SNR representa el número de serie único para dicho modelo fabricado. Por último, el dígito denominado como CHK en la figura 3.1, representa el código de verificación para los 14 dígitos anteriores, calculado mediante el algoritmo *Luhn*.

La identidad de usuario en GSM se obtiene de la tarjeta SIM que se introduce en un ME determinado. Dicha tarjeta SIM es, de hecho, una tarjeta inteligente identificada por un número único denominado ICCID. La SIM es básicamente un microcontrolador con memoria interna que realiza las funciones de HSM y contiene el identificador único para un contrato de suscripción con un operador de telefonía móvil determinado. Dicho identificador, denominado IMSI y cuya composición se detalla en la figura 3.2, posee una longitud

[1]En la práctica, cuando el terminal es en cierta medida subvencionado por un operador determinado, es posible que venga bloqueado para permitir únicamente dicho operador

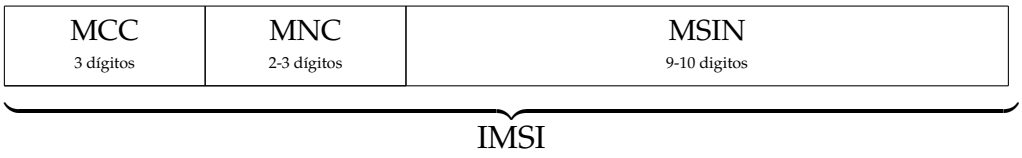


Figura 3.2: Estructura y composición del IMSI

máxima de 15 dígitos y se compone de los campos MCC, MNC y MSIN.

Los campos MCC y MNC representan respectivamente los código de país y operador de telefonía móvil nacional, representando en su conjunto a un operador móvil concreto. Por su parte, el campo MSIN contiene el identificador único del contrato de suscripción a dicho operador de telefonía móvil.

La tarjeta SIM también contiene la clave secreta, denominada K_i, utilizada para los procesos de autenticación y cifrado que serán detallados más adelante en las secciones 3.3.1 y 3.3.3. La SIM se encuentra equipada con una pequeña memoria utilizada para el almacenamiento de mensajes de texto y contactos.

Con el objetivo de ofrecer cierto nivel de protección contra el posible fraude derivado del robo o pérdida de la tarjeta, la SIM ofrece la posibilidad de autenticación local mediante un código PIN de 4 dígitos o un código PUK de mayor longitud.

En esencia, la tarjeta SIM representa la identidad de un usuario en la red GSM y dicha identidad se encuentra ligada a un número de teléfono ofrecido por el operador de telefonía móvil para dicho código IMSI en particular. Cada usuario puede tener diversas identidades en uno o más operadores GSM, poseyendo varias tarjetas SIM en uno o varios ME. Cada identidad se encontrará identificada por su código IMSI y tendrá asignada un número de teléfono dentro de la red GSM.

Al conjunto de equipamiento móvil (ME) y tarjeta SIM, se le denomina Estación Móvil o MS. Una MS se comunica con una o más torres GSM denominadas Estaciones Base o BTS, mediante el protocolo de radio frecuencia GSM, el cual será analizado posteriormente en detalle en la sección 3.2.

Cada BTS se identifica con un número de 14 dígitos denominado CGI, com-

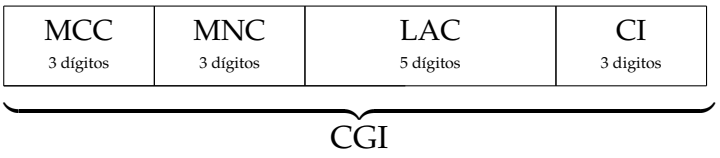


Figura 3.3: Estructura y composición de Identificador Global de Célula (CGI)

puesto del identificador del área donde la BTS se encuentra localizada, denominado LAI, y el número de identificación de la célula, denominado CI. A su vez, el LAI se encuentra compuesto de los campos MCC, MNC y LAC, tal y como se detalla en la figura 3.3.

La BTS se encuentra conectada a un Controlador de Estación Base, denominado BSC. El BSC supervisa y coordina una serie de estaciones base conectándolas con el Sistema de Conmutación Central, denominado NSS. Al conjunto de BTS y BSC se le denomina Subsistema de Estación Base o BSS.

El NSS representa el centro neurálgico de la red GSM de un operador, interconectando todos los BSS entre ellos y la red de telefonía pública y es el encargado de ofrecer todos los servicios a los usuarios de la red GSM. El NSS se encuentra formado por los siguientes componentes:

MSC El Centro de Conmutación Móvil es el componente principal al que se conectan todos los BSS. En colaboración con VLR y el GMSC, se encarga del establecimiento y encaminamiento de las llamadas, junto con la adquisición de datos para la facturación.

GMSC La Puerta de Enlace del Centro de Conmutación Móvil es la encargada de encaminar las llamadas hacia y desde la red de telefonía pública denominada PSTN.

HLR El Registro de Localización Local se encarga de coordinar los diferentes VLRs y almacenar toda la información relativa a los contratos de suscripción de los usuarios de la red GSM. Gracias al HLR es posible relacionar los códigos IMSI con sus respectivos números de teléfono asociados.

VLR El Registro de Localización de Visitantes almacena toda la información

de las SIMs de los usuarios que se encuentran en una área determinada. Entre otras funciones, el VLR será el encargado de relacionar los identificadores temporales TMSI con los identificadores permanentes IMSI. Dicho mecanismo se describe en detalle en la sección 3.3.

AuC El Centro de Autenticación es una pieza fundamental de los procesos de autenticación y cifrado utilizados en la red GSM. El AuC almacena la clave secreta K_i de todos los IMSIs registrados en el sistema, siendo también el encargado de la ejecución de los algoritmos A3/A8 utilizados por los procesos de cifrado y autenticación que serán descritos en detalle en la sección 3.3.

EIR El Registro de Identificadores de Equipamiento contiene todos los IMEI de los MS suscritos a la red GSM. Como parte del registro en la red GSM, los ME transmiten el IMSI, que es gestionado por el HLR, y el IMEI, que es autenticado y almacenado en el EIR. Uno de los objetivos del EIR es poder denegar el acceso a la red GSM en base al IMEI, posibilitando la generación de una lista negra, por ejemplo para ser usada para el bloqueo de teléfonos robados.

3.2. El protocolo de radio GSM

El protocolo GSM de ETSI es un protocolo complejo cuya descripción detallada comprende decenas de documentos y una totalidad de miles de páginas. A esta complejidad se le añade la escasez de herramientas de bajo coste para realizar investigación experimental, junto con ciertas complejidades legales relativas al hecho de que las frecuencias de GSM se encuentren reguladas.

Para los propósitos de esta tesis, en la presente sección se analizará el protocolo de radio GSM, denominado como la interfaz *Um*, encargado de la comunicación por radiofrecuencia entre los teléfonos móviles (MS) y las estaciones base (BTS). GSM opera en frecuencias reguladas, donde en diferentes áreas geográficas se ha reservado porciones del espectro de radio para uso exclusivo

Banda	Frec. subida (Mhz)	Frec. bajada (Mhz)	ARFCN
GSM-450	450.6 - 457.6	460.6 - 467.6	259 - 293
GSM-480	479.0 - 486.0	489.0 - 496.0	306 - 340
GSM-850	824.2 - 849.2	869.2 - 894.2	128 – 251
GSM-900	890.0 - 915.0	935.0 - 960.0	1 - 124
EGSM-900	880.0 - 915.0	925.0 - 960.0	975 – 1023, 0-124
GSM-1800	1710.2 - 1784.8	1805.2 - 1879.8	512 - 885
GSM-1900	1850.2 - 1909.8	1930.2 - 1989.8	512 - 810

Tabla 3.1: Bandas de frecuencias asignadas al protocolo de radio GSM

de la telefonía móvil GSM.

La tabla 3.1 muestra los diferentes rangos de frecuencia asignados al protocolo GSM en las diferentes áreas geográficas a nivel mundial. En Europa, las bandas utilizadas son EGSM-900, con el rango de frecuencias 880 - 915 Mhz y 925 - 960 Mhz, y GSM-1800 con el rango de 1710.2 – 1784.8 MHz y 1805.2 – 1879.8 MHz.

El protocolo de radio GSM es dúplex completo, de tal manera que se utilizan frecuencias separadas para el envío y la recepción. Las comunicación en la dirección $MS \rightarrow BTS$ se denomina enlace de subida, o *uplink*, y la comunicación $BTS \rightarrow MS$ se denomina enlace de bajada, o *downlink*. Por lo tanto, cada banda reservada a GSM define rangos diferentes de frecuencias para los enlaces de subida y bajada, tal y como se aprecia en la figura 3.1.

GSM emplea tanto Acceso Múltiple por División de Frecuencia (FDMA) como Acceso Múltiple por División de Tiempo (TDMA). Cada banda GSM se divide, siguiendo FDMA, en diferentes canales de 200 kHz de ancho de banda cada uno, denominados ARFCN. Dentro de la terminología GSM, los valores de los ARFCN son absolutos, por lo que, dado un valor determinado para un ARFCN, no existirá ambigüedad respecto a la banda GSM a la que pertenece. Por ejemplo, el ARFCN 2 pertenece a la banda GSM-900, como se puede apreciar en la tabla 3.1.

La frecuencia exacta de un ARFCN determinado puede calcularse de la siguiente forma, siendo S y B el inicio de la frecuencia correspondiente a la banda para los enlaces de subida y bajada respectivamente.

$$\begin{aligned} F_{Subida} &= S + 0{,}2 \times ARFCN \\ F_{Bajada} &= B + 0{,}2 \times ARFCN \end{aligned}$$

Cada ARFCN se divide a su vez en varios canales lógicos mediante la aplicación de TDMA creando 8 ranuras de tiempo. Cada teléfono móvil (ME) conectado a la red queda asignado a una de dichas ranuras, por lo que el ME simplemente ignorará las otras 7. Por lo tanto, el ME únicamente recibirá o transmitirá $\frac{1}{8}$ del tiempo. Con el objetivo de facilitar la implementación del TDMA en los terminales móviles, se sigue la Transmisión en dos sentidos por División de Tiempo (TDD), por lo que existe un desplazamiento de 3 ranuras temporales entre los canales de subida y bajada, de tal manera que el terminal no se encuentre obligado a implementar dúplex completo para la transmisión y recepción simultánea.

En GSM, al igual que en otras implementaciones de TDMA como por ejemplo DECT, la transmisión dentro de las mencionadas ranuras temporales da lugar a lo que se conoce como ráfagas o *bursts*. Cada ráfaga acomoda una trama GSM, con una duración de 4615 milisegundos, la cual es el resultado de la modulación en GMSK de un conjunto de bits que componen el paquete transmitido.

Las tramas se agrupan en *multi-tramas*, las cuales pueden ser de 2 tipos, *multi-tramas* de tráfico o *multi-tramas* de control. Las *multi-tramas* de tráfico contienen un total de 26 tramas cada una y las de control contienen 51 tramas. A su vez, ambos tipos de *multi-trama* se agrupan en *super-tramas* de control y tráfico, con un tamaño de 26 y 51 *multi-tramas* respectivamente. Por lo tanto, la cantidad de tramas contenidas en ambos tipos de *supertramas* será idéntica (51×26 tramas). Por último, las *super-tramas* se agrupan en *hiper-tramas* de 2048 super-tramas cada una.

Una *hiper-trama* contiene por tanto un total de 2,715,648 ($26 \times 51 \times 2048$) tramas, con una duración de 3 horas 28 minutos 53 segundos y 760 milise-

gundos. El número de una trama determinada respecto a su localización en la *hiper-trama* que la contiene, es utilizado como Vector de Inicialización para el algoritmo de cifrado, tal y como se describe más detalladamente en la sección 3.3.3. Por lo tanto, el IV se repetirá a intervalos regulares correspondientes con la duración de la *hiper-trama*.

Con el objetivo de mejorar el rendimiento de la transmisión de voz, respecto a efectos de propagación impredecibles que puedan afectar a ciertas frecuencias, GSM soporta el protocolo conocido como Salto Lento de Frecuencia o SFH. Mediante SFH, cada trama de tráfico durante una llamada de voz puede ser transmitida en una frecuencia diferente. De esta manera, y dado que la codificación de voz en GSM implementa redundancia, se mejora la calidad de transmisión ante la presencia de interferencias o problemas de transmisión en determinadas frecuencias.

El salto de frecuencias es completamente opcional para la BTS, pero si ésta lo solicitara, el ME deberá implementarlo en la comunicación. A pesar de que no fuera diseñado con tal propósito, el SFH en GSM ha resultado ser un problema para ataques de interceptación pasivos, tal y como se demuestra más adelante en la sección 3.7.

En cada una de las 8 ranuras temporales de un ARFCN determinado, se pueden transmitir tramas pertenecientes a uno de los diferentes canales lógicos[2] definidos en GSM. Dichos canales se agrupan en dos grandes tipos, canales de tráfico y canales de control.

Los canales de tráfico, denominados TCH, se encargan principalmente de la transmisión de la voz digitalizada y los datos de los usuarios. A su vez, se dividen en dos tipos, *Full Rate* (TCH/F) con una velocidad de 22.8 Kbps o *Half Rate* (TCH/H) a 11.4 Kbps. Como se ha descrito anteriormente, las tramas pertenecientes a canales de tráficos se agrupan, de 26 en 26, en multi-frames de tráfico los cuales, a su vez, se agrupan, de 51 en 51, en super-frames del mismo tipo. Los canales TCH/H utilizarán únicamente una ranura temporal cada 2

[2]Los canales lógicos en GSM no deben ser confundidos con los 8 *canales lógicos* generados en cada ARFCN a nivel de transmisión tal y como se ha explicado previamente.

tramas.

Por otra parte, los canales de control se encargan de las tareas relacionadas con la gestión de las llamadas y conexiones. Existen 10 tipos de canales de control divididos en 3 categorías, Canales de Difusión (BCH), Canales Comunes de Control (CCCH) y Canales de Control Dedicados (DCCH).

Los Canales de Difusión son los encargados de transmitir a los ME información relativa a la BTS y a la red GSM del operador. En estos canales no existe enlace de subida, son únicamente de bajada. Dentro de esta categoría podemos encontrar los siguientes tipos:

- Canales de Control de Difusión (BCCH), encargados de transmitir los parámetros públicos de configuración relativos a la BTS, en particular el MCC, MNC, LAC y CI, descritos en la sección 3.1.
- Canales de sincronización (SCH), donde se difunde información sobre el número de trama actual, de modo que los ME puedan sincronizarse con la BTS.
- Canales de Corrección de Frecuencia (FCCH), donde se transmite la frecuencia exacta en la que opera la BTS, a fin de que los ME puedan ajustarse con exactitud a la misma.

Por su parte, los Canales de Comunes de Control se encargan principalmente del envío y recepción de información de control para la gestión de identidades, conexiones y llamadas. Existen los siguientes tipos de Canales Comunes de Control.

- Canales de Llamada (PCH), también conocidos como Canales de *Paging*, los cuales sustentan el mecanismo principal para la gestión de identidades en GSM, en lo relativo a llamadas y conexiones entrantes. Su funcionamiento se describe en detalle posteriormente en la demostración de ataques contra la privacidad de GSM en la secciones 3.5 y 3.7.
- Canales de Acceso Aleatorio (RACH), son los encargados de iniciar acciones en la red a petición del MS, como por ejemplo el establecimiento

de una llamada saliente.

- Canales de Adjudicación de Acceso (AGCH), que actúan como paso intermedio en el establecimiento de llamadas o conexiones provenientes de los canales RACH o PCH, con el objetivo de acordar un nuevo canal de control para la comunicación.

Por último, los Canales de Control Dedicado se dividen en los siguientes tipos.

- Canales Dedicados de Control (SDCCH), son los encargados del paso final en el establecimiento de llamadas o conexiones, identificando y autenticando al MS, así como inicializando el mecanismo criptográfico que cifrará la comunicación, si la hubiera. También son los encargados de asignar un canal de tráfico para la comunicación. Los SMS también puede ser enviados directamente por el SDCCH.

- Canales Lentos Asociados de Control (SACCH), en colaboración con un SDCCH o un TCH, son los encargados, entre otras funciones, de la transmisión de información relativa a la gestión de potencia de la señal y las frecuencias de BTS vecinas. También son capaces de transmitir SMS simultáneamente a una llamada en curso.

- Canales Rápidos Asociados de Control (FACCH), a expensas de ranuras temporalmente utilizadas por un canal de tráfico, capaces de enviar mensajes urgentes de control, tales como desconexiones de llamadas.

3.3. Mecanismos de seguridad y privacidad en GSM

3.3.1. Autenticación de Equipos Móviles

La autenticación del Equipo Móvil (MS) por parte de una red GSM, es uno de los mecanismos de seguridad más importantes, ya que gracias a dicha autenticación se evita el fraude telefónico. En efecto, sin la autenticación del MS

por parte del operador GSM, sería posible para un atacante realizar llamadas a cargo de un tercero.

La tarjeta SIM de un Equipo Móvil y el Centro de Autenticación AuC, juegan un papel vital dentro del proceso de autenticación del ME. Cuando el MSC requiere la autenticación de un ME determinado, solicita primeramente al VLR los valores de autenticación relativos a su IMSI o TMSI. El VLR solicita al AuC los valores de autenticación para dicho ME, conocidos como el trío de autenticación, los cuales son calculados de la siguiente forma:

$$\begin{aligned} RAND &= Random() \\ SRES &= A_3(K_i, RAND) \\ K_c &= A_8(K_i, RAND) \end{aligned}$$

RAND es un número aleatorio de 128-bits. SRES tiene una longitud de 32-bits, representa la respuesta al desafío de autenticación y es generado mediante el algoritmo de cifrado A_3 en base a RAND y a la clave secreta K_i de 128-bits asociada a dicho IMSI/TMSI. K_c es la clave simétrica de sesión de 64-bits resultante de aplicar el algoritmo A8 a K_i y $RAND$.

Armado con esta información, el MSC procede a solicitar al ME la autenticación enviándole el número aleatorio RAND generado, junto con un Número de Secuencia de Clave de Cifrado de 3 bits de longitud denominado CKSN. El ME solicita a la tarjeta SIM el cálculo de SRES y K_c en base al RAND recibido. En efecto, la tarjeta SIM posee la clave K_i de 128-bits asociada al IMSI e implementa los algoritmos A3 y A8, por lo que es capaz de calcular SRES y K_c de la siguiente manera:

$$\begin{aligned} SRES &= A_3(K_i, RAND) \\ K_c &= A_8(K_i, RAND) \end{aligned}$$

Utilizando los valores SRES y K_c, el ME puede completar la autenticación enviado el valor SRES calculado de vuelta al MSC, el cual verificará que concuerda con el recibido del UaC. En caso positivo, el MSC enviará a la BTS correspondiente el valor de K_c calculado por el UaC, concluyendo satisfactoriamente la autenticación del ME. Llegado a este punto, tanto el ME como la

BTS en la que se encuentra registrado, comparten una clave K_c común con la que se podrán cifrar las comunicaciones de radio.

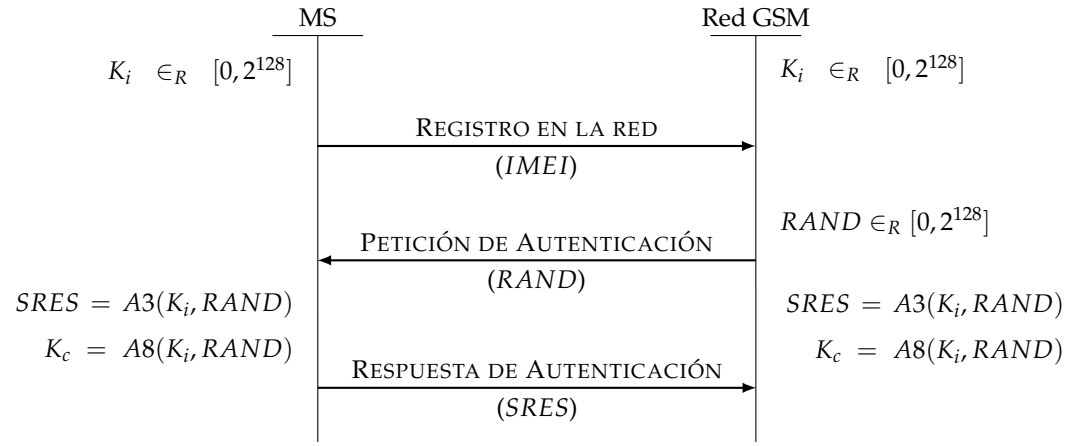


Figura 3.4: Autenticación del MS en GSM

La figura 3.4 presenta un esquema del proceso de autenticación del ME en la red GSM. Como se puede apreciar, la tarjeta SIM del ME actúa como un HSM, computando los algoritmos A_3 y A_8 internamente. La clave K_i de 128-bits asociada al IMSI se encuentra almacenada en la tarjeta, pero nunca sale de ella, ya que es utilizada solo internamente para la ejecución de los algoritmos A_3 y A_8. De hecho, en el sistema GSM, la clave K_i únicamente reside dentro de la memoria interna de las tarjetas SIM y en el Centro de Autenticación.

Dado que, tanto las claves K_i como los algoritmos A3 y A8 se encuentran únicamente bajo el control del operador de telefonía, no es necesario que exista interoperabilidad en lo referente a su implementación específica. En efecto, siempre que el número y tamaño de los parámetros de entrada y salida se encuentren conformes con el estándar, la implementación interna no supondría un problema de interoperabilidad, ya que el operador de telefonía controlaría tanto la implementación existente en sus tarjetas SIM, como la efectuada por el UaC. Sin embargo, en la práctica, la casi totalidad de operadores siguen la sugerencia de ETSI y usan el algoritmo COMP128 en alguna de sus versiones [154].

COMP128 es una implementación de los algoritmos A3 y A8 definida en el estándar GSM. A pesar de que dicha implementación se encontrara disponible bajo acuerdos de confidencialidad, en [22] los autores la derivaron por ingeniería inversa y publicaron una implementación de referencia del algoritmo. Dicha implementación sería conocida como COMP128-1, de primera generación, y posteriormente sucedida por COMP128-2, COMP128-3 y COMP128-4. En la sección 3.4 se analizan las vulnerabilidades encontradas en las diferentes versiones de COMP128.

3.3.2. Privacidad de las identidades móviles

El código IMSI, como se ha descrito anteriormente, identifica de forma única al suscriptor de telefonía móvil. Utilizando dicho código IMSI, un atacante, capaz de interceptar pasivamente el intercambio de paquetes GSM entre los MS y las BTS, sería capaz de localizar y seguir los movimientos de los usuarios GSM. En efecto, el código IMSI es permanente y se encuentra relacionado con un número de teléfono, por lo que sería factible para un atacante encontrar el IMSI asociado a un número de teléfono determinado y seguir los movimientos geográficos del poseedor de dicho número, utilizando el IMSI como si de un GPS se tratase.

La privacidad de las identidades de los suscriptores móviles era uno de los objetivos del estándar, motivo por el cual se diseñó un mecanismo de anonimización basado en el uso de identidades temporales mediante códigos TMSI.

El TMSI es el código de Identidad Temporal del Suscriptor Móvil que es asignado por el VLR a los suscriptores móviles durante el proceso de registro en la red GSM. El MS envía el código IMSI y, como parte del proceso de registro, recibe un código TMSI de naturaleza temporal. Desde ese momento, el direccionamiento a dicho MS se realizará utilizando su código TMSI en lugar del IMSI.

Dado que la relación TMSI-IMSI es almacenada en el VLR, el TMSI solo será válido en un área determinada, por lo que tendrá que ser actualizado cuando MS cambie a un área diferente. En cualquier momento, la red GSM

podrá actualizar el TMSI a uno nuevo. En la práctica, el tiempo de vida el código TMSI varía de un operador a otro.

3.3.3. Cifrado de comunicaciones

Con el objetivo de proteger la confidencialidad de las comunicaciones GSM, la interfaz de radio entre el MS y la BTS puede ser cifrada con un algoritmo denominado A5.

Al contrario que los algoritmos A3 y A8, A5 es implementado en el ME, en lugar de en la tarjeta SIM, y su implementación debe garantizar la interoperabilidad entre los diferentes ME del mercado y las BTS utilizadas por los operadores de GSM.

A5 utiliza una clave secreta de 64-bits y un Vector de Inicialización de 22 bits para generar 228 bits de *keystream*. En su utilización en GSM, el valor K_c derivado con el algoritmo A8, tal y como se ha descrito en la sección 3.3.1, será utilizado como clave secreta de 64-bits. El Vector de Inicialización de 22-bit será determinado por el número de trama GSM a cifrar o descifrar. De esta manera, el algoritmo A5 deberá ser ejecutado una vez por cada nueva trama GSM.

Los 228-bit de *keystream* producidos por la ejecución de A5 se dividen en 2 mitades de 114-bits cada una. La primera mitad de *keystream* será utilizada para cifrar o descifrar las tramas GSM enviadas por la BTS al MS, realizando una operación lógica XOR bit a bit entre el *keystream* y el texto plano o cifrado. El cifrado y descifrado de las tramas enviadas por el MS a la BTS, se realizará realizando la misma operación con los últimos 114 bits del *keystream*.

Existen 4 variantes del algoritmo A5 definidas en el estándar GSM. La primera de ellas, denominada como A5/0, es en realidad la ausencia de cifrado. En efecto, el estándar GSM permite las comunicaciones en texto plano. Es posible, por tanto, que una red GSM determinada no ofrezca ningún tipo de cifrado, hecho documentado en algunos países.

El primer algoritmo de cifrado definido dentro de la familia A5, es el A5/1. A5/1 es el algoritmo de cifrado por excelencia en GSM, siendo implementado

por la práctica totalidad de los operadores europeos, que lo utilizan como único algoritmo de cifrado en sus redes GSM. El algoritmo A5/1 será analizado en detalle posteriormente en la sección 3.7.2.

El algoritmo A5/2, cuya estructura interna similar a A5/1 será analizada posteriormente en la sección 3.7.2, es una variante debilitada del A5/1 creada con el objetivo de permitir su exportación fuera de los límites de la Unión Europea. Efectivamente, tal y como será analizado en detalle en la sección 3.4, el algoritmo A5/2 es extremadamente débil, siendo vulnerable a diferentes tipos de criptoanálisis documentados en la literatura. Desde hace unos años[3], las entidades de estandarización ETSI y 3GPP han iniciado el proceso de retirada del algoritmo A5/2 para los nuevos terminales (ME) y las redes GSM existentes.

Por último, el algoritmo A5/3, también llamado KASUMI, es el más seguro de las variantes A5. A5/3 fue diseñado por Mitsubishi Electric y se encuentra basado en el algoritmo MISTY1. En GSM, A5/3 es utilizado para la transmisión de datos en GPRS/EDGE, donde el tamaño de la trama a cifrar es de 384 bits, en lugar de los 114 de las tramas de voz.

3.4. El estado del arte

3.4.1. A5/1 y A5/2

El diseño y funcionamiento interno del algoritmo de cifrado A5/1 y A5/2, inicialmente opaco a la comunidad investigadora, fue revelado en líneas generales al dominio público en [4]. Posteriormente, en [23] los autores publicaron los detalles del algoritmo, obtenido mediante un proceso de ingeniería inversa de una implementación hardware, así como una implementación software de referencia y unos vectores de prueba. Esta última implementación sería posteriormente confirmada como correcta por la organización GSM en [15]. El algoritmo publicado revelaría que el diseño del A5/1 ofrecido en [4] sería correcto, con la excepción de la posición específica de los bits utilizados para la decisión

[3]Más información sobre el proceso de retirada del algoritmo A5/2 puede encontrarse en `http://security.osmocom.org/trac/wiki/A52_Withdrawal`

de desplazamiento irregular de registros y la retro-alimentación lineal.

El diseño de A5/2, esquematizado en la figura 3.13, era revelado junto con el de A5/1 en [23]. Siendo concebido como una variante débil de este último para fines de exportación, no resultaría sorprendente que demostrara una fortaleza notablemente menor a la de demostraría A5/1.

El primer criptoanálisis de A5/2 llega en [64], inmediatamente tras la publicación del algoritmo, donde se describía un ataque de texto plano conocido capaz de derivar la clave K_c tras 2^{16} operaciones, utilizando 2 tramas que se encontrasen separadas por 2^{11} tramas.

Posteriormente, en [127], se presentaba un criptoanálisis de naturaleza algebraica donde el estado interno del algoritmo era descrito por un sistema de ecuaciones cuadráticas capaz de predecir los sucesivos bits de salida de *keystream*, analizando 4 tramas GSM (de 114 bits cada una), y con una complejidad de 2^{17} eliminaciones de Gauss en matrices de tamaño $(719 \times 400)^2$.

En [9] los autores presentan un nuevo ataques contra A5/2 que no requiere conocimiento de texto plano. El ataque propuesto es capaz de recuperar la clave utilizada analizando tan solo 8 tramas GSM, con una complejidad de 2^{16}, 2^{28} bytes de memoria y un computo previo de 2^{47} operaciones XOR (unas 5 horas). Para ello utiliza el campo CRC de las tramas GSM que forma parte del texto plano cifrado con A5/2.

Dado que el ataque no requiere de conocimiento del texto plano de ninguna de las tramas, su aplicación práctica es tremendamente efectiva y posibilita la ruptura de una conversación GSM cifrada con A5/2 en tiempo real utilizando un PC estándar.

El primer ataque criptográfico contra A5/1 fue apuntado directamente en [4], donde se menciona brevemente la viabilidad de recuperar el contenido de los registros R1 y R2 mediante búsqueda exhaustiva y derivar el contenido de R3 utilizando el *keystream*. Sin embargo, aún sería necesario explorar algunos bits adicionales de R3 con el objetivo de determinar algunas decisiones de desplazamiento irregular, por lo que la búsqueda exhaustiva debería ser expandida a un total de 2^{45} a 2^{52}candidatos, tal y como sería apuntado posteriormente

por [15] y [128].

El primer ataque criptográfico convenientemente documentado contra A5/1 vendría de la mano de Golic et al [66], donde se presenta un primer ataque contra el algoritmo revelado inicialmente en [4]. Los autores describen un método de criptoanálisis capaz de derivar un conjunto de ecuaciones lineales independientes mediante el análisis de la decisión de desplazamiento irregular para cada uno de los 3 registros y los bits de salida.

El ataque es capaz de derivar la clave K_c mediante la resolución de $2^{40,16}$ sistemas de ecuaciones. Golic también apunta en su artículo la reducción del espacio posible de estados internos en cada ronda de reloj, así como la posibilidad de aplicación de un ataque basado en situaciones de compromiso de espacio tiempo (TMTO), sentando las bases de los ataques futuros que le sucederían.

En términos prácticos, dicho ataque supone una mejora marginal con respecto al apuntado por [15], ya que a pesar de mejorar el tamaño del espacio de estados a ser explorado exhaustivamente, de 2^{45} a 2^{40}, cada evaluación resulta más costosa, ya que involucra la utilización de un sistema de ecuaciones lineales.

Posteriormente, en [22] los autores publicaron que el algoritmo COMP128 en su versión 1, utilizado para derivar la clave K_c de A5/1 partiendo de la clave permanente K_i almacenada en la tarjeta SIM del suscriptor, estaba diseñado para mantener a ceros los 10 bits más significativos de la clave de 64 bits. Por lo tanto, la longitud efectiva de la clave A5/1 sería de tan solo 2^{54}, facilitando su ataque por búsqueda exhaustiva.

Tras la publicación del algoritmo exacto del A5/1 en [23], vendría el primer criptoanálisis [15] directamente aplicable al algoritmo real implementado en los sistemas GSM. En [15] los autores presentan 2 nuevos tipos de ataques basados en situaciones de compromiso de espacio tiempo donde es posible romper A5/1 casi en tiempo real utilizando 2^{42} cálculos previos y 292 GB (o 2^{42} cálculos y 146GB) de almacenamiento permanente. El ataque es capaz de derivar la clave tras el análisis de 2^{15} y 2^{9} tramas GSM respectivamente.

En [13] se presenta una nueva técnica de criptoanálisis del A5/1, capaz de recuperar el estado interno del algoritmo y derivar la clave realizando rondas inversas siguiendo la técnica apuntada por [15]. Como resultado, el ataque es capaz de determinar la clave K_c de 64 bits tras el análisis de $2^{20,8}$ *keystreams* y el cómputo de $2^{39,91}$ rondas de A5/1. En comparación con los ataques anteriores presentados por [4] y [15], donde la complejidad del ataque en número de rondas A5/1 es del orden de 2^{47}, el ataque descrito por [13] supone una mejora substancial.

En [43], Ekdahl et al. describen un nuevo ataque criptográfico contra A5/1 cuya complejidad es prácticamente independiente de la longitud de los registros LFSR utilizados en el algoritmo. El ataque, basado en los ataques de correlación[4] descritos en [101] y [83], utiliza la mala inicialización de la clave (K_c e IV) existente en A5/1. Su eficacia es dependiente del número de rondas previas a la producción del primer bit de *keystream* (100 en A5/1). El ataque propuesto es capaz de recuperar la clave con una probabilidad cercana al 50 % tras analizar unos 70000 *keystreams*.

Posteriormente, en [99], los autores presentan un segundo ataque de correlación contra A5/1 que mejora al presentado por Ekdahl et al. alcanzando una efectividad del 60 % con 10000 *keystreams*.

En [8] los autores presentan una evolución de los ataques de correlación demostrados en [101] y [99]. El ataque propuesto es capaz de determinar la clave de A5/1 requiriendo el análisis de 1500-2000 tramas, con una efectividad superior al 90 %

Esta primera generación de ataques presenta ciertas dificultades en lo referente a su implementación práctica para la ruptura de claves A5/1 en entornos operacionales GSM reales. Con la excepción de [66], los ataques asumen que el atacante es capaz de obtener gran cantidad de *keystreams*, o con determinadas condiciones de difícil cumplimiento en la práctica, pertenecientes a la

[4]El ataque Coisel-Sanchez contra el *DECT Standard Cipher* presentado en el capítulo 2 de la presente tesis se basa también en los mismos principios de los ataques de correlación utilizados contra A5/1.

misma clave K_c. En GSM, cada trama es cifrada con 114 bits provenientes del *keystream* resultante de la aplicación de A5/1 sobre la K_c de 64 bits que cifra la comunicación y el IV de 22 bits proveniente del número de trama que se cifra o descifra. El conocimiento del texto plano correspondiente a una trama cifrada, permitiría a un atacante obtener la porción de *keystream* correspondiente.

Sin embargo, la predicción del texto plano de tramas GSM individuales, si bien es posible en casos específicos, es inviable para conjuntos masivos de tramas, condición requerida por la mayoría de los ataques referenciados anteriormente para la obtención de los respectivos *keystreams*.

En [128] se describe por primera vez un ataque efectivo contra A5/1 que requiere únicamente conocimiento de 64 bits de *keystream* para una clave K_c y un IV determinados. Los autores se basan en el ataque descrito por [66] y lo implementan siguiendo una técnica de compromiso de hardware-software donde determinadas tareas son ejecutadas en una implementación hardware en FPGA. Como resultado, el ataque es capaz de romper la clave K_c en una media de 2.5 días utilizando para ello únicamente 64-bits consecutivos de *keystream*, obtenible mediante el conocimiento del texto plano correspondiente a una única trama GSM cifrada.

Recientemente, en [131], se ha presentado una implementación del ataque de Pornin utilizando GPUs. La implementación es capaz de romper una clave A5/1 en unas 8 horas siguiendo una técnica similar a la descrita por [128].

En [69] se documenta la creación de COPACABANA, un clúster compuesto por 120 FPGAs y su aplicación para un ataque del tipo compromiso-espacio-tiempo contra A5/1, que requiere de 114 bits de keystream (una trama cifrada con texto plano conocido). Dicho ataque es similar al propuesto por [76] y posteriormente por [116].

En [63] los autores presentan un nuevo ataque desarrollado sobre la plataforma COPACABANA, que se basa en el trabajo previo de [84] y permite romper la clave K_c en unas 7 horas de cómputo, utilizando únicamente un *keystream* de 64 bits y ningún tipo de cómputo previo.

En [9] los autores presentan un nuevo ataque contra A5/1 que utiliza el

CRC de las tramas, el cual forma parte del texto plano, y no requiere de texto plano conocido. El ataque, tras un proceso de cómputo previo relativamente largo con un almacenamiento permanente de 4.5 TB, sería capaz de derivar la clave K_c mediante el análisis de 2^{12} tramas cifradas (unos 5 minutos de conversación).

En [76], los autores revelan un proyecto para la aplicación de forma práctica de un ataque del tipo TMTO [72] contra A5/1 en PCs, utilizando una técnica similar a la propuesta por [69] sobre la plataforma COPACABANA. Un total de 68 FPGAs son utilizadas para computar tablas *rainbow* capaces de derivar el estado interno del A5/1 para 64-bit determinados de *keystream*. Dichas tablas nunca fueron liberadas y los autores no realizaron ninguna publicación posterior al respecto.

Debido a ello, Nohl et al. iniciaron un proyecto denominado *The A5 Cracking Project* [116], con el objetivo de crear, en un esfuerzo comunitario, un total de 2TB de tablas para el ataque a A5/1, utilizando las técnicas de *puntos distinguidos* [118] y *tablas rainbow* [14]

Utilizando la implementación que el equipo del proyecto desarrolló a tal efecto [148], es posible atacar A5/1 utilizando un *keystream* conocido de 114 bits de longitud. La creación de las tablas y el funcionamiento del proceso se encuentran parcialmente documentados en [112] y [97].

Más recientemente, en [142], se propone una evolución del ataque a A5/1 sugerido originalmente en [4], donde por cada posible combinación de bits de R1 (2^{19}) se pre-calcula el contenido de R2 y R3 para un conjunto determinado de bits de *keystream*, ocupando como resultado 5.65GB de memoria. En una segunda fase, el ataque determina un conjunto de $2^{48,5}$ estados internos candidatos entre los que el correcto se encontrará incluido con un 100 % de probabilidad.

En la sección 3.7.2 del presente capítulo se analizan estas últimas técnicas de criptoanálisis y se demuestra para la interceptación pasiva experimental de conversaciones cifradas en GSM.

3.4.2. A3 y A8

Los algoritmos A3 y A8 fueron liberados por primera vez al dominio público en [22], donde los autores presentaron una implementación software del algoritmo COMP128 derivado de una tarjeta SIM mediante un proceso de ingeniería inversa. Los autores también revelaron que, en el uso de COMP128 por el algoritmo A8 para la generación de la clave K_c que será usada por A5/1, los 10 bits más significativos de la nueva clave son siempre fijados a 0, reduciendo de esta manera la longitud efectiva de K_c de 64 a 54 bits.

Esta primera versión del algoritmo COMP128 fué rápidamente criptoanalizada en [24], donde se describe un ataque del tipo texto plano elegido, capaz de derivar la clave K_i de la tarjeta enviando un conjunto de unos 2^{17} solicitudes de cifrado a la misma. El ataque requiere acceso físico a la tarjeta y tiene una duración aproximada de 8 horas [71]. En [163] el autor propone un nuevo ataque basado en los mismos principios que ofrece unos resultados similares.

En [132] los autores presentan un nuevo tipo de ataque de canal lateral, denominado ataque de partición, contra implementaciones reales de COMP128 en tarjetas SIM. El ataque demostrado es capaz de recuperar la clave K_i de 128-bit utilizando tan solo 8 peticiones a la tarjeta SIM.

Más recientemente, en [113] los autores demuestran cómo es posible extraer la clave K_i de la tarjeta SIM mediante la introducción de un software malicioso en la tarjeta que abuse de ciertas vulnerabilidades existentes en la mayoría de las Máquinas Virtuales Java (JVM). Dicho ataque puede potencialmente ser realizado remotamente desplegando la aplicación Java maliciosa mediante un SMS de actualización en una técnica similar a la utilizada por [108] o [2] para inducir cambios en el comportamiento de la tarjeta SIM.

3.4.3. Interceptación de comunicaciones GSM

Los ataques contra los algoritmos de cifrado A5/1 y A5/2 fueron calificados inicialmente de teóricos[5] por la industria GSM, en base a la dificultad exis-

[5]En la nota de prensa número 30 de 2009 de GSMA titulada *"GSMA Statement on Media Reports Relating to the Breaking of GSM Encryption"*, los ataques existentes contra GSM y A5/1 son

tente en su aplicación práctica, derivada de la falta de herramientas hardware y software para la monitorización e interacción con comunicaciones GSM.

En [76], los autores presentarían públicamente un resumen de los últimos avances desarrollados en el uso de herramientas software libre y hardware de bajo coste para ejecutar en la práctica ataques contra comunicaciones GSM.

Posteriormente, en [116], se presenta un resumen de los principales métodos existentes para la interceptación práctica de comunicaciones GSM, utilizando tanto métodos activos como pasivos, y se anuncia el comienzo del proyecto *A5/1 Cracking* con la intención de demostrar que el ataque práctico conversaciones GSM cifradas con A5/1 es factible utilizando herramientas libres de bajo coste. Paralelamente, en [146], se revisan las diferentes herramientas existentes para la interacción y monitorización de GSM, incluyendo la modificación de terminales móviles disponibles en el mercado de consumo.

En lo referente a la monitorización de tráfico GSM, el primer método asequible a la comunidad investigadora, usa la interfaz de servicio que ciertos modelos de terminales ofrecían como soporte al proceso de ingeniería y depuración de errores. El más difundido fue el Nokia 3310, que incorporaba dicha función y para el cual se liberó una implementación [120] capaz de permitir la conexión de un terminal móvil a un PC para la monitorización de todo el tráfico recibido por el mismo. A pesar de que mediante este método únicamente fuera posible recibir los paquetes GSM dirigidos a dicho terminal (o enviados por éste) y aquellos difundidos a todos los terminales, la disponibilidad de dicha herramienta sería decisiva como soporte a ulteriores investigaciones de seguridad GSM por parte de la comunidad académica.

La capacidad de monitorizar conversaciones ajenas fue ofrecida mediante una implementación de un receptor GSM [147] software basado en SDR para el dispositivo USRP versión 1. Utilizando dicho software es posible escuchar a un ARFCN determinado y decodificar los paquetes GSM transmitidos. Al ser una implementación basada totalmente en software, se posibilita la recepción

---

calificados como de meramente teóricos con nula aplicación práctica, no presentando ningún tipo de riesgo para la privacidad de los usuarios GSM.

de cualquier ranura temporal de cualquier canal físico GSM en cualquiera de las bandas soportadas por el USRP v1, incluyendo todas aquellas usadas por GSM.

Dicha implementación para la interceptación pasiva de tráfico, presenta sin embargo dos limitaciones. Por una parte es únicamente capaz de monitorizar el tráfico del canal de bajada ($BTS \rightarrow MSE$), no siendo posible monitorizar el tráfico originado por el terminal móvil. Por otro lado, no es posible la interceptación pasiva de tráfico en el caso en el que la BTS utilice salto de frecuencia, ya que la implementación del ataque solamente puede escuchar en un canal físico de forma simultánea.

Recientemente, en [29], se demuestra la utilización de varios USRP en paralelo para la captura de varios canales físicos de forma simultanea, posibilitando la interceptación de llamadas incluso en aquellos casos donde el salto de frecuencia se encuentra activado. En [41] se explora la viabilidad de analizar la comunicación y seguir el salto de frecuencias de la misma manera que opera un terminal móvil.

La popularización de USRP como dispositivo SDR de bajo coste posibilitó también el desarrollo de una implementación software de una BTS GSM [26][6], capaz de funcionar como una estación base GSM. Utilizando dicha implementación [110], se posibilita el aprovechamiento de la ausencia de autenticación mutua entre el MSE y la BTS para suplantar la identidad de esta última [116]. De la misma manera, sería posible capturar identidades IMSI, tal y como sería demostrado por [162] y [116]. En [124] se demuestra cómo es posible interceptar de forma activa llamadas efectuadas por un MSE, suplantando la identidad de una BTS legítima, permitiendo la conexión de cualquier IMSI y conectando la llamada mediante el uso de una PBX con una línea telefónica fija o móvil. De forma similar al ataque presentado en [124], en [126] el autor presenta un ataque activo emulando una BTS legítima donde se interceptan las comunicaciones de datos GPRS de la víctima, utilizando para ello una imple-

[6]En [26] los autores explícitamente rechazan analizar el potencial de su implementación para llevar a cabo ataques contra el protocolo GSM.

mentación de controlador de estación base [121] y una micro BTS. En [154] se resumen algunos de los diferentes ataques existentes contra el protocolo GSM.

Más recientemente, en [161], se presenta un software de banda base completo para ciertos terminales móviles, denominado *OsmocomBB*, concebido como reemplazo del software de banda base existente en ciertos terminales de bajo coste. En [68] se demuestran ciertos ataques activos de denegación de servicio contra la infraestructura GSM utilizando dicho software, con implicaciones en la privacidad de los usuarios.

Simultáneamente, en [115] los autores revisan las diferentes opciones mencionadas anteriormente para la interceptación pasiva de conversaciones GSM y demuestran la viabilidad de la interceptación pasiva de conversaciones utilizando el software de banda base referenciado anteriormente.

La utilización de un terminal GSM con chipset TI-Calypso equipado con el software de banda base OsmocomBB [122] permite únicamente la monitorización de un máximo de 200Khz, correspondiente a un único canal ARFCN GSM. Dentro de dicho ancho de banda, los autores demuestran cómo modificando el software de banda base es posible monitorizar de forma simultánea las 8 ranuras temporales del canal físico. Con una pequeña modificación hardware, eliminando un filtro, el terminal puede ser utilizado también para la monitorización del canal de subida.

Más recientemente, en [109], se demuestra cómo es posible suplantar la identidad de una BTS utilizando OsmocomBB en un terminal móvil compatible. El mismo autor demuestra en [65] un efectivo ataque de Denegación de Servicio sobre redes GSM, mediante el abuso activo del mecanismo de *paging* de la red.

En [114] se demuestra un ataque de suplantación de la identidad de un usuario de la red, donde se abusa de la ausencia de autenticación en todas las llamadas y, mediante la ruptura de la clave K_c del algoritmo A5/1, se consigue realizar llamadas fraudulentas. Los autores también sugieren formas de defensa contra los ataques existentes a la privacidad de los usuarios GSM, en particular, en lo referente a la actualización del algoritmo de cifrado y a la mi-

nimización de la cantidad de texto plano predecible. Con el objetivo de monitorizar el progreso de los operadores en la implementación de contramedidas contra los ataques a la privacidad de usuarios GSM, los autores anuncian un proyecto comunitario[7] donde se recopilan estadísticas reales de tráfico.

3.4.4. Localización de usuarios GSM

La seguridad del sistema de señalización por canal común número 7, también llamado SS7, utilizado para la comunicación de señalización entre los diferentes operadores de telefonía móvil, ha sido cuestionada por la comunidad académica en [106] y la viabilidad de su abuso para la localización no autorizada de usuarios GSM ha sido demostrada en numerosas ocasiones [90], [44], [92], [91].

En [89] los autores demuestran cómo es posible determinar la presencia de un usuario GSM en un área determinada, monitorizando de forma pasiva el tráfico del canal de bajada de las estaciones base [115]. A diferencia de trabajos previos, [30] o [37], el trabajo de los autores se centra en la localización geográfica de usuarios por parte de terceros mediante la monitorización de su tráfico de bajada GSM, de forma similar a lo que otros investigadores demuestran en [77] aplicado a tráfico WiFi.

Recientemente, en [6], los autores demuestran que el mecanismo de protección de la privacidad de identidades de usuarios en GSM, mediante el uso de seudónimos itinerantes, en numerosas ocasiones no se encuentra implementado correctamente por parte de los operadores y resulta ineficaz como medida de prevención del seguimiento de usuarios.

3.5. Seguimiento experimental de identidades GSM

En esta sección se exploran experimentalmente los riesgos que el protocolo GSM presenta para la privacidad de los usuarios en lo referente al seguimiento de sus identidades. El propósito de esta sección es contribuir a la validación de

[7] GSM-MAP. `http://gsmmap.org`

la hipótesis de la tesis y consecución de sus objetivos, por lo que el escenario que se utiliza es aquél donde el atacante hace uso de dispositivos SDR de bajo coste y lleva a cabo el ataque sin poseer grandes recursos.

El teléfono GSM se ha convertido en un elemento indispensable en el día a día de miles de millones de usuarios a nivel mundial. Para la mayoría de éstos, es un elemento que, al igual que fuera el reloj en antaño, llevan consigo continuamente.

En este contexto, localizar geográficamente el terminal móvil GSM equivale a localizar al usuario al que éste pertenece. Las implicaciones para la privacidad de los suscriptores son obvias en el caso en que un atacante pudiera seguir los movimientos geográficos de un determinado terminal GSM.

En lo referente al seguimiento de identidades GSM utilizando el enlace de radiofrecuencia entre el MSE y la BTS, el código IMSI presenta el riesgo más elevado. Dicho código, único a nivel mundial para cada suscriptor GSM, posee la característica de que es invariante en el tiempo y cuando es utilizado para el direccionamiento, es transmitido sin cifrado.

El protocolo GSM, al igual que ocurre con cualquier otro protocolo de radio, es susceptible de ser interceptado remotamente de forma pasiva, dado que sus ondas de radio se propagan a lo largo de varios kilómetros. Por lo tanto, cada vez que el código IMSI es transferido por el terminal móvil, el usuario anuncia de forma inadvertida su presencia. En efecto, cualquier atacante podría fácilmente determinar la posición geográfica de cualquier usuario GSM en tiempo real utilizando dicha información, con consecuencias obvias para la privacidad de los usuarios.

Con el objetivo de minimizar este problema, el protocolo GSM prevé la utilización de TMSIs, a modo de seudónimos, con un tiempo de vida limitado. Tal y como se ha descrito en la sección 3.3.2, durante el proceso de registro del MS en la red GSM, el VLR asignará un código TMSI al código IMSI recibido en el proceso de registro. De esta manera, el IMSI es únicamente enviado por el MS durante el proceso de registro. Una vez el TMSI es asignado, todo el direccionamiento será realizado utilizando este último.

En el caso en el que un suscriptor reciba un SMS o llamada entrante, la red GSM enviará un mensaje de *paging* dirigido al código TMSI correspondiente al número llamado. Dicho mensaje de *paging* será transmitido por el PCH, el cual es continuamente monitorizado por todos los terminales MS conectados a dicha BTS. Únicamente aquel MS cuyo TMSI coincida con el requerido, contestará al mensaje.

Los mensajes de *paging* no van cifrados y circulan por un canal cuya naturaleza es pública, en el sentido en el que es monitorizado de forma automática por todos los MS conectados a la BTS. A pesar de que un terminal GSM procese de forma automática dichos mensajes y solo reaccione ante aquellos que se refieran a su propio TMSI, un atacante podría recibirlos remotamente sin filtrado, con el objetivo de determinar los TMSIs localizados en dicha área geográfica.

Con el objetivo de ilustrar la capacidad de un atacante de monitorizar los mensajes de *paging*, se ha realizado un experimento donde el canal de bajada de BTS cercanas es monitorizado de forma pasiva utilizando la plataforma de banda base OsmocomBB [122], en un PC conectado con un terminal móvil Motorola C118 conectado vía puerto serie (utilizando un conversor USB a puerto serie con soporte para velocidades altas de transferencia de datos).

Se ha instalado el *firmware* de nivel 1 en el terminal de tal forma que todos los mensajes GSM de capa 1 sean transmitidos al PC mediante el puerto serie para ser recibidos por el *framework* de OsmocomBB. Con el objetivo de preservar la privacidad de las identidades reales, se procede a eliminar en tiempo real cualquier identidad IMSI o TMSI recibida como parte del flujo de proceso de datos. A efectos del experimento, se contabilizarán la cantidad de TMSIs e IMSIs observados, pero no sus valores concretos.

La figura 3.5 muestra en verde la cantidad de mensajes de *paging* acumulados en los últimos 60 segundos por un periodo de 4 días. Se puede apreciar como la actividad disminuye notablemente durante la noche, alcanzando su punto más bajo sobre las 4 de la mañana. Si bien la gran mayoría de mensajes de *paging* son realizados por TMSI, existe una minoría, representados en azul,

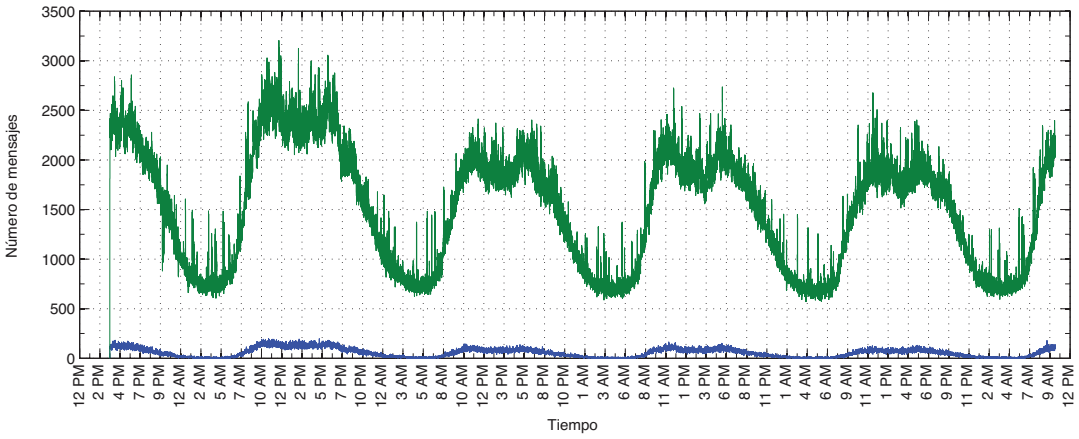


Figura 3.5: Cantidad de mensajes de *paging* acumulados en los últimos 60 segundos por un periodo de 4 días

que son realizados por IMSI.

El objetivo de la asignación de los identificativos TMSI es garantizar la privacidad respecto a la identidad de los usuarios. Por ello, el tiempo de vida de los TMSI es limitado, de tal manera que pasado cierto tiempo el TMSI será renovado a petición de la red GSM y uno nuevo será asignado al usuario. De esta forma, la privacidad de los usuarios queda protegida, ya que se imposibilita la realización de un seguimiento y localización de los mismos por parte de terceros.

Por el contrario, el identificativo IMSI es permanente ya que identifica de forma única el contrato de suscripción a la red GSM. Según la teoría, el IMSI solamente debería ser enviado "por el aire" durante la fase de registro del terminal MS en la red GSM. En efecto, como parte del proceso de registro el MS transmitirá el código IMSI a la red GSM y recibirá de ésta el TMSI correspondiente. A partir de ese momento, el direccionamiento a dicho IMSI, en forma por ejemplo de mensajes de *paging* será realizado por TMSI.

Sin embargo, en la práctica se han observado situaciones donde, sin la existencia de ningún tipo de ataque activo, la red GSM utiliza el IMSI para el direccionamiento en los mensajes de *paging*, en lugar del IMSI, para algunos suscriptores. En la figura 3.5 se puede observar cómo existe una proporción relevante de mensajes de *paging* que utilizan códigos IMSI (mostrado en color

azul) con respecto al total de mensajes de *paging* (mostrado en color verde).

Existen varias explicaciones para la existencia de dicha "anomalía". Cuando un teléfono móvil se desconecta de la red GSM sin haber realizado correctamente el proceso de salida, pasado un tiempo, el almacenamiento del TMSI asociado a dicho IMSI expirará en el VLR. A partir de ese momento, los mensajes de *paging* dirigidos a dicho móvil comenzarán a utilizar su IMSI. Como además el móvil no se encuentra conectado y no responderá a los mensajes, la red GSM comenzará un proceso de búsqueda de su localización, de tal manera que los mensajes de *paging* se extenderán progresivamente a áreas mayores. Por lo tanto, dichos mensajes de *paging* que utilizan IMSI, no necesariamente corresponderán a MS que se encuentren cercanos a la BTS que los transmite.

El ratio de IMSI/TMSI mostrado por la figura 3.5 se puede considerar normal y correspondiente a una red GSM "sana". Sin embargo, incluso en estas circunstancias, dichos IMSI viajan sin ningún tipo de protección y pueden ser interceptados fácilmente como se ha demostrado en este experimento.

Sin embargo, existen otros casos donde el ratio IMSI-TMSI es desproporcionadamente alto, alcanzando incluso tasas del 20 %-60 % respecto a la totalidad de mensajes de *paging*. Dado que la memoria del VLR es volátil y limitada, en ocasiones la cantidad de suscriptores en un área determinada puede ser superior a la capacidad máxima de la memoria que el VLR utiliza para almacenar las asociaciones de TMSI-IMSI. En estos casos la red GSM comenzará a utilizar el IMSI en los mensaje de *paging*. En aquellas ocasiones en las que el VLR no se encuentre disponible, bien sera durante un reinicio manual o un fallo de operación, la red GSM comenzará a utilizar el direccionamiento por IMSI.

Estas circunstancias, no afectan a la propia funcionalidad del servicio GSM, tal y como es percibido por los usuarios, de tal manera que la amenaza que suponen a la privacidad no será percibida por los mismos.

Lamentablemente, el mecanismo de protección de privacidad por TMSI es ineficiente ante un tipo de ataque activo que abusa de una de las mayores vulnerabilidad del protocolo GSM, la falta de autenticación mutua entre el terminal móvil (MS) y la red GSM. En efecto, tal y como se ha descrito en

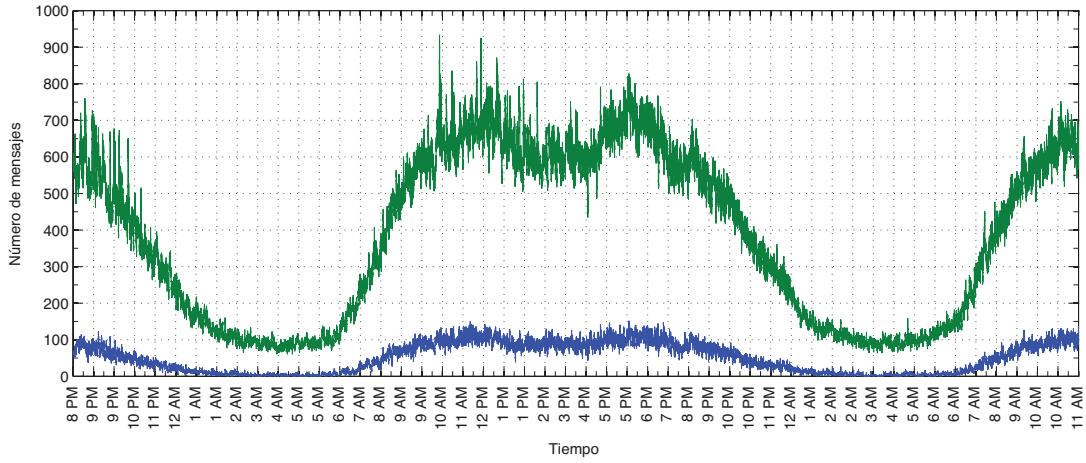


Figura 3.6: Cantidad de mensajes de *paging* acumulados en los últimos 60 segundos por un periodo de día y medio

la sección 3.3.1, la red GSM es capaz de autenticar al terminal móvil siguiendo el procedimiento de Autenticación del MS, esquematizado previamente en la figura 3.4. Sin embargo, GSM no prevé la autenticación de la red GSM por parte del terminal móvil MS.

Gracias a esta vulnerabilidad, un atacante puede crear una BTS falsa y modificarla de tal manera que se anuncie como una legítima perteneciente al operador GSM de la víctima.

Cuando el terminal móvil inicia el proceso de registro en una red GSM, primero escanea todos los ARFCNs con el objetivo de identificar todas las estaciones base disponibles en el área geográfica donde se encuentra. Por cada BTS, el MS determinará el operador al que pertenece en base al CGI que es difundido por ésta, el cual se compone del MCC, MNC, LAC y CI, descritos previamente en la sección 3.1.

Una vez identificado el conjunto de BTS del operador al que pertenece la línea, el MS determinará la estación más adecuada, en base a parámetros tales como la potencia y calidad de la señal recibida, y se conectará a ésta. Sin embargo, debido a la ausencia de autenticación de la red GSM por parte del MS, este último no tendrá forma de saber si dicha BTS es legítima o no.

En efecto, es posible para un atacante, determinar los parámetros MCC, MNC, LAC y CI del conjunto de BTS disponibles en un área geográfica deter-

minada para un operador GSM concreto.

Con el objetivo de evaluar el grado de amenaza que este vector de ataque supone para la privacidad de las identidades de los usuarios, así como la dificultad y recursos necesarios para llevar a cabo el ataque, se ha diseñado el siguiente experimento.

La víctima del ataque es representada por un terminal móvil se encuentra conectado a una red de telefonía GSM. No existe ningún tipo de requisito para el teléfono o el operador. El experimento ha sido repetido utilizando diversos tipos de terminales y operadores, con el objetivo de verificar los resultados bajo diferentes condiciones. Dado que el terminal y su tarjeta SIM forman parte del experimento, se conoce de antemano su IMSI, lo cual es necesario para validar el resultado.

Se desarrolla un *IMSI-catcher* utilizando como base el software OpenBTS [26] bajo GNU/Linux. La configuración específica utilizada en OpenBTS simula los parámetros MCC, MNC, LAC y CI de la célula cuya identidad se pretende suplantar. Como hardware se utiliza un PC estándar y el USRP N210, cuyas especificaciones eran descritas en las sección 1.2 del capítulo 1.

OpenBTS ha sido configurado para rechazar cualquier intento de registro a la red por parte de terminales móviles, no sin antes almacenar el IMSI que será enviado como parte del inicio del proceso de registro por parte de aquellos terminales que intenten conectarse.

El objetivo del ataque consiste en que la BTS falsa, simulada por la implementación realizada del *IMSI-catcher*, se camufle como perteneciente a un operador GSM determinado. Como parte del funcionamiento de los terminales GSM, la potencia de la señal recibida por parte de las diferentes BTS del operador utilizado, es continuamente monitorizada a fin de detectar cambios de zona asegurando la estabilidad y calidad de las comunicaciones.

De esta manera, el teléfono es capaz de reconectar de manera transparente a otra BTS, en caso de que, en términos de calidad de señal, el cambio sea beneficioso. Como parte de la información transmitida por una BTS, como se ha descrito en la sección 3.1, se incluye la lista de BTS vecinas. Dicho conjunto

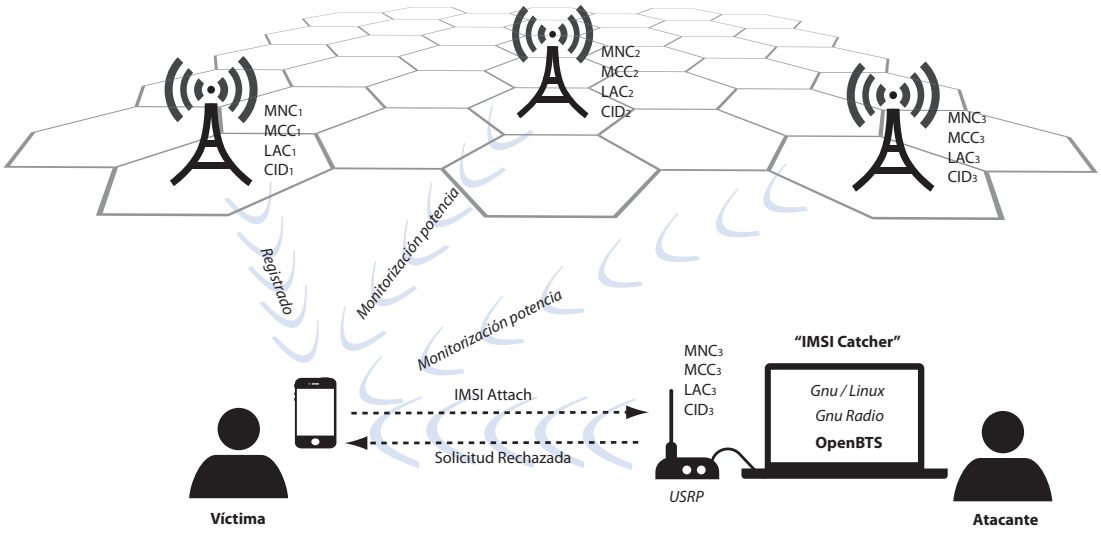


Figura 3.7: Descripción del vector de ataque utilizado en el experimento realizado para la captura de identidades IMSI

de BTS será el utilizado por el MS para buscar una BTS alternativa con una mejor señal. En el caso de que ninguna de ellas se encuentre disponible, el MS considerará otras BTS, dando prioridad a aquellas de su operador GSM.

La figura 3.7 esquematiza el escenario del ataque y el modo de operación del *IMSI Catcher* desarrollado. Los resultados experimentales demuestran que la eficacia del *IMSI Catcher* depende en gran medida de su integración adecuada en el entorno en el que opera, en lo referente a la estructura de las células del operador GSM al que suplanta la identidad. Los resultados óptimos se obtienen principalmente en dos situaciones, cuando no existe cobertura de señal para ningún operador o cuando la BTS falsa se camufla como aquella en la lista de BTS vecinas con señal más débil.

En este último caso, si la señal original es lo suficientemente débil, no existirán interferencias con la señal del *IMSI Catcher*, el cual utilizará los valores MNC, MCC, LAC y CID pertenecientes a la original. Al estar incluido en la lista de BTS vecinas registrada por el MS y emitir con una señal más potente que la de la BTS sobre la que se encuentra registrado el teléfono, éste se intentará asociarse a ella en cuestión de minutos.

En cualquiera de los casos, cuando el MS intenta registrarse en la BTS falsa siguiendo el procedimiento, envía el IMSI. En este punto, la BTS falsa rechaza

el registro del MS, de tal manera que éste continúe registrado a la original. El usuario del MS no habrá observado ninguna anomalía en el funcionamiento del servicio o el terminal, pero el IMSI habrá sido capturado en el proceso.

En el experimento desarrollado se demuestra la facilidad con la que se puede desarrollar un *IMSI Catcher* utilizando Software Libre, un ordenador estándar y una Radio Definida por Software (USRP) de bajo coste. El protocolo GSM y los teléfonos GSM no cuentan actualmente con ningún mecanismo para evitar este ataque a la privacidad de las identidades de usuarios GSM.

3.6. Interceptación activa de comunicaciones GSM

En la sección 3.5 se ha descrito el desarrollo de un *IMSI Catcher* utilizando software libre y Radios Definidas por Software de bajo coste. Dicho ataque funciona debido a la ausencia de autenticación mutua entre la red GSM y los teléfonos móviles.

Dicha vulnerabilidad de GSM puede ser aprovechada, de la misma manera, para la interceptación activa de datos y llamadas de voz. En el caso del *IMSI Catcher*, el registro del MS en la BTS era rechazado, ya que el objetivo era capturar el IMSI. Sin embargo, si la finalidad del ataque es la de la interceptación del tráfico del MS, es posible permitir el registro en la BTS de tal forma que el MS pueda completar el registro.

En este experimento, se parte de la implementación del *IMSI Catcher* realizada en la sección 3.5, extendiendo su configuración para permitir el registro de un IMSI específico, que será aquél perteneciente al teléfono móvil que representará en el experimento a la víctima del ataque. Utilizando esta implementación, en combinación con otras medidas adicionales orientadas a limitar el rango de acción de las transmisiones del experimento, se consigue interceptar específicamente las llamadas efectuadas por el teléfono objeto del análisis experimental. En un escenario real, un atacante podría simplemente permitir el registro de cualquier IMSI, consiguiendo de esta forma interceptar el tráfico de cualquier teléfono que se encuentre en las inmediaciones.

Dado que en el escenario del ataque, el adversario no conoce la clave K_i almacenada en la tarjeta SIM de la víctima, no será posible utilizar ningún tipo de cifrado en la red, ya que la negociación de la clave de sesión K_c requeriría que ambos BTS y MS compartieran K_i. Sin embargo, esto no representa un problema, ya que GSM no requiere de la presencia de cifrado en su modalidad A5/0. Por lo tanto, OpenBTS no autenticará al MS y empleará el modo A5/0 sin cifrado.

Al igual que en el escenario del *IMSI Catcher* descrito en la sección 3.5, se configura OpenBTS con los parámetros MNC, MCC, LAC y CID de la estación base GSM cuya identidad se pretende suplantar. Una vez el teléfono móvil de la víctima se registre en nuestra célula GSM simulada, ya no se encontrará conectado a la red de su operador GSM, por lo que no podrá recibir llamadas o mensajes entrantes. Sin embargo, cualquier mensaje enviado o llamada efectuada por el MS, serán recibidos por OpenBTS.

Con el objetivo de re-encaminar la llamada o SMS efectuados por el MS a la red de telefonía pública, se utiliza el software Asterisk[8] con el módulo *chan_dongle*[9]. OpenBTS es configurado para reencaminar las llamadas entrantes a Asterisk, el cual las encamina hacia la red pública GSM mediante la utilización de un módem GSM Huawei modelo K3520 equipado con una tarjeta SIM del operador GSM.

Todo el proceso de reencaminado es realizado de forma transparente durante el ataque, a la vez que el audio de la conversación mantenida es registrado por OpenBTS. De esta forma, el atacante puede escuchar cualquier conversación telefónica perteneciente a una llamada efectuada por la víctima.

El proceso del ataque simulado en el experimento se esquematiza en la figura 3.8.

El experimento ha sido efectuado con diferentes terminales móviles y tarjetas SIM con resultados positivos en todos los casos. La figura 3.9 muestra el audio extraído por el interceptador GSM activo de una de las llamadas efec-

[8]Más información sobre Asterisk puede ser consultada en `http://www.asterisk.org/`

[9]Para más información sobre el módulo *chan_dongle* consultar `https://code.google.com/p/asterisk-chan-dongle/`

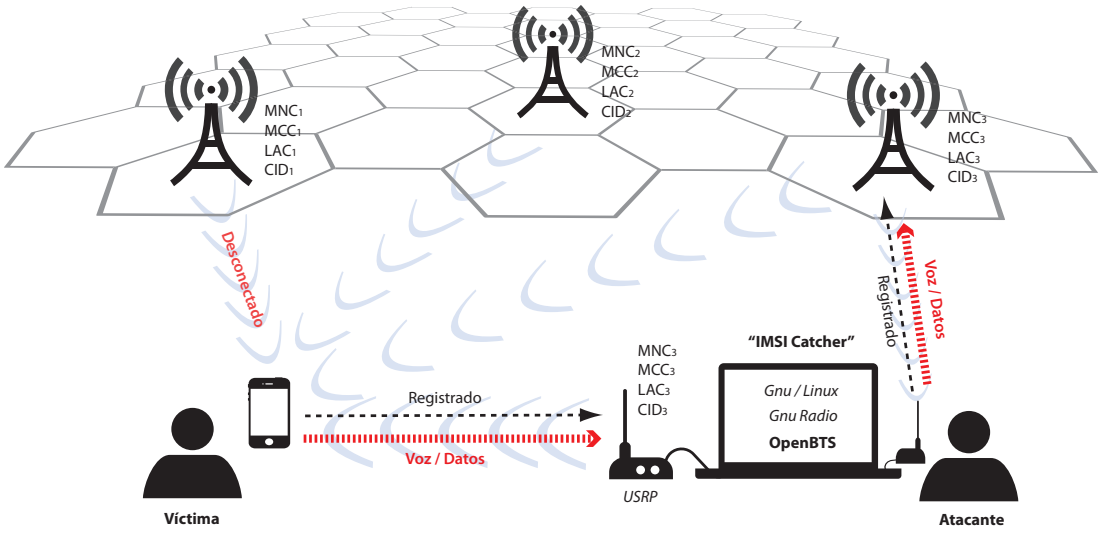


Figura 3.8: Arquitectura y escenario de uso del sistema de interceptación activa GSM desarrollado

tuadas durante los experimentos. En ninguno de los casos el terminal interceptado mostraba ningún signo indicando la ausencia de cifrado en la red GSM.

Como efecto secundario, a pesar de que el proceso sea completamente transparente para la víctima que efectúa la llamada, el destinatario de la misma se podría percatar que la recibe desde otro número diferente. En efecto, la llamada es efectuada a través de la red GSM utilizando la tarjeta SIM del atacante, por lo que la señalización indicará el número asociado al IMSI del atacante como el número llamante. El atacante podría ocultar el número marcando la señalización correspondiente, pero, en principio, no podría suplantar el de la víctima.

En una continuación de los experimentos se evalúa la viabilidad de interceptar la conexión a Internet sostenida mediante GPRS sobre la red GSM. Para ello, se parte de la configuración anterior y se activa el soporte soporte de GPRS de OpenBTS, a la vez que se configura el sistema GNU/Linux subyacente para el encaminado hacia Internet mediante NAT, vía una conexión de datos móvil independiente.

En el caso del experimento llevado a cabo, dicha conexión a Internet se efectúa mediante una conexión 3G utilizando un módem 3G USB. El encaminado es realizado mediante la activación de *ip_forwarding* del kernel de Linux,

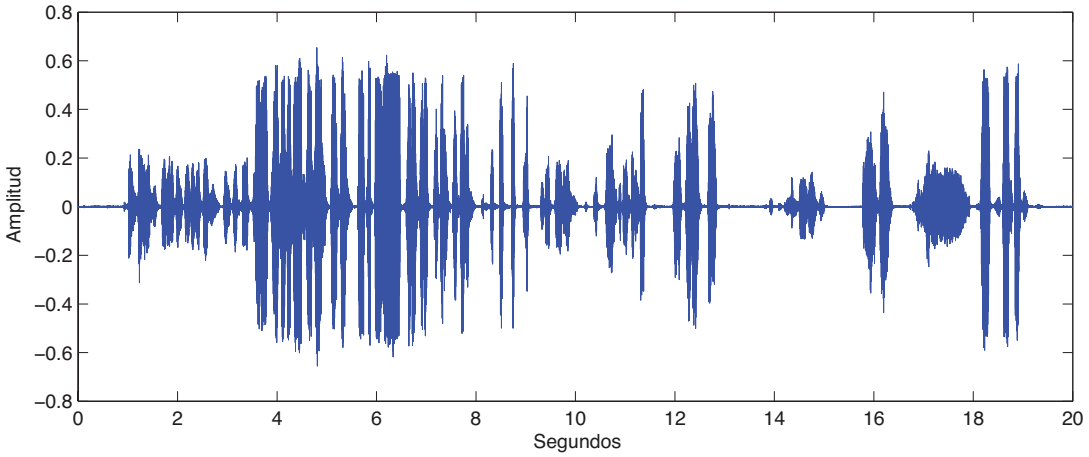


Figura 3.9: Audio extraído en un ataque simulado usando el interceptador GSM activo desarrollado.

junto con *iptables* para realizar una traducción de puertos origen (SNAT).

Como resultado, toda la comunicación de datos de la víctima con Internet se realiza a través del interceptador GSM, el cual realiza las funciones de encaminador de Internet. En este escenario, el atacante obtiene acceso a todas las comunicaciones de datos y se sitúa en posición de realizar ataques activos del tipo Hombre-en-el-Medio (MiTM). El resultado del experimento es positivo para todos los terminales analizados y su efectividad se ha demostrado interceptando conexiones HTTP para la extracción de la navegación efectuada en Internet por la víctima simulada en el experimento.

Llegado a este punto, donde se demuestra la capacidad para interceptar las comunicaciones de datos, las implicaciones de dicho ataque en la seguridad y privacidad de las comunicaciones de la víctima son similares a las del escenario de interceptación de comunicaciones WiFi, descrito en el capítulo 4.

3.7. Interceptación pasiva de comunicaciones GSM

En la presente sección se investiga experimentalmente la interceptación práctica de comunicaciones GSM de forma pasiva. Al contrario que el ataque de interceptación activo descrito en la sección 3.6, donde se realizaba una su-

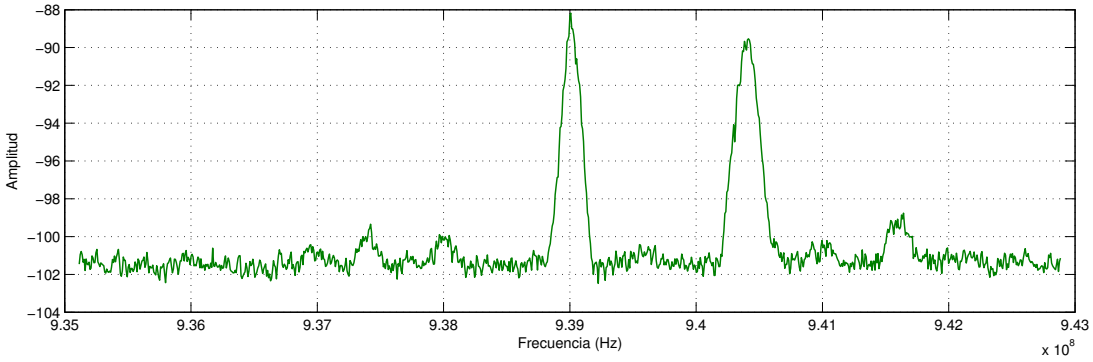


Figura 3.10: Porción del espectro de la banda GSM-900 mostrando varias BTS activas

plantación de la identidad de una célula GSM, en esta sección se explora una alternativa menos intrusiva orientada a la captura y descifrado del tráfico que un usuario mantiene con una estación base legítima.

El objetivo perseguido en esta sección es la evaluación del grado de efectividad de llevar a cabo un ataque de este tipo sobre redes GSM, así como la viabilidad de efectuarlo utilizando Radios Definidas por Software en combinación con los modestos recursos provistos por un ordenador personal estándar.

En lo referente a las herramientas y el hardware requerido para la captura pasiva de comunicaciones GSM con modestos recursos, se han identificado dos métodos principales. El primero de ellos consiste en la utilización de un terminal móvil con un procesador de banda base basado en el chip *Calypso* en combinación con el software desarrollado para su utilización como sistema de monitorización, denominado *OsmocomBB* [122].

El segundo método involucra el uso de dispositivos SDR de bajo coste cuya repercusión potencial sobre la seguridad y privacidad de las comunicaciones GSM, es objeto de investigación en la presente tesis. Tanto el RTL-SDR, en sus diferentes variantes de sintonizador, como la gama de dispositivos USRP, como el N210 utilizado en los experimentos realizados en la presente tesis, son capaces de monitorizar el espectro de frecuencias asignado a GSM, siendo capaces de monitorizar varios canales GSM de forma simultánea. La figura 3.10 muestra una porción del espectro de la banda de GSM-900 siendo monitoriza-

do por un USRP, donde se pueden observar claramente la existencia de varias estaciones base GSM activas.

La herramienta de código abierto Airprobe [147] permite la captura de tráfico GSM y posterior decodificación de paquetes, mediante el uso de dispositivos SDR, siendo compatible con prácticamente cualquier dispositivo SDR capaz de funcionar con GNU Radio.

3.7.1. Interceptación de comunicaciones no cifradas

Con el objetivo de demostrar la viabilidad de interceptación de tráfico GSM no cifrado utilizando el dispositivo SDR N210, se diseña el siguiente experimento que simula una situación real que pudiera darse en cualquiera de los países donde se utiliza la modalidad de cifrado A5/0.

Primeramente se simula un escenario donde un ordenador equipado con un USRP n210 representa una célula GSM configurada para utilizar cifrado A5/0. Para ello se utiliza el software *OpenBTS*, en una configuración similar a aquella utilizada para el experimento de interceptación activo descrito en la sección 3.6. Uno de los terminales móviles configurado para utilizar la red GSM creada a tal efecto, es utilizado para representar la víctima del ataque, la cual efectúa una llamada que será encaminada a su destinatario por el sistema *OpenBTS/Asterisk*.

El atacante es simulado utilizando otro USRP N210 en combinación con el software *AirProbe*, el cual captura de forma remota la comunicación entre el MS de la víctima y la BTS.

Dado que la red GSM opera en modo A5/0, no existe ningún cifrado que proteja la seguridad de las comunicaciones, por lo que es perfectamente posible interceptar la totalidad del tráfico de control y voz. Una vez interceptado, se recupera el contenido de voz y, tras aplicar el códec de descompresión, se obtiene el audio de la conversación simulada.

La figura 3.11 muestra una gráfica de la amplitud sobre el tiempo del audio capturado durante el experimento.

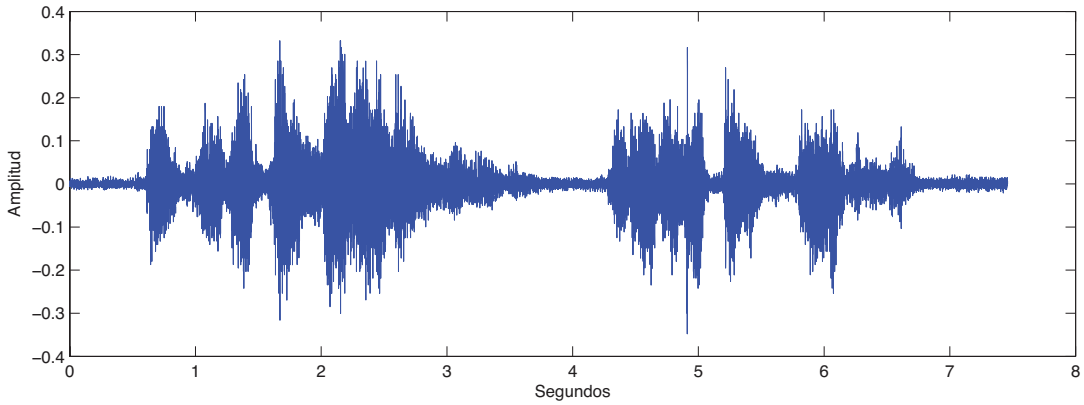


Figura 3.11: Audio extraído de la comunicación no cifrada interceptada.

3.7.2. Interceptación de comunicaciones cifradas

Dentro de las diferentes modalidades de cifrado GSM, A5/1 es, actualmente, la utilizada por la gran mayoría de los operadores a nivel mundial y la casi totalidad de las redes GSM europeas. El propósito de esta sección es contribuir a los objetivos de la tesis investigando la viabilidad práctica de efectuar ataques pasivos contra la privacidad del contenido de las comunicaciones GSM en redes cifradas con A5/1, utilizando para ello SDR de bajo coste y recursos de cómputo modestos.

Dado que la capacidad de interceptar comunicaciones GSM no cifradas bajo estas condiciones ha quedado demostrada en la sección 3.7.1, esta sección se centra en el ataque al algoritmo de cifrado A5/1 en condiciones reales de operación en redes GSM.

A5/1 es un algoritmo de cifrado de flujo que utiliza una clave secreta de 64 bits, denominada K_c, y un vector de inicialización de 22 bits, denominado IV, para generar 228 bits de *keystream*.

En el modo de operación utilizado en GSM, la clave K_c es derivada durante el proceso de autenticación del MS, descrito en la sección 3.3.1, mientras que el IV es calculado como el contador global de tramas en función del número de trama a cifrar o descifrar, según el algoritmo descrito en la figura 3.12.

El cifrado de una trama GSM se realiza ejecutando una operación lógica XOR bit a bit, entre los 114 bits que componen la trama y los primeros o últi-

Algoritmo 3: Cálculo del contador global de tramas

$T1 \leftarrow F_n / 1326$;

$T2 \leftarrow F_n mod 26$;

$T3 \leftarrow F_n mod 51$;

return $(T1 << 11)|(T3 << 5)|T2$;

Figura 3.12: Algoritmo para el cálculo del contador global de tramas

mos 114 bits del *keystream*, dependiendo si la trama es enviada por la BTS o el MS respectivamente. De esta manera, aquellas tramas emitidas por la BTS serán cifradas con los primeros 114 bits, mientras que las emitidas por el MS utilizaran los últimos 114 bits.

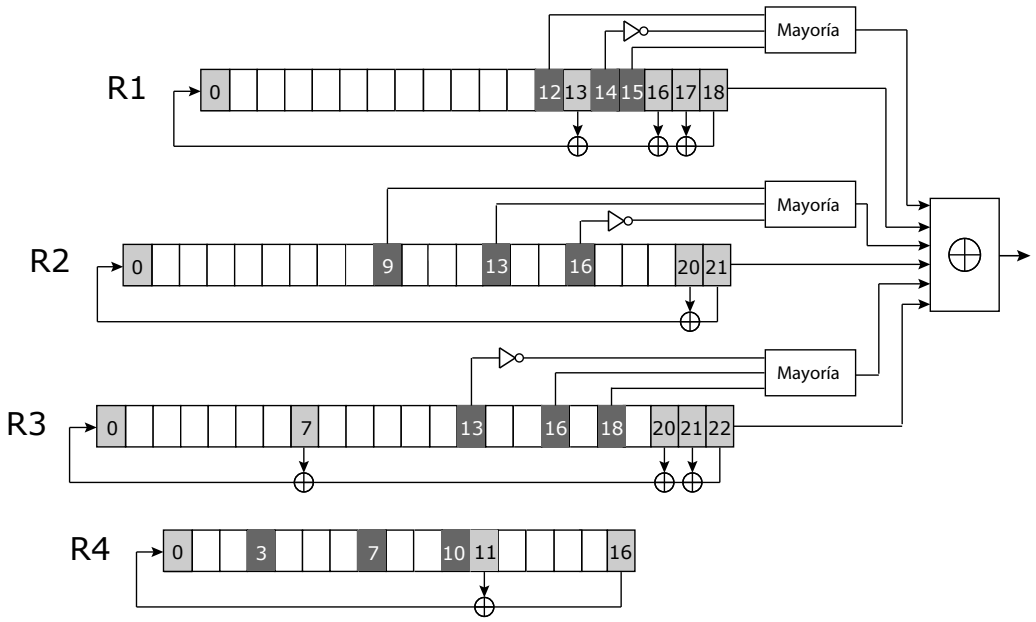


Figura 3.13: Esquema del cifrado A5/2

El algoritmo A5/1 fue diseñado para poder ser implementado de forma eficiente utilizando hardware de bajo coste. A pesar de formar parte integral del estándar GSM, los detalles de A5/1 solo pueden ser obtenidos mediante acuerdos de confidencialidad, algo que imposibilitó totalmente su escrutinio

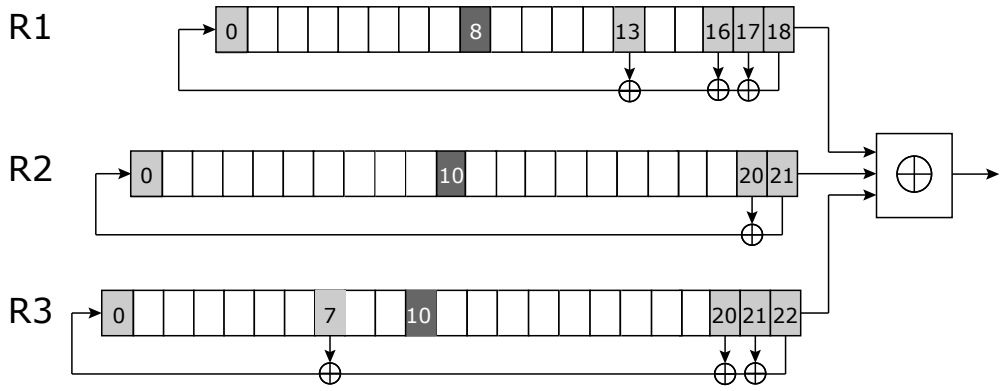


Figura 3.14: Esquema del cifrado A5/1

LFSR	Longitud	Polinomio de retroalimentación
R1	19 bits	$x^{19} + x^{18} + x^{17} + x^{14} + 1$
R2	22 bits	$x^{22} + x^{21} + 1$
R3	23 bits	$x^{23} + x^{22} + x^{21} + x^{8} + 1$

Tabla 3.2: Descripción de los polinomios de los registros de desplazamiento con retroalimentación lineal utilizados en A5/1

por parte de la comunidad académica, hasta que en 1999 Briceno et al [23] publicara por primera vez una implementación de referencia junto con varios vectores de prueba. Acto seguido a dicha publicación, comenzaron a aparecer los primeros ataques criptográficos. La descripción detallada del estado del arte en lo referente a la seguridad de A5/1 se describe en la sección 3.4.

La figura 3.14 detalla la estructura interna del algoritmo de cifrado A5/1. Como se puede apreciar, A5/1 utiliza 3 registros de desplazamiento con retroalimentación lineal, denominados como $R1$, $R2$ y $R3$, con longitudes 19, 22 y 23 bits respectivamente. Los polinomios característicos de los registros de A5/1 se detallan en la tabla 3.2.

A5/1 utiliza un desplazamiento irregular para sus registros, el cual es con-

Registros desplazados	Condición
$R1$, $R2$	$C_1 == C_2 ! = C_3$
$R1$, $R3$	$C_1 == C_3 ! = C_2$
$R2$, $R3$	$C_2 == C_3 ! = C_1$
$R1$, $R2$, $R3$	$C_1 == C_2 == C_3$

Tabla 3.3: Algoritmo de desplazamiento irregular de registros en A5/1

trolado por los valores de ciertos bits de $R1$, $R2$ y $R3$, denominados como C_1, C_2 y C_3, resaltados en fondo negro en la figura 3.14. La tabla 3.3 muestra los resultados de la decisión del desplazamiento irregular para cada uno de los registros en cada ronda de A5/1. Asumiendo una distribución uniforme de los bits de los registros, la probabilidad de que cada registro sea desplazado en la ronda será de 2/3 %.

El estado interno del algoritmo A5/1 se inicializa a partir de la clave K_c de 64-bits y el vector de inicialización IV de 22-bits, en un proceso que dura un total de 88 rondas. El IV se concatena con K_c, donde el bit menos significativo corresponderá a aquel del K_c y el más significativo al del IV. Partiendo de un estado donde todos los registros valen cero, los 88 bits del K_c y el IV se cargan progresivamente, en un total de 88 rondas con desplazamiento regular de los LFSR, mediante una operación lógica XOR con la posición cero de cada registro $R1$, $R2$ y $R3$.

Al estado interno del A5/1 resultante se le denomina estado inicial del A5/1 y depende únicamente de los 64 bits de K_c y los 22 bit de IV utilizados. En un último paso antes de producir el primer bit de *keystream*, se ejecutan un total de 100 rondas con desplazamiento irregular de registros, ignorando los bits producidos. La ronda 101 producirá el primer bit del *keystream* y sucesivas rondas producirán hasta un total de 228-bits.

Una de las mayores vulnerabilidades de A5/1 es precisamente consecuencia del limitado tamaño de su estado interno. En efecto, A5/1 posee un estado interno de 64-bits, bastante más pequeño que el existente en el algoritmo de

cifrado DSC, objeto de análisis en el capítulo 1 de la tesis. La presencia de un estado interno tan reducido ha sido aprovechada en la literatura [14, 116] para proponer un práctico ataque *Time-Memory-Tradeoff* capaz de relacionar las combinaciones del estado interno con el *keystream* correspondiente. Para entender el ataque es necesario comprender la función de los 64-bits de estado interno en la generación del *keystream*.

Como se ha explicado anteriormente, en una primera fase de la ejecución de A5/1, la clave de 64-bits y el vector de inicialización de 22-bits son cargados en el estado interno. Por lo tanto, los 88 bits de entrada a A5/1 son reducidos a un espacio efectivo de 64 bits. En la segunda fase del cifrado, donde un total de 100 rondas son ejecutadas, el espacio interno resultante del proceso, a pesar de continuar siendo de 64-bits, tan solo tendrá una longitud efectiva de 61-bits. Este fenómeno es debido al hecho que en una ronda i, no todas las combinaciones posibles del estado interno para dicha ronda, denominados S_i donde i es el número de ronda, serán posibles.

En efecto, dado que el número de desplazamientos de los registros, en cada ronda, es irregular, siendo dependiente del resultado de la función mayoritaria sobre los valores específicos de ciertos bits de los tres registros, existirán valores S_i para los cuales no exista ningún S_{i-2} o S_{i-3} que satisfaga el resultado esperado de la función mayoritaria. Dicho conjunto de S_i imposibles de generarse, aumenta con la cantidad de rondas totales.

Para ilustrar este fenómeno se ha realizado un experimento donde se generan S_x aleatorios para proceder a realizar el proceso inverso del desplazamiento irregular de registros, explorando iterativamente todos los posibles estados antecedentes de forma recursiva con el objetivo de determinar los posibles estados originales tras 100 desplazamientos inversos.

La figura 3.15 muestra la evolución de la reducción del espacio interno para el rango de desplazamientos inversos 0 - 250. De los resultados experimentales se desprende que tras las 100 rondas del A5/1 previas a la generación del primer bit de *keystream*, el estado interno efectivo se reduce a un 16 %, pasando de 64 al equivalente de 61 bits. Pasadas 244 rondas, en la posición de la gene-

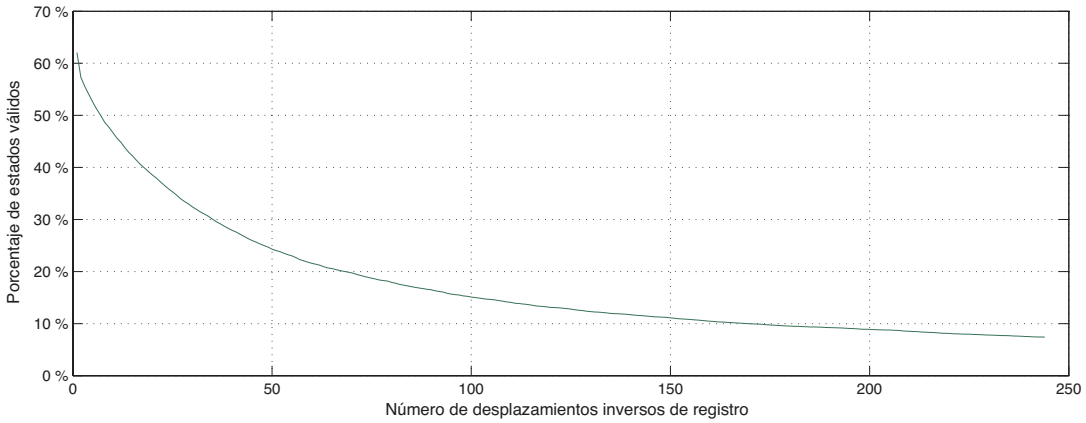


Figura 3.15: Evolución del porcentaje de estados internos válidos en función del número de desplazamientos de registros internos efectuados, en una simulación sobre 10,000 estados aleatorios.

ración del primer bit del *keystream* utilizado para el canal de subida, el espacio interno efectivo se reduce a un 8 %.

La primera propuesta de aplicación práctica de un ataque tipo Time-Memory-Tradeoff en GSM, no tomaba en cuenta este hecho qué sería revelado posteriormente en la lista de correo del proyecto A5/1. Dicho proyecto resultante de la propuesta de Nohl en [112] ha generado en un esfuerzo comunitario, 1.7TB de tablas *rainbow*, que relacionan estados internos posibles de A5/1 con sus 64 bits de *keystream*.

Con el objetivo de medir el grado de efectividad de dichas tablas *rainbow* en la ruptura de claves A5/1 en GSM, se ha realizado un nuevo experimento donde se generan iterativamente claves K_c de forma aleatoria, junto con números de trama también aleatorios, con el objetivo de generar pares de *keystream* para su utilización en canales de subida y baja respectivamente. Por cada *keystream* generado, se utilizan los 1.7 TB de tablas *rainbow* para tratar de establecer el estado interno del A5/1 que ha dado lugar a dicho *keystream*.

Por cada estado interno determinado, se realizan los desplazamientos inversos necesarios para derivar el estado interno anterior a la ejecución de las 100 rondas iniciales donde los bits producidos son descartados. En el caso de los *keystream* del canal de bajada, el número de rondas será de 100, mientras

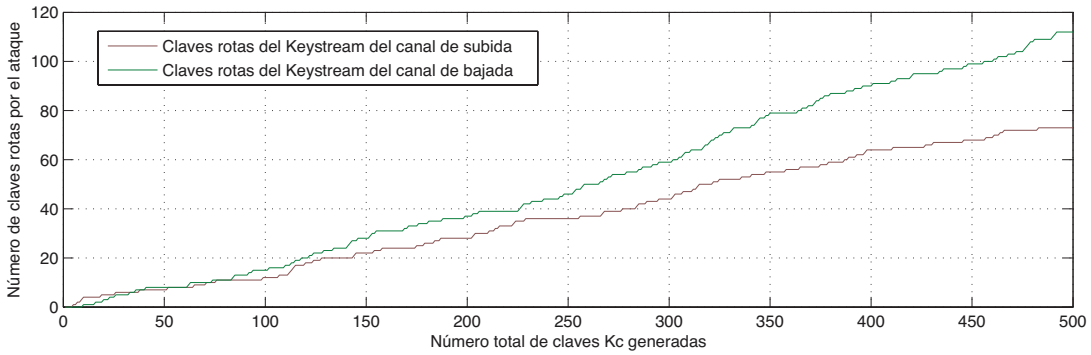


Figura 3.16: Cantidad de claves K_c rotas por el ataque de las tablas *rainbow* para el experimento con un total de 500 claves aleatorias

que en el de los del canal de subida el número de rondas será 244 (100+144).

Una vez determinado el estado interno inicial del A5/1 para un *keystream* determinado, se procede a derivar el K_c correspondiente tomando en cuenta los 22 bits del número de trama utilizado para la generación del *keystream*.

La figura 3.16 muestra la evolución de la cantidad de claves K_c rotas mediante el procedimiento descrito anteriormente, sobre el total de claves K_c generadas. La línea verde muestra las claves rotas mediante el uso del *keystream* del canal de bajada, mientras que la línea roja hace lo propio con el de subida. Se puede apreciar como la efectividad del ataque ronda el 22 % atacando *keystream* perteneciente al canal de bajada y el 15 % cuando se ataca el de subida.

En términos prácticos, la probabilidad de romper una clave utilizando dicho ataque es superior al 80 %, dado que cada paquete GSM se divide en 4 tramas. Si se conoce el contenido en texto plano de un paquete GSM que se observa en su forma cifrada, es posible obtener 4 *keystreams* de dicho paquete.

En GSM existen varios paquetes pertenecientes a información difundida sobre la BTS sobre los parámetros de la célula, cuyo contenido prácticamente no varía en intervalos cortos de tiempo. Previamente al inicio del cifrado, dichos paquetes pueden ser observados en texto plano. Dado que aparecen a intervalos regulares, es posible predecir, en base a su número de trama, cuando volverán a ser transmitidos una vez ha comenzado el cifrado.

La figura muestra el contenido de varios paquetes GSM del tipo *5ter* trans-

Tipo	Nº trama	Contenido
5ter	45484	000203034906064000084ac42000000000000000080040
5ter	49466	000103034906064000084ac42000000000000000080040
5ter	49560	000103034906064000084ac42000000000000000080040

Tabla 3.4: Ejemplo de paquetes GSM con texto plano conocido

mitidos por una BTS [10]. Se puede apreciar que el contenido de todos los paquetes es idéntico, con la excepción del segundo byte.

Dicho byte pertenece al campo *timing advance*, o avance temporal, que es utilizado por la BTS para informar al MSE del retardo estimado de propagación. El MSE utilizará dicha información para transmitir anticipadamente con el objetivo de que el paquete sea recibido dentro de la ranura temporal asignada.

Dada la naturaleza dinámica de dicho parámetro, su valor es susceptible de cambiar constantemente. No existirá forma para el atacante de saber el valor de dicho byte en el paquete cifrado. Sin embargo, dada la reducida cantidad de valores posibles, el mecanismo más efectivo será la generación de varios *keystream* posibles para proceder al ataque individual de cada uno. Cada clave K_c obtenida puede ser confirmada descifrando un segundo paquete GSM. Cada paquete cifrado transmitido por la BTS con texto plano conocido dará lugar a un total de $4 \times ta$ *keystreams*, siendo *ta* el número de valores posibles del campo *timing advance*.

Con el objetivo de ilustrar este ataque en una situación práctica, sin arriesgar la privacidad de usuarios reales de redes GSM, se realiza un experimento utilizando el dataset público ofrecido por los desarrolladores del proyecto *Airprobe*.

En dicho dataset es constituido por la grabación de la señal de radio en I/Q

[10]Con el objetivo de preservar la privacidad de usuarios reales, los paquetes mostrados pertenecen a un dataset público (ruCTF 2013) de la señal de una BTS simulada registrada en formato *I/Q*

de una BTS mientras se realiza una llamada de voz. En el presente experimento se utiliza dicha señal GSM para recrear desde el inicio todo el proceso de ataque a una célula GSM real.

En un primer paso se utiliza el software *Airprobe* para extraer la señalización contenida en la ranura temporal 0 en la configuración combinada *B* (FCCH + SCH + BCCH + CCCH). En el número de trama 862221 se observa un mensaje de asignación donde la BTS emplaza al MSE a utilizar la ranura temporal 1 del ARFC 725 (el actual) sin salto de canal. Dicho mensaje responde a la solicitud de canal por parte del MSE, con referencia 13553, para el establecimiento de la llamada de voz.

Se procede a la decodificación del SDCCH/8 en la ranura temporal 1. En el número de trama 862315, en la sub-ranura 1, se observa el comando de inicio del modo cifrado. Previamente, 70 tramas antes, en la misma sub-ranura se observa un paquetes GSM de información de sistema tipo 5 con número de trama 862245. El contenido del paquete es ”0001030349061d9f6d181080000000000000000000000000” donde el segundo byte indica el valor del *timing advance*, 0x01.

El mensaje de inicio del modo cifrado indica que a partir de la trama número 862315, la comunicación con el MSE se realizará en modo cifrado utilizando el algoritmo A5/1, tal y como se indica en la información contenida en el paquete. Las tramas de tipo 5, que hasta ahora eran transmitidas al MSE en plano (como es el caso de la número 862245), serán transmitidas cifradas.

Como se ha explicado anteriormente, el contenido de dichas tramas de información es estático en intervalos cortos de tiempo, con la excepción del byte perteneciente al *timing advance*. En el caso del mensaje tipo 5 anteriormente referenciado, su contenido informa de los ARFC de las BTS vecinas (en este caso 730, 733, 734, 741 y 746) cuyos valores no cambiarán en sucesivos mensajes del tipo 5.

En GSM, dichos mensajes se retransmiten a intervalos regulares. El siguiente se espera en el número de trama 862449 con un contenido similar, salvo el valor del *timing advance*, pero cifrado. El método más eficaz para la predicción

N. trama	C. trama	*Keystream*
862446	1332352	011010100011101111111010011110011010000 011001000000010011100110001010011111110 111110001100000110101010110100010101111
862447	1332385	011100001111011111111101111011010101010 001101101110100100011001011011110100010 100101100110011101001001101011101110010
862448	1332418	011110100011100011110000000101111100000 010010111110100001000100010100000101010 011011010110110000101111101101111100001
862449	1332451	101111111110010000001001001111100100000 111010001101011011100001111011001100110 010000000110001100101101000001000110000

Tabla 3.5: *Keystream* calculados para ser utilizados en el ataque para la recuperación del K_c

del contenido del nuevo mensaje, a fin de lanzar el ataque de texto plano conocido y recuperar el *keystream* utilizado para cifrarlo, es probar con los valores cercanos al *timing advance* del último mensaje recibido (en este caso 1).

Por lo tanto, se generan 16×4 *keystreams* posibles resultantes de realizar una operación lógica XOR entre el contenido cifrado observado para las tramas 862446, 862447, 862448 y 862449, y su posible contenido en texto plano para los valores de *timing advance* entre 0 y 0x0F. En efecto, cada paquete GSM es transmitido en 4 tramas, por lo que el paquete correspondiente al número de trama 862449 se distribuye en las tramas 862446, 862447, 862448 y 862449.

La tabla 3.5 muestra los *keystream* calculados para el valor de *timing advance* 0, resultado de la operación lógica XOR entre el texto plano esperado y el texto cifrado observado en dichos número de trama. Se realiza una operación similar con el resto de valores del *timing advance*, hasta obtener un total de 64 *keystreams*.

Se lanza el ataque de recuperación del estado interno del A5/1 para cada uno de dichos 64 *keystreams*. Dada la limitada eficacia del ataque, descrita anteriormente y representada en la figura 3.16, cada *keystream* perteneciente a las 4 tramas con el *timing advance* correcto, tendrá cerca de un 16 % de probabilidades de ser roto por el ataque. Para el resto de textos planos incorrectos, existirá una probabilidad baja de obtener falsos positivos.

b7092ab2c95c8632	1ef00bab3bac7002	9f5b403557b2964d	951ad422968807d3
c7566abf4df703e9	65a653acc6d8ae88	6673710d9bee51c9	2bfe626cb5e72b56
7aaf0d4260375851	6f9029615e694697	f47a34bbab586c8e	be6b9347fbc14e3e
32aaa8b1e64eabdc	d294fb047f8899e2	623825688d4202cc	4a71638438c45644
1af8c23678a0bba3	3a46173afc346376	93bf36230ec49546	890e5b7f3ec88637
b4932ec3375b57a9	e40d1832aac648c7		

Tabla 3.6: Posibles valores de K_c para el escenario del ataque pasivo a A5/1

Con el objetivo de descartar dichos falsos positivos, por cada *keystream* para el cual sea determinado un estado interno, se recuperan los K_c posibles y se eliminan falsos positivos verificando el descifrado de otro *keystream* perteneciente al mismo paquete.

Siguiendo este procedimiento se determina que el *timing advance* correcto es el 0 (siendo 1 el del anterior paquete tipo 5 sin cifrar). De los 4 *keystreams* mostrados en la tabla 3.5, el ataque únicamente ha podido romper el cuarto, determinando que el estado interno con valor $d5eb21665d2b8f25$ genera el citado *keystream* en su posición 13 (encontrado en la tabla 172).

Una vez determinado el valor del estado interno, se procede a desplazar inversamente todos los registros del A5/1 un total de 113 veces, ya que el *keystream* es perteneciente al canal de bajada y el ataque lo ha encontrado en la posición 13. Finalmente se derivan los K_c posibles para su contador de trama, el 1332451. La tabla 3.6 muestra los 21 K_c posibles que producen el citado *keystream*.

Utilizando el *keystream* con número de trama 1332418, para el mismo *timing advance*, se prueban todos los posibles K_c para encontrar aquel que produce dicho *keystream* correctamente para el mencionado número de trama. Mediante este procedimiento se determina que el K_c correcto es el $1ef00bab3bac7002$.

Una vez obtenida la clave con la que se ha cifrado la comunicación con el MSE, se procede a descifrar los paquetes cifrados, en la sub-ranura 1, posteriores al número de trama 862315 donde comenzaba el cifrado. Se observa un comando de asignación con número de trama 862519 donde se emplaza al

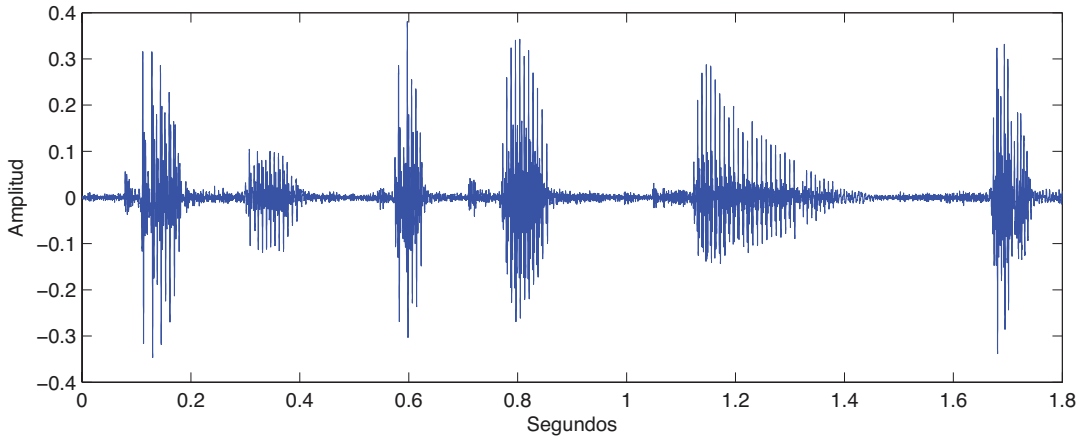


Figura 3.17: Audio extraído de la comunicación cifrada

MSE a la utilización de la ranura temporal 5 para la transmisión de la voz de la llamada. La llamada era formalmente iniciada en el número de trama 862366, también cifrado, donde se configuraba la conexión y se enviaba el número llamante.

Con el objetivo de recuperar el audio transmitido en la llamada, se decodifica a continuación la ranura temporal del mismo ARFC, descifrando los paquetes con el mismo valor de K_c. El resultado es la voz codificada en GSM RPE-LTP. La figura 3.17 muestra el audio extraído de la conversación simulada.

3.7.3. Conclusiones

En esta sección se han evaluado experimentalmente diversos ataques contra la seguridad y privacidad de las comunicaciones efectuadas sobre redes GSM. En línea con los objetivos de la tesis, descritos en la sección 1.1 del capítulo 1, el propósito perseguido con los experimentos realizados es determinar el grado efectivo de seguridad y privacidad de las comunicaciones efectuadas sobre el protocolo GSM, ampliamente difundido a nivel europeo y mundial, a la vista de la amplia disponibilidad de dispositivos SDR de bajo coste y nuevas técnicas para el ataque a los mecanismos criptográficos.

A la luz de los resultados experimentales obtenidos, se puede concluir que

el protocolo GSM ofrece unas garantías alarmantemente escasas para la protección de la privacidad de los usuarios y sus comunicaciones. Ha quedado demostrado que prácticamente cualquier adversario, independientemente de su nivel de recursos, es capaz de llevar a cabo ataques activos muy efectivos para la captura de identidades de usuario GSM y la interceptación de sus comunicaciones de voz y datos.

De la misma manera, se ha demostrado que la interceptación pasiva de comunicaciones cifradas con A5/1, utilizado en la práctica totalidad de las redes de los operadores GSM europeos, es factible utilizando SDR de bajo coste e implementaciones de ataques documentados en la literatura, las cuales han demostrado ser efectivas en escenarios reales y con recursos limitados para llevar a cabo el ataque.

Dada la amplia difusión de la tecnología GSM en Europa, las conclusiones obtenidas en la investigación realizada en este capítulo, suponen una importante contribución a la validación de la hipótesis formulada en la tesis y la consecución de los objetivos propuestos.

Capítulo 4

WiFi

Este capítulo de la tesis se centra en la investigación del grado efectivo de seguridad y privacidad de las comunicaciones personales efectuadas sobre redes inalámbricas basadas en el protocolo IEEE 802.11, también conocidas como redes WiFi.

La hipótesis y objetivos perseguidos son los descritos en la sección 1.1 del capítulo 1, aplicados a la tecnología WiFi, y la metodología aplicada será aquella definida en la sección 1.2 del mismo capítulo.

En el caso particular de las redes WiFi, la viabilidad del abuso e interceptación de comunicaciones no cifradas por parte de terceros con escasos recursos, se encuentra ya establecida dada la amplia gama de herramientas actualmente disponibles a tal efecto.

Debido a ello, la práctica totalidad de redes inalámbricas WiFi desplegadas en entornos residenciales utilizan cifrado con el objetivo de proteger la seguridad y privacidad de las comunicaciones. Por lo tanto, la contribución del presente capítulo a la validación de la hipótesis de la tesis y consecución de sus objetivos, se centrará directamente en la investigación de la efectividad de las medidas de cifrado previstas en el estándar.

Primeramente, se realiza una investigación sobre la seguridad del cifrado WEP soportado por las redes WiFi [1]. Para ello, se investigan diferentes ataques

[1]La investigación relativa a WEP fue realizada al inicio de la investigación de esta tesis, en un contexto donde no existían herramientas capaces de atacarlo de forma efectiva y en

de criptoanálisis y se desarrolla una herramienta, denominada como *Weplab*, que implementa de forma práctica una serie de ataques criptográficos contra WEP, siendo capaz de criptoanalizar la clave de cifrado WEP analizando tráfico cifrado en situaciones de despliegue reales.

A continuación, se explora la aplicación práctica de ataques de canal lateral basados en el análisis de tráfico cifrado y se demuestra cómo ninguno de los cifrados actualmente soportados por el protocolo 802.11 es capaz de ofrecer una protección efectiva de la privacidad de las comunicaciones ante este tipo de ataques. Para ello, se diseñan una serie de experimentos donde se busca establecer la viabilidad de realizar este tipo de ataques para la determinación de los sitios web visitados por un nodo de una red WiFi cifrada, sin utilizar para ello excesivos recursos.

4.1. Estado del arte

4.1.1. Ataques a los algoritmos de cifrado y autenticación

El algoritmo de cifrado RC4, utilizado para la protección de la privacidad de las comunicaciones en el protocolo WEP, fue diseñado por Ron Rivest en 1987, siendo inicialmente su diseño opaco al escrutinio de la comunidad académica, dado que se encontraba protegido por secreto comercial. En [5] se publicó de forma anónima en Internet una implementación software del algoritmo, supuestamente derivada mediante ingeniería inversa de una implementación software o hardware.

El primer criptoanálisis del cifrado RC4 sería publicado en [134] y [158], donde los autores demuestran experimentalmente correlaciones entre los primeros bytes del *keystream* y combinaciones lineales de bytes de la clave RC4 utilizada.

En [78], RC4 fue adoptado oficialmente por IEEE como algoritmo de cifrado en el estándar WEP, cuyo propósito sería garantizar la privacidad de las comunicaciones inalámbricas del protocolo 802.11 con el objetivo de equipar

ausencia de dispositivos en el mercado que implementara el nuevo cifrado WPA

su privacidad y seguridad a las comunicaciones mediante efectuadas mediante cable.

El primer análisis crítico de la seguridad del estándar WEP fue publicado por [21], donde los autores repasan las diversas vulnerabilidades existentes en el diseño del protocolo, muchas de las cuales serían posteriormente demostradas en ataques activos prácticos contra comunicaciones 802.11 cifradas con WEP. El mismo año, en [111], se presenta una implementación de varios ataques de fuerza bruta y diccionario contra claves WEP de 40-bit y se demuestran vulnerabilidades en el algoritmo de conversión de la contraseña suministrada por el usuario en la clave WEP de ciertas implementaciones, que permiten la reducción del espacio de estados a ser explorado.

Posteriormente, en [62], se publica el que sería el primer ataque de solo texto cifrado contra las comunicaciones inalámbricas protegidas por WEP, que sería denominado en lo sucesivo como el ataque FMS. El ataque descrito utiliza un byte de texto plano predecible derivado de las cabeceras de las tramas para derivar el primer byte del *keystream*, resultado de la ejecución del cifrado RC4 con la clave (de 40 o 104 bits) y el vector de inicialización correspondiente a cada paquete analizado. Los autores demuestran la existencia de las llamadas *claves débiles*[2], dentro del protocolo de cifrado RC4 en su aplicación a WEP, que son aquellas que, debido a ciertos valores presentes en el vector de inicialización, posibilitan la utilización de una ecuación lineal capaz de correlacionar bytes de la clave con el primer byte de *keystream*, con una probabilidad de acierto de un 5 %. Los autores describen la posibilidad de derivar la clave de cifrado WEP capturando gran cantidad de tráfico cifrado, con el objetivo de obtener suficientes paquetes cifrados con *IV débiles* como para poder realizar el ataque estadístico.

Los autores declaran explícitamente en su artículo que su investigación tiene un carácter teórico y no ha sido probada experimentalmente en la práctica contra comunicaciones cifradas con el protocolo WEP. La primera implemen-

[2]La existencia de *claves débiles* ya había sido revelada anteriormente de forma genérica para RC4 por [134] y [158]

tación pública del ataque sería demostrada por [129], seguida por [67] [3], donde los autores demuestran cómo es posible recuperar una clave WEP de 104-bits mediante el análisis de una gran cantidad de paquetes cifrados, del orden de 4 millones para alcanzar una probabilidad del 50 % de recuperación de clave.

En [75] se presenta un análisis crítico del trabajo de [62] y de las implementaciones existentes por utilizar un modo de operación muy limitado al determinar claves débiles mediante patrones estáticos en el IV utilizado, tal y como era sugerido en el trabajo de [62]. En su artículo, el autor describe la extensión el ataque FMS a otros bytes del *keystream* y propone una nueva implementación del ataque FMS tomando en cuenta también el segundo byte de *keystream*. Su implementación es capaz de romper la clave utilizando de 500.000 a 2 millones de IVs.

En [150] y [149] se describe una implementación del ataque FMS que no sería liberada y, al igual que en [75], se sugiere una optimización del ataque FMS original para el caso en el que el primer byte de *keystream* dependa de 2 valores en lugar de 3. El ataque es refinado en [17], donde el autor continúa el trabajo previo de [75, 150] y analiza este último caso especial del ataque FMS, a la vez que determina nuevas clases de claves débiles no dependientes de valores del vector de inicialización.

En [136] se publica la primera versión de *Weplab*, la implementación del ataque a WEP desarrollada por el autor de la presente tesis y descrita en la sección 4.2. La versión inicial de *Weplab* implementa el ataque FMS extendido al primer y segundo byte del *keystream*. De esta forma se convierte en la primera implementación para GNU/Linux capaz de derivar la clave WEP con un rendimiento similar a la implementación descrita por [75]. Poco después, se publica *Aircrack* [38], el cual implementa un nuevo ataque activo capaz de acelerar la recolección de muestras, re-inyectando tráfico cifrado capturado de forma previa [4] con el objetivo de generar paquetes cifrados de respuesta por

[3] Stubblefield et al publicaría posteriormente los resultados de una implementación [150], la cual nunca sería liberada.

[4] WEP no ofrece protección contra ataques de repetición de tráfico, por lo que es posible para un atacante capturar un paquete cifrado en un momento t y re-inyectarlo en la red en

parte de nodos de la red WEP.

Posteriormente, en [88], se publicarían de forma anónima un conjunto de ecuaciones lineales, que serían conocidas en adelante como el ataque KoreK [5], comprendiendo todas las ecuaciones descritas previamente en [75] y [17], y proponiendo otras clases nuevas. Las nuevas clases de ataques fueron implementadas de forma independiente en *Weplab* y *Aircrack*, mejorando el rendimiento de ambas implementaciones y posibilitando ataques prácticos a transmisiones de datos cifradas por WEP analizando de 150.000 a 300.000 muestras. En la sección 4.2 se dan más detalles sobre el criptoanálisis implementado en *Weplab* y sus resultados experimentales. El conjunto de ecuaciones lineales propuestas por KoreK fueron posteriormente analizadas en detalle en [28], donde el autor analiza, a su vez, las implementaciones realizadas en *Weplab* y *Aircrack*.

En [87] se publica un nuevo ataque activo contra WEP, que sería denominado en lo sucesivo como ataque *chopchop*. El ataque es una aplicación invertida del ataque de inducción descrito por [7], donde se utiliza a un nodo de la red como oráculo, de tal forma que el atacante pueda inyectar un paquete cifrado y saber si éste posee un CRC válido tras su descifrado. De esta manera, partiendo de un paquete cifrado con un IV determinado, es posible recuperar su *keystream* correspondiente, lo cual no solamente posibilita el descifrado del mismo, sino también el cifrado e inyección de cualquier paquete en la red, utilizando el mismo IV y *keystream*.

En respuesta a los ataques prácticos existentes contra el protocolo WEP, la *IEEE Alliance* publica dos nuevos estándares para el cifrado de las comunicaciones en redes WiFi 802.11, que serán conocido como WPA y WPA2 [79]. WPA es concebido como una solución puramente software que utiliza el protocolo RC4 de WEP, unido a una renovación automática de claves y protecciones adicionales contra ataques de repetición, de tal manera que sea compatible con las implementaciones hardware existentes de WEP. Por otro lado, WPA2 introdu-

$t + n$ para cualquier valor de n

[5]Las ecuaciones fueron enviadas por correo electrónico al autor de esta tesis mientras realizaba la investigación en el criptoanálisis de WEP, descrita en la sección posterior del presente capítulo, y publicadas posteriormente en un conocido foro público de seguridad [88]

ce AES como algoritmo de cifrado utilizado para la protección de las comunicaciones. Actualmente WPA y WPA2 forman parte de las especificaciones del estándar IEEE 802.11 [80].

Un nuevo conjunto de vulnerabilidades WEP, denominados como ataques de fragmentación, son descritos en [18], donde los autores determinan como utilizar el soporte para la fragmentación de paquetes en el protocolo 802.11 con el objetivo de poder inyectar paquetes de tamaño $(n-4) \times 16$ poseyendo n bytes de *keystream*, mediante su separación en 4 fragmentos con idéntico IV. Utilizando este ataque en combinación con los ataques pasivos al cifrado RC4, es posible atacar a una red WEP y recuperar la clave de cifrado en cuestión de minutos.

La nueva generación de ataques contra el algoritmo de cifrado RC4 aplicado a WEP, vienen de la mano de [98], [82] y [86]. El nuevo tipo de ataque presentado no depende de la existencia de claves débiles, por lo que las correlaciones encontradas pueden aplicarse a mayor cantidad de paquetes cifrados. En [153] los autores describen una extensión del ataque reduciendo la dependencia sobre bytes previos de la clave y realizan una implementación que será conocida como el ataque PTW. En las pruebas experimentales, su implementación es capaz de romper una clave WEP de 104-bit en 40.000 muestras, con una probabilidad del 50 %, o en 65.000 muestras, con un 90 % de probabilidad.

Posteriormente, en [125], se descubren nuevos ataques para los primeros 3 bytes de la clave, dependientes del primer byte de *keystream*. En [156] se identifican dos nuevas debilidades y se describe el ataque VX, capaz de criptoanalizar una clave WEP con una probabilidad de éxito superior al 50 %, analizando tan solo 2^{15} paquetes cifrados en condiciones ideales.

Un nuevo ataque, publicado en [152], permite reducir la cantidad de paquetes cifrados a 24.200 para obtener una probabilidad del 50 % de romper la clave WEP. La cifra sería de nuevo reducida a 9.800 en [141], donde los autores proponen un nuevo enfoque experimental basado en *caja negra* para la búsqueda de relaciones lineales en WEP.

Tras algunas publicaciones previas [61, 104], con escaso o nulo impacto práctico en la seguridad de WPA, en [107] se describe el primer ataque con dimensión práctica contra los protocolos WPA y WPA2, en su modalidad WPA-PSK[6]. El ataque es implementado inicialmente por las herramientas [135] y [165], y posteriormente por [96] y [39]. El ataque consiste en la observación pasiva de proceso de autenticación de un cliente legítimo en la red WPA con el objetivo de lanzar una búsqueda exhaustiva fuera de línea de la clave PSK. Dado que una búsqueda exhaustiva de la clave de 2^{256} sería totalmente inviable, el ataque utiliza un diccionario y realizar el proceso de generación de la clave PSK partiendo de la frase de paso, consistente en el cálculo de 4096 iteraciones de HMAC-SHA1 sobre el ESSID y un *salt*. En [133] se propone el cálculo previo de este último proceso, mediante una técnica de compromiso espacio-tiempo, con el objetivo de acelerar el proceso, a pesar de que las tablas generadas sean dependientes del nombre (ESSID) de la red WPA[7].

En [152], los autores extienden una vulnerabilidad existente en WEP [87] para atacar redes protegidas con WPA-TKIP, mediante la recuperación del *keystream* utilizado para el cifrado de un paquete determinado y su utilización posterior para inyectar hasta un total de 7 nuevos paquetes arbitrarios en la red. El ataque requiere que la red soporte Calidad de Servicio 802.11e (QoS) y posea una periodicidad de renovación de claves relativamente larga. El ataque tiene una duración variable de entre 12 y 15 minutos. Posteriormente, el ataque es mejorado en [119], donde se describe la utilización de un escenario de hombre en el medio (MiTM) para lanzar un nuevo ataque de falsificación de mensaje, donde el soporte de 802.11e QoS en la red objetivo del ataque ya no es necesario. El tiempo de ejecución requerido por el nuevo ataque es del

[6]WPA-PSK y WPA2-PSK son las modalidades de despliegue usadas en la práctica totalidad de las instalaciones WiFi domésticas. En WPA-PSK el punto de acceso y los clientes comparten una clave secreta a modo de *secreto compartido*.

[7]El proceso de compromiso espacio-tiempo descrito tiene una utilidad cuestionable, no solamente por el hecho de ser dependiente del un ESSID determinado, sino por el hecho de que realmente se tratan de tablas de búsqueda ordinarias, que relacionan frases de paso candidatas con sus equivalentes claves PSK de 256-bit

orden de 1 a 2 minutos. Posteriormente en [70], los autores presentan un ataque similar, donde se descifran paquetes ACK de DHCP para posibilitar la inyección de paquetes de un tamaño superior de hasta 596 bytes.

En [11] se describe de forma teórica una vulnerabilidad de WPA-PSK que permite el descifrado de cualquier paquete transmitido desde el punto de acceso al cliente, utilizando una vulnerabilidad encontrada en el algoritmo *Michael* que permite la inserción de contenido en un paquete, de tal forma que se pueda mantener su código *Michael* constante al añadir datos una cadena de datos determinada. Al igual que sucedía en [152], el ataque requiere de la presencia de *802.11e QoS* en la red WPA. El ataque sería posteriormente demostrado experimentalmente en [155] sobre escenarios reales de redes WPA.

Más recientemente, en [1], se describe una nueva vulnerabilidad en WPA denominada *Hole 196* [8]. Un atacante en posesión de la clave de la red, y por lo tanto capaz de recibir y enviar paquetes a la misma, es capaz de utilizar la clave de red de grupo (GTK) para realizar mediante ARP del tipo MiTM y recibir descifrados por parte del punto de acceso, paquetes cifrados de otros clientes.

En [157] y [144], los autores descubrían de forma independiente una interesante y efectiva vulnerabilidad en el diseño e implementaciones del protocolo WPS que permite, mediante un ataque activo de varias horas de duración, la recuperación de la clave PSK de una red WiFi protegida con WPA-PSK o WPA2-PSK. Más recientemente, en [20], se descubre una nueva vulnerabilidad presente en gran cantidad de implementaciones del protocolo WPS por parte de los diferentes fabricantes, basada en la generación de número pseudoaleatorios predecibles por un atacante. Utilizando dicha vulnerabilidad es posible recuperar la clave PSK de un punto de acceso vulnerable de forma casi instantánea.

Con la excepción del ataque de diccionario contra la autenticación del clien-

[8] La denominación *Hole 196* se deriva de la página del estándar 802.11 (página 1232, IEEE 802.11 Standard, Revisión 2007), donde se encontraban las especificaciones que daban lugar a la vulnerabilidad.

te a la red WPA, los ataques existentes contra WPS y las contraseñas por defecto de las redes WPA [40], el resto de ataques descritos por la comunidad académica ofrecen importantes dificultades en su aplicación práctica, ya sea por los requisitos, complejidad de ejecución o tiempo. Por ello, se concluye que actualmente WPA y WPA2 ofrecen un grado adecuado de protección de las comunicaciones WiFi domésticas, siempre y cuando la frase de paso sea lo suficientemente compleja, no encontrándose basada en parámetros observables de la red, y WPS se encuentre completamente desactivado.

4.1.2. Ataques de fuga de información en WiFi

Una de las primeras aproximaciones genéricas al empleo práctico de análisis de tráfico para implementar ataques del tipo canal lateral capaces de afectar la privacidad de las comunicaciones cifradas, fue descrita por [32], seguida de [151] y [74]. En dichas publicaciones, los autores describen diversas vulnerabilidades existentes en varios protocolos de comunicación cifrada de datos y demuestran ataques efectivos contra comunicaciones cifradas por TLS/SSL, capaces de determinar los sitios web visitados por un usuario determinado. A su vez, los autores proponen diversas contramedidas de carácter genérico orientadas a mitigar el riesgo de dicho tipo de ataques. El ataque de otros protocolos de comunicación cifrada ya había sido previamente descritos en [145], donde se demostraba la viabilidad de inferir el tamaño de una contraseña de usuario mediante el análisis de los tamaños de paquetes y latencias en conexiones cifradas SSH.

En [16] se describe la utilización de los tamaños de paquetes cifrados y el tiempo relativo de llegada de los mismos, como base para el análisis de tráfico, con el objetivo de la identificación de los sitios webs visitados bajo una conexión cifrada SSL entre el cliente y un proxy. Los autores también mencionan la posibilidad de aplicar el mismo método para conexiones WEP. Por el contrario, en [93] se demuestra cómo el tiempo relativo de llegada no es imprescindible para realizar una clasificación efectiva de los sitios web visitados bajo una conexión cifrada. Los autores describen dos clasificadores, *Naive Ba-*

yes y *Jaccard's coheficient*, para la clasificación del tráfico cifrado únicamente en base al tamaño de los paquetes observados.

En [139] se demuestra la aplicación de técnicas de análisis de tráfico para la determinación del contenido multimedia que es transmitido en tiempo real de forma cifrada por un dispositivo inteligente [9] perteneciente al Internet de las Cosas (IoT). Posteriormente, en [164], se demuestra un ataque de la misma naturaleza, capaz de detectar determinadas frases habladas dentro de una conexión VoIP cifrada.

En [31] se describe un ataque de canal lateral contra comunicaciones cifradas con el objetivo de determinar las acciones que un usuario de la red toma en una determina aplicación web. Los autores demuestran cómo es posible determinar las búsquedas efectuadas por un usuario en Google mediante el análisis del orden y tamaño de paquetes cifrados intercambiados por un host determinado de una red WPA/WPA2.

Posteriormente, en [73], los autores describen un método genérico para la determinación de sitios web visitados bajo conexiones cifradas, basado en la utilización de un clasificador Naïve-Bayes multinomial sobre los tamaños de los paquetes IP. HMM es posteriormente propuesto como clasificador en [27], donde se utiliza para la determinación del sitio web, en función de las diferentes secciones de la web que son accedidas, las cuales son clasificadas utilizando distancias *Damerau-Levenshtein* sobre una representación intermedia de las trazas generadas por los tamaños de los paquetes y su orden derivados de la carga de páginas individuales.

Más recientemente, en [103], se describe un efectivo ataque práctico de análisis de tráfico sobre conexiones SSL cifradas, capaz de determinar las secciones que un usuario visita en un sitio web determinada. Los autores presentan un escenario realista de ataque donde el estado de salud del usuario puede ser inferido en función de las páginas visitadas por el mismo en la web

[9] El dispositivo analizado se llama *SlingBox* y ofrece la capacidad de transmitir en tiempo real, de forma cifrada, a un PC o dispositivo móvil el contenido de audio/vídeo de una entrada analógica o digital.

Algoritmo 4: Algoritmo de preparación de clave RC4 (KSA)

for $i = 0$ *to* 255 **do**
    $S[i] \leftarrow i$;
$j \leftarrow 0$;
for $i = 0$ *to* 255 **do**
    $j \leftarrow (j + S[i] + K[i \ mod \ len(K)]) \ mod \ 256$;
    $S[i] \leftrightarrow S[j]$;

Figura 4.1: Algoritmo de preparación de clave RC4 (KSA)

del hospital.

4.2. Criptoanálisis del algoritmo de cifrado WEP

4.2.1. El algoritmo de cifrado RC4 en WEP

RC4 es un algoritmo de cifrado de flujo que toma una clave de longitud variable y genera un flujo de bits pseudo-aleatorio de longitud arbitraria, denominado *keystream*. Las operaciones de cifrado y descifrado son realizadas mediante la ejecución de una operación lógica XOR, bit a bit, entre el *keystream* y el texto plano o cifrado respectivamente.

El algoritmo de cifrado posee un estado interno, (S, i, j), compuesto por dos punteros de 8 bits cada uno, denominados i y j, y un vector de 256 bytes, denominado S, donde el valor de cada elemento es único en todo el vector. El proceso de generación de *keystream* se compone de 2 pasos principales, el algoritmo de preparación de la clave (KSA), encargado de utilizar la clave para generar el estado interno inicial del algoritmo (S_0), y el algoritmo de generación pseudo-aleatoria (PRGA), cuya misión es generar el *keystream* utilizando, y actualizando, el estado interno (S_i, i, j).

La figura 4.1 describe en pseudocódigo el algoritmo KSA. Primeramente se

Algoritmo 5: Alg. de generación pseudo-aleatoria (PRGA)

$i \leftarrow 0$;

$j \leftarrow 0$;

while $i = 0$ *to* $len(keystream)$ **do**

  $i \leftarrow (i + 1)$ *mod* 256;

  $j \leftarrow (j + S[i])$ *mod* 256;

  $S[i] \leftrightarrow S[j]$;

  $O[i] \leftarrow S[(S[i] + S[j])$ *mod* 256];

Figura 4.2: Algoritmo de generación pseudo-aleatoria RC4 (PRGA)

inicializa el vector S con valores incrementales de 0 a 255. A continuación, se procede a permutar S en función del valor de la clave, recorriendo iterativamente los 256 elementos de S e intercambiado cada valor con el apuntado por una posición determinada en función uno de los bytes de la clave, según se describe en pseudocódigo en la figura 4.1.

Como resultado, el vector S contendrá una permutación de los valores de 0 a 255, donde cada valor aparecerá en una única posición del array.

Por cada byte de *keystream* producido por el algoritmo PRGA, se actualiza primeramente el estado interno del algoritmo (S, i, j), compuesto por el vector S y los punteros i y j inicializados a cero tras el algoritmo KSA. Todas las operaciones de suma y resta serán calculadas en módulo 256, i denota el número de byte de *keystream* a ser producido (empezando por 1) y S_i denota el valor del vector S listo para la producción del byte i de *keystream*. Tras la actualización de (S, i, j), el byte número i de *keystream* se calcula como $S_i[(S_i[i] + S_i[j])]$, donde $S_i[x]$ denota el valor del byte en la posición número x del vector S_i.

Previamente, el estado interno (S, i, j) se actualiza incrementando i en 1, recalculando j como $j + S_{i-1}[i]$ y actualizando S (de S_{i-1} a S_i) mediante el intercambio de los valores en las posiciones $S_{i-1}[i]$ y $S_{i-1}[j]$. La figura 4.2 describe en pseudocódigo el algoritmo PRGA para la producción de los bytes de

keystream.

WEP utiliza el cifrado RC4, siempre en módulo 256, para la protección de la confidencialidad de las comunicaciones, junto con un CRC de 32-bits para garantizar su integridad, denominado ICV. En el modo de operación de WEP, las claves utilizadas en RC4 pueden tener la longitud de 64 o 128 bits, de los cuales los 24 bits iniciales son conformados por el vector de inicialización, denominado IV, que viaja en texto plano en la cabecera del paquete. El resto de la clave, cuya longitud de 40 o 104 bits determina la longitud efectiva de la clave, es denominada R_k y permanece estática a no ser que sea cambiada por el usuario en todos los dispositivos de la red.

El proceso de cifrado y descifrado en WEP se realiza componiendo la clave correspondiente del paquete concatenando el IV con R_k ($IV||R_k$), ejecutando el algoritmo de cifrado RC4 para producir el *keystream* correspondiente y realizando la operación lógica XOR contra los datos a cifrar y descifrar.

4.2.2. Ataque FMS

En [62] se describe un ataque de solo texto cifrado contra comunicaciones inalámbricas 802.11 cifradas con WEP, que sería conocido en lo sucesivo como el ataque FMS.

Un atacante que intercepte pasivamente tráfico WEP cifrado puede fácilmente determinar el primer byte del *keystream* utilizado para cifrar dicho paquete, dado que el primer byte del paquete WEP en texto plano tendrá siempre el valor 0xAA, ya que dicho valor perteneciente a la cabecera SNAP es una constante.

Un byte de *keystream* es generado por el algoritmo PRGA de RC4 como $S[(S[i] + S[j])]$ y dependerá por lo tanto de 3 valores: $S[i]$, $S[j]$, y $S[S[i] + S[j]]$, contenidos en las posiciones i, j y $S[i] + S[j]$ respectivamente.

En el caso del primer byte de *keystream*, denominado como o_1, los punteros i y j toman los valores 1 y $S[1]$ respectivamente. Por lo tanto, los valores $S[i]$, $S[j]$ y $S[S[i] + S[j]]$, serán respectivamente $S[1]$, $S[S[1]]$ y $S[(S[1] + S[S[1]])]$.

Dado que en WEP los 3 bytes del IV que viajan en plano en la cabecera

de cada paquete son los 3 primeros bytes de la clave RC4, es posible ejecutar con sus valores las 3 primeras rondas del algoritmo KSA. En estas 3 primeras rondas, i habrá tomado los valores 0, 1 y 2, intercambiando las respectivas posiciones del vector S actualizándolo de S_0 a S_2, siendo S_x el estado del vector S en la ronda x de KSA. Si las posiciones 1, $S_2[1]$ y $S_2[1] + S_2[S_2[1]]$ son menores que 3, S_2 se considera un estado resuelto ya que la probabilidad de que al finalizar las 256 rondas de KSA sus valores no hayan sido alterados mediante un nuevo intercambio que afecte a sus posiciones en S, será de $e^{-3} \simeq 5\,\%$.

En este último caso, se puede establecer la siguiente ecuación lineal que relaciona al primer byte de *keystream*, o_1, con las mencionadas posiciones de S.

$$o_1 \quad = \quad S_2[S_2[1] + S_2[S_2[1]]]$$

Si $S_2[1] + S_2[S_2[1]] = 3$ entonces, en la siguiente ronda del algoritmo KSA, la posición que contendrá el valor o_1 al final de la ronda, será intercambiada por el valor contenido en la posición apuntada por el puntero j en la ronda 3, denominado como j_3, cuyo valor es calculado como $j_2 + S_2[3] + K[0]$, siendo j_x el valor del punto j en la ronda x del KSA y $K[0]$ el primer byte de la clave WEP.

Por lo tanto, la ecuación anterior, en el caso de que las condiciones $S_2[1] + S_2[S_2[1]] == 3$ y $S_2[1] < 3$ se cumplan, se puede derivar la siguiente ecuación lineal sobre el primer byte de la clave, donde $S_2^\star[x]$ denota la posición que ocupa el valor x en el vector S_2:

$$K[0] \quad = \quad (S_2^\star[o_1] - j_2 - S_2[3])$$

Dicha ecuación lineal se cumplirá el $e^{-3} \simeq 5\,\%$ de las veces siempre que el estado S_2 se considere resuelto.

El caso particular descrito para el primer byte de la clave, se puede generalizar a cualquier byte. Si se cumplen la condiciones $S_{p-1}[1] < p$ y $(S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]) = p$, siendo p el número de byte de la clave RC4 (cuyos 3 primeros bytes son el IV), la ecuación $K_0 = (S_{p-1}^\star[o_1] - j_2 - S_{p-1}[3])$ se verifica con una probabilidad del $e^{-3} \simeq 5\,\%$.

En el artículo que describe el ataque FMS [62], se define una clave débil como aquella cuyo IV cumple con el patrón $(p, 255, X)$, donde X puede tomar cualquier valor. En efecto, si $IV_0 = p$ y $IV_1 = 255$ entonces, dado el algoritmo KSA, necesariamente $S_{p-1}[1] < p$ y $(S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]) = p$.

Detectando los paquetes que contengan un IV que de lugar a una clave débil para un byte determinado de la clave WEP, es posible utilizar la ecuación lineal para calcular el valor correcto de dicho byte. Dado que la ecuación se verifica con una probabilidad de $e^{-3} \simeq 5\,\%$, tras el análisis de varias muestras con diferentes IV débiles para el mismo byte, se puede estimar como correcto aquel que haya sido determinado en más ocasiones.

El ataque FMS fue implementado en [129] y [67], donde se utilizaba el patrón de IV, $(p, 255, X)$, para la determinación de claves débiles y se realiza el proceso de votación descrito anteriormente para la determinación de todos los bytes de la clave.

4.2.3. Más allá del ataque FMS

El método para la detección de claves débiles, en base al patrón de IV descrito en [62], fue utilizado por fabricantes y desarrolladores de controladores para prevenir la generación de IV que dieran lugar a claves débiles. Un ejemplo de ello se encuentra en el kernel de Linux donde la prevención de claves débiles se realiza de utilizando el siguiente código[10].

```
static inline bool ieee80211_wep_weak_iv(u32 iv, int keylen)
{
        /*
        * Fluhrer, Mantin, and Shamir have reported weaknesses in the
        * key scheduling algorithm of RC4. At least IVs (KeyByte + 3,
        * 0xff, N) can be used to speedup attacks, so avoid using them.
        */
        if ((iv & 0xff00) == 0xff00) {
                u8 B = (iv >> 16) & 0xff;
                if (B >= 3 && B < 3 + keylen)
                        return true;
                }
```

[10]El extracto de código pertenece al fichero *linux/net/mac80211/wep.c* del kernel de Linux en su versión 3.16.

```
                return false;
    }
```

Dicho filtrado previene de forma efectiva el ataque FMS tal y como es implementado en [129] y [67].

La claves débiles determinadas por el patrón $(p, 255, X)$ son, sin embargo, un subconjunto del total de claves débiles determinadas por un estado S_{p-1} resuelto, determinado por cumplir las condiciones $S_{p-1}[1] < p$ y $(S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]) = p$. El patrón de IV tiene la propiedad de que es capaz de detectar claves débiles para un byte de clave determinado de una forma independiente al resto de bytes anteriores de la misma. Sin embargo, de esta forma el total de claves débiles será de un máximo de 256 para cada byte de la clave, existiendo muchas otras claves débiles dependientes de bytes previos de la clave, que sin cumplir el patrón de IV, verifican las condiciones mencionadas anteriormente. En efecto, la detección del total de claves débiles para el byte número b de la clave, implica la ejecución de b rondas de KSA, siendo necesario conocer el valor de los bytes anteriores de la clave para ejecutar el proceso.

A pesar de que este hecho se desprendiera del artículo original [62] que describe el ataque FMS, las implementaciones [129] y [67] que le sucedieron, así como las medidas preventivas como aquella utilizada en el kernel de Linux, utilizan el patrón de IV descrito en [62], limitándose se esta manera a un subconjunto de claves débiles respecto del total de claves vulnerables al ataque FMS.

Este hecho, fue denotado por [75] quien también propuso la extensión del ataque al segundo byte del *keystream*. Al igual que el primero, el segundo byte de *keystream* puede ser determinado realizando un XOR del segundo byte del paquete cifrado con la constante 0xAA perteneciente a la cabecera SNAP.

Un segundo byte de *keystream* es generado por el algoritmo PRGA de RC4 como $S_{p-1}[(S_{p-1}[2] + S_{p-1}[j_2])]$. Dado que $j_2 = S_{p-1}[1] + S_{p-1}[2]$, el segundo byte de *keystream* será calculado como $S_{p-1}[(S_{p-1}[2] + S_{p-1}[S_{p-1}[1] + S_{p-1}[2]])]$ y dependerá de las siguientes 4 posiciones de S_{p-1}: $S_{p-1}[2]$, $S_{p-1}[1]$, $S_{p-1}[S_{p-1}[1] +$

$S_{p-1}[2]]$, y $S_{p-1}[S_{p-1}[2] + S_{p-1}[S_{p-1}[1] + S_{p-1}[2]]]$.

Para que las 4 posiciones de S_{p-1} involucradas en el cálculo del segundo byte del *keystream* tengan una probabilidad del $e^{-4} \simeq 2\,\%$ permanecer con los mismos valores tras todas las rondas del PRGA, las siguientes condiciones han de ser satisfechas: $S_{p-1}[1] \neq 2$, $S_{p-1}[2] \neq 0$, $S_{p-1}[2] + S_{p-1}[1] < p$ y $S_{p-1}[2] + S_{p-1}[S_{p-1}[1] + S_{p-1}[2] = p$.

En los casos descritos anteriormente, se puede derivar la siguiente ecuación lineal sobre el byte p de la clave, donde $S^{\star}_{p-1}[X]$ denota la posición que ocupa el valor X en el vector S_{p-1}:

$$K[p] \quad = \quad (S^{\star}_{p-1}[o_1] - j_{p-1} - S_{p-1}[p])$$

Otra mejora sobre el ataque FMS original al primer byte de *keystream*, fue ya apuntada por [75] y [150], donde se sugería la existencia de un posible caso especial del ataque FMS, donde dos de las tres posiciones de S que determinaban los valores en base a los cuales se calcula o_1, fueran idénticas. Posteriormente, en [17], se analizaban las condiciones bajo las cuales dicho caso sería posible.

En el ataque FMS, se utilizan las posiciones 1, $S_{p-1}[1]$ y $S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]$. La probabilidad de éxito del ataque es calculada como $e^{-3} \simeq 5\,\%$, dado que depende de los valores contenidos en las 3 posiciones. Por otro lado, si dos de las tres posiciones fueran iguales, la probabilidad de éxito del ataque sería de $e^{-2} \simeq 13\,\%$, al depender únicamente de dos elementos de S.

De las condiciones del ataque FMS se desprende que $S_{p-1}[1] + S_{p-1}[S_{p-1}[1]] = p$, por lo que para que dos posiciones de las tres sean iguales, las únicas posibilidades serían [11] $S_{p-1}[S_{p-1}[1]] = p$ y $S_{p-1}[S_{p-1}[1]] = 1$.

$S_{p-1}[1] = 1$ resulta en una imposibilidad, ya que $S_{p-1}[1] + S_{p-1}[S_{p-1}[1]] = 1 + S_{p-1}[1] = 2$, en lugar de p.

Por otro lado $S_{p-1}[S_{p-1}[1]] = p$ sería ciertamente posible en el caso en el que $S_{p-1}[1] = p$. Resolviendo el sistema de ecuaciones se obtiene que $o_1 = S_{p-1}[S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]] = S_{p-1}[1] = p$. Por lo tanto las condiciones para el caso especial del ataque FMS al primer byte del *keystream* serían $S_{p-1}[1] = p$ y $o_1 = p$.

[11] $S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]$ es la posición p por lo que obviamente no puede ser la posición 1

4.2.4. *Weplab*, implementando un ataque FMS mejorado

Con el objetivo de validar la hipótesis de que el protocolo WEP no ofrece suficientes garantías para la protección de la seguridad y privacidad de las comunicaciones en situaciones reales de utilización del protocolo, se desarrolla una herramienta capaz de capturar tráfico WiFi cifrado con WEP y experimentar diferentes tipos de ataques dirigidos a la obtención de la clave de cifrado.

La implementación, denominada como *Weplab*, persigue los siguientes objetivos.

1. Determinar la viabilidad de criptoanalizar claves de cifrado WEP de 40 y 104 bits, utilizando recursos estándar, mediante el análisis de tráfico cifrado en escenarios reales de utilización.

2. Servir como plataforma para la investigación experimental de técnicas de criptoanálisis del cifrado RC4 en su utilización en el protocolo WEP.

A pesar de que el núcleo de la implementación este compuesto por el ataque de criptoanálisis, se implementan también los ataques diccionario y de búsqueda exhaustiva. Dado que el presente capítulo se refiere al criptoanálisis de WEP, no se elaborarán los detalles referentes el resto de ataques.

La primera implementación de criptoanálisis realizada, se centra en la aplicación del ataque FMS en su completa extensión, tal y como se describe en la secciones 4.2.2 y 4.2.3 del capítulo actual.

El método de ataque desarrollado es modular, permitiendo la acomodación de nuevas relaciones lineales entre bytes del *keystream* y bytes de la clave. Por cada paquete cifrado capturado, se derivan los dos primeros bytes de *keystream* mediante una operación lógica XOR entre los dos primeros bytes del paquete cifrado y el valor conocido de la cabecera SNAP. A continuación, se recorren iterativamente los diferentes números de byte de la clave y se determinan la cantidad de relaciones lineales, o ataques, aplicables en base a sus condiciones. Por cada ataque válido, se determina el valor del byte probable y se le vota con un peso proporcional a la probabilidad de éxito estimada para el ataque en cuestión.

Una vez finalizado el proceso para todos los bytes de la clave, se procede a ordenar la lista de valores por cada byte de la clave, en función a la cantidad de votos recibidos. Con el objetivo de combinar el proceso de criptoanálisis con una búsqueda exhaustiva, a fin de maximizar la probabilidad de éxito, se exploran los n primeros candidatos por cada byte de la clave, siendo n la profundidad seleccionada.

Dado que el proceso de criptoanálisis del byte número b de la clave, depende del valor de todos los anteriores, tal y como se describe en la sección 4.2.2, la búsqueda de la clave correcta dentro de los candidatos es un proceso recursivo, donde se prueban secuencialmente diferentes claves candidatas. Por cada posible valor de clave, bien sea para 40 o 104 bits, se prueba el descifrado del paquete y se utiliza el CRC a modo de validación del proceso. En caso de que el CRC sea positivo, se utilizan otros paquetes adicionales para descartar falsos positivos. El proceso de búsqueda puede optimizarse de forma opcional, realizando una búsqueda exhaustiva de los últimos n bytes de la clave, en lugar de criptoanalizarlos.

Para más información sobre los diferentes ataques soportados, su implementación específica y la gama de parámetros configurables que guían el ataque, se puede consultar el código fuente de *Weplab* en [136].

La figura 4.3 muestra la efectividad obtenida en las pruebas experimentales para el criptoanálisis de claves de 104-bits utilizando el ataque FMS implementado, en base al número de paquetes cifrados analizados. Para el experimento se ha utilizado una generación aleatoria de vectores de inicialización para un un total de 2000 claves aleatorias. Las diferentes líneas muestran la evolución del porcentaje de efectividad en función de la cantidad de paquetes cifrados analizados, para los diferentes valores estáticos de profundidad.

La figura 4.4 muestra los resultados para un experimento similar donde la generación de los vectores de inicialización de los paquetes es realizada de forma secuencial, partiendo de un primer IV aleatorio.

Con el objetivo de mejorar el rendimiento de la implementación realizada, se implementa el ataque al segundo byte de *keystream*, tal y como se describe

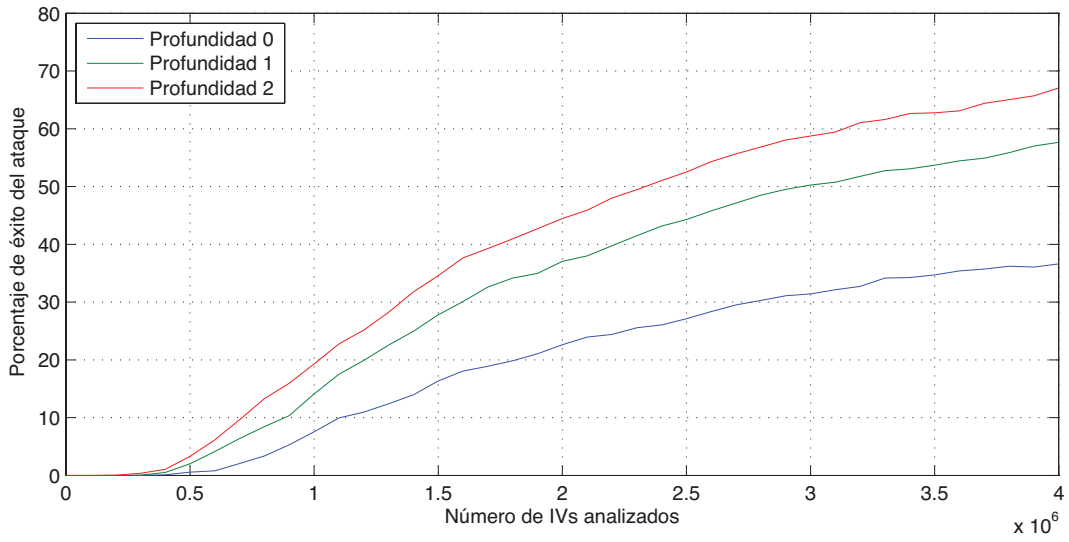


Figura 4.3: Rendimiento del ataque FMS implementado en *Weplab* para IVs generados aleatoriamente y un total de 2000 claves aleatorias

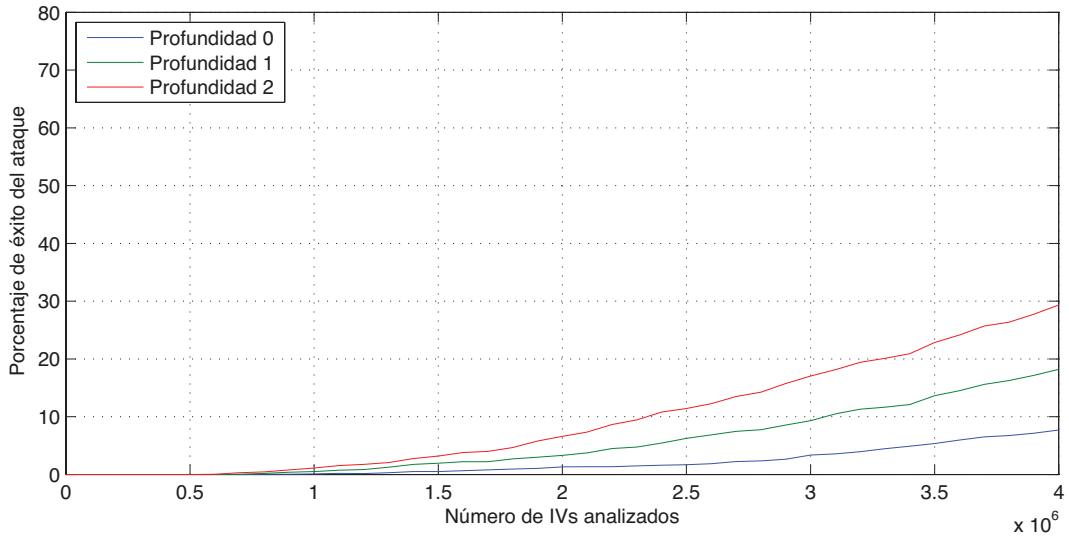


Figura 4.4: Rendimiento del ataque FMS implementado en *Weplab* para IVs generados secuencialmente (LE) y un total de 2000 claves aleatorias

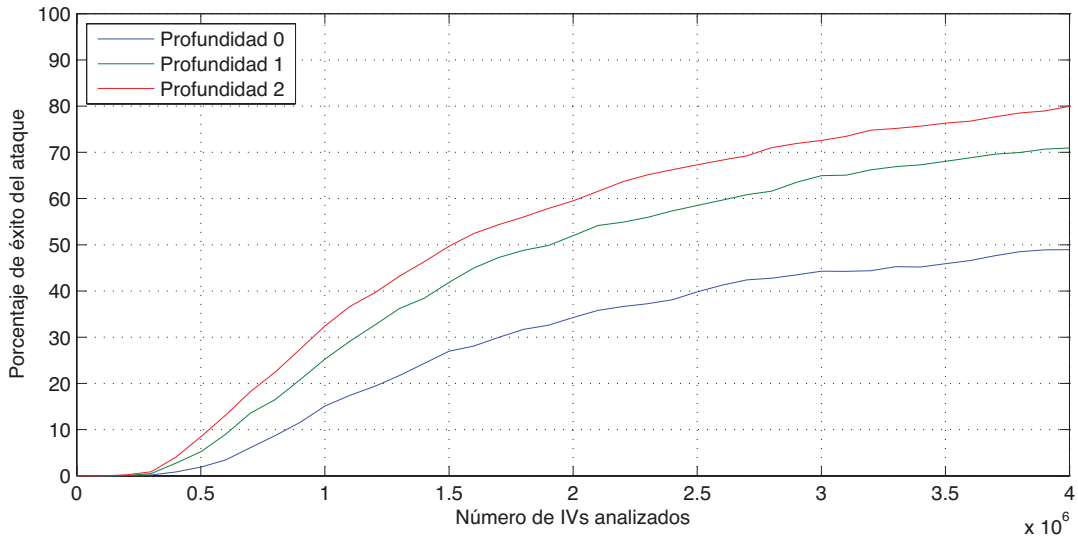


Figura 4.5: Rendimiento de los ataques combinados al primer y segundo byte del *keystream* implementados en *Weplab* para IVs generados aleatoriamente y un total de 2000 claves aleatorias

en la sección 4.2.3. La arquitectura modular de la implementación permite su combinación con el ataque FMS, de tal manera que los votos emitidos por ambos ataques, con sus respectivos pesos, son utilizados para la generación de la lista de valores candidatos para cada número de byte de la clave.

La figura 4.5 muestra el rendimiento de la nueva implementación, para un total de 2000 claves aleatorias y una generación también aleatoria de los vectores de inicialización.

En la figura 4.6 se muestran los resultados para el mismo experimento, generando es esta ocasión los vectores de inicialización de modo secuencial.

4.2.5. Mejorando *Weplab* con los ataques KoreK

Las primeras versiones de *Weplab* implementan el ataque completo FMS al primer byte y el equivalente al segundo byte del *keystream*, ofreciendo resultados positivos en el criptoanálisis práctico de claves de 40 y 104-bits, a partir de 500.000 IV capturados.

La herramienta *Aircrack* surge posteriormente a *Weplab* implementando de forma similar los ataques criptográficos pasivos contra WEP, junto con un ata-

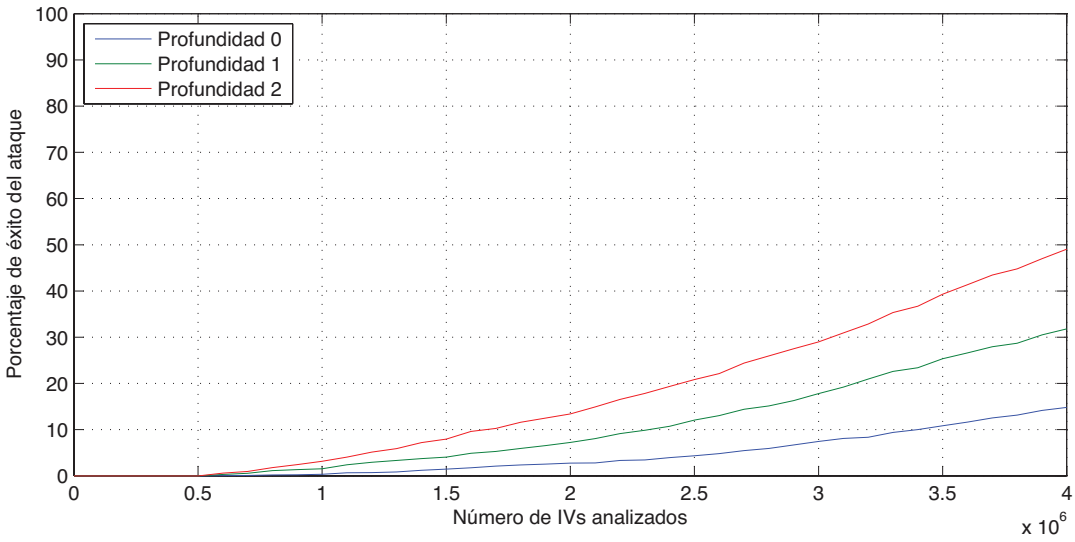


Figura 4.6: Rendimiento de los ataques combinados al primer y segundo byte del *keystream* implementados en *Weplab* para IVs generados secuencialmente (LE) y un total de 2000 claves aleatorias

que activo de repetición que permite acelerar la recuperación de IV de una red WEP objetivo.

Tras publicar el código fuente de varias versiones iniciales de *Weplab* y recibir evaluaciones independientes de sus resultados por parte de otros investigadores de seguridad, un investigador bajo el pseudónimo de KoreK contacta por email al autor de esta tesis para sugerir mejoras en los ataques implementados. Korek consolida los ataques existentes y los integra con nuevas clases de ataques, tanto para el primero como el segundo byte de *keystream*. Los ataques son implementados en *Weplab* y *Aircrack*, con ligeras diferencias, resultando en una mejora drástica en la efectividad de las implementaciones en la ruptura de claves WEP.

Los ataques KoreK no fueron publicados de forma formal, por lo que su implementación en *Weplab* y *Aicrack* se convertiría durante varios años en su referencia documental. Posteriormente, en [28], se describe en detalle el funcionamiento de los 17 ataques KoreK y se analiza su implementación y resultados en *Weplab* y *Aircrack*.

Dado que posteriormente, en [28], los autores analizarían en detalle los ata-

ques KoreK junto con su implementación en *Weplab*, la presente sección realiza una descripción comparativa de los mismos, para centrarse en la implementación experimental integrada en *Weplab*, detallando los resultados obtenidos. Para más detalles sobre el conjunto de ataques denominado como KoreK se puede consultar el código fuente de *Weplab*, publicado en [136], y el análisis posterior en [28].

El conjunto de ataques KoreK se dividen en tres clases, los que atacan al primer byte de *keystream*, denominado como o_1, los que atacan al segundo byte, denominado como o_2, y los negados que son aquellos que, utilizando tanto o_1 como o_2, son capaces de determinar ciertos valores de la clave con baja probabilidad de ser los correctos.

La tabla 4.1 muestra el primer grupo de ataques KoreK, compuesto por 8 ataques todos los cuales utilizan el valor del primer byte de *keystream*, denominado como o_1. En la tabla 4.1 se muestra la denominación de cada ataque, el número de byte de *keystream* que utiliza y la probabilidad de éxito aproximada.

Por cada ataque de la tabla, se muestra en la última columna su expresión matemática para las condiciones de determinación de claves débiles y el cálculo del valor probable para el byte de la clave atacado. j_p denota el valor del puntero j en la ronda p del KSA. $S_{p-1}[x]$ denota el valor de la posición x del vector S en su ronda $p-1$ del KSA, siendo p el valor de la clave siendo atacado. De la misma manera, $S^{\star}_{p-1}[x]$ denota la posición que ocupa el valor x en el vector S_{p-1}.

El primer ataque, denominado como *A_s5_1*, es básicamente el ataque FMS completo, tal y como es implementado en *Weplab*, según es descrito en la sección 4.2.4, junto con una pequeña optimización que redunda en una ligera mejora en su probabilidad de éxito en la predicción del byte de la clave. En efecto, la inclusión de las condiciones $S^{\star}_{p-1}[o1] \neq 1$ y $S^{\star}_{p-1}[o1] \neq S_{p-1}[S_{p-1}[1]]$ para la detección de clave débiles, es capaz de detectar y descartar aquellos casos en los que los valores $S_{p-1}[1]$ y $S_{p-1}[S_{p-1}[1]]$ vayan a ser sobrescritos. En las pruebas experimentales realizadas con ambas implementaciones en *Weplab*, la mejora de la probabilidad llevada a cabo por dichas optimizaciones se sitúa en

Denom.	Estabilidad	o_x	Prob.	Expresión matemática
A_s5_1	Estable	1º	5 %	$S_{p-1}[1] < p$ $S_{p-1}[1] + S_{p-1}[S_{p-1}[1]] = p$ $S^{\star}_{p-1}[o1] \neq 1$ $S^{\star}_{p-1}[o1] \neq S_{p-1}[S_{p-1}[1]]$ $K_p = S^{\star}_{p-1}[o_1] - S_{p-1}[p] - j_{p-1}$
A_s13	Estable	1º	13.75 %	$S_{p-1}[1] = p$ $o_1 = p$ $K_p = S^{\star}_{p-1}[0] - S_{p-1}[p] - j_{p-1}$
A_u13_1	Inestable	1º	13.75 %	$S_{p-1}[1] = p$ $o_1 = 1 - p$ $K_p = S^{\star}_{p-1}[o_1] - S_{p-1}[p] - j_{p-1}$
A_u5_1	Inestable	1º	5 %	$S_{p-1}[1] = p$ $o_1 \neq 1 - p$ $o_1 \neq p$ $S^{\star}_{p-1}[o_1] < p$ $S^{\star}_{p-1}[(S^{\star}_{p-1}[o_1] - p)] \neq 1$ $K_p = S^{\star}_{p-1}[(S^{\star}_{p-1}[o_1] - p)] - S_{p-1}[p] - j_{p-1}$
A_u5_2	Inestable	1º	5 %	$S^{\star}_{p-1}[o_1] = 2$ $S_{p-1}[p] = 1$ $K_p = 1 - S_{p-1}[p] - j_{p-1}$
A_u13_2	Inestable	1º	13.75 %	$S^{\star}_{p-1}[p] = p$ $S_{p-1}[1] = 0$ $o_1 = p$ $K_p = 1 - S_{p-1}[p] - j_{p-1}$
A_u13_3	Inestable	1º	13.75 %	$S_{p-1}[p] = p$ $o_1 = S_{p-1}[1]$ $S_{p-1}[1] = 1 - p$ $K_p = 1 - S_{p-1}[p] - j_{p-1}$
A_u5_3	Inestable	1º	5.07 %	$S_{p-1}[p] = p$ $S_{p-1}[1] \geq -p$ $S_{p-1}[1] = S^{\star}_{p-1}[o_1] - p$ $S_{p-1}[1] \neq 1$ $K_p = 1 - S_{p-1}[p] - j_{p-1}$

Tabla 4.1: Ataques Korek al primer byte de *keystream*, o_1

el rango del 0.1 %-0.2 %.

El ataque denominado como *A_s5_1* es la implementación del caso especial del ataque FMS descrito en la sección 4.2.4, donde el cálculo de o_1 depende únicamente de los valores de 2 posiciones de S, por lo que la probabilidad de éxito mejora del 5 % al 13 %.

A_u13_1 se basa en los principios del ataque anterior, realizando un intento de predecir el valor de j_p utilizando la condición $S_{p-1}[1] = p$ y $o_1 = 1 - p$. En este caso, en la primera ronda de PRGA, la posición 1 de S será intercambiada con la posición p, ya que $S_p[1] = p$. Por lo tanto, debe cumplirse que $S_p[p] = 1 - p$ para que el valor o_1 sea correcto.

Dado el intercambio entre las posiciones p y 1, tras el primer intercambio en PRGA previo al cálculo de o_1, se cumplirá que $S[1] = 1 - p$ y $S[p] = p$. o_1 será calculado como $o_1 = S[S[1] + S[S_{p-1}[1]]] = S[1 - p + S[p]] = S[1 - p + p] = S[1] = 1 - p$. La probabilidad de éxito del ataque es del 13.75 %.

El ataque *A_u5_1* utiliza un principio similar con la condición $Sp - 1[1] = p$. Las condiciones $o_1 \neq 1 - p$ y $o_1 \neq p$ son utilizadas para evitar solapamiento con los ataques *A_u13_1* y *A_s13* respectivamente. Las condiciones adicionales tiene como objetivo optimizar la probabilidad de éxito detectando compromisos potenciales de las posiciones de S utilizadas en futuras rondas del KSA.

Los ataques *A_u5_2*, *A_u13_3*, *A_u13_4* y *A_u5_3* asumen $j_p = 1$ y utilizan las condiciones descritas en la tabla 4.1 para determinar bytes de la clave con las probabilidades 5.07 %, 13.75 %, 13.75 % y 5.07 % respectivamente.

Las tablas 4.2 y 4.3 muestran el siguiente grupo de ataques KoreK, los cuales utilizan el valor del segundo byte de *keystream*, denominado como o_2. El primero de ellos, denominado *A_s*3 es básicamente el ataque FMS extendido a o_2, tal y como ha sido descrito en la sección 4.2.3, añadiendo una pequeña optimización en su rendimiento al verificar con las condiciones, $S^{\star}_{p-1}[o_2] \neq 1$, $S^{\star}_{p-1}[o_2] \neq 2$ y $S^{\star}_{p-1}[o_2] \neq S_{p-1}[2] + S_{p-1}[1]$,que las posiciones de S utilizadas no hayan sido sobrescritas en rondas posteriores.

El ataque *A_u15* supone que $j_p = 2$ y utiliza la condición $o_2 = 0$ y $S_{p-1}[p] = 0$. Para generar o_2, el algoritmo PRGA realiza 2 rondas en las que j toma los

Denom.	Estabilidad	o_x	Prob.	Expresión matemática
A_s3	Estable	2º	5.13 %	$S_{p-1}[1] \neq 2$ $S_{p-1}[2] \neq 0$ $S_{p-1}[2] + S_{p-1}[1] < p$ $S_{p-1}[2] + S_{p-1}[S_{p-1}[2] + S_{p-1}[1]] = p$ $S^{\star}_{p-1}[o_2] \neq 1$ $S^{\star}_{p-1}[o_2] \neq 2$ $S^{\star}_{p-1}[o_2] \neq S_{p-1}[2] + S_{p-1}[1]$ $K_p = S^{\star}_{p-1}[o_2] - S_{p-1}[p] - j_{p-1}$
A_u15	Inestable	2º	13.75 %	$o_2 = 0$ $S_{p-1}[p] = 0$ $S_{p-1}[2] \neq 0$ $K_p = 2 - S_{p-1}[p] - j_{p-1}$
A_s5_2	Estable	2º	5.07 %	$S_{p-1}[1] > p$ $S_{p-1}[2] + S_{p-1}[1] > p$ $o_2 = S_{p-1}[1]$ $S_{p-1}[S_{p-1}[1] - S_{p-1}[2]] \neq 1$ $S_{p-1}[S_{p-1}[1] - S_{p-1}[2]] \neq 2$ $K_p = S^{\star}_{p-1}[(S_{p-1}[1] - S_{p-1}[2])] - S_{p-1}[p] - j_{p-1}$
A_s5_3	Estable	2º	5.07 %	$S_{p-1}[1] > p$ $S_{p-1}[2] + S_{p-1}[1] = p$ $o_2 = 2 - S_{p-1}[2]$ $S^{\star}_{p-1}[o_2] \neq 1$ $S^{\star}_{p-1}[o_2] \neq 2$ $K_p = S^{\star}_{p-1}[(2 - S_{p-1}[2])] - S_{p-1}[p] - j_{p-1}$

Tabla 4.2: Ataques Korek al segundo byte de *keystream* o_2 (1/2)

Denom.	Estabilidad	o_x	Prob.	Expresión matemática
A_s4_13	Estable	2º	13.85 %	$p = 4$ $S_{p-1}[1] = 2$ $o_2 = 0$ $K_p = S^{\star}_{p-1}[0] - S_{p-1}[p] - j_{p-1}$
A_s4_13	Inestable	2º	5.13 %	$p = 4$ $S_{p-1}[1] = 2$ $o_2 \neq 0$ $S^{\star}_{p-1}[o_2] = 0$ $j_1 = 2$ $K_p = S^{\star}_{p-1}[254] - S_{p-1}[p] - j_{p-1}$
A_s4_13	Inestable	2º	5.13 %	$p = 4$ $S_{p-1}[1] = 2$ $o_2 \neq 0$ $S^{\star}_{p-1}[o_2] = 2$ $j_1 = 2$ $K_p = S^{\star}_{p-1}[255] - S_{p-1}[p] - j_{p-1}$
A_s4_13	Inestable	2º	5.25 %	$p > 4$ $S_{p-1}[1] = 2$ $S_{p-1}[4] + 2 = p$ $S^{\star}_{p-1}[o2] \neq 1$ $S^{\star}_{p-1}[o2] \neq 4$ $K_p = S^{\star}_{p-1}[o_2] - S_{p-1}[p] - j_{p-1}$

Tabla 4.3: Ataques Korek al segundo byte de *keystream* o_2 (2/2)

valores $S[1]$ y $S[2]$ respectivamente, ya que en cada ronda j se actualiza como $j = j + S[i]$. Dada la condiciones del ataque $S_{p-1}[p] = 0$ y la suposición de $j_p = 2$, en la segunda ronda de PRGA $S[2] = 0$ por lo que j mantendrá su valor. El ataque depende de 2 valores y su probabilidad estimada es de 13.75 %.

El siguiente ataque, denominado *A_s5_2* define una serie de condiciones para determinar el valor de j en la segunda ronda del PRGA, que será calculado como $j = S[1] + S[2]$, en caso que $S[1]$ no haya sido intercambiado en la primera ronda. Para asegurar esto último, se utiliza la condición $S_{p-1}[1] > p$. La segunda condición, $S_{p-1}[2] + S_{p-1}[1] = p$ es utilizada para inducir el intercambio de $S[2]$ por $S[p]$ en la segunda ronda del PRGA. Finalmente, la condición $o_2 = S_{p-1}[1]$, junto con las anteriores, permite determinar j_p como $S^{\star}_{p-1}[(S_{p-1}[1] - S_{p-1}[2])]$. El resto de condiciones son optimizaciones orientadas a la detección del compromiso en las posiciones $S_{p-1}[1]$, $S_{p-1}[2]$ y $S_{p-1}[p]$, de las cuales depende el ataque. Al depender de 3 valores, la probabilidad de éxito es del 5 %.

El ataque *A_s5_3* detecta cuando $o_2 = S_{p-1}[p]$, forzando $j = p$ en la segunda ronda del PRGA, con las primeras 2 condiciones del ataque de forma similar al modo empleado por el ataque *A_s5_2*. Si $j_p = 2 - S_{p-1}[2]$ entonces $S_p[p] = S_{p-1}[2 - S_{p-1}[2]]$ y en la segunda ronda, donde $j = p$, o_2 será calculado como $o_2 = S[S[2] + S[p]] = S_{p-1}[S_{p-1}[2] + 2 - S_{p-1}[2]] = S_{p-1}[2]$, asumiendo que $S_{p-1}[1]$, $S_{p-1}[2]$ y $S_{p-1}[p]$ no hayan cambiados en rondas posteriores del KSA. Para que esto ocurra $S^{\star}_{p-1}[o_2] \neq 1$ y $S^{\star}_{p-1}[o_2] \neq 2$. El ataque por tanto tiene una probabilidad estimada del 5 %.

Los ataques *A_4_s13*, *A_4_u5_1* y *A_4_u5_2* son aplicables cuando $p = 4$ y $S_{p-1}[1] = 2$, con el objetivo que en la segunda ronda del PRGA se cumpla $S[4] = 2$ ya que en la ronda previa $S[2] = 2$ al ser intercambiada con $S_{p-1}[1]$. En *A_4_s13* se asume que $j_4 = S^{\star}_{p-1}[0]$, de tal manera que $S_p[4] = 0$ y estableciendo la condición $o_2 = 0$, tras la segunda ronda del PRGA, $o_2 = S[S[2] + S[4]] = S[0 + 2] = S[2] = 0$. El ataque depende de la inmutabilidad de 2 elementos de S_{p-1} y tiene por tanto una probabilidad estimada del 13.75 %.

El ataque *A_4_u5_1* utiliza un procedimiento similar asumiendo $j_4 = S^{\star}_{p-1}[254]$,

Denom.	Estabilidad	Prob.	Expresión matemática
A_neg_1	Estable	99 %	$S_{p-1}[2] = 0$ $S_{p-1}[1] = 2$ $o_1 = 2$ $K_p \neq 1 - S_{p-1}[p] - j_{p-1}$ $K_p \neq 2 - S_{p-1}[p] - j_{p-1}$
A_neg_2	Estable	99 %	$S_{p-1}[2] = 0$ $o_2 = 0$ $S_{p-1}[1] \neq 2$ $K_p \neq 2 - S_{p-1}[p] - j_{p-1}$
A_neg_3	Estable	99 %	$S_{p-1}[1] = 1$ $o_1 = S_{p-1}[2]$ $K_p \neq 1 - S_{p-1}[p] - j_{p-1}$ $K_p \neq 2 - S_{p-1}[p] - j_{p-1}$
A_neg_4	Estable	99 %	$S_{p-1}[1] = 0$ $S_{p-1}[0] = 1$ $o_1 = 1$ $K_p \neq 1 - S_{p-1}[p] - j_{p-1}$

Tabla 4.4: Ataques Korek invertidos

en el caso que $o_2 \neq 0$, con el objetivo de evitar solapamiento con el ataque anterior. En caso que $S^{\star}_{p-1}[o_2] = 0$, tras el PRGA $o_2 = S[S[2] + S[4] = S[2 + 254] = S[0]$. La condición adicional, $j_1 = 2$, se utiliza a modo de optimización para el descarte de falsos positivos. El ataque *A_4_u5_2* utiliza exactamente el mismo mecanismo pero con $S^{\star}_{p-1}[o_2] = 2$, de tal forma que, tras el PRGA, $o_2 = S[S[2] + S[4] = S[2 + 255] = S[1]$. En ambos casos, al depender del valor de 3 posiciones de S_{p-1}, la probabilidad estimada de los ataques es de 5 %.

Por su parte, el ataque *A_u5_4* es aplicable cuando $p > 4$ y similarmente al grupo de ataques anteriores, utiliza la condición $S_{p-1}[1] = 2$ para obtener $S[4] = 2$ tras la segunda ronda del PRGA. Mediante la condición $S_{p-1}[4] = p - 2$, tras la segunda ronda de PRGA, $o_2 = S[S[2] + S[4] = S[2 + p - 2] = S[p]$. Las dos condiciones adicionales optimizan la probabilidad del ataque detectando sobrescrituras en las posiciones de S utilizadas. El ataque tiene una probabilidad estimada del 5 %.

El ultimo grupo de ataques, denominados ataques invertidos, se muestran

en la tabla 4.4. El objetivo de estos ataques no es detectar los valores más probables para los bytes de la clave, sino aquellos más improbables. Para ello, los ataques buscan condiciones bajo las cuales ciertos valores para los diferentes bytes de la clave sean imposibles.

A pesar de que en *Weplab* todos los ataques invertidos sean implementados bajo diferentes ramas de ejecución condicionales pertenecientes al mismo ataque, con el objetivo de facilitar su descripción, en la presente sección se describe cada conjunto de condiciones como un ataque invertido independiente.

El primer ataque, denominado *A_neg_1*, si $S_{p-1}[2] = 0$ y $S_{p-1}[1] = 2$ entonces, si $o_1 = 2$, necesariamente el valor de j_p es tal que no ha alterado el valor de las posiciones $S_{p-1}[1]$ y $S_{p-1}[2]$, ya que $o_1 = S[S[1] + S[S[1]] = S[2 + S[2]] = S[2 + 0] = 2$ [12]. De esta manera es posible determinar 2 valores de K_p que darían lugar a tales modificaciones.

El ataque *A_neg_2* utiliza las condiciones $S_{p-1}[2] = 0$ y $o_2 = 0$. Si $S_{p-1}[1] \neq 2$, en la segunda ronda del PRGA, $o_2 = S[0 + S_{p-1}[1]] = S[S_{p-1}[1]] = 0$, dado que en la primera ronda del PRGA $S[S[1]] = S[1]$. Por lo tanto, cualquier valor de K_p que de lugar a la modificación posterior de $S_{p-1}[2]$, es inválido.

A_neg_3 utiliza los casos donde $S_{p-1}[1] = 1$ y $o_1 = S_{p-1}[2]$, de tal manera que si $S_{p-1}[1]$ ha sido alterado, no podría darse que $o_1 = S_{p-1}[S_{p-1}[1] + S_{p-1}[S_{p-1}[1]]] = S_{p-1}[1 + S_{p-1}[1]] = S_{p-1}[2]$. Mediante este ataque es posible descartar aquellos valores de K_p que alteren durante KSA el valor de $S_{p-1}[1]$ o $S_{p-1}[2]$.

Finalmente, en el ataque *A_neg_4* se utilizan las condiciones $S_{p-1}[1] = 0$ y $S_{p-1}[0] = 1$, bajo las cuales tras el intercambio en la primera ronda de PRGA, $o_1 = S[S[1] + S[0]] = 0$, siendo, por tanto, $o_1 = 0$ la última condición del ataque. El ataque permite descartar los valores de K_p que alteren $S_{p-1}[1]$, ya que $o_1 = 1$ es posible aunque el valor $S_{p-1}[1]$ haya sido alterado.

La figura 4.7 muestra el porcentaje de claves rotas para un total de 2000 claves generadas aleatoriamente con una generación también aleatoria de los

[12] El valor $S_{p-1}[1] = 2$ pasa a $S[2]$ en el intercambio realizado en la primera ronda del PRGA, ya que $j = S_{p-1}[1] = 2$ y $i = 1$

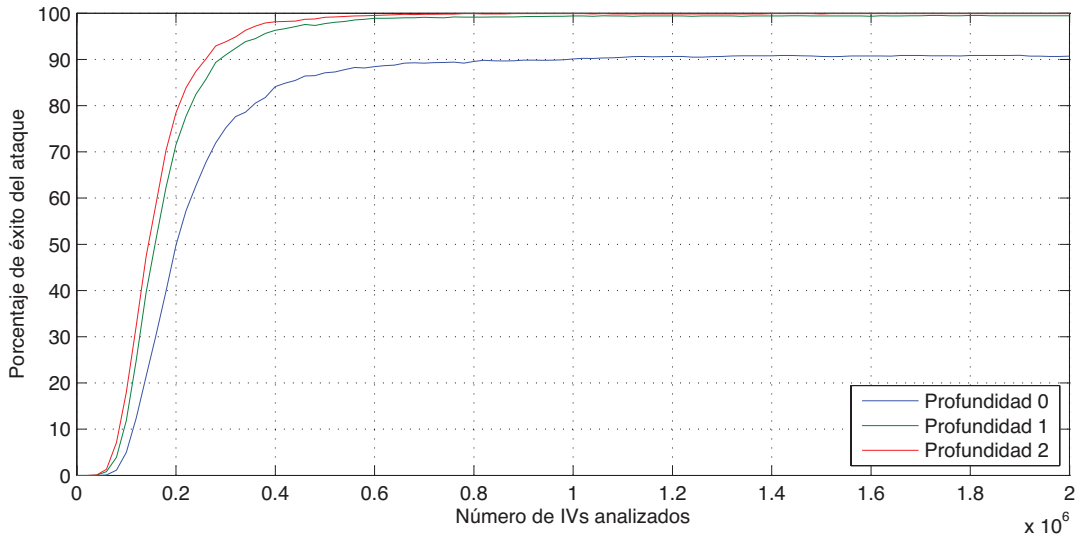


Figura 4.7: Rendimiento de los ataques KoreK para IVs generados aleatoriamente y un total de 2000 claves aleatorias

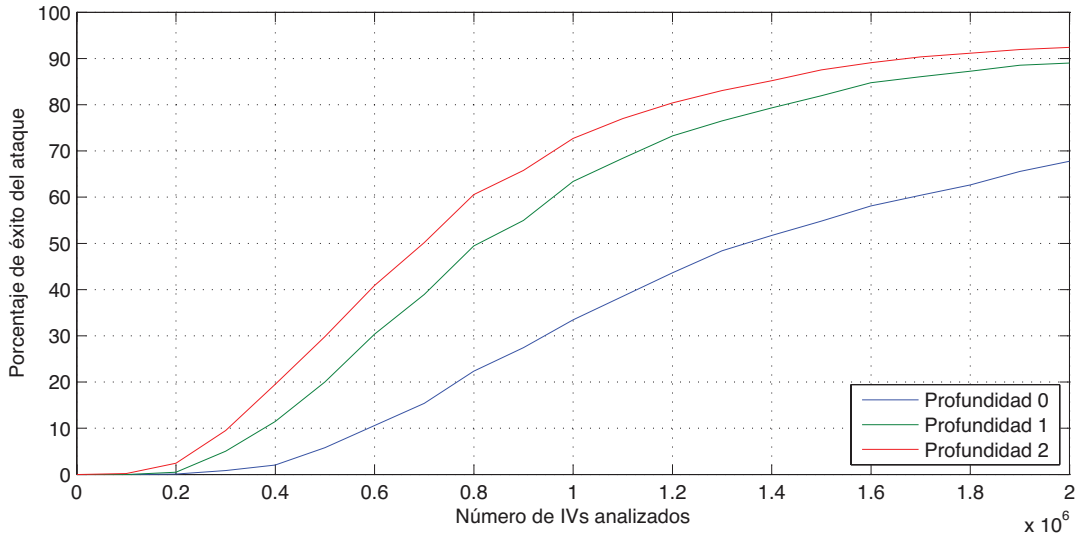


Figura 4.8: Rendimiento de los ataques KoreK para IVs generados aleatoriamente y un total de 2000 claves aleatorias

IVs. La figura 4.8 muestra la evolución de la precisión del ataque para 2000 claves generadas aleatoriamente con generación secuencial de los vectores de inicialización.

En ambas pruebas experimentales realizadas se aplica un factor de profundidad estático y no se utiliza búsqueda exhaustiva de los últimos bytes de la clave. Los resultados pueden ser mejorados utilizando un factor de profundidad dinámico, en base al porcentaje de votos obtenidos por los candidatos más probables.

En [153] se muestran el rendimiento de la implementación del ataque FMS y los ataques KoreK realizada en *Aircrack*. Las diferencias en su implementación específica, junto con una aproximación diferente en el marco del ataque general, derivan en diferencias de rendimiento con respecto al obtenido con la implementación de *Weplab*.

4.3. Fugas de información en redes 802.11 cifradas mediante ataques de canal lateral

La tecnología WiFi 802.11 se ha convertido en el protocolo de comunicación por excelencia en el entorno de la casa inteligente. Tras más de 10 años desde su desarrollo, el estándar 802.11 ha evolucionado a redes de alta velocidad, con el último estándar 802.11ac alcanzando el gigabit por segundo [33], y se ha posicionado como el líder del mercado en lo referente las comunicaciones IP en entornos domésticos.

Actualmente, con la excepción de la telefonía inalámbrica en el hogar que se encuentra liderada por el protocolo DECT, objeto de estudio del capítulo 2 de la presente tesis, el protocolo WiFi es utilizado como medio de intercomunicación entre los diferentes dispositivos inteligentes del hogar, ya sean bien ordenadores y tablets, o dispositivos más específicos como televisiones inteligentes o aparatos de cocina.

A lo largo de los varios años de vida del estándar 802.11, su capacidad para la protección de la seguridad y privacidad de las comunicaciones ha ido

evolucionando. En el análisis del estado del arte, realizado en la sección 4.1, se detallan los ataques existentes contra las diferentes variantes de cifrado y protocolos asociados, y en la sección 4.2 se describe el trabajo de investigación realizado en el criptoanálisis del protocolo de cifrado WEP.

En contraste con el escrutinio al que los protocolos de seguridad WiFi se han visto sometidos por parte de la comunidad académica, la aplicación práctica de ataques de canal lateral para la extracción de información privada en la observación pasiva de comunicaciones WiFi cifradas, no ha sido explorada tan en profundidad.

En la sección 4.1, se analizaba la literatura existente al respecto, donde diferentes autores describen el problema de forma general en comunicaciones cifradas y presentan vectores de ataques capaces de extraer información privada del usuario mediante la observación de comunicaciones cifradas con otros protocolos. El trabajo más cercano al presentado en esta sección de la tesis, es [31], donde los autores demuestran la aplicación práctica de dicha familia de ataques sobre comunicaciones WiFi cifradas para la detección de las acciones de un usuario en una determinada aplicación web mediante el análisis de tráfico cifrado.

En el contexto actual, donde todos los dispositivos de la casa inteligente se comunican entre ellos y con Internet utilizando la red inalámbrica WiFi, dada la cada vez mayor expansión del concepto de casa inteligente y su integración en nuestra vida cotidiana, la protección de la información privada que circula por vías inalámbricas adquiere una importancia fundamental.

En esta sección se exploran las limitaciones inherentes a la tecnología WiFi para la protección de información privada en el contexto emergente de la casa inteligente. De demuestra que los ataques de canal lateral basados en el análisis pasivo de tráfico cifrado son efectivos para la obtención de información privada de los habitantes del nuevo hogar inteligente.

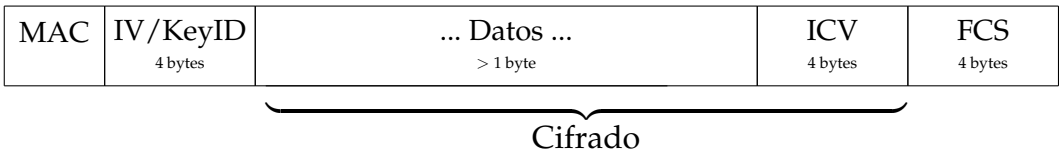


Figura 4.9: Estructura de la trama 802.11 con WEP TKIP

4.3.1. Limitaciones del protocolo 802.11 WiFi en la protección de la privacidad de las comunicaciones

A pesar de que el protocolo 802.11 soporta un conjunto de algoritmos criptográficos orientados a la protección de la seguridad y privacidad de las comunicaciones efectuadas por los nodos integrantes de la red, las características técnicas inherentes al diseño del protocolo lo hacen vulnerable a ataques de canal lateral basados en análisis de tráfico cifrado.

Esta familia de ataques, en lugar de utilizar vulnerabilidades existentes en los algoritmos de cifrado y autenticación, o en sus respectivas implementaciones, es capaz de extraer información útil mediante análisis del tráfico cifrado con el objetivo de inferir información privada relativa a los datos transferidos o el comportamiento de los usuarios que utilizan, directa o indirectamente, la red inalámbrica.

Actualmente, el protocolo WiFi soporta, en instalaciones domésticas, 6 niveles de seguridad en base al cifrado utilizado por la red: WEP, WPA-PSK TKIP, WPA-PSK AES, WPA2-PSK TKIP, WPA2-AES y ausencia de cifrado. En todas las modalidades operativas mencionadas, el algoritmo de cifrado, bien sea RC4 o AES, es utilizado en modo de cifrado de flujo, de tal manera que el paquete WiFi transmitido por la red contendrá en su interior los datos cifrado referentes a la información en texto plano, junto con una serie de cabeceras de control pertenecientes al protocolo específico de cifrado utilizado según dicta el estándar 802.11.

La figura 4.9 describe la composición de un paquete 802.11 cifrado mediante el protocolo WEP. El primer campo, común a todos los paquetes 802.11, independientemente de su tipo de cifrado, es denominado dirección MAC y

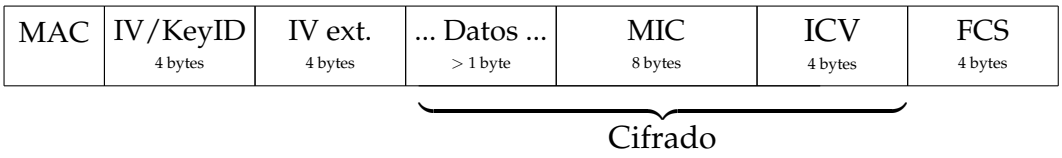


Figura 4.10: Estructura de la trama 802.11 con WPA TKIP

Figura 4.11: Estructura de la trama 802.11 con WPA AES

tiene como objetivo especificar el destino del paquete a nivel de enlace de datos. El segundo campo, denominado como *IV/KeyID*, contiene el vector de inicialización WEP específico para ese paquete, junto con el identificador de la clave. El tercer campo contiene los datos cifrados a ser transferidos y su tamaño será variable en función de la cantidad de carga útil de información que lleve el paquete. El cuarto campo, denominado como *ICV*, contiene el valor de verificación de integridad para los datos cifrados y el último campo es formado por el verificador de integridad de la trama, denominado *FCS*.

La figura 4.10 muestra la composición de un paquete cifrado mediante WPA-PSK [13] TKIP, cuya composición se describe en la figura 4.10, donde se puede observar una estructura semejante a la utilizada en WEP, con dos campos adiciones denominados como *IV Extendido* y *MIC*.

En la variante WPA-PSK AES, los campos IV y IV extendido son reemplazados por la cabecera *CCMP*, a la vez que el campo ICV desaparece, ya que su funcionalidad es integrada en el campo *MIC*. La estructura del paquete WPA-PSK AES se muestra en la figura 4.11.

En todas las modalidades de cifrado, se verifican dos hechos que posibilitan hasta cierto punto la inferencia de información privada mediante el análisis

[13] La estructura del paquete cifrado con TKIP es similar tanto en las implementaciones WPA1 como WPA2

de las comunicaciones cifradas. Primeramente, la cabecera MAC viaja en texto plano, por lo que puede ser observada por un atacante que sea capaz de interceptar remotamente de forma pasiva el trafico cifrado transmitido por una red WiFi.

A pesar de que un atacante no pueda determinar el contenido de las comunicaciones transferidas cuando éstas utilizan cifrado, el análisis de las direcciones *MAC* involucradas en las comunicaciones observadas permite la extracción de información útil del entorno de la red. Mediante este análisis es posible recrear la estructura interna de la red, enumerando los dispositivos de red existentes y determinando los flujos de datos observados.

A modo de ejemplo, un atacante podría determinar la presencia de los habitantes de la vivienda inteligente al detectar tráfico proveniente de la dirección MAC de uno de sus teléfonos móviles. Mediante la triangulación en base a la potencia de la señal recibida, es posible determinar en que área de la vivienda en concreto se encuentra el dispositivo. El análisis del tráfico generado por los diferentes dispositivos inteligentes conectados a la red inalámbrica permitiría la creación de perfiles de comportamiento de los ocupantes de la vivienda.

Por si misma, la información extraída mediante el análisis de las direcciones MAC es ciertamente limitada. Sin embargo, existe otra característica, común a los diferentes modos de operación del cifrado en WiFi, que permite extraer mucha más información relativa a los datos que son transferidos: el tamaño de los paquetes cifrados.

En las diferentes estructuras de paquete correspondientes a los diversos tipos de cifrado, todos los campos poseen un tamaño fijo, con excepción del campo que contiene los datos cifrados del paquete. Este último contendrá el resultado de la aplicación de algoritmo de cifrado (RC4 o AES) sobre los datos a ser cifrados. Dado que el algoritmo de cifrado es utilizado como un cifrado de flujo, el tamaño de los datos cifrados, denominado como S_c, será siempre igual al tamaño de los datos en plano, denominado como S_p, más el tamaño de los campos fijos para el modo de operación de cifrado en vigor, denominado como S_h. Formalmente, $S_c = S_p + S_h$, siendo S_h constante para cada uno de

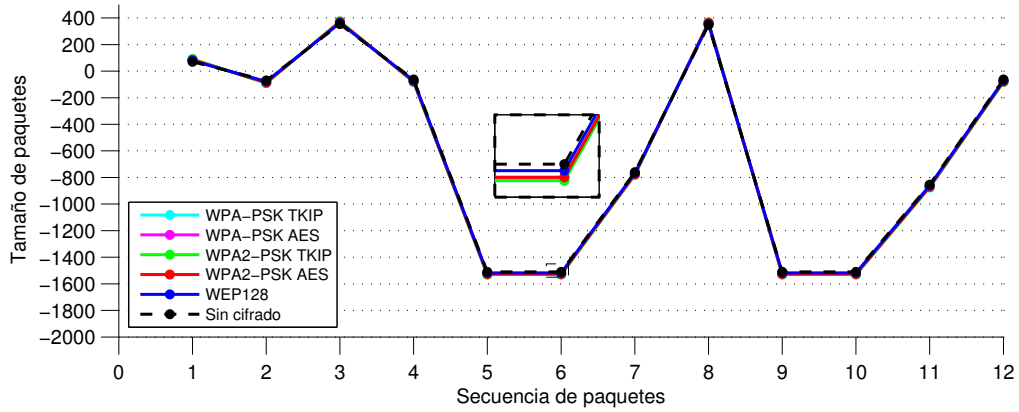


Figura 4.12: Señal generada a partir de la observación del tamaño y orden de paquetes cifrados intercambiados entre un cliente y un servidor para la descarga de un sitio web, bajo cada uno de los tipos posibles de cifrado en WiFi.

los modos de operación del cifrado en WiFi.

Con el objetivo de ilustrar este último punto, se diseña un sencillo experimento donde se simula la presencia de un atacante que observa y registra de forma pasiva el intercambio del paquetes 802.11 de usuario que visualiza un sitio web desde un ordenador. A fin de simplificar el escenario, se utiliza la página por defecto del servidor web *Apache*, que únicamente contiene dos imágenes y una mínima cantidad de texto en formato HTML.

El experimento se realiza bajo los diferentes tipos de cifrado WiFi, incluyendo la ausencia total de cifrado. La figura 4.12 representa la señal extraída del intercambio de paquetes observado entre el ordenador y el servidor durante la visualización de la web simulada. El eje vertical, x, representa el orden de los paquetes y el eje horizontal, y, el tamaño de paquete observado. Los tamaños positivos representan el tamaño de paquetes pertenecientes a la descarga, es decir, originados por el servidor web y transmitidos hacia el cliente. Por otra parte, los tamaños negativos representan el tamaño de los paquetes transferidos desde el cliente hacia el servidor. Cada color representa la señal extraída para el tipo específico de cifrado aplicado en la red inalámbrica, tal y como muestra la leyenda del gráfico.

Se puede observar cómo la sobrecarga de tamaño introducida por las ca-

beceras de los paquetes es de tamaño fijo y constante para cada tipo de red, cumpliéndose que $S_c = S_P + S_h$. Los grandes valores positivos son debidos a contenido específico que es descargado desde la web. En cierta medida, los tamaños de dichos paquetes se convierten en una huella del contenido descargado. Por ejemplo, en el caso de la web del experimento, los tamaños de ambas imágenes contenidas en ella son representados por los picos negativos en el gráfico. El canal de subida, compuesto por la información enviada por el cliente en sus peticiones al servidor, también contiene información útil, sobre todo en el caso de páginas web complejas que hagan uso de contenido dinámico, tal y como será demostrado en las secciones posteriores.

4.3.2. Una metodología para la extracción de huella digitales de sitios web

El análisis de las direcciones MAC origen y destino de los paquetes cifrados interceptados de forma pasiva en una red inalámbrica 802.11, posibilita la reconstrucción de los flujos de datos entre dispositivos de la misma red y hacia Internet. Filtrando mediante la dirección MAC del encaminador de Internet, es posible separar aquel tráfico que tiene como origen y destino Internet. Incluyendo en el filtro la dirección MAC de otro dispositivo de interés de la red, por ejemplo una televisión inteligente o un portátil, se posibilita la discriminación de la navegación web que el usuario efectúa desde dicho dispositivo.

La figura 4.13 muestra la señal extraída experimentalmente de una red inalámbrica para un portátil donde el usuario visita con un navegador [14] consecutivamente 3 sitios webs diferentes. La señal de la figura muestra el tamaño de los paquetes, positivo para el canal de subida de datos y negativos para el de descarga, sobre el orden de recepción por parte del atacante pasivo simulado.

En este caso particular, donde no existe otro tráfico simultaneo adicional desde el mismo dispositivo, es posible determinar claramente la visita a cada

[14] En el caso concreto de este pequeño experimento, el usuario utiliza el navegador Firefox bajo GNU/Linux.

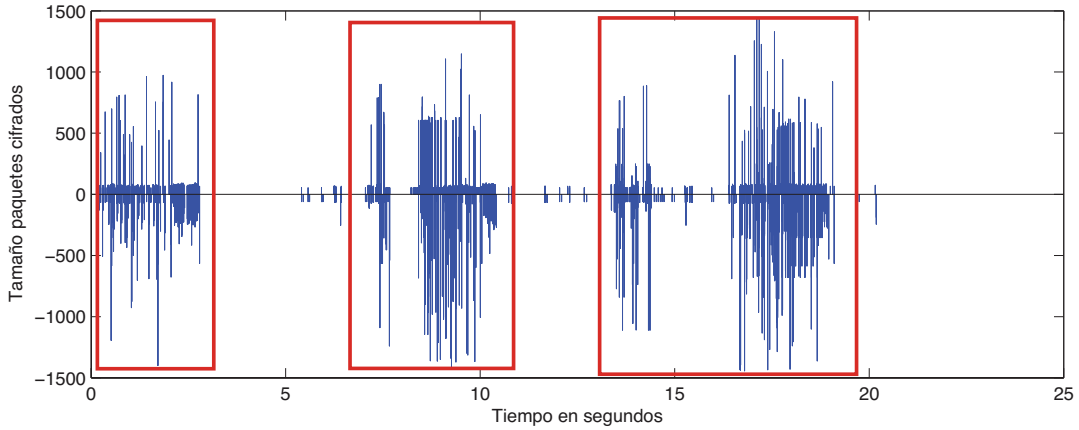


Figura 4.13: Señal Wifi (no filtrada) extraída de la visita consecutiva de 3 sitios web diferentes.

web. Cada una de ellas presenta un patrón específico, representativo del contenido descargado y la forma de interacción con el navegador, derivada de la interpretación del código HTTP y ejecución de código *javascript*.

Sin embargo, sucesivas descargas del mismo sitio web no presentan exactamente el mismo patrón. El motivo de ello es la existencia de factores no deterministas que modifican el patrón de descarga, tales como la perdida de paquetes, peticiones concurrentes hacia el servidor o la ejecución de código dinámico concurrente en el cliente (ie: javascript). Existirá una mayor variabilidad si existe contenido dinámico altamente volátil, como por ejemplo anuncios, o alteraciones en el contenido de la propia página, como por ejemplo ocurre en sitios de noticias.

En cualquier caso, a pesar de los factores mencionados anteriormente, existe una serie de elementos de un sitio web que permanecen más estáticos a lo largo del tiempo, generando un patrón definido que puede ser utilizado a modo de huella digital del sitio web. Este es el caso de los elementos de la plantilla web utilizada, incluyendo código HTML, javascript y imágenes.

La metodología utilizada para la extracción de la huella digital del sitio web, consiste en el filtrado de la señal generada por el tamaño y orden de los paquetes cifrados observados, mediante la detección de aquellos tamaños de paquetes que no se encuentran presentes en sucesivas adquisiciones pertene-

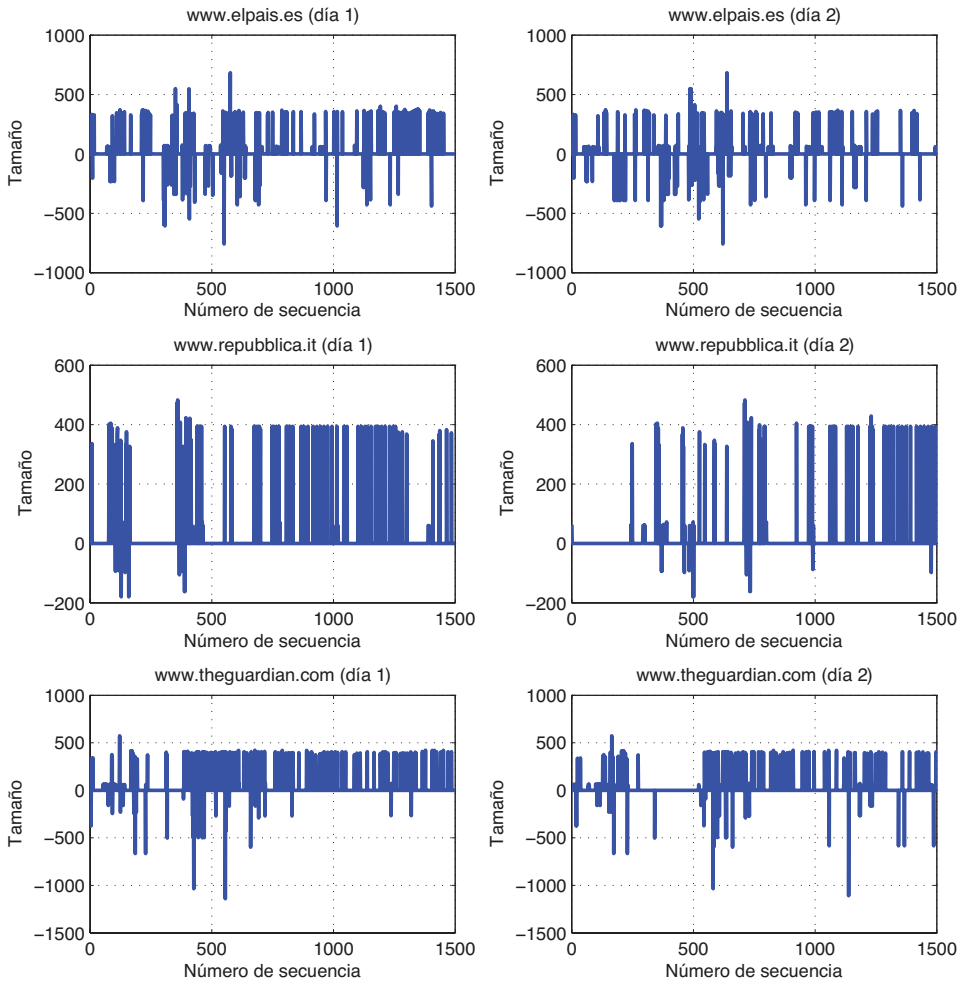


Figura 4.14: Señal filtrada perteneciente a adquisiciones realizadas de tráfico de diferentes sitios web en diferentes días.

cientes al mismo sitio web, ya sea el mismo día o diferentes días. De forma adicional, se elimina el tamaño máximo y mínimo de paquetes, ya que éstos pertenecen al MTU del canal y el ACK del protocolo TCP respectivamente y no añaden un valor informativo relevante.

La figura 4.14 muestra la señal extraída, tras el proceso de filtrado, de 3 periódicos en línea de diferentes nacionalidades en días diferentes. Los valores positivos se refieren a paquetes transmitidos en la dirección de subida (cliente hacia el servidor) y los valores negativos a la dirección de descarga (servidor hacia el cliente). A pesar de que en diferentes días el contenido de la pági-

na del periódico haya cambiado, se puede observar claramente como existen patrones característicos a cada uno en la señal obtenida.

Un último procesado de la señal con vistas a su utilización experimental en los escenarios de ataques en 4.3.3, consiste en la división de la señal en 3 áreas de igual longitud y el cálculo de la concatenación de los histogramas para cada una de las áreas. El resultado será la huella digital del sitio web.

4.3.3. Un escenario experimental de ataque para la detección de sitios web visitados por el usuario de la casa inteligente.

La casa inteligente es un escenario interesante para la aplicación de los ataques de canal lateral mediante análisis de tráfico para la detección de los sitios web visitados por el usuario utilizando cualquiera de los dispositivos inteligentes del ecosistema, tales como la *SmartTV* o el ordenador personal.

El experimento realizado se centra en la detección de las visitas realizadas a periódicos en línea, dado que este escenario supone una importante amenaza a la privacidad de los ciudadanos, al permitir a un atacante inferir información sensible sobre el usuario de la red WiFi, tal como su afiliación política.

El caso específico de los periódicos en línea presenta también una serie de dificultades añadidas en lo referente a la aplicación de la metodología de extracción de la huella digital, dado que su contenido es altamente volátil al encontrarse compuesto de noticias que cambian en cuestión de horas.

Dentro de este contexto, se presentan 2 escenarios de ataque, bajo los cuales un atacante pasivo intercepta las comunicaciones WiFi cifradas pertenecientes a un usuario que visita un determinado periódico en línea. En el primer escenario, el atacante tiene como objetivo determinar qué periódico ha sido visitado, de un total de N periódicos candidatos. Para ello, el atacante utiliza aprendizaje supervisado generando previamente al ataque, durante la fase de aprendizaje, un modelo del tráfico generado por cada uno de los sitios considerados. Posteriormente, cuando el tráfico del usuario es capturado, se utiliza un

Clasificador	Precisión %
k-NN	83.33
SVM con kernel lineal	89.00
Discriminante lineal de cuadrados mínimos de Fisher	100.00

Tabla 4.5: Precisión de los diferentes clasificadores en la detección del sitio web accedido

clasificador para determinar a qué periódico en línea pertenece. Este escenario es denominado como clasificación.

En el segundo escenario, el objetivo del atacante sería comprobar si el usuario esta visitado un determinado sitio web. Para ello, se utiliza una búsqueda de patrones simple, donde la huella del sitio extraída de forma independiente por el atacante, es comparada con aquella extraída de la señal capturada de la víctima. Este escenario es denominado como verificación.

En el escenario de verificación se utilizan 3 clasificadores diferentes, kNN [3], SVM con kernel lineal [25] y el discriminante lineal de cuadrados mínimos de Fisher [159]. Dado que, tanto SVM como Fisher se tratan de clasificadores binarios, se utilizan $\frac{N\times(N-1)}{2}$ clasificadores entrenados sobre todas las posibles combinaciones de clases y se toma como correcta aquella clasificación que resulte en una mayor puntuación.

En todos los casos, se utiliza una primera fase donde se entrena el clasificador con el conjunto de aprendizaje, compuesto por N sitios web ya clasificados, y una segunda fase donde se clasifica una nueva muestra no clasificada. Para la fase de aprendizaje se utiliza tráfico capturado en dos días diferentes, con un total de 10 adquisiciones por día para un conjunto de 10 periódicos en línea, haciendo un total de 200 instancias. La clasificación es realizada con el tráfico capturado un tercer día.

La tabla 4.5 muestra los resultados obtenidos en el escenario de clasificación, para cada uno de los 3 clasificadores utilizados. Las respectivas matrices de confusión se muestran en las tablas 4.6, 4.7 y 4.8.

		Sitio web predecido									
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Sitio web real	#1	30	0	0	0	0	0	0	0	0	0
	#2	0	30	0	0	0	0	0	0	0	0
	#3	0	0	30	0	0	0	0	0	0	0
	#4	2	0	0	0	0	0	20	8	0	0
	#5	0	0	0	0	30	0	0	0	0	0
	#6	0	0	0	0	0	30	0	0	0	0
	#7	0	0	0	0	0	0	30	0	0	0
	#8	0	0	0	0	0	0	0	30	0	0
	#9	0	0	0	0	0	19	0	0	10	1
	#10	0	0	0	0	0	0	0	0	0	30

Tabla 4.6: Matriz de confusión para el clasificador *k*-NN

		Sitio web predecido									
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Sitio web real	#1	28	0	0	0	0	0	0	0	2	0
	#2	0	30	0	0	0	0	0	0	0	0
	#3	0	0	30	0	0	0	0	0	0	0
	#4	0	0	0	0	0	0	30	0	0	0
	#5	0	0	0	0	30	0	0	0	0	0
	#6	0	0	0	0	0	30	0	0	0	0
	#7	0	0	0	0	0	0	30	0	0	0
	#8	1	0	0	0	0	0	0	29	0	0
	#9	0	0	0	0	0	0	0	0	30	0
	#10	0	0	0	0	0	0	0	0	0	30

Tabla 4.7: Matriz de confusión para el clasificador de Maquinas de Vector Soporte

		Sitio web predecido									
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Sitio web real	#1	30	0	0	0	0	0	0	0	0	0
	#2	0	30	0	0	0	0	0	0	0	0
	#3	0	0	30	0	0	0	0	0	0	0
	#4	0	0	0	30	0	0	0	0	0	0
	#5	0	0	0	0	30	0	0	0	0	0
	#6	0	0	0	0	0	30	0	0	0	0
	#7	0	0	0	0	0	0	30	0	0	0
	#8	0	0	0	0	0	0	0	30	0	0
	#9	0	0	0	0	0	0	0	0	30	0
	#10	0	0	0	0	0	0	0	0	0	30

Tabla 4.8: Matriz de confusión para el clasificador Fisher

Se demuestra, por tanto, la viabilidad de determinar el sitio web visitado dentro de un conjunto de sitios web candidatos, incluso aunque el contenido del sitio web haya cambiado y el aprendizaje supervisado haya sido realizado en días previos.

En lo referente al escenario de verificación, donde el atacante tiene como objetivo determinar si el usuario ha visitado una determinada web, se calcula una métrica de distancia entre la señal obtenida de la monitorización pasiva de las comunicaciones cifradas del usuario, denominada como t, y la señal adquirida del sitio web por el atacante, denominada como p. Si n es el número de muestras de las que se compone la huella, p_i y t_i son los elementos en la posición i de p y t, la métrica se calcula de la siguiente manera.

$$S(p,t) = \sum_{i=1}^{n} \text{mín}(p_i, t_i)$$

p y t son normalizados a 1 de tal manera que $S(p,t)$ tome valores entre 0 y 1.

La figura 4.15 muestra el ratio de falsos positivos y falsos negativos para la aplicación de la técnica descrita en el escenario de verificación. El EER, donde

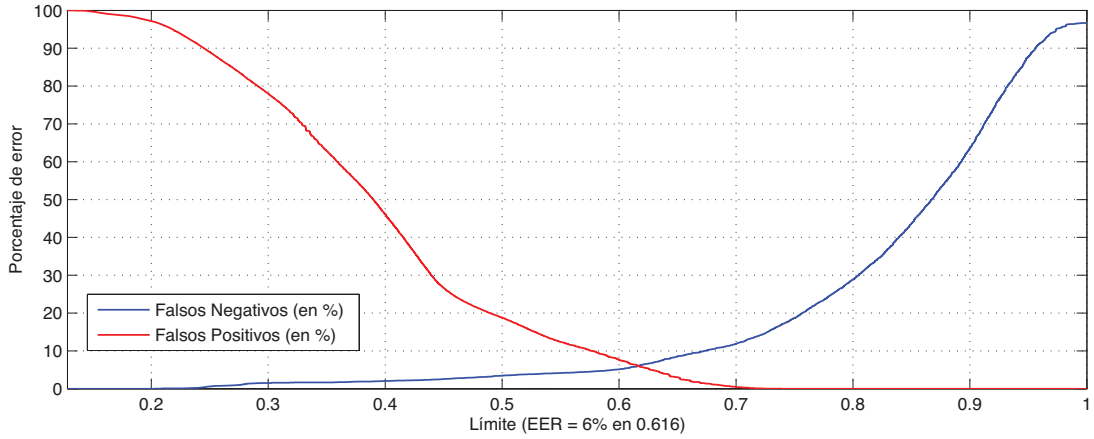


Figura 4.15: Ratio de Falsos Positivos y Falsos Negativos

el ratio de falsos positivos es igual al de falso negativos, se sitúa en tan solo 6 %, tal y como se aprecia en la figura. Un límite mayor reduciría aun más el número de falsos positivos, a expensas de un mayor porcentaje de falsos negativos.

En ambos escenarios, el atacante no requiere conocimiento de la clave de la red WiFi cifrada y no utiliza ninguna vulnerabilidad en el protocolo para realizar el descifrado de los datos. Actualmente ninguno de los mecanismos de seguridad incluidos en el protocolo WiFi protege ante este tipo de ataques de análisis de tráfico.

4.3.4. Conclusiones

Este capítulo de la tesis se centra en el análisis de la efectividad de las medidas de seguridad previstas en el protocolo WiFi para la protección de la seguridad y privacidad de las comunicaciones personales.

Dada la viabilidad de interceptar comunicaciones de datos en redes inalámbricas no cifradas, utilizando hardware estándar disponible en ordenadores personales y software libre, el presente capítulo de la tesis se centra en la investigación de la efectividad del cifrado soportado por el protocolo WiFi para la protección de la seguridad y privacidad de las comunicaciones personales.

Tras un análisis del estado del arte al respecto, en la sección 4.2 se desarrolla

un criptoanálisis del protocolo de cifrado WEP y se demuestra experimentalmente su efectividad para la interceptación pasiva de tráfico WEP cifrado en situaciones reales. La implementación realizada en dicha sección, denominada como *Weplab*, ha sido publicada en [136], convirtiéndose en la primera implementación de un ataque criptográfico contra WEP para GNU/Linux, capaz de criptoanalizar de forma efectiva la clave de cifrado en situaciones reales. Actualmente la implementación realizada forma parte de conocidas distribuciones GNU/Linux, como Ubuntu, y ha sido referenciada por diversas publicaciones.

La investigación realizada sobre el criptoanálisis de WEP, supone una sólida contribución a validación de la hipótesis de la presente tesis y la consecución de sus objetivos, al demostrar que el cifrado WEP no ofrece garantías efectivas de la seguridad y privacidad de las comunicaciones personales, incluso ante atacantes con escasos recursos.

En la sección 4.3 se exploran los límites inherentes a los diferentes tipos de cifrado soportados en las redes WiFi en lo referente a su capacidad de protección de la privacidad de las comunicaciones ante ataques de canal lateral basados en el análisis del tráfico. En dicha sección se describe y demuestra un ataque de este tipo, capaz de determinar los sitios web visitados por un usuario de un dispositivo inteligente conectado a una red WiFi cifrada. Se demuestra que un atacante con escasos recursos es capaz de determinar el sitio web visitado, en un escenario donde la clave de cifrado es desconocida para el atacante.

Las implicaciones del ataque descrito en la privacidad de las comunicaciones personales efectuadas vía inalámbrica en el contexto de la casa inteligente son particularmente relevantes. Se ha demostrado que ninguno de los diferentes modos de cifrado soportados actualmente por WiFi ofrece un nivel de protección suficiente ante este tipo de ataques. Los resultados de la investigación han sido publicados en [138].

Capítulo 5

Contramedidas y propuestas de mejora

En el presente capítulo se analizan los resultados obtenidos en la investigación realizada en los capítulos 2, 3 y 4, con el objetivo de determinar los fallos de diseño e implementación de los cuales han derivado las vulnerabilidades presentes en los diferentes protocolos e identificar una serie de recomendaciones generales a ser tenidas en cuenta por futuros protocolos y estándares.

En base a la investigación realizada y los resultados obtenidos, descritos en los capítulos anteriores, se proponen una serie de contramedidas a ser consideradas para su inclusión en las implementaciones actuales, orientadas a mitigar el riesgo existente para la seguridad y privacidad de las comunicaciones personales que tienen lugar sobre las tecnologías inalámbricas objeto de esta tesis.

5.1. Lecciones aprendidas y sugerencias para futuros estándares e implementaciones

La investigación realizada sobre la seguridad de la generación actual de estándares DECT, GSM y WiFi, en especial en lo referente a sus vulnerabilidades y limitaciones, permite identificar un conjunto de importantes *lecciones aprendidas* valiosas como punto de partida en el diseño e implementación de

futuros protocolos.

En la presente sección se recogen una serie de requisitos o principios de seguridad que, en opinión del autor de esta tesis, deberían ser tenidos en cuenta en el diseño de futuros estándares.

1. Utilización de cifrado en las comunicaciones.

Una de las conclusiones principales alcanzadas en la investigación realizada en esta tesis, es la capacidad que la nueva generación de dispositivos de Radio Definida por Software de bajo coste posee para la monitorización del conjunto actual de protocolos inalámbricos utilizados en las comunicaciones personales.

La presencia de esta nueva tecnología elimina las barreras de acceso tradicionales, presentes en forma de costosos recursos hardware, requeridas por un atacante para llevar a cabo ataques contra la privacidad de las comunicaciones efectuadas por vías inalámbricas.

A la luz de los resultados arrojados por la investigación realizada en la presente tesis respecto a la aplicación de dispositivos SDR para la interceptación de comunicaciones personales efectuadas por vías inalámbricas, el uso de cifrado en las comunicaciones debe ser requisito obligatorio para futuros estándares e implementaciones, tanto para el contenido (datos o voz) como para los datos de control intercambiados.

2. La seguridad de un algoritmo de cifrado debería estar basado exclusivamente en su clave.

Este principio, conocido como el *Principio de Kerckhoff*, fue formulado por primera vez por Auguste Kerckhoffs en 1883 [85] y establece que un sistema criptográfico debe ser seguro incluso si todos sus detalles, excepto la clave utilizada, son conocidos públicamente. Claude Shannon reformularía el principio en 1949 [143] en lo que se conocería como *la máxima de Shannon*, que establece que el diseño de seguridad de un sistema debe realizarse bajo la hipótesis de que el adversario conocerá todos sus detalles.

Las conclusiones alcanzadas tras el análisis de los resultados obtenidos en

los capítulos 2 y 3 determinan que los respectivos algoritmos criptográficos utilizados en ambos estándares para el cifrado de las comunicaciones no son efectivos en la protección de la privacidad y seguridad de las mismas. En estos capítulos de la tesis se demuestra la viabilidad de interceptar comunicaciones cifradas mediante el ataque a las debilidades existentes en los protocolos criptográficos utilizados por ambos estándares.

A pesar de que los algoritmos DSC, utilizado en DECT, y A5/1, utilizado en GSM, sean diferentes en lo referente a sus detalles y funcionamiento, ambos presentan obvias similitudes en su diseño y estructura interna. Sin embargo, la mayor similitud reside en la no conservación del *Principio de Kerckhoff* y la *máxima de Shannon* en sus respectivos diseños. En ambos casos, los detalles de los algoritmos de cifrado, a pesar de ser referenciados en los respectivos estándares públicos, se encontraban disponibles únicamente bajo acuerdos de confidencialidad, resultando opacos al escrutinio por parte de la comunidad académica.

Una vez que sus diseños fueron derivados mediante procesos de ingeniería inversa y una implementación de referencia fuera publicada, los primeros ataques de criptoanálisis no tardarían en llegar, revelando debilidades que terminarían derivando en una ruptura completa de su seguridad, tal como se demuestra en la presente tesis.

En opinión del autor de esta tesis, la aplicación del *Principio de Kerckhoff* y la *máxima de Shannon* durante el proceso de diseño de ambos algoritmos, utilizando algoritmos de cifrado públicos o disponibilizando el diseño de éstos a la comunidad académica durante el proceso, hubiera prevenido la situación actual.

Desde ese punto de vista, se recomienda evitar la utilización de algoritmos de cifrado propietarios cuya implementación no sea pública, siendo preferible el uso de algoritmos de cifrado estándar, cuyo diseño e implementación sean conocidos por la comunidad académica. Idealmente, se deberían considerar algoritmos sin vulnerabilidades estructurales conocidas, con cierto bagaje en lo referente a su resistencia a intentos de criptoanálisis por parte de la comu-

nidad académica y con los que se posea experiencia en su utilización práctica en sistemas de producción con gran número de usuarios.

3. Utilización de longitudes de clave de al menos 256-bit para algoritmos de cifrado simétrico.

Otra debilidad presente en ambos algoritmos de cifrado DSC y A5/1 objeto de investigación en la presente tesis, es su escasa longitud de clave y tamaño del estado interno del algoritmo de cifrado. En ambos casos, el tamaño de clave utilizada posee una longitud de 64-bits, la cual dista mucho de poder considerarse segura actualmente ante ataques de búsqueda exhaustiva.

Según el NIST [10] y las recomendaciones de la Agencia Europea de Seguridad de Redes (ENISA) [45], la longitud mínima recomendada para algoritmos de cifrado simétricos se encuentra fijada actualmente en 128-bits. En previsión del posible escenario de procesamiento cuántico, el tamaño recomendado para su utilización en futuros algoritmos debería fijarse en 256-bits [12].

4. Perfect Forward Secrecy.

Secreto-perfecto-hacia-delante, también denominado como *Perfect Forward Secrecy (PFS)*, puede ser definido como la resistencia de un sistema criptográfico ante un ataque donde un adversario sea capaz de recuperar una clave que le permita descifrar cualquier mensaje cifrado pasado o futuro.

Ninguno de los sistemas criptográficos utilizados en los protocolos DECT, GSM y WiFi, analizados en la tesis, implementa el principio de PFS. Como consecuencia, en todos los casos existe una única clave que, en posesión de un atacante, permite el descifrado de comunicaciones cifradas interceptadas en el pasado o a ser interceptadas en el futuro.

A excepción del protocolo WEP, el resto de protocolos criptográficos utilizados para el cifrado de las comunicaciones por parte de los diferentes estándares analizados, implementa secreto-hacia-delante, también conocido como *Forward Secrecy (FS)*. Como consecuencia, el cifrado de las comunicaciones es efectuado utilizando una clave de sesión específica para una comunicación, de tal manera que si ésta se ve comprometida, el conocimiento de la misma no com-

promete la seguridad de conversaciones pasadas o futuras que puedan haber sido o ser interceptadas. En efecto, los ataques de criptoanálisis demostrados en los capítulos 2 y 3 de la tesis, son capaces de derivar la clave de sesión y descifrar únicamente la comunicación actual

Sin embargo, al no implementar *Perfect Forward Secrecy*, en ambos casos existe una clave *permanente* (UAK en caso de DECT y K_i en caso de GSM), la cual en caso de ser comprometida, permitiría a un atacante descifrar cualquier conversación cuyas claves de sesión hayan sido derivadas de la misma. En efecto, el ataque contra el mecanismo de emparejamiento criptográfico demostrado en el capítulo 2 deriva dicha clave, posibilitando el descifrado de cualquier otra futura comunicación cifrada. Similarmente, un atacante con acceso físico a un teléfono DECT, una tarjeta SIM o un dispositivo WiFi, podría potencialmente extraer las respectivas claves UAK, K_i y PSK y descifrar conversaciones que pudieran haber sido interceptadas y almacenadas en formato cifrado, meses o años antes.

Es la opinión del autor de esta tesis, que *Perfect Forward Secrecy* debería considerarse como un requisito indispensable a ser implementado por sistemas criptográficos de futuros estándares, con el objetivo de ofrecer un grado efectivo de protección de la seguridad y privacidad de las comunicaciones personales efectuadas por medios inalámbricos.

5. Autenticación mutua.

Otro requisito de seguridad importante, con el objetivo de evitar ataques tipo MiTM donde la identidad de la estación base es suplantada por un atacante, es la existencia de autenticación mutua entre los dispositivos de red y la estación base.

La ausencia de autenticación mutua es uno de los principales problemas de seguridad de GSM[1], donde los terminales móviles no tienen forma de auten-

[1]El problema no solo se encuentra presente en GSM, sino también en menor medida en DECT, donde hasta hace unos años la autenticación de la estación base por parte de un terminal DECT no era considerada como obligatoria en el estándar, por lo que posibilitaba la interceptación activa de comunicaciones para llamadas salientes suplantando su identidad.

ticar la identidad de las células GSM, lo que abre las puertas a varios ataques efectivos de interceptación activa de comunicaciones y revelado de identidades de usuarios móviles, tal y como se ha demostrado en el capítulo 3.

6. Diffie-Hellman como mecanismo de intercambio de claves en los emparejamientos de dispositivos.

En el capítulo 2 se describía y demostraba experimentalmente un ataque contra el mecanismo criptográfico de emparejamiento de dispositivos DECT, capaz de derivar la clave UAK intercambiada durante el proceso, posibilitando la interceptación de subsecuentes comunicaciones cifradas.

El ataque es posible porque todos los parámetros involucrados en el cálculo de la clave criptográfica intercambiada son observables por un atacante pasivo que se encuentre monitorizando las comunicación inalámbrica. La única excepción es el propio código PIN definido por el usuario, el cual puede ser fácilmente determinado por búsqueda exhaustiva, tal y como se demuestra en el capítulo 2.

En base a estos resultados, se recomienda la utilización de protocolos de intercambio de claves basados en Diffie-Hellman [42], de tal manera que el intercambio sea seguro incluso ante la presencia de un atacante pasivo que pueda observar el intercambio de mensajes efectuado vía inalámbrica durante el proceso.

7. Utilización de seudónimos de corta duración.

De la investigación de las amenazas de privacidad presentes en los diferentes protocolos de comunicación inalámbrica objeto de la tesis, se desprende que existe un riesgo relevante para la privacidad de los usuarios debido a la existencia de identificadores estáticos cuyo análisis remoto por parte de un atacante puede ser utilizado para inferir información privada de sus utilizadores.

Mientras que DECT y WiFi no utilizan ningún tipo de seudónimos para la protección de la identidad de sus usuarios, GSM prevé el uso de identificadores temporales. Desafortunadamente, frecuentemente el ciclo de vida de dichos identificadores es demasiado largo y el mecanismos previsto por

el estándar puede ser esquivado con el objetivo de obtener el identificativo permanente, tal y como se demuestra experimentalmente en el capítulo 3.

Para futuros estándares de comunicación, se recomienda la utilización de un sistema basado en seudónimos en forma de identificadores temporales con un ciclo corto de vida. El diseño del protocolo debe ser tal, que se evite la transmisión insegura de identificadores permanentes.

5.2. Contramedidas aplicables a los protocolos actuales

La actualización de los algoritmos y protocolos criptográficos existentes en los estándares de comunicación inalámbrica DECT, GSM y WiFi, representa un complicado desafío dada la inmensa cantidad de redes y dispositivos a nivel mundial que hacen uso de dichas tecnologías. Existen más de 1000 millones de dispositivos DECT vendidos y un crecimiento anual de 100 millones de nuevos dispositivos. La red GSM es utilizada actualmente al 95 % de la población mundial [81] y el número total de dispositivos WiFi vendidos durante el año 2014 se estima en varios miles de millones[2].

Uno de los principales males de los que adolece el ecosistema mundial del Internet de las Cosas es la ausencia de mecanismos para la actualización de las implementaciones de los diferentes estándares de comunicación inalámbrica. Un teléfono DECT es un ejemplo de sistema embebido que, en la gran mayoría de los casos, no soporta la actualización de su *firmware* y permanecerá toda su vida útil con la misma implementación del protocolo DECT. Dichos dispositivos, no solo son incapaces de acomodar nuevos estándares de comunicación, o modificaciones en los estándares existentes, sino que irán acumulando vulnerabilidades en sus implementaciones, que amenazarán su propia seguridad

[2]Según el informe de prensa publicado por Gartner el 13 de Febrero de 2014, el número total de *SmartPhones* vendidos en 2013 es estimado en unos 1000 millones, superando por primera vez al número total de ordenadores personales. `http://www.gartner.com/newsroom/id/2665715`

y la del ecosistema.

El reemplazo de los protocolos de comunicación inalámbrica analizados en la presente tesis es una tarea compleja, dada la inmensidad de redes desplegadas utilizando dichas tecnologías y los miles de millones de dispositivos existentes en el mercado que las implementan.

Como alternativa, es posible introducir nuevas versiones de los estándares con renovadas características de seguridad, a la vez que se mantienen las existentes (o ausentes). La idea detrás de dicha aproximación es permitir la coexistencia de dispositivos antiguos y nuevos garantizando su interoperabilidad. Este modelo ha sido implementado, por ejemplo, en la industria de telefonía móvil, donde las redes GSM, 3G y 4G coexisten. Si bien, en términos de funcionalidad, ésta sería una opción viable, en términos de seguridad y privacidad, presenta serias limitaciones.

Por ejemplo, actualmente en telefonía móvil es posible encontrarse dentro de ciudades bajo cobertura GSM y 3G, de tal manera que si se utiliza un terminal compatible con 3G (por ejemplo un Iphone), las llamadas efectuadas y recibidas serán realizadas utilizando este protocolo, no siendo susceptibles de ser interceptadas mediante el abuso de las vulnerabilidades existentes en GSM. Sin embargo, dado que el terminal móvil siempre presenta soporte para el protocolo GSM, un atacante puede fácilmente interferir las frecuencias 3G para eliminar la disponibilidad de cobertura 3G y forzar el uso de la red GSM.

El mismo ataque puede ser realizado a nivel de negociación de protocolo de cifrado cuando varios cifrados de diferente nivel de seguridad sean soportados en el estándar. Este es el caso de GSM y WiFi, donde es posible forzar la negociación de un modo de cifrado más débil susceptible de ser atacado.

Sin embargo, el grado de nivel de seguridad de un protocolo o sistema no debe ser entendido como una medida binaria (blanco o negro), sino como la medida resultante de un proceso de análisis de riesgos, donde se toma en cuenta las vulnerabilidades conocidas, la dificultad de su abuso y las amenazas existentes. Cualquier medida orientada a reducir cualquiera de estos factores, redundará positivamente en el riesgo de seguridad resultante.

Por ello, la actualización, progresiva o no, de los protocolos inalámbricos debe ser acompañada por una serie de medidas orientadas a la reducción de las vulnerabilidades existentes, contribuyendo de esta forma a la reducción del riesgo para la seguridad y privacidad de las comunicaciones personales. A continuación, se enumeran una serie de contramedidas que, en base a la investigación de esta tesis, tienen la capacidad de contribuir a la reducción del riesgo de seguridad y privacidad existente en los estándares actuales.

Se realiza una excepción en el caso del protocolo WEP, objeto de análisis en la primera mitad del capítulo 4, ya que en opinión del autor de esta tesis, el uso de éste debe ser evitado completamente, dados los serios riesgos para la seguridad y privacidad de las comunicaciones que entraña su utilización.

1. Renegociación frecuente de claves de sesión.

Varios de los ataques descritos y demostrados en esta tesis tienen como objetivo criptoanalizar la clave de sesión utilizada para cifrar una comunicación determinada. Para ello, un atacante observa remotamente una cierta cantidad de tráfico cifrado y lo analiza con el objetivo de determinar la clave mediante el proceso de criptoanálisis. Una vez determinada la clave, ésta permitirá al atacante descifrar toda la conversación.

La cantidad específica de tráfico cifrado requerida por el atacante para llevar a cabo el proceso es específica al tipo y efectividad del proceso de criptoanálisis llevado a cabo. En el caso del cifrado DSC del protocolo DECT, el criptoanálisis más efectivo conocido hasta la fecha, desarrollado como parte de esta tesis en el capítulo 2, requiere de unos 4 minutos de conversación cifrada, en el mejor de casos, para obtener una probabilidad superior al 50 % de obtener la clave.

Una contramedida eficaz contra aquellos ataques de criptoanálisis dirigidos a la recuperación de la clave de sesión, consiste en la renegociación frecuente de dicha clave, de tal manera que ésta cambie en varias ocasiones durante la conversación en curso. En el caso específico de DECT, si la clave es renegociada automáticamente cada 30 segundos, la probabilidad de éxito del criptoanálisis sería del 0 %, según los resultados experimentales obtenidos en

el capítulo 2.

En GSM la contramedida propuesta tendría un efecto positivo más marginal, dada la naturaleza del ataque criptográfico demostrado en el capítulo 3, al no requerir tanta cantidad de tráfico cifrado. A pesar de ello, dicha contramedida dificultaría el ataque a la vez que limitaría su impacto, ya que la recuperación de una clave de sesión solo permitiría el descifrado de un fragmento de la comunicación.

2. Minimización del texto plano conocido.

Los ataques más efectivos contra el cifrado de DECT (DSC) y GSM (A5/1), demostrados en los capítulos 2 y 3, se basan en conocimiento de texto plano por parte del atacante. En términos prácticos, esto último implica que el atacante debe averiguar fragmentos de texto plano para los cuales posea el equivalente cifrado. Para ello, se emplean típicamente datos de control con carácter predecible que puedan ser identificados en su forma cifrada.

Una contramedida efectiva para dificultar dichos ataques consiste en la minimización de la cantidad de texto plano predecible, sobre todo en lo referente a datos de control. En el caso específico de DECT, las fuentes de texto plano más viables son aquellas constituidas por los intervalos de silencio y los mensajes de control, por lo que éstos deben ser aleatorizados.

Con respecto a GSM, los diferentes tipos de mensajes de control, tales como *System Information* o *Cipher Mode Complete*, son suministradores fiables de texto plano predecible, por lo que es preciso tomar medidas para aleatorizar su contenido, incluyendo el *padding* de los mensajes y sus campos libres. A pesar de que medidas similares ya hayan sido propuestas [114], actualmente la inmensa mayoría de los operadores móviles europeos no implementan medidas efectivas de este tipo, siendo sus redes vulnerables a ataques pasivos de interceptación GSM, como el demostrado en la sección 3.7 del capítulo 3.

3. Evolución hacia los algoritmos más fuertes de cifrado previstos en los estándares.

En el caso de DECT, las nuevas versiones del estándar incluyen nuevos

algoritmos de autenticación y cifrado, denominados DSAA2 y DSC2 respectivamente. Sin embargo, en el momento de finalizar esta tesis, no existe aún ningún dispositivo del mercado que los implemente.

De forma similar, GSM incluye actualmente un tercer modo de cifrado denominado como A5/3, que no presenta las vulnerabilidades existentes en A5/1 y A5/2. Lamentablemente, dicho algoritmo goza actualmente de un soporte prácticamente nulo entre las redes GSM que operan en territorio europeo.

En lo referente a WiFi, afortunadamente el protocolo WPA presenta una alternativa viable al protocolo WEP, cuyo uso debe ser evitado dadas las serias vulnerabilidades demostradas en el capítulo 4.

En cualquiera de los casos, la evolución hacia algoritmos de cifrado más seguros, a pesar de que contribuya a mitigar el riesgo de privacidad y seguridad, queda lejos de eliminarlo, dado que muchas de las vulnerabilidades demostradas en ésta tesis se refieren al propio protocolo y no tan solo al algoritmo de cifrado.

4. Implementaciones seguras.

De nada sirve que existan medidas de seguridad, efectivas o no, en los estándares, si éstas no son posteriormente implementadas de forma correcta. Muchos de los ataques existentes contra los protocolos actuales utilizan fallos de implementación con el objetivo de atacar el sistema.

Al margen de la propia seguridad del dispositivo, bien sea estación base o terminal móvil, que pueda comprometer de forma obvia las comunicaciones, existen ciertos factores relevantes a ser tenidos en cuenta por implementadores y operadores.

- Autenticación de operaciones. Siempre que sea posible es preciso autenticar tanto al cliente como al servidor en todas las operaciones realizadas. En el caso específico de DECT, muchas implementaciones del mercado son vulnerables a ataques MiTM debido a que no realizan una autenticación de la estación base, la cual se prevé en el estándar.

- Números aleatorios. La seguridad criptográfica de un sistema se ve completamente comprometida si un adversario es capaz de predecir los números que deberían ser generados de forma aleatoria por los participantes de la comunicación como parte del protocolo criptográfico. Existen ejemplos documentados de como dicho vector de ataque puede comprometer la seguridad de una comunicación DECT [105] o WiFi [20].

- Claves por defecto o predecibles. En un sistema criptográfico perfecto, toda la seguridad reside en la clave de cifrado. Si existen claves por defecto o fácilmente predecibles por un atacante, toda la seguridad del sistema se verá comprometida. Éste es desafortunadamente el caso de muchas instalaciones WiFi, donde la contraseña por defecto del encaminador o punto de acceso es generada matemáticamente en base a parámetros observables o predecibles por un atacante.

5. Utilización de seudónimos con ciclo de vida corto.

Siempre que sea posible dentro de las limitaciones impuestas por el estándar correspondiente, las implementaciones deberían utilizar seudónimos con un ciclo de vida corto, con el objetivo de proteger la identidad, y por extensión privacidad, de los utilizadores de las redes.

En GSM, que prevé el uso de identificadores temporales a modo de seudónimos, es posible renovarlos más frecuentemente, por ejemplo siempre que exista una comunicación, ya que actualmente numerosos operadores mantienen el TMSI constante durante horas e incluso días.

En redes DECT y WiFi, cuyos estándares no prevén el uso de seudónimos, se debería explorar la posibilidad de cambiar los identificadores permanentes manejados por las diferentes implementaciones, siempre que ésto no interfiera con el funcionamiento del protocolo. Por ejemplo, en WiFi sería factible que cada dispositivo cambiara periódicamente su dirección MAC por una aleatoria, ya que esto dificultaría la ejecución de ataques contra la privacidad, como el demostrado en el capítulo 4, y no resultaría un problema de interoperabili-

dad con otros dispositivos de la red [3], ya que el protocolo ARP resolvería de forma automática la asociación de la nueva MAC para aquellos dispositivos que posean la clave de cifrado de la red.

6. Sacrificio de protocolos y/o modalidades de operación inseguras.

Existen una serie de protocolos dentro de los estándares de comunicación inalámbrica analizados en la presente tesis, cuya mera existencia supone una grave amenaza para la seguridad y privacidad de las comunicaciones. El autor de esta tesis tiene la opinión de que el sacrificio de las funcionalidades ofrecidas por estos protocolos debería considerarse seriamente, a fin de reducir el riesgo que éstos suponen para la seguridad y privacidad de las comunicaciones.

Dentro de los posibles candidatos, se han identificado los siguientes por su alto riesgo para la privacidad de las comunicaciones.

- GSM A5/0. GSM soporta el modo de operación A5/0 que representa la ausencia de cifrado. Al margen del grave riesgo de privacidad que dicho modo de operación representa para las comunicaciones efectuadas por el usuario, el hecho de que los terminales móviles existentes en el mercado lo implementen, posibilita la ejecución de efectivos ataques activos de interceptación, como el descrito y demostrado experimentalmente en el capítulo 3. Adicionalmente, ninguno de los teléfonos móviles a los que el autor de esta tesis ha tenido acceso ha sido capaz de mostrar un signo de aviso ante la presencia de una red en modalidad A5/0 durante los experimentos realizados en el capítulo 3.

- GSM A5/2. El modo de cifrado A5/2 de GSM fue creado como la variante débil del cifrado A5/1, objeto de investigación en el capítulo 3, y su soporte por parte de los terminales móviles del mercado supone un riesgo

[3]Existe una obvia incompatibilidad con el filtrado MAC en caso de que éste se encuentre activo. Sin embargo, en la opinión del autor de esta tesis, el filtrado MAC es una medida totalmente ineficaz para la protección de la seguridad y privacidad de la red, por lo que su sacrificio en aras de un uso dinámico de direcciones MAC sería beneficioso

a la seguridad de las comunicaciones, ya que posibilita ciertos tipos de ataques activos. Desde hace unos años [4] las entidades de estandarización ETSI y 3GPP han iniciado el proceso de retirada del algoritmo A5/2 para las nuevos terminales (ME) y las redes GSM existentes.

- WPS (*Wi-Fi Protected Setup*). El protocolo WPS adolece de serios problemas de seguridad y su utilización en la modalidad de código PIN puede suponer una grave amenaza para la privacidad y seguridad de las redes WiFi, en aquellos casos donde no existan contramedidas específicas.

[4] Más información sobre el proceso de retirada del algoritmo A5/2 puede encontrarse en `http://security.osmocom.org/trac/wiki/A52_Withdrawal`

Capítulo 6

Conclusiones y futuras líneas de trabajo

6.1. Conclusiones

En respuesta al objetivo principal de la presente tesis, los resultados obtenidos validan la hipótesis formulada de que el conjunto de tecnologías de transmisión inalámbricas utilizadas en comunicaciones personales, DECT, GSM y WiFi, no ofrece actualmente suficientes garantías en lo referente a la protección de la seguridad y la privacidad de las comunicaciones.

Se ha demostrado que la nueva generación de dispositivos de Radio Definida por Software de bajo coste supone un aumento del riesgo para la privacidad de las comunicaciones en las tecnologías analizadas, ya que permiten llevar a cabo ataques activos y pasivos de interceptación, los cuales requerían tradicionalmente de la utilización de costoso hardware especializado.

La implementación desarrollada como parte de la investigación realizada en el capítulo 2, ha demostrado ser eficaz en la interceptación pasiva de comunicaciones DECT mediante el uso de dispositivos SDR de bajo coste. Similarmente, los resultados experimentales del capítulo 3 arrojan resultados semejantes para la interceptación activa y pasiva de comunicaciones GSM.

En lo referente a las medidas de seguridad, principalmente autenticación y cifrado, contempladas en los diferentes protocolos objeto de investigación en

esta tesis, se ha demostrado que éstos no pueden considerarse seguros en el contexto actual, resultando ineficaces para la protección de la privacidad de las comunicaciones, incluso ante atacantes casuales con escasos recursos a su alcance.

En el capítulo 2 se han analizado los ataques existentes contra el algoritmo de cifrado DSC de DECT y se ha desarrollado un nuevo método de criptoanálisis que ha demostrado ser cuatro veces más eficaz que el mejor ataque documentado en la literatura. En el mismo capítulo también se ha demostrado de forma experimental la interceptación pasiva de comunicaciones DECT cifradas atacando el protocolo de emparejamiento de dispositivos previsto en el estándar DECT-GAP.

En lo referente a GSM, en el capítulo 3 se ha demostrado experimentalmente como la ausencia de autenticación mutua puede ser fácilmente aprovechada por un atacante para realizar la interceptación pasiva de comunicaciones. Los experimentos realizados en el ataque al algoritmo de cifrado A5/1, han demostrado la viabilidad de realizar una interceptación pasiva de las comunicaciones, rompiendo la clave de cifrado de la comunicación en cuestión de minutos.

En el capítulo 4 se ha explorado y desarrollado una novedosa implementación[1] que es capaz de derivar la clave criptográfica utilizada en una red cifrada con protocolo WEP mediante el análisis de varios minutos de tráfico cifrado bajo escenarios reales. Posteriormente, en el mismo capítulo, se ha explorado el riesgo que los ataques de canal lateral basados en el análisis del tráfico cifrado presentan sobre la privacidad de las comunicaciones realizadas sobre redes WiFi en entornos domésticos, aun cuando éstas utilicen un cifrado considerado como seguro. Se ha demostrado cómo es posible determinar los sitios web

[1]La investigación sobre la seguridad de WEP fue realizada en el comiendo del doctorado en un contexto en el que los ataques de criptoanálisis contra el cifrado WEP eran considerados meramente teóricos por parte de la industria con difícil aplicación práctica. La implementación realizada del criptoanálisis de WEP, descrita en la sección 4.2, fue publicada [136], anteriormente al lanzamiento de otras herramientas similares como *Aircrack*, convirtiéndose en la herramienta más eficaz para GNU/Linux en el ataque criptográfico al protocolo WEP.

que son visitados por un usuario de la red, incluso cuando ésta utilice cifrado y la seguridad de la clave no se vea comprometida.

Una conclusión importante a la que ha llegado el autor de la presente tesis, es que actualmente no son necesarios grandes recursos o hardware especializado para la interceptación de comunicaciones personales sobre protocolos DECT, GSM y WiFi. Este hecho, en combinación con las vulnerabilidades demostradas en sus algoritmos y protocolos de cifrado, eleva de forma alarmante el riesgo para la seguridad y privacidad de las comunicaciones personales llevadas a cabo por vía inalámbrica de forma cotidiana por cientos de millones de usuarios.

En base a las conclusiones obtenidas en la investigación realizada, en el capítulo 5 se han propuesto una serie de contramedidas a ser consideradas por las implementaciones existentes, con el objetivo de mitigar el riesgo que las vulnerabilidades de los protocolos de comunicación inalámbrica investigados representan para la privacidad y seguridad de las comunicaciones personales. Sin embargo, es la opinión del autor de la presente tesis, que la única solución realista es el reemplazado de los protocolos actuales o al menos, donde sea posible, sus algoritmos y protocolos de seguridad. A tal efecto, en el mismo capítulo, se han identificado una serie de requisitos que el diseño de una nueva generación de algoritmos y protocolos de seguridad debería cumplir, para poder proteger de forma efectiva la seguridad y privacidad de las comunicaciones personales efectuadas por vías inalámbricas.

6.2. Futuras líneas de trabajo

El conjunto de estándares de comunicación inalámbrica actuales involucrados en la transmisión de datos personales se encuentra actualmente en plena evolución. La necesidad de mayor ancho de banda en transmisiones inalámbricas y menor consumo energético, ha derivado en el desarrollo de una nueva generación de estándares, entre los que se encuentran LTE, DECT ULE y la eventual evolución del estándar de comunicaciones inalámbricas WiFi.

En el contexto actual, la seguridad y privacidad de las comunicaciones efectuadas por vía inalámbrica ha adquirido una relevancia particular. La evolución natural de la investigación realizada en esta tesis se dirige hacia la aplicación práctica de las lecciones aprendidas y conclusiones alcanzadas, a la nueva generación de estándares de comunicaciones, con el objetivo de alcanzar en ésta las mayores garantías de privacidad y seguridad.

De forma complementaria, se pueden considerar nuevas líneas de investigación, fuera de los límites naturales de la presente tesis, enfocadas en el estudio de la seguridad de los propios dispositivos involucrados en las comunicaciones inalámbricas, incluyendo su componente software. Una nueva línea de investigación actualmente abierta pretende cubrir el amplio abanico de aplicaciones móviles existentes para teléfonos inteligentes, investigando posibles métodos para la detección automatizada y colaborativa de amenazas para la privacidad y seguridad de los usuarios.

Otra posible vertiente de investigación se encuentra enfocada al análisis de las implicaciones de seguridad y privacidad derivadas del concepto emergente de la Casa Inteligente. A este respecto, la investigación realizada en esta tesis, centrada en la seguridad de los protocolos inalámbricos utilizados para la intercomunicación de dispositivos en dicho contexto, podría ser extendida para incluir la seguridad de los propios dispositivos embebidos, los cuales, por su naturaleza, son actualmente considerados como uno de los principales puntos débiles del ecosistema de IoT.

Finalmente, en vista de la reciente eclosión de dispositivos ponibles considerados como pertenecientes al ámbito de salud, tales como los relojes inteligentes, futuras líneas de investigación podrán ser concebidas con el objetivo de comprender mejor sus implicaciones potenciales para la privacidad de sus usuarios, contexto en el cual la presente tesis podrá considerarse como punto de partida.

Bibliografía

[1] S. Ahmad. WPA too. Hole 196. In *DefCon 18*, 2010.

[2] B. Alecu. SMS fuzzing – SIM toolkit attack. In *DefCon 21*, 2013.

[3] N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46:175–185, 1992.

[4] R. Anderson. An implementation of A5. Newsgroup sci.crypt,alt.security,uk.telecom, 1994.

[5] Anónimo. Thank you Bob Anderson. Cypherpunks mailing list, 1994.

[6] M. Arapinis, L. I. Mancini, E. Ritter, and M. Ryan. Privacy through pseudonymity in mobile telephony systems. In *21st Annual Network and Distributed System Security Symposium (NDSS'14)*, 2014.

[7] W. A. Arbaugh. An inductive chosen plaintext attack against WEP/-WEP2. *IEEE document*, 802(01):230, 2001.

[8] E. Barkan and E. Biham. Conditional estimators: An effective attack on A5/1. In *Proceedings of the 12th International Conference on Selected Areas in Cryptography*, SAC'05, pages 1–19, Berlin, Heidelberg, 2006. Springer-Verlag.

[9] E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *Journal of Cryptology*, 21(3):392–429, Mar. 2008.

[10] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. NIST special publication 800-57. Technical report, National Institute of Standards and Technology, 2012.

[11] M. Beck. Enhanced TKIP Michael attacks. `http://download.aircrack-ng.org/wiki-files/doc/enhanced_tkip_michael.pdf`, 2010. Accedido el 11-08-2014.

[12] D. J. Bernstein. Grover vs. Mceliece. In *Proceedings of the Third International Conference on Post-Quantum Cryptography*, PQCrypto'10, pages 73–80, Berlin, Heidelberg, 2010. Springer-Verlag.

[13] E. Biham and O. Dunkelman. Cryptanalysis of the A5/1 GSM stream cipher. In *Proceedings of the First International Conference on Progress in Cryptology*, INDOCRYPT '00, pages 43–51, London, UK, UK, 2000. Springer-Verlag.

[14] A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '00, pages 1–13, London, UK, UK, 2000. Springer-Verlag.

[15] A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of A5/1 on a PC. In *Proceedings of the 7th International Workshop on Fast Software Encryption*, FSE '00, pages 1–18, London, UK, UK, 2001. Springer-Verlag.

[16] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine. Privacy vulnerabilities in encrypted HTTP streams. In *Proceedings of the 5th International Conference on Privacy Enhancing Technologies*, PET'05, pages 1–11, Berlin, Heidelberg, 2006. Springer-Verlag.

[17] A. Bittau. Additional weak IV classes for the FMS attack. `http://www0.cs.ucl.ac.uk/staff/a.bittau/sorwep.txt`, 2003. Accedido el 10-08-2014.

[18] A. Bittau, M. Handley, and J. Lackey. The final nail in WEP's coffin. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, SP '06, pages 386–400, Washington, DC, USA, 2006. IEEE Computer Society.

[19] E. Blossom. GNU Radio: Tools for exploring the radio frequency spectrum. *Linux Journal*, 2004(122):4–, June 2004.

[20] D. Bongard. Offline bruteforce attack on WiFi Protected Setup. In *Passwordscon 2014*, 2014.

[21] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, pages 180–189, New York, NY, USA, 2001. ACM.

[22] M. Briceno, I. Goldberg, and D. Wagner. An implementation of the GSM A3A8 algorithm. `https://www.unix-ag.uni-kl.de/~conrad/krypto/misc/a5.html`, 1998. Accedido el 02-08-2014.

[23] M. Briceno, I. Goldberg, and D. Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 voice privacy encryption algorithms. `http://www.scard.org/gsm/a51.html`, 1999. Accedido el 02-08-2014.

[24] B. Brumley. A3/A8 & COMP128. T-79.514 Special Course on Cryptology, 2004.

[25] C. J. C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998.

[26] D. A. Burgess, H. S. Samra, and otros. The OpenBTS project. `http://cs.ru.ac.za/research/g09b0279/UsefulPapers/OpenBTSProject.pdf`, 2008.

[27] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 605–616, New York, NY, USA, 2012. ACM.

[28] R. Chaabouni. Break WEP faster with statistical analysis. Master's thesis, École Polytechnique Fédérale de Lausanne, 2006.

[29] N. Chan and H. Wong. Código fuente de iListener. `https://github.com/RabbitNick/iListener`, 2013.

[30] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for GSM phones. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, UbiComp'06, pages 225–242, Berlin, Heidelberg, 2006. Springer-Verlag.

[31] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, pages 191–206, Washington, DC, USA, 2010. IEEE Computer Society.

[32] H. Cheng and R. Avnur. Traffic analysis of SSL encrypted web browsing. `http://citeseer.ist.psu.edu/656522.html`, 1998. Accedido el 16-08-2014.

[33] CISCO. 802.11ac : The Fifth Generation of Wi-Fi. Technical report, CISCO Networks, 2012.

[34] I. Coisel and I. Sanchez. Practical interception of DECT encrypted voice communications in Unified Communications environments. In *Proceedings of the IEEE Joint Intelligence & Security Informatics Conference*, IEEE ICCST 2014, 2014.

[35] I. Coisel and I. Sanchez. Improved cryptanalysis of the DECT Standard Cipher. In *Cryptographic Hardware and Embedded Systems*, CHES 2015, 2015.

[36] G. Combs. Manual de usuario de WireShark. `https://www.wireshark.org/docs`, 1998–2014.

[37] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. Identification via location-profiling in GSM networks. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, WPES '08, pages 23–32, New York, NY, USA, 2008. ACM.

[38] C. Devine. Aircrack, yet another WEP cracking tool for Linux. `http://www.netstumbler.org/unix-linux/aircrack-yet-another-wep-cracking-tool-for-linux-t11878.html`, 2004. Accedido el 12-08-2014.

[39] C. Devine and T. D'Otreppe. Código fuente de Aircrack-ng. `http://www.aircrack-ng.org`, 2005–2014.

[40] K. Devine. Default key algorithm in Thomson and BT home hub routers. `http://www.gnucitizen.org/blog/default-key-algorithm-in-thomson-and-bt-home-hub-routers`, 2008. Accedido el 12-08-2014.

[41] B. Diaconescu. Evaluating GSM A5/1 security on hopping channels. `http://yo3iiu.ro/blog/wp-content/uploads/2012/04/Evaluating_GSM_hopping1.pdf`, 2011. Accedido el 02-07-2014.

[42] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[43] P. Ekdahl and T. Johansson. Another attack on A5/1. *IEEE Transactions on Information Theory*, 49(1):284–289, Jan. 2003.

[44] T. Engel. Locating mobile phones using Signalling System 7. In *25th Chaos Computer Congress*, 2008.

[45] ENISA. Algorithms, Key Sizes and Parameters Report. Technical report, European Union Agency for Network and Information Security, 2013.

[46] ETSI. New Generation DECT, Part 2: Support of transparent IP packet data. Technical report, European Telecommunications Standards Institute, 2007.

[47] ETSI. New Generation DECT, Part 4: Software Update Over The Air (SUOTA). Technical report, European Telecommunications Standards Institute, 2009.

[48] ETSI. DECT, Common Interface (CI), Part 1: Overview. Technical report, European Telecommunications Standards Institute, 2013.

[49] ETSI. DECT, Common Interface (CI), Part 2: Physical Layer (PHL). Technical report, European Telecommunications Standards Institute, 2013.

[50] ETSI. DECT, Common Interface (CI), Part 3: Medium Access Control (MAC) layer. Technical report, European Telecommunications Standards Institute, 2013.

[51] ETSI. DECT, Common Interface (CI), Part 4: Data Link Control (DLC) layer. Technical report, European Telecommunications Standards Institute, 2013.

[52] ETSI. DECT, Common Interface (CI), Part 5: Network (NWK) layer. Technical report, European Telecommunications Standards Institute, 2013.

[53] ETSI. DECT, Common Interface (CI), Part 6: Identities and addressing. Technical report, European Telecommunications Standards Institute, 2013.

[54] ETSI. DECT, Common Interface (CI), Part 7 : Security features. Technical report, European Telecommunications Standards Institute, 2013.

[55] ETSI. DECT, Common Interface (CI), Part 8: Speech and audio coding and transmission. Technical report, European Telecommunications Standards Institute, 2013.

[56] ETSI. DECT, Generic Access Profile (GAP). Technical report, European Telecommunications Standards Institute, 2013.

[57] ETSI. Sitio web oficial de ETSI DECT. http://www.etsi.org/technologies-clusters/technologies/dect, 2013. Accedido el 10-08-2014.

[58] ETSI. New Generation DECT, Part 1 : Wideband speech. Technical report, European Telecommunications Standards Institute, 2014.

[59] ETSI. New Generation DECT, Part 3 : Extended wideband speech services. Technical report, European Telecommunications Standards Institute, 2014.

[60] ETSI. New Generation DECT, Part 5: Additional feature set nr. 1 for extended wideband speech services. Technical report, European Telecommunications Standards Institute, 2014.

[61] N. Ferguson. Re: DOS attack on WPA 802.11? Lista de distribución de correo cryptography@wasabisystems.com, 2002.

[62] S. R. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, SAC '01, pages 1–24, London, UK, UK, 2001. Springer-Verlag.

[63] T. Gendrullis, M. Novotný, and A. Rupp. A real-world attack breaking A5/1 within hours. In *Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '08, pages 266–282, Berlin, Heidelberg, 2008. Springer-Verlag.

[64] I. Goldberg, D. Wagner, and L. Green. The real-time cryptanalysis of A5/2. In *Proceedings of Rump session of Crypto'99*, volume 99, pages 239–255, 1999.

[65] N. Golde, K. Redon, and J.-P. Seifert. Let me answer that for you: Exploiting broadcast information in cellular networks. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pages 33–48, Berkeley, CA, USA, 2013. USENIX Association.

[66] J. D. Golic. Cryptanalysis of alleged A5 stream cipher. In *Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'97, pages 239–255, Berlin, Heidelberg, 1997. Springer-Verlag.

[67] T. S. Group. Código fuente de AirSnort. `http://airsnort.shmoo.com`, 2002.

[68] Grugq. Base jumping: Attacking GSM base station systems and mobile phone base bands. In *BlackHat 2010 Lecture Notes*, 2010.

[69] T. Güneysu, T. Kasper, M. Novotný, C. Paar, and A. Rupp. Cryptanalysis with COPACOBANA. *IEEE Transactions on Computers*, 57(11):1498–1513, Nov. 2008.

[70] F. M. Halvorsen, O. Haugen, M. Eian, and S. F. Mjølsnes. An improved attack on TKIP. In *Proceedings of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age*, NordSec '09, pages 120–132, Berlin, Heidelberg, 2009. Springer-Verlag.

[71] H. Handschuh and P. Paillier. Reducing the collision probability of alleged COMP128. In *Proceedings of the The International Conference on Smart Card Research and Applications*, CARDIS '98, pages 366–371, London, UK, UK, 2000. Springer-Verlag.

[72] M. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, Sept. 2006.

[73] D. Herrmann, R. Wendolsky, and H. Federrath. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, CCSW '09, pages 31–42, New York, NY, USA, 2009. ACM.

[74] A. Hintz. Fingerprinting websites using traffic analysis. In *Proceedings of the 2Nd International Conference on Privacy Enhancing Technologies*, PET'02, pages 171–178, Berlin, Heidelberg, 2003. Springer-Verlag.

[75] D. Hulton. Practical exploitation of RC4 weaknesses in WEP environments. `http://www.dartmouth.edu/~madory/RC4/wepexp.txt`, 2002. Accedido el 08-08-2014.

[76] D. Hulton and S. Miller. Intercepting mobile phone traffic. In *BlackHat Europe 2008*, 2008.

[77] N. Husted and S. Myers. Mobile location tracking in metro areas: Malnets and others. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 85–96, New York, NY, USA, 2010. ACM.

[78] IEEE. 802.11-1997 - IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, Institute of Electrical and Electronics Engineers, 1999.

[79] IEEE. ANSI/IEEE standard 802.11 i: Amendment 6: Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications. Technical report, Institute of Electrical and Electronics Engineers, 2004.

[80] IEEE. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Technical report, Institute of Electrical and Electronics Engineers, 2007.

[81] ITU. ICT facts and figures. `http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf`, 2014. Accedido el 15-07-2014.

[82] R. J. Jenkins. ISAAC and RC4. http://burtleburtle.net/bob/rand/isaac.html, 1996. Accedido el 10-08-2014.

[83] T. Johansson and F. Jönsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'99, pages 347–362, Berlin, Heidelberg, 1999. Springer-Verlag.

[84] J. Keller and B. Seitz. A hardware-based attack on the A5/1 stream cipher. *ITG FACHBERICHT*, pages 155–158, 2001.

[85] A. Kerckhoffs. *La cryptographie militaire*. University Microfilms, 1978.

[86] A. Klein. Attacks on the RC4 stream cipher. *Des. Codes Cryptography*, 48(3):269–286, Sept. 2008.

[87] KoreK. Chopchop (experimental WEP attacks). http://www.netstumbler.org/showthread.php?t=12489, 2004. Accedido el 11-08-2014.

[88] KoreK. Next generation of WEP attacks? http://www.netstumbler.org/news/next-generation-of-wep-attacks-t12277.html, 2004. Accedido el 11-08-2014.

[89] D. F. Kune, J. Koelndorfer, and Y. Kim. Location leaks on the GSM air interface. In *Proceedings of the 19th Network and Distributed System Security Symposium*, 2012.

[90] P. Langlois. SCTPscan - finding entry points to SS7 networks & telecommunication backbones. In *BlackHat Europe 2007*, 2007.

[91] P. Langlois. Getting in the SS7 kingdom: hard technology and disturbingly easy hacks to get entry points in the walled garden. In *HackitoErgoSum 2014*, 2014.

[92] P. Langlois and V. Brunet. SCCP hacking, attacking the SS7 & SIGTRAN applications one step further and mapping the phone system. In *26th Chaos Communication Congress*, 2009.

[93] M. Liberatore and B. N. Levine. Inferring the source of encrypted HTTP connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 255–263, New York, NY, USA, 2006. ACM.

[94] A. Linz and A. Hendrickson. Efficient implementation of an I-Q GMSK modulator. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(1):14–23, Jan 1996.

[95] S. Lucks, A. Schuler, E. Tews, R.-P. Weinmann, and M. Wenzel. Attacks on the DECT authentication mechanisms. In *Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, CT-RSA '09, pages 48–65, Berlin, Heidelberg, 2009. Springer-Verlag.

[96] L. Lueg. Código fuente de Pyrit. `https://code.google.com/p/pyrit`, 2008–2014.

[97] Magnus Glendrange;, K. Hove, and E. Hvideberg. Decoding GSM. Master's thesis, Norwegian University of Science and Technology, 2010.

[98] I. Mantin. A practical attack on the fixed RC4 in the WEP mode. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3788 LNCS, pages 395–411, 2005.

[99] A. Maximov, T. Johansson, and S. Babbage. An improved correlation attack on A5/1. In *Proceedings of the 11th International Conference on Selected Areas in Cryptography*, SAC'04, pages 1–18, Berlin, Heidelberg, 2005. Springer-Verlag.

[100] P. McHardy, A. Schuler, and E. Tews. Interactive decryption of DECT phone calls. In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, WiSec '11, pages 71–78, New York, NY, USA, 2011. ACM.

[101] W. Meier and O. Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, 1(3):159–176, Jan. 1989.

[102] A. Mengele and E. Tews. Security of Digital Enhanced Cordless Telecommunication (DECT) devices for residential use. Master's thesis, Technische Universität Darmstadt, 2009.

[103] B. Miller, L. Huang, A. D. Joseph, and J. D. Tygar. I know why you went to the clinic: Risks and realization of HTTPS traffic analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8555 LNCS, pages 143–163, 2014.

[104] V. Moen, H. Raddum, and K. J. Hole. Weaknesses in the temporal key hash of WPA. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):76–83, Apr. 2004.

[105] H. G. Molter, K. Ogata, E. Tews, and R.-P. Weinmann. An efficient FPGA implementation for an DECT brute-force attacking scenario. In *Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications*, ICWMC '09, pages 82–86, Washington, DC, USA, 2009. IEEE Computer Society.

[106] T. Moore, T. Kosloff, J. Keller, G. Manes, and S. Shenoi. Signaling System 7 (SS7) network security. In *The 2002 45th Midwest Symposium on Circuits and Systems*, volume 3 of *MWSCAS-2002*, pages III–496–III–499 vol.3, Aug 2002.

[107] R. Moskowitz. Weakness in passphrase choice in WPA interface. `http://wifinetnews.com/archives/2003/11/weakness_in_passphrase_choice_in_wpa_interface.html`, 2003. Accedido el 14-08-2014.

[108] C. Mulliner, N. Golde, and J.-P. Seifert. SMS of death: From analyzing to attacking mobile phones on a large scale. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 24–24, Berkeley, CA, USA, 2011. USENIX Association.

[109] S. Munaut. Further hacks on the Calypso platform. In *29th Chaos Communication Congress*, 2012.

[110] R. Networks. Código fuente de OpenBTS. `http://openbts.org`, 2007–2014.

[111] T. Newsham. Cracking WEP keys. In *BlackHat 2001 Lecture Notes*, 2001.

[112] K. Nohl. Attacking phone privacy crypto basics time-memory trade-offs. In *BlackHat 2010 Lecture Notes*, 2010.

[113] K. Nohl. Rooting SIM cards. In *BlackHat 2013 Lecture Notes*, 2013.

[114] K. Nohl and L. Melette. Defending mobile phones. In *28th Chaos communication congress*, 2011.

[115] K. Nohl and S. Munaut. GSM Sniffing. In *27th Chaos communication congress*, 2010.

[116] K. Nohl and C. Paget. Gsm: Srsly. In *26th Chaos Communication Congress*, 2009.

[117] K. Nohl, E. Tews, and R.-P. Weinmann. Cryptanalysis of the DECT Standard Cipher. In *Proceedings of the 17th International Conference on Fast Software Encryption*, FSE'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[118] P. Oechslin. Making a faster cryptanalytic time-memory trade-off. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer Berlin Heidelberg, 2003.

[119] T. Ohigashi and M. Morii. A practical message falsification attack on WPA. In *Procedings of Joint Workshop on Information Security*, 2009.

[120] Osmocom. Código fuente de Nokia DCT3-GSMTAP. `http://bb.osmocom.org/trac/wiki/dct3-gsmtap`, 2004.

[121] Osmocom. Código fuente de OpenBSC. http://openbsc.osmocom.org/trac/wiki/OpenBSC, 2008–2014.

[122] Osmocom. Código fuente de OsmocomBB. http://bb.osmocom.org/trac, 2011–2014.

[123] Osmocom. Código fuente de OsmocomSDR. http://sdr.osmocom.org/trac, 2011–2014.

[124] C. Paget. Practical cellphone spying. In *DefCon 18*, 2010.

[125] G. Paul, S. Rathi, and S. Maitra. On non-negligible bias of the first output byte of RC4 towards the first three bytes of the secret key. *Des. Codes Cryptography*, 49(1-3):123–134, Dec. 2008.

[126] D. Perez and J. Pico. A practical attack against GPRS / EDGE / UMTS / HSPA. In *BlackHat 2011 Lecture Notes*, 2011.

[127] S. Petrovic and A. Fúster-Sabater. Cryptanalysis of the A5/2 algorithm. *IACR Cryptology ePrint Archive*, 2000:52, 2000.

[128] T. Pornin and J. Stern. Software-hardware trade-offs: Application to A5/1 cryptanalysis. In *Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '00, pages 318–327, London, UK, UK, 2000. Springer-Verlag.

[129] A. Rager. Código fuente de WEPCrack. http://sourceforge.net/projects/wepcrack, 2002.

[130] B. Ramamurthi, K. Giridhar, and M. Srinivas. DSP-based digital FM demodulation for GMSK signals. *Sadhana*, 21(1):101–112, 1996.

[131] M. Rangelov. Código fuente de HashKill. https://github.com/gat3way/hashkill, 2011–2013.

[132] J. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely. Partitioning attacks: or how to rapidly clone some GSM cards. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 31–41, 2002.

[133] RenderLab. Church of WiFi WPA-PSK lookup tables. http://www.renderlab.net/projects/WPA-tables/. Accedido el 14-08-2014.

[134] A. Ross. A class of weak keys in the RC4 stream cipher. 2 envíos a sci.crypt, message-id 43u1eh$1j3@hermes.is.co.za and 44ebge$llf@hermes.is.co.za, 1995.

[135] M. Rossberg. Código fuente de KisMAC. http://kismac-ng.org, 2006–2011.

[136] I. Sanchez. Código fuente de Weplab. http://weplab.sourceforge.net, 2004–2006.

[137] I. Sanchez, G. Baldini, D. Shaw, and R. Giuliani. Experimental passive eavesdropping of Digital Enhanced Cordless Telecommunication voice communications through low-cost Software-Defined Radios. *Security and Communication Networks*, 2014.

[138] I. Sanchez, R. Satta, I. N. Fovino, G. Baldini, G. Steri, D. Shaw, and A. Ciardulli. Privacy leakages in smart home wireless technologies. In *Proceedings of the 48th IEEE International Carnahan Conference on Security Technology*, IEEE ICCST 2014, 2014.

[139] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 5:1–5:16, Berkeley, CA, USA, 2007. USENIX Association.

[140] A. Schuler, E. Tews, and R.-P. Weinmann. A hacker's view of DECT. In *25th Chaos Computer Congress*, 2008.

[141] P. Sepehrdad, S. Vaudenay, and M. Vuagnoux. Discovery and exploitation of new biases in RC4. In *Proceedings of the 17th International Conference on Selected Areas in Cryptography*, SAC'10, pages 74–91, Berlin, Heidelberg, 2011. Springer-Verlag.

[142] J. Shah and A. Mahalanobis. A new guess-and-determine attack on the A5/1 stream cipher. *arXiv preprint arXiv:1204.4535*, 2012.

[143] C. E. Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[144] T. N. Solutions. Código fuente de Reaver. `https://code.google.com/p/reaver-wps`, 2011.

[145] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, SSYM'01, Berkeley, CA, USA, 2001. USENIX Association.

[146] D. Spaar. Playing with the GSM RF interface. In *26th Chaos Communication Congress*, 2009.

[147] SRLabs. Código fuente de AirProbe. `http://gts.sourceforge.net`, 2010–2011.

[148] SRLabs. Código fuente de Kraken. `https://opensource.srlabs.de/projects/a51-decrypt`, 2012.

[149] A. Stubblefield, J. Ioannidis, and A. D. Rubin. A key recovery attack on the 802.11b Wired Equivalent Privacy protocol (WEP). *ACM Transactions on Information and System Security*, 7(2):319–332, May 2004.

[150] A. Stubblefield, A. Stubblefield, J. Ioannidis, J. Ioannidis, A. D. Rubin, and A. D. Rubin. Using the Fluhrer, Mantin, and Shamir attack to break WEP. In *Proceedings of the 10th USENIX Security Symposium*, 2001.

[151] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, SP '02, pages 19–, Washington, DC, USA, 2002. IEEE Computer Society.

[152] E. Tews and M. Beck. Practical attacks against WEP and WPA. In *Proceedings of the Second ACM Conference on Wireless Network Security*, WiSec '09, pages 79–86, New York, NY, USA, 2009. ACM.

[153] E. Tews, R.-P. Weinmann, and A. Pyshkin. Breaking 104 bit WEP in less than 60 seconds. In *Proceedings of the 8th International Conference on Information Security Applications*, WISA'07, pages 188–202, Berlin, Heidelberg, 2007. Springer-Verlag.

[154] F. van den Broek. Catching and understanding GSM-signals. Master's thesis, Radboud University Nijmegen, 2010.

[155] M. Vanhoef and F. Piessens. Practical verification of WPA-TKIP vulnerabilities. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 427–436, New York, NY, USA, 2013. ACM.

[156] S. Vaudenay and M. Vuagnoux. Passive-only key recovery attacks on RC4. In *Proceedings of the 14th International Conference on Selected Areas in Cryptography*, SAC'07, pages 344–359, Berlin, Heidelberg, 2007. Springer-Verlag.

[157] S. Viehböck. Brute forcing Wi-Fi Protected Setup. `http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf`, 2011. Accedido el 04-03-2014.

[158] D. Wagner. Weak keys in RC4. Newsgroup sci.crypt, 1995.

[159] A. Webb and K. Copsey. *Statistical Pattern Recognition*. Wiley, 2011.

[160] M. Weiner, E. Tews, B. Heinz, and J. Heyszl. FPGA implementation of an improved attack against the DECT Standard Cipher. In *Proceedings of the 13th International Conference on Information Security and Cryptology*, ICISC'10, pages 177–188, Berlin, Heidelberg, 2011. Springer-Verlag.

[161] H. Welte and S. Markgraf. OsmocomBB running your own GSM stack on a phone. In *27th Chaos Communication Congress*, 2010.

[162] H. Welte and D. Spaar. Running your own GSM network. In *25th Chaos communication congress*, 2008.

[163] S. Wray. COMP128: A birthday surprise. `http://www.stuartwray.net/comp128-a-birthday-surprise-rev.pdf`, 2003. Accedido el 15-07-2014.

[164] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 35–49, Washington, DC, USA, 2008. IEEE Computer Society.

[165] J. Wright. Código fuente de CoWPAtty. `http://wirelessdefence.org/Contents/coWPAttyMain.htm`, 2007.