UNIVERSIDAD CARLOS III DE MADRID

Ph.D. THESIS

Lightweight Cryptography in Radio Frequency Identification (RFID) Systems

Author:
Pedro Peris López

Supervisors:
Dr. D. Julio C. Hernández Castro
Dr. D. Arturo Ribagorda Garnacho

Computer Science Department

Leganés, October 2008

UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Criptografía Ligera en Dispositivos de Identificación por Radiofrecuencia - RFID

Autor:
Pedro Peris López

Directores:
Dr. D. Julio C. Hernández Castro
Dr. D. Arturo Ribagorda Garnacho

Departamento de Informática

Leganés, Octubre 2008

TESIS DOCTORAL

LIGHTWEIGHT CRYPTOGRAPHY IN
RADIO FREQUENCY IDENTIFICATION (RFID) SYSTEMS

Autor: Pedro Peris López

Directores: Dr. D. Julio C. Hernández Castro
Dr. D. Arturo Ribagorda Garnacho

Firma del Tribunal Calificador:

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de 2008

To my 7 iron

Acknowledgements

I am indebted to many people for the help I have received to complete this thesis.

Firstly, I would like to thank my PhD supervisors, Arturo y Julio. A special mention goes to Julio for believing in me from the start and for supporting me through the good and the bad times. I have learned great things these years, and working alongside you has been an extremely rewarding task and a real honour (even if we disagree at times). I hope I will continue to learn and work with you. Thank you, Julio.

Secondly, I should like to thank the rest of the research team: Benjamin, Almudena, Agustín, Anabel, Chema, Jorgue and Eduardo. I cannot leave out Esther, with whom I have shared so many moments since our first days in the lab. Thank you for your understanding and support, and for putting up with my music (Philip Glass, CocoRosie, Rufus Wainwright, etc.), which was not always easy. Nor can I forget Juan, who has supported me constantly and from whom I have learnt many things - I hope I will keep learning from you.

Finally, I wish to say thanks to my family. Firstly to my parents, who let me leave my small home town to come here to the capital to study. Thank you for your confidence in me and for supporting me always. Last but not least I should like to thank the two people who live with me and tolerate me! Thank you Mary (Bonnie) for always being there, encouraging and helping me. A million thanks to my brother (Diego - TxP), with whom I have shared so many of life's moments and who is always there when I need him (and much, much more).

Finally I should like to say thanks to my close friend Nacho, and to Paco for initiating me and being my teacher in the great art that is golf.

Abstract

This thesis examines the security issues of Radio Frequency Identification (RFID) technology, one of the most promising technologies in the field of ubiquitous computing. Indeed, RFID technology may well replace barcode technology. Although it offers many advantages over other identification systems, there are also associated security risks that are not easy to address.

RFID systems can be classified according to tag price, with distinction between high-cost and low-cost tags. Our research work focuses mainly on low-cost RFID tags. An initial study and analysis of the state of the art identifies the need for lightweight cryptographic solutions suitable for these very constrained devices. From a purely theoretical point of view, standard cryptographic solutions may be a correct approach. However, standard cryptographic primitives (hash functions, message authentication codes, block/stream ciphers, etc.) are quite demanding in terms of circuit size, power consumption and memory size, so they make costly solutions for low-cost RFID tags. Lightweight cryptography is therefore a pressing need.

First, we analyze the security of the EPC Class-1 Generation-2 standard, which is considered the universal standard for low-cost RFID tags. Secondly, we cryptanalyze two new proposals, showing their unsuccessful attempt to increase the security level of the specification without much further hardware demands. Thirdly, we propose a new protocol resistant to passive attacks and conforming to low-cost RFID tag requirements. In this protocol, costly computations are only performed by the reader, and security related computations in the tag are restricted to very simple operations. The protocol is inspired in the family of Ultralightweight Mutual Authentication Protocols (UMAP: M2AP, EMAP, LMAP) and the recently proposed SASI protocol. The thesis also includes the first published cryptanalysis of

SASI under the weakest attacker model, that is, a passive attacker. Fourthly, we propose a new protocol resistant to both passive and active attacks and suitable for moderate-cost RFID tags. We adapt Shieh et.'s protocol for smart cards, taking into account the unique features of RFID systems. Finally, because this protocol is based on the use of cryptographic primitives and standard cryptographic primitives are not supported, we address the design of lightweight cryptographic primitives. Specifically, we propose a lightweight hash function (Tav-128) and a lightweight Pseudo-Random Number Generator (LAMED and LAMED-EPC). We analyze their security level and performance, as well as their hardware requirements and show that both could be realistically implemented, even in low-cost RFID tags.

This document is submitted as PhD thesis in the Computer Science Programme 2007-8 at Carlos III University of Madrid.

Resumen

Esta tesis se centra en el estudio de la tecnología de identificación por radiofrecuencia (RFID), la cual puede ser considerada como una de las tecnologías más prometedoras dentro del área de la computación ubicua. La tecnología RFID podría ser el sustituto de los códigos de barras. Aunque la tecnología RFID ofrece numerosas ventajas frente a otros sistemas de identificación, su uso lleva asociados riesgos de seguridad, los cuales no son fáciles de resolver.

Los sistemas RFID pueden ser clasificados, atendiendo al coste de las etiquetas, distinguiendo principalmente entre etiquetas de alto coste y de bajo coste. Nuestra investigación se centra fundamentalmente en estas últimas. El estudio y análisis del estado del arte nos ha permitido identificar la necesidad de desarrollar soluciones criptográficas ligeras adecuadas para estos dispositivos limitados. El uso de soluciones criptográficas estándar supone una aproximación correcta desde un punto de vista puramente teórico. Sin embargo, primitivas criptográficas estándar (funciones resumen, código de autenticación de mensajes, cifradores de bloque/flujo, etc.) exceden las capacidades de las etiquetas de bajo coste. Por tanto, es necesario el uso de criptografía ligera.

A continuación, se resume la motivación de este trabajo, la metodología seguida, las principales aportaciones de esta tesis y, finalmente, se muestran las conclusiones más importantes.

0.1 Motivación

La tecnología RFID puede transformar los procesos de identificación, ofreciendo numerosas ventajas frente a otros sistemas. Actualmente los sistemas de identificación más extendidos son los códigos de barra. La prin-

cipal diferencia entre estas dos tecnologías radica en el nivel de identificación conseguido. Los códigos de barras permiten la identificación del tipo, mientras que las etiquetas RFID permiten una identificación inequívoca. En otras palabras, si se emplea la tecnología RFID se pueden distinguir dos items del mismo tipo.

Sin embargo, no todo son ventajas. El uso de la tecnología RFID trae asociado ciertos riesgos de seguridad. Privacidad y trazabilidad son los más importantes, pero existen otros que es necesario mencionar: ataques físicos, falsificación, denegación de servicio, etc. A su vez, el coste de las etiquetas está retrasando la introducción de esta tecnología. Cada vez que una nueva tecnología aparece (ej. bluetooth, wireless, etc.), las principales preocupaciones son el precio y la operatividad dejando la seguridad a un lado. Para evitar errores pasados, son necesarias soluciones criptográficas con el objetivo de proporcionar un nivel de seguridad adecuado. Este nivel de seguridad será diferente dependiendo de la clase del RFID. Es importante comprender que no todas las clases de RFID tienen porqué proporcionar el mismo nivel de seguridad.

Desde 2003, se han publicado numeros artículos centrados en la seguridad. La gran mayoría de estas propuestas no abordan de forma realista las fuertes limitaciones (ej. computacionales, circuitería, consumo, etc.) de este tipo de dispositivos. A pesar que desde un punto de vista teórico estas propuestas tienen sentido; en general no es posible su aplicabilidad en un gran número de etiquetas RFID de bajo coste. Están basadas en primitivas criptográficas, pero sin embargo no se sugieren primitivas criptográficas ligeras y las primitivas criptográficas estándar exceden las capacidades de las etiquetas. A su vez, a menudo, no se especifica para qué clase de etiqueta son adecuadas las propuestas. La clase de la etiqueta determina un gran número de parámetros tales como las operaciones soportadas en la misma o el tipo de ataques frente a los que debe ser resistente.

En 2003, Vajda et al. publicaron el primer artículo en el que se proponía el uso de criptografía ligera. El año siguiente, Juels introdujo el concepto de criptografía minimalista. En 2005, no hubo ninguna propuesta en este área de investigación, por el contrario hubo numerosas propuestas basadas en el uso de funciones resumen. Sin embargo, este área de investigación ha atraído cierto interés, recientemente, desde la publicación de varios protocolos ligeros por parte del autor de esta tesis. Uno de estos

primeros trabajos fue publicado en la conferencia RFIDSec, considerada el evento anual más importante sobre seguridad en dispositivos RFID. Desde la publicación de estos trabajos, algunos autores, incluido el autor de esta tesis, han intentado realizar avances en el desarrollo de la criptografía ligera para las etiquetas RFID de bajo coste.

0.2 Metodología y Planificación

El desarrollo de esta tesis se puede dividir en dos bloques fundamentales:

Tecnología RFID En primer lugar, fue necesaria una comprensión de la tecnología RFID. Un buen punto de partida fue la lectura del libro titulado: "RFID Handbook: fundamentals and applications in contactless smart-cards and identification". Al mismo tiempo, se consultaron otros libros y muchos artículos. Además, se estudiaron los estándares relacionados con esta tecnología con el fin de profundizar en su conocimiento. Tomando en consideración la tecnología RFID, consideramos oportuno centrar nuestra investigación en aspectos de seguridad para este tipo de dispositivos. Durante este periodo, consultamos, como un primer paso, el sitio web mantenido por G. Avoine (`http://lasecwww.epfl.ch/~gavoine/rfid/`) el cual reúne un gran número de artículos sobre privacidad y seguridad en sistemas RFID. La lectura de estos artículos y muchos otros nos han permitido conocer las ultimas novedades en este área de investigación. Puede decirse que familiarizarse con estas tres áreas (Tecnología RFID, Estándares RFID y Seguridad RFID) duraron aproximadamente un año y medio. Esta fase es realmente un proceso continuo, especialmente debido a que la tecnología RFID es un tema candente y se producen nuevos avances y desarrollos constantemente.

Avances en Criptografía Ligera Después de haber comprendido y analizado todas las propuestas publicadas, decidimos centrar nuestra investigación en las etiquetas de bajo coste. A partir de los estudios mencionados anteriormente, identificamos la absoluta necesidad de avanzar en el desarrollo de soluciones criptográficas ligeras. En primer lugar, realizamos un análisis de seguridad del estándar EPC Class-1 Generation-2. Consideramos importante este análisis debido a que

esta especificación puede ser considerada como el estándar universal para las etiquetas de bajo coste. El análisis anterior mostró importantes riesgos de seguridad. Algunos autores, siendo conscientes de ello, han propuesto nuevos esquemas mejorados bajo el marco de esta especificación. En esta tesis criptoanalizamos las dos propuestas más recientes en esta área de investigación, identificando importantes vulnerabilidades. En tercer lugar, diseñamos un protocolo resistente frente a ataques pasivos y adecuado para las etiquetas de bajo coste. Ya que el diseño de un protocolo nuevo no es una tarea fácil, fueron realizadas diferentes propuestas hasta que alcanzamos el protocolo finalmente propuesto en esta tesis. A continuación, nos planteamos el reto de diseñar un nuevo protocolo resistente frente a ataques tanto pasivos como activos. Este nuevo protocolo está inspirado en el protocolo para smart-cards diseñado por Shien et al., pero adaptado para los sistemas RFID. Como el protocolo anterior se basa en el uso de primitivas criptográficas, consideramos necesario el diseño de primitivas criptográficas ligeras. Esta es un area de investigación en el que apenas ha habido propuestas en el ámbito de las etiquetas de bajo coste. En concreto, proponemos dos nuevas primitivas, una función resumen y un generador de números pseudoaleatorios. Una vez diseñados, analizamos su seguridad en profundidad. A su vez, los requisitos hardware asociados fueron cuidadosamente examinados, comprobando siempre que estos no superaban las capacidades de las etiquetas de coste moderado. Este periodo de investigación duró alrededor de 2 años y medio.

La *Tabla 9* incluye la duración de cada actividad.

0.3 Principales Aportaciones

El resumen de las aportaciones novedosas de esta tesis pueden agruparse en tres puntos fundamentales:

Estado de la Cuestión Comenzamos nuestro trabajo con un exhaustivo estado del arte sobre la tecnología RFID. En primer lugar, en el *Capítulo 1* estudiamos los sistemas RFID, componentes y métodos de comunicación. En segundo lugar, en el *Capítulo 2* examinamos los princi-

Table 9: Duración de las Actividades

	Actividad	**Duración**
Parte 1	Tecnología RFID	4 meses
	A. Estándares RFID	3 meses
	B. Seguridad RFID	10 meses
Parte 2	A. EPC	3 meses
	B. Cryptoanálisis: EPC^{+}	6 meses
	C. Protocolo: etiquetas de bajo coste	6 meses
	D. Protocolo: etiquetas de coste moderado	6 meses
	E. Función Resumen Ligera	5 meses
	F. PRNG Ligero	5 meses

pales estándares relacionados con la tecnología RFID. En tercer lugar, un estudio de los principales ataques relacionados con los sistemas RFID se puede encontrar en el *Capítulo 3*. Por ultimo, en el *Capítulo 5* revisamos y examinamos las principales propuestas -incluyendo una extensa sección dedicada a criptografía ligera- cuyo objetivo es dotar de seguridad a la tecnología RFID. De este trabajo resultó la publicación de un artículo en la conferencia internacional "Personal Wireless Communications" [164] y dos capítulos publicados por Auerbach Publications en el libro titulado: "Security in RFID and Sensor Networks" [170, 171].

Estándar EPC-C1G2 En el *Capítulo 4*, realizamos un completo estudio y análisis del estándar EPC-C1G2 (ISO 18000-6C) debido a que éste puede ser considerado como el estándar universal para las etiquetas RFID de bajo coste. Parte de este trabajo fue publicado por Auerbach Publications como un capítulo en el libro titulado: "The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems" [169].

Criptografía Ligera: etiquetas RFID de no alto coste Para contribuir en el avance del área de investigación conocida como criptografía ligera para sistemas RFID proponemos las siguientes aportaciones relevantes incluidas en el *Capítulo 6*.

Criptoanálisis del EPC+ Algunos autores han propuestos nuevos esquemas -sin alterar significativamente el marco del estándar EPC-C1G2- para corregir el nivel de seguridad insatisfactorio de la

especificación original. Criptanalizamos las dos propuestas más recientes concluyendo que ambas propuestas poseen importantes fallos de seguridad. De esta investigación resultó la publicación de un artículo en la revista "Computer Standards & Interfaces" [168], y un segundo artículo en la conferencia internacional "RFIDSec" [172].

Protocolo Ultraligero Una parte importante de nuestra actividad investigadora se centró en el diseño de un protocolo de autenticación ultraligero resistente frente a ataques pasivos. Debido a que el diseño de una solución segura es un reto extremadamente difícil, fueron propuestos varios esquemas hasta que alcanzamos el protocolo propuesto en esta tesis. En 2006, propusimos una familia de protocolos ultraligeros (denominados genéricamente la familia de protocols UMAP para abreviar). La familia de protocolos UMAP fueron publicados en varias conferencias internacionales dentro de las cuales se incluye la prestigiosa conferencia internacional "RFIDSec" [161, 162, 163]. Posteriormente varios investigadores publicaron ataques activos y pasivos contra la familia de protocolos ultraligeros UMAP. Finalmente, Chien publicó el protocolo SASI fuertemente inspirado en la familia UMAP. Este protocolo da un paso más hacia el objetivo de diseñar un protocolo ultraligero seguro. Sin embargo, en esta tesis mostramos un criptanálisis del protocolo SASI bajo la débil suposición de un atacante pasivo. Este trabajo ha sido enviado a la revista "IEEE Transactions on Dependable and Secure Computing" (Mayo del 2008). Por ultimo, a modo de conclusión de toda esta investigación, proponemos un nuevo protocolo, llamado Gossamer e inspirado tanto en SASI como en la familia de protocolos UMAP.

Protocol Resistente a Ataques Activos Identificamos la necesidad de diseñar un protocolo seguro para las etiquetas de coste moderado. En esta clase de etiquetas, son aplicables tanto los ataques pasivos como los activos. En vez de empezar desde cero, proponemos un protocolo inspirado en el esquema para smart-cards diseñado por Shieh et al., pero adaptado para las etiquetas de coste moderado. De este trabajo resultó la publicación de un

artículo en la conferencia internacional "Security in Ubiquitous Computing (SecUbiq)" [165].

Primitivas Ligeras En el *Capítulo 5* realizamos un estudio completo sobre primitivas criptográficas ligeras. El análisis revela que mientras que ha habido un gran avance en el diseño de cifradores de bloque/flujo ligeros, queda mucho trabajo por hacer en el caso de funciones resumen ligeras y PRNG ligeros. Proponemos dos nuevas primitivas criptográficas con el objetivo de avanzar en este area de investigación. En primer lugar, debido a que el protocolo que proponemos para las etiquetas de coste moderado está basado en una función resumen, proponemos una novedosa función resumen ligera, llamada Tav-128. Esta primitiva ligera junto con el protocolo mencionado anteriormente fue publicado en la conferencia internacional "SecUbiq" [165]. En segundo lugar, nos centramos en los PRNG ligeros ya que su uso fue ratificado por el estándar EPC-C1G2. En esta especificación se exigen tres requisitos al PRNG pero no se propone ningún algoritmo capaz de satisfacerlos. A pesar de que existen diversos productos comerciales fabricados conformes al estándar, ningún algoritmo ha sido hecho público. Avanzando en esta dirección, proponemos dos novedosos PRNG ligeros (LAMED y LAMED-EPC). De esta investigación resultó la publicación de un artículo en la revista "Computer Standards & Interfaces" [166].

Por ultimo, nos gustaría hacer hincapié en el impacto en la comunidad investigadora de todos nuestros trabajos publicados y relacionados con esta tesis. Concretamente, estos artículos han sido citados en 64 ocasiones (Consultado en Junio del 2008: `http://scholar.google.es/scholar?hl=es&lr=&q=pedro+peris-lopez&btnG=Buscar&lr=`). Además, hemos creado una página Web (`http://www.lightweightcryptography.com/`) con el objetivo de difundir nuestra actividad investigadora lo más ampliamente posible.

0.4 Conclusiones

La tecnología RFID es una tecnología prometedora que podría revolucionar los sistemas de identificación. Actualmente, los códigos de barras son el sistema de identificación más extendido. Esta es una tecnología muy consolidada, que está retrasando la implantación de la tecnología RFID. Sin embargo, la tecnología RFID ofrece ventajas importantes sobre los códigos de barras. La principal ventaja es que los items etiquetados mediante tecnología RFID se pueden identificar de forma inequívoca. Otra característica importante es que no es necesaria la intervención humana. Como veremos en el *Capítulo 2*, los datos pueden ser leídos de forma automática, sin necesidad de visión directa, a una velocidad de cientos por segundo y desde una distancia de varios metros.

Uno de los obstáculos que está retrasando la penetración de los sistemas RFID es su coste. El precio de las etiquetas para su implantación masiva debe estar en el rango de 0,05 a 0,1 €. Para reducir el coste, los expertos creen que este podría ser repartido a lo largo de todo su ciclo de vida: producción, distribución, venta y reciclado. Actualmente, sin embargo todos los costes recaen en la fase de producción. Además, todo cambio tecnológico trae costes asociados a las empresas. La forma de abordar este problema no es la misma en todas las compañías. Algunas empresas, simplemente, esperan a ver que van a hacer sus rivales. Por el contrario, en compañías avanzadas, los lideres empresariales están estudiando el retorno de inversion (ROI) asociado con este cambio tecnológico. De hecho, la penetración de la tecnología RFID difiere significativamente de un país a otro. En España, las compañías están principalmente desarrollando proyectos pilotos para analizar el uso de esta tecnología (ej. Grupo Pascual, Grupo KH Lloreda, etc.).

0.4.1 Riesgos y Amenazas

Existe un gran desconocimiento acerca de los riesgos reales asociados con la tecnología RFID. La gran mayoría de los artículos encontrados en los medios de comunicación dan descripciones incorrectas y frecuentemente catastróficas de esta tecnología. Esto no significa que los sistemas RFID no puedan ser atacados. La tecnología RFID es una tecnología inalámbrica. Algunos de los ataques encontrados en esta tecnología son similares

a los que suceden en otras tales como wireless, bluetooth, etc. Además, las etiquetas RFID tienen similitudes con las smart-cards debido a las limitaciones del chip. Por lo tanto, no todos los ataques asociados con esta tecnología son nuevos.

Cuando se pregunta a la gente acerca de los riesgos de seguridad asociados con la tecnología RFID, la mayoría dice lo mismo: "privacidad". Se trata de una respuesta evidente debido a que la tecnología RFID es una tecnología pervasiva. Cuando nos referimos a privacidad, realmente consideramos dos conceptos: datos y localización. Sin embargo, estos no son las únicos problemas que deben ser tenidos en cuenta a la hora de diseñar un sistema RFID. Como los sistemas RFID están compuestos de tres componentes principales (etiqueta, lector y base de datos), cada componente debe ser analizado. En el *Capítulo 3*, realizamos un exhaustivo análisis de cada uno de los posibles riesgos.

Otro aspecto importante es el nivel de seguridad de los sistemas RFID. Desde un punto de vista teórico, es razonable que todos los sistemas sean resistentes frente a ataques activos y pasivos. Sin embargo, existen diferentes clases de etiquetas RFID y su área de aplicación no es la misma. De hecho, el nivel de seguridad de un sistema será un balance entre confidencialidad, integridad y disponibilidad.

0.4.2 Estándares y Soluciones Propuestas

La introducción de cualquier tecnología está asociada con el desarrollo de estándares. Inicialmente, las empresas usaban soluciones propietarias debido a la falta de armonización. Hoy en días las cosas están cambiando al menos para las etiquetas de bajo coste. Esto se debe en parte a la importante labor realizada por las organizaciones EPCGlobal e ISO. Este trabajo dio lugar al estándar EPC Class-1 Generation-2 ratificado por ambas organizaciones. El estándar puede ser considerado como el estándar universal para las etiquetas de bajo coste. En el *Capítulo 4*, analizamos en detalle la seguridad de este estándar, reflejando importantes fallos de seguridad. Motivados por este insatisfactorio nivel de seguridad, algunos investigadores han publicado ligeras modificaciones del estándar. Sin embargo, todos los esquemas fallaron en sus objetivos como veremos en los *Capítulos 4* y *6*.

Consideramos el estándar EPC-C1G2 un buen punto de partida para la construcción de un estándar seguro para las etiquetas de bajo coste. De

hecho, esperamos que el nivel de seguridad sea incrementado en la proxima generación (Gen-3). Algunos de los problemas de seguridad de este estándar son debidos a su diseño. En concreto, el estándar está centrado en el fabricante mientras que los usuarios no son considerados. En nuestra opinión, la operatividad del estándar es importante, pero los usuarios deberían ser tenidos en cuenta. Por ejemplo, el EPC no puede ser transmitido como texto en claro comprometiendo la privacidad del usuario.

Aparte del EPC-C1G2, existen otros estándares relacionados con esta tecnología, debido a su heterogeneidad. Estos pueden ser clasificados en cinco áreas principales (tarjetas de circuito integrado sin contacto, identificación de animales, gestión de items, campo cercano y EPC), como veremos en el *Capítulo 2*. Además, regulaciones regionales (ECC y ETSI en Europa) imponen restricciones adicionales a la implantación de estos sistemas. Estas restricciones están principalmente relacionadas con la planificación y uso del espectro radioeléctrico.

La tecnología RFID no es nueva, el primer artículo relacionado se publicó en los años cincuenta. Sin embargo, la investigación sobre esta tecnología y seguridad ha recibido una atención considerable desde el año 2003. En los años 2003 y 2004, se publicaron 10 y 30 artículos respectivamente, en esta área de investigación. Esta cifra aumentó a 75, 90, y 85, respectivamente en los tres siguientes años (`http://www.avoine.net/rfid/download/bib/bibliography-rfid.pdf`; consultada en Junio del 2008). De hecho, la seguridad en RFID es un tema de interés en un gran número de conferencias. Muchos de estos trabajos están centrados en seguridad, y la variedad de estas propuestas es muy diferente. Algunos autores han propuesto el uso de soluciones no criptográficas tales como jaulas de Faraday, declaración de derechos, etc. Otros autores han propuesto soluciones basadas en técnicas criptográficas. Estas soluciones son muy diversas, estando basadas algunas de estas en cifradores de bloque, generador de números pseudo-aleatorios, e incluso en criptografía asimétrica. Sin embargo, la solución más ampliamente usada está basada en el uso de una función resumen. Todos los protocolos anteriores comparten la particularidad común de ser protocolos de una sola ronda. En una aproximación completamente diferente tenemos la familia de human protocols que están basados en la ejecución iterada de una ronda muy simple. Además, sólo tres entidades (base de datos, lector y etiqueta) están involucradas

en todos los protocolos mencionados. Sin embargo, existen nuevas áreas de aplicación más allá de la autenticación mutua, y por ejemplo, algunos protocolos se centran en el problema de proporcionar una evidencia de la lectura simultánea de dos o más etiquetas.

0.4.3 Protocolos Ligeros

La mayoría de las propuestas que tratan de dotar de seguridad a las etiquetas RFID cometen dos errores. Primero, proponen un protocolo para las etiquetas RFID sin especificar para qué clase de etiquetas está destinado. Esto es un punto muy importante, ya que el número de recursos disponibles (memoria, circuito y consumo de energía, etc.) dependerá enormemente de esta decisión. Por tanto, no todas las etiquetas soportan el mismo tipo de operaciones. Por ejemplo, la criptografía de clave pública es aplicable para las etiquetas de alto coste pero excede las capacidades de las etiquetas de bajo coste. Además, cada clase de etiqueta tendrá un nivel de seguridad diferente. No es razonable que a las etiquetas de bajo coste (ej. paquete de galletas) se les exija el mismo nivel de seguridad que a las etiquetas de alto coste (ej. pasaporte electrónico). En segundo lugar, los protocolos propuestos no son realistas respecto a los recursos de las etiquetas. Como ya hemos mencionado anteriormente, la solución más ampliamente usada se basa en la utilización de una función resumen. A pesar de ello muchos autores afirman que sus protocolos son adecuados para las etiquetas de bajo coste. Sin embargo, un máximo de 4K puertas lógicas se pueden dedicar a funciones de seguridad en esta clase de etiquetas. Como veremos en el *Capítulo 6*, un gran número de recursos (más de 9K puertas lógicas) son necesarios para implementar funciones resumen criptográficas estándar. Por otra parte, no se proponen funciones resumen ligeras. Por tanto, la criptografía ligera es necesaria.

Con el fin de evitar errores del pasado, los requisitos y restricciones del sistema deben ser fijados inicialmente. En nuestro trabajo, hemos discriminado entre dos clases de etiquetas: etiquetas de bajo coste y etiquetas de coste moderado. Para cada clase de etiqueta se han especificado las siguientes características: fuente de alimentación, circuitería, distancia de lectura, precio, etc. En el *Capítulo 6* pueden verse los detalles. La principal diferencia entre estas dos clases de etiquetas es el número de puertas lógicas que pueden ser dedicados a tareas relacionas con la seguridad. Las

etiquetas de bajo coste únicamente usan operaciones eficientes (250K-4K puertas lógicas) y las etiquetas de coste moderado soportan primitivas criptográficas ligeras ($<$ 6K puertas lógicas). Otro punto interesante es el nivel de seguridad. Los ataques han sido divididos en activos y pasivos. Las etiquetas de bajo coste deben ser resistentes frente a ataques pasivos y las etiquetas de coste moderado frente a ataques pasivos y activos. Por ultimo mencionar que recientemente Chien propuso una clasificación alternativa de las etiquetas basada en las operaciones soportadas en las mismas (véase *Capítulo 6*).

0.4.4 Problemas Sociales

Incluso considerando que los problemas tecnológicos podrían eventualmente ser resueltos, la implantación de los sistemas RFID a gran escala no sería una realidad si no se educa al ciudadano acerca de sus beneficios y riesgos potenciales, y si no podemos garantizar un nivel de seguridad adecuado.

Gunter et al. realizaron un interesante estudio empírico acerca de la tecnología RFID y la percepción de control desde la perspectiva del consumidor [87]. La percepción de control pude ser vista como la creencia que el consumidor tiene en que el entorno electrónico actuará sólo en los casos explícitamente permitidos. Dos factores principales son considerados responsables de la percepción de pérdida de privacidad. En primer lugar, una cierta aprensión a "ser accedido", de no controlar la entrada en el entorno. Un atacante puede determinar el comportamiento personal y trazar los movimientos de los poseedores de las etiquetas sin ser detectado. En segundo lugar, "difusión, uso y mantenimiento de la información", esto es, el uso espurio de la gran cantidad de datos que pueden ser adquiridos de las personas. Dos tecnologías para aumentar la privacidad (PET) fueron propuestas a los usuarios. En el modelo orientado a usuario, los consumidores tienen un control total sobre el sistema (ej. mecanismo de autenticación). Por el contrario, el control de acceso es delegado a un agente (sistema de gestión de la identidad con protección de la identidad) en el modelo orientado a agente. Se analizó empíricamente cuál de estos dos mecanismos incrementaría la aceptación para el consumidor de la tecnología RFID. La *Figura 7.1* (véase *Capítulo 7*) muestra la percepción de control en cada aproximación. El estudio reflejó que los usuarios preferían (73.4%) desactivar

las etiquetas después de comprar un producto. Un 18% confiaban en PET (usuario o modelo de agente), y un 8.6% estaban indecisos.

En resumen, los avances tecnológicos deben reflejarse en la sociedad, y este aspecto no debe ser descuidado. Después de todo, los ciudadanos tienen la última palabra para decidir el futuro de una tecnología determinada.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objectives

This thesis examines cryptography for RFID systems. RFID security is an important research subject today, as evidenced by the great number of research papers published in the past two years (over 200). RFID technology has similarities with other technologies (wireless, bluetooth), but it also has unique features that must be taken into account when designing protocols.

Some papers do refer to attacks on RFID systems, but so far there has not been a rigorous study on the subject. Understanding attacks is the first step towards creating mechanisms to secure the technology. More thorough study is therefore necessary. The extent and impact of an attack can vary considerably; some focus on a particular component of the system (eg. the tag) whereas others target the whole system. As RFID technology is a pervasive technology, privacy is one of main problems to be solved. Privacy concerns both information and location. Although these are the risks most often referred to in the literature, there are other equally important problems to address. To aid the reader in his comprehension of the matter, the threats are grouped according to the unit involved. First we examine threats related to tags and readers such as eavesdropping, cloning, replay and relay attacks. Then we look at the threats to the back-end database (eg. ONS attack, virus).

Due to the diversity of RFID technology, there are a great number of stan-

dards in existence. The EPC Class-1 Generation-2 standard (the EPC-C1G2 standard for short) is one of the most significant. The computational and storing restrictions of Class-1 RFID tags are very severe, which rules out the use of standard cryptographic solutions. Some experts consider this standard as the universal for low-cost RFID tags. Since its publication, various researchers have pointed out its security risks, but a rigorous security analysis still awaits. This necessity has provided the motivation for our analysis, which concludes that security is very deficient. We would hope that the results of our analysis be taken into account in future versions (Generation-3) of the specification, which we believe must be able to provide greater security. In fact, it is incomprehensible that user information and location privacy should not be guaranteed.

Several authors have proposed solutions intending to provide satisfactory security within the framework of the EPC-C1G2 standard. In 2007, Chien et al. proposed a mutual authentication protocol advancing on Duc et al.'s scheme which had proved vulnerable to attacks such as DoS attacks against both tags and readers. Additionally, disguise tags are not detected and forward secrecy is not guaranteed. These protocols assume that tags support on-chip a cyclic redundancy code and a pseudo-random generation function. The same year, Konidola et al. presented a tag-reader mutual authentication scheme at the prestigious RFID security conference (RFIDSec'07). The scheme can be considered an improved version of an earlier proposal which was insecure even under a passive attack. In the scheme, a PadGen chain is introduced to protect access and kill passwords. We have cryptanalyzed these two proposals and are able to point out important security flaws which are difficult to solve. Consequently, all protocols proposed within the framework of the EPC-C1G2 specification (that are known to us) fall short of the security objectives.

Cryptography seems inevitable as a way to make RFID technology secure. From a theoretical point of view, standard cryptography may be a correct approach. However, it demands resources far in excess of those available on many tags. Low-cost RFID tags are very constrained devices, having severe storage, circuitry and power consumption limitations. Therefore, lightweight cryptographic techniques appear to be the most appropriate solution for non-high-cost RFID tags.

Many research articles do not specify the class of tag for which the proposed protocol is appropriate. Resources vary with the class of tag -in other words, tag class determines the kind of operations that can be supported. This thesis focuses on non-high-cost RFID tags with a distinction between low-cost and moderate-cost tags. We examine both kinds and identify their requirements and restrictions.

Having established the requirements, one protocol is proposed for each class of tag. Prior to defining the protocol, an adversary model is identified. The protocol is then defined, being broken down into its different stages. A security and performance analysis follows. It should be noted that there are greater security demands on moderate-cost tags than on low-cost RFID tags. Indeed, moderate-cost tags must resist both passive and active attacks whereas low-cost tags need only resist passive attacks.

The security of the proposed protocol, which corresponds to the requirements of moderate-cost tags, is based on a secure one-way hash function. However, the resources demanded by standard cryptographic primitives far exceed those of moderate-cost tags. To date, no lightweight hash function has been proposed that is suitable for RFID tags. So research must concentrate efforts on designing lightweight primitives. With the aim of advancing this research area, a new lightweight hash function, named Tav-128, is therefore proposed. Its design is an attempt to avoid the errors occurring in certain primitives proposed in the past (such as the use of a very linear expansion algorithm). A security analysis, examining its output over a low entropy input, is then performed. Finally, careful consideration is given to the resources required for its implementation.

Additionally, the EPC-C1G2 standard ratified the use of PRNGs for low-cost RFID tags. PRNGs conforming to this specification should meet three conditions of randomness. These are clearly specified, but no algorithm was proposed. Although there are some commercial products conforming to the EPC-C1G2 specification, the algorithms of the supported PRNGs have not been made public. So the design of public algorithms would be desirable. As a step in the right direction, we propose a new lightweight PRNG conforming to the standard. Indeed, two PRNGs have been designed: LAMED (32-bits) and LAMED-EPC (16-bits). Genetic programming was employed in their construction. Once presented, an in-depth

security analysis is made, which also looks at LAMED-EPC's compliance with the specification. Finally, an architectural design is proposed, which gives an overestimation of the number of logic gates necessary for implementation.

1.2 Motivation

RFID is one of the most promising technologies in the field of ubiquitous computing. Indeed, it may well transform identification processes. The technology offers many advantages over other identification systems. At the moment, the dominant identification system is the barcode. The main difference between the two technologies is that barcodes only allow type identification whereas RFID tags allow unequivocal identification. This means that two items of the same type can be distinguished if RFID technology is employed.

However, RFID technology has its disadvantages too. There are associated security risks. Privacy and tracking are the principal issues, but there are others that need to be considered too: physical attacks, counterfeiting, denial of service, etc. Additionally, the cost of RFID tags is an obstacle to technological advance. When a new technology appears (eg. bluetooth, wireless technology), the main concerns are cost and efficiency rather than security. We believe that past errors should be avoided. Cryptographic solutions are therefore required to provide adequate levels of security. Depending on the RFID class, the security level will vary. It is important to realise that not all tag classes need to guarantee the same security level.

Since 2003, there have been a great number of research publications focusing on security. The vast majority of these do not realistically address the severe limitations of these devices (storage, circuitry, power consumption, etc.). Although from a theoretical point of view the proposals make sense, their application in a great number of low-cost tags is not possible. They are based on cryptographic primitives, but lightweight cryptographic primitives are not discussed and standard cryptographic primitives exceed tag capabilities. Additionally, it is often not specified for which class of tag the proposals are suitable.

In 2003, Vajda et al. published the first article proposing the use of lightweight cryptography. The following year, Juels introduced the concept of minimalist cryptography. In 2005, there was no proposal in this area, the majority of proposals being based on the use of hash function. Nevertheless, this research has attracted some interest recently when certain lightweight protocols were proposed by the author of this thesis, in 2006. Of these protocols, one was presented in the RFIDSec conference, considered the most important event in the field of RFID security. Since the publication of these works, certain authors including the author of this thesis have tried to advance the development of lightweight cryptography for non-high-cost RFID tags.

1.3 Methods and Schedule

The development of the thesis is structured in two main blocks:

RFID Technology First, an understanding of RFID technology was necessary. A good starting point was reading "RFID Handbook: fundamentals and applications in contactless smart-cards and identification". Other books and many articles were also read in the same period. Knowledge of the technology was further extended by studying related standards. We considered it appropriate to focus our investigation of RFID technology on the security aspects of these RFID devices. During this same period, our first step was to consult the website maintained by Gildas Avoine (`http://lasecwww.epfl.ch/~gavoine/rfid/`) which gathers a great number of articles about security and privacy in RFID systems. Reading these and many other articles provided us with the very latest news in this research area. Research activity in the three areas (RFID technology, standards and security) could be said to have taken about a year and a half, although in reality the phase is an ongoing one, because RFID technology currently attracts a great deal of attention and new advances happen by the month.

Advances in Lightweight Cryptography Having understood and analyzed all the published proposals, we decided to focus investiga-

tion on non-high-cost RFID tags. The studies mentioned previously highlighted the need for advance in the development of lightweight cryptographic solutions. First, a security analysis of the EPC Class-1 Generation-2 standard was performed. We consider the analysis important because this specification can be considered as the universal standard for low-cost RFID tags. The analysis revealed important security risks. Several authors, being aware of these, proposed improved schemes under the framework of the specification. We cryptanalyzed the two most recent proposals in this field and identified important vulnerabilities. Thirdly, we designed a new protocol resistant to passive attacks and suitable for low-cost RFID tags. As the design of a new protocol is not an easy task, different proposals were made until we arrived at the proposed protocol in this thesis. We then went on to consider the great challenge of designing a new protocol resistant to both passive and active attacks. This new protocol is inspired in Shieh et al.'s protocol for smart cards, but adapted to RFID systems. As this protocol is based on cryptographic primitives, we considered the design of lightweight cryptographic primitives to be an imperative. This is a research area where hardly any proposals have considered non-high cost RFID tags. Specifically, we proposed two new lightweight primitives (a hash function and a pseudorandom number generator). After their design, a thorough security analysis was performed. Hardware complexity was also examined, to check that the resources required did not exceed the capabilities of moderate-cost RFID tags. This period of research lasted just over two years and a half.

To conclude, *Table 1.1* shows the duration of each activity.

1.4 Organization

The remainder of the thesis is organized as follows:

In *Chapter 2*, RFID systems are introduced. First, an overview of RFID components (tags, readers, back-end database) is given. Secondly, communication methods and interfaces are described. Then standards related to RFID

Table 1.1: Activity Duration

	Intended Activity	**Duration**
Part 1	A. RFID technology	4 months
	B. Standards	3 months
	C. RFID Security	10 months
Part 2	A. EPC	3 months
	B. Cryptanalysis: EPC^{+}	6 months
	C. Protocol: low-cost RFID tags	6 months
	D. Protocol: moderate-cost RFID tags	6 months
	E. Lightweight hash function	5 months
	F. Lightweight PRNG	5 months

technology are outlined. To clarify the explanation, they have been classified in five main groups: contactless integrated circuit cards, animals, item management, near field communication, and electronic product code.

Chapter 3 describes in detail the kind of attacks that RFID systems can suffer. First, the main privacy issues (privacy and tracking) are examined. Then attacks related to tags and readers are analyzed. Finally, attacks connected with back-end databases are studied.

In *Chapter 4*, the EPC Class-1 Generation-2 (IS0 18000-6C) standard is analyzed in depth. First, the main differences between Generation-2 and Generation-1 tags are described. Secondly, the main concepts of the standard are explained. Then a detailed description of the messages exchanged in the different operations is given. With understanding of the procedures now gained, a security analysis is carried out. Current proposals to improve upon the security of the standard are also considered, and their security aspects examined.

Chapter 5 introduces the main proposals to date for solutions to the security problems discussed in *Chapter 3*. These are divided into different groups (classical cryptography, hash function based schemes etc.) to facilitate the reader's understanding. Note that a whole section with the main proposals for lightweight cryptography appears at the end of the chapter. The basic principles and a critical review of each proposal are included. Although the full details are not given, readers may consult the bibliography entries for a more in-depth consideration of the subject.

Chapter 6 looks at solutions based on lightweight cryptography. First, the

necessity of lightweight cryptography for non-high-cost RFID tags is explained. Secondly, we cryptanalyze two novel authentication protocols under the EPC-C1G2 specification. Then an ultralightweight protocol resistant to passive attacks and conforming to low-cost RFID tags requirements is proposed. The cryptanalysis of the recently proposed and innovative ultralightweight protocol SASI is also incorporated in this section. Finally, a new protocol resistant to both passive and active attacks, and conforming to moderate-cost RFID tag restrictions, is presented. Since this last protocol is based on the use of a hash function and a PRNG, two new lightweight primitives suitable to moderate-cost RFID tags are also put forward.

Chapter 7 summarizes the conclusions that have been drawn previously in the corresponding chapters of the thesis. The concept of social problems is then introduced. Finally, there is an extensive bibliography for the reader to study the subject in depth.

1.5 Evaluation Methods

In the early part of the research period dealing with lightweight cryptography (*2.A* and *2.B*), the security of the EPC Class-1 Generation-2 specification was examined. First, the specification was considered in detail: physical layer, tag identification layer, tag memory, tag states and slot counter, etc. Then the messages exchanged in the different operations (select, inventory, access) were studied. Once the procedures were understood, their security vulnerabilities were identified. Additionally, some researchers have tried to enhance the security level of this standard. After studying these protocols (EPC+ for short), their security was assessed, particularly their resistance to standard attacks such as privacy, tracking etc.

In the subsequent part of the research period dealing with lightweight cryptography (*2.C* and *2.D*), two new lightweight protocols for non-high-cost RFID tags were designed. These were specifically suited to low-cost and moderate-cost RFID tags respectively. First, to define the protocol, the requirements of each tag class were identified. These requirements were then all evaluated after design of the protocol. Since designing a new protocol is not an easy task, different versions of the protocols were created until a definitive version could be identified. The protocols were submit-

ted for publication to recognized peer-reviewed conferences or journals, a process that allows the exchange of new ideas and allows potential vulnerabilities to be detected by the research community. Most importantly, it encourages advance in the creation of lightweight cryptography for RFID systems.

As mentioned previously, protocol design is a complex and challenging task. An in-depth security analysis must be carried out on completion of the design. In our case, the proposed protocol was analyzed for its robustness to the following attack types:

- ✓ User Privacy
- ✓ Location Privacy
- ✓ Data Integrity
- ✓ Mutual Authentication
- ✓ Forward Security
- ✓ Replay Attack
- ✓ Forgery Resistance
- ✓ Data Recovery
- ✓ Active Attacks
- ✓ Etc

Once the security analysis was complete, the performance of the protocol was studied. This was an important process because it included assessment of compliance with tag class requirements. Computational, storage, and communication overhead are some of the parameters included in the analysis. Finally, the protocol was compared with the main proposals found in the research literature.

Finally, two new primitives (a hash function and a PRNG) were proposed (*2.E* and *2.F*). Designing these primitives without exceeding the capabilities of constrained tags (moderate-cost) is a challenge. We endeavoured to avoid errors encountered in previous primitive designs. Once the design of the primitive was completed, a security analysis was carried out. The statistical properties of the outputs generated by the primitive were analyzed using four well-known stringent randomness test suites: ENT [216], DIEHARD [149], NIST [205], and David Sexton's battery [6]. The conformity of the PRNG primitive with the randomness requirements established

in the EPC-C1G2 standard was also scrutinized. Finally, these new primitives were submitted for publication to the recognized conferences and journals for analysis by the research community.

The implementation of these functions was another important issue. Functions should be as efficient as possible from a hardware perspective. An architectural design was proposed for each lightweight primitive. An overestimation of the gate count and throughput was obtained. Finally, the primitive was compared with other relevant proposals.

1.6 Main Contributions

The summary of the novel contributions of this thesis can be stated in three main points as:

State of the Art We started out with a comprehensive survey of the state of the art concerning with RFID technology: RFID systems, components and communications methods in *Chapter 1*; the main standards related to RFID technology in *Chapter 2*; and a comprehensive study of the main attacks on RFID systems in *Chapter 3*. Finally, in *Chapter 5* we listed and analyzed the main proposals (including an extensive section on lightweight cryptography) for making RFID technology secure. This work led to the publication of a paper for the International Personal Wireless Communications Conference [164] and of two book chapters in *Security in RFID and Sensor Networks* [170, 171] published by Auerbach Publications.

EPC-C1G2 Standard As the EPC-C1G2 (ISO 18000-6C) is regarded as the universal standard for low-cost RFID tags, a comprehensive study and analysis was completed (*Chapter 4*). Part of this is going to appear as a book chapter in *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems* [169] to be published by Auerbach Publications.

Lightweight Cryptography for Low-cost RFID Tags We proposed relevant contributions advancing research into lightweight cryptography for RFID systems (see *Chapter 6*), as follows:

Cryptanalysis of EPC$^+$ Without significantly altering the framework of the EPC-C1G2 standard, some authors have proposed new schemes to correct the unsatisfactory level of security in the specification. We cryptanalyzed the two most recent EPC+ proposals and concluded that both show significant security flaws. This investigation resulted in the publication of two articles: in the Computer Standards & Interfaces Journal [168] and in the International Conference on RFID Security [172].

An Ultralightweight Protocol An important part of our research activity was centered on the design of an ultralightweight protocol resistant to passive attacks. Since designing a secure solution is a great challenge, several schemes were proposed until the protocol proposed in this thesis (Gossamer protocol) was created. In 2006, we proposed a family of ultra-lightweight protocols (the UMAP protocols for short). The UMAP protocols were published in several international conferences including the International Conference on RFID security [161, 162, 163]. Several researchers published passive and active attacks against the UMAP family of protocols. Finally, Chien et al. published the protocol SASI which is a step towards designing a secure ultralightweight protocol. However, we have shown in this thesis a cryptanalysis of SASI under the weak assumption of a passive attacker. This work was submitted to IEEE Transactions on Dependable and Secure Computing (May 2008). The fruit of all this research is a new protocol called Gossamer, inspired in SASI and in the UMAP family of protocols, and presented in this thesis.

A Protocol Resistant to Active Attacks We identify the necessity of a secure protocol design for moderate-cost RFID tags. Both passive and active attacks apply to this class of tag. Rather than starting from scratch, we proposed a protocol inspired by Shieh et al.'s protocol for smart cards but adapted for moderate-cost RFID tags. This work led to a publication in the International Conference on Security in Ubiquitous Computing (SecUbiq) [165].

Lightweight Primitives In *Chapter 5*, a thorough study of lightweight cryptographic primitives was made. The anal-

ysis reveals that whilst great strides have been made in lightweight stream/block cipher design, much work remains to be done in the case of lightweight hash functions. We proposed two new lightweight primitives to advance research in this area. First, since the protocol for moderate cost RFID tags mentioned earlier is based on a hash function, a novel lightweight hash function, Tav-128, was proposed. This lightweight primitive was published together with the protocol at the SecUbiq conference [165]. Secondly, we looked at lightweight PRNGs, as their use for low-cost RFID tags is ratified by the EPC-C1G2 standard. In this specification three conditions are required for the PRNG, yet no algorithms have been proposed. So far, no public algorithm has been published despite the existence of several commercial products. To this end, two novel lightweight PRNGs (LAMED and LAMED-EPC) were proposed. This led to the publication of an article in the Computer Standards & Interfaces Journal [166].

Finally, we would like to highlight the impact that our work has made on the research community in connection with this thesis. Our articles have been cited 64 times (Consulted in June 2008: `http://scholar.google.es/scholar?hl=es&lr=&q=pedro+peris-lopez&btnG=Buscar&lr=`). Additionally, a Web site (`http://www.lightweightcryptography.com/`) was created to spread our research activity as widely as possible.

Chapter 2

RFID Systems

2.1 Introduction

At the moment, the most extended identification systems are barcodes (*Figure 2.1*). Initially, there were two standards: the Universal Product Code (UPC, United States) and the European Article Number (EAN, Europe). Although at first EAN was only taken on by twelve European countries, by the end of 2004 more than one hundred countries all over the world had already adopted this standard. Finally, when the United States decided to adopt the European-born standard, UPC and EAN merged, giving rise to what is nowadays known as GS1 [60].

Recently, there has been mass deployment of Radio Frequency Identification systems (RFID) (*Figure 2.2*). These systems comprise Radio Frequency (RF) tags or transponders, and RF readers or transceivers. Tag readers broadcast an RF signal to access resistant data stored in tags. One of the

Figure 2.1: Barcode

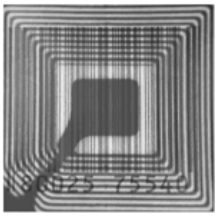


Figure 2.2: RFID Tag

main differences between tags and barcodes is that RFID tags provide an unique identifier, or a pseudonym that allows access to this unique identifier. The use of RFID tags offers several advantages over barcodes; data can be read automatically, without line of sight and through a non-conducting material such as cardboard or paper, at a rate of hundreds of times per second, and at a distance of several meters.

RFID systems are becoming valuable tools in processes such as manufacturing, provision chain management and stock control. Around 5 billion barcodes are read daily, so efficiency gains from using RFID tags could lower the cost of tagged items substantially [195]. The penetration of RFID systems is nowadays mainly limited by privacy concerns and by their cost, which must be between 0.05 and 0.1 € to be considered economically viable. Additionally, in order to take full advantage of the potential offered by RFID tags, the identification of an item must be maintained throughout its life cycle: production, distribution, sale and recycling (*Figure 2.3*).

Figure 2.3: Life Cycle of an Object

The low cost demanded for RFID tags causes them to be very resource limited. Typically, they can only store hundreds of bits, have roughly between 5K and 10K logic gates, and a maximum communication range of a few meters. Within this gate counting, only between 250 and 4K gates can be devoted to security functions. It is interesting to recall that for a standard implementation of the Advanced Encryption Standard (AES) between 20K and 30K gates are needed. Additionally, power restrictions should be taken into account, since most RFID tags in use are passive. Nor can these systems be expected to store passwords securely, because tags are not at all resistant to tampering attacks.

In spite of all these limitations, the penetration of RFID technology is increasing steadily. Experts believe that both systems will coexist for some time and that finally, RFID tags will completely replace classical barcodes. Some developments and current uses of this technology are:

- The US Department of Defense and Wal-Mart require all their major suppliers to use RFID technology in their supply chains [18].
- Delta Airlines is testing RFID for luggage control [20].
- Michelin is planning to build RFID tags into its tyres [17].
- The European Central Bank wants to attach a tag into 500 € bank notes [112].
- The Vatican's library has installed RFID technology for tracking books [19].

However, the implantation of RFID systems is not all smooth sailing, as certain organizations like CASPIAN [41] and FOEBUD [75] are strongly against their mass use.

2.2 Overview of RFID Systems

2.2.1 RFID System Components

RFID systems are made up of three main components, briefly described bellow: the transponder or RFID tag, the transceiver or RFID reader, and

the back-end database.

Transponder or RFID Tag In an RFID system, each object will be labeled with a tag. Each tag contains a microchip with some computation and storage capabilities, and a coupling element, such as an antenna coil for communication. Tags can be classified according to two main criteria:

1. **The type of memory**: The memory element serves as writable and non-writable data storage. Tags can be programmed to be read-only, write-once read-many, or fully rewritable. Depending on the kind of tag, tag programming can take place at the manufacturing level or at the application level.
2. **The source of power:** A tag can obtain power from the signal received from the reader, or it can have its own internal source of power. The way the tag gets its power generally defines the category of the tag.

   Passive RFID tags Passive tags do not have an internal source of power. They harvest their power from the reader that sends out electromagnetic waves. They are restricted in their read/write range as they rely on RF electromagnetic energy from the reader for both power and communication.

   Semi-passive RFID tags Semi-passive tags use a battery to run the microchip's circuitry but communicate by harvesting power from the reader signal.

   Active RFID tags Active tags possess a power source that is used to run the microchip's circuitry and to broadcast a signal to the reader.

Transceiver or RFID Reader RFID readers are generally composed of an RF module, a control unit, and a coupling element to interrogate electronic tags via RF communication. Readers may have better internal storage and processing capabilities, and frequently connect to back-end databases. Complex computations, such as all kind of cryptographic operations, may be carried out by RFID readers, as they do not usually have more limitations than those found in modern hand-

Table 2.1: Communication Methods

Passive	Passive backscatter or inductive coupling
Semi-passive	Passive backscatter or inductive coupling
Active	Transmits and receives RF signal

held devices or PDAs. Under this assumption, complexity is transferred out of the restricted tags.

Back-end Database The information provided by tags is usually an index to a back-end database (pointers, randomized IDs, etc.). This limits the information stored in tags to only a few bits, which is a sensible choice due to severe tag limitations in processing and storing. It is generally assumed that the connection between readers and back-end databases is secure, because processing and storing constraints are not so tight in readers, and common solutions such as SSL/TLS can be used.

2.2.2 Passive Communications Methods

The communication between a passive tag and a reader consists of energy transfer as well as data transfer. Energy is transferred using coupling via electromagnetic fields [82]. An electromagnetic field, as the name implies, has an electrical component and a magnetic component. RFID tags uses either the electric field or the magnetic field or both to receive energy from the reader. There are various methods of transferring the data to a reader, but passive tags usually use backscatter or inductive coupling. *Table 2.1* summarizes the different communication methods for different tag types.

Passive Backscatter Passive tags communicate with a reader using passive backscatter, also called modulated backscatter (see *Figure 2.4*). The reader transmits a continuous wave RF signal into the reading environment. When a tag appears in the area, it receives the reader signal and demodulates it (or breaks it up) into patterns of ones and zeros. This data is used as commands to inform the tag what operation to perform. By detuning and tuning its antenna very rapidly, the tag

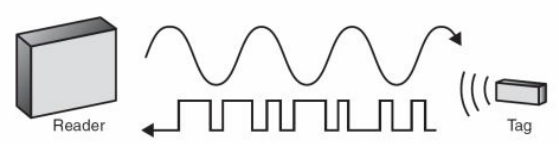


Figure 2.4: Passive Backscatter [230]

modulates the signal and reflects it, in a pattern of ones and zeros, back to the reader.

Inductive Coupling (Magnetic coupling) The electrical current flow through a conductor generates a magnetic field around the conductor. However, there is a further point to this phenomenon. When a conductor is exposed to a magnetic field, the magnetic field produces a current flow in the conductor. This is known as inductive coupling, because a current is generated by the influence of the magnetic field.

This communication process is used by Low Frequency (LF) and High Frequency (HF) band RFID devices. The RFID reader's antenna uses current to generate the magnetic field. The antenna on the RFID tag, when exposed to the magnetic field generated by the reader's antenna, generates a current in the tag that powers the tag circuitry. Circuitry on the RFID tag switches the impedance load of the tag's antenna, according to the data stream, causing modulation of the magnetic field joining the reader and tag. The modulation is demodulated by circuitry in the RFID reader, and the data is transmitted to the user (see *Figure 2.5*).

Electromagnetic Coupling Ultra High Frequency and Microwave tags usually use electromagnetic coupling. They can utilize the electric field as well as the magnetic field for energy. Because of the nature of the magnetic field, it can be utilized only at short distances from the source. That is why LF and HF tags, which mainly use magnetic fields, have short read ranges. Sometimes, where UHF and microwave tags are in near proximity to the radiating source, they can also use the magnetic field if the antenna design allows it. For longer read ranges, the magnetic fields get weaker and the electric field is used to resonate the tag's antenna in a specific frequency band.

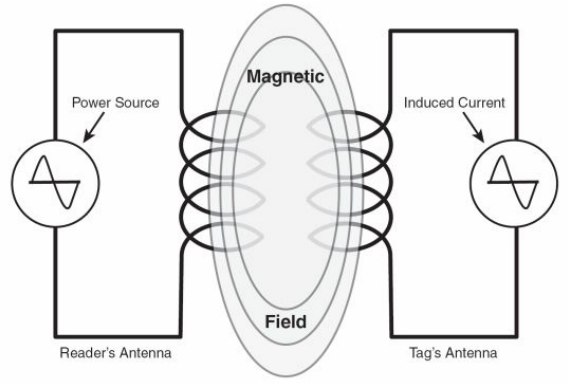


Figure 2.5: Inductive Coupling [230]

2.2.3 RFID System Interface

In this section, we focus exclusively on passive RFID tags, since we consider that these will be the first to be massively deployed and form part of our daily lives. Additionally, these low-cost RFID systems are very limited on resources, which forces some interesting trade-offs in their designs.

Transceiver/Transponder Coupling Communication Passive RFID tags obtain their operating power by harvesting energy from the RF signal of the reader by means of passive backscatter or inductive coupling. They have a reading distance of up to 3.3m.

The signal sent from readers to tags must be used simultaneously to transmit both information and energy. However, readers normally operate in Industrial Scientific-Medical (ISM) bands, so there are restrictions in the bandwidth and in the transmitted power. Tags, on the other hand, are not subject to these limitations.

Data Coding The exchange of data between the reader and the tag, and vice versa, must be produced efficiently; so both coding and modulation are used. The coding/modulation is defined according to the existing limitations in the backward and the forward channel. Readers will be able to transmit greater power, but will have bandwidth limitations. Tags, which are passive, will not have bandwidth limitations.

Table 2.2: Coding Techniques

Channel	Usual Coding
Forward Channel	Manchester or NRZ
Backward Channel	PPM or PWM

As a coding mechanism, level codes (Non-Return-to-Zero, NRZ; and Return to Zero, RZ) or transition codes (Pulse Pause Modulation, PPM; Pulse Weight Modulation, PWM; and Manchester) are mostly used. These coding techniques are shown in *Table 2.2*.

Modulation The modulation scheme determines how the bitstream is transmitted between readers and tags, and vice versa. Three possible solutions exist: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK). The choice of modulation type is based on power consumption, reliability and bandwidth requirements.

Tag Anti-collision Collisions in RFID systems happen when multiple tags simultaneously answer a reader signal. Methods used to solve this kind of problem, allowing reliable communication between readers and tags, are referred to as anti-collision methods. The anti-collision algorithms used in RFID systems are quite similar to those applied in networks, but they take into account that RFID tags are generally more limited than the average network device. Two approaches are used: probabilistic or deterministic. In practice, however, a combination of both is used to solve the problem.

Reader Anti-collision In this case, several readers interrogate the same tag at the same time. This is known in the bibliography as the *Reader Collision Problem*. One possible solution to this problem consists of allocating frequencies over time to a set of readers by either a distributed or a centralized approach.

Frequencies and Regulations Most RFID systems operate in ISM bands [102]. ISM bands are designated by the International Union of Telecommunications and are freely available for use by low-power, short-range systems. The most commonly used ISM frequencies for

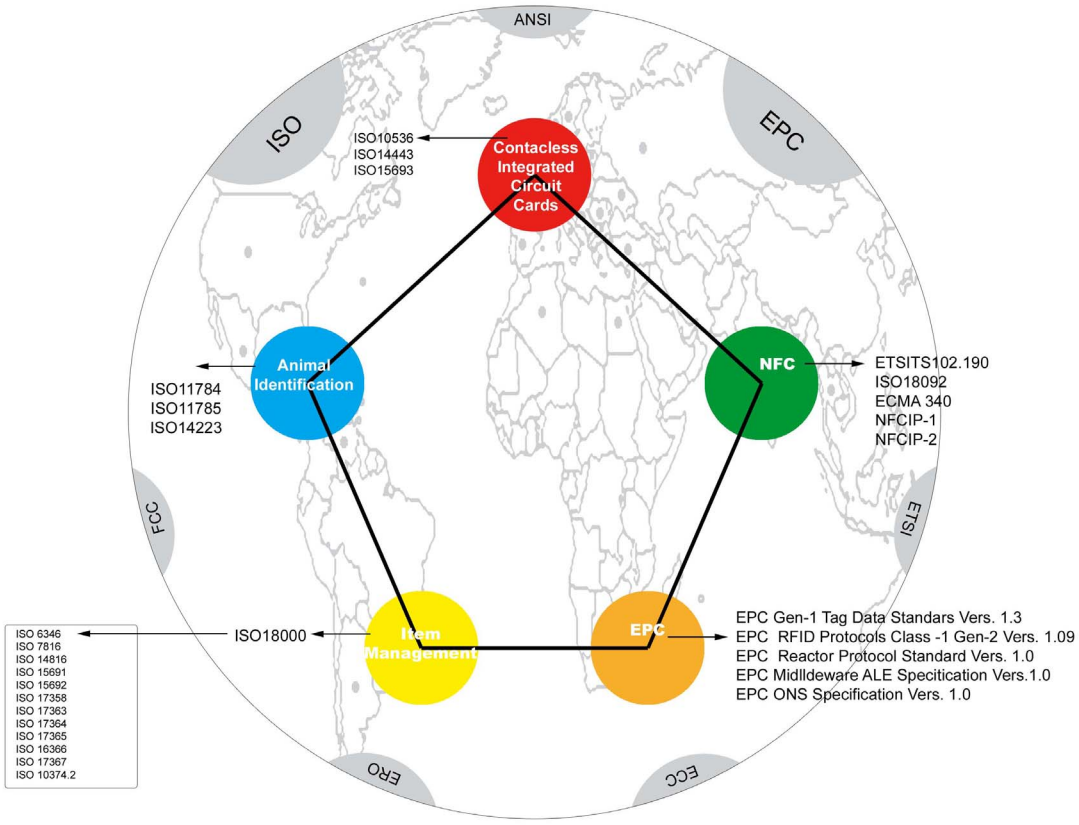


Figure 2.6: RFID Standards

RFID systems are 13.56 MHz and 865-868 MHz (only in Europe). Each band has its own radiation power and bandwidth regulations.

2.3 RFID Standards

RFID systems do not lack standards. Those standards typically describe the physical and the link layers, covering aspects such as the air interface, anti-collision mechanisms, communication protocols and security functions. Nevertheless, not everything is well covered, and there is a certain absence of standardization in testing methods and application data (notably in protocols and application programming interfaces). *Figure 2.6* summarizes the most important standards [164].

Many organizations making standards around RFID base their standards on existing ones developed by the ISO/IEC, and then augment or modify them to meet the needs of their particular application or design.

The International Organization for Standardization (ISO) is the world's leading developer of international standards. ISO technical standards specify the requirements for products, services, processes, materials, and systems. ISO has also developed standards for good(s) conformity assessment, managerial, and organizational practices.

The International Electrotechnical Commission (IEC) is a leading global organization that prepares and publishes international standards for all electrical, electric, and related technologies. The IEC promotes international cooperation on electrotechnical standardization, such as the assessment of conformity to standards in the fields of electricity, electronics, and related technologies such as RFID.

The EPCglobal has also contributed to RFID standardization. EPC is a joint venture between the EAN International and the Uniform Code Council (UCC). It was chartered to establish and support the Electronic Product Code (EPC) Network as the global specification and leading to the global worldwide standard (ISO) for immediate, automatic and accurate identification of any item in the supply chain, and has become the major organization for the development of RFID specifications.

2.3.1 Contactless Integrated Circuit Cards

ISO 7810 defines a special type of identification card without contact. According to the communication range, three types of cards can be distinguished:

Close-coupled Cards (ISO 10536) These are cards that operate at a very short distance from the reader (< 1 centimeter).

Proximity Cards (ISO 14443) These are cards that operate at an approximate distance of 10 centimeters from the reader. They can be considered as a high-end RFID transponder since they have a microprocessor.

Vicinity Cards (ISO 15693) These are cards that operate at distances greater than one meter. Unlike previous cards (ISO 14443), they usually only incorporate inexpensive state machines, instead of microprocessors.

2.3.2 RFID in Animals

ISO 11784, ISO 11785, and ISO 14223 standardize tags for animal identification in the frequency band below 135 KHz. Initially, standards define an identifier of 64 bits. In ISO 14223, greater blocks for reading and writing, as well as blocks of protected writing, are allowed. There are hardly any differences between the communication protocols defined in ISO 14223 and ISO 18000-2.

2.3.3 Item Management

ISO 18000 defines the air interface, collision detection mechanisms, and the communication protocol for item tags in different frequency bands. The reference architecture is described in part 1, and parts 2-6 specify the characteristics for the different frequency bands. Specifically, part 2 defines tags for low frequency ($<$ 135 KHz). Part 3-1 for HF systems (13.56 MHz) is compatible with ISO 15693 (but with more flexibility in tag design), and part 3-2 specifies a next generation RFID system in the same frequency band with higher bandwidth and faster scanning of multiple tags. Part 4 specifies 2.4 GHz systems: in mode 1, a passive backscatter system and in mode 2, active tags with long range and high-data rates. Part 5 for the 5.8 GHz band is currently withdrawn. Part 6 defines a passive backscatter system around 900 MHz. Part 7 specifies a RFID system with active tags and long range in the 433 MHz band. Finally, ISO/IEC 18000 is employed in conjunction with a great number of application oriented standards:

- ISO/IEC 6346 Freight containers - Coding and marking.
- ISO/IEC 7816 (1-12) Identification cards - Integrated circuit(s) cards with contacts.

- ISO/IEC 1736X and ISO 10374.2 Pertain to large shipping container applications of RFID.
- ISO/IEC 14816 Road traffic and transport telematics. Automatic vehicle and equipment identification.
- ISO/IEC 15691/15692 Deal with the data protocol/application interface as well as encoding rules and memory functions for item management.
- 10374 Define uses of RFID for freight container identification.
- ISO/IEC 17358 Applications requirements include hierarchical data mapping.
- ISO/IEC 17363 Freight containers.
- ISO/IEC 17364 Returnable transport items.
- ISO/IEC 17365 Transport units.
- ISO/IEC 17366 Product packaging.
- ISO/IEC 17367 Product tagging.
- ISO/IEC 10374.2 RFID freight container identification.

2.3.4 Near-Field Communication (NFC)

NFCIP-1 NFC is designed for interactions between tags and electronic devices in close proximity ($<$ 10 cm). The standards ETSI TS 102.190, ISO 18092, and ECMA 340 identically define the Near Field Communications Interface and Protocol-1 (NFCIP-1).

These protocols describe the air interface, initialization, collision avoidance, a frame format, and a block-oriented data-exchange protocol with error handling. Additionally, they describe two different communication modes: active and passive.

NFCIP-2 The Near Field Communication Interface and Protocol-2 (NFCIP-2) specifies the communication mode selection mechanism (ECMA

352). NFCIP-2 compliant devices can enter in three different communication modes: NFCIP-1, ISO 14443, and ISO 15693. All these modes operate at 13.56 MHz and are designed not to disturb other RF fields at the same frequency.

2.3.5 Electronic Product Code (EPC)

The Auto-ID (Automatic Identification) Center was created in October 1999 at the MIT Department of Mechanical Engineering, by a number of leading figures. At the beginning, EPC was developed by the Auto-ID Center. The Auto-ID Center officially closed 26th October, 2003. The center had completed its work and transferred its technology to EPCglobal [64]. EPCglobal is a joint venture between EAN International and the Uniform Code Council (UCC).

2.3.5.1 Tag Data Standard

The EPC Tag Data Standard version 1.3 defines standardized EPC tag data, including how it is encoded on the tag and how it is encoded for use in the information systems layer of the EPC Network System [67]. Standardized EPC data consists of an EPC identifier, which uniquely identifies an individual object, as well as a filter value, which enables effective and efficient reading of the EPC tag. Additionally, some types of EPC tags (Class-1 Generation-2) allow user defined data. Specifically, an EPC number contains:

Header: identifies the length, type, structure, version and generation.

Manager Number: identifies the company or company entity.

Object Class: is similar to a Stock Keeping Unit (SKU).

Serial Number: is the specific instance of the object class being tagged.

The EPC Tag Data Standard version 1.3 does not provide specific guidance for EPC Class-1 Generation-2 Tags. Specifically, the standard defines the following encoding schemes:

GTIN Global Trade Item Number is the globally unique number used to identify trade items, products or services. This term also refers to the entire family of UCC.EAN. A GTIN is a numeric structure containing 8 digits (EAN-8), 12 digits (EAN-12), 13 digits (EAN-13) or 14 digits (EAN-14).

SSCC Serial Shipping Container Code is a fixed length, 20-digit number that contains no classifying elements. It has a trailing check digit and a leading extension digit, which a company can use for internal needs. The central 18 digits represent the company prefix and the serial reference. The SSCC is different for each carton and shipping container, regardless of its content. The SSCC is specially used for tracking cartons containing custom quantities of mixed products.

GLN Global Location Number is a globally unique 13-digit number that provides a standard means of identifying any legal, functional or physical location within a business or organizational entity such as a company (legal entities), a specific department within a legal entity (functional entities), a loading dock (physical entities), etc.

GRAI Global Returnable Asset Identifier provides the unique identification of a returnable asset. A returnable asset is a reusable package or transport equipment of a certain value. The identifier is composed of a company prefix, and a individual asset type. Optionally a serial number, allowing its unique identification, can be included.

GIAI Global Individual Asset Identifier is used to uniquely identify an entity that is part of the fixed inventory of a company. This GIAI can be used to identify any fixed asset of an organization. A fixed asset is defined as: any property used in carrying out the operation of a business, which will not be consumed through use or converted into a cash during the current fiscal period. The identifier is composed of a company prefix and an individual asset reference. The holder of the company prefix determines the numbering of the individual asset reference.

GID General Identifier is defined for a 96-bit EPC, and is independent of any existing identity specification or convention. In addition to the header which guarantees uniqueness in the EPC namespace, the GID

is composed of three fields: the general manager number, object class and serial number.

2.3.6 Tag Protocol

The EPC Class-1 Generation-2 specification defines the physical and logical requirements for a passive backscatter, Interrogator-Talk-First (ITF), radio frequency identification system operating in the 860-960 MHz frequency range [61]. The system comprises interrogators (also known as readers), and tags (also known as labels).

This standard has been ratified by both EPCglobal and ISO. In fact, it can be considered as the universal standard for low-cost RFID tags. Due to its importance, a security analysis of this specification is tackled in *Chapter 4*.

2.3.7 EPCglobal Architecture

The EPC Network (EPCglobal Architecture Framework) is a technological application that will allow organizations to increase their efficiency as a greater visibility of their product information is obtained. This new global and open standard mixes low-cost RFID technology, existing communications networks, and the electronic product code to create precise, effective and real time information. The EPCglobal network is made up of six fundamental components [65]:

EPC (EPC Tag Data Standards Version 1.3 [67]). The Electronic Product Code (EPC) is the next generation in the product identification. The EPC code is composed of a collection of numbers that unequivocally identify an item in the supply chain.

Tag or Labels EPC (EPC Class-1 Generation-2 Version 1.09 [61]). In this system, barcodes are replaced by tags (radio frequency chip joined to an antenna). Each tag stores an electronic product code.

Readers (EPC Reader Protocol Standard, Version 1.1 [66]). The barcodes readers are replaced by RFID readers which are composed of one or several antennas. When a group of items are near a reader, each one

is activated and its stored information collected. EPC readers are located in strategic localizations to facilitate the tracking of item movements.

Middleware (The Application Level Events (ALE) Specification Version 1.0 [62]). Software technology designs to manage and transfer information to avoid the overload of public and corporative networks. The EPC middleware uses a distributed architecture that works in different computers throughout an organization.

EPC-IS (EPC Information Services Specification Version 1.0 [68]). The EPC Information Service allows users to interchange information included in the EPCs between commercial partners by means of the EPC network.

EPC Discovery Service A group of services that allow users to find and obtain access to the associated information of an specific EPC.

ONS - Object Naming Service (EPC Object Naming Service (ONS) Specification Version 1.0 [63]). The ONS service is similar to the Domain Name Service (DNS). Using this service, the information associated with the EPC can be acquired. Specifically, the location of the EPC-IS that stores this information is provided.

2.3.8 Region Regulations

Some standardization and regulatory organizations work within a region. These organizations are either governmental bodies or independent organizations that base their specifications and regulations on international standards. The most significant organizations are outlined bellow:

FCC The Federal Communications Commission is the regulatory authority for the US government for regulating the use of RFID technology. The FCC issues standards that apply to RFID technologies, including RS-232 and RS-485 communications protocols of FCC Part 15 for RF transmissions. FCC Part 15, section 15.247 define the conditions under which RFID devices operating at UHF frequencies in the Industrial, Scientific, and Medical (ISM) bands can operate. This section

also defines operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz. The 902-928MHz band, or UHF, offers optimum range of operation and is usually preferred for supply chain applications. Additionally, new regulations for the use of improved RF identification systems in conjunction with commercial shipping containers (433.5-434.5 MHz) have been adopted.

GASB 34 Government Accounting Standards Boards (GASB) Statement No. 34, which is typically known as GASB 34. This standard requires states and local governments to report the value of their infrastructure assets (including bridges, roads, water, etc). Additionally, the standard demands an asset management system in which RFID technology plays an important role.

ANSI American National Standards Institute. This organization promotes interpretability and compatibility of electronic devices. RFID devices have to be compliant with ANSI NCITS 256, a standard for item management that describes three 2.4 GHz interfaces, 2 UHF interfaces, and the 13.56 MHz interface.

ERO and ECC The European Radiocommunications Offices (ERO) supports the Electronics Communications Committee (ECC), formerly called the European Radiocommunications Committee (ERC). The main task of the ECC is to develop radiocommunications policies and coordinate frequency, regulatory, and technical matters for the allocation and utilization of the 9 KHz to 275 GHz frequency range.

ETSI The European Telecommunications Standards Institute is another regulatory Agency in Europe that regulates the use of RFID technology. The ETSI standards relevant to RFID operation in the UHF bands are defined in EN 300 220.

Chapter 3

Attacking RFID Systems

3.1 Introduction

3.1.1 Background

Press stories about RFID often give inaccurate descriptions of the possibilities that exist for abuse of this technology. They sometimes predict a world where all our possessions will have a unique identification tag: clothes, books, electronic items, medicines, etc. For example, an attacker outside your house equipped with a commercial reader would be able to draw up an inventory of all your possessions, and particular information such as your health and lifestyle could also be revealed. Also, it is said that this technology allows "Big Brother" to know when you are in public places (office, cinemas, stores, pubs, etc.), tracking all your movements and compromising your privacy in terms of your whereabouts (location).

RFID technology is a pervasive technology, perhaps one of the most pervasive in history. While security concerns about the possibility of abuse are legitimate, misinformation and hysteria should be avoided. One should be aware that ways of collecting, storing and analyzing vast amounts of information about consumers and citizens existed before the appearance of RFID technology. For example, we usually pay with credit cards, give our names and address for merchandizing, use cookies while surfing the Internet, etc.

In this chapter we give an overview of the risks and threats related to RFID technology, helping the reader to become better acquainted with this technology. Although the privacy issues are the main focus in literature [36, 50, 95, 97, 112, 122, 126, 180, 185, 186, 193, 199], there are other risks that should be considered when designing a RFID system.

3.1.2 Attack Objectives

The objectives of each attack can be very different. It is important to identify the potential targets in order to understand all the possible attacks. The target can be the complete system (i.e. disrupt the whole of a business system) or only a section of the entire system (i.e. a particular item).

A great number of information systems focus solely on protecting the transmitted data. However, when designing RFID systems, additional objectives, such as tracking or data manipulation should be considered. Imagine the following example in a store: an attacker modifies the tag content of an item reducing its price from 100 to 9.90 €. This leads to a huge loss for the store. In this scenario, the data may be transmitted in a secure form and the database has not been manipulated. However, fraud is carried out because part of the system has been modified. Therefore, in order to make a system secure, all of its components should be considered. Neglecting one component, whatever the security level of the remaining components, could compromise the security of the whole system.

As shown in the above example, the attack may be perpetrated to steal or reduce the price of a single item, while other attacks could aim to prevent all sales at a store. An attacker may introduce corrupt information in the database to render it inoperative. Some attacks, such as the Faraday cage or active jamming, are inherent in the wireless technology employed. Other attacks focus on eliminating physical access control, and ignore the data. Some involve fraudulent border crossing, identity stealing from legitimate e-passports, etc.

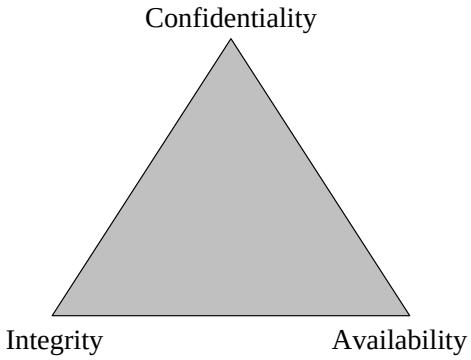


Figure 3.1: Three Pillars of Security: The CIA Triad

3.1.3 Security Needs

As any other mission-critical system, it is important to minimize the threats to the confidentiality, integrity, and availability (CIA) of data and computing resources. These three factors are often referred to as "The Big Three". *Figure 3.1* illustrates the balance between these three factors.

However, not all systems need the same security level. For example, not all systems need 99.999% availability or requiere that its users be authenticated via retinal scans. Because of this, it is necessary to analyze and evaluate each system (sensitivity of the data, potential loss from incidents, criticality of the mission, etc.) to determine the exact confidentiality, integrity, and availability requirements. To give another example, the security requirements of tags used in e-passports should not equal those employed in the supply chain (i.e. tag compliant to EPC Class-1 Generation-2).

Confidentiality: The information should be accessible only to those authorized for access. Privacy information, such as the static identifiers transmitted by tags, fits into the confidentiality dimension. Both users and companies consider this issue of utmost importance. Furthermore, RFID technology allows the tracking of items. From a user perspective tracking should be avoided. However, companies may take advantage of it in controlling the movements of materials in the supply chains, increasing the productivity of their processes.

Integrity The assurance that the messages transmitted between two parties are not modified in transit. Additionally, some systems provide the authenticity of messages. The recipient is sometimes even able to prove that a message was originated by the purported sender and is not a forgery (non-repudiation). An example of this kind of attack is the spoofing attack.

Availability System availability is whether (or how often) a system is available for use by its intended users. This factor will determine the performance and the scalability level of the system. DoS attacks are usual threats against availability (i.e. active jamming of the radio channel or preventing the normal operation of vicinity tags by using some kind of blocker tag).

Each time a new technology is implanted, contingency plans for various points of failure should be designed. We recommend periodical security audits in order to review the security polices, procedures and IT infrastructures. As has been frequently mentioned, RFID technology may be a replacement for barcode technology. Nevertheless, new risk scenarios should be considered with its implantation. For example, consider the repercussions of a barcode reader failing or an RFID reading going down. When a barcode reader fails, an operator can manually enter the codes into the terminal and the system works, albeit with relatively slowness. On the other hand, if the RFID reader is processing high volumes of items and these items are moving at high speed, the consequences will be much worse. Security needs should therefore be considered a priority.

3.2 Main Security Concerns

3.2.1 Privacy

> *No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks [1].*

> *Whereas data-processing systems are designed to serve man; whereas they must, whatever the nationality or residence of individuals, respect their fundamental rights and freedoms, notably the right to privacy, and contribute to economic and social progress, trade expansion and the well-being of individuals [2].*

Privacy has no definite boundaries and its meaning is not the same for different people. In general terms, it is the ability of an individual or group to keep their lives and personal affairs out of public view, or to control the flow of information about themselves.

The invasion on privacy by governments, corporations or individuals is controlled by a country's laws, constitutions or privacy laws. For example, taxation processes normally require detailed private information about earnings. The EU Directive 95/46/EC [2] on the protection of individuals with regard to the processing of personal data and the free movement of this, limits and regulates the collection of personal information. Additionally, Article 8 of the European Convention of Human Rights identifies the right to have private and family life respected. Within this framework, monitoring the use of e-mails, internet or phones in the workplace, without notifying employees or obtaining their consent can result in legal action.

RFID technology is a pervasive technology, and seems destined to become more and more so. As Weiser already predicted in 1991, one of the main problems that ubiquitous computing has to solve is privacy [225]. Leakage of information is a problem that occurs when data sent by tags reveals sensitive information about the labeled items. Products labeled with insecure tags reveal their memory contents when queried by readers. Usually, readers are not authenticated and tags answer in a completely transparent and indiscriminate way.

As an example of the threat this could pose, consider the pharmaceutical sector where tagged medication is planned for the immediate future. Imagine that when you leave the chemist's with a given drug -say an antidepressive or AIDS treatment, an attacker standing by the door equipped with a reader could find out what kind of medication you have just bought. In a similar scenario, thieves equipped with tag readers could search people, selecting those with multiple tagged bank bills to rob, and they would

know how much they would earn with each robbery.

Advanced applications, where personal information is stored in the tags, have appeared recently. E-passports are a good example of this sort of application. As part of its US-VISIT program, the United States government mandated the adoption of e-passports by the twenty-seven nations in its Visa-Waiver Program. A combination of RFID technology and biometric technology are employed [98, 111, 126]. The RFID tags store the same information that is printed on its first page (name, data of birth, passport number, etc) as well as biometric information (facial image). In phase-2 of the European e-passport project [10], the biometric data from two fingerprints, which is very sensitive information, will also be stored.

Several organizations like CASPIAN [41], and FOEBUD [7] are strongly against the massive deployment of RFID technology. They believe that RFID technology will lead to a significant loss of citizens' privacy. Some of CASPIAN's activities include successful boycott campaigns against important companies like Benetton [32, 106], Tesco [212], and Gillette [85], to name but a few. Additionally, a book titled "SPYCHIPS: How Major Corporations and Government Plan to Track your Every Move with RFID" and published in 2005 [21], has contributed to promoting suspicion about RFID technology.

Another example of objection to RFID technology is the case of California State Senator Joe Simitian (Senate Bill 682), who planned to restrict the use of identification systems based on RFID technology: "The act would prohibit identity documents created, mandated, or issued by various public entities from containing a contactless integrated circuit or other device that can broadcast personal information or enable personal information to be scanned remotely" [69]. Due to significant industry opposition, Bill 682 was stalled in the Assembly Appropriations Committee and an important missed deadline resulted in the expiry of the Bill. Legislative manoeuvring allowed the resurrection of the case by means of Bill 768 [189]. This bill was finally vetoed by California Governor Arnold Schwarzenegger. In particular, Bill 768 proposed to:

1. Criminalize the "skimming" of personal data from RFID-enable iden-

tification documents.

2. Implement specific provisions to ensure the security of data contained in such identification documents.

3. Impose a three-year moratorium on the use of RFID technology in certain types of government issued identification documents.

In 2002, Garfinkel proposed a set of rights that should be upheld by any system that uses RFID technology [78]. Consumers should have:

1. The right to know whether products contain RFID tags.

2. The right to have RFID tags removed o deactivated when they purchase products.

3. The right to use RFID-enabled services without RFID tags.

4. The right to access an RFID tag's stored data.

5. The right to know when, where and why the tags are being read.

These rights are not necessarily considered as the basis for a new law, but as a framework for voluntary guidelines that companies wishing to deploy this technology may adopt publicly.

3.2.2 Tracking

Location information is a set of data describing an individual's location over a period of time [56]. The resolution of the system (time and localization) depends on the technology used to collect data.

Indeed, location privacy can be viewed as a particular type of privacy information [33]. A secondary effect of wireless communication is that information can be made public and collected. In a mobile phone context, regions are divided up into cells. Each time a phone enters a new cell, the mobile is registered. Mobile phone operators record handset location information and supply it to third parties (i.e. police, the company that subscribed the localization service, etc). Other techniques such as triangulation can be used to increase the precision of the system. The new localization services

(i.e. third-generation mobile phones) allow an accuracy of a few meters by means of the incorporation of a Global Positioning System (GPS) receiver. In data network context, Wireless 802.11 Ethernet cards obtain connectivity by registering with access points which could be used to locate a network device.

RFID technology is not a high-tech bugging device. It does not possess GPS functionality or the ability to communicate with satellites. RFID tags do not have the storage and transmission capability for large quantities of information. An RFID system is normally composed of only three components: tags, readers and a back-end database. Readers are connected, using a secure channel, to the database. When a database is present in the system, tags might only transmit an identifier. This identifier is used as a index-search in the database to obtain all the information associated with the tag. Therefore only people with access to the database can obtain the information about the labeled item.

Most of the time, tags provide the same identifier. Although an attacker cannot obtain the information about the tagged item, an association between the tag and its holder can easily be established. Even where individual tags only contain product codes rather than a unique serial number, tracking is still possible using an assembly of tags (constellations) [223]. To clarify the potential risks of tracking, some examples are given:

Wall-Mart Wall-Mart is an American public corporation, currently one of the world's largest. It has concentrated on streamlining the supply chain, which is why it encourages all its suppliers to incorporate RFID technology. The substitution of barcodes by RFID tags allows an increase in the reading-rate of the pallets as they move along the conveyor belt. RFID readers can automatically scan these as they enter or leave the warehouse, saving time and improving product flow. Right now, RFID technology is used at pallet level. Individual packaging is the next logical step.

Individual Product Packaging Imagine that your Tag Heuer bifocals possess a tag, and this tag stores a 96-bit static identifier, allowing an attacker to establish a link between the identifier and you. On associa-

tion, an attacker could know when you passed through a given place, for example when you enter or leave your home, when you arrive at or leave your office etc. Even worse, the attacker could place several readers in your favorite mall. He could collect data over a long time (data, time, shop, etc.) acquiring a consumer profile of you. Finally, he could send you unsolicited personalized advertising information depending on your shopping habits.

E-passports Since October 2006, USA required the adoption of e-passports by all the countries in its Visa-Waiver Program. The ICAO standard specifies one mandatory cryptographic feature (passive authentication) and two optional cryptographic features (basic access control and active authentication). Passive authentication only demonstrates that tag content is authentic but it does not prove that the data container is secure. Basic authentication ensures that tag content can only be read by an authorized reader. Additionally, a session key is established, encrypting all the information exchanged between the tag and the reader. Active authentication is an anti-cloning feature, but it does not prevent unauthorized readings. Independently of the security mechanism used, tracking is possible. The electronic chip required by the ICAO must conform to ISO/IEC 14443 A/B already adopted in other applications [4, 11]. The collision avoidance in ISO 14443 uses unique identifiers that allow readers to distinguish one tag from another [111]. However, this identifier will allow an attacker to unequivocally identify an e-passports's holder. One simple countermeasure is to generate a new random identifier each time the tag is read.

As has been shown, RFID technology is not the first one that permits the tracking of people (i.e. video surveillance, mobile phone, Wireless 802.11 Ethernet cards , GPS, etc.). Nevertheless, the equipment used to track people holding RFID tags is not very expensive. If we return to the example of tracking in a mall, we will understand one of the principal differences between RFID and other localization technologies. The vast majority of malls have a video surveillance system. You can be filmed in all the supermarket sections in which you buy an item. Then, the information obtained by the system (images) has to be processed to obtain your consumer profile.

Table 3.1: Tag Frequencies and Reading Distances

Frequency band	Frequency	Distance	Energy Transfer
Low (LF)	125 KHz	1 - 90 cm, typically around 45 cm	Inductive coupling
High (HF)	13.56 MHz	1 - 75 cm, typically under 40 cm	Inductive coupling
Ultra High (UHF)	865 - 868 MHz 902 - 928 MHz 433 MHz	Up to 9 m	Electromagnetic coupling
Microwave (μW)	2.45 Ghz 5.8 GHz	Typically 0.3 - 0.9 m	Electromagnetic coupling

However, if RFID technology were employed, data could be automatically collected without the need for subsequent data processing as in video systems.

3.3 Tags and Readers

3.3.1 Operating Frequencies and Reading Distances

RFID tags operate in four primary frequency bands [230]:

1. Low Frequency or LF (120-140 KHz).
2. High Frequency or HF (13.56 MHz).
3. Ultra High Frequency or UHF (860-960 MHz).
4. Super High Frequency/Microwave (2.45 GHz and above).

The characteristics of different frequencies are summarized in *Table 3.1*.

Low Frequency (LF) Tags These tags operate at 120-140 KHz. They are generally passive and use near field inductive coupling. So they are suited for applications reading small amounts of data at relatively slow speeds and at short distances. Their read range varies from 1 to 90 cm, typically below 45 cm. LF tags do not support simultaneous tag reads. LF tags are relatively costly because they require

a longer, more expensive copper antenna. They penetrate materials such as water, tissue, wood and aluminium. Their most common applications are in animal identification, automobile security, electronic article surveillance, commerce and other areas.

High Frequency (HF) Tags These tags operate at 13.56 MHz. They are typically passive and frequently use inductive coupling. HF tags penetrate materials well, such as water, tissue, wood, aluminium, etc. Their data rates are higher than LF tags and their cost is lower due to the simple antenna design. Their read ranges varies from 1 cm to 75 cm, typically under 40 cm. HF tags are used in smart shelf, smart cards, libraries, baggage handling, and similar applications.

Ultra High Frequency (HF) Tags UHF active and passive tags can operate at different frequencies. UHF active tags operate at 433 MHz, and UHF passive tags usually operate at 860-960 MHz. Generally, passive UHF tags are not very effective around metals and water. They perform well at distances greater than 90 cm. UHF passive tags usually reach about 9 m. UHF tags have good non-line-of-sight communication, have a high data rate, and can store relatively large amounts of data.

Super High Frequency/Microwaves Tags These tags operate at frequencies of 2.45 GHz and above (also 5.8GHz) and can be either active or passive. Their characteristics are similar to those of UHF tags. However, they have faster read rates and are less effective around metals and liquids than tags of lower frequencies. These tags can be smaller in size compared to LF, HF and UHF tags and are used for electronic toll collection as well as for the tracking of shipping containers, trains, commercial vehicles, parking, etc. The read range varies from 0.3 m to 0.9 m for passive tags, and is very dependent on design. Active systems also use microwave frequency.

3.3.2 Eavesdropping

RFID technology operates through radio, so communication can be surreptitiously overheard. In [182], the possible distances at which an attacker

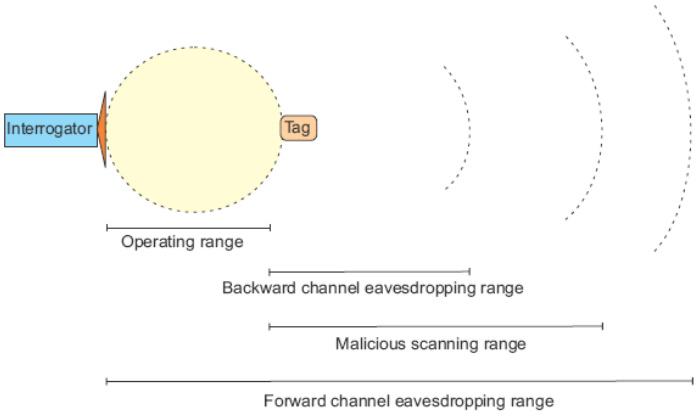


Figure 3.2: Eavesdropping Range Classification [182]

can listen to the messages exchanged between a tag and a reader are categorized (see *Figure 3.2*).

Forward Channel Eavesdropping Range In the reader-to-tag channel (forward channel) the reader broadcasts a strong signal, allowing its monitoring from a long distance.

Backward Channel Eavesdropping Range The signal transmitted in the tag-to-reader (backward channel) is relatively weak, and may only be monitored in close proximity to the tag.

Operating Range The read ranges shown in the above section are the operating read range using sales-standard readers.

Malicious Scanning Range An adversary may build his own reader archiving longer read ranges, especially if regulations about radio devices are not respected. A conversation between a reader and a tag can be eavesdropped over a greater distance than is possible with direct communication. For example, tags compliant to ISO 14443 have a reading distance of around 10 cm (using standard equipment). However, Kfir et al. showed that this distance can be increased to 55 cm employing a loop antenna and signal processing [120].

Eavesdropping is particular problematic for two reasons:

1. Feasibility: it can be accomplished from long distances.
2. Hard Detection: it is purely passive and does not imply any kind of power signal emission.

Eavesdropping attacks are a serious threat mainly when sensitive information is transmitted on the channel. To give an example, we consider the use of RFID technology in payments cards (RFID credit cards) [24]. In an eavesdropping attack, information exchanged between the credit card reader and the RFID credit card is captured. Heydt-Banjamin et al. showed how this attack can be carried out [96]. An antenna was located next to an off-the-shelf RFID credit card reader. The radio signal picked up by the antenna was processed to translate it into human readable form. In particular, the following pieces of data were captured: cardholder name, complete credit card number, credit card expiry date, credit card type, and finally information about software version and supported communications protocols. As the above example shows, eavesdropping attacks should therefore be considered and treated seriously.

3.3.3 Authentication

Entity authentication allows the verification of the identity of one entity by another. The authenticity of the claimed entity can only be ascertained for the instant of the authentication exchange. A secure means of communication should be used to provide authenticity of the subsequent data exchanged. To prevent replay attacks, a time variant parameter, such as a time stamp, a sequence number, or a challenge may be used. Messages exchanged between entities are called tokens. At least one token has to be exchanged for unilateral authentication and at least two tokens for mutual authentication. An additional token may be needed if a challenge has to be sent to initiate the protocol.

In RFID context, the first proposals found in the literature are based on unilateral authentication [71, 160, 215]. However, the necessity of mutual authentication has been confirmed in many publications [47, 107, 155, 163]. In ISO/IEC 9784, different mechanisms for entity authentication are described [3]:

- Part 1: General model.
- Part 2: Entity authentication using symmetric techniques.
- Part 3: Entity authentication using a public key algorithm.
- Part 4: Entity authentication using a cryptographic check function.

The use of a cryptographic check function seems to be the most adequate solution for RFID. Due to the fact that standard cryptographic primitives exceed the capabilities of a great number of tags, the design of lightweight primitives is imperative, at least for low-cost RFID tags.

The two entities (claimant/verifier) share a secret authentication key. An entity corroborates its identity by demonstrating knowledge of the shared key. This is accomplished by using a secret key with a cryptographic check function applied to specific data to obtain a cryptographic check value. This value can be recalculated by the verifier and compared with the received value. The following mechanisms, as shown in *Figure 3.3*, are possible.

3.3.4 Skimming

> *Takashimaya, one of the largest retailers in Japan, now sells anti-skimming cards called "Sherry" at their department stores. Consumers can just put the cards in their wallets in order to prevent their RFID-chipped train passes, etc. from skimming attacks.*
> *The anti-skimming card functions by creating a reverse electromagnetic field like Taiyo's technology [5].*

Eavesdropping is the opportunistic interception of information exchanged between a legitimate tag and legitimate reader. However, skimming occurs when the data stored on the RFID tag is read without the owner's knowledge or consent. An unauthorized reader interacts with the tag to obtain the data. This attack can be carried out because most of the tags broadcast their memory content without requiring any kind of authentication.

One interesting project is Adam Laurie's RFIDIOt [131]. Specifically, RFIDIOt is an open source library for exploring RFID devices. Several experiments with readers operating at 13.56 MHz and 125/134.2 KHz are shown.

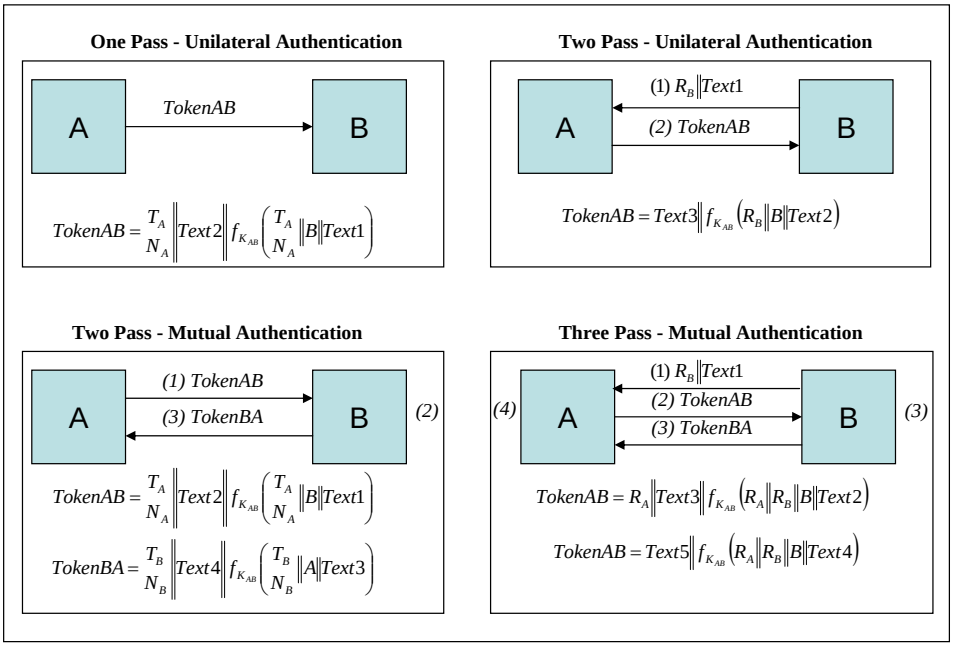


Figure 3.3: Entity Authentication Mechanisms

The number of standards supported by the library is around 50. Some examples of the attacks carried out are the following:

Non-authentication In 2004, Verychip received approval to develop a human-implant RFID microchip [16]. About twice the length of a grain of rice, the device is typically implanted above the triceps of an individual's right arm. Once scanned at the proper frequency, the Verichip answers with an unique 16-digit number which can correlate the user to the information stored on a database. The type of tag used by Verichip appears to be an EM4x05. This kind of tag can be read simply with the program "readlfx.py", obtaining the following information: card ID, tag type, application identifier, country code, national ID.

Password Authentication Since 2003, the Oyster card has been used on Transport for London and National Rail services. The Oyster card is a contactless smart card, with a claimed proximity range of about 8cm, and based on Philips's MIFARE® standard [13]. A code for attack-

ing this kind of card is included in the project. The sample program "bruteforce.py" can be run against it, and it will try to log in the sector 0 by choosing random numbers as the key.

Nowadays, the security of e-passports have aroused a great interest [40, 88, 98, 111]. Skimming is problematic because e-passports possess very sensitive data. The mandatory passive authentication mechanism demands the use of digital signatures. A reader will be able to verify that the data came from the correct passport-issuing authority. However, digital signatures do not link data to a specific passport. Additionally, if only passive authentication is supported, an attacker equipped with a reader could obtain sensitive information such as your name, birthday or even your facial photograph. This is possible because readers are not authenticated -in other words, the tag answers indiscriminately. Certain projects exist which openly give the code needed to read e-passports: RFIDIOt (Adam Laurie) [131], OpenMRTD (Harald Welte) [226], JMRTD (SoS group, ICIS, Radbound University) [198].

3.3.5 Cloning and Physical Attacks

Symmetric-key cryptography can be used to avoid tag cloning attacks. Specifically, a challenge-response mechanism like the following can be employed. First, the tag is singulated from many by means of a collision-avoidance protocol like the binary tree walking protocol. The tag (T_i) shares the key (K_i) with the reader. Afterwards, the following messages are exchanged:

1. The reader generates a fresh random number (R) and transmits it to the tag.
2. The tag computes $H = g(K_i, R)$ and sends it back to the reader.
3. The reader computes $H' = g(K'_i, R)$ and checks its equality with H.

The g function can be implemented by a hash function or, alternatively, by an encryption function. Note that if the g function is well constructed and appropriately deployed, it is infeasible for an attacker to impersonate the

tag. Because standard cryptographic primitives (hash functions, message authentication codes, block/stream ciphers, etc.) are extravagant solutions for low-cost RFID tags on account of their demand for circuit size, power consumption and memory size [108], the design of new lightweight primitives is pressing.

For some kinds of tags, resources are not so restricted. However, their cost is much higher than low-cost RFID tags (i.e. tags used in supply chain). An example of these sort of tags are e-passports. The active authentication method is an anti-cloning feature. The mechanism relies on public cryptography. It works by forcing e-passports to prove possession of a private key:

1. The tag generates an 8-byte nonce and sends it to the tag.
2. The tag digitally signs this value using its private key, and transmits it to the reader.
3. The reader can verify the correctness of the response with the public key supposedly associated with the passport.

Tamper-resistant microprocessors are used to store and process private and sensitive information, such as private keys or electronic money. The attacker should not be able to retrieve or modify this information. To achieve this objective, chips are designed so that the information is not accessible using external means and can only be accessed by the embedded software, which should contain the appropriate security measures.

Making simple electronic devices secure against tampering is very difficult, as a great number of attacks are possible, including [221]:

✓ Mechanical Machining
✓Laser Machining
✓Energy Attacks
✓Temperature Imprinting
✓Probe Attacks
✓Active or Injector Probes
✓Water Machining
✓Shaped Charge Technology
✓Radiation Imprinting
✓High Voltage Imprinting
✓Passive Probes
✓Pico Probes

✓Energy Probes
✓Manual Material Removal
✓Clock Glitching
✓Electronic Beam Read/Write
✓Imaging Technology
✓ Matching Methods
✓High or Low Voltage
✓Circuit Disruption
✓IR Laser Read/Write

As sensitive information such as cryptographic keys are stored on the chips, tamper resistant devices may be designed to erase this information when penetration of their security encapsulation or out-of specification environmental parameters is detected. Some devices are even able to erase all their information after their power supply has been interrupted.

In the RFID context we have to distinguish between low-cost RFID tags and tags used in applications without severe price restrictions. Low-cost RFID tags are very constrained resources (storing, computing and energy consumption). These kinds of tags are usually non-resistant to physical attacks. An example are tags compliant with the EPC Class-1 Generation-2 specification [61]. High-cost tags, sometimes called contact-less chips or smart cards, are not so restrictive regarding resources. However, price increases from 0.05 €to several euros. For example, the chips used in e-passports have an EAL 5+ security level, the highest security level for chips [133]. Therefore, an attacker will not be able to acquire the private key used in private authentication in order to avoid cloning attacks. The plusID tag, manufactured by Bradcom, is another example of tamper resistant tags [206]. Initially, its security level was 2 (tamper evidence) according to Federal Information Processing Standards (FIPS), but it was finally increased to level 3 (tamper-resistant).

3.3.6 Replay and Relay Attacks

A replay attack copies a stream of messages between two parties and replays it to one or more parties. A generalized definition of a replay attack could be the following: an attack on a security protocol using replay of messages from a different context into the intended (or original and expected) context, thereby fooling the honest participant(s) into thinking they have

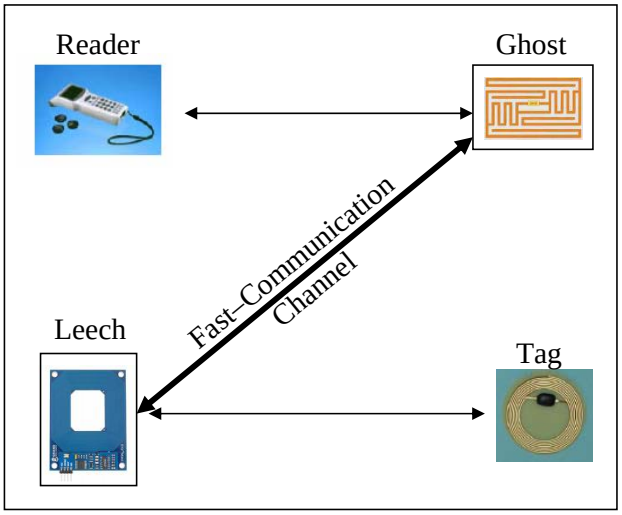


Figure 3.4: Relay Attacks

successfully completed the protocol run [147]. An exhaustive classification of replay attacks can be found in [209].

Common techniques to avoid replay attacks are incremental sequence number, clock synchronization, or a nonce. In [25], a set of design principles for avoiding replay attacks in cryptographic protocols is presented. In a RFID context, clock synchronization is not feasible since passive RFID tags cannot make use of clocks, as these kind of tags do not have an on-board power source. Incremental sequence such as session tokens may be a straightforward solution if tracking is not considered a threat. Therefore, the use of a nonce seems to be the most suitable option for RFID tags.

A number of factors combine to make relay attacks on RFID technology possible. Tags are read over a distance and activated automatically when close to a reader. Therefore, an attacker could communicate with a tag without the knowledge of its owner.

Two devices, as shown in *Figure 3.4*, are involved in the relay attack: the ghost and the leech [120]. The ghost is a device which fakes a card to the reader, and the leech is a device which fakes a reader to the card. A fast communication channel between the legitimate reader and the victim card is created by the ghost and the leech:

1. The legitimate reader sends a message (A) to the ghost.
2. The ghost receives it and forwards this message (A) to the leech through the fast communication channel (minimum delay).
3. The leech fakes the real reader, and sends the message (A) to the legitimate tag.
4. The legitimate tag computes a new message (B) and transmits it to the leech.
5. The leech receives it and forwards this message (B) to the ghost through the fast communication channel.
6. The ghost forwards this message (B) to the real reader.

This sort of attack dispels the assumption that readers and tags should be very close to communicate. Additionally, even if communications were encrypted, the attack is feasible because messages are only relayed through a fast communication channel, without requiring knowledge of their contents. In [89], a practical relay attack against ISO 14443 compliant tags is described.

3.3.7 Hiding

RFID technology uses electromagnetic radio waves. Labeled items can be therefore protected by insulating them from any kind of electromagnetic radiation:

Faraday Cage A Faraday cage or shield is a container made of conducting material, or a mesh of such material. This blocks out radio signals of certain frequencies. There are currently a number of companies that sell this type of solution [15, 152].

Passive Jamming Each time a reader wants to interact with a single tag, the tag will have to be singulated from a population of tags. A collision-avoidance protocol such as Aloha or Binary tree walking may be employed. To conceal the presence of a particular tag, this could simulate the full spectrum of possible tags in the singulation

phase, hiding its presence. This concept was first introduced by Juels et al. as the "Blocker tag" [113]. In 2004, a variant of the blocker concept, named "soft blocking", was introduced [110]. This involves software (or firmware) modules that offer a different balance of characteristics from ordinary blockers.

Active Jamming Another way of achieving isolation from electromagnetic waves is disturbing the radio channel -known as active jamming of RF signals. This disturbance may be implemented with a device that actively broadcasts radio signals, so as to completely disrupt the radio channel, thus preventing the normal operation of RFID readers. However, in most cases government regulations on radio emissions (power and bandwidth) should be violated [130].

3.3.8 Deactivation

Some methods exist for deactivating tags and rendering them unreadable. The most common method consists on generating a high-power RF field that induces sufficient current to burn out a weak section of the antenna. The connection between the chip and the antenna is cut off, rendering it useless. This method is usually chosen to address privacy concerns and to deactivate tags that are used to label individual items or prevent thefts in stores.

The benefits of using RFID technology in a store are clear. However, the deactivation of tags may be malicious. The necessary technology can be available to anyone. The usual range of a "kill" signal is only a few centimeters. However, designing and building a high-gain antenna with a high power transmitter is easy. Using batteries, it could probably fit into a back pack. Then an attacker entering a store could kill all the tags, causing widespread retail chaos. A practical implementation of this sort of attack is the RFID-Zapper project [51, 153].

Karjoth et al. proposed the use of physical RFID structures that permit a consumer to disable a tag by mechanically altering it [116]. In "clipped tags", the consumer can physically separate the body (chip) from the head (antenna) in an intuitive way. Such separation provides visual confirmation that the tag has been deactivated. Then, the tag can be reactivated by means

of physical contact. The reactivation requires deliberate actions on the part of its owner. Indeed, reactivation cannot be carried out without the owner's knowledge unless the item is stolen.

To avoid wanton deactivation of tags, the use of kill passwords has been proposed. Tags compliant to the EPC Class-1 Generation-2 implement this feature [61]. When an EPC tag receives the "kill" command, it renders itself permanently inoperative. However, to protect tags from malicious deactivation, the kill command is PIN protected. One of the main problems linked to this kind of solutions is password management. Employing the same password for all tags is a very naive solution: if a single tag is compromised, all the tags would be at risk. Another straightforward solution is that each tag has a different password, with the associated management and scalability problems.

The potential benefits of RFID technology usage are reduced if tags are permanently deactivated. Instead of killing tags, they could be put to sleep, rendering them only temporarily inoperative. As with the killing process, sleeping/waking up tags will not offer real protection if anyone is able to accomplish these operations. So some form of access control, such a PINs, will be needed to sleep/wake up a tag.

3.3.9 Cryptographic Vulnerabilities

In the 19th century, Kerckhoffs sets out the principles to the security of cryptography systems [119]:

1. *The system must be practically, if not mathematically, indecipherable.*
2. *It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.*
3. *Its key must be communicable and retainable without the help of written notes, and changeable or able to be modified at the will of the correspondents.*
4. *It must be applicable to telegraphic correspondence.*
5. *It must be portable, and its usage and function must not require the concourse of several people.*

6. *Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.*

RFID tags are very constrained devices, with restrictions in power consumption, storage and circuitry. Due to these severe limitations, some commercial RFID tags support weak cryptographic primitives, and thus vulnerable authentication protocols. Additionally, some of these cryptographic primitives are proprietary. The use of proprietary solutions is not really inadequate if algorithms are published to be analyzed by the research community. However, as time has shown, the security of an algorithm cannot reside in its "obscurity". A system relying on security through obscurity may have theoretical or actual security vulnerabilities, but its owners or designers incorrectly believe that the flaws are unknown, and that attackers are unlikely to find them [119].

Texas Instruments manufactures a low-frequency tag, named Digital Signature Transponder (DST). The DST executes a challenge-response protocol. The reader and the DST share a secret key K_i. The reader sends a challenge R to the DST. The DST computes an encryption function of the challenge $C = e_{K_i}(R)$ and sends this value to the reader. The reader computes $C' = e_{K'_i}(R)$ and compares this value with the received value. The challenge is 40 bits in length, and the output of the encryption function is 24 bits length. The length of the K_i is only 40 bits. It is a very short length. The National Institute of Standards and Technology (NIST) [8] and the ECRYPT EU Network of Excellence on cryptography [12] recommended in 2005 a key length of 80 bits for a minimal level of general purpose protection, and 112 bits for the following ten years.

The most common uses of DST are the following:

1. The DST is employed as a theft-deterrent (immobilizer keys) in automobiles, such a Ford and Toyota vehicles.

2. The DST serves as a wireless payment device (speedpass), which can be used by more than seven million individuals in around 10,000 Exxon and Mobile gas stations.

Texas Instruments has not published details of the encryption algorithm, basing itself on security through algorithm obscurity. A team of researchers at Johns Hopkins University and RSA Laboratories discovered, however, serious security vulnerabilities in the DST [37]. In particular, a successful reverse engineering of the DST encryption algorithm was accomplished. First, a rough schematic of the cipher was obtained from a published Texas Instruments presentation. With the reverse-engineering of the cipher, they showed that a 40-bit key length was inadequate, and that the cipher was not only vulnerable to brute-force attacks. The proposed attack can be divided into three phases:

Reverse Engineering They were equipped with a DST reader and some blank DST tags. With the reader and the blank tags, the output of the encryption function, with any key and challenge, could be obtained. Using specific key/challenge pairs and centering on the schematic of the encryption, operational details of the algorithm were derived.

Key Cracking After determining the encryption algorithm, a programmed hardware "key cracker" was implemented to recover the unique cryptographic key of the DST. The cracker operated by brute force (full space of 2^{40}). Given two input-output pairs, it took around 30 minutes to recover the secret key.

Simulation They programmed a hardware device with the recovered key from the DST. This device could impersonate the original DST.

The research on the DST exemplifies the importance of Kerckhoffs's principles. Another significant and similar example is the proprietary CRYPTO1 encryption algorithm used in Philips Mifare cards, which has been recently reverse-engineered [117, 159]. We recommend the publication of any used algorithms. Open algorithms can be analyzed and refined by the scientific community, bolstering confidence in their security.

3.4 Back-end database

3.4.1 Tag Counterfeiting and Duplication

Since the incorporation of RFID technology in sensitive applications such as passports [178] or pharmaceutical pedigrees [220], the possibility of creating counterfeiting tags has unleashed some concerns.

Here are some arguments that may dissuade users from being too alarmist [86]:

1. Usually, each tag has an unique identifier (ID) that allows its unequivocal identification. To counterfeit a tag, one would have to modify the identity of an item, which generally implies tag manipulation. The tag (ID) implementation may vary in each manufacturer as well as in each product. The major manufacturers first programme the tag and then lock it. So resistance to these attacks lies in the lock. In most cases, it is not possible to unlock the tag without using invasive techniques. These techniques are not commonly available to the general public.

2. RFID tags are generally sold pre-programmed with their identifiers, this being one of the phases of the normal production process. The ID format usually accords with a standard. The non-availability of blank tags will therefore reduce the possibility of counterfeiting.

3. An alternative is the design of blank tags. However, even with the equipment necessary for IC fabrication, designing these kind of chips is not an easy task.

Despite the difficulty of counterfeiting tags, on some occasions tags have been duplicated. It is a similar problem to that of credit card fraud where a card is duplicated and possibly used in multiple places at the same time. As duplicate tags cannot be operatively distinguished, the back-end database should detect rare conditions. An example of a rare condition is the following: a tag cannot be in the toll gate on the Madrid-Barcelona motorway and fifteen minutes later in the toll gate of Almería-Sevilla motorway. The design of back-end database should be considered case by case [213].

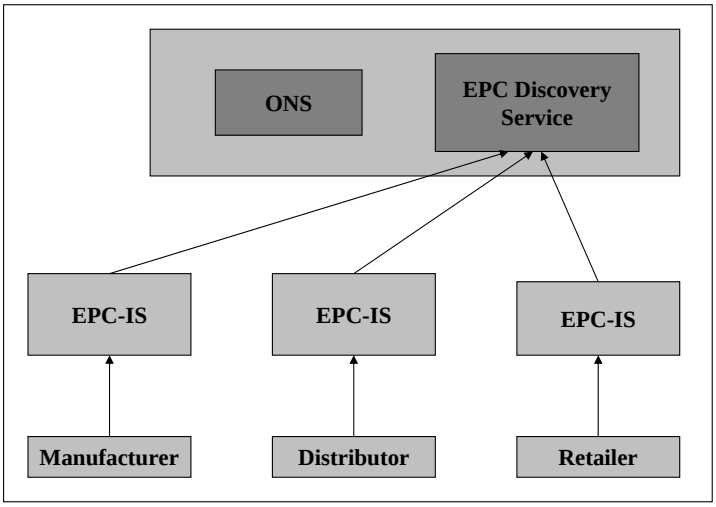


Figure 3.5: EPCglobal Network

3.4.2 EPC Network: ONS Attacks

The EPCglobal network is made up of three key elements, as displayed in *Figure 3.5*:

1. EPC Information Services.
2. EPC Discovery Services.
3. Object Name Service.

When an RFID tag is manufactured with an EPC, the EPC is registered within the ONS. The RFID tag is attached to a product, and the EPC becomes a part of that product as it moves through the supply chain. The particular product information is added to the manufacturer's EPC-IS, and the knowledge that this data exists within the manufacturer's EPC-IS is passed to the EPC Discovery Service.

The ONS is a distributed but authoritative directory service that routes request for information about EPCs. Existing or new network resources can be employed to route the requests. The ONS is similar to DNS in both technology and functionality. When a query is sent to the ONS including the EPC code, one or more localizations (Uniform Resource Locator or URL)

where information about items reside, are returned. The ONS service is divided in two layers. First, the Root ONS, which is the authoritative directory of manufacturers whose products may have information on the EPC Network. Second, the Local ONS, which is the directory of products for that particular manufacturer.

As the ONS can be considered a subset of the DNS, the same security risks are applicable. In 2004, a threat analysis of the Domain Name System was published as RFC 3833 [23]. Some of the principal threats identified were the following:

1. Packet interception: manipulating IP packets carrying DNS information.
2. Query prediction: manipulating the query/answer schemes of the DNS protocol.
3. Cache Poisoning: injecting manipulated information into DNS caches.
4. Betrayal by trusted server: attacker controlling DNS servers in use.
5. Denial of service (DoS): DNS is vulnerable to DoS as happens in any other network service. Additionally, the DNS itself might be used to attack third parties.

However, there are some risks that are particular to the ONS service [70]:

1. Privacy: There are many situations where the EPC of an RFID tag can be considered highly sensitive information. Sensitive information can be obtained even knowing just part of the EPC. For example, knowing only the class of the identifier, you can find out the kind of object. To obtain the information associated with a tag, the EPC-IS has to be located. Even if the connections to the EPC-IS are secured (i.e. with SSL/TLS), the initial ONS look-up process is not authenticated nor encrypted in the first place. Therefore, sensitive information passes in clear on the channel (middleware - networks - DNS server).
2. Integrity: The correctness and the completeness of the information should be guaranteed. An attacker controlling intermediate DNS

servers or launching a successful man-in-the-middle attack could forge the list of URLs (i.e. a fraudulent server). To prevent this attack, an authentication mechanism should be used for the EPC-IS.

3. Availability: If the adoption of the EPC network is widespread, there will be a great number of companies dependent on network services. ONS will become a service highly exposed to attacks. These could include Distribute Denial-of-Service (DDoS) attacks, that reduce the functioning of the server or its network connection by issuing countless and intense queries, or targeted exploits that shut down the server software or its operating system.

3.4.3 Virus Attacks

The RFID tag memory generally contains an unique identifier, but additional data may be stored. This data size varies from a few bytes to several kilobytes. The memory where this additional information is stored is rewritable. The information sent by the tags is usually implicity trusted by the database, which implies some security threats [184, 213]:

1. Buffer Overflow: Buffer overflow is one of the most frequent sources of security vulnerabilities in software. Programming languages, such as C or C++, are not memory safe. In other words, the length of the inputs are not checked. An attacker could introduce an input that is deliberately longer, writing out of the buffer. As program control data is often located in memory areas adjacent to data buffers, the buffer overflow may lead the program to execute an arbitrary code. As a great number of tags have severe storage limitations, resource rich tag simulating devices could be utilized [105].

2. Code Insertion: An attacker might inject malicious code into an application, using any script language (i.e. CGI, Java, Perl, etc.). RFID tags with data written in a script language could perform an attack of this kind. Imagine that the tags used for tracking baggage in the airport contain the airport destination in its data field. Each time a tag is read, the back-end system fires the query, "select * form location_table where airport =< tag data>". Imagine that an attacker

stores in one piece of baggage "MAD;shutdown". When this data is read, the database will be shutdown and the baggage system will crash.

3. SQL Injection. SQL injection is a type of code insertion attack, executing SQL code in the database that were not intended. The main objectives of these attacks are the following: enumerate the database structure, retrieve authorized data, make unauthorized modifications or deletions, etc. RFID tags could contain data for a SQL injection attack. Storage limitation is not a problem, as it is possible to do a lot of harm with a very small amount of SQL commands. For example, the SQL "drop table <*tablename*>" will delete a specified database table.

Summarizing, an RFID tag is an unsecured and untrusted data source. So the information obtained from such devices should be analyzed until there is sufficient evidence that the data is accurate. However, this is not a new concept, as in all information systems the input data should be examined to ensure that it will not cause problems.

Chapter 4

EPC Class-1 Generation-2

4.1 Introduction

There are multiple standards related to RFID technology. In this chapter, the EPC Class-1 Generation-2 is examined. This standard can be considered as the "universal" standard for Class-1 RFID tags. Class-1 RFID tags are very limited both in their computational and storage capabilities. Because of these severe restrictions, the usage of standard cryptographic primitives is not possible. However, RFID tags are susceptible to attacks also found in other technologies such as wireless, bluetooth, smart-cards, etc. Therefore, once the EPC Class-1 Generation-2 specification is explained, a security analysis will reveal its weak points. Furthermore, current proposals to enhance its security level are presented and analyzed. Finally, the chapter is finished by identifying some open research issues to increment the security of low-cost RFID tags.

The benefits of standards are clear, and assumed by almost everyone. The growth of any new technology is in many cases due in part to the establishment of open standards. Back in 2003, there was a clear lack of harmonization and major RFID vendors offered mainly proprietary systems. Fortunately, things are quickly changing. Nowadays, EPCglobal [64] and ISO [103] have joined forces to publicize and harmonize the use of RFID technology.

EPCglobal is a joint venture between EAN International and the Uniform

Code Council (UCC). EPCglobal fulfills the industry's need for an effective RFID-network standard with its EPC (Electronic Product Code) Network. Within EPCglobal, the Hardware Action Group (HAG) develops specifications for hardware components of the EPC Network, including tags and readers. The EPC system defines four RFID tag classes:

- **Class-1: Identity Tags.** Passive-backscatter tags with the following minimum features:
  - An Electronic Product Code (EPC) identifier.
  - A Tag identifier (TID).
  - A kill function that permanently disables the tag.
  - Optional password-protected access control, and optional user memory.
- **Class-2: Higher functionality tags.** Passive tags with all the aforementioned features, also including the following:
  - An extended TID.
  - Extended memory.
  - Authenticated access control.
  - Additional features (TBD) as will be defined in the Class-2 specification.
- **Class-3.** Semi-passive tags with all the aforementioned features, also including the following:
  - An integral power source.
  - An integrated sensing circuitry.
- **Class-4.** Active tags with all the aforementioned features and also including the following:
  - Tag-to-Tag communications.
  - Active communications.
  - Ad-hoc and networking capabilities.

4.2 Generation-2 vs Generation-1

One of the most important standards proposed by EPCglobal is the EPCglobal Class-1 Gen-2 RFID specification [61]. This standard was adopted by EPCglobal in 2004 and was sent to ISO. Eighteen months later (March-April'2006), it was ratified by ISO and published as an amendment to its 18000-6 standard.

These specifications provide a great advance to consolidate the adoption of RFID technology [211]. Where previously there were several specifications such as EPC Class-1 and EPC Class-0, a single UHF specification is now established. In order to ease a worldwide deployment, emerging UHF regulations in different regions have been taken into account. Additionally, the best features of the preceding specifications have been improved, and a range of future applications including higher-function sensor tags have been foreseen.

4.2.1 Read and Write Speed

Generation-1 provides a single communication speed, providing satisfactory speed and adequate robustness for most applications. On the other hand, four different communication speeds are available in Generation-2 to provide more flexibility for different operational environments. Gen-2 tags have a maximum theoretical reading speed of around 1000 tags per second (when insulated from RF noise), but in very noisy environments that speed is reduced to around 100 tags/sec. The read speed of Gen-2 is then about twice as fast as in Gen-1 in real conditions, with average read rates of around 500 tags per second.

Furthermore, Gen-2 specifies the speed at which tags can be programmed. The specification dictates that tags should be writable at a minimum rate of about 5 per second, setting 30 tag/sec as the objective value in optimum conditions.

4.2.2 Robust Tag Counting

Tags compliant with Gen-1 specification are singulated by means of binary tree walking protocols with persistent sleep/wake states. The Gen-2 specification is based on the principle that tags experiencing brief moments of power can be read. In particular, the Q *protocol*, which is based on simple query/acknowledgment exchanges between reader and tags, is employed. Q is a parameter that a reader uses to regulate the probability of tag response. Briefly, a reader sends a $Query$ to a tag and the tag loads a Q-bit random number (or pseudo-random number) into its slot counter. The tag responds with a random number when the value in its slot counter is fixed to zero. The reader then sends an acknowledge that includes the tag's random number, which sends back its ID (i.e. EPC). The process continues until all tags in the reader's field have been counted.

In Gen-1, tags are switched to sleep mode after they have been read, facilitating the reading of tags that have not yet been counted. To begin a new count, a wake-up message is sent to tags in order to wake them to be ready for reading. Multiple wake-up, count and sleep cycles are necessary to ensure that all tags in a reader field have been read. Gen-2 refines this process by introducing a dual state, avoiding the necessity of a wake-up command. Under this approximation, tags change their state each time they are read. As the reader counts tags in "A" state, those change automatically to "B" state, and viceversa. Gen-2 repeats counts of "A" and "B" until all tags have been identified. Therefore, these two mechanisms both allow an increase in reading speed and make sure all tags have been counted.

4.2.3 Dense Reader Operation

If there are many readers operating and querying in close proximity at the same time, this can drown out the weak responses of tags. In US, frequency hopping is used, as there are not severe bandwidth requirements in the UHF band. In Europe, the band available to RFID is relatively narrow, so this approach is not possible. Readers are required to "listen before talk"; first they determine that the channel is not already in use, only then they may start communication.

Gen-2 tries to improve its features under dense reader operation in several ways. First, the available RF bandwidth is used as efficiently as possible; only minimal data is exchanged between readers and tags. Second, Gen-2 provides new radio signaling techniques which easily allow the isolation of tag's response, even in noisy conditions. In particular, "Miller sub-carrier" or "FM0" are employed. Third, three modes of operation are available: single reader, multi-reader and dense-reader. Finally, Gen-2 verifies data as in Gen-1 specification to ensure accurate reads in noisy environments, and adds a feature that confirms when tags have been correctly written.

4.2.4 Parallel Counting

It is possible that several readers communicate simultaneously with the same tag. In this situation, a tag might change its state in the middle of another reader's query, causing the loss of the tag by the second reader. To solve this problem, tags support "sessions" allowing a single tag to communicate with two or more readers. Up to four logical sessions can be assigned to be assigned to different readers.

4.3 EPC Class-1 Generation-2 Specification

The EPC Class-1 Generation-2 specification [61], in the following named as EPC-C1G2, defines the physical and logical requirements for RFID systems operating in the 860-960 MHz frequency range. These systems are made up of two main components: interrogators, also known as readers, and tags also known as labels.

Modulating a RF signal (860-960 MHz), a reader transmits information to a tag. As tags are passive, all of their operating energy is received from reader's RF waveform. Indeed, both information and operating energy are extracted from the signal sent by the reader.

Furthermore, as tags do not have a power source, tags can only answer after a message is sent by the reader. These kind of systems are known as Interrogator-Talk-First (ITF). A reader receives information from a tag by transmitting a continuous-wave RF signal to the tag. The tag backscatters a

signal to the reader by means of the modulation of the reflection coefficient of its antenna. Communications are half-duplex, so the reader talks and the tag listens, or viceversa.

4.3.1 Physical Layer

A reader sends information to one or more tags by modulating an RF carrier using Double-Sideband Amplitude Shift Keying $(DSB-ASK)$, Single-Sideband Amplitude Shift Keying $(SSB-ASK)$ or Phase-Reversal Amplitude Shift Keying $(PS-ASK)$ using Pulse-Interval Encoding (PIE) format. Tags receive their operating power from this RF signal.

In order to receive information from a tag, readers transmit an unmodulated RF carrier and listen for a backscattered replay. Tags send information by backscatter-modulating the amplitude/phase of the RF carrier. The encoding format is either FM0 or Miller-modulated subcarrier.

4.3.2 Tag-Identification layer

A reader interacts with tags using three basic operations:

- **Select.** The operation of choosing a subset of the tag population for inventory and access. Working with databases, this operation is similar to selecting records.
- **Inventory.** The operation of identifying tags. Specifically, after the exchange of several messages (an inventory round), the tag sends to the reader the PC, EPC and a CRC-16 values. An inventory round operates in one and only one session at a time.
- **Access.** The operation of communicating (reading from and/or writing to) with a tag is comprised of multiple commands. Tags have to be unequivocally identified before access.

4.3.3 Tag Memory

Tag memory is logically separated into four banks, as illustrated in *Figure 4.1*.

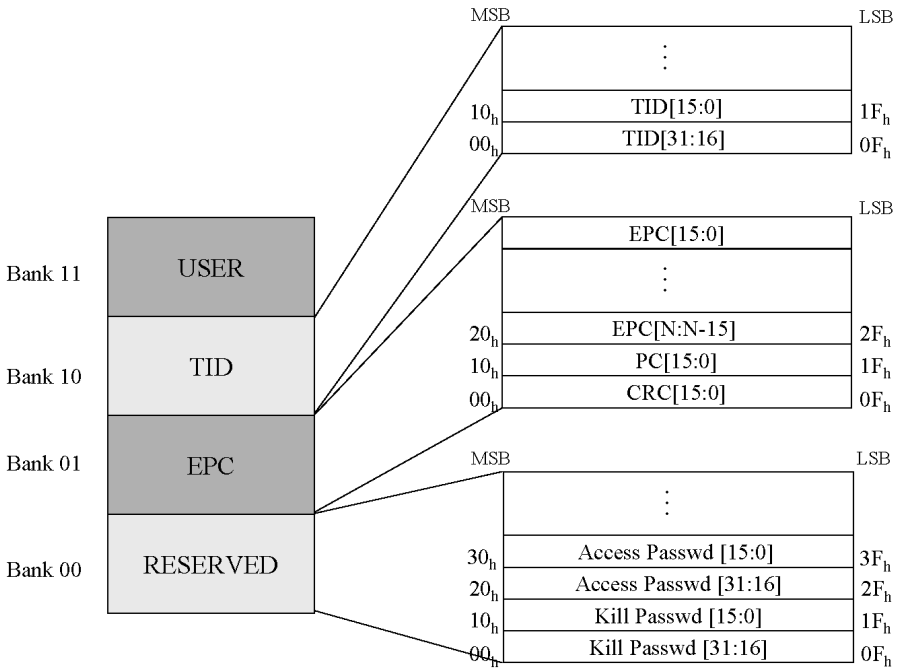


Figure 4.1: Logical Memory Map

- **Reserved memory (Bank 00).** This area of memory shall contain the kill and access passwords. Unless this memory locations have nonzero values, the kill and access commands will not be accepted. Furthermore, these locations cannot be locked or protected without invoking the access command.

  The kill and access passwords are 32-bits values. Once the tag receives the kill password, it is rendered silent thereafter. Tags with a nonzero access password have to receive this before transitioning to a secure state.

- **EPC memory (Bank 01).** This area of memory shall contain a CRC-16 checksum. Specifically, the complement-one of the precursor defined in ISO/IEC 13239 is computed. A basic integrity check is implemented by using a CRC-16 checksum of the PC and EPC values that a tag backscatters during an inventory operation.

  In the same block we can find the Protocol Control (PC) bits and a code (such as an EPC) that unequivocally identifies the object to

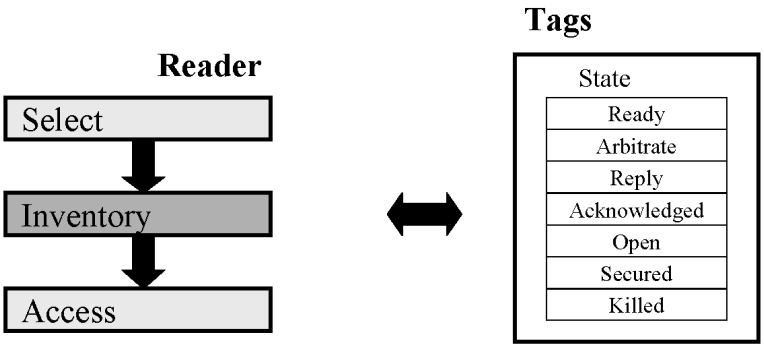


Figure 4.2: Tags State Diagram

which the tag is attached. The PC is subdivided into the EPC length field, RFU, and a Number System Identifier (NSI).

- **TID memory (Bank 10).** This area of memory shall contain an 8-bit ISO/IEC 15693 class identifier. Additionally, sufficient information to unequivocally identify the custom commands and/or optional features supported by the tag is also stored.

- **User (Bank 11).** This area of memory allows user-specific data storage. The memory organization is user-defined.

4.3.4 Tag States and Slot Counter

As defined in EPC-C1G2, tags shall implement different states, as displayed in *Figures 4.2* and *4.3*.

- **Ready state.** After being energized, a tag that is not killed shall enter a ready state. The tag shall remain in this ready state until it receives an accurate $Query$ command. Tag loads a Q-bit number from its RNG, and transitions to the arbitrate state if the number is non-zero, or to the replay state if the number is zero.

- **Arbitrate state.** A tag in an arbitrate state shall decrement its slot counter every time it receives a $QueryRep$, transitioning to the replay state and backscattering a $RN16$ when its slot counter reaches 0000_h.

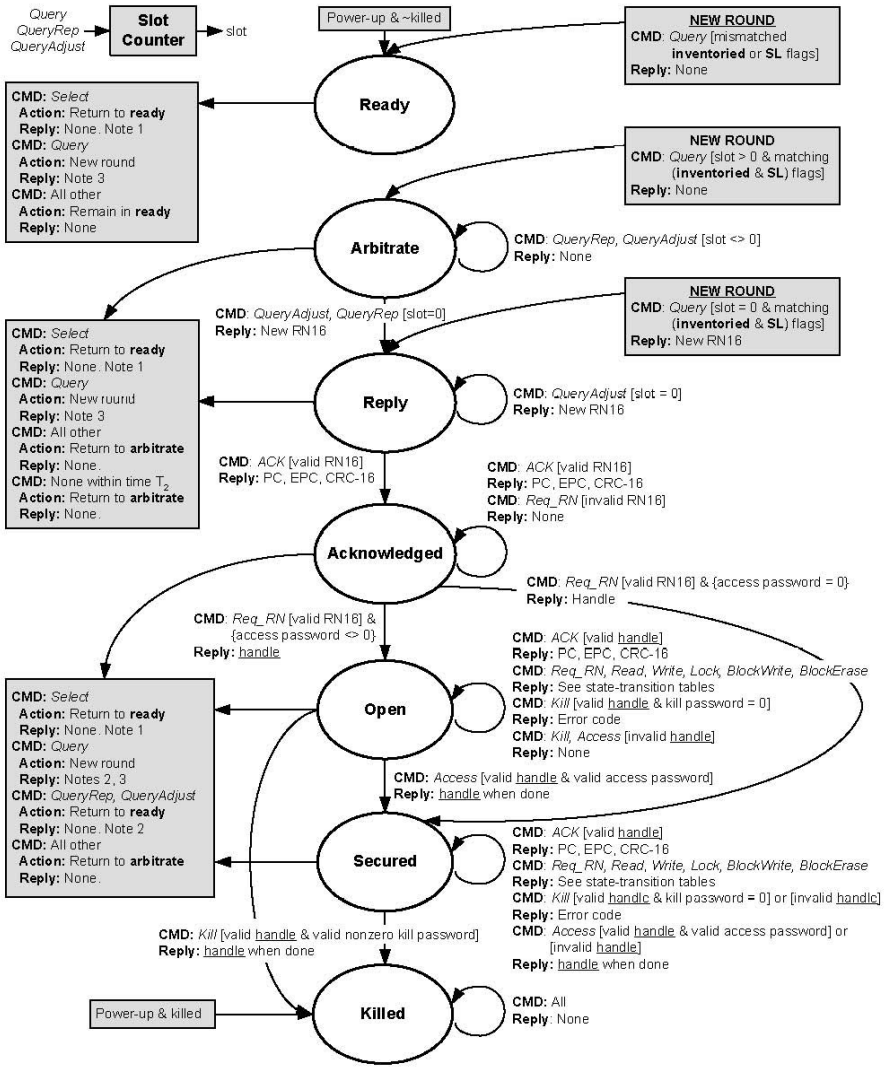


Figure 4.3: Interrogator/Tag Operations and Tag State [61]

- **Replay state.** A tag shall backscatter a $RN16$, once entering in replay state. If the tag receives a valid acknowledge (ACK), it shall transition to acknowledge state, backscattering its PC, EPC, and CRC-16. Otherwise, the tag remains in arbitrate state.

- **Acknowledge state.** A tag in acknowledge state may transition to any state except killed.

- **Open state.** After receiving a Req_RN command, a tag in acknowledge state whose access password is nonzero shall transition to open state. The tag backscatters a new $RN16$ that both reader and tag shall use in subsequence messages. Tags in an open state can execute all access commands except $Lock$ and may transition to any state except acknowledge.

- **Secured state.** A tag in acknowledge state whose access password is zero shall transition to secured state, upon receiving a Req_RN command. The tag backscatters a new $RN16$ that both reader and tag shall use in future messages. A tag in the open state whose access password is nonzero shall transition to a secured state, after receiving a valid access command, which include the same $handle$ that was previously backscattered when it transitioned from acknowledge state to the open state. Tags in secured state can execute all access commands and may transition to any state except open or acknowledge.

- **Killed state.** Once a kill password is received by a tag in either open state or secured state, it shall enter the killed state. Kill permanently disables a tag. A tag shall notify the reader that the killed operation was successful, and shall not respond to any reader thereafter.

Tags shall implement a 15-bit slot counter. Once a $Query$ or $QueryAdjust$ command is received, a tag shall load into its slot counter a value between 0 and $2^Q - 1$ obtained from tag's PRNG. Q is a integer in the range (0, 15). A $Query$ specifies Q, and a $QueryAdjust$ may modify Q from the prior $Query$.

4.3.5 Managing Tag Populations

As shown in *Figure 4.2*, tag populations are managed using three basic operations. Each of these operations comprise one or more commands. The operations are defined as follows:

- **Select.** The process that allows a reader to select a subset of the tag population for inventory and access. The select command is $Select$. A reader may use one or more $Select$ commands to select a particular tag previous to inventory.
- **Inventory.** The process that allows a reader to identify a tag. The inventory operation is started by transmitting a $Query$ command in one of the four sessions that the tag can handle. One or more tags may replay. The reader isolates a single tag response, and requests the PC, EPC, CRC-16 from the tag. An inventory round can only operate one session at a time.

  The inventory command set is comprised of the following commands: $Query$, $QueryAdjust$, $QueryRep$, ACK, $NACK$. All of these commands are mandatory.
- **Access.** The process that allows a reader to interact (read from or write to) with individual tags. Tags have to be unequivocally identified prior to access. There are multiple access commands, some of which employ a one-time-pad-based cover coding in the $R \Rightarrow T$ link. The set of access commands is composed of the following mandatory commands: Req_RN, $Read$, $Write$ and $Lock$. Additionally, there are other optional commands described in the specification: $Access$, $Blockwrite$ and $BlockErase$.

4.4 Pseudo-Random Number Generators

According to EPC-C1G2, tags shall be able to generate 16-bit random or pseudo-random numbers (RN16), and shall have the ability to extract Q-bit subsets from a RN16 to preload into its slot counter. Additionally, tags should be able to temporally store at least two RN16s while powered, for

example a handle and a 16-bit cover-code during password transactions. The generator (RN16) should meet the following randomness criteria:

- **Probability of a single RN16:** The probability that any RN16 drawn from the RNG has value RN16 = j for any j, shall be bounded by $0.8/2^{16} <$ P(RN16 = j) $< 1.25/2^{16}$.
- **Probability of simultaneously identical sequences:** For a tag population of up to 10,000 tags, the probability that any of two or more tags simultaneously generate the same sequence of RN16s shall be less than 0.1%, regardless of when the tags are energized.
- **Probability of predicting an RN16:** An RN16 drawn from a tag's RNG 10ms after the end of T_r, shall not be predictable with a probability greater than 0.025% if the outcomes of prior draws from RNG, performed under identical conditions, are known.

4.5 Security Analysis and Open Issues

In this section we see in detail the messages exchanged in the different operations between tags and readers. Once the procedures have been understood, the security faults can easily be identified.

4.5.1 Inventory Procedure

A $Query$ initiates an inventory round and decides which tags participate in the round. After receiving the $Query$ tags shall pick up a random value in the range $(0, 2^Q - 1)$, and load this value into its slot counter. Tags that pick up a nonzero value shall transition to the arbitrate state and await a $QueryAdjust$ or $QueryResp$ command. Assuming that a single tag answers (slot_counter = 0), the query-response algorithm proceeds as follows:

- $T \Rightarrow R$: The tag backscatters an $RN16$ and enters in replay state.
- $R \Rightarrow T$: The reader acknowledges the tag with an ACK containing the same $RN16$.

- $T \Rightarrow R$: The acknowledged tag transitions to the acknowledged state, and backscatters its PC, EPC, and CRC-16
- $R \Rightarrow T$: The reader sends a $QueryAdjust$ or $QueryRep$, so the identified tag inverts its inventory flag (i.e $A \rightarrow B$ or $B \rightarrow A$) and transitions to the ready state.

The security of this procedure is almost non-existent. Imagine the following scenario, where a passive eavesdropper is listening in the channel. Under these simple and realistic conditions the following security drawbacks are presented:

- Tags do not transmit the EPC in a secure way. Instead, the EPC is transmitted in plain-text. So one of the most important concerns about the use of RFID technology is not accomplished. In other words, the privacy information of the tag is easily jeopardized.
- Every time a tag is interrogated, it always transmits the same EPC. Due to the fact that the EPC transmitted by a tag is fixed, a tag may be associated with its holder, allowing its tracking. Suppose that your watch has a tag compliant with the EPC-C1G2. An attacker may place readers in your favorite entertainment places: cinemas, pubs, shops, etc. During two months the attacker's readers store the day and hours when you stay at these places. Then, the attacker collects all this information, obtaining a consumer profile of you, which is very valuable information for a great number of companies. This is only one example of how privacy location may be compromised. Therefore, another of the main citizen concerns about the implantation of RFID technology is not guaranteed, either.

The above scenario shows how privacy and location privacy are not guaranteed even under a very weak attack scenario, where there is only a passive eavesdropper. To avoid these two connected problems, researchers proposed the use of pseudonyms. In general terms, a pseudonym is a fictitious name. In RFID context, a pseudonym is interpreted as an anonymized static identifier. However, under these conditions only privacy is guaranteed, as no private information is passed on the channel. An additional

requirement is needed to prevent an attacker from being able to track a holder's tag. Specifically, every time the tag is interrogated, the tag has to transmit a fresh new pseudonym.

The most commonly-found solution in the literature consists on repeatedly applying a hash function to the static identifier (i.e $pseudonym_n = hash^n(EPC)$). Since the work of Sarma [193] in 2002, there has been a huge number of solutions based on this idea [49, 93, 136, 160, 228]. Other authors have even proposed using both hash functions and pseudo-random number generators [59, 183].

The usage of pseudonyms may be a good and interesting solution. However, hash functions have not been ratified by the EPC-C1G2 specification. As we have mentioned earlier, standard hash functions greatly exceed the capabilities of low-cost RFID tags. Therefore, an interesting issue may be to design efficient and lightweight hash functions for RFID environments.

4.5.2 Access Procedures

After acknowledging a tag, a reader may want to access the tag. The access command set comprises Req_RN, $Read$, $Write$, $Kill$, $Lock$, $Access$, $Blockwrite$ and $Blockerase$. The above commands can be computed under the following conditions:

Command	State
Req_RN	Acknowledged, Open or Secured
$Read$	Secured
$Write$	Secured
$BlockWrite$	Secured
$BlockErase$	Secured
$Acess$	Open or Secured
$Kill$	Open or Secured
$Lock$	Secured

Note that $Read$, $Write$, $Blockwrite$ and $BlockErase$ commands may also be executed from the open state when allowed by the lock status of the memory location.

A reader starts an access to a tag in the acknowledged state as set out bellow:

- $R \Rightarrow T$: The reader sends a Req_RN to the acknowledged tag.
- $T \Rightarrow R$: The tag generates and stores a new $RN16$ (denoted $handle$). Then, the tag backscatters the $handle$, and transitions to open state if its access password is nonzero, or to the secured state if its access password is zero.
- R: Now or at some later point, the reader may start further access commands.

All access commands sent to a tag (in open or secured state) include the $handle$ as a parameter. The tag shall verify the handle every time an access command is received. Access commands with an incorrect handle are ignored. The tag's answers should also include the $handle$. This value is fixed for the entire duration of an access sequence.

Tags and readers can communicate indefinitely in the open and secured states. If the reader wants to terminate communication, one of the followings messages would have to be sent: $Query$, $QueryAdjust$, $QueryRep$, or NAK.

4.5.2.1 Write, Kill and Access Commands

Everytime a $Write$, $Kill$ or $Access$ command is sent to a tag, a 16-bit word (either data or half-password) is transmitted over the channel. As confidential information is sent, the reader "obscures" it by means of a cover-code. In general terms, to cover-code data or a password, the reader first requests a random number from the tag. The reader computes a bit-wise xor of the word with this pseudorandom number, and transmits the cover-code string to the tag. Finally, the tag uncovers the received word by performing a bitwise XOR with the original random number. Specifically, as described in the specification, the following sequence of messages are exchanged:

- $R \Rightarrow T$: The reader sends a Req_RN to the acknowledged tag.

- $T \Rightarrow R$: The tag generates a new $RN16$ and backscatters it to the reader.
- $R \Rightarrow T$: The reader computes a 16-bit cipher-text, which is composed of the bitwise XOR of the 16-bit word to be transmitted with the new RN16. The reader sends a command to the reader which includes as a parameter the cipher-text and the $handle$.
- $T \Rightarrow R$: The tag decrypts the received cipher-text performing a bitwise XOR of it with the original RN16.

Kill and access passwords are 32-bits length. Therefore, to kill or access a tag, a reader shall follow a multi-step procedure: the first containing the 16 MSB of the kill or access password xored with a RN16, and the second containing the 16 LSB of the kill or access password xored with a different $RN16$.

The security margin of a protocol using a 16-bit PRNG is usually bounded by $\frac{1}{2^{16}}$. Furthermore, the access and kill password are 32-bit values. The use of 32-bits random numbers would avoid the multi-step procedure for using the access, kill and write command. We recommend the usage of 32-bits PRNG, which increases the security level and provides greater flexibility. Therefore, the design of lightweight 32-bits PRNG can be considered as an interesting open issue in RFID security. This PRNG has to obey the randomness criteria specified in the EPC-G1G2 standard [61] and fits in the severe computational and storing requirements for low-cost RFID tags.

The RFID channel asymmetry does not prevent the listening of the backward channel (tag-to-reader channel) by an attacker. Indeed, the security of the above mechanisms can be easily jeopardized even by a passive eavesdropper. In this scenario, the attacker can obtain the random number ($RN16$) sent by the tag (answer to Req_RN message). Once obtained, the attacker can decrypt the message sent by the reader using a simple XOR. Therefore the access password, kill password or sent data, can be easily acquired by means of listening to the channel and performing a bitwise XOR. Summarizing, cover-coding does not provide any kind of security protection for data or passwords.

A naive solution to protect data and passwords would be the use of conventional encryption. However, standard cryptographic ciphers lie beyond the capabilities of low-cost RFID tags. So, an open issue is the design of new secure ciphers conforming to the severe restrictions of these environments. Another possible solution may consist of more sophisticated challenge-response protocols such as the Gossamer protocol proposed in this thesis (see *Chapter 6*).

4.5.2.2 Read Command

Read allows a reader to read part or all of tag's Reserved, EPC, TID or User memory. A read command has the following fields:

MemoryBank	Specifies whether the reader accesses Reserved, EPC, TID or User access memory.
WordPtr	Specifies the starting word address for the memory read, where words are 16-bit in length.
WordCount	Specifies the number of 16-bit words to be read.

The read command also includes tag's handle and a CRC-16. Before receiving a $Read$ command, the tag first verifies that all memory words exist and none are read-locked. Second, the tag backscatters a header (a 0-bit), the requested memory words, and its $handle$.

The information exchanged between readers and tags is transmitted in plain-text. So, if the tag requests access to private information, it could be obtained by an attacker listening to the channel. In the case that the access password was equal to zero, the $Reading$ command could be accomplished from open state. This will imply that an adversary may gain access to the tag's data without prior authentication. Once the tag is in open state, an attacker may send messages to the tag trying to obtain all its stored information. To avoid this kind of attack, we recommend locking the memory when the access password is fixed to zero, in order to discard $Read$ commands.

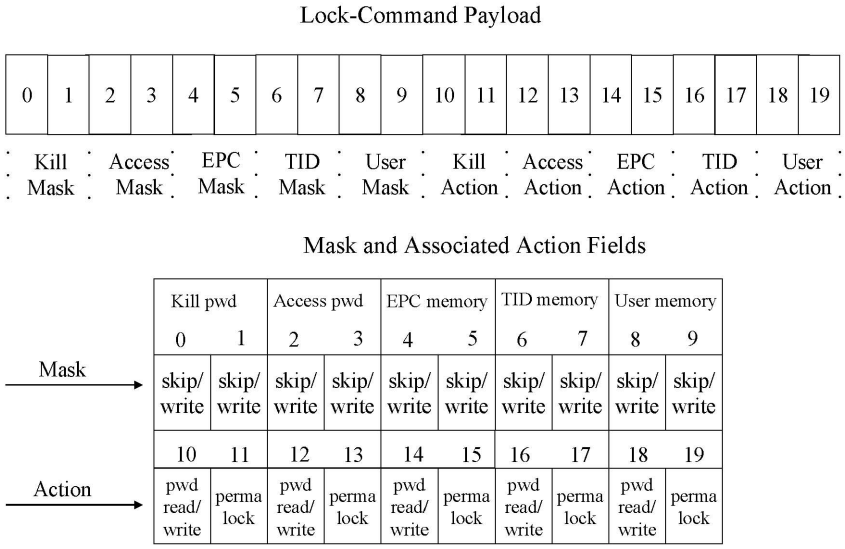


Figure 4.4: Lock-Command Payload and Masks

4.5.2.3 Lock Command

Only tags in the secured state shall execute a *Lock* command. Using the *Lock* command the reader is able to:

- Lock individual passwords, thereby preventing or allowing subsequent reads and/or writes of that password.
- Lock individual memory banks, therefore preventing or allowing subsequent writes to that bank.
- Permalock (make permanently unchangeable) the lock status for a password or memory bank. Permalock bits, once asserted, cannot be deasserted. If a tag receives a *Lock* whose payload attempts to deassert a previously asserted permalock bit, a tag will ignore the command and backscatter an error code.

As displayed in *Figure 4.4*, *Locks* contains a 20-bit payload defined as follows:

- The first 10 payload bits are *Mask* bits, which are interpreted by tags as follows:

  - $Mask = 0$: Ignore the associated action field and retain the current lock setting.
  - $Mask = 1$: Implement the associated action field and overwrite the current lock setting.

- The last 10 payload bits are $Actions$ bits, which are interpreted by tags as follows:
  - $Action = 0$: Deassert lock for the associated memory location.
  - $Action = 1$: Assert lock or permalock for the associated memory location.

Specifically, the functionality of various actions fields is described below:

pwd-write	**permalock**	**Description**
0	0	Associated memory bank is writeable from either the open or secured states.
0	1	Associated memory back is permanently writeable from either the open or secured states and may never be locked.
1	0	Associated memory back is writeable from the secured state but not from the open state.
1	1	Associated memory back is not writeable from any state.

pwd-read/ write	**permalock**	**Description**
0	0	Associated password location is readable and writeable from either the open or secured states.
0	1	Associated password location is permanently readable and writeable from either the open or secured states and may never be locked.
1	0	Associated password location is readable and writeable from the secured state but not from the open state.

1	1	Associated password location is not readable or writeable from any state.

The security of the $Lock$ command lies in the security of the access password, due to the fact that $Lock$ command can be only executed when the tag is in the secured state. For example, a DoS attack can be accomplished once the access password is obtained. Once the tag is in secure state, the attacker can send a $Lock$ command, whose payload is $FFFFF_h$. This command will entail the permanent lock of the memory, preventing the tag's functionality. Note that the permanent lock is an irreversible action.

4.5.2.4 BlockWrite and BlockErase Commands

As optional commands, readers and tags may implement $BlockWrite$ and $BlockErase$. These commands allow a reader/tag to write/read multiple words in tag's Reserved, EPC, TID, or User memory using a single command. These commands have the following fields:

MemoryBank	Specifies whether the $BlockWrite/BlockErase$ accesses Reserved, EPC, TID or User access memory. Both commands shall apply to a single memory bank. Successive $BlockWrite/BlockErase$ may apply to different banks.
WordPtr	Specifies the starting word address for the memory write/read, where words are 16-bit length.
WordCount	Specifies the number of 16-bit words to be written/read.
Data	Only employed in the $BlockWrite$ command. It contains the 16-bit words to be written, and shall be $16 \times WordCount$ bits in length.

The security of the $BlockWrite$ command is also non-existent. Tags send information (16-bit words) over the channel in plain text. Therefore if private information passes on the channel, it may be captured by a passive attacker listening to the channel. Similarly, an active attacker can easily modify messages.

The $BlockErase$ command is not encrypted upon transmission over the channel. As the information is transmitted in clear text, an attacker may be able to easily modify the messages. For example, an attacker may send the following message: 0xC8 - 01 - EPC - 0x00 - 0x8 - $handle$. Once the tag receives the previous message, the EPC memory will be erased, leaving a non-operative tag.

In the case that the access password was equal to zero, the $Blockwrite$ and $BlockErase$ command can be accomplished from open state. So, in this situation, an adversary may acquire or erase the tag's data without previous authentication. We recommend deactivating these two commands when access password is fixed to zero. In fact, a more secure option will be to limit the usage of these two commands, independently of whether a tag is in secured or open state, only when non-sensitive information is concerned.

4.6 EPC Class-1 Generation-2$^+$

The vast majority of studies on the design of security protocols for RFID either do not conform to the EPC-C1G2 specification or suffer from serious security flaws. In this section, we present some recent proposals that try to raise the security level of low-cost RFID tags and were developed "conforming" to EPC-C1G2.

4.6.1 Strengthening EPC Tags

In [109], Juels claims that EPC tags which do not have any explicit authentication functionality are vulnerable to cloning attacks. Tags emit its EPC in a promiscuous mode and readers accept the validity of the EPC at face value. The result is that tags compliant with EPC-C1G2 are vulnerable to elementary cloning attacks. Juels introduces the concept of skimming, the process of scanning a tag and obtaining its EPC for the purpose of cloning.

Several algorithms were proposed, distinguishing between two types of tags. A basic tag is one that carries only the mandatory features of the EPC-C1G2 specification. An enhanced EPC tag additionally includes the optional access-control function.

We now describe the $BasicTagAuth$ protocol and the $EnhancedTagAuth$ protocol. In a system with N tags, the integer i (with $1 \leqslant i \leqslant N$) denotes the unique index of an EPC tag. T_i and K_i denotes the EPC identifier and the valid kill PIN of tag i respectively. PIN-test(K) denotes an EPC-tag command that causes a tag to output a bit-response b: 0 if the kill PIN is correct, and 1 otherwise.

4.6.1.1 Basic TagAuth Protocol

T:	$T \leftarrow T_i$
$T \Rightarrow R$:	T
R:	if $T = T_x$ for some $1 \leq x \leq N$ the $i \leftarrow x$
	else output "unknown tag" and halt
R:	$(j,\{P_i^{(1)}, P_i^{(2)},...,P_i^{(q)}\}) \leftarrow GeneratePINSet(i)[q]$
	$M \leftarrow$ valid
	for $n = 1$ to q do
$R \Rightarrow T$:	$PIN\text{-}test(P_i^{(n)})$
$T \Rightarrow R$:	b
	if $b == 1$ and $n \neq j$ then $M \leftarrow$ invalid
	if $b == 0$ and $n = j$ then $M \leftarrow$ invalid
R:	output M;

The key idea of this protocol is the presentation of spurious PINs as a means of testing their authenticity. The q value is a security parameter that specifies the number of spurious PINs to be generated.

An attacker that performs skimming attacks can only create a cloned device that attempts to guess the correct PIN-trial j uniformly at random. The probability of successful attack (the cloned tag appearing to be valid) is clearly just $1/q$. However, the protocol is very inefficient (time-consuming) as large number of messages (q) are exchanged between tags and readers to provide an adequate security level.

4.6.1.2 Enhaced TagAuth Protocol

Enhanced EPC tags have both access and kill PIN. Juels proposes a mutual authentication protocol, in which the access PIN (A_i) serves to authenticate

the reader and the kill PIN (K_i) in turn serves to authenticate the tag.

T:	$T \leftarrow T_i$
$T \Rightarrow R$:	T
R:	if $T = T_x$ for some $1 \leq x \leq N$ the $i \leftarrow x$, $A \leftarrow A_i$
	else output "unknown tag" and halt
$R \Rightarrow T$:	A
T :	if $A = A_i$ then $K \leftarrow K_i$
	else $K \leftarrow \phi$
$T \Rightarrow R$:	K
T :	if $K = K_i$ then output "valid"
	else output "invalid"

Both of Juels's proposals prevent a cloned tag from impersonating legitimate EPC-C1G2 tags. Although this problem constitutes a very interesting issue, we believe that focusing only on cloning, and forgetting other important problems such as privacy, tracking, denial of service, etc. should not be the correct approach.

4.6.2 Shoehorning Security into the EPC Standard

Juels et al. examine various ways by means of which RFID tags might perform cryptographic functionality while remain compliant with EPC-C1G2 standard [27]. Its key idea resides in taking an expansive view of EPC tag memory. Instead of considering this memory merely as a form of storage, they use it as an input/output medium for interfacing with a cryptographic module within the tag. Therefore, read/write commands may carry out cryptographic values associated, such as messages in a challenge-response protocol.

Juels et al. claim that EPC-C1G2 is a very limited protocol for entity authentication. Specifically, tags are not authenticated in the specification, which facilitates counterfeiting. As an example, this simple challenge-response protocol is presented:

- $R \Rightarrow T$: C_R
- $T \Rightarrow R$: EPC, R_T

$R_T = H(K_{TS}, C_R)$, and $H()$ is a cryptographic primitive like a block-cipher, K_{TS} is some secret key known only to the tag and the reader, and C_R is a unique challenge.

To implement the previous protocol, commands designed for other purposes will have to be reused or one should define custom or new commands. The task of defining the use of one protocol to carry the data units of another is often designed as protocol convergence. Concisely, the ISO 7816 command set to accomplish entity authentication of the tag is proposed.

Juels et al.'s work signals the need for mutual authentication between tags and readers. However, the proposal is based on the assumption that EPC-C1G2 tags might support on-board cryptographic modules. We consider that this is not realistic, at least at the present time. Moreover, this proposal is focused on tag authentication (counterfeiting). However, there are a great number of other security concerns that should be considered before proposing a protocol converge. Otherwise, Juels et al.'s proposal can be a good starting point.

4.6.3 Enhancing Security of EPC-C1G2

Duc et al. proposed a tag-to-backend server authentication protocol [158]. During manufacturing time, EPC, and tag's access PIN are assigned by the manufacturer. Then, it chooses a random seed and stores $K_1 = f(seed)$ in the tag's memory and the corresponding back-end server's database entry. The authentication protocols is described as follows, where $f()$ denotes a PRNG function.

$R \Rightarrow T$:	Query request
T:	Compute $M_1 = CRC(EPC\|\|r) \oplus K_i$ and $C = CRC(M_1 \oplus r)$ where r is a nonce.
$T \Rightarrow R \Leftrightarrow S$:	M_1, C and r

S:	From each tuple (EPC,K_i) in its database, the server verifies whether the equation $M_1 \oplus K_i = CRC(EPC\|\|r)$ and $C = CRC(M1 \oplus r)$ hold. If a match is found, then the tag is successfully identified and authenticated, and the server will forward tag's information to the reader and proceed to the next step; otherwise the process is stopped.
$S \Leftrightarrow R \rightarrow T$	M_2 If R desires to perform read/write operation to tag's memory, it requests an authentication token to M_2 from S where $M_2 = CRC(EPC\|\|PIN\|\|r) \oplus K_i$. Then, R sends M_2 to T. The tag receives M_2 and computes its M_2, using its local values (PIN, r, EPC, K_i), and verifies whether the received M_2 equals the local one. If so, the tag will accept the end session command in the next step.
$S \leftarrow R \rightarrow$ T:	$EndSession$ command Upon receiving this command, both server and tag update their shared keys as $K_{i+1} = f(K_i)$.

The security of Duc et al.'s protocol greatly depends on key synchronization between tags and back-end server. The last message of the protocol is comprised of an $EndSession$ command, which is sent to both tags and readers. An interception of one of these messages will cause a synchronization loss between the tag and the server. Therefore, the tag and the server cannot authenticate each other any more. Additionally, this fault may be exploited as follows: the $EndSession$ message to server may be intercepted avoiding its key updating and then a counterfeiting tag can replay old data (M_1, r, C) leading to its correct authentication.

Furthermore, the forward secrecy is not guaranteed. Forward secrecy is the property that guarantees that the security of the messages sent today will still be valid tomorrow. If the tag is compromised, obtaining (EPC, PIN, K_i), an attacker can verify whether past communications came from the same tag. This attack is based in the symmetry of messages M_1 and M_2. Imagine that an attacker has stored old ($M1$, $M2$, r) messages. Now, he computes $M_1 \oplus M_2 = CRC(EPC \oplus r) \oplus CRC(EPC\|\|PIN\|\|r)$, and using the compromised values (EPC, PIN, K_i) and the eavesdropped r, he could verify that these messages came from the same tag.

Chapter 5

Proposed Solutions for Securing RFID Technology

In this chapter we present the best solutions proposed so far for solving the security problems and threats associated with the use of RFID systems. Our objective is not to give a detailed explanation of each solution, but to provide the reader with the fundamental principles and a critical review of every proposal, as well as the bibliography to be checked in case the reader should wish to pursue particular aspects of this subject further.

5.1 Kill Command

This solution was proposed by the Auto-ID Center [43] and EPCglobal. In this scheme, each tag has an unique password, for example of 24 bits, which is programmed at the time of manufacture. On receiving the correct password, the tag will deactivate forever.

5.2 The Faraday Cage Approach

Another way of protecting the privacy of objects labeled with RFID tags is by isolating them from any kind of electromagnetic wave. This can be done using what is known as a Faraday Cage (FC), a container made of

metal mesh or foil that cannot be penetrated by radio signals (of certain frequencies).

5.3 The Active Jamming Approach

Another way of obtaining isolation from electromagnetic waves, and an alternative to the FC approach, is by disturbing the radio channel, a method known as active jamming of RF signals. This disturbance may be accomplished with a device that actively broadcasts radio signals, so disrupting the radio channel completely, thus preventing the normal operation of RFID readers.

5.4 Blocker Tag

If more than one tag answers a query sent by a reader, it detects a collision. The most important singulation protocols are Aloha (13.56 MHz) and the tree-walking protocol (915 MHz). Juels [113] used this feature to propose a passive jamming approach based on the tree-walking singulation protocol, called blocker tag. A blocker tag simulates the full spectrum of possible serial numbers for tags. In [110], Juels and Brainard proposed a weaker privacy-protection mechanism, called soft blocking. Soft blockers simply show the privacy preferences of their owners to RFID readers.

5.5 Bill of Rights

In [78], Garfinkel proposed a so-called RFID Bill of Rights that should be upheld when using RFID systems. He did not try to convert these rights into Laws, but proposed them as a framework that companies voluntarily and publicly should adopt.

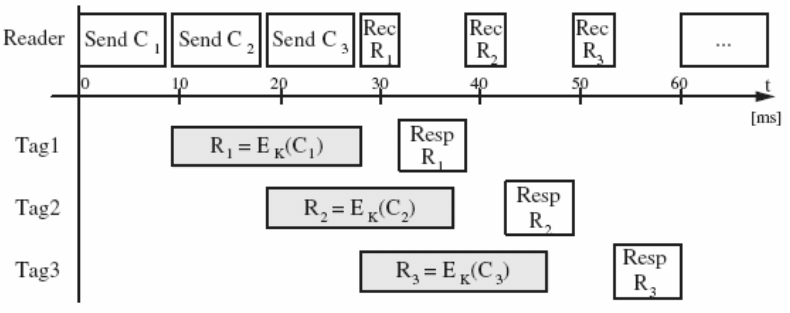


Figure 5.1: Interleaved Challenge-Response Protocol [72]

5.6 Classic Cryptography

Rewritable Memory In 2003, Kinoshita [121] proposed an anonymous-ID scheme. The fundamental idea of his proposal is to store an anonymous ID, $E(ID)$, for each tag, so that an adversary cannot find out the real ID of the tag. E may represent a public or a symmetric key encryption algorithm, or a random value linked to the tag ID. In order to solve the tracking problem, the anonymous ID stored in the tag must be renewed by re-encryption as frequently as possible.

Symmetric Key Encryption Feldhofer [72] proposed an authentication mechanism based on a simple two-way challenge-response algorithm (as illustrated in *Figure 5.1*). The main problem with this approach is that it requires AES to be implemented in an RFID tag. In the next section the main block and stream ciphers suitable for low-cost RFID systems are presented.

Public Key Encryption There are solutions that use public-key encryption, based on the cryptographic principle of re-encryption. Readers interested in the precise details can read Juels's paper [112]. Two other interesting papers that tackle the subject of re-encryption are [83] and [192]. Nowadays, other authors have proposed the use of public cryptographic based on elliptic curve cryptography (see *Section 5.8*)

5.7 Symmetric Ciphers

Block and stream ciphers are two categories of ciphers used in classic cryptography.

Block cipher A block cipher is a type of symmetric-key encryption algorithm that transforms a fixed-length block of plaintext data into a block of ciphertext data of the same length. The transformation is controlled using a second input -the secret key. The fixed-length is usually the same as the block size, 64-bits being a value normally found in many ciphers of the past. However, block size has nowadays been increased to 128-bits as processors are more powerful and attacks of complexity 2^{64} become feasible.

Techniques known as modes of operation have to be used when we encrypt messages longer than the block size. To be useful, a mode must be at least as secure and efficient as the underlying cipher.

Stream cipher A stream cipher is a type of symmetric encryption, typically much faster than block ciphers. In contrast to block ciphers that operate on a block of data, stream ciphers operate on smaller units of plaintext, usually bits or bytes. If we encrypt several times any particular plain text with a block cipher using the same key and an ECB as the mode of operation, the same ciphertext is obtained. However, with a stream cipher, the transformation of these smaller plaintext units will vary, depending on when they are encountered during the encryption process.

A stream cipher generates a keystream, in other words a sequence of bits used as a key. Encryption is generally performed by combining the keystream with the plain text, and usually the bitwise XOR operation is employed. When the keystream is independent of the plaintext and ciphertext, the stream cipher is said to be synchronous. Alternatively, it can depend on the data and its encryption, in which case the stream cipher is said to be self-synchronizing.

5.7.1 AES

In 2005, Feldhofer et al. proposed a hardware implementation of the Advanced Encryption Standard (AES) which is optimized for low-resource requirements [74]. The authors claim that the proposed implementation will serve for considerable time as a reference for AES-128 implementations that support encryption and decryption including key setup. The two main objectives of this implementation are optimization of the silicon area (smallest possible footprint) and power consumption. High data throughput, on other hand, is of minor importance.

Most AES operations are byte oriented, executing efficiently on 8-bit processors. As 8-bit operations can be combined to form 32-bit operations, AES can be efficiently implemented on 32-bit processors too. However, the most common implementation found is a 128-bit architecture. With this architecture, a higher degree of parallelism is obtained, permitting higher throughput. Feldhofer et al. decided to implement AES with encryption and decryption using a fixed key size of 128 bits. The low-power requirements do not permit use of 128-bits operations and even a 32-bit architecture still exceeds the restrictions. An 8-bit architecture, as seen in *Figure 5.2*, was finally proposed. For more details, the reader is referred to the original paper. We summarize now the most relevant aspects from their implementation:

Die Size The core occupies a silicon area of 0.25 mm^2 on a 0.35μm CMOS. This compares roughly to 3.4K gate equivalents, or to the size of a grain of sand.

Performance The functionality of cheap devices, even at very low supply voltages, has been tested. The chip works correctly with a supply voltage higher than 0.65V. The encryption of one 128-bit block requires 1032 clock cycles including loading data and data reading. The maximum throughput for encryption is 9.9 Mbps. Similar values are obtained for decryption.

Power Consumption A charge transfer method has been employed to measure the power consumption on the AES chip. The mean current consumption of the chip measured is 3.0 μA at the target clock

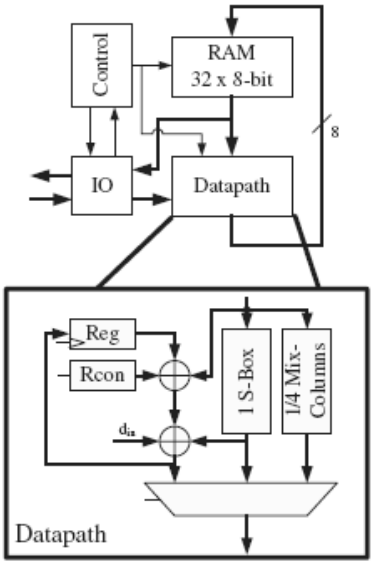


Figure 5.2: Architecture of the 8-bit AES module [74]

Table 5.1: AES-128: Performance Comparison

AES-128 version	Technology (μm)	Area (GEs)	Throughput (Mbps)	Max. Frequency (MHz)	Power (μW)
Feldhofer [74]	0.35	3400	9.9	80	4.5
Satoh [194]	0.11	5400	311	130	—
Mangard [177]	0.6	7000	70	50	—

frequency of 100 KHz and a supply voltage of 1.5V.

Finally, we can compare the Feldhofer et al. proposal with other efficient AES implementations. *Table 5.1* summarizes the specifications of these proposals. The Feldhofer proposal requires the lower chip area for its implementation, with a reduction of at least 40%. From a throughput perspective, Satoh [194] and Mangard [177] are superior. However, 10Mbps may be enough for most of the intended applications. Finally, the Satoh and Mangard proposals do not include power consumption results since they did not manufacture their design in silicon.

5.7.2 DES and its Variants

In [176] Poschmann et al. proposed a serialized version of DES which processes 4-bit and 6-bit data words instead of 32 bit or 48 bits (see the original paper for details). This implementation requires only 2,310 gate equivalents, and 144 clock cycles are consumed to encrypt a plain-text. The security of this cipher is limited by the use of a 56-bit key. A brute-force attack using software takes a few months and hundreds of PCs, but only few days with a special purpose machine such as COPACOBANA [129]. So the above implementation is intended for applications demanding short-term security or when the protected contents have a relatively low value. A key-whitening technique can be employed when a higher security level is required, yielding DESX:

$$DESX_{k.k1.k2}(x) \quad = \quad k2 \oplus DES_k(k1 \oplus x)$$

The key space increases from 56 bits to 184 bits. However, the security level of DESX is bounded by 118 bits due to the time-memory trade-offs [132]. This scheme demands around 14% additional extra gates because of the use of the bank of XOR gates and the additional registers.

The same authors proposed a compact version called DESL (DES Lightweight extension). DESL is based on the DES design, but the eight original S-boxes are replaced by a single new one (7 S-boxes and a multiplexer are eliminated). The S-box has been optimized allowing the cipher to be resistant against common attacks (i.e. linear and differential cryptanalysis, Davies-Murphy attack). This DES variant demands 1,850 gate equivalents, consuming 144 clock cycles to encrypt a block. Finally, authors proposed the use of key whitening when a higher security level is required. This version, called DESXL, requires 2170 equivalent gates, consuming the same number of clock cycles. *Table 5.2* summarizes the specifications of the different DES variants.

5.7.3 PRESENT

Recently, Bogdanov et al. proposed an ultra-lightweight block-cipher named PRESENT [35] and inspired by the AES finalist candidate SERPENT.

Table 5.2: DES Variants: Performance Comparison

DES version	Technology (μm)	Area (GEs)	Clock cycles	Current consumption (μA)
DESL	0.18	1,848	144	0.89
DES	0.18	2,309	144	1.19
DESX	0.18	2,629	144	—
DESXL	0.18	2,168	144	—

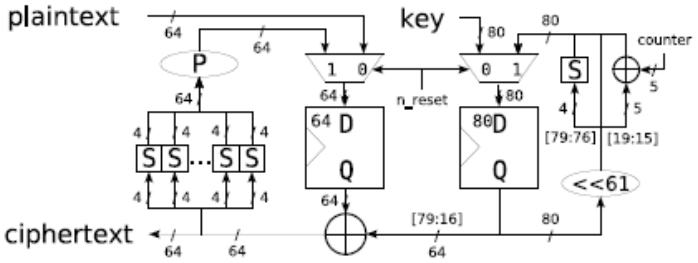


Figure 5.3: PRESENT Cipher [35]

This cipher is an example of a substitution-permutation network (SPN) and consists of 32 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. *Figure 5.3* depicts the data path of an area-optimized version of PRESENT-80. As any other SPN, the cipher comprises three stages: a key mixing step, a substitution layer, and a permutation layer. For the key mixing, authors choose a simple XOR operation. The key schedule consists basically of a 61-bit rotation together with an S-box and a round counter (PRESENT-80 uses a single S-box, whereas PRESENT-128 demands two S-boxes). The substitution layer comprises 16 S-boxes with 4-bit inputs and outputs (4 x 4). The authors recommend the version with the 80-bit key for constrained devices (i.e. low-cost RFID tags or sensor networks). The implementation of PRESENT-80 requires around 1.6K equivalent logic gates, and 32 clock cycles are needed to encrypt a 64-bit plain text (200 Kbps). PRESENT-80 is therefore more efficient than AES-128 [74] from a hardware perspective. However, from a security perspective, AES has been studied for many years and a deeper security analysis of PRESENT is required [217].

5.7.4 Other Block Ciphers

Recently, many block ciphers (i.e. HIGHT, Clefia, SEA, TEA, etc.) have been proposed for they use in RFID systems. HIGHT is block cipher with 64-bit block length and 128-bit key length [99]. This cipher requires 3048 gates on 0.25μm technology and consumes 34 clock cycles for encryption. Clefia is a new 128-bit block cipher supporting key lengths of 128, 192, and 256 bits [197]. The authors claim this cipher achieves enough resistance against known attacks and performs well both in hardware and software. The circuit of CLEFIA with 128-bit key by area optimization requires around 6,000 gates. The Scalable Encryption algorithm ($SEA_{n,b}$) is a block cipher targeted for small embedded applications [145, 201]. This cipher can be parameterized according to the plaintext size n, key size n, and the processor (or word) size b. Although it was initially designed for software implementations, its implementation in a FPGA device has been also examined [146]. Finally, the Tiny Encryption Algorithm (TEA) and its variants (XTEA and XXTEA) are optimized for software architectures [53, 54, 55]. The TEA family uses only addition, XOR and shift, so that those can be implemented efficiently on 8-bit platforms.

5.7.5 Grain

Grain [91, 92] is a stream cipher designed for restricted hardware environments and submitted to eSTREAM in 2004 by Martin Hell, Thomas Johansson and Willi Meier. It has been selected as Phase 3 Focus Candidate for Profile 2 by the eSTREAM project and finally in the eSTREAM Profile 2 (Hardware) Portafolio. This cipher is suitable for devices in which gate count, power consumption and memory are very limited.

We shall give a brief description of the Grain cipher (the reader is referred to the original paper for more details). Grain cipher is a bit oriented synchronous stream cipher. So the key stream is generated independently from the plain text. Two shift registers are employed, one with linear feedback (LFSR) and one with nonlinear feedback (NFSR). The NFSR is used in combination with a nonlinear output function to introduce nonlinearity in the cipher. In order to balance the state of the NSFR, the input to the NFSR is masked with the output of the LFSR. Both shift registers are 80 bit in size.

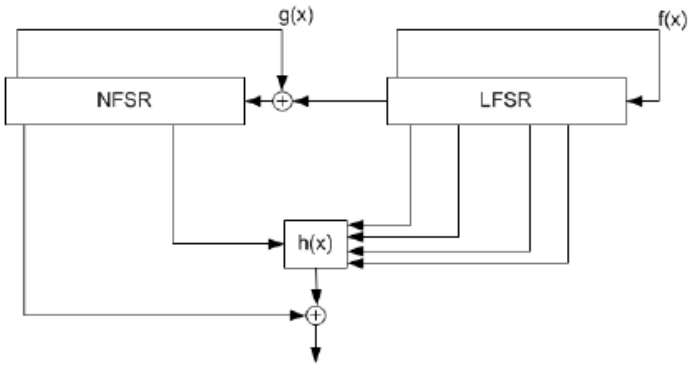


Figure 5.4: Grain Cipher [91]

Table 5.3: Gate Count and Throughput of Grain

t	Gate count	Throughput (Mbit/s)		
		MAX 3000A	MAX II	Cyclone
1	1450	49	200	282
2	1637	98.4	422	576
4	2010	196	632	872
8	2756	—	1184	1736
16	4248	—	2128	3136

The key size is 80 bits and the IV size is specified to be 64 bits. *Figure 5.4* shows the architecture of the cipher.

In the initial proposal, both shift registers are regularly clocked so the cipher will output 1 bit/clock. Johansson and Meier looked at how to increase the speed of the cipher at the expense of more hardware. Specifically, this can be done by just implementing the feedback function and the output function several times. To achieve the objective, in registers each bit is shifted t steps instead of one when the speed is increased by a factor t. An example of the architecture proposed when the speed is doubled is shown in *Figure 5.5*.

The authors studied the hardware complexity for its implementation. They elaborated a design based on standard FPGA architectures. The design has been implemented in three different FPGA families. *Table 5.3* summarizes the results obtained.

Finally, the authors claim that the Grain design provides much better secu-

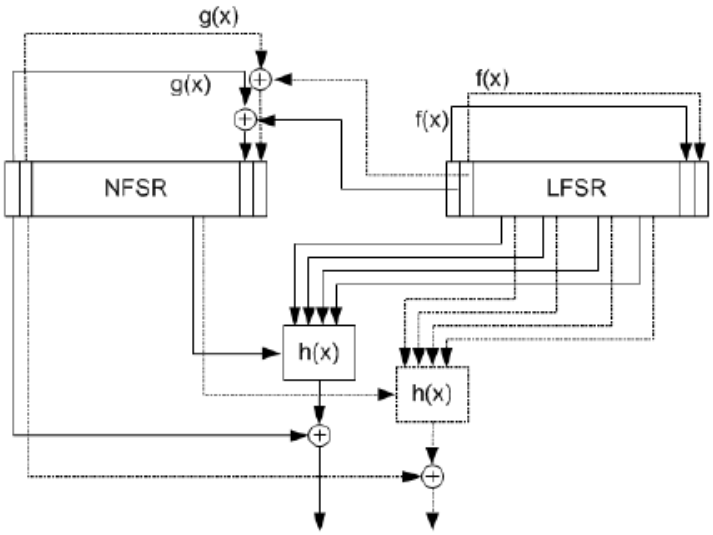


Figure 5.5: Grain cipher when at double speed [91]

rity than both E0 and A5/1 while maintaining a low gate count.

5.7.6 Trivium

Trivium is a synchronous stream cipher designed to provide a flexible trade-off between speed and gate count in hardware, and a reasonably efficient software implementation [58]. It has one of the simplest architectures of the eSTREAM candidates and is consequently particularly easy to implement. Trivium generates up to 2^{64} bits of output from an 80-bit key and an 80-bit IV. As for most stream ciphers, this process consists of two phases: first the internal state of the cipher is initialized using the key and the IV, then the state is repeatedly updated and used to generate keystream bits.

Trivium's 288-bit internal state consists of three shift registers of different lengths. The key stream generation consists of an iterative process which extracts the values of 15 specific state bits and uses them both to update 3 bits of the state and to compute 1 bit of keystream. The state bits are then rotated and the process is repeated until the requested $N < 2^{64}$ bits of keystream have been generated. *Figure 5.6* shows a graphical representation of the keystream generation. To initialize the cipher, the key and IV

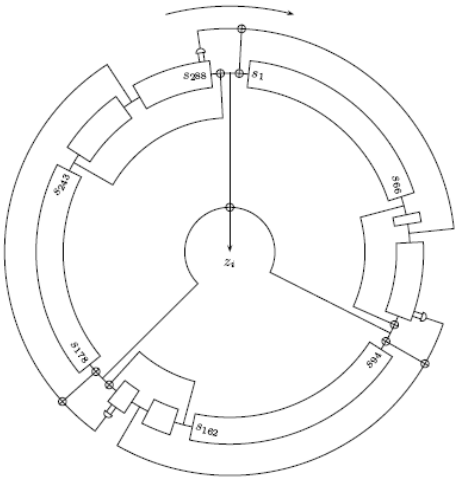


Figure 5.6: Trivium Cipher [58]

Table 5.4: Performance of Trivium

Version	Gates				Bits/cycle
Trivium	3488	(288 Flip-Flops	+ 3 AND	+ 11 XOR)	1
Trivium-8	3712	(288 Flip-Flops	+ 24 AND	+ 88 XOR)	8
Trivium-16	3968	(288 Flip-Flops	+ 48 AND	+ 176 XOR)	16
Trivium-32	4480	(288 Flip-Flops	+ 96 AND	+ 352 XOR)	32
Trivium-64	5504	(288 Flip-Flops	+ 192 AND	+ 704 XOR)	64

are written into two of the shift registers and the remaining bits are set to a fixed pattern of zeros and ones. The cipher state is then updated 4 x 288 = 1152 times in the same way as explained above, so that every bit of the internal state depends on every bit of the key and of the IV in a complex nonlinear way.

Authors suggest a bit-oriented architecture for compact implementation. Additionally, the parallelization of operations allows power-efficient and fast implementations. As any state bit is not used for at least 64 iterations after its modification, up to 64 iterations can be computed at once. So the 3 AND gates and 11 XOR gates in the original scheme are duplicated a corresponding number of times. *Table 5.4* summarizes the results obtained.

5.7.7 Other Stream Ciphers

The ECRYPT Stream Cipher Project is a multi-year effort to identify new stream ciphers that might become suitable for widespread adoption. Profile 2 is oriented to hardware applications with restricted resources. Besides Grain and Trivium, other candidates for Profile-2 are the following:

Cipher	**Authors**
DECIM	Côme Berbain, Olivier Billet, Anne Canteaut, Nicolas Courtois, Blandine Debraize, Henri Gilbert, Louis Goubin, Aline Gouget, Louis Granboulan, Cédric Lauradoux, Marine Minier, Thomas Pornin and Hervé Sibert
Edon80	Danilo Gligoroski, Smile Markovski, Ljupco Kocarev and Marjan Gusev
F-FCSR	Thierry Berger, François Arnault and Cédric Lauradoux
MICKEY	Steve Babbage and Matthew Dodd
Moustique	Joan Daemen and Paris Kitsos
Pomaranch	Cees Jansen, Tor Helleseth and Alexander Kolosha
Trivium	Christophe De Cannière and Bart Preneel

The eSTREAM portfolio finalized on May 2008 and the following ciphers have been chosen for the Profile-2: F-FCSR-H v2, Grain v1, MICKEY v2, and Trivium. The reader is referred to `http://www.ecrypt.eu.org/stream/` for more details. Additionally, hardware implementation and performance metrics of these algorithms have been assessed in detail [39, 84, 188].

5.8 Asymmetric Ciphers

The major problem related to the use of symmetric key systems is the need to share a secret key, this being required for both encryption and decryption. In asymmetric, (or public key), cryptography this is not an issue in the same way. Two keys, mathematically related, are used and work together. For example, a plain text encrypted with one of the keys can only be correctly decrypted with the other associated key. Of these two keys one will be typically be kept private by its owner and the other will be made

Table 5.5: ECC Performance Comparison

Source	Field	Total Area (GEs)	Technology (μm)	Frequency (MHz)
Kumar et al. [128]	$GF(2^{113})$	10,113	0.35	13.560
	$GF(2^{131})$	11,970	0.35	13.560
	$GF(2^{163})$	15,064	0.35	13.560
	$GF(2^{193})$	17,723	0.35	13.560
Batina et al. [30, 31]	$GF(2^{67})^2$	12,944	0.25	0.175
	$GF(2^{131})$	14,735	0.25	0.175
Gaubatz et al. [79]	$GF(p_{100})$	18,720	0.13	0.5
Wolkerstorfer [80]	$GF(p^{191})$	23,000	0.35	68.500
Öztürk et al. [57]	$GF(p_{166})$	30,333	0.13	20.000

as widely-known as possible. So there is almost no need to share keys, avoiding the risk of compromising security.

Whether a public-key cryptosystem can be implemented on an RFID tag or not remains an open problem. Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem for this kind of device. Compared to traditional cryptosystems like RSA or discrete logarithms, ECC offers equivalent security with smaller operand length, which results not only in lower computational requirements, but also power consumption and storage. ECC has been commercially accepted and recently endorsed by the US government.

In [30], the most recent works in this research area are described. Batina et al. investigated several options considering ECC over F_{2^p}, p a prime, operands ranging between 130 a 140 bits of length, and composite fields. Instead, Kumar et al. proposed ECC over binary fields ($F_{2^{131}}$ for short-term security – $F_{2^{163}}$ for medium-term security). The authors claim that the use of binary fields rather than prime fields offers two main advantages: carry free arithmetic and simplified squaring arithmetic. A performance comparison of the main contributions of ECC implementations for constrained devices is summarized in *Table 5.5*.

5.9 Schemes Based on Hash Functions

One of the more widely used proposals to solve the security problems that arise from RFID technology (privacy, tracking, etc) is the use of hash functions.

Hash Lock Scheme Weis [224] proposed a simple security scheme based on one-way hash functions. Each tag has a portion of memory reserved for storing a temporary *metaID*, and operates in either a locked or an unlocked state. The reader hashes a key *K* for each tag, and each tag holds a *metaID* (*metaID = hash(K)*). While locked, a tag answers all queries with his *metaID* and offers no other functionality. To unlock a tag, the owner queries the back-end database with the *metaID* from the tag, looks up the appropriate key and sends the key to the tag. The tag hashes the key and compares it to the stored *metaID*.

Randomized Hash Lock Scheme One of the problems of the aforementioned solution is that it allows the tracking of individuals. To avoid this, the *metaID* should be changed repeatedly in an unpredictable way. In order to solve this problem, Weis [224] proposed an extension of the hash lock scheme. It requires that tags have a hash function and a pseudo-random number generator.

Hash-Chain Scheme Ohkubo in [160] suggested a list of five points that must be satisfied in all security designs of RFID schemes: keep complete user privacy, eliminate the need for extraneous rewrites of the tag information, minimize the tag cost, eliminate the need for high power of computing units, and provide forward security. In [160], a hash-chain scheme was proposed in which two hash functions (*G* and *H*) are embedded in the tag.

The secret key s is stored in the tag's memory, which is linked to the object's ID server, which manages the link between the secret key s and the object's ID. Tag responds to reader queries by generating a hash value *a=G(s)* of secret s, computing new secret *s'=H(s)*, and overwriting the memory with new secret s'. The reader sends the output from the tag to the server and makes a request for revealing

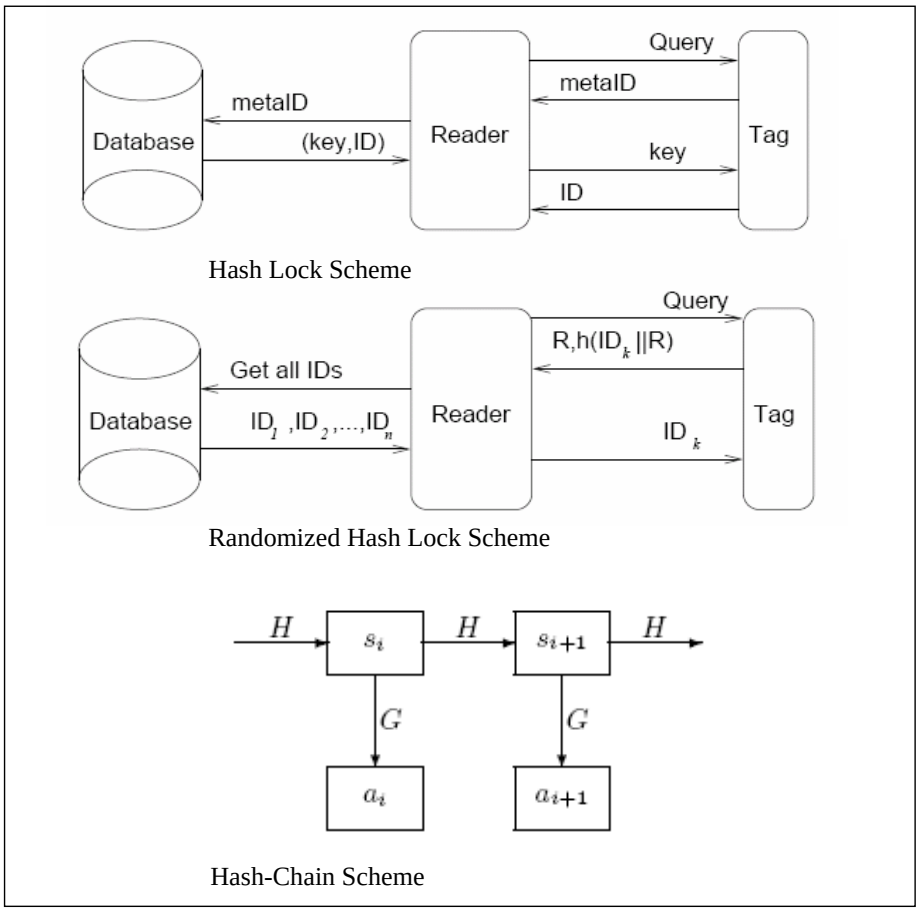


Figure 5.7: Schemes Based on Hash Functions [160, 224]

the ID. The server identifies the ID of the tag from *a=G(s)*, received from the reader, and sends the ID back to the reader.

The above three schemes are illustrated in *Figure 5.7*. Some other recent published works on the use of hash functions are [49, 93, 136, 228].

5.10 Schemes Based on Pseudo-Random Functions

Molnar [155] proposed a scheme for mutual authentication between tags and readers, with privacy for the tag. This protocol uses a shared secret s and a Pseudo-Random Function (PRF) to protect the messages exchanged

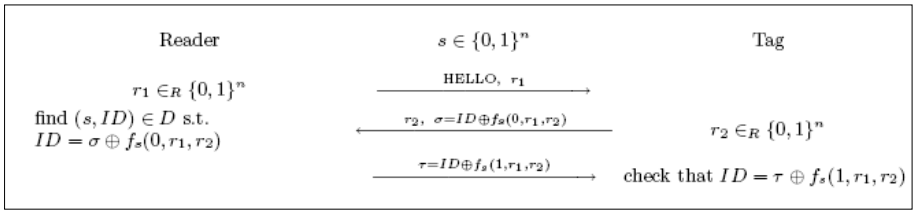


Figure 5.8: PRF-based Private Authentication Protocol [155]

between the tag and the reader. The protocol consists in a three message exchange (see *Figure 5.8*). The reader sends a *hello* message along with a random number r_1. The tag responds to the reader queries by sending another random number r_2 and σ ($\sigma = ID \oplus f_s(0, r_1, r_2)$). The reader identifies the ID of the tag, and then sends τ ($\tau = ID \oplus f_s(1, r_1, r_2)$) in order the tag to be able to authenticate the reader.

Finally, other proposals are based on the use of both hash functions and PRNGs [59, 183].

5.11 Optimization of Server Search

One of the main drawbacks of the hash schemes already proposed is that the load of the server (for identifying tags) is proportional to the number of tags. Molnar [155] proposed a new scheme to reduce this load, which is named *Tree-Based Private Authentication*. In this solution each tag is identified with a leaf of the tree. A tag stores the secrets ($\log n$) from the root to the leaf of the tree. The reader, when it wants to authenticate itself with a tag, starts at the root (left or right) and if the reader and the tag successfully authenticate using one of the secrets (left or right), the reader continues to the next level of the tree. If the reader fails to convince the tag at any level, the tag rejects the reader. If the reader passes all secrets in the path, the tag accepts the reader.

In the tree-based private authentication, the reader acts as a relay, passing the pseudonym from the tag to the Trusted Center (TC). It requires a costly interaction between the reader and the TC every time the tag is read. In order to reduce the burden on the TC, an off-line delegation scheme has

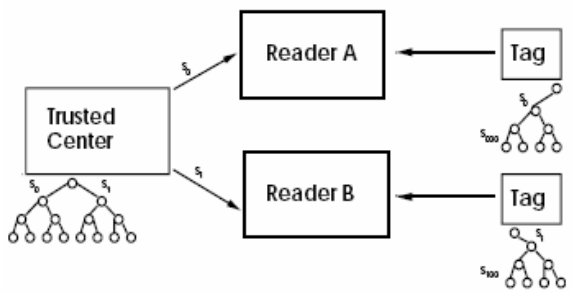


Figure 5.9: Trusted Center Delegates Access to Two Different Readers [154]

been proposed [154]. In this scheme, the TC can compute a time limit secret that only allows the ability to disambiguate pseudonyms for a particular tag a limited number of times. In particular, the TC computes a secret that allows recognizing the next q pseudonyms from this tag. We illustrate in *Figure 5.9* how off-line delegation works with two different readers and a TC.

Another interesting proposal is the work of Gildas and Oechslin [26], where a time-space trade-off is proposed.

5.12 Lightweight cryptography

5.12.1 Naïve Proposals

In 2003, Vajda and Buttyán proposed a set of lightweight challenge and response authentication protocols [215]. They started analyzing the following simple protocol:

$$\begin{aligned} R \rightarrow T &: x \oplus k = a \\ T \rightarrow R &: f(x) \oplus k = b \end{aligned} \tag{5.1}$$

where R and T symbolize the reader and the tag, respectively; k is the secret key shared between both entities; x is a n-bit random challenge; and f is a n-bit to n-bit mapping function.

The mutual information ($I(h,k) = H(x \oplus f(x))$) between the key and the exchanged messages ($h = (a,b)$) is the entropy of $x \oplus f(x)$. Bearing this in

mind, there are several ways to strengthen the security of the above protocol:

1. **Non-linearity.** The set of preimages will be more hard to compute by an attacker if a nonlinear function f is used.
2. **Mixed operations.** XOR operation (which is linear over binary vectors) could be replaced by modular addition or modular integer powering. In this case, the analysis and combination of messages would not be so easy.
3. **Compression.** Authentication does not need to be based on invertible transformations.
4. **Keys.** Different keys may be used in the two directions.

Protocol 1: XOR
A protocol based on the XOR operation and with different keys used in both directions is proposed below:

$$\begin{aligned} R \rightarrow T &: x \oplus k_1 \\ T \rightarrow R &: x \oplus k_2 \end{aligned} \tag{5.2}$$

The above protocol is probably secure if the keys are randomly selected in each run of the protocol. As low-cost RFID tags are very restricted devices, a probably secure key update algorithm is necessary. The authors then proposed a new protocol, where a lightweight block stream generator with secret seed value k^0 is employed:

$$\begin{aligned} R \rightarrow T &: x \oplus k_i \\ T \rightarrow R &: x \oplus k_0 \end{aligned} \tag{5.3}$$

where $k^i = \prod(k^{(i-1)})$, and $\prod$: $\{0,1\}^n \rightarrow \{0,1\}^n$ is a permutation, a special stream generator that expands seed k^0.

Protocol 2: Subset
The following protocol was proposed, where a tag sends back a m-bit por-

tion of the challenge as a replay:

$$\begin{aligned} R \rightarrow T &: x \oplus k \\ T \rightarrow R &: f(x) = (x_{L,x_{R[0..7]}}, x_{L,x_{R[8..15]}}, ..., x_{L,x_{R[8m..8m+7]}}) \end{aligned} \tag{5.4}$$

The challenge is divided into two parts: $x = (x_L, x_R)$. The j-th byte of x_R, denoted by $x_{R,[8j..8j+7]}$, addresses a bit of x_L, denoted by $x_{L,x_{R,[8j..8j+7]}}$, which is considered to be the j-th bit of the output vector. The following parameters are assumed: $n = 384$ ($= 256 + 128$), $|x_L| = 256$, $|x_R| = 128$ bits, and m=16. Under these conditions, the probability that an attacker will successfully impersonate a tag using a random response is bounded by $2^{-m} = 2^{-16}$. This value is unacceptable for standard cryptographic applications but it may suffice for some RFID applications.

Protocol 3: Squaring
A protocol based on the squaring of a $2n$ bit portion of the key shared between the tag and reader was proposed:

$$\begin{aligned} R \rightarrow T &: x \\ T \rightarrow R &: k_L \oplus ((k_R + x)^2 \ mod \ 2^n) \end{aligned} \tag{5.5}$$

where k_L and k_R are two halves of a $2n$ bit secret key $k = (k_L, k_R)$, and the symbol "$+$" represents integer addition.

Protocol 4: Knapsack

$$\begin{aligned} R \rightarrow T &: d \oplus k, \kappa(x, d) \\ T \rightarrow R &: x \oplus k' \end{aligned} \tag{5.6}$$

where k is an m-bit secret key and k' is an n-bit secret key, x is an n-bit challenge, and d is an m-bit trapdoor. κ is a punctured multiplicative knapsack; in order words, a public set of n s-bit prime numbers, stored both by the reader and the tag.

R selects randomly $n/2$ elements from this set (knapsack) and multiplies together the selected primes. The n-bit challenge x contains 1 at those bit positions which correspond to the primes selected (an order is assumed

among the primes) and 0 at the remaining positions. R chooses t integers randomly from the range of 1, 2, ... , $s \cdot n/2$, and marks bits of binary representation of the product at bit positions corresponding to the selected integers. The marked bits are deleted and the binary string is shrunk in size accordingly. The resultant punctured string is the output of mapping κ. Trapdoor d consists of the integers used in puncturing, by appending these integers in order. It follows that the output of κ has length $s \cdot n/2 - t$ bits, furthermore the trapdoor is $m = t \cdot \log(s \cdot n/2)$ bits long. When the tag receives the reader's message, it knows the punctured position according to the trapdoor, and the punctured bits will be found by exhaustive search.

All of these protocols assume that each tag shares a secret with the reader, so it is quite difficult for the reader to find which secret corresponds to which tag. To limit the impact of a potential discovery by an adversary, they should be changed regularly and kept secure during distribution and in service. The process of selecting, distributing and storing keys is known as key management; this is difficult to achieve reliably and securely.

The aim of the above schemes is tag authentication, but the security level is very low and can be breached by a powerful adversary. Additionally, they do not solve important problems such as reader-to-tag authentication or tracking, to name just a few issues.

5.12.2 List of Identifiers

In [107], Juels proposed a solution based on the use of pseudonyms, without using hash functions at all. The RFID tag stores a short list of random identifiers or pseudonyms (α_1, α_2, α_3, ..., α_k). When the tag is queried, it emits the next pseudonym in the list. An adversary can, however, gather all the names on the list by querying a tag multiple times. Then the fraudulent tag could impersonate a honest tag. This is the sort of cloning attack to which standard RFID tags with static identifiers are vulnerable (i.e. EPC-C1G2 specification).

To prevent such an attack, some solutions were proposed later: tags could release their name only at a certain prescribed rate, or pseudonyms could be refreshed only by authorized readers. If the second solution is employed, a mutual authentication between the reader and the tag is required.

Juels proposed a lightweight mutual authentication protocol based on the release of keys shared between both parties. The verifier authenticates to a tag by releasing a key β_i, which is unique to a pseudonym α_i. Once the verifier has authenticated to the tag, the tag authenticates itself to the reader by releasing an authentication key γ_i. Like β_i, this authentication key γ_i is unique to a pseudonym α_i. After mutual authentication, key (β_i,γ_i) and pseudonym (α_i) updating is accomplished. The reader transmits one-time padding data that the tag uses in the updating stage. Although encryption is not explicitly involved by means of one-time pads, it is equivalent to encryption. Pads can be considered keys used to "encrypt" and thereby update the α_i, β_i and γ_i values. Indeed, each tag stores a series of pads. The stored pads are updated with new material on each authentication. This new pad material is sent in clear on the channel, but the updating procedure ensures that it will be used only after a certain number (m) of updates. This number should be chosen such that an adversary cannot observe m consecutive authentications.

As it has been shown, Juels's protocol does not require the use of any cryptographic primitive. However, it involves the exchange of four messages and needs key updating, which may be costly and difficult to perform securely. Moreover, the assumption that an attacker can not observe m consecutive authentications does not hold in many real scenarios.

5.12.3 Abstractions of Integers Arithmetics

Lemieux and Tang proposed a mutual authentication scheme based on infinite, non-associative, and usually non-abelian structures that authors have termed Abstractions of Integer Arithmetic (AIA) and Partial Abstractions of Integer Arithmetic (PAIA) [138]. To give an easily understandable explanation of the proposed protocol, the scheme is introduced to the reader using integer arithmetic. However, infinite non-abelian groupoids or quasigroups should be used to achieve a secure solution (see the original paper for more details).

The proposed authentication protocol is based on the multiplication of two integers, as displayed in *Figure 5.10*. We assume that Alice and Bob share a secret number $K = k_n...k_2k_1$ and a common secret digit d. In each round, a

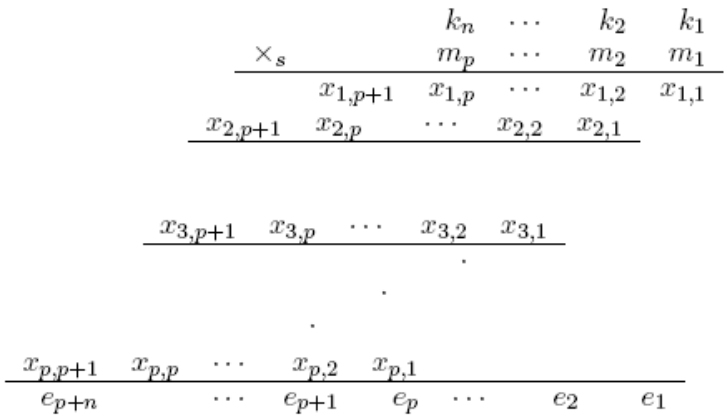


Figure 5.10: Multiplication of Two Integers

randomly digit is concatenated on the left hand side of the number M. M is initialized with the digit $m_1 = d$, which is only shared between Alice and Bob. In each round:

1. Alice randomly generates a new digit m_i, concatenates this value to $M = m_i m_{i-1} ... m_1$ and computes $E = K \times M$. Then the pair (e_i, m_i) is transmitted to Bob, where e_i is the i^{th} digit of E.

2. Bob computes the same product and checks the received e_i. If it is correct, he randomly generates m_{i+1} and repeats the process.

After r rounds of consecutive success, both parties are convinced that they share the same secret number K and secret digit d. Encryption and decryption of a message can be computed in a similar manner as authentication. In this case, m_i represents the next digit in the message that Alice wishes to send Bob. She encrypts it and the ciphertext e_i is sent to Bob. Bob decrypts e_i to recover m_i and the process continues with the next digit m_{i+1}.

The authentication and encryption/decryption schemes, based on integer arithmetic, are completely non secure. Two main assumptions are necessary to guarantee an appropriate security level. First, given numbers E and M, one cannot divide E and M even if it is know that M is a factor of E. Secondly, multiplication is not commutative. Therefore, AIAs or PIAs are required to securely implement the protocol.

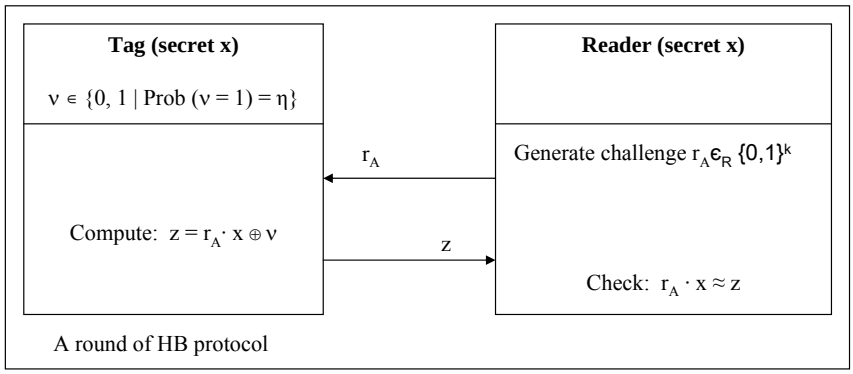
5.12.4 Human Protocols

In [222], Weis introduced the concept of human-computer authentication protocols due to Hopper and Blum, adapted to low-cost RFIDs (HB protocol). The security of the proposed protocol is rooted in the *Learning Parity with Noise Problem*, whose hardness over random instances still remains an open question.

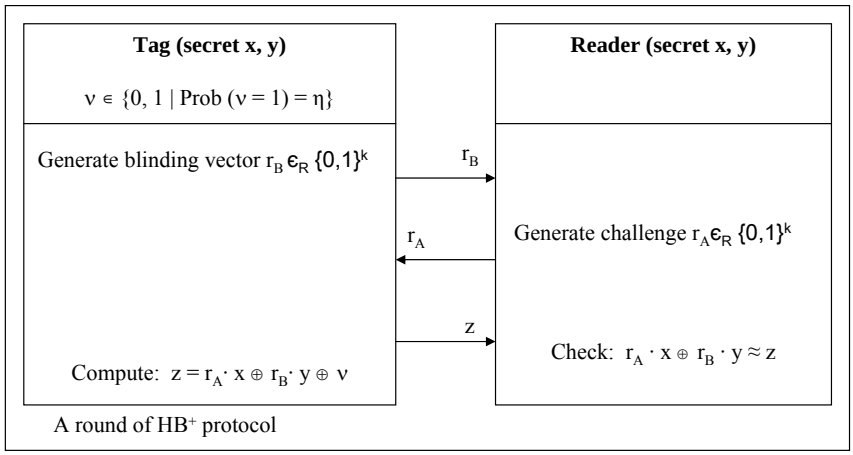
Figure 5.11 (A) illustrates a single round of the HB authentication protocol [114]. Suppose that the reader and the tag share a k-bit secret x, and the tag would like to authenticate itself to the reader. The reader selects a random challenge $a \epsilon (0,1)^k$ and sends it to the tag. The tag responds to the reader challenge by computing the binary inner-product $a \cdot x$ and injecting noise into the result. The tag intentionally sends the wrong response with probability $\eta \epsilon (0, \frac{1}{2})$. This interaction must be repeated q rounds and the reader will authenticate the tag's identity if fewer than $q\eta$ of its responses are incorrect.

The above protocol is resistant to passive attacks, but not to active attacks. Weis et al. proposed a new version of its protocol (HB^+) to offer protection against active attacks [114]. The main differences with respect to the HB protocol are the following: they introduce another k-bit secret key (y) shared between the reader and the tag. The tag and not the reader initiates the protocol, transmitting a k-bit blinding vector. Finally, z is computed as the scalar product of the newly introduced secret key (y) and the blinding vector transmitted by the tag, xored with the z in HB. A round of HB^+ is illustrated in *Figure 5.11 (B)*. Although Juels et. al claimed that HB^+ is resistant to active attacks, Gilbert et al. showed how a man-in-the-middle attack can be accomplished [81].

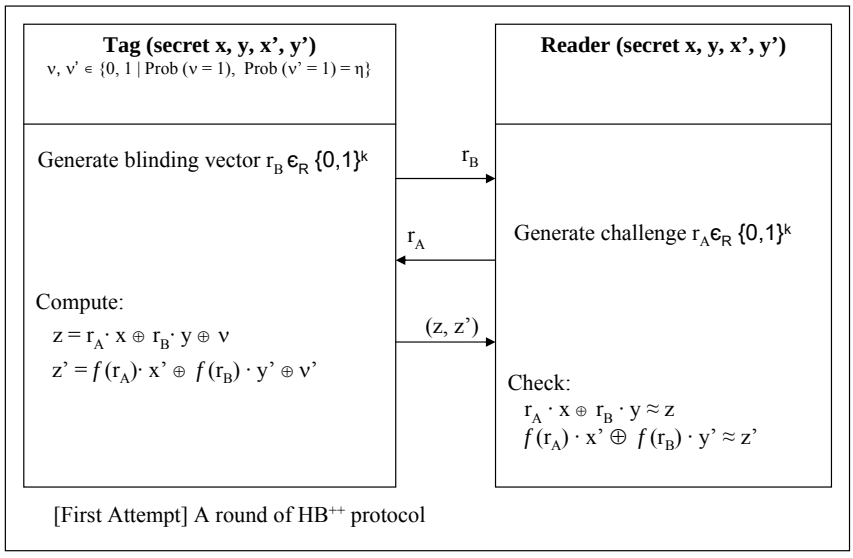
In order to avoid Gilbert et al.'s attack on HB^+, Bringer et al. [38] proposed two protocols (HB^{++}[first attempt] and HB^{++} (*Figures 5.11 (C)* and *5.12 (A)*) that protect against such man-in-the-middle attacks. However, these protocols are vulnerable to attacks from an adversary that pretends to be a genuine reader [175]. Piramuthu proposed a new protocol (see *Figure 5.12 (B)*) inspired by the HB^{++} protocol. The main changes introduced are as follows:



(a) Fig. 5.11 (A)

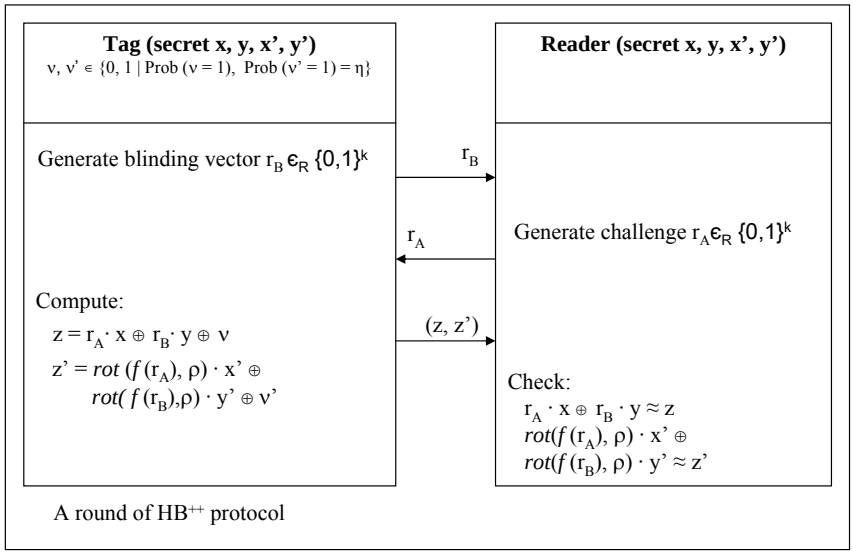


(b) Fig. 5.11 (B)

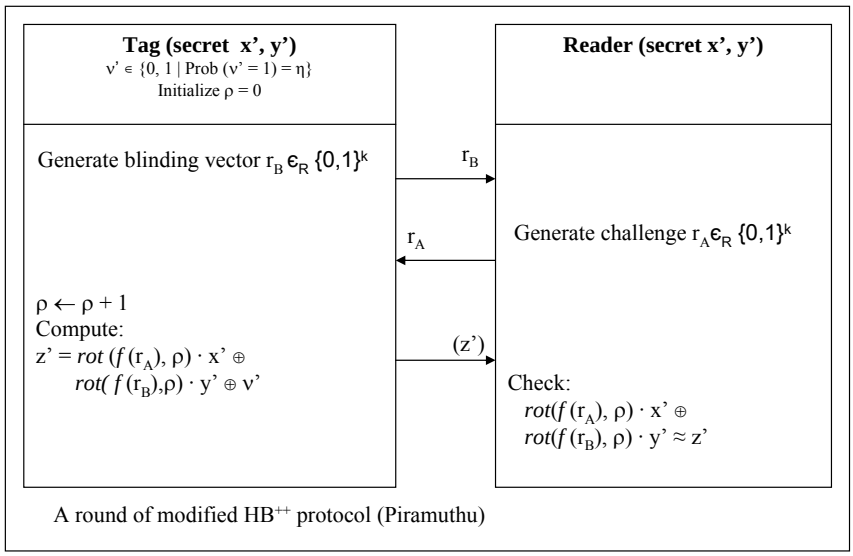


(c) Fig. 5.11 (C)

Figure 5.11: Human Protocols (I)



(a) Fig. 5.12 (A)



(b) Fig. 5.12 (B)

Figure 5.12: Human Protocols (II)

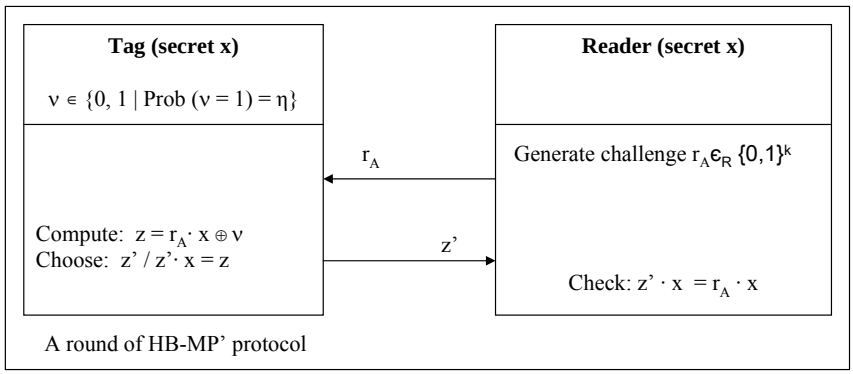
1. To thwart the attacker when an adversary pretends to be a valid reader, z and the related vectors (x, y) and ν were omitted. Additionally, the protocol is kept more lightweight.

2. In order to prevent the use of the same ρ until protocol completion, updating of ρ is accomplished every time z is computed.

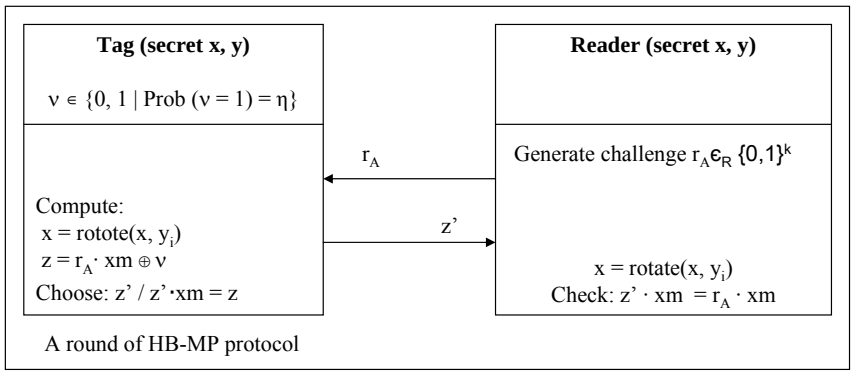
Recently, Munilla and Peinado proposed another protocol [156]. First a protocol inspired by HB and named HB-MP' was proposed (see *Figure 5.13 (A)*). The authors acknowledge that the above protocol was vulnerable to a simple man-in-the-middle attack, just like the initial HB$^+$ protocol. To avoid this weakness, a new protocol named $HB\text{-}MP$ was worked out (see *Figure 5.13 (B)*). We briefly describe a round of the $HB\text{-}MP'$ protocol: suppose that the reader and the tag share a k-bit secret x, and the tag would like to authenticate itself to the reader. The reader selects a random k-bit binary vector a and sends it to the tag. The tag computes the binary inner-product $a \cdot x$ and injects noise into this result. Then, the tag looks for a k-bit binary vector b such that $b \cdot x = z$. The tag sends back b to the reader. The reader checks the equality of $b \cdot x$ and $a \cdot x$. If it is correct, the tag is authenticated. This protocol differs slightly from the protocols based on the LPN problem. However, the authors maintain that the problem of finding x, knowing the vectors a and b, is at least at difficult as solving the LPN problem. In 2008, Leng et al. exposed a man-in-the-middle attack against HB-MP and proposed an enhanced version of the aforemention protocol, called the HB-MP$^+$ protocol [139].

5.13 Simultaneous Reading

In 2004, Juels introduced the problem of providing a proof for the simultaneous reading of two RFID tags [115]. In other words, a proof that a pair of RFID tags has been scanned simultaneously, but not necessarily by the same reading device. Concretely, Juels denominated his proof as "yoking proof" (applying "yoke" with its meaning "to join together"). *Figures 5.14* and *5.15*, summarize the proposed solutions related to the simultaneous reading problem.

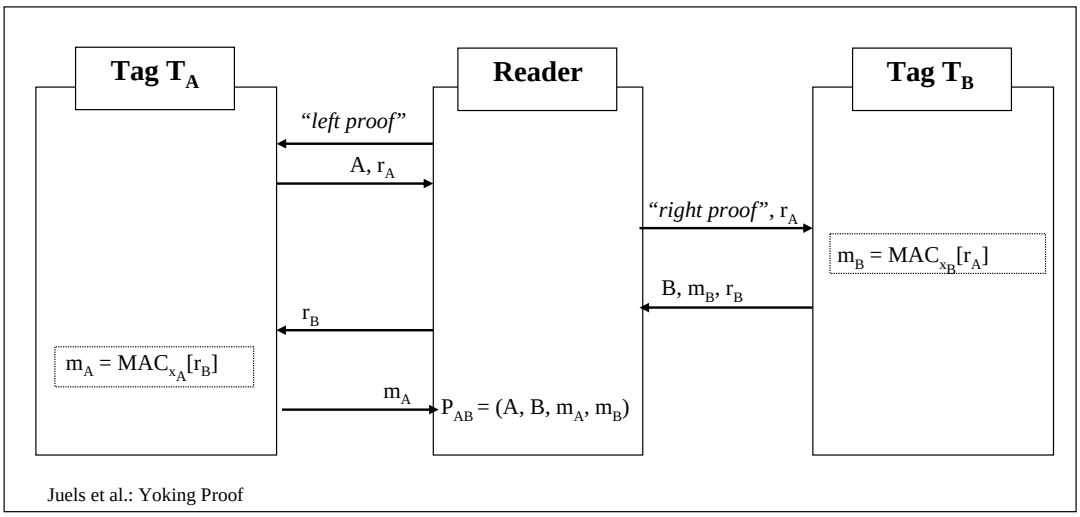


(a) Fig. 5.13 (A)

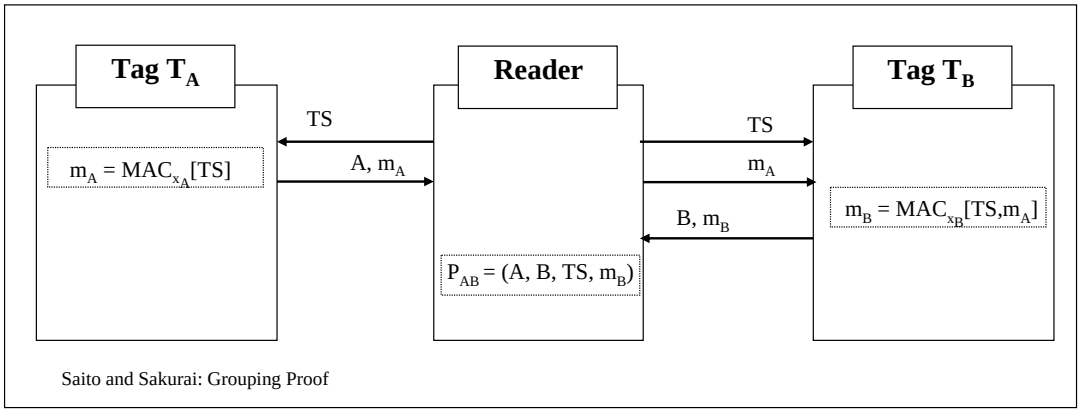


(b) Fig. 5.13 (B)

Figure 5.13: Human Protocols (III)

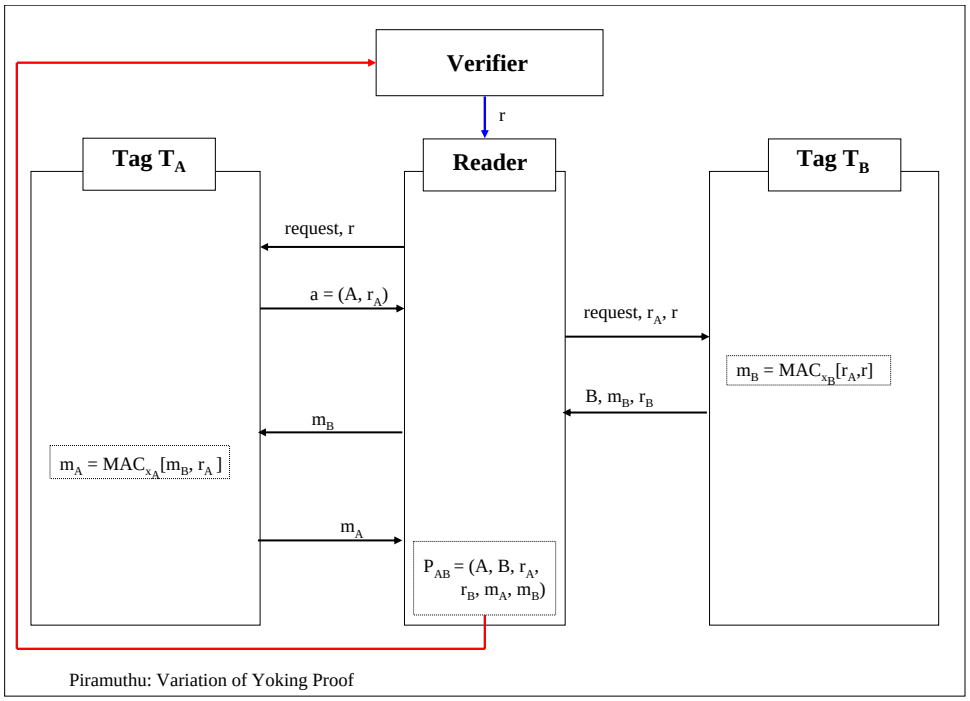


(a) Fig. 5.14 (A)

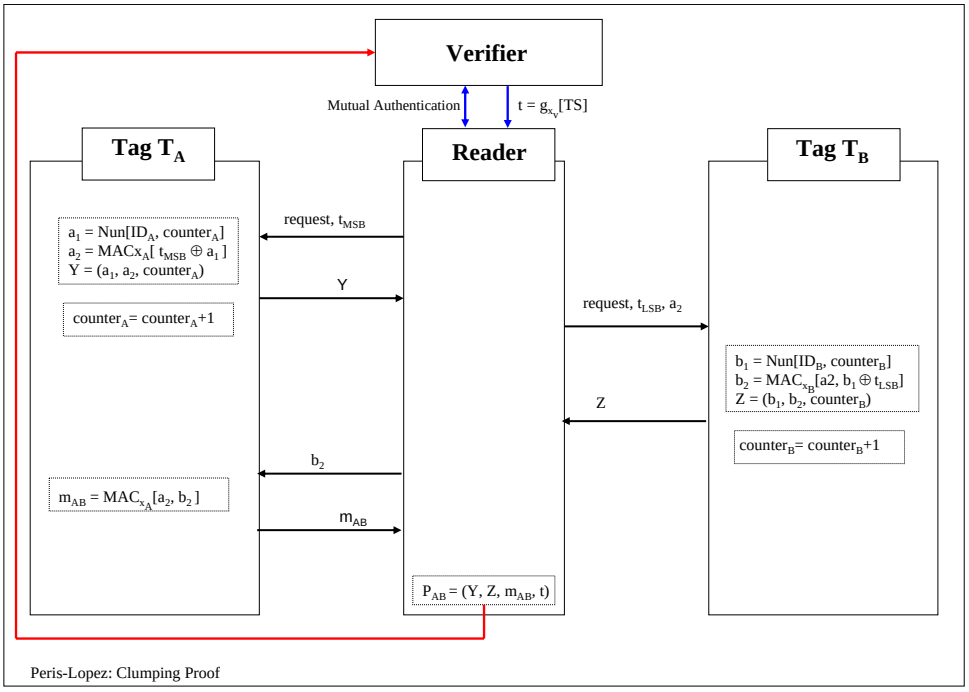


(b) Fig. 5.14 (B)

Figure 5.14: Simultaneous Reading (I)



(a) Fig. 5.15 (A)



(b) Fig. 5.15. (B)

Figure 5.15: Simultaneous Reading (II)

In 2005, Saito et al. presented a replay attack against the yoking proof [191]. Additionally, Peris et al. showed that the possession of a yoking proof only implies that the involved tags have been read within a specific interval, but it does not imply that the two tags have been simultaneously read [167].

Yoking proofs are based on message authentication codes and random numbers. Saito and Sakurai proposed a yoking proof, which they called "grouping proofs", using timestamps [191] to be immune to replay attacks. However, Piramuthu demonstrated [174] that grouping proofs are vulnerable to replays attacks too.

In 2006, a new proof inspired by Juels's yoking proofs was proposed by Piramuthu [174]. Although Piramuthu's proposed scheme seemed to be resistant to replay attacks, Peris et al. showed its vulnerability to multi-proof session attacks. Furthermore, Piramuthu claims that privacy and location privacy is guaranteed in his scheme, which is not the case as tags transmit their static identifiers in clear [167].

Finally, in 2007, Peris et al. proposed a new anonymous proof, named "clumping proof", that solves the multi-proofs session attack and provides privacy while also protecting against tracking [167].

Chapter 6

Lightweight Cryptography for Low-cost RFID Tags

6.1 Introduction

The major challenge in trying to provide security for low-cost RFID tags is their very limited computational capabilities (storage, circuitry and power consumption), which makes them unable to perform the most basic cryptographic operations.

Before designing a new protocol, the requirements and restrictions of a system should be analyzed. The security level of an RFID tag used in an e-passport should not be the same as that of a low-cost tag employed in the supply chain (i.e. tags conforming to the EPC-C1G2 specification). To clarify the kind of systems we refer to as low-cost/high-cost RFID tags, *Table 6.1* summarizes their specifications, these being relevant to current-commercial RFID tags.

In [46], Chien proposed a tag classification mainly based on which were the operations supported on-chip. High-cost tags are divided into two classes: "full-fledged" and "simple". Full-fledged tags support on-board conventional cryptography like symmetric encryption, cryptographic one-way functions and even public key cryptography. Simple tags can support random number generators and one-way hash functions. Likewise, there are two classes for low-cost RFID tags."Lightweight" tags are those whose

chip supports a random number generation and simple functions like a Cyclic Redundancy Code (CRC) checksum, but not a cryptographic hash function. "Ultralightweight" tags can only compute simple bitwise operations like XOR, AND, OR, etc. These ultralightweight tags represent the greatest challenge in terms of security, because of the widespread use that is anticipated for them and their very limited capabilities.

Table 6.1: Specifications for Low-cost and High-cost RFID Tags

	Low-cost RFID Tag	**High-cost RFID Tag**
Standards	EPC Class-1 Generation-2 ISO/IEC 18006-C	ISO/IEC 14443 A/B
Power Source	Passively powered	Passively powered
Storage	32 - 1K bits	32 KB - 70 KB
Circuitry (security processing)	250 - 4K gates Standard cryptographic primitives cannot be supported	Microprocessor Implement 3DES, SHA-1, RSA
Reading Distance (commercial devices)	Up to 3 m	About 10 cm
Price	0.05 - 0.1 €	Several euros
Physical Attacks	Not resistant	Tamper resistance EAL 5+ security level
Resistance to Passive Attacks	Yes	Yes
Resistance to Active Attacks [52, 111, 118, 158]	No	Yes

In spite of the severe restrictions of low-cost RFID tags, most of the proposed solutions are based on the use of hash functions [49, 93, 136, 160, 193, 228]. Although this apparently constitutes a good and secure approach, engineers face the non-trivial problem of implementing cryptographic hash functions with only 250-4K gates [179]. In most of the aforementioned proposals, no explicit algorithms were suggested and finding one is not an easy issue, since traditional hash functions cannot be used (SHA-256, SHA-1, MD5, etc.). The best implementation of SHA-256 requires around 11K gates and 1120 clock cycles to perform a hash calculation on a 512-bit data block [73]. As the number of resources needed is much greater than those of a low-cost RFID tag, it may seem natural to propose the use of smaller hash functions. However, neither functions such as SHA-1 (8.1K gates, 1228 clock cycles) or MD5 (8.4K gates, 612 clock cycles) fit in a tag [73]. Recently, some authors have suggested the use of a "universal hash function" [229]. Although this solution only needs around 1.7K gates, a more thorough security analysis is necessary and this has not yet been done. Additionally, this function only has a 64-bit output, which does not guarantee adequate

security because finding collisions is a relatively easy task because of the birthday paradox (around 2^{32} operations).

Finally, none of the proposals comply with the EPC-C1G2 specification, because hash functions are not supported on Gen-2 RFID tags.

6.2 Cryptanalysis

Cryptanalysis is the science that evaluates the promises of cryptography. For example when you buy an air flight over the internet and pay with your credit card, you would like that transaction to be secure. Transaction security depends on the security mechanisms used. Cryptanalysis focuses on evaluating the strength of cryptographic primitives and protocols [173, 200, 207].

Security issues do not seem to be properly addressed in the EPC-C1G2 standard. A number of papers, such as [27, 109, 124, 158] , have proposed new protocols to enhance the security of this standard. Unfortunately, due to weaknesses that have been exposed in them (see *Chapter 4*), these protocols fall short of meeting the desired security objectives. In this section, we show the security vulnerabilities of two recent authentication protocols within the framework of the EPC-C1G2 specification [168, 172].

6.2.1 Chien et al. Protocol

In [47], Chien et al. proposed a mutual authentication protocol for improving the security of EPC-C1G2. Their scheme consists of two phases: an initialization phase and an authentication phase.

Initialization Phase
For each tag denoted as T_i, the server randomly selects an initial authentication key K_{i_0} and a initial access key P_{i_0}. These two values are stored in the tag together with the EPC (EPC_i). The authentication and access key are updated after each successful authentication. For each tag, the server S (back-end database) maintains a record of six values: (1) EPC_i; (2) the old authentication key for this tag (K_{old}), which is initially set to K_{i_0}; (3) P_{old} denotes the old access key for this tag, which is initially set to P_{i_0}; (4) K_{new}

denotes the new authentication key, which is initially set to K_{i_0}; (5) P_{new} denotes the new authentication key, which is initially set to P_{i_0}; (6) $Data$ denotes all the information about the tagged object.

The (n+1) Authentication Phase

$R \rightarrow T_i$:	N_1 The reader sends a random nonce N_1 as a challenge to the tag.
$T_i \rightarrow R \rightarrow S$:	M_1, N_1, N_2 The tag generates a random number N_2, computes $M_1 = CRC(EPC_i \|\| N_1 \|\| N_2) \oplus K_{i_n}$, and sends the value back to the reader, which will forward these values to the server. The reader interactively selects an entry $(EPC_i, K_{old}, K_{new}, P_{old}, P_{new}, Data)$ from its database, computes $I_{old} = M_1 \oplus K_{old}$ and $I_{new} = M_1 \oplus K_{new}$, and checks whether any of these two equations hold $I_{old} = CRC(EPC_i \|\| N_1 \|\| N_2)$ $I_{new} = CRC(EPC_i \|\| N_1 \|\| N_2)$. This is designed to be a way of avoiding desynchronization attacks. The process is repeated until a match is found in the database, thus implying a successful authentication of the tag. If no match is found, a failure message is sent to the reader, and the authentication process is stopped.
$S \rightarrow R$:	M_2, $Data$ After a successful authentication, the server computes $M_2 = CRC(EPC_i \|\| N_2) \oplus P_{old}$ or $M_2 = CRC(EPC_i \|\| N_2) \oplus P_{new}$, depending on which value (K_{old}, K_{new}) satisfies the equation in the previous step. It also updates $K_{old} = K_{new}$, $P_{old} = P_{new}$, $K_{new} = PRNG(K_{new})$ and $P_{new} = PRNG(P_{new})$. The server sends M_2, $Data$ to the reader.
$R \rightarrow T_i$:	M_2 Upon receiving M_2, the tag verifies whether the equation $M_2 \oplus P_{i_n} = CRC(EPC_i \|\| N_2)$ holds. If so, it updates its keys $K_{i_{n+1}} = PRNG(K_{i_n})$ and $P_{i_{n+1}} = PRNG(P_{i_n})$.

6.2.1.1 Cyclic Redundancy Codes - CRC's

A Cyclic Redundancy Code (CRC) is a checksum algorithm that can be used to detect transmission errors (typically one or two bit flips, or bursts) in a very efficient way. CRCs operate by interpreting input binary sequences as polynomial coefficients that they divide over a prefixed polynomial in order to obtain a remainder, which, in its binary expression, constitutes the crc value.

CRCs are completely linear, so they shouldn't be used in cryptographic applications as they cannot detect malicious changes by a knowledgeable attacker [22, 181, 203, 227]. To illustrate this property, the Hamming Distance (HD) can be used. The HD of a CRC polynomial is the minimum possible number of bit errors that is undetected by computing the CRC. For example, if a CRC has a HD of 3, any combinations of 1 or 2 bit errors will be detected, but there is at least one combination of 3 bit errors that will pass undetected. Cryptographic hash functions should therefore be used for this purpose.

Computing a crc value for a given binary stream is essentially dividing the polynomial associated with this stream by another fixed polynomial (that depends on the particular CRC implementation) and computing a remainder. The stream should be multiplied by x^N (being N the degree of the crc polynomial) prior to division. That is to say, computing the crc of a polynomial $i(x)$ is basically finding a remainder $r(x)$ so that,

$$i(x) \cdot x^N = d(x) \cdot p(x) + r(x) \quad with \ \ |r(x)| < |p(x)| \tag{6.1}$$

The reader is referred to [210] where a detailed explanation about the election of the $p(x)$ polynomial is included. Some popular polynomials are the following:

CRC-8 $= x^8 + x^5 + x^4 + 1$

CRC-12 $= x^{12} + x^{11} + x^3 + x^2 + x + 1$

CRC-16 $= x^{16} + x^{15} + x^2 + 1$

CRC-CCITT $= x^{16} + x^{12} + x^5 + 1$

CRC-32 $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + +x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

CRC-64 $= x^{64} + x^4 + x^3 + x + 1$

The EPC-C1G2 specification proposes the use of CRC-CCITT which detects all single and double errors, all errors with an odd number of bits, all burst errors of length 16 or less, 99.997 percent of 17-bit error bursts, and 99.998 percent of 18-bit and longer bursts.

The hardware requirements of a CRC generator are not very demanding. Specifically, an n bit CRC consists of an n-bit shift register with some XOR gates, as illustrated in *Figure 6.1.* To compute the CRC:

1. Load the register with the Preset value.
2. Augment the message by appending N zeros to the end of it ($i(x) \cdot x^N$).
3. While (more message bits)
    3.1 Shift the register left by one bit, reading the next bit of the augmented message into register bit position 0.
    3.2 If (a 1 bit popped out of the register during the above step)
        $Register$ = $Register$ XOR $p(x)$
    End While
4. The register contains the remainder.

CRC Properties

Due to their linearity, CRCs have some properties that, from the security point of view, one can label as bad. In fact, we will show that one of these "bad" properties (derived from their linear structure) will be enough to successfully attack Chien et al.'s mutual authentication protocol in various ways.

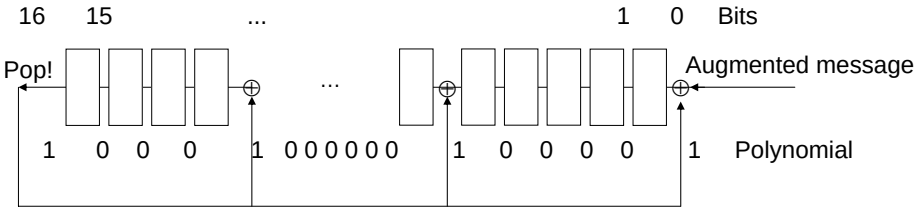


Figure 6.1: CRC-CCITT Implementation

Theorem 1. *For any CRC (independent of its divider polynomial) and for any values a, b, c and $d \in F_2[x]$, it holds that:*

$$CRC(a\|b) \oplus CRC(c\|d) = CRC(a \oplus c\|b \oplus d) \tag{6.2}$$

Proof. From the definition in *Equation 1* above, one can write:

$$CRC(a\|b) = (a \cdot x^N \oplus b) \cdot x^N \oplus d_1(x) \cdot p(x) \tag{6.3}$$

$$CRC(c\|d) = (c \cdot x^N \oplus d) \cdot x^N \oplus d_2(x) \cdot p(x) \tag{6.4}$$

for certain polynomials $d_1(x)$ and $d_2(x) \in F_2[x]$. Substituting these values in the left side of *Equation 2* we obtain the following:

$$(a \cdot x^N \oplus b) \cdot x^N \oplus d_1(x) \cdot p(x) \oplus (c \cdot x^N \oplus d) \cdot x^N \oplus d_2(x) \cdot (x) \tag{6.5}$$

Rearranging terms in this expression we get:

$$((a \oplus c) \cdot x^N \oplus (b \oplus d)) \cdot x^N \oplus (d_1(x) \oplus d_2(x)) \cdot p(x) \tag{6.6}$$

that is the corresponding expression for $CRC(a \oplus c\|b \oplus d)$ (analogously to *Equation 3* and *4*). □

Corollary 1. *In particular, if in Equation 1 we have $a = c$, then,*

$$\begin{aligned} crc(a\|b) \oplus crc(a\|d) &= crc(a \oplus a\|b \oplus d) = crc(0\|b \oplus d) = \\ &= crc(b \oplus d) \end{aligned} \tag{6.7}$$

because $0 \cdot x^N \equiv 0 \cdot p(x)$

This is the property we will use to our advantage in attacking Chien et al.'s protocol (and, for that matter, any other protocol relying on the use of a CRC as a means of concealing secrets). It is important to point out that this holds for every CRC implementation, independently of its length and crc polynomial (CRC-8, CRC-16, CRC-32, CRC-64, etc.).

6.2.1.2 Vulnerabilities of Chien's Protocol

In this section we will analyze the most important vulnerabilities in Chien et al.'s protocol:

1. Unequivocal Identification
There is a fundamental difference between barcode technology and RFID. Barcodes use Universal Product Codes (UPC) to identify the class of items. RFID technology replaces UPC with the Electronic Product Code (EPC) that allows the unequivocal identification of tagged items. The Tag Data Specification [67] does not provide any specific guidance for using EPCs in UHF Class-1 Generation-2 tags. So in the following we assume that EPCs will be managed in the same way as they were in the EPC-C1G1 standard. So the EPC is composed of the following fields (identical to those of the General Identifier, GID-96):

- $Header$ is set to the fixed hexadecimal value 0x35 (8-bits).
- $General\ manager$ identifies a company, manager or organization (28-bits).
- $Object\ class$ is used by an EPC managing entity to identify class or "type" of thing (24-bits).
- $Serial\ number$ is unique within each object class (36-bits).

Static identifiers (EPC-96) represent valuable information that should be transmitted on the channel guaranteeing confidentiality and, at the same time, avoiding the tracking of its holders. Researchers have proposed a number of pseudonym-based solutions to provide privacy protection (data and location). The most commonly used solution in the literature for pseudonym updating consists of repeatedly applying a hash function to

the static identifier (i.e. $pseudonym_i = hash^i(EPC)$). However, hash functions have not been ratified by the EPC-C1G2 specification because of the inherent computational limitations of low-cost RFID tags. As we saw in *Chapter 4*, tags conforming to EPC-C1G2 only support on-board a 16-bit Pseudo-Random Number Generator and a 16-bit Cyclic Redundancy Code (CRC) checksum.

In the inventory command, as described in the EPC-C1G2 specification, tags transmit their EPC as plain text. Chien et al. propose that tags transmit $M_1 = CRC(EPC||n_1||n_2) \oplus K_{i_n}$ instead, where the nonce n_1 is generated by the reader and the nonce n_2 is generated by the tag. Message M_1, concatenated with these two nonces n_1 and n_2 is sent to the back-end database. This scheme presents a serious security failure.

An EPC has the first 8 bits of the header fixed, while the remaining 88 bits are variable. So, there are 2^{88} possible identifiers. However, tags support on-board a 16-bit CRC (ISO/IEC 13239, p(x)=$x^{16} + x^{12} + x^5 + 1$, Preset=0xFFFF, Residue=0x1D0F). So the 2^{88} possible EPC values collapse in only 2^{16} possible values when the CRC is applied to the EPC ($M_1 = CRC(EPC\ ||n_1||n_2) \oplus K_{i_n}$).

Weakness 1: Chien et al's protocol does not guarantee the unequivocal identification of tagged items, which is an essential property in authentication protocols.

We have simulated a population of N tags. For each tag, the values of EPC, K, P were randomly initialized. These values will be stored both in each tag and at the back-end database. Upon initialization, we simulate the reading of these N tags. For each reading, the following process is repeated:

(1) Reading of tag_x
(2) M_1 = CRC(EPC_x || n_1 || n_2) $\oplus$ K_x
(3) Send M_1, n_1, n_2 to the back-end database
(4) for(x'==1,x'< N,x'++)
  M_1'= CRC($EPC_{x'}$ || n_1 || n_2) $\oplus$ $K_{x'}$
  if ((x' != x) && (M_1' == M_1)) collision++;
(5) if collision>0 "Failed Unequivocal Identification"

I.E. Failed Unequivocal Identification

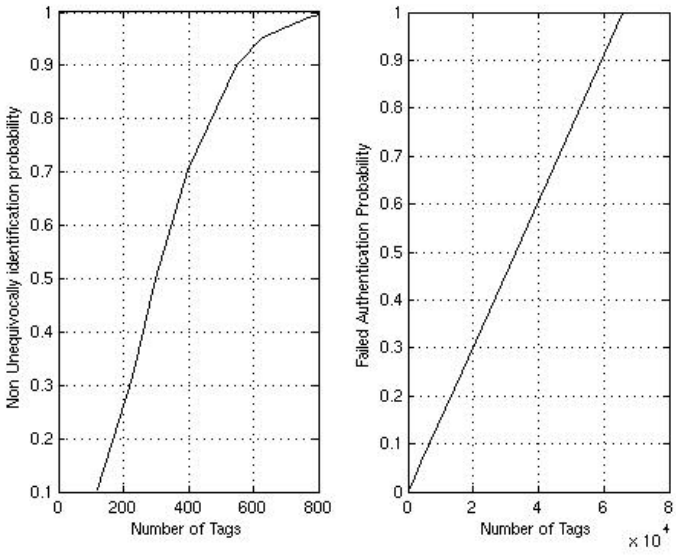
EPC$_x$= 0xe48862a92b704993e0698583	EPC$_{x'}$=0xf1af12caee0319f564f89098
K$_x$= 0x9cf5	K$_{x'}$= 0x2336
n1 = 0xbdc5	n1 = 0xbdc5
n2 = 0xa6f4	n2 = 0xa6f4
M1 = CRC(EPC$_x$||n1||n2) $\oplus$ K$_x$	M1'= CRC(EPC$_{x'}$||n1||n2) $\oplus$ K$_{x'}$

M1 = M1'=0xa2b2

The above process is repeated T times ($T = 10^4$) in order to obtain an estimation of the non-unequivocal identification probability (P_{NUI}). We have simulated the above experiment with eight different values ($N = 118$, 226, 301, 397, 549, 626, 769, 800). The values obtained are summarized in *Figure 6.2 (A)*, and fit perfectly with the values obtained for the birthday paradox with a group of N tags and $d = 2^{16}$ boxes:

$$p(N;d) = \begin{cases} 1 - \prod_{k=1}^{N-1}(1 - \frac{k}{d}) & N \leq d \\ 1 & N > d \end{cases} \tag{6.8}$$

These results are hardly surprising, since each time a tag (tag_x) is read, we search if the equality $M_1 = CRC(EPC_x||n_1||n_2) \oplus K_x == CRC(EPC_{x'}||n_1||n_2) \oplus K_{x'} = M_1'$ holds for $x' \neq x$. As specified in EPC-C1G2, the CRC is 16-bits length. The kill and access passwords are 32-bits length in the specification. However, only one half (MSB or LSB) is included in each message. Chien et al. do not state the key length of the authentication key (k_x) nor that of the access key (P_x). From the above, we can assume that these keys are 16-bit length. Moreover, the keys are xored with a CRC of 16-bit length, so the previous assumption seems consistent with their usage. Finally, if we assume that the $CRC()$ and K_i are uniformly distributed (which may well not be the case, specially in the case of CRCs), the probability that at least two of N randomly selected tags have the same index ($M_1 = M_1'$), is exactly that of the birthday paradox with parameters N and $d = 2^{16}$. Therefore, we have demonstrated that tags cannot be unequivocally identified under these conditions. For example, with a population of tags greater than 300, we have at least one non-unequivocal



(a) Fig. 6.2 (A) (b) Fig. 6.2 (B)

Figure 6.2: Non Unequivocal Identification and Failed Authentication

identification with a probability over 0.5 ($P_{NUI} > 0.5$). Similarly, even for a relatively low number of tags (600) the probability of non-unequivocal identification rises to more than 90% (see *Figure 6.2 (A)*).

Finally, even if we considerably relax the requirements of our RFID system even allowing the existence of collisions in the identification process (although if the collision probability is low, this seems a very unusual decision), Chien's protocol could present serious operational problems as the number of tags increases, because the probability of failed authentication rises quickly. To verify this, we carried out an experiment similar to that described above, although in this case each time a tag is read the absolute number of collisions in the database is computed. The experiment has been simulated with six different values ($N = 2^7$, 2^{10}, 2^{12}, 2^{14}, 2^{15}, 2^{16}), and repeated T times ($T = 10^4$). The obtained results are displayed in *Figure 6.2 (B)*. So, the probability of failed identifications in a population of N tags, is described by the following equation:

$$P_{FI}(N) = \begin{cases} 2^{log_2(N)-16} & N \leq 2^{16} \\ 1 & N > 2^{16} \end{cases} \tag{6.9}$$

I.E. Several Failed Identification of $\text{Tag}_{0x29a51b66cf66663d2dc9f16f}$

Messages Transmitted

Reader $\leftrightarrow \text{Tag}_{0x38304699082162af23df77a1}$:	M1	n1	n2	M2
Reader $\leftrightarrow \text{Tag}_{0x9f52dc22daa678fe85d68b23}$:	$M1'$	$n1'$	$n2'$	$M2'$
Reader $\leftrightarrow \text{Tag}_{0xdb58b949c3a24a24484999a8}$:	$M1''$	$n1''$	$n2''$	$M2''$

EPC_x = 0x38304699082162af23df77a1
K_x = 0xabd8
n1 = 0x1b17 n2 = 0x2b72

$M1 = CRC(EPC_x||n1||n2) \oplus K_x$

$EPC_{x'''}$ = 0x29a51b66cf66663d2dc9f16f
$K_{x'''}$ = 0xa27b
n1 = 0x1b17 n2 = 0x2b72

$M1''' = CRC(EPC_{x'}||n1||n2) \oplus K_{x'''}$

M1 = M1$'''$=0xe5ce

$EPC_{x'}$ = 0x9f52dc22daa678fe85d68b23
$K_{x'}$ = 0x61b8
$n1'$ = 0x8eac $n2'$ = 0xfb81

$M1' = CRC(EPC_{x'}||n1'||n2') \oplus K_{x'}$

$EPC_{x'''}$ = 0x29a51b66cf66663d2dc9f16f
$K_{x'''}$ = 0xa27b
$n1'$ = 0x8eac $n2'$ = 0xfb81

$M1''' = CRC(EPC_{x'''}||n1'||n2') \oplus K_{x'''}$

M1$'$ = M1$'''$=0x5b41

$EPC_{x''}$ = 0xdb58b949c3a24a24484999a8
$K_{x''}$ = 0xc14e
$n1''$ = 0x8017 $n2''$ = 0xf1c3

$M1''=CRC(EPC_{x''}||n1''||n2'') \oplus K_{x''}$

$EPC_{x'''}$ = 0x29a51b66cf66663d2dc9f16f
$K_{x'''}$ = 0xa27b
$n1''$ = 0x8017 $n2''$ = 0xf1c3

$M1'''=CRC(EPC_{x'''}||n1''||n2'') \oplus K_{x'''}$

M1$''$ = M1$'''$=0x7058

2. Tag Impersonation and Forward Secrecy

Each tag shares with the reader some private information: EPC, authentication key (k_x) and the access key (P_x). This information is used to build messages M_1 and M_2 in order to prove its authenticity. However, a passive attacker eavesdropping the backward and forward channel (see [182] for a eavesdropping range classification) will be able to supplant a legitimate tag as described below:

Weakness 2: Chien et al's protocol does not guarantee the non-impersonation of legitimate tags.

Proof: In order to accomplish this attack, an adversary only needs to listen to an iteration between the reader and the legitimate tag.

(1) $R \rightarrow T:$ $\quad n_1$
(2) $T \rightarrow R:$ $\quad M_1 = CRC(EPC_{\text{x}}||n_1||n_2) \oplus K_{\text{x}}, n_2$

At this point, the attacker isolates the legitimate tag, preventing it from operating. He has the following information: M_1, n_1, and n_2. With this, the attacker should be able to build message $M_1' = CRC(EPC_{\text{x}}||n1'||n2')$ when queried by the reader. Although the attacker does not know the private information stored in the tag (EPC, K_x, and P_x), message M_1' can be easily computed as described below. *Corollary 1* states:

$$crc(a||b) \oplus crc(a||d) = crc(b \oplus d) \tag{6.10}$$

As this holds for every a, b, d $\in F_2[x]$, if b and d are the concatenation of some other variables ($b = b1||b2$, $d = d1||d2$), the above expression also holds and can be rewritten as:

$$\begin{aligned} &crc(a\|b) \oplus crc(a\|d) = crc(a\|b1\|b2) \oplus crc(a\|d1\|d2) = \\ &crc((b1\|b2) \oplus (d1\|d2)) = crc(b1 \oplus d1\|b2 \oplus d2) \end{aligned} \tag{6.11}$$

So the difference between the known value M_1 and the new challenge M_1' is exactly $M_1 \oplus M_1' = CRC(EPC||n_1||n_2) \oplus CRC(EPC||n_1'||n_2')$ and, substituting in *Equation 6.11*, we get

$$CRC(EPC||n_1||n_2) \oplus CRC(EPC||n_1'||n_2') = CRC(n_1 \oplus n_1'||n_2 \oplus n_2') \tag{6.12}$$

So, message M_1' can be obtained doing an XOR between message M_1 and the easily computable value $CRC(n_1 \oplus n_1'||n_2 \oplus n_2')$ (because all nonces are transmitted in clear and the CRC function is public). Therefore, the

identity of a legitimate tag could be easily impersonated. An ANSI-C code with the implementation of this attack is available in `http://163.117.149.208/rfid/chien/attack2.c`.

I.E. Tag Impersonation

EPC_x = 0xe34f5cdd919f4f2f9211678fe K_x = 0xb224 Reader → Tag: n1 = 0xb3e2 Tag → Reader: M1 = 0x21b4, n2 = 0x5fa4 (Isolate the tag)
Reader → Attacker: $n1^{'}$ = 0x77d8 Attacker → Reader: $M1^{'}$ = M1 ⊕ CRC($n1 \oplus n1^{'} \| \| n2 \oplus n2^{'}$) = 0x21b4 ⊕ 0x5e73 = **0x7fc7** $n2^{'}$ = 0xf0e2 Database: Check $M1^{'}$ = CRC(EPC||$n1^{'}$ || $n2^{'}$) ⊕ K_x = = 0xcde3 ⊕ 0xb224 = **0x7fc7** Tag is impersonated!

Additionally, the scheme does not provide forward secrecy protection. Suppose that an adversary listens to an iteration between a legitimate reader and a legitimate tag (M1, n1, n2, M2) and stores these values. Then, the tag which is not resistant to physical attacks is compromised, the EPC being obtained by the attacker. At this point, the attacker will be able to obtain the secret keys (K_x and P_x) and to generate future $M1^{'}$, M1″, etc. A detailed explanation of this attack is described below:

I.E. Forward Secrecy

EPC_x = 0x4d3174f00cf844e4ce5fb064 K_x = 0x1479 P_x = 0xe04d Reader → Tag: n1 = 0x119b Tag → Reader: M1 = 0x1b36, n2 = 0x8a4b Reader → Tag: M2 = 0xc57a ... Tag is compromised obtaining its EPC

Attacker: M1, M2, n1, n2, EPC_x are known
Obtaining the keys:
K_x=CRC(EPC || n1 || n2) $\oplus$ M1 = 0x1479
P_x=CRC(EPC || n2) $\oplus$ M2 = 0xe04d

$K_{x'}$ = PRNG(K_x) = 0xa586
M1$'$ = CRC(EPC || n1′ || n2′) $\oplus$ $K_{x'}$

The attacker is able to generate future M1 messages

i.e. n1′ = 0xfa4b n2′ = 0x4b88
M1$'$ = CRC(EPC || n1′ || n2′) $\oplus$ $K_{x'}$ =
= 0x3c64 $\oplus$ 0xa586 = 0x99e2

3. Back-end Database Impersonation

In the above section we focused on message M_1 sent by the tag when queried by the reader. In this case we concentrate on message M_2, generated by the back-end database. The attacker should be able to generate this message in order to impersonate a legitimate back-end database.

Weakness 3: Chien et al.'s protocol is vulnerable to back-end database impersonation.

Proof: For the attacker, it is enough to listen to an iteration between a legitimate tag and a reader-database in order to exploit this vulnerability:

(1) $R \rightarrow T$: n_1
(2) $T \rightarrow R$: $M_1 = CRC(EPC_x||n_1||n_2) \oplus K_x, n_2$
(3) $R \rightarrow Database$: M_1 , n_1, n_2
(4) $Database \rightarrow R$: $M_2 = CRC(EPC_x||n_2) \oplus P_x$
(5) $R \rightarrow T$: M_2

The attacker has to block or disrupt the radio channel to obstruct the correct reception of message 5. The objective of this is to prevent the legitimate tag from updating its key. At this point, the attacker could supplant the back-

end database without knowing all its private information. In the next tag reading, the database will receive M_1', n_1', n_2'. The fraudulent database has to compute the message M_2'. But from *Corollary 1*, the following expression can be derived:

$$M_2 \oplus M_2' = CRC(EPC||n2) \oplus CRC(EPC||n2')$$
$$CRC(EPC||n2) \oplus CRC(EPC||n2') = CRC(n2 \oplus n2') \quad (6.13)$$

So, message M_2' can be obtained by means of an XOR between the previous M_2 message listened to in the air channel and the easily computed value $CRC(n_2 \oplus n_2')$. Message M_2' will be sent to the tag, which will authenticate the fraudulent back-end database and update its keys. An ANSI-C code with the implementation of this attack is available in `http://163.117.149.208/rfid/chien/attack3.c`.

I.E. Database Impersonation

EPC_x = 0x52c3e4175b97de07f22f9db0	K_x = 0xf6dd P_x = 0xca39
Reader → Tag:	n1 = 0x04a6
Tag → Reader:	M1 = 0x7a98, n2 = 0xa833
Reader → Tag:	M2 = 0x25f6 (**blocked!**)
...	
Attacker → Tag:	n1′ = 0xf556
Tag → Attacker:	M1′ = 0x47dc, n2′ = 0xae5c
Attacker → Tag:	M2′ = M2 ⊕ CRC(n2 ⊕ n2′) = **0x1219**
Tag:	Check M2′ = CRC(EPC || n2′) ⊕ P_x =
	= 0xd820 ⊕ 0xca39 = **0x 1219**
Back-end database is impersonated!	

4. Tracking or Private Location

Protection against tracking is not guaranteed when tags answer reader queries with the same identifier. In Chien et al.'s protocol, nonces n_1 and n_2 are employed in each session to ensure freshness. With this, it seems that the tag's private location is assured. This is not the case, as explained below:

Weakness 4: Chien et al.'s protocol does not guarantee the location privacy of tags

Proof: The success of this attack depends on preventing tag key updating. Moreover, if the population of tags is greater than 2^{16}, the normal operation of the protocol will hamper the key update operation (non-unequivocal identification). In the back-end database a pair of keys (new, old) are stored for each tag key. Chien et al. claim that the storage of these two keys frustrates DoS attacks. To provide this property, the fact that a tag may sometimes use the same key (message M_2 was incorrectly received) to compute the message M_1 is considered as a normal operation. In fact, Chien does not specify the maximum number of times that a tag can be authenticated with the same authentication key. Imagine that the reader captures two non-consecutive iterations, upon non-updating key condition (an ANSI-C code with the implementation of this attack is available in `http://163.117.149.208/rfid/chien/attack4.c`):

(1) $R \rightarrow T$: $n1$
(2) $T \rightarrow R$: $M_1 = CRC(EPC_{\text{x}}||n_1||n_2) \oplus K^n_{\text{x}}, n_2$
....
(1) $R \rightarrow T$: $n1'$
(2) $T \rightarrow R$: $M'_1 = CRC(EPC_{\text{x}}||n'_1||n'_2) \oplus K^n_{\text{x}}, n'_2$
....

Now, the attacker computes the XOR of messages M_1 and M'_1. If messages M_1 and M'_1 came from the same tag, the key K^n_{x} is cancelled when the XOR is computed. By means of *Equation 6.11*, the attacker can verify if answers arise from the same tag:

$$M_1 \oplus M'_1 = CRC(n_1 \oplus n'_1||n_2 \oplus n'_2) \tag{6.14}$$

I.E. Private Location Jeopardized

EPC$_x$ = 0x26d4caaaaa59d9AAAa3afbeaf871fb35c K$_x$ = 0x650b
Upon non-updating key condition ...

Reader → Tag:	n1 = 0x1305
Tag → Reader:	M1 = 0x6b3c, n2 = 0xb642
...	
Reader → Tag:	n1′ = 0x1ea4
Tag → Reader:	M1′ = 0x1e33, n2′ = 0xf4a7
...	
Attacker:	A = M1 ⊕ M1′ = **0x750f**
	B = CRC ($n1 \oplus n1^{'} \| n2 \oplus n2^{'}$) = **0x750f**
	A==B ⇒ Tag answers provide for the same tag!

A similar attack can be accomplished using messages sent by the database (M_2). Suppose that a crooked reader interrogates a tag: reader sends the message M_1 to the database. The database authenticates the tag, and sends back message M_2. M_2 is stored by the reader, and a wrong M_2 message is sent to the tag avoiding its key updating. Then, the above process is repeated obtaining messages M_1', and M_2' with nonces n_1' and n_2'. At this point, the attacker computes the XOR of messages M_2 and M_2' to check if they came from the same tag. From *Corollary 1*, the following equality has to be fulfilled:

$$M_2 \oplus M_2' = CRC(n_2 \oplus n_2') \tag{6.15}$$

5. Back-end Database Auto-desynchronization

To defend against a DoS attack, Chien et al. propose that the back-end database maintains a pair of keys (new, old) for each tag key. This assumption allows the server to authenticate tags and re-synchronize these each time they suffer a DoS attack. However, the normal operation of the protocol results in synchronization loss between the database and the tags due to the non-unequivocal identification property.

Weakness 5: Chien et al.'s protocol is vulnerable to auto-desynchronization attacks.

We have simulated a population of N tags. For each tag the EPC, K, and P values are randomly initialized. These values will be stored both in the tag

and in the back-end database. Upon initialization, we simulate the reading of N tags. For each reading, the following process is repeated:

(1) Reading of tag_x
(2) M = CRC(EPC_x || n_1 || n_2) $\oplus$ K_x
(3) Send M, n_1, n_2 to the back-end database
(4) while ((x'<N)&&(output==0))
{M'= CRC($EPC_{x'}$ || n_1 || n_2) $\oplus$ $K_{x'}$
if (M' == M) autodesyn[x']++; output=1
x'++;}

Upon the reading of the N tags, we compute the number of times that auto-desynchronization occurred. After the reading of a tag its keys are updated both in the database and in the tag. An additionally wrong update, during the reading of a different tag, will cause a loss of synchronization for that tag. Therefore, the number of tags whose keys have been updated two or more times constitute the number of desynchronized tags. So the probability of auto-desynchronization (P_{ADS}) can be defined as:

$$P_{ADS} = \frac{1}{N}\sum_{x=1}^{N}(autodesyn[x] - 1) \tag{6.16}$$

The above process is repeated T times ($T = 10^3$) in order to obtain an estimation of the P_{ADS}. We have simulated the experiment with different values (N = 2^{14}, 2^{15}, 2^{16} ... 2^{18}), as displayed in *Figure 6.3*. The results indicate that if we have a population of $N \geq 2^{17}$ tags, the probability of auto-desynchronization is greater than 0.5. This probability increases to 0.93 if the population is $N \geq 2^{18}$.

6.2.1.3 Remarks

We have shown that all these security weaknesses of Chien protocol are related to the use of the CRC. Some of them (non-unequivocal identification and autodesynchronization) could have been solved simply by using larger CRCs (well above the 16-bit CRC proposed in the standard). The rest of the security problems highlighted are due to the bad (linear) properties of CRCs and will not be solved by changing the CRC length. Indeed, we

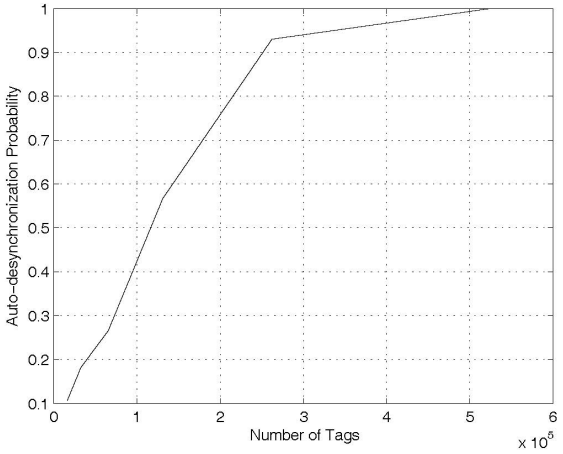


Figure 6.3: Auto-desynchronization Probability

doubt that CRCs should be used in any security protocol at all, and its use should be confined to guarantee an error-free communication channel.

6.2.2 Konidala and Kim Protocol

A first initiative by Konidala and Kim [124] tried to solve the security shortcomings of the EPC-C1G2 specification by presenting a mutual authentication scheme (TRMA) to protect the tag's access password. However, Lim and Li showed how a passive attacker can recover this [143]. Recently, Konidala and Kim proposed a new version of the TRMA scheme (TRMA$^+$) in which the tag access and kill password are used for authentication [125].

6.2.2.1 The Original TRMA Scheme and its Extension

For completeness and readability we will first provide a brief description of the original TRMA scheme and its extended version TRMA$^+$.

Original TRMA Scheme

A brief description of the TRMA scheme follows. For further details, the reader is referred to the original work in [124].

Tag ⇒ Reader: EPC, RN_1^{Tag}, RN_2^{Tag}

First the tag is singulated and backscatters its EPC number. Then the reader sends two $ReqRN$ commands to the tag, which responds by backscattering two generated 16-bit random numbers: RN_1^{Tag} and RN_2^{Tag}.

Reader ⇒ Tag: RN_1^{Rdr}, RN_2^{Rdr}, $CCPwd_{M1}$, $CCPwd_{L1}$, RN_3^{Rdr}, RN_4^{Rdr}

The reader also generates two 16-bit random numbers: RN_1^{Rdr} and RN_2^{Rdr}. The four random numbers and the access password are used to construct $CCPwd_{M1}$ and $CCPwd_{L1}$ responses:

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \quad (6.17)$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \quad (6.18)$$

where $APWD_M$ and $APWD_L$ are the 16 most significant and 16 least significant bits of the access password, respectively. $PAD_i = PadGen(RN_i^{Tag}, RN_i^{Reader})[APWD]$, where $PadGen(.)$ is a specially designed pad generation function. Next, two 16-bit random numbers (RN_3^{Rdr}, RN_4^{Rdr}), which will be used in tag authentication, are generated and transmitted to the tag.

Tag: Verify $CCPwd_{M1}$ and $CCPwd_{L1}$. If both values are correct, the process continues. Otherwise, the process is aborted.

Tag ⇒ Reader: RN_3^{Tag}, RN_4^{Tag}, $CCPwd_{M2}$, $CCPwd_{L2}$

The tag also generates two new random numbers (RN_3^{Tag}, RN_4^{Tag}), and builds answers $CCPwd_{M2}$ and $CCPwd_{L2}$.

$$CCPwd_{M2} = APWD_M \oplus PAD_3 \quad (6.19)$$

$$CCPwd_{L2} = APWD_L \oplus PAD_4 \quad (6.20)$$

These new random numbers and answers are sent to the reader.

Reader: Verify $CCPwd_{M2}$ and $CCPwd_{L2}$. If both values are correct, the tag is authenticated. Otherwise an alarm is raised.

TRMA$^+$ Scheme

In [143], Lim and Li uncovered weaknesses in Konidala and Kim's TRMA scheme. It was found that a passive attacker can recover the tag's access password by eavesdropping over a single run of the protocol and performing correlation analysis on the captured information. In [125], Konidala and Kim proposed an improved version that uses the tag's access and kill passwords. The authors proposed using a PadGen chain of length 2. The outer PadGen is computed over the kill password, while the inner PadGen over the access password. The new scheme is essentially the same as the original TRMA scheme, but the cover-coding pad PAD_i ($i = \{1, 2, 3, 4\}$) is computed differently, as follows:

$$PAD_i = PadGen(PadGen(RN_i^{Tag}, RN_i^{Reader})[APWD], RN_i^{Tag})[KPWD] \quad (6.21)$$

Pad Generation Function - PadGen(.)

The $PadGen$ is a pad generation function that produces a 16-bit pad used to cover-code the two 16-bit access password halves ($APWD_M$ and $APWD_L$). PadGen takes two 16-bit input arguments and operates on a 32-bit password ($KPWD$ or $APWD$) according to the input. The two input arguments are used as location indexes to retrieve individual bits from the access/kill password stored in those locations.

A detailed description of PadGen is provided bellow. The 32-bit $XPWD$ (where $XPWD \in \{APWD,\ KPWD\}$) is represented in binary (or Base 2) as

$$\begin{aligned} XPWD &= XPWD_M \,||\, XPWD_L \\ XPWD_M &= b_0 b_1 b_2 .....b_{13} b_{14} b_{15} \\ XPWD_L &= b_{16} b_{17} b_{18} .....b_{29} b_{30} b_{31} \end{aligned}$$

where each $b_i \in \{0, 1\}$. Let us also represent the 16-bit random numbers RN_i^{Tag} and RN_i^{Rdr} in hexadecimal (or Base 16) representations as

$$\begin{aligned} RN_i^{Tag} &= H_{i,0}^{Tag} H_{i,1}^{Tag} H_{i,2}^{Tag} H_{i,3}^{Tag} \\ RN_i^{Rdr} &= H_{i,0}^{Rdr} H_{i,1}^{Rdr} H_{i,2}^{Rdr} H_{i,3}^{Rdr} \end{aligned}$$

where each $H_{i,j}^{Tag}$ and $H_{i,j}^{Rdr}$ is a hexadecimal digit, i.e. $H_{i,j}^{Tag}, H_{i,j}^{Rdr} \in \boldsymbol{H}_{16} = \{0x0, 0x1, 0x2, ..., 0xD = 13, 0xE = 14, 0xF = 15\}$.

$PadGen(RN_i^{Tag}, RN_i^{Rdr})[XPWD]$ would then be computed as follows:

$$\begin{aligned} & PadGen(RN_i^{Tag}, RN_i^{Rdr})[XPWD] \\ = \; & b_{H_{i,0}^{Tag}} b_{H_{i,1}^{Tag}} b_{H_{i,2}^{Tag}} b_{H_{i,3}^{Tag}} \,||\, b_{H_{i,0}^{Tag}+16} b_{H_{i,1}^{Tag}+16} b_{H_{i,2}^{Tag}+16} b_{H_{i,3}^{Tag}+16} \,|| \\ & b_{H_{i,0}^{Rdr}} b_{H_{i,1}^{Rdr}} b_{H_{i,2}^{Rdr}} b_{H_{i,3}^{Rdr}} \,||\, b_{H_{i,0}^{Rdr}+16} b_{H_{i,1}^{Rdr}+16} b_{H_{i,2}^{Rdr}+16} b_{H_{i,3}^{Rdr}+16} \quad [Base\ 2] \\ = \; & P_0 P_1 P_2 P_3 \quad [Base\ 16] \end{aligned}$$

for some $P_0, P_1, P_2, P_3 \in \boldsymbol{H}_{16}$.

As an example, let us consider $PadGen$(7E2B$_h$, 2B5F$_h$)$[XPWD]$ with $XPWD_M = 1110\ 0101\ 0100\ 1000_2$ and $XPWD_L = 1110\ 1000\ 1100\ 1010_2$.

- 7E2B$_h$ = 7^{th} 14^{th} 2^{nd} 11^{th} location of $XPWD_M$ = 1010_2
- 7E2B$_h$ = 7^{th} 14^{th} 2^{nd} 11^{th} location of $XPWD_L$ = 0110_2
- 2B5F$_h$ = 2^{nd} 11^{th} 5^{th} 15^{th} location of $XPWD_M$ = 1010_2
- 2B5F$_h$ = 2^{nd} 11^{th} 5^{th} 15^{th} location of $XPWD_L$ = 1000_2

Combining the 4 results above, we have a 16-bit pad value $PadGen$(7E2B$_h$, 2B5F$_h$)$[XPWD]$ = 1010 0110 1010 1000 = A6A8$_h$

6.2.2.2 Attacks on TRMA$^+$

In this section we describe how an attacker can recover the 32 bits of the tag's access and kill passwords in the TRMA$^+$ scheme.

Access Password Attack (LSB)

The attack is outlined in the following figure. T, R and A represent the tag, the reader and the attacker respectively.

(1) $T \rightarrow R$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2)	...
(3) $A \rightarrow R$:	$\{EPC, RN_1^{Tag'}, RN_2^{Tag'}\}$
(4) $R \rightarrow A$:	$\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$

Scenario: An adversary eavesdrops on an authentication session between a genuine reader and a genuine tag to obtain a valid EPC. This tag then becomes the target of the attack. With the obtained EPC, the adversary performs an active attack by masquerading as the target tag and participating in the TRMA$^+$ protocol with a genuine reader. The adversary sends the message $\{EPC, RN_1^{Tag'}, RN_2^{Tag'}\}$ to the reader such that all the hexadecimal digits in each of $RN_1^{Tag'}$ and $RN_2^{Tag'}$ have the same value:

$$RN_i^{Tag'} = RRRR_h \quad [Base\ 16] \tag{6.22}$$

where $R \in \boldsymbol{H}_{16}$ and $RN_1^{Tag'}$ may or may not equal $RN_2^{Tag'}$. Next, the adversary receives the response provided by the reader $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \tag{6.23}$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \tag{6.24}$$

and for $i \in \{1, 2\}$,

$$PAD_i = PadGen(PadGen(RN_i^{Tag'}, RN_i^{Rdr})[APWD], RN_i^{Tag'})[KPWD] \tag{6.25}$$

Let $PadGen(RN_i^{Tag'}, RN_i^{Rdr})[APWD] = V_0V_1V_2V_3$ for some hexadecimal digits $V_0, V_1, V_2, V_3 \in \boldsymbol{H}_{16}$. Substituting this and (6.22) into (6.25), we have

$$\begin{aligned} PAD_i &= PadGen(V_0V_1V_2V_3, RRRR)[KPWD] \quad [Base\ 16] \\ &= k_{V_0}k_{V_1}k_{V_2}k_{V_3} \,||\, k_{V_0+16}k_{V_1+16}k_{V_2+16}k_{V_3+16} \,||\, k_Rk_Rk_Rk_R \,|| \\ &\quad k_{R+16}k_{R+16}k_{R+16}k_{R+16} \quad [Base\ 2] \\ &= P_0P_1P_2P_3 \quad [Base\ 16] \end{aligned}$$

where each k_j is the j^{th} bit in the kill password. We observe that all the bits in each of the hexadecimal digits P_2 and P_3 are the same, i.e. $P_2, P_3 \in \{0000_b = 0_h,\ 1111_b = \mathrm{F}_h\}$. This leads to $P_2P_3 \in \{00_h, 0\mathrm{F}_h, \mathrm{F}0_h, \mathrm{FF}_h\}$. Assuming that P_2P_3 takes each value with equal probability, the adversary can then use this to obtain the 8 least significant bits of $APWD_L$ and

$APWD_M$ by computing the following:

$$APWD_M[8...15] = \begin{cases} CCPwd_{M1}[8...15] \oplus \text{0x00} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0x0F} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xF0} & \text{with } p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xFF} & \text{with } p = 2^{-2} \end{cases} \tag{6.26}$$

$$APWD_L[8...15] = \begin{cases} CCPwd_{L1}[8...15] \oplus \text{0x00} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0x0F} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0xF0} & \text{with } p = 2^{-2} \\ CCPwd_{L1}[8...15] \oplus \text{0xFF} & \text{with } p = 2^{-2} \end{cases} \tag{6.27}$$

Summarizing, the adversary can obtain the 8 least significant bits of APW_M and APW_L with probability 2^{-2} for each. The attack is more powerful if the random numbers are such that $RN_1^{Tag\prime} = RN_2^{Tag\prime}$. Under this condition, an adversary can also extract the following 16 bits of the access password with probability 2^{-2}:

$$\begin{aligned} &APWD_M[8...15] \,||\, APWD_L[8...15] \\ &= \begin{cases} CCPwd_{M1}[8...15] \oplus \text{0x00} \,||\, CCPwd_{L1}[8...15] \oplus \text{0x00} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0x0F} \,||\, CCPwd_{L1}[8...15] \oplus \text{0x0F} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xF0} \,||\, CCPwd_{L1}[8...15] \oplus \text{0xF0} & p = 2^{-2} \\ CCPwd_{M1}[8...15] \oplus \text{0xFF} \,||\, CCPwd_{L1}[8...15] \oplus \text{0xFF} & p = 2^{-2} \end{cases} \end{aligned} \tag{6.28}$$

Hence an active attacker can gather vast amounts of information about the tag's access password in a single run of the TRMA$^+$ protocol.

Access Password Attack (MSB)
The attack is outlined in the following figure. Details are provided below.

(1) $T \rightarrow A$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(2a) $A \rightarrow R$:	$\{EPC, RN, RN\}$
(2b) $R \rightarrow A$:	$\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$
(3a) $A \rightarrow R$:	$\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$
(3b) $R \rightarrow A$:	$\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr\prime}, RN_2^{Rdr\prime}, RN_3{}^{Rdr\prime}, RN_4{}^{Rdr\prime}\}$
(4) $A \rightarrow T$:	$\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr\prime}, RN_2^{Rdr\prime}, RN, RN\}$
(5) $T \rightarrow A$:	$\{RN_3^{Tag}, RN_4^{Tag}, CCPwd_{M2}, CCPwd_{L2}\}$

Scenario: An adversary intercepts and modifies the content of the message sent by a genuine tag. The random numbers picked up by the adversary are then set to RN before being forwarded to the reader. Specifically, the random number RN must satisfy the following equation:

$$RN = RRRR_h \quad [Base\ 16] \tag{6.29}$$

where $R \in \boldsymbol{H}_{16}$. The adversary receives the response provided by the legitimate reader: $\{CCPwd_{M1}, CCPwd_{L1}, RN_1^{Rdr}, RN_2^{Rdr}, RN_3^{Rdr}, RN_4^{Rdr}\}$, where

$$CCPwd_{M1} = APWD_M \oplus PAD_1 \tag{6.30}$$

$$CCPwd_{L1} = APWD_L \oplus PAD_2 \tag{6.31}$$

and for $i \in \{1, 2\}$,

$$PAD_i = PadGen(PadGen(RN, RN_i^{Rdr})[APWD], RN)[KPWD] \tag{6.32}$$

In a different, parallel authentication session, the adversary forwards the initial message sent by the tag $\{EPC, RN_1^{Tag}, RN_2^{Tag}\}$ to the legitimate reader. The reader's response $\{CCPwd_{M1}{}', CCPwd_{L1}{}', RN_1^{Rdr'}, RN_2^{Rdr'}, RN_3^{Rdr'}, RN_4^{Rdr'}\}$ is received by the adversary, who then sets the random numbers $RN_3^{Rdr'}$ and $RN_4^{Rdr'}$ to RN. (Note that these two random numbers will be used by the tag to compute its response to the reader). The modified message is then forwarded to the genuine tag, which responds by sending the message: $\{CCPwd_{M2}, CCPwd_{L2}, RN_3^{Tag}, RN_4^{Tag}\}$, where

$$CCPwd_{M2} = APWD_M \oplus PAD_3 \tag{6.33}$$

$$CCPwd_{L2} = APWD_L \oplus PAD_4 \tag{6.34}$$

and for $i \in \{3, 4\}$,

$$PAD_i = PadGen(PadGen(RN_i^{Tag}, RN)[APWD], RN_i^{Tag})[KPWD] \tag{6.35}$$

In such an attack scenario, the adversary can derive the following:

1. Information from the computation of PAD$_{i\in\{1,2\}}$

$$\begin{aligned}
& PadGen(RN, RN_i^{Rdr})[APWD] \\
= \;& PadGen(RRRR, H_{i,0}^{Rdr}H_{i,1}^{Rdr}H_{i,2}^{Rdr}H_{i,3}^{Rdr})[APWD] \quad [Base\ 16] \\
= \;& a_R a_R a_R a_R \,||\, a_{R+16}a_{R+16}a_{R+16}a_{R+16} \,||\, a_{H_{i,0}^{Rdr}}a_{H_{i,1}^{Rdr}}a_{H_{i,2}^{Rdr}}a_{H_{i,3}^{Rdr}} \,||\, \\
& a_{H_{i,0}^{Rdr}+16}a_{H_{i,1}^{Rdr}+16}a_{H_{i,2}^{Rdr}+16}a_{H_{i,3}^{Rdr}+16} \quad [Base\ 2] \\
= \;& V_0V_1V_2V_3 \quad [Base\ 16] \qquad (6.36)
\end{aligned}$$

where we observe that all four bits in each of V_0 and V_1 have the same value, i.e. $V_0, V_1 \in \{0_h, \mathrm{F}_h\}$ or $V_0V_1 \in \{00_h, 0\mathrm{F}_h, \mathrm{F}0_h, \mathrm{FF}_h\}$ (as in the previous attack on the LSB). Then,

$$\begin{aligned}
& PAD_{i\in\{1,2\}} \\
= \;& PadGen(PadGen(RN, RN_1^{Rdr})[APWD], RN)[KPWD] \\
= \;& PadGen(V_0V_1V_2V_3, RRRR)[KPWD] \quad [Base\ 16] \\
= \;& k_{V_0}k_{V_1}k_{V_2}k_{V_3} \,||\, k_{V_0+16}k_{V_1+16}k_{V_2+16}k_{V_3+16} \,||\, k_R k_R k_R k_R \,||\, \\
& k_{R+16}k_{R+16}k_{R+16}k_{R+16} \quad [Base\ 2] \qquad (6.37)
\end{aligned}$$

Assuming that the values of V_0 and V_1 are taken randomly from the set $\{0_h, \mathrm{F}_h\}$, they would be equal half of the time, i.e. with probability 0.5. If $V_0 = V_1$, then $k_{V_0} = k_{V_1}$. On the other hand, if $V_0 \neq V_1$, then assuming that the bits in the kill password are perfectly random, we would have $k_{V_0} = k_{V_1}$ with probability 0.5. Hence:

$$Prob(k_{V_0} = k_{V_1}) \;=\; (0.5)(1) + (0.5)(0.5) = 0.75 \qquad (6.38)$$

Similarly, $Prob(k_{V_0+16} = k_{V_1+16}) = 0.75$.

2. Information from the computation of PAD$_{i\in\{3,4\}}$

$$\begin{aligned}
& PadGen(RN_i^{Tag}, RN)[APwd] \\
= \;& PadGen(H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag}, RRRR)[APWD] \quad [Base\ 16] \\
= \;& a_{H_{i,0}^{Tag}}a_{H_{i,1}^{Tag}}a_{H_{i,2}^{Tag}}a_{H_{i,3}^{Tag}} \,||\, a_{H_{i,0}^{Tag}+16}a_{H_{i,1}^{Tag}+16}a_{H_{i,2}^{Tag}+16}a_{H_{i,3}^{Tag}+16} \,||\, \\
& a_R a_R a_R a_R \,||\, a_{R+16}a_{R+16}a_{R+16}a_{R+16}[Base\ 2] \\
= \;& S_0S_1S_2S_3 \quad [Base\ 16] \qquad (6.39)
\end{aligned}$$

where $S_2, S_3 \in \{0_h, \text{F}_h\}$ or $S_2S_3 \in \{00_h, 0\text{F}_h, \text{F}0_h, \text{FF}_h\}$. Furthermore, we note that $V_0 = S_2$ and $V_1 = S_3$. Next, we derive

$$\begin{aligned}
& PAD_{i\in\{3,4\}} \\
= \;& PadGen(PadGen(RN_i^{Tag}, RN)[APWD], RN_i^{Tag})[KPWD] \\
= \;& PadGen(S_0S_1S_2S_3, H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag})[KPWD] \quad [Base\ 16] \\
= \;& k_{S_0}k_{S_1}k_{S_2}k_{S_3} \;||\; k_{S_0+16}k_{S_1+16}k_{S_2+16}k_{S_3+16} \;||\; k_{H_{i,0}^{Tag}}k_{H_{i,1}^{Tag}}k_{H_{i,2}^{Tag}}k_{H_{i,3}^{Tag}} \;||\; \\
& k_{H_{i,0}^{Tag}+16}k_{H_{i,1}^{Tag}+16}k_{H_{i,2}^{Tag}+16}k_{H_{i,3}^{Tag}+16} \quad [Base\ 2] \qquad (6.40)
\end{aligned}$$

As in the earlier case for the computation of $PAD_{i\in\{1,2\}}$, assuming that $S_2 = S_3$ half of the time, we would have $Prob(k_{S_2} = k_{S_3}) = 0.75$ and $Prob(k_{S_2+16} = k_{S_3+16}) = 0.75$.

3. Combining both sets of information

Since $V_0 = S_2$ and $V_1 = S_3$, we then have

$$\begin{aligned}
k_{V_0} = k_{S_2} \;&=\; k_{V_1} = k_{S_3} \quad p = 0.75 \\
k_{V_0} = k_{S_2} \;&\neq\; k_{V_1} = k_{S_3} \quad p = 0.25
\end{aligned}$$

and

$$\begin{aligned}
k_{V_0+16} = k_{S_2+16} \;&=\; k_{V_1+16} = k_{S_3+16} \quad p = 0.75 \\
k_{V_0+16} = k_{S_2+16} \;&\neq\; k_{V_1+16} = k_{S_3+16} \quad p = 0.25
\end{aligned}$$

However, instead of considering these two sets of relations separately, we combine them to give four possible cases. Their corresponding probabilities are computed as follows:

- **Case 1:** $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$.

  The two relations will always hold if $V_0 = V_1$ (which also implies $S_2 = S_3$). When $V_0 \neq V_1$ (and $S_2 \neq S_3$), the probability that $k_{V_0} = k_{V_1}$ (and $k_{S_2} = k_{S_3}$) is 0.5. Similarly, the probability that $k_{V_0+16} = k_{V_1+16}$ (and $k_{S_2+16} = k_{S_3+16}$) is also 0.5. Hence the probability that this case will occur is $(0.5)(1) + (0.5)(0.5)(0.5) = 0.625$.

- **Case 2:** $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} =$

k_{S_3+16}.

This case will only occur when $V_0 \neq V_1$ (and $S_2 \neq S_3$). In such a situation, the probability that $k_{V_0} = k_{V_1}$ (and $k_{S_2} = k_{S_3}$) is 0.5, and the probability that $k_{V_0+16} \neq k_{V_1+16}$ (and $k_{S_2+16} \neq k_{S_3+16}$) is 0.5. Hence the probability that the two relations will hold is $(0.5)(0.5)(0.5) = 0.125$.

- **Case 3:** $k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$.

  This case is similar to Case 2 and occurs when $V_0 \neq V_1$ ($S_2 \neq S_3$), $k_{V_0} \neq k_{V_1}$ ($k_{S_2} \neq k_{S_3}$) but $k_{V_0+16} = k_{V_1+16}$ ($k_{S_2+16} = k_{S_3+16}$). The resulting probability for this case is $(0.5)(0.5)(0.5) = 0.125$.

- **Case 4:** $k_{V_0} = k_{S_2} \neq k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} \neq k_{V_1+16} = k_{S_3+16}$.

  This case is also similar to Case 2 and occurs when $V_0 \neq V_1$ ($S_2 \neq S_3$), $k_{V_0} \neq k_{V_1}$ ($k_{S_2} \neq k_{S_3}$) and $k_{V_0+16} \neq k_{V_1+16}$ ($k_{S_2+16} \neq k_{S_3+16}$). It yields a probability of $(0.5)(0.5)(0.5) = 0.125$.

Based on this information, the 8 most significant bits of $APWD_M$ and $APWD_L$ can be given by

$$APWD_M[0...7] \,||\, APWD_L[0...7] \;=\; A \oplus mask \,||\, B \oplus mask \quad (6.41)$$

where

$$A \;=\; (CCPwd_{M1}[0..7] \wedge \text{0xCC}) \vee (CCPwd_{M2}[0..7] \wedge \text{0x33}) \quad (6.42)$$

$$B \;=\; (CCPwd_{L1}[0..7] \wedge \text{0xCC}) \vee (CCPwd_{L2}[0..7] \wedge \text{0x33}) \quad (6.43)$$

and $\wedge$ denotes the bitwise logical AND operation, $\vee$ denotes the bitwise logical OR operation. The $mask$ in (6.41) can take a number of probable values depending on whether Case 1, 2, 3 or 4 holds:

- If **Case 1** holds, i.e. $k_{V_0} = k_{S_2} = k_{V_1} = k_{S_3}$ and $k_{V_0+16} = k_{S_2+16} = k_{V_1+16} = k_{S_3+16}$, then $mask \in \{$0x00, 0x0F, 0xF0, 0xFF$\}$. In this case, if the adversary were to select a mask from the specified set of values, the probability of a successful attack to recover all 16 bits of the access

password would be

$$\begin{aligned} & Prob(\text{successful recovery of all bits in } APWD_M[0...7] \,||\, APWD_L[0...7]) \\ = \;& 0.625 \times 1/4 \\ = \;& 0.15625 \end{aligned} \tag{6.44}$$

- If **Case 2** holds, then $mask \in$ {0x05, 0x0A, 0xF5, 0xFA} and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.
- If **Case 3** holds, then $mask \in$ {0x50, 0x5F, 0xA0, 0xAF} and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.
- If **Case 4** holds, then $mask \in$ {0x55, 0x5A, 0xA5, 0xAA} and the probability of a successful attack would be $0.125 \times 1/4 = 0.03125$.

In summary, with equations (6.41), (6.42) and (6.43), the probability of a successful attack for any selected mask would be given by

$$Prob(\text{successful recovery of all bits in } APWD_M[0...7] \,||\, APWD_L[0...7])$$
$$= \begin{cases} \frac{5}{2^5} = 0.15625 & \text{if} \quad mask \in \{\text{0x00, 0x0F, 0xF0, 0xFF}\} \\ \frac{1}{2^5} = 0.03125 & \text{if} \quad mask \in \{\text{0x05, 0x0A, 0x50, 0x55, 0x5A, 0x5F, 0xA0, 0xA5,0xAA, 0xAF 0xF5, 0xFA}\} \end{cases}$$

Hence, in order to maximize the probability of success of an attack, the adversary should select $mask$ from the set {0x00, 0x0F, 0xF0, 0xFF}. In any case, this attack results in 16 possible values for the most significant bits of $APWD_M$ and $APWD_L$. Together with the 4 possible values for the least significant bits of $APWD_M$ and $APWD_L$ obtained from the earlier attack, the adversary can narrow down the possible values for the access password from 2^{32} to $16 \times 4 = 2^6$, which is a tremendous reduction.

Kill Password Attack

We have shown how an attacker is able to obtain the 8 least significant bits of $APWD_M$ and $APWD_L$, each with probability 2^{-2}. This advantage can be employed by an adversary to recover the full 32 bits of the kill password with the same probability. The attack is described below.

Given that the adversary knows the access password of the target tag, the pads used to cover-code the MSB and LSB of the access password can be obtained as follows:

$$PAD_1[8...15] = CCPwd_{M1}[8...15] \oplus APWD_M[8...15] \quad (6.45)$$
$$PAD_2[8...15] = CCPwd_{L1}[8...15] \oplus APWD_L[8...15] \quad (6.46)$$
$$PAD_3[8...15] = CCPwd_{M2}[8...15] \oplus APWD_M[8...15] \quad (6.47)$$
$$PAD_4[8...15] = CCPwd_{L2}[8...15] \oplus APWD_L[8...15] \quad (6.48)$$

where the 8 bits in each pad PAD_i are bits selected from different memory locations in the kill password:

$$\begin{aligned} PAD_i &= PadGen(----, RN_i^{Tag})[KPWD] \\ &= PadGen(----, H_{i,0}^{Tag}H_{i,1}^{Tag}H_{i,2}^{Tag}H_{i,3}^{Tag})[KPWD] \quad [Base\ 16] \end{aligned}$$

Hence,

$$\begin{aligned} PAD_i[8...15] = {} & k_{H_{i,0}^{Tag}}k_{H_{i,1}^{Tag}}k_{H_{i,2}^{Tag}}k_{H_{i,3}^{Tag}} \,||\, \\ & k_{H_{i,0}^{Tag}+16}k_{H_{i,1}^{Tag}+16}k_{H_{i,2}^{Tag}+16}k_{H_{i,3}^{Tag}+16} \quad [Base\ 2] \end{aligned}$$

We now have the following equations relating a bit in the kill password to a bit in each PAD_i $(i \in \{1,2,3,4\})$:

$$\begin{aligned} k_{H_{i,0}^{Tag}} &= PAD_i[8] & k_{H_{i,0}^{Tag}+16} &= PAD_i[12] \\ k_{H_{i,1}^{Tag}} &= PAD_i[9] & k_{H_{i,1}^{Tag}+16} &= PAD_i[13] \\ k_{H_{i,2}^{Tag}} &= PAD_i[10] & k_{H_{i,2}^{Tag}+16} &= PAD_i[14] \\ k_{H_{i,3}^{Tag}} &= PAD_i[11] & k_{H_{i,3}^{Tag}+16} &= PAD_i[15] \end{aligned}$$

where each bit $PAD_i[n]$ can be computed using one of equations (6.45), (6.46), (6.47) or (6.48). For example, $k_{H_{1,1}^{Tag}} = PAD_1[9] = CCPwd_{M1}[9] \oplus APWD_M[9]$ and $k_{H_{4,0}^{Tag}+16} = PAD_4[12] = CCPwd_{L2}[12] \oplus APWD_L[12]$. Hence, once an adversary has obtained the LSB of the access password, he can obtain the bits in the kill password. For a complete recovery of the entire kill password, there are two possible approaches:

Passive Attacker In this case, an adversary eavesdrops over multiple ses-

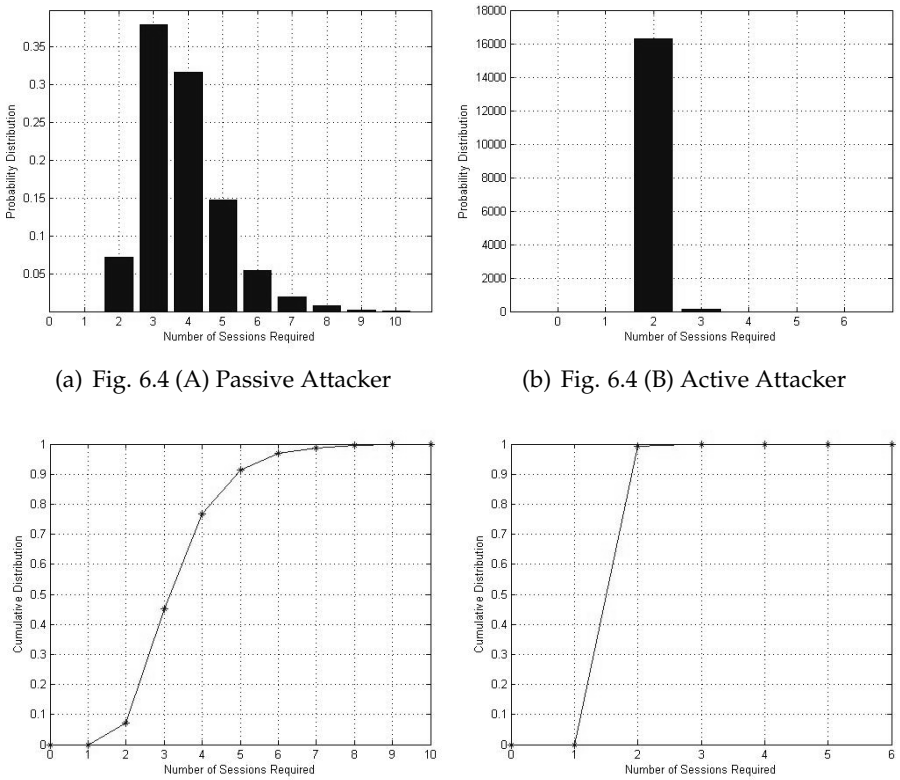
sions of the protocol in order to obtain the full 32 bits of the kill password. From our experiments, in which we simulated 20,000 executions of the attack, we find that the average number of sessions required to obtain the full kill password is 4. *Fig. 6.4 (A)* and *6.4 (C)* show the probability distribution and the cumulative distribution for the number of sessions required.

Active Attacker In the active attack, the adversary modifies and manipulates the random numbers RN_1^{Tag} and RN_2^{Tag} to lead the legitimate reader to select bits in the kill password such that those bit locations where the value of the bit is already known are avoided. Hence, the number of sessions required to obtain the full 32 bits of the kill password would be reduced. From the results of our experiments, we find that the average number of attack sessions required to obtain the full password is 2, i.e. a 50% reduction compared to the passive attack. *Fig. 6.4 (B)* and *6.4 (D)* show the results.

6.2.2.3 Additional Comments

The TRMA+ scheme uses the access and kill passwords defined in the EPC specification, these being shared between legitimate entities (tags and readers). The authors suggested the use of a PadGen chain to protect both passwords. This function, however, is not secure enough, as the cover codes generated depend on random numbers selected by the tag/reader. We have found that there is a high probability of an attacker acquiring the access and kill passwords after some computations. The most and least significant eight bits of the access password can be obtained with a probability of 2^{-5} and 2^{-2} respectively. Once the attacker knows the LSB of the access password, the 32 bits of the kill password can be derived with a probability of 2^{-2}. The efficiency of the attack on the kill password depends on whether the attack is passive or active. A passive attacker has to eavesdrop on an average of 4 protocol rounds, and this number is reduced to 2 when the attacker can modify and manipulate the exchanged messages.

In summary, the security of the EPC-C1G2 specification is inadequate. So far, most of the proposals that aim to increase security whilst conforming



(a) Fig. 6.4 (A) Passive Attacker

(b) Fig. 6.4 (B) Active Attacker

(c) Fig. 6.4 (C) Passive Attacker

(d) Fig. 6.4 (D) Active Attacker

Figure 6.4: Probability and cumulative distributions for the number of sessions required for a successful attack.

to the standard have failed. Providing greater security in future standards is indeed an important challenge.

6.3 Passive Attacks

We believe, as do many other authors, that the security of low-cost RFID tags can be greatly improved without using classic cryptographic primitives (i.e. block/stream ciphers, hash functions, etc.). This is why we present Gossamer, a new protocol inspired in the recently published SASI scheme [46], which constituted an advancement on the UMAP family of protocols. Despite doing so, SASI was not designed with sufficient care and a passive attacker can obtain the secret static identifier of the tag (ID)

after observing several consecutive authentication sessions, as shown in *Section 6.3.2.1.*

6.3.1 A Family of Ultralightweight Mutual Authentication Protocols

In 2006, Peris et al. proposed a family of Ultralightweight Mutual Authentication Protocols (henceforth referred to as the UMAP family of protocols). Chronologically, M^2AP [163] was the first proposal, followed by EMAP [162] and LMAP [161].

These protocols are based on the use of pseudonyms to guarantee tag anonymity. Specifically, an index-pseudonym is used by an authorized reader to retrieve the information associated with a tag (tag identification phase). Additionally, a key -divided in several subkeys- is shared between legitimate tags and readers (back-end database). Both readers and tags use these subkeys to construct the messages exchanged in the mutual authentication phase.

In line with their real processing capabilities, tags only support on-board simple operations. Indeed, these protocols are based on bitwise XOR ($\oplus$), bitwise OR ($\vee$), bitwise AND ($\wedge$) and addition mod 2^m. By contrast, only readers need to generate pseudorandom numbers; tags only used them for creating fresh messages to the protocol.

In the UMAP family of protocols, the proposed scheme consists of three stages. First, the tag is identified by means of the index-pseudonym. Secondly, the reader and the tag are mutually authenticated. This phase is also used to transmit the static tag identifier (ID) securely. Finally, the index-pseudonym and keys are updated (the reader is referred to the original papers for more details).

6.3.1.1 Security Analysis of the UMAP Protocols

Since the publication of the UMAP family of protocols, their security has been analyzed in depth by the research community. In [140, 141] a desynchronization attack and a full disclosure attack are presented. These require an active attacker and several incomplete run executions of the protocol to

disclose the secret information on the tag. Later, Chien et al. proposed - based on the same attack model- a far more efficient full-disclosure attack [48]. Additionally, Bárász et al. showed how a passive attacker (an attack model that may be, in certain scenarios, much more realistic) can find out the static identifier and particular secrets shared by the reader and the tag after eavesdropping on a few consecutive protocol rounds [28, 29].

This leads us to the following conclusions: first, we must define what kind of attack scenarios are applicable. In our opinion, ultralightweight RFID tags have to be resistant to passive attacks but not necessarily to active attacks, because of their severe restrictions (storage, circuitry and power consumption, etc.). Regarding passive attacks, we can affirm the following:

- The UMAP family of protocols is based on the composition of simple operations like bitwise AND, XOR, OR and sum mod 2^m. Because all of these are triangular functions (T-functions) [123], the information does not propagate well from left to right. In other words, the bit in position i in the output only depends on bits j = 0,..., i of the input words.
- The use of the bitwise AND or OR operations to build public submessages is a weakness common to all these protocols. When a bitwise AND (OR) operation is computed even over random inputs, the probability of obtaining a one (zero) is $\frac{3}{4}$. In other words, the result is strongly biased. This poor characteristic is the basis of all the passive attacks proposed so far.

6.3.2 SASI Protocol

In 2007 Chien proposed a very interesting ultralightweight authentication protocol providing Strong Authentication and Strong Integrity (SASI) for very low-cost RFID tags [46]. We briefly describe the messages exchanged between the reader (or back-end database) and the tag (see *Figure 6.5*).

An index-pseudonym (IDS), the tag's private identification (ID), and two keys (k_1/k_2) are stored both on the tag and in the back-end database. Simple bitwise XOR ($\oplus$), bitwise AND ($\wedge$), bitwise OR ($\vee$), addition 2^m and left rotate (Rot(x,y)) are required on the tag. Additionally, random number gen-

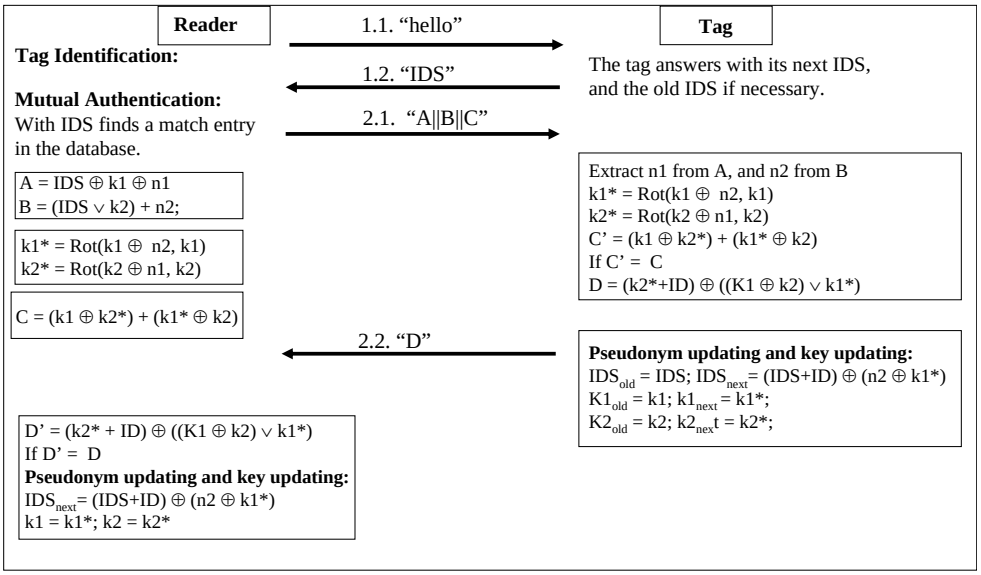


Figure 6.5: SASI Protocol

eration (i.e. n_1 and n_2) is required on the reader. The protocol is divided into three states: tag identification, mutual authentication and updating phase. In the identification phase, the reader (R) sends a "hello" message to the tag (T), and the tag answers with its IDS. The reader then finds, in the back-end database, the information associated with the tag (ID and k_1/k_2), and the protocol continues to the mutual authentication phase. In this, the reader and the tag authenticate each other, and the index-pseudonym and keys are subsequently updated:

R $\rightarrow$ **T** : $A||B||C$

The reader generates nonces n_1 and n_2 to build the submessages as follows:
$A = IDS \oplus k_1 \oplus n_1$; $B = (IDS \vee k_2) + n_2$; $C = (k_1 \oplus k_2^*) + (k_2 \oplus k_1^*)$;
where $k_1^* = Rot(k_1 \oplus n_2, k_1)$; $k_2^* = Rot(k_2 \oplus n_1, k_2)$;

Tag From messages A and B, the tag can obtain values n_1 and n_2 respectively. Then it locally computes C' and checks if the result is equal to the received value. If this is the case, it sends D and updates the values of IDS, k_1 and k_2:
$D = (k_2^* + ID) \oplus ((k_1 \oplus k_2) \vee k_1^*)$;
$IDS^{next} = (IDS + ID) \oplus (n_2 \oplus k_1^*)$; $k_1^{next} = k_1^*$; $k_2^{next} = k_2^*$;

$\mathbf{T} \rightarrow \mathbf{R} : D$

Reader Verifies D and, if it is equal to the result of its local computation, updates IDS, k_1 and k_2 in the same way as the tag.

6.3.2.1 Cryptanalysis of the SASI Protocol

It is important to note that the way in which rotations should be performed is not specified in the original paper. The first researchers to publish a weakness (two desynchronization attacks) in the protocol needed to contact the author to clarify this [204]. After a private communication, the author stated that the rotation used in the protocol is:

- $Rot(A, B) = A << wht(B)$, where $wht(B)$ stands for the Hamming weight of vector B, instead of the more natural and common definition:
- $Rot(A, B) = A << (B \bmod N)$ for a given value of N (96 in our case).

An important observation is that the number of positions rotated, which is guided by the Hamming weight of its second argument, is far from being uniform. In fact, if we assume this second argument B to be random, then the probability that the rotation amount takes value k is given by the formula:

$$Prob(wht(B) = k) = \frac{\binom{96}{k}}{2^{96}}$$

which attains a maximum for $k = \frac{96}{2} = 48$ with an associated probability of 0.0812219, or around 8% of the times.

Firstly, for simplicity, we will work with the second definition of Rot, i.e. the usual one. We will then be able to show how both definitions result in the same security weaknesses, and that the attack presented here can reveal the secret ID in both cases.

6.3.2.2 Analytical Results

From the analysis of the UMAP family of protocols, we conclude that it is necessary to incorporate a non-triangular function in order to increase the

security of ultralightweight protocols. At first sight, the SASI protocol complies with this requirement as it includes the left rotate operation (which is non triangular). However, a logical way of attacking SASI is to consider what happens when rotations are not performed, that is, when the amount of rotation given by the second argument is zero modulo 96. In these cases, the proposed protocol uses exactly the same set of operations that enable attacks on the UMAP family of protocols. We therefore have:

$$\begin{aligned} k_1^* &= Rot(k_1 \oplus n_2, k_1) = Rot(k_1 \oplus n_2, k_1 \bmod 96) = \\ &= Rot(k_1 \oplus n_2, 0) = k_1 \oplus n_2 \end{aligned} \tag{6.49}$$

Similarly,

$$k_2^* = Rot(k_2 \oplus n_1, k_2) = k_2 \oplus n_1 \tag{6.50}$$

This has a particularly adverse impact on the index pseudonym (IDS) update process, since:

$$\begin{aligned} IDS^{next} &= (IDS + ID) \oplus (n_2 \oplus k_1^*) = (IDS + ID) \oplus (n_2 \oplus\ k_1 \oplus n_2) \\ &= (IDS + ID) \oplus k_1 \end{aligned} \tag{6.51}$$

So we are left with $ID = IDS^{next} \oplus k_1 - IDS$ and we can take full advantage of the knowledge that $k_1 = k_2 = 0 \bmod 96$ to conclude that, with a given probability (see *Table 6.2*), which in turn only depends on the value of N ($N = 96$ in this case, although other values could be used for recovering more bits), it holds that:

$$ID \bmod 96 \approx (IDS^{next} - IDS) \bmod 96 \tag{6.52}$$

As both values IDS^{next} and IDS are public and easily observable by snooping on two consecutive authentication sessions, relation (6.52) allows us to recover the $log_2(96) \approx 6.58$ less significant bits of the secret ID.

The only question that remains is recognizing when the conditions $k_1 = 0 \bmod 96$ and $k_2 = 0 \bmod 96$ hold, since k_1 and k_2 are secrets that only the tag and the reader should know. Fortunately, this is possible by simply checking if certain relations hold that only involve public values. If $K_1 =$

$K_2 = 0 \bmod 96$ then expressions (6.49) and (6.50) hold, so:

$$C \quad = (k_1 \oplus k_2^*) + (k_2 \oplus k_1^*) = k_1 \oplus k_2 \oplus n_1 + k_2 \oplus k_1 \oplus n_2 \tag{6.53}$$

which implies that:

$$C \bmod 96 \quad = k_1 \oplus k_2 \oplus n_1 + k_2 \oplus k_1 \oplus n_2 \bmod 96 \approx n_1 + n_2 \bmod 96 \tag{6.54}$$

The value of $n_1 + n_2 \bmod 96$ can also be probabilistically obtained from the observed values of public messages A, B and IDS because:

$$A = IDS \oplus k_1 \oplus n_1 \Rightarrow n_1 = A \oplus IDS \oplus k_1 \tag{6.55}$$

and then we can get that:

$$n_1 \bmod 96 \quad = A \oplus IDS \oplus k_1 \bmod 96 \approx A \oplus IDS \bmod 96 \tag{6.56}$$

since by hypothesis, $k_1 = 0 \bmod 96$.

Similarly, we can obtain that, as $B = (IDS \vee k_2) + n_2$, then:

$$n_2 \approx (B - IDS) \bmod 96 \tag{6.57}$$

Summarizing, we can conclude that if $k_1 = k_2 = 0 \bmod 96$ then, with the probability given in *Table 6.2*:

$$C \bmod 96 \quad \approx n_1 + n_2 \bmod 96 \approx (A \oplus IDS) + (B - IDS) \bmod 96 \tag{6.58}$$

What remains is to passively snoop on multiple authentication sessions and, for each one, verify if condition (6.58) holds. If this is the case, one can compute the value $(IDS^{next} - IDS) \bmod 96$ and, from this, directly approximate $ID \bmod 96$.

Only one last tweak is needed to perform a successful attack: just by chance, expression (6.58) will be true even if the two preconditions $k_1 = 0 \bmod 96$ and $k_2 = 0 \bmod 96$ are not simultaneously true, and this will lead us to a possibly wrong estimation for $ID \bmod 96$. This is, however, easily fixed by simply observing many values of $(IDS_{next} - IDS) \bmod 96$ when equation (6.58) holds, since the true value of $ID \bmod 96$ is likely to be the

Table 6.2: Probabilities of expressions (6.52), (6.54), (6.56), (6.57) and (6.58) simultaneously holding for different forms of N, given that $K_1 = K_2 = 0 \bmod N$

N	2^t	$3 * 2^t$	$4 * t + 10$	$2 * t + 5$
Probability	1.00	0.33	$2 * N^{-1}$	N^{-1}

1. For $i = 0$ to 96
2. $\quad Observations[i] = 0$
3. Repeat a sufficiently high number of times the following steps:
4. $\quad$ Observe an authentication session and get IDS, A, B and C
5. $\quad$ Check expression (6.58) for these values
6. $\quad$ If this is not the case, go to step 4
7. $\quad$ Perform the following tasks:
8. $\quad\quad$ Wait for the authentication session to finish
9. $\quad\quad$ Send the tag a hello message to obtain IDS^{next}
10. $\quad\quad$ Compute $x = (IDS^{next} - IDS) \bmod 96$
11. $\quad\quad$ Increment $Observations[x]$
12. Find m, the maximum of the values in $Observations$
13. Conjecture that $m = ID \bmod 96$

Figure 6.6: Outline of the Attack

most common. This fact has been experimentally verified and leads to the attack described in *Figure 6.6*.

6.3.2.3 Efficiency Analysis

The attack presented could be performed not only for recovering $log_2(96)$ bits of the secret value ID, but also works for other moduli, with varying probabilities as in *Table 6.2.* In particular, expressions (6.52), (6.54), (6.56), (6.57) and (6.58) all hold with probability one for moduli that are a power of two, so this allows us to recover many more bits (i.e. $log_2(256) = 8$, $log_2(512) = 9$, $log_2(1024) = 10$, etc.) if necessary. In these cases, we need, of course, to observe more authentication sessions to recover more ID bits.

As a rule of thumb guide, we have concluded, after extensive experimentation, that an attacker following this procedure is able, on average, to recover the $log_2(S)$ least significant bits of ID after observing around $\mathcal{O}(S)$

Table 6.3: Attack Success Probabilities

Number of sessions	Prob. Modular rotation	Prob. Hamming rotation
2^5	0.08	0.12
2^6	0.09	0.12
2^7	0.16	0.19
2^8	0.19	0.25
2^9	0.5	0.36
2^{10}	0.5	0.53
2^{11}	0.76	0.58
2^{12}	0.93	0.74
2^{13}	0.97	0.92
2^{14}	1.00	0.99

authentication sessions.

6.3.2.4 The Case of the Hamming Rotation

We have found through experiment that exactly the same approach as described above can break the version of this protocol that uses rotation to the left by the amount given by the hamming weight of its second argument. In this case, the efficiency of the attack is is slightly less efficient –see *Table 6.2* for an exact description of the probabilities of the main equations to hold. Some overall figures are given in *Table 6.3*, which shows the probability of success for recovering $5 = log_2(32)$ ID bits for different rotation definitions.

6.3.2.5 Additional Remarks

We have presented an attack against a new and quite interesting ultra-lightweight authentication protocol. Even though the definition of the rotation in the original paper was a little obscure, we have found experimentally that the attack is successful for both definitions.

The inclusion of rotations in these very lightweight protocols appears necessary to guarantee security, but it is not enough in itself. Certain changes in the design would, however, have hindered our attack considerably. We briefly describe these:

- The IDS update should be improved as it is dependent on n_2 and k_1^*,

which is again a function of n_2. This is instrumental in our attack and, in any case, leads to a variety of poor statistical properties.

- The definition of k_1^* and k_2^* should be rethought, as in their current form there is a kind of distributive property ($k_1^* = Rot(k_1 \oplus n_2, k_1) = Rot(k_1, k_1) \oplus Rot(n_2, k_1)$) that could facilitate attacks. This could be avoided by, for example, using addition instead of XOR as the inner operation, though part of the problem remains. Ideally, a more complex key scheduling should be devised, but of course this will mean additional cost.

- The use of the bitwise OR operation should be performed with extreme care, as the resulting messages are strongly biased. As an example, in the current protocol definition, n_2 could be approximated with very good precision simply by computing $n_2 \approx B - 1$. Note that message D suffers from a similar problem.

- The use of bitwise AND operation would produce similar undesirable effects. These two operations should only be included in the inner parts of the algorithm, and every effort should be made to disguise their output into seemingly random output when constructing public messages such as B and D.

In fact, a more general version of this attack is even possible. This alternative is, however, significantly less efficient than the attack scheme described previously. It simply consists of observing and storing the different values of equation (6.52). In a well-designed protocol, these should approximately follow a uniform distribution, but we have observed experimentally that this is far from being the case. Following this extremely simple approach, with neither approximations nor preconditions, we are able to recover up to 4 bits of the secret ID after around 2^{10} authentication sessions with a 100% success probability, a fact that could lead to a very straightforward tracking attack.

Finally, we can conclude that SASI protocol is definitely an interesting step in the right direction towards fully secure ultralightweight protocols, but it still falls short of the security requirements for such schemes.

6.3.3 Gossamer Protocol

As a consequence of observations in *Section 6.3.2.5*, we have derived a new protocol, called Gossamer[1], which is inspired by the SASI scheme but hopefully devoid of its weaknesses. Our main aim was to define a protocol with adequate security level and which can be realistically be employed in ultralightweight RFID tags.

6.3.3.1 Model Suppositions

Each tag stores a static identifier (ID), an index-pseudonym (IDS) and two keys (k_1/k_2) in its memory. This information is also stored in the back-end database. The IDS is employed as a search index to allocate, in the database, all the information linked with each tag. These elements have a length of 96 bits, compatible with all the encoding schemes (i.e. GTIN, GRAI) defined by EPCGlobal. Additionally, tags are given the added requirement of storing the old and potential new values of the tuple (IDS, k_1, k_2), to avoid desynchronization attacks. In spite of this, resiliency against attacks which involve tag manipulation are not considered as these devices are not at all tamper-resistant.

For the implementation of the proposed protocol, only simple operations are available on tags, in accordance with their restrictions: specifically, bitwise XOR ($\oplus$), addition mod 2^m ($+$), and $Rot(x, y)$. To avoid ambiguity, $Rot(x, y)$ is defined to perform a circular shift on the value of x, (y mod N) positions to the left for a given value of N (in our case 96).

Random number generation, required in the protocol to supply freshness, is a costly operation, so it is performed by the reader. Moreover, random numbers cannot be indiscriminately employed because their use increases both memory requirements and message counts (which could be costly in certain applications). To significantly increase security, we have also added a specially designed and very lightweight function called $MixBits$. In [94], a detailed description of the methodology used -basically, to evolve compositions of extremely light operands by means of genetic programming,

[1]Gossamer: Noun describing a thin film of cobwebs floating in the air (this meaning dates from the 14th century) and an adjective meaning light, delicate, thin enough to let light through, nearly transparent.

in order to obtain highly non-linear functions- is included. $MixBits$ has an extremely lightweight nature, as only bitwise right shift ($>>$) and addition operations are employed. Specifically,

```
Z = MixBits(X,Y)
----------------------------
Z = X;
for(i=0; i<32; i++) {
Z = (Z>>1) + Z + Z + Y ;}
----------------------------
```

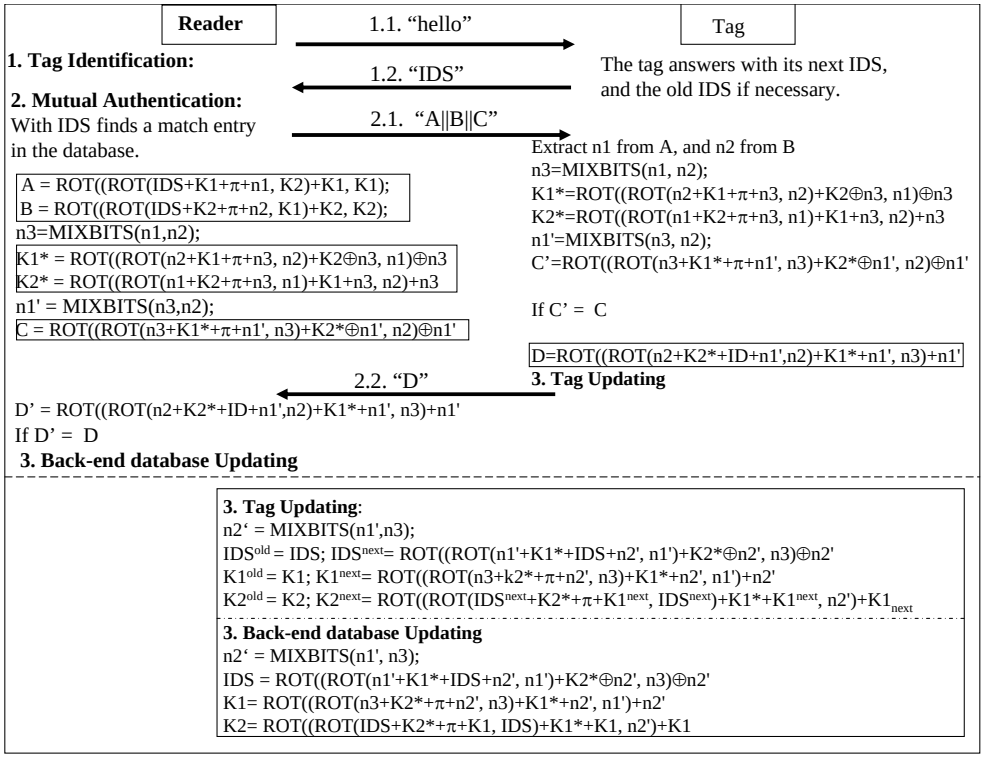
Communications have to be initiated by readers, since tags are passive. The communication channel between the reader and the database is generally assumed to be secure, but the channel between the reader and the tag can be eavesdropped on. Attacks involving modification of the exchanged messages, the insertion of fraudulent new messages, or message blocking (active attacks), can be discounted.

6.3.3.2 The Protocol

The protocol comprises three stages: tag identification phase, mutual authentication phase, and updating phase. *Figure 6.7* shows the exchanged messages.

Tag Identification The reader first sends a "hello" message to the tag, which answers with its potential next IDS. With it, the reader tries to find an identical entry in the database. If this search succeeds, the mutual authentication phase starts. Otherwise the identification is retried but with the old IDS, which is backscattered by the tag upon request.

Mutual Authentication With IDS, the reader acquires the private information linked to the tag, identified from the database. Then the reader generates nonces n_1 and n_2 and builds and sends to the tag $A||B||C$ (see *Figur 6.7*). Note that the equations used in the generation of public messages, as do those used in the computation of internal values, generally follow the scheme bellow:



† π = 0x3243F6A8885A308D313198A2 ($L = 96$ bits)

Figure 6.7: Gossamer Protocol

$$n_{i+2} = MIXBITS(n_i, n_{i+1}) \quad (6.59)$$

$$M_i = ROT((ROT(n_{i+1} + k_i + PI + n_{i+2}, n_{i+1}) + K_{i+1} \oplus n_{i+2}, n_i) \oplus n_{i+2} \quad (6.60)$$

$$M_{i+1} = ROT((ROT(n_i + K_{i+1} + PI + n_{i+2}, n_i) + K_i + n_{i+2}, n_{i+1}) + n_{i+2} \quad (6.61)$$

From submessages A and B, the tag extracts nonces n_1 and n_2. Then it computes n_3/n_1' and k_1^*/k_2^* and builds a local version of submessage C'. This is compared with the received value. If it is verified, the reader is authenticated. Finally, the tag sends message D to the reader. On receiving D, this value is compared with a computed local version. If comparison is successful, the tag is authenticated; otherwise the protocol is abandoned.

Index-Pseudonym and Key Updating After successfully completing the mutual authentication phase between reader and tag, they locally update IDS and keys (k_1/k_2) as indicated in *Figure 6.7*. As we have just seen, submessages C/D allow reader/tag authentication, respectively. Moreover, the use of submessages C/D results in confirmation of synchronization for the internal secret values (n_3/n_1' and k_1^*/k_2^*) used in the updating phase, preventing straightforward desynchronization attacks.

6.3.3.3 Security Analysis

We will now analyze the security of the proposed scheme against relevant attacks:

Data Confidentiality All public messages are composed of at least three secret values shared only by legitimate readers and genuine tags. Note that we consider private information (ID, k_1, k_2), random numbers (n_1, n_2), and internal values (n_3, n_1', n_2', k_1^*, k_2^*) as secret values. The static identifier and the secret keys cannot, therefore, be easily obtained by an eavesdropper.

Tag anonymity Each tag updates IDS and private keys (k_1, k_2) after successful authentication, and this update process involves random numbers (n_3, n_1', n_2'). When the tag is interrogated again, a fresh IDS is backscattered. Additionally, all public submessages ($A||B||C||$ and D) are anonymized by the use of random numbers (n_1, n_2, n_3, n_1'). Tag anonymity is thus guaranteed, and location privacy of the tag owner is not compromised.

Mutual Authentication and Data Integrity The protocol provides mutual authentication. Only a legitimate reader possessing keys (k_1, k_2), can build a valid message $A||B||C$. Similarly, only a genuine tag can derive nonces n_1, n_2 from $A||B||C$, and then compute message D.

Messages C and D, which involve the internal secret values (n_3, n_1', k_1^*, k_2^*) and nonces (n_1, n_2), allow data integrity to be checked. Note that these values are included in the updating equations (potential next index-pseudonym and keys).

Replay attacks An eavesdropper could store all the messages exchanged in a protocol run. To impersonate the tag, he could replay message D. However, this response would be invalid as different nonces are employed in each session -this will frustrate this naive attack. Additionally, the attacker could pretend that the reader has not accomplished the updating phase in the previous session. In this scenario, the tag is identified by the old index-pseudonym and the attacker may forward the eavesdropped values of $A||B||C$. Even if this is successful, no secret information is disclosed and the internal state is unchanged in the genuine tag, so all these attacks are unsuccessful.

Forward Security Forward security is the property that guarantees the security of past communications even when a tag is compromised at a later stage. Imagine that a tag is exposed one day, making public its secret information (ID, k_1, k_2). The attacker still cannot infer any information from previous sessions as two unknown nonces (n_1, n_2) and five internal secret values (n_3, n_1', n_2', k_1^*, k_2^*) are involved in the message creation (mutual authentication phase). Additionally, these internal values are employed in the updating phase. Consequently, past communications cannot be easily jeopardized.

Updating Confirmation The Gossamer protocol assumes that tags and readers share certain secret values. As these values are locally updated, synchronization is mandatory. Submessages C and D provide confirmation of the internal secret values (n_3, n_1', k_1^*, k_2^*) and nonces (n_1, n_2). These values are employed in the updating stage. So the correct update of values IDS and keys (k_1, k_2) is implicitly ensured by submessages C and D.

Unintentional transmission errors can happen in the received messages since a radio link is used. This is an extremely serious issue for message D, since it can result in a loss of synchronization. However, the tuple (IDS, k_1, k_2) is stored twice in the tag memory -once with the old values, the other with the potential next values. With this mechanism, even in the event that message D is incorrectly received, the tag and the reader can still authenticate with the old values. So the reader and the tag will be able to recover their synchronized state.

Table 6.4: Performance Comparison of Ultralightweight Authentication Protocols

	U-MAP family [163, 162, 161]	SASI [46]	Gossamer
Resistance to Desynchronization Attacks	No	No	Yes
Resistance to Disclosure Attacks	No	No	Yes
Privacy and Anonymity	Yes	Yes	Yes
Mutual Authentication and Forward Security	Yes	Yes	Yes
Total Messages for Mutual Authentication	4-5L	4L	4L
Memory Size on Tag	6L	7L	7L
Memory Size for each Tag on Database	6L	4L	4L
Operation Types on Tag	$\oplus$, $\vee$, $\wedge$, +	$\oplus$, $\vee$, $\wedge$, +, Rot[2]	$\oplus$, +, Rot[3], $MixBits$

[1] L designates the bit length of variables used
[2] $Rot(x, y) = x << wht(y)$, being $wht(y)$ the Hamming weight of vector y
[3] $Rot(x, y) = x << (y \bmod L)$ for a given value of L -in our case $L = 96$

6.3.3.4 Performance Analysis

Our proposed protocol is now examined from the point of view of computational cost, storage requirements and communication cost. Additionally, *Table 6.4* compares the most relevant ultralightweight protocol proposals from a performance perspective.

Computational cost The protocol we have proposed only requires simple bitwise XOR, addition 2^m, left rotation, and the $MixBits$ function on tags. These operations are very low-cost and can be efficiently implemented in hardware.

When comparing Gossamer with the protocol SASI, we can observe that the bitwise AND and OR operations are eliminated, and the light $MixBits$ operation is added for increased security. $MixBits$ is very efficient from a hardware perspective. The number of iterations of this function is optimized to guarantee a good diffusion effect. Specifically, it consumes $32 \times 4 \times (96/m)$ clock cycles, m being the word length used to implement the protocol (i.e. $m = 8$, 16, 32, 64, 96). As this may have a cost impact on temporal requirements, we have minimized the number of $MixBits$ calls.

Storage requirement Each tag stores its static identifier (ID) and two

records of the tuple (IDS, k_1, k_2) -with old and potential new values. A 96-bit length is assumed for all elements in accordance with EPC-Global. The ID is a static value, thus stored in ROM. The remaining values ($96 \times 6 = 576$ bits) are stored in a rewritable memory because they need to be updated.

In the protocol SASI, two temporal nonces are linked to each session. We include an additional value derived from the previous nonces ($n_{i+2} = MixBits(n_i, n_{i+1})$). As these nonces are updated three times in the internal steps of the protocol, our scheme is roughly equivalent to the use of five fresh random numbers. So, with the relatively light penalty of storing an extra nonce, the security level seems to be notably increased.

Communication cost The proposed protocol performs mutual authentication and integrity protection with only four messages, so in this sense it is similar to the SASI scheme. In the identification phase, a "hello" and IDS message are sent over the channel. Messages $A||B||C$ and D are transmitted in the authentication phase. So a total of 424 bits are sent over the channel - considering 5 bytes for the "hello" message.

6.3.4 Concluding Comments

From the UMAP family of protocolos, we can infer the following:

Interest The protocols arouse interest in the design of new ultra-lightweight protocols. Indeed, they have inspired the proposal of other protocols [46, 142]. Additionally, as can be seen below, the security of the UMAP family of protocols has been carefully examined by the research community.

Security Weaknesses The security of the UMAP family of protocols has been analyzed under different assumptions. First, security vulnerabilities were revealed under the hypothesis of an active attacker [140, 141, 48]. Secondly, Bárász et al. showed how a passive attacker can disclose part of the secret information stored in the tag's memory [28, 29].

As mentioned in *Section 6.3.3.1*, only attacks that do not alter or interfere with communications are considered a real threat in most scenarios. In other words, active attacks are discounted when designing a protocol to meet the requirements of ultralightweight RFID tags.

Operations Only bitwise AND, XOR, OR and sum mod 2^m are required for the implementation of the UMAP protocol family. At first sight, the choice seems well-conceived as these operations can be efficiently implemented in hardware. However, they are all T-functions, which have a very poor diffusion effect; the information does not propagate well from left to right. Also, as a consequence of the use of bitwise AND and OR operations in the generation of certain messages, the latter were highly biased. These two operands should therefore be avoided in messages passed on the channel, but may be used in inner parts of the protocol.

The protocol SASI was a step further towards a secure protocol compliant with real ultralightweight tag requirements. However, as shown in *Section 6.3.2.1*, a passive attacker can obtain the secret ID by observing several consecutive authentications sessions. Despite this, we consider that the protocol design shows some interesting new ideas (specifically, the inclusion of rotations). The analysis of SASI and the UMAP protocol family has led to the proposal of Gossamer, a new protocol inspired by SASI and examined here both from the security and performance perspective. Indeed, the resources needed for the implementation of Gossamer are very similar to those of SASI the scheme, but Gossamer seems to be considerably more secure because of the use of dual rotation and the $MixBits$ function. The price to be paid, of course, is the throughput (number of authenticated tags per second) of the Gossamer protocol. However, preliminary estimations seem to show that the commonly required figure of 100 responses per second is still achievable.

6.4 Active Attacks

In *Section 6.3* we have seen how the security level of low-cost RFID tags can be increased using non-cryptographic primitives. Since the publication of

Table 6.5: Specifications for Moderate-cost RFID Tags

	Moderate-cost RFID Tag
Power Source	Passively powered
Storage	32 - 1K bits
Circuitry (security processing)	Up to 6K gates Lightweight cryptographic primitives
Reading Distance (comercial devices)	Up to 3 m
Physical Attacks	Not resistant
Resistance to Passive Attacks	Yes
Resistance to Active Attacks	Yes

the predecessors of Gossamer protocol (UMAP family and SASI) the interest by non-cryptographic primitives (lightweight protocols) have increased [47, 125, 124, 134, 137].

On the other hand, as we see in *Chapter 3*, the security level of the different RFID tags classes should not necessarily be equal. In *Section 6.1*, we point out that low-cost RFID tags should be resistant to passive attacks but not to active attacks. However, high-cost RFID tags are designed to be resistant to passive and active attacks as tradicional cryptographic primitives are supported on-board.

In this section we are going to see a different class of RFID tags, in the following denominated as "moderate-cost" RFID tags. This class of tags should not be vulnerable both passive and active tags. However, moderate-cost tags do not support on-board standard cryptographic primitives differently from what it happens in high-cost RFID tags. Its security resides on lightweight cryptographic primitives. So, for this purpose, the designing of new lightweight cryptographic primitives is imperative. The specifications of these tags are similar to low-cost RFID tags with the particularity that the number of logic gates devoted to security tasks is superior. This limit is fixed to 4K gates for low-cost RFID tags. We consider that around 6K gates can be dedicated to security for moderate-cost RFID tags, that means an increase of around 50%. *Table 6.5* summarizes the characteristics of this class of tags.

The standard EPC-C1G2 ratified the use of PRNG for low-cost RFID tags, so its use for moderate-cost RFID tags is completely justified. Additionally, another primitive like a keyed hash function or a cipher should be employed in the protocol design. Indeed, the research area of designing

lightweight primitives is becoming importance. The ECRYPT Stream Cipher Project (profile-2 for embedded devices) is only an example of it [14].

Instead of beginning from scratch, we have tried to avoid past errors in the designing of our protocol. The kind of attacks applicable to RFID technologies are not much different to those that can happen in wireless, bluetooth, or smart-card systems. In deed, we have found interesting resemblances in the field of smart-card security, which is by now a consolidated technology. Since the pioneer work of Lamport (1981) where he proposed a remote authentication scheme, many researchers suggested alternative schemes improving the efficiency and security of remote authentication processes. Recently, Shieh et al. have proposed a very interesting scheme in their work entitled "Efficient remote mutual authentication and key agreement" [196]. This protocol is considered to be one of the most secure an efficient security protocols for smart-cards. Taking advantage of this work, we have updated their protocol to the special features of RFID systems. The resulting protocol is not only resistant to the standard passive attacks, such as privacy, tracking and eavesdropping, etc. but also to active attacks.

6.4.1 Review of Shieh et al.'s Scheme

The security of Shieh et al.'s scheme (2006) is based on the use of secure one-way hash functions (Merkle, 1989; NIST FIPS PUB 180, 1993; Rivest, 1992). Time stamps are used but no time-synchronization is required. The scheme consists of two phases: the registration phase, and the login and key agreement phase.

6.4.1.1 Registration Phase

Assume an user U_i submits his identity ID_i and password PW_i to the server over a secure channel for registration. If the request is accepted, the server computes $R_i = h(ID_i \oplus x) \oplus PW_i$ and issues U_i a smart-card containing R_i and $h()$, where $h()$ is a one-way hash-function, x is the secret key maintained by the server, and the symbol "$\oplus$" denotes the exclusive-OR operation.

(1) U_i	$\rightarrow$	Server:	ID_i, T_u, MAC_u
(2) Server	$\rightarrow$	U_i:	T_u, T_s, MAC_s
(3) U_i	$\rightarrow$	Server:	T_s, MAC''_u

$a_i = h(ID_i \oplus x)$ $\quad$ $MAC_u = h(T_u||a_i)$

$MAC_s = h(T_u||T_s||a'_i)$ $\quad$ $MAC''_u = h(T_s||(a_i + 1))$

Figure 6.8: Messages Transmitted in Shieh's Scheme

6.4.1.2 Login and Key Agreement Phase

Figure 6.8 is an illustration of messages transmitted during the login and key agreement phase in Shieh's scheme. When user Ui wants to login to the server, he first inserts his smart-card into a card reader then inputs his identity ID_i and password PW_i. Next, the smart-card performs the followings steps:

1. Compute $a_i = R_i \oplus PW_i$.
2. Acquire current time stamp T_u, store T_u until the end of the session, and compute $MAC_u = h(T_u||a_i)$.
3. Send message (ID_i, T_u, MAC_u) to the server.

After receiving message (ID_i, T_u, MAC_u) from U_i, the server performs the following steps to assure the integrity of the message, answer to U_i, and challenge U_i to avoid replay attacks:

1. Check the freshness of T_u. If T_u has already appeared in a current execution session of user U_i, reject U_i's login request and stop the session. Otherwise T_u is fresh.
2. Compute $a'_i = h(ID_i \oplus x)$, $MAC'_u = h(T_u||a'_i)$ and check whether MAC'_u is equal to the received MAC_u. If it is not, reject U_i's login and stop the session.
3. Acquire current time stamp T_s. Store temporarily paired time stamps (T_u, T_s) and ID_i for freshness checking until the end of the session. Compute $MAC_s = h(T_u||T_s||a'_i)$ and session key $K_s = h((T_u||T_s) \oplus a'_i)$. Then, send the message (T_u, T_s, MAC_s) back to U_i.

On receiving the message (T_u, T_s, MAC_s) from the server, the smart-card performs the following steps to authenticate the server, achieves a session key agreement, and answers to the server.

1. Check if the received T_u is equal to the stored T_u to assure the freshness of the received message. If is not, report login failure to the user and stop the session.
2. Compute $MAC'_s = h(T_u||T_s||a_i)$ and check whether it is equal to the received MAC_s. If not, report login failure to the user and stop. Otherwise conclude that the responding party is the real server.
3. Compute $MAC''_u = h(T_s||a_i{+}1)$ and session key $K_s = h((T_u||T_s)\oplus a_i)$, then send the message (T_s, MAC''_u) back to the server.

When the message (T_s, MAC''_u) from U_i is received, the server performs the following steps to authenticate U_i and achieve key agreement:

1. Check if the received T_s is equal to the stored T_s. If it fails reject U'_i login request and stop the session.
2. Compute $MAC'''_u = h(T_s||(a'_i + 1))$ and check whether this is equal to MAC''_u. If it is not, reject U_i's login request and stop the session. Otherwise, U_i is a legal user and U_i's login is permitted. At this moment, mutual authentication and session key agreement between U_i and the server are achieved.

6.4.2 Our scheme

In this section, a new protocol adapted to RFID systems and resistant to passive and active attacks (inspired in Shieh et al.'s protocol) is proposed. First, we will mention some peculiarities of RFID systems which should be considered in the new design. These will force changes in the protocol which will be presented next.

In Shieh et al.'s protocol, when the user wants to login in the server *"he first inserts the card into a card-reader..."*. In a RFID system, tags (T) will be equivalent to smart-cards and readers to card-readers respectively. Note RFID

readers (R) are assumed to be connected to back-end databases (B) over a secure channel. Additionally, both devices have "non-limited" computing and storing capabilities. In the following, when we refer to a RFID reader an entity composed by a reader and a back-end database is considered.

Additionally, there are significant differences between smart-card and RFID systems. RFID technology operates through the radio channel, so communication could be eavesdropped. Another particularity is the asymmetry of the communication channel, which allows monitorization of the forward channel (reader-to-tag) from a long-range distance than the backward channel (tag-to-reader). Smart-cards are usually tamper resistant devices, which is not the case of RFID tags. Furthermore, when then smart-card is inserted in the reader an user intervention is necessary, entering his identity and password. In RFID technology, however, interactions between tags and readers are automatic.

Taking into account all these considerations, Shieh et al's scheme has been adapted. Our proposed scheme consists of two phases: the registration phase, and the mutual authentication and index-pseudonym update phase. The following symbols have been used:

x_i:	secret key maintained by the reader	N_z:	random number generated by z
$h()$:	secure one-way hash function	$\oplus$:	exclusive-OR operation
$\|\|$:	string concatenation operation		

6.4.2.1 Registration Phase

The user or holder of the tag submits his static identifier ID_i[2] and a freely chosen password PW_i to the reader over a secure channel for registration. If the request is accepted, the reader generates a random index-pseudonym IDS_i^0 and computes $a_i = h(ID_i \oplus x_i)$[2]. The tag will replace its identifier ID_i by IDS_i^0 and store a_i. The IDS_i^n will be used as searching-index of a database in which all the sensitive information (ID_i, x_i, PW_i) and the temporary data session (N_{T_i}, N_R) associated with each tag are stored. IDS_i^{new}

[2] A 64-bit length identifier is compatible with all the encoding schemes (SGTIN, SSCC, GLN, etc) defined by EPCGlobal [61]. Due to this reason, we assume that tag static identifier (ID_i), and index-pseudonyms (IDS_i^n) are 64-bit length. Additionally, the secret key x_i is xored with ID_i to compute a_i, so x_i length is also set to 64-bits.

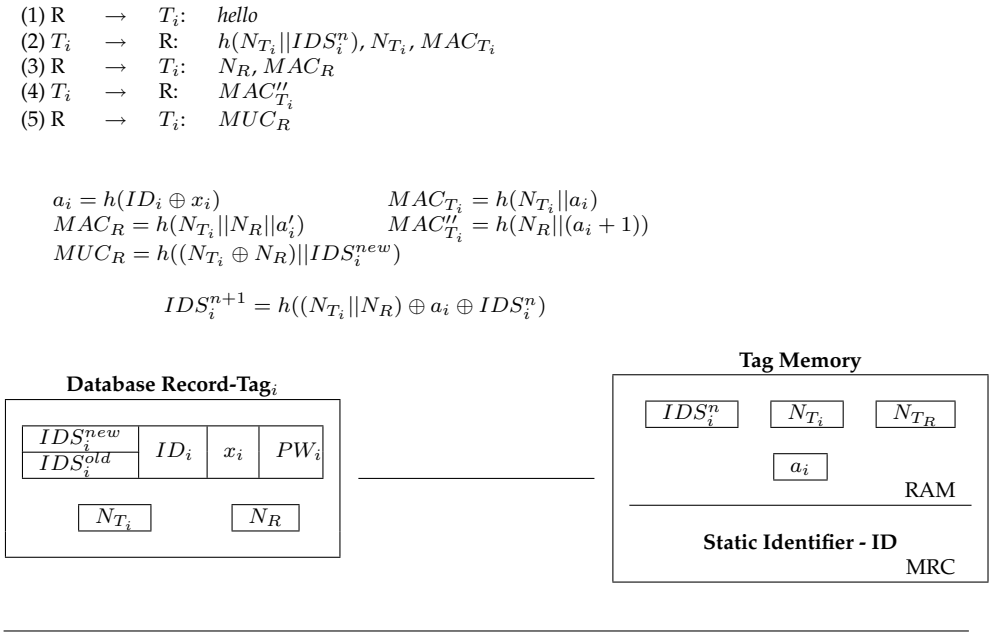


Figure 6.9: Messages Transmitted in our Protocol

and IDS_i^{old} are initially set to IDS_i^0. The password PW_i will be used by the holder of the tag (over a secure channel) to temporarily deactivate the tag. In this case, a_i will be replaced by $R_i = a_i \oplus PW_i$.

6.4.2.2 Mutual Authentication and Index-Pseudonym Update

The messages exchanged in our scheme are shown in *Figure 6.9*. First, the reader usually applies a probabilistic (i.e. Aloha-based algorithm) or determinist (i.e. Binary tree-walking protocol) collision avoidance protocol to singulate a tag out of many [193]. Upon singulation condition, the reader will send a "$hello$" message to the tag. To start the mutual authentication, the tag accomplishes the following steps:

1. Generate a random number N_{T_i}[3], and store N_{T_i} temporarily until the end of the session.

[3]Tags conforming with EPC Class-1 Gen-2 specification support a 16-bit PRNG [67]. We suggest that 32-bit PRNGs should be supported, as mentioned in [100, 158]. So, 32-bit length could be an adequate value to N_{T_i} and N_R.

2. Compute $h(N_{T_i}||IDS_i^n)$, and $MAC_{T_i} = h(N_{T_i}||a_i)$.

3. Send message $(h(N_{T_i}||IDS_i^n), N_{T_i}, MAC_{T_i})$ to the reader and wait for response.

Once the previous message is received, its integrity is checked and the reader answer includes a challenge to avoid replay attacks:

1. Check the newness of N_{T_i}. If N_{T_i} has already come out in a current mutual authentication, the protocol is stopped at this point. Otherwise N_{T_i} is fresh.

2. Compute $p' = h(N_{T_i}||IDS_i^{new})$ and $p'' = h(N_{T_i}||IDS_i^{old})$ and check wether any of the two values is equal to the received $h(N_{T_i}||IDS_i^n)$. The above procedure is repeated for each entry (row) in the database until a match is found. If not found, the protocol is stopped at this point.

3. Compute $a_i' = h(ID_i \oplus x_i)$, $MAC'_{T_i} = h(N_{T_i}||a_i')$, and check if it is equal to MAC_{T_i}. If not, the protocol is stopped and a check over tag deactivation is taken by computing $R_i' = a_i' \oplus PW_i$, $MAC'_{T_i} = h(N_{T_i}||R_i')$ and verifying if it is equal to MAC_{T_i}. A match will imply that the tag has been deactivated temporally by its holder.

4. Acquire a fresh random number N_R[2]. For avoiding replay attacks, the pair (N_{T_i}, N_R) is stored until the end of the session.

5. Compute $MAC_R = h(N_{T_i}||N_R||a_i')$. Then, send the message (N_R, MAC_R) back to the tag and wait for response.

After receiving the message (N_R, MAC_R), the following steps are accomplished to authenticate the reader, achieve new material to update the index-pseudonym, and finally answer to the reader:

1. Compute $MAC'_R = h(N_{T_i}||N_R||a_i)$ and check if its value is equal to the received MAC_R. If not, stop the protocol at this point. Note that the newness of this message is guaranteed by N_{T_i}. For preventing loss of synchronization attacks, N_R is also stored in the tag.

2. Compute $MAC''_{T_i} = h(N_R||(a_i + 1))$ and send it back to the reader.

When the message MAC''_{T_i} is received, the reader computes $MAC'''_{T_i} = h(N_R||(a'_i+1))$ and checks whether it is equal to MAC''_{T_i}. If not, the protocol is stooped. At this point, both the reader and the tag have mutually authenticated. Additionally, both possess two nonces (N_{T_i}, N_R), which have been exchanged. Shieh et al. proposed using this fresh material to establish a session key agreement. In our case this material is employed to update the index-pseudonym. Obviously, the tag and reader have to be synchronized.

The glib solution for the synchronization problem will be to update the index-pseudonym in the tag when message 4 is sent, and this updating will be performed in the reader when checking this message. Under this scenario an attacker (active attack) could intercept message 4 avoiding the update of the index-pseudonym in the reader with the consequently losing of synchronization. A naïve solution will consist of assuming that after the end of the protocol, completion messages are sent between the involved entities. However, these messages could be also intercepted. Additionally, note that tags are much more constrained devices than readers. For this reason, a new message 5 has been added to the protocol (Message Update Code - MUC), and readers will have to store the old and the new index-pseudonym to prevent the interception of this message. To complete the protocol, the following steps are performed by the reader:

1. Store the current session index-pseudonym $IDS_i^{old} = IDS_i^{new}$ to avoid desynchronization attacks.

2. Compute the new index-pseudonym $IDS_i^{new'} = h((N_{T_i}||N_R) \oplus a'_i \oplus IDS_i^{new})$.[4]

3. Compute $MUC_R = h((N_{T_i} \oplus N_R)||IDS_i^{new'})$ and send it to the tag, including the two nonces exchanged between reader and tag and the new index-pseudonym.

When the message MUC_R is received from reader, the tag accomplishes the following steps to verify a successfully index-pseudonym update has

[4]If tags support on board the proposed *Tav*-128 hash function, a_i's length will be fixed to 128-bits ($a_i = h(ID_i \oplus x_i)$). In this case, we suggest the following update equation: $IDS_i^{new'} = h((N_{T_i}||N_R) \oplus a'_i[0{:}63] \oplus a'_i[64{:}127] \oplus IDS_i^{new})$.

been performed in the reader:

1. Compute the potential-new index-pseudonym $IDS_i^{n+1} = h((N_{T_i}||N_R) \oplus a_i \oplus IDS_i^n)^4$.
2. Compute $MUC_R'' = h((N_{T_i} \oplus N_R)||IDS^{n+1})$ and check whether MUC_R'' is equal to MUC_R. If this is the case, update the index-pseudonym.

6.4.2.3 Security Analysis

The robustness of the proposed protocol against the main important attacks is analyzed in the following.

1. User Privacy
   Tag ID_i must be kept secure to guarantee user's privacy. In order to protect it, both the tag's memory and the radio channel have been taken into account. In the registration phase, the static identifier ID_i and the password PW_i are submitted to the reader over a secure channel. To avoid physical access to the static identifier, ID_i is replaced by the hash of $ID_i \oplus x_i$. Note, x_i is a secret key only known by the reader. Additionally, and similarly to what happens in e-passports, we recommended the ID_i to be printed as a machine-readable code as illustrated in *Figure 6.9*. In the radio channel, the value of IDS_i^n is protected by the use of a secure one-way hash function $h()$. In the same way, a_i can not be derived from the messages authentication codes MAC_{T_i}, MAC_R and MAC_{T_i}''.
2. Location Privacy
   The secure protection of tag information does not ensure location privacy. Constant answers would allow an attacker to identify each tag with its holder. To protect the index-pseudonym only its hash is transmitted. As the index-pseudonym is not updated until the completion of the protocol and the protocol may be accidentally or intentionally interrupted, the hash of the IDS_i concatenated with nonce N_{T_i} is really sent. Similarly, a_i is anonymized by means of

the use of message authentication codes where a kind of challenge-response nonces are included. Finally, sending the message update code $MUC_R = h((N_{T_i} \oplus N_R)||IDS^{n+1})$, the new index-pseudonym is hidden. So, in order to avoid tracking, all the information is anonymized.

3. Data Integrity
   Based on the use of a mutual authentication approach, our protocol guarantees data integrity between tag and reader. On the other hand, tag's memory is rewritable so modifications are possible. In this memory, both a_i and the index-pseudonym IDS_i^n are stored. If an attacker does succeed in modifying this part of the memory, the reader would not recognize the tag, having to carry out the registration phased again (see *Section 6.4.2.1*).

4. Mutual Authentication
   Due to the fact that both tag and reader authenticate each other, by means of message authentication codes MAC_R and MAC''_{T_i}, mutual authentication is accomplished. These message authentication codes include a_i, a secret only shared between them, preventing any other to create correct $MACs$, and in this way guaranteeing the legitimacy of each part. Therefore it is infeasible for a fraudulent reader or tag to impersonate another entity.

5. Replay Attack
   Our protocol is based on a challenge-response scheme, so replay attacks are prevented because challenges are different each time and long enough to prevent attacks based on storing them. In our scheme, any replay attack will not be able to correctly answer the challenges that form part of the protocol. In message 2, the tag sends ($h(N_{T_i}||IDS_i^n)$, N_{T_i}, MAC_{T_i}) where a nonce N_{T_i} is included. Therefore, the reader must include N_{T_i} in the answer message, so in message 3 the reader sends (N_R, $MAC_R = h(N_{T_i}||N_R||a'_i)$), including not only the response nonce N_{T_i} but also the challenge nonce N_R. Then, the tag sends $MAC''_{T_i} = h(N_R||(a_i + 1))$ back, including N_R, to the reader. So, only legitimate parties (reader+tag) can send valid answers as challenge nonces are joined with the message authentication codes requiring the knowledge of a_i.

6. Forgery Resistance
   All the sensitive information stored in the tag (IDS_i^n, a_i) is never sent in clear over the communication channel. In all cases, this information is concatenated with a nonce and hashed before passed on the channel. So the simple copy of information by eavesdropping is not useful to an adversary.

7. Active Attacks

   (a) Man-in-the-middle Attack: If an attacker tries to impersonate a legitimate reader to obtain information from a tag, perhaps to be able to impersonate it in a future. This kind of attack is not feasible because all messages include a message authentication code, which requires the knowledge of the secret a_i shared only between the tag and the reader. In the previous scenario, the fraudulent reader will not be able to generate message 3, so the capture of the message 4 sent back by the tag will be a vain attempt. Moreover, in future sessions, a new challenge would be used by the reader preventing any advantage from knowing old messages.

   (b) Parallel Session: Because of the asymmetric structure of the message authentication codes $MAC_{T_i} = h(N_{T_i}||a_i)$ and $MAC''_{T_i} = h(N_R||a_i + 1)$ this attack fails. Another important point is that both reader and tag store the session nonces, N_{T_i} and N_R.

   (c) Synchronization Loss: The tag updates the index-pseudonym only when the message update code (MUC) is received. An attacker could interrupt this message, trying to desynchronize reader and tag. To avoid this sort of attack, each time the reader updates the index-pseudonym, the old index-pseudonym is still maintained. Under the interception of the MUC from the reader, the tag will use the old index-pseudonym to build $h(IDS_i^n||N_{T_i})$. When the reader checks its integrity, it first will try with the new index-pseudonym, and if it fails, then he will try with the old index-pseudonym. Next, the rest of the protocol will be accomplished ensuring the recovery of synchronization loss.

6.4.3 Lightweight Hash-Function

Informally, a cryptographic hash function is a transformation that takes a variable-length input and returns a fixed-size string, which is called the hash value. Specifically, it is commonly assumed that a cryptographic hash function should meet the following prerequisites:

- Preimage resistant: given h, it should be hard to find any m such that $h = hash(m)$.
- Second preimage resistant: given an input m_1, it should be hard to find another input, m_2 (not equal to m_1) such that $hash(m_1) = hash(m_2)$.
- Collision-resistant: it should be hard to find two different messages m_1 and m_2 such that $hash(m_1) = hash(m_2)$.

For a hash output of n bits, compromising these should require 2^n, 2^n, and $2^{n/2}$, respectively. Additionally, some precautions should be taken when a new protocol is designed. Since most hash functions are built using the Merkle-Damgard construction, these are vulnerable to length-extension attacks: given $h(m)$ and $length(m)$ but not m, by choosing a suitable m' an attacker can calculate $h(m||m')$, where $||$ denotes concatenation.

Because a key is not necessary, hash functions are considered a better choice from the implementation point of view in the RFID security community. As a result, most of the proposed protocols are based on the use of hash functions. However, traditional cryptographic primitives excess the capabilities of low-cost RFID tags, and even that of moderate-cost RFID tags. As the protocol we proposed in *Section 6.4.2* is based on the use of a hash function, a new lightweight hash function, named *Tav*-128, is proposed. Next, the code of this functions is included:

```
/*****************************************************************************/
Process the input a1 modifying the accumulated hash a0 and the state
/*****************************************************************************/
void tav(unsigned long *state, unsigned long *a0, unsigned long *a1)

{ unsigned long h0,h1; int i,j,r1,r2,nstate;
/* Initialization */
 r1=32; r2=8; nstate=4;
 h0=*a0; h1=*a0;

/* A - Function */ for(i=0;i<r1;i++){h0=(h0<<1)+((h0+(*a1))>>1);}
/* B - Function */ for(i=0;i<r1;i++){h1=(h1>>1)+(h1<<1)+h1+(*a1);}

/* C and D - Function */ for(j=0;j<nstate;j++) {
 for(i=0;i<r2;i++)
    {
    /* C - Function */
    h0^=(h1+h0)>>3;
    h0=((((h0>>2)+h0)>>2)+(h0<<3)
           +(h0<<1))^0x736B83DC;
    /* D - Function */
    h1^=(h1^h0)>>1;
    h1=(h1>>4)+(h1>>3)+(h1<<3)+h1;
    } // round-r2
     state[j]+=h0;
     state[j]^=h1;
} // state

/* a0 updating */
 *a0=h1+h0;
}

/*****************************************************************************/
Initialization of the state and a0 with random values obtained from
www.random.org
/*****************************************************************************/
void init_state(unsigned long *state, unsigned long *a0)

 {
     state[0]=0xa92be51d;
     state[1]=0xba9b1ef0;
     state[2]=0xc234d75a;
     state[3]=0x845c2e03;
        a0[0]=0x768c7e74;
}
```

6.4.3.1 *Tav*-128 Design and Security Analysis

Some of the recent cryptanalytic attacks on many of the most important hash functions [218, 219] rely on the fact that these constructions generally use a very linear (LFSR-based) message expansion algorithm. In order to avoid this, we have decided to make the expansion of the *Tav*-128 hash function (corresponding to algorithms C and D) highly nonlinear. As, on the other hand, the resulting function should be very efficient and lightweight both from the gate count and the throughput point of view, we have found these functions by evolving compositions of extremely lightweight operands by means of genetic programming, as described in [94].

We have also tried to include a filter phase (corresponding to algorithms A and B in the code) in the input of the *Tav*-128 function, to avoid giving the attacker direct access to any bit of the internal state. Without this possibility, some attacks that have been found on other cryptographic primitives in the past are precluded: decreasing the control that the attacker has over the hash functions inputs complicates his task significantly.

An output length of 128 bits was found to be a reasonable compromise between speed and robustness to realistic attacks in the intended scenarios. Additionally, we propose the use of eight rounds in the internal loop (r_2 parameter) for having an adequate security margin, although we have found that even with six rounds (which will significantly improve its performance) the overall scheme seems to be secure.

We have performed an additional security analysis of *Tav*-128, consisting of examining the statistical properties of its output over a very low entropy input. Specifically, 2^{25} 32-bit inputs have been generated by means of an incremental counter (x, $x+1$, $x+2$, etc.). After randomly initializing (with values obtained from `http://randomnumber.org`) the internal state and the accumulated hash $a0$ value, we compute the output of *Tav*-128 for each counter value input ($Tav(x)$, $Tav(x+1)$, $Tav(x+2)$, etc.). The resulting hashes have been analyzed with two well-known suites of randomness tests, namely ENT [216] and DIEHARD [149]. The results are presented in *Tables 6.6* and *6.7*. *Tav*-128 also passed the very demanding -because it is oriented to cryptographic applications- NIST [205] statistical

Table 6.6: Results Obtained with ENT (Tav-128)

Test	Tav-128
Entropy	7.999999 bits/byte
Compression Rate	0%
χ^2 Statistic	269.73 (50%)
Arithmetic Mean	127.4993
Monte Carlo π Estimation	3.14178848 (0.01%)
Serial correlation Coefficient	-0.000073

battery. We have computed 100 p-values for each test, being all the results compatible with a uniform $U(0,1)$. The whole report is available in `http://163.117.149.137/tav/` due to the huge amount of p-values generated.

The author of this thesis acknowledges that successfully passing these statistical batteries, even over a very low-entropy input, does not prove security, but it does point out the nonexistence of trivial weaknesses.

6.4.3.2 Hardware Complexity

One of the most relevant aspects considered in the design of *Tav*-128 was the sort of operations that can be employed. As tags are very restricted computationally, only simple operations have been used. For example, multiplication has been ruled out due to its high cost [144]. Specifically, the following operators have been finally used: right shifts, bitwise XOR, and addition mod 2^{32}. The necessary architecture to implement *Tav*-128 can be divided into two main blocks:

- **Memory Blocks.** All the used variables are stored in this part: state (128-bits), accumulated hash a0 (32-bits), internal variables $h0$ (32-bits) and $h1$ (32-bits), and the input $a1$ (32-bits).
- **Arithmetic Logic Unit.** In this unit the addition mod 2^{32} and the bitwise XOR operation are implemented. As the $h0$ and $h1$ functions consist of three or more components, an auxiliary register to store the intermediate results is necessary.

Although we have not implemented *Tav*-128 in hardware, an overestimation of its gate count can be easily obtained. The function bitwise XOR

Table 6.7: Results Obtained with the Diehard Suite (Tav-128)

	Tav-128
Test	**p-value**
Birthday Spacings	0.725
	0.868
GCD	0.229
	0.138
Gorilla	0.779
Overlapping Permutations	0.823
	0.849
	0.349
	0.897
	0.939
Ranks of 31×31 and 32×32 Matrices	0.556
	0.241
Ranks of 6×8 Matrices	0.315
Monkey Tests on 20-bit Words	0.312
Monkey Test OPSO, OQSO, DNA	OK
Count the 1's in a Stream of Bytes	0.473
Count the 1's in Specific Bytes	OK
Parking Lot Test	0.235
Minimum Distance Test	0.580
Random Spheres Test	0.912
The Squeeze Test	0.487
Overlapping Sums Test	0.106
Runs Up and Down Test	0.147
The Craps Test	0.3211
	0.067
	0.775
	0.261
Overall KS p-value	**0.826**

requires 32 logic gates as we are operating with 32-bit variables. For implementing the add with carry circuit, a parallel architecture is proposed. Six logic gates are needed for each bit added in parallel[5]. The registers will be implemented by means of flip-flops. A gate count of 8 has been chosen for implementing a flip-flop as in [90]. So, 2304 logic gates are necessary to store the memory block and the auxiliary register. Additionally, around 50 extra logic gates are employed to control the internal state of the hash function. Therefore, 2578 logic gates are needed for implementing *Tav*-128.

Another key aspect to consider is throughput. We reckon that 1568 clock cycles are consumed in executing one *Tav*-128 hash. Due to the fact that low-cost RFID tags imply serious powers restrictions, we assume that the clock frequency is set to 100 KHz. Under this conditions, the throughput obtained by a tag that would have on-chip *Tav*-128 will be around 65 hashes/sec. It is generally accepted that at least between 50-100 tags should be authenticated per second [187]. In order words, a tag may use up at the most 2000 clock cycles ($@100Khz$) to answer a reader. In some applications 65 hashes/sec may not be enough, so we have analyzed how to increment the speed of *Tav*-128. In the initial proposed scheme, we have a parameter (r_2), which fits the number of rounds computed in the C and D algorithms. This parameter has been initially fixed to eight rounds in order to guarantee a high avalanche effect. After accomplishing a deeper study, we have determined that r_2 may be reduced to six rounds. So the speed of the tag will be incremented by 25% or in other words, the tag may compute around 80 hashes/sec. Note that for non-high speed demanding applications, we recommend to fix r_2 to eight rounds.

Finally, an overestimation of the total number of gates needed to implement the proposed protocol can be computed. The protocol is based on a hash functions and a PRNG. Tav-128 can be implemented with around 2.6K gates. As we will see in the next section, a lightweight PRNG, conforming with EPC-C1G2 [61] specification, demands around 1.6K gates to be implemented. Therefore, in total 4.2K gates are needed to security tasks, which is inferior than limit fixed initially. In conclusion, the proposed protocol is suitable for moderate-cost RFID tags.

[5] $S = A \oplus [B \oplus C_{ENT}]$ $C_{SAL} = BC_{ENT} + AC_{ENT} + AB$

6.5 Pseudo-Random Number Generation

6.5.1 Introduction

The need for random and pseudo-random numbers arises in many cryptographic applications. In fact, the usage of Pseudo-Random Number Generators (PRNGs) in RFID systems has been proposed almost from the start. In 2003, Weis et al. proposed the randomized hash-locking scheme, based on a hash function and a random number generator in order to prevent tracking, but limiting its applicability only to small tag populations [224]. Molnar et al. proposed a simple protocol for enhancing passwords in RFID tags [155]. There are others papers where the use of a PRNG has been proposed [45, 59, 135, 183, 214]. Nowadays, the used of a PRNG has been ratified by EPCGlobal (EPC-C1G2) and ISO (ISO/IEC 18000-6C). As we saw in the *Chapter 4*, a generator conforming with these specifications [61, 104], should meet the following randomness criteria:

- **Probability of a single RN16:** The probability that any RN16 drawn from the RNG has value RN16 = j for any j, shall be bounded by:

$$\frac{0.8}{2^{16}} < \mathrm{P}(\mathrm{RN16} = j) < \frac{1.25}{2^{16}} \tag{6.62}$$

- **Probability of simultaneously identical sequences:** For a tag population of up to 10,000 tags, the probability that any of two or more tags simultaneously generate the same sequence of RN16s shall be less than 0.1%, regardless of when the tags are energized.

- **Probability of predicting an RN16:** An RN16 drawn from a tag's RNG 10ms after the end of Tr, shall not be predictable with a probability greater than 0.025% if the outcomes of prior draws from RNG, performed under identical conditions, are known.

So far, no public algorithm conforming to EPC-C1G2 specification has been published. On the other hand, one can find commercial tags that obey this specification [101, 208]. However, the algorithms of the PRNGs supported are not public, and if you try to obtain them a negative answer is received. Manufacturers should learn of past disasters, such as Texas DST

and Philips Mifare cards, whose security resided on its obscurity and was quickly broken [37, 117, 159]. Motivated by this necessity, a new PRNG conforming to the standard has been proposed [166].

6.5.2 Experimentation Issues

The methodology to obtain the core of our PRNG is based on the use of Genetic Programming (GP). GP is a stochastic population-based search method devised in 1992 by John R. Koza [127]. The technique evolves computer programs instead of just particular solutions to a specific problem as in GA. As PRNGs are designed and implemented as computers programs, the use of GP in the problem is justified.

We have used the `lil-gp` library [9] for our experimentation. Next, we briefly describe how its parameters have been adjusted to our particular problem.

Function Set These functions are the building blocks of the individual we will obtain. We decided to include only very efficient operations easy to implement in hardware: **vrotd** (one-bit right rotation), **xor** (addition mod 2), **and** (bitwise AND), **or** (bitwise OR), and **not** (bitwise NOT). The **sum** (sum mod 2^{32}) operator is also necessary, in order to avoid linearity. Multiplication was excluded due to its high cost [144].

Terminal Set The terminals will be represented by two 32-bit unsigned integers (a_0, a_1). We also included Ephemeral Random Constants (ERCs), which are constant values (in our problem, 32-bit random values) that GP uses to try to generate better individuals.

Fitness Function We use the Avalanche Effect to evaluate the nonlinearity of our generator. In fact, an even more demanding property will be used: the Strict Avalanche Criterion [76], which can be mathematically described by:

$$\forall x, y | H(x, y) = 1, \quad H(F(x), F(y)) \approx B\left(n, \frac{1}{2}\right) \tag{6.63}$$

To measure the proximity of the distribution of the computed Hamming distances to the sought theoretical binomial $B(n, 1/2)$ a χ^2

goodness-of-fit test statistic is employed. Specifically, the proposal fitness function was the following:

$$Fitness = 10^6/\chi^2 \tag{6.64}$$

It was necessary to amplify the fitness function (multiplying by 10^6) because the initial values of the χ^2 statistic were extremely high, making the fitness negligible at the beginning of the evolution process. In more detail, the fitness of each individual was calculated as follows: we used the Mersenne Twister generator [151] to randomly generate the pair (a_0, a_1). The output O_0 for this input is stored. Then we randomly flipped one single bit of this two 32-bit input and we obtained a new output O_1. Now we stored the Hamming distance between those two output values $H(O_0, O_1)$. This process is repeated a number of times ($2^{11} = 2048$ was experimentally proved to be enough) and each time a Hamming chi-square statistic is obtained.

Tree Size Limitations The depth and/or the number of nodes of the individuals should be limited. We tried both limiting the depth and not limiting the number of nodes, and vice versa. The best results were consistently obtained by using the latter option. We allowed the PRNG to use up to 65 nodes for trying to ensure a high degree of Avalanche Effect and robustness without exceeding the processing and temporal requirements of a low-cost RFID tag.

When the parameters were adjusted, we ran 20 experiments with different seeds for generating the initial population ($seed_i = (\pi * 100000)^i \pmod{1000000}$), with a population size of 500 individuals, a crossover probability of 0.8, a reproduction probability of 0.2, and an ending condition of reaching 2000 generations. These parameters were experimentally found to be adequate for our purposes. The best individual found following the approach described above has an Avalanche Effect of 15.9707 (16.00 being the optimal value) and presents a χ^2 goodness-of-fit test statistic of 5.1175 for a χ^2 probability distribution with 32 degrees of freedom implying that, with probability 0,9999999853, the computed Hamming distances come from a Binomial distribution B(32,1/2).

```
=== BEST-OF-RUN ===
generation: 1172
nodes: 65
depth: 39
hits: 1597070
TOP INDIVIDUAL: -- #1 --
hits: 1597070
raw fitness: 195409.7232
standardized fitness: 195409.7232
adjusted fitness: 195409.7232

=== TREE: ===
(xor (sum a1 a0) (vrotd (vrotd (sum (xor a0 a1)(vrotd (xor a1 (vrotd (vrotd
(vrotd (vrotd (sum (sum a0 a1) (vrotd (vrotd (sum (vrotd (vrotd (vrotd
(xor (sum a1 a0) (vrotd (vrotd (sum (sum a0 a1) (vrotd (vrotd (sum (vrotd
(vrotd (vrotd (vrotd (xor (sum a1 a0) (vrotd (vrotd (vrotd (sum (xor a0 a1)
(vrotd (vrotd (vrotd (vrotd (vrotd (sum a1 a0)))))))))))))))) a1)))))))))
a1)))))))))))))
```

6.5.3 Design Specification

The seed of the PRNG will consist of an initialization vector (iv) and a key (s). The iv may be public, but it is very important that it is never reused together with the same key. It can also be kept secret, effectively extending the keylength up to 64-bits, depending on the security needs of the specific application. The key is a secret only known by an authorized reader and the tag. Usually, the secret (s) will be set at the time of manufacture, and will be stored in the associate row of the back-end database. In *Equation 6.65*, and *6.66*, we show the proposed update function for the internal state of LAMED.

$$a_0^{n+1} = \begin{cases} a_1^n + iv & \text{if } n \text{ is odd} \\ a_1^n \oplus iv & \text{if } n \text{ is even} \end{cases} \tag{6.65}$$

$$a_1^{n+1} = \begin{cases} 0(a_0^n, a_1^n) \oplus s & \text{if } n \text{ is odd} \\ 0(a_0^n, a_1^n) + s & \text{if } n \text{ is even} \end{cases} \tag{6.66}$$

The output length is 32 bits. As the specification EPC-C1G2 proposes the use of a 16-bit PRNG, we have designed a 16-bit version of our PRNG, named LAMED-EPC, with an additional XOR operation before its final output. The 32-bit output is divided in two halves, $\text{MSB}_{31:16}$ and $\text{LSB}_{15:0}$. These two halves will then be xored in order to obtain a 16-bit output with higher entropy. In this way, our proposal is EPC-C1G2 compliant and has

the additional advantage that a 32-bit PRNG is also supported, which could be relevant for certain applications and also increases its flexibility and, probably, its longevity, as mentioned in [100, 158]. Furthermore, the access and kill PIN are 32-bit values. The use of 32-bit random numbers would avoid the complex multi-step procedure for using the access and kill command proposed in the standard. It is important to recall, however, that the security margin of a protocol using a 16-bit PRNG is usually bounded by $\frac{1}{2^{16}}$. Moreover, a generic time-memory-data trade-off attack costs $O(2^{\frac{n}{2}})$, see [34], where n is the number of inner state variables in the PRNG. In LAMED-EPC, with a public iv, which is the weakest security configuration possible, the total of state variables is 32. Thus, the expected complexity of a time-memory is lower limited by $O(2^{16})$.

6.5.4 Standard Security Analysis

We have performed an extensive security analysis of LAMED[6], consisting of examining the statistical properties of the output over a random initialization of the initialization vector (iv) and the key (s) obtained from `http://randomnumber.org`. Unfortunately, it is not possible to prove randomness, because there is no efficient deterministic definition of this rather abstract concept. Instead, scientists usually limit themselves to using batteries of randomness tests to verify that the output of a given function "seems" random, meaning the used tests cannot distinguish it from a truly (theoretical) random variable. In 2001, the National Institute of Standards and Technology (NIST) proposed a comprehensive suite of randomness tests suitable for the evaluation of PRNGs used in cryptographic applications [190]. Additionally, there is another very stringent set of randomness tests called Diehard, developed by Marsaglia [148, 149]. We have also used a battery of tests named ENT [216], and a very recent set of randomness tests proposed by Sexton [6]. However, none of these test suites ensure, when successfully passed, that a given generator is useful for all kind of applications. On the other hand, systematically passing the NIST and Diehard batteries provides evidence in favour of a good degree of output randomness.

[6]The whole report is available in `http://163.117.149.208/rfid/lamed/`

Table 6.8: Results Obtained with ENT (LAMED)

Test	LAMED	LAMED-EPC
Entropy	7.999999 bits/byte	7.999999 bits/byte
Compression Rate	0%	0%
χ^2 Statistic	256.90 (50%)	246.61 (50%)
Arithmetic Mean	127.5024	127.4980
Monte Carlo π Estimation	3.141474228 (0.00%)	3.141796646 (0.01%)
Serial correlation Coefficient	-0.000023	0.000015

Two files of 300MB and 4GB have been generated with LAMED and LAMED-EPC, the latter only being used in Sexton's battery as it needs a huge amount of data to run. Results obtained with ENT, Diehard and David Sexton's battery are presented in *Tables 6.8*, *6.9*, and *6.10* respectively. When several p-values were produced in the same test, we summarized them by a Kolmogorov-Smirnov p-value (marked with *), that should be greater than 0.05 to be considered successful. LAMED also passed the very demanding (being angled to cryptographic applications) NIST statistical battery. We have computed 100 p-values for every test in the statistical suite; the proportion of successful ones is presented in *Table 6.11*. If this proportion is lower than 0.96, the whole test is considered to have failed. From these results, we can conclude that LAMED's output successfully passed all the randomness tests.

6.5.5 Compliance to EPC-C1G2 Security Requirements

In this section we will study the compliance of LAMED-EPC with EPC-C1G2. This study will be started analyzing the probability of a single 16-bit random number. The standard asks that "the probability that any RN16 drawn from the RNG has value RN16 = j for any j, shall be bounded by $0.8/2^{16} < \text{P(RN16 = } j) < 1.25/2^{16}$". In order to verify this property, five files of 2^{30} bytes have been generated and analyzed. These files have been obtained with different secret keys and initialization vectors, in every case taken from `http://random.org/`. From this analysis, we can conclude that the probability of any 16-bit random number drawn from LAMED-

Table 6.9: Results Obtained with the Diehard Suite (LAMED)

	LAMED	**LAMED-EPC**
Test	**p-value**	**p-value**
Birthday Spacings	0.261	0.192
GCD and Gorilla	0.778*	0.608*
Overlapping Permutations	0.311*	0.564*
Ranks of 31×31 and 32×32 Matrices	0.699*	0.587*
Ranks of 6×8 Matrices	0.521	0.947
Monkey Tests on 20-bit Words	0.312*	0.758*
Monkey Test OPSO	0.436*	0.751*
Monkey Test OQSO	0.742*	0.835*
Monkey Test DNA	0.688*	0.231*
Count the 1's in a Stream of Bytes	0.664	0.789
Count the 1's in Specific Bytes	0.586*	0.680*
Parking Lot Test	0.433	0.117
Minimum Distance Test	0.411	0.03
Random Spheres Test	0.788	0.50
The Squeeze Test	0.841	0.449
Overlapping Sums Test	0.173	0.003
Runs Up and Down Test	0.191	0.859
The Craps Test	0.443*	0.539*
Overall KS p-value	**0.778**	**0.792**

Table 6.10: Results Obtained with David Sexton's Battery (LAMED)

	LAMED	**LAMED-EPC**
Test	**p-value**	**p-value**
Bit Runs Test	0.925*	0.726*
Frequency Test	0.016*	0.962*
Bit Test	0.375*	0.748*
Sum Test	0.841*	0.18*
Matrix Test	0.432*	0.857*
Prediction Test	0.119*	0.529*
And Test	0.778*	0.856
Up/Down Test	0.699*	0.355*
Rect. Distance Test	0.018	0.798
Collision Test	0.577*	0.362*
Offset Xor Test	0.865*	0.723*
Mod Test	0.230*	0.637*

Table 6.11: Results Obtained with the NIST Suite (LAMED)

	LAMED	**LAMED-EPC**
Test	**Proportion**	**Proportion**
Frequency	0.98	0.98
Block-frequency	0.98	1.00
Cumulative-sums	0.98, 0.98	0.98, 0.98
Runs	1.00	0.99
Longest-run	1.00	0.99
Rank	0.98	0.99
Fft	0.99	0.98
Overlapping-templates	0.98	1.00
Universal	0.96	0.98
Apen	0.99	1.00
Serial	0.97, 1.00	0.99, 0.97
Linear-complexity	0.99	0.98
Random-excursions	0.97, 0.98 1.00, 0.97 1.00, 1,00 0.97 1.00	0.97, 1.00 0.97 0.96 1.0, 0.98 0.97 0.98
Random-excursions-variant	1.00, 1.00, 1.00 0.98, 1.00, 1.00 1.00, 1.00, 1.00 1.00, 1.00, 1.00 1.00, 0.98, 0.97 0.98, 1.00, 0.97	1.00, 1.00, 1.00 0.98, 0.98, 1.00 1.00, 0.98, 0.98 1.00, 1.00, 1.00 1.00, 0.98, 0.98 1.00, 1.00, 1.00

Table 6.12: Serial Correlation Test (LAMED-EPC)

	LAMED-EPC		
Experiment	**Bit**	**Byte**	**16-bit**
1-Experiment	-0.000002	0.000154	0.000065
2-Experiment	-0.000015	0.000028	0.000028
3-Experiment	-0.000013	-0.000008	-0.000157
4-Experiment	-0.000006	0.000079	0.000074
5-Experiment	-0.000053	0.000088	0.000047

EPC is, in fact, bounded by:

$$\frac{0.96}{2^{16}} < P_{LAMED-EPC}(RN16 = j) < \frac{1.05}{2^{16}} \tag{6.67}$$

Another interesting property asked for in the EPC-C1G2 standard concerns the probability of predicting a random number. The specification in this context determines that "a RN16 could not be predicted with probability greater than 0.025% if the outcomes of prior draws from RNG, performed under identical conditions, are known". In order to check if this property holds, some tests have been completed:

Serial Correlation Test This quantity measures the extent to which each n-bit output depends upon the previous n-bit output. For random sequences, this value (which can be positive or negative) should be very close to zero. As an example, a non-random n-bit stream such as a counter will yield a serial correlation coefficient of about 0.5. Five files of 2^{25} bytes haven been generated (with different secret keys and initialization vectors). From each file, the serial correlation (bit, byte, 16-bit) has been obtained. *Table 6.12* shows the obtained results.

Bit-Byte Prediction Test from David Sexton's battery Many algorithms are used to predict the value of each bit (respectively, byte) of the sequence from the beginning of the sequence to the end. Specifically, various algorithms are used to predict the value of each bit (byte) from the beginning of the sequence to the end. In a random sequence the probability of success of any algorithm is 1/2 (respectively 1/256). The number of successes is counted and a chi-squared statistic is computed. The following tests have been made:

- Bit Prediction A Test: the numbers of zeros and ones in all the previous bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted. Otherwise the prediction is the same as for the previous bit.
- Bit Prediction B Test: the numbers of zeros and ones in the previous 9 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Bit Prediction C Test: the numbers of zeros and ones in the previous 17 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Bit Prediction D Test: the numbers of zeros and ones in the previous 33 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Bit Prediction E Test: the numbers of zeros and ones in the previous 65 bits are counted. If the ones outnumber the zeros, a zero is predicted; if the zeros outnumber the ones, a one is predicted.
- Byte Prediction A Test: the next byte is predicted to be equal to all the previous bytes bitwise XORed together. The first byte of the sequence is predicted to equal zero.
- Byte Prediction B Test: the next byte is predicted to be equal to the sum of all the previous bytes, modulo 256. The first byte of the sequence is predicted to equal zero.
- Byte Prediction C Test: the next byte value is predicted to be zero until the first zero is found. From that point on, the next byte value is predicted to be the byte value whose last appearance was furthest back in the sequence.
- Byte Prediction D Test: a given byte value is predicted to be followed by the same byte value it was followed by the last time it appeared in the sequence. A byte value that has not previously appeared in the sequence is predicted to be followed by the byte value of the first byte in the sequence. The first byte of the sequence is predicted to equal zero.
- Byte Repetition Test: This test is equivalent to a byte prediction test where each byte is predicted to be equal to its preceding

Table 6.13: Prediction Tests as in David Sexton's Battery (LAMED-EPC)

	LAMED-EPC
Test	**p-value**
Bit Prediction A Test	0.8421
Bit Prediction B Test	0.6966
Bit Prediction C Test	0.8499
Bit Prediction D Test	0.8081
Bit Prediction E Test	0.4742
Byte Prediction A Test	0.3263
Byte Prediction B Test	0.6074
Byte Prediction C Test	0.5686
Byte Prediction D Test	0.3254
Byte Repetition Test	0.4184

byte. The first byte of the sequence is predicted to equal the last byte of the sequence.

A file of 2^{32} bytes was generated to check these prediction tests, and the results obtained are summarized in *Table 6.13*.

Lineal Predictor Another interesting approach for finding a good predictor for a given function is to try to approximate it by a linear relation, similarly to what is done in linear cryptanalysis for block ciphers [150, 202]. In order to obtain the linear bias of LAMED-EPC, the following experiment has been accomplished: two 16-bit masks (A, B) have been randomly picked, and two consecutive outputs have been generated (O_i, O_{i+1}). With these two masks and outputs, the equality $A * O_i = B * O_{i+1}$ is evaluated. The process is repeated 2^n times, counting the numbers of successes (m). The $*$ symbolizes scalar product, with a mod 2 operation being carried out after addition. The bias is defined as:

$$BIAS = \frac{1}{2^{-log_2(|\frac{m}{2^n} - \frac{1}{2}|)}} \tag{6.68}$$

We have randomly tested many pairs of different masks, A and B. For each pair, 2^{25} 16-bit outputs have been generated, and for consecutive outputs the previous expression ($A * O_i = B * O_i$) was evaluated. From the above results, we can gather that the bias of LAMED-EPC

is limited by:

$$BIAS_{LAMED-EPC} < \frac{1}{2^{11.77}} \tag{6.69}$$

which implies that the security margin, given by the number of observations needed for predicting the next output value with a good accuracy is around $(2^{11.77})^2$, which is well over the 2^{-16} limit that any protocol using this PRNG will have (see [150, 202]).

Differential Analysis A differential analysis is a form of attack in which the differences between consecutive values are used to gain additional knowledge about the system. Two different analysis have been proposed, where O_i refers to the i^{th} output provided by LAMED-EPC:

- The simpler and generally most useful is the XOR analysis where $O_i \oplus O_{i+1}$ is studied.
- Another standard analysis is that of the difference $(O_i - O_{i+1})$ mod 2^{16}.

For each of the two analysis, the following experiments were carried out four times: First, the secret key and initialization vector were randomly set. Next, a sequence consisting of 2^{25} outputs was generated. Finally, the xor and difference values were computed. In *Table 6.14*, the statistical properties are summarized.

Summarizing, the probabilities associated with LAMED output are well within the limits set by the specification. Our prediction analysis gives no indication that the output could be predicted significantly better by the knowledge of prior outputs without knowing the secret key and the IV, than just by chance. Although the period of LAMED has not been exactly determined, in *Section 6.5.4* a file of 2^{30} bytes (4 Gb) was analyzed without finding any evidence that those bytes behave differently from what would be expected from a random variable, thus proving that the period was greater and, in any case, sufficiently great for the intended application. Let's consider the following simple scenario: a tag having LAMED-EPC on chip, and a reader which interrogates this tag every 5 milliseconds. Under these conditions, the reader could continuously interrogate the tag for

Table 6.14: Analysis of the XOR and Substraction (LAMED-EPC)

Experiment	**Xor ($0_i \oplus 0_{i+1}$)**	**Difference ($0_i - 0_{i+1}$)**
Experiment #1		
Entropy	7.999999 bits/byte	7.999999 bits/byte
Compression Rate	0%	0%
χ^2 Statistic (byte)	267.81 (50%)	259.79 (50%)
χ^2 Statistic (16-bit)	65370.5898 (67.55%)	65209.1562 (62.26%)
Arithmetic Mean	127.5060	127.4997
Monte Carlo π Estimation	3.141522675 (0.00%)	3.141581433 (0.00%)
Serial correlation Coefficient	-0.000041	-0.000093
Experiment #2		
Entropy	7.999999 bits/byte	7.999999 bits/byte
Compression Rate	0%	0%
χ^2 Statistic (byte)	247.43 (50%)	267.88 (50%)
χ^2 Statistic (16-bit)	65484.3281 (55.60%)	65607.109375 (42.14%)
Arithmetic Mean	127.4969	127.4977
Monte Carlo π Estimation	3.142054749 (0.01%)	3.141537534 (0.01%)
Serial correlation Coefficient	0.000097	0.000049
Experiment #3		
Entropy	7.999999 bits/byte	7.999999 bits/byte
Compression Rate	0%	0%
χ^2 Statistic (byte)	240.76 (50%)	264.96 (50%)
χ^2 Statistic (16-bit)	65673.964844 (35.09%)	65182.406250 (83.56%)
Arithmetic Mean	127.4910	127.4954
Monte Carlo π Estimation	3.141983490 (0.01%)	3.141622471 (0.00%)
Serial correlation Coefficient	-0.000078	0.000112
Experiment #4		
Entropy	7.999999 bits/byte	7.999999 bits/byte
Compression Rate	0%	0%
χ^2 Statistic (byte)	259.95 (50%)	259.79 (50%)
χ^2 Statistic (16-bit)	65346.839844 (69.88%)	65614.957031 (41.29%)
Arithmetic Mean	127.4997	127.5001
Monte Carlo π Estimation	3.142095337 (0.02%)	3.141177521 (0.01%)
Serial correlation Coefficient	0.000064	0.000044

at least fifteen days, which clearly meets the needs of the vast majority of applications.

6.5.6 Hardware Complexity

In this section, we explain in detail one architectural design for LAMED. As mentioned in *Chapter 4*, Class-1 Generation-2 tags have severe temporal requirements, for around 450 tags should be readable every second. Power consumption is another important restriction. Tags are passive, so we should limit their power consumption as much as possible. One of the parameters with a major influence on this target is clock frequency. Following other authors in the RFID area [72], we assume that clock frequency must be in the range of KHz, at some value around 100 KHz, implying a clock cycle consumption of 0.01 ms. With these conditions, a tag can use up to 200 clock cycles (2 ms) for the whole random number generation phase. To obey all these restrictions, we have decided to process 32-bit streams in parallel. Next, a code for the implementation of LAMED is included, where ^ is the **xor** operator and `vrotdk(v,k)` means rotations of v, k times.

```
------------------------------------------------------------
                  #1 If n is odd
                  #2     a0=a1+iv
                  #3     a1=out^s
                  #4 If n is even
                  #5     a0=a1^iv
                  #6     a1=out+s
------------------------------------------------------------
  #1  aux1 = a0 + a1;              #12 aux3 = aux3 ^ aux1;
  #2  aux2 = a0 ^ a1;              #13 aux3 = vrotdk(aux3,3);
  #3  aux3 = vrotdk(aux1,5);       #14 aux3 = aux3 + a1;
  #4  aux3 = aux3 + aux2;          #15 aux3 = vrotdk(aux3,2);
  #5  aux3 = vrotdk(aux3,3);       #16 aux3 = aux3 + aux1;
  #6  aux3 = aux3 ^ aux1;          #17 aux3 = vrotdk(aux3,4);
  #7  aux3 = vrotdk(aux3,4);       #18 aux3 = aux3 ^ a1;
  #8  aux3 = a1 + aux3;            #19 aux3 = vrotd(aux3);
  #9  aux3 = vrotdk(aux3,2);       #20 aux3 = aux3 + aux2;
  #10 aux3 = aux3 + aux1;          #21 aux3 = vrotdk(aux3,2);
  #11 aux3 = vrotdk(aux3,2);       #22  out = aux1 ^ aux3;
------------------------------------------------------------
```

The architecture of LAMED, see *Figure 6.10*, can be divided into four main parts:

Input Selection Unit The PRNG will be initialized with a 32-bit initialization vector (iv). Furthermore, the tag has s stored, a 32-bit secret which is only known by the tag and authorized readers. After initialization, the state of the PRNG will be updated as in *Expression 6.65* and *6.66*.

Arithmetic Logic Unit (ALU) Due to the structure of the PRNG, we have only included the **xor** and **sum** operators.

Registers We have used three registers for the core of our PRNG. Two will be used to store `a0+a1` (`aux1`) and `a0^a1` (`aux2`) for the generation process of each 32-bit output. Additionally, once an output has been obtained, these registers will be used for the temporal storage of the new two inputs (updated state) to the PRNG. The third register will be used to execute the right rotations and to store the intermediate results. `aux1` and `aux2` must be updated after the generation of a new 32-bit stream. Moreover, two additional registers will be used to store the initialization vector (iv) and the secret (s).

16-bit Unit Output This unit performs a split XOR operation. The 32-bit output is divided in two halves, $MSB_{31:16}$ and $LSB_{15:0}$, and the XOR of these two halves will be outputted.

Directly derived from all of the above, we reckon that 186-194 clock cycles (1.86-1.94 ms) are needed for generating each 32-16 output with LAMED or LAMED-EPC. Besides, these results imply a throughput of between 17.2 and 8.2 kbps respectively. We can then conclude that the temporal requirements are accomplished with enough margin in both cases.

Another important aspect we should consider its gate counting. An overestimation of this factor is presented in *Table 6.15*. In this calculation, an extra 20% of logic gates are added for control functions, and 8 additional gates are needed for implementing a flip flop as in [90].

Table 6.15: Number of Logic Gates (LAMED)

	LAMED		**LAMED-EPC**	
Architecture Units	**Gate Counting**		**Gate Counting**	
Arithmetic Logic Unit	XOR	32 LG	XOR	32 LG
	SUM	192 LG	SUM	192 LG
16-bit Unit	—		16 LG	
Update Unit	10 LG		10 LG	
Registers	Aux1-2	512 LG	Aux1-2	512 LG
	Aux3	264 LG	Aux3	264 LG
	Aux4-5	512 LG	Aux4-5	512 LG
Control (20%)	44 LG		50 LG	
Total	**1566 Logic Gates**		**1585 Logic Gates**	

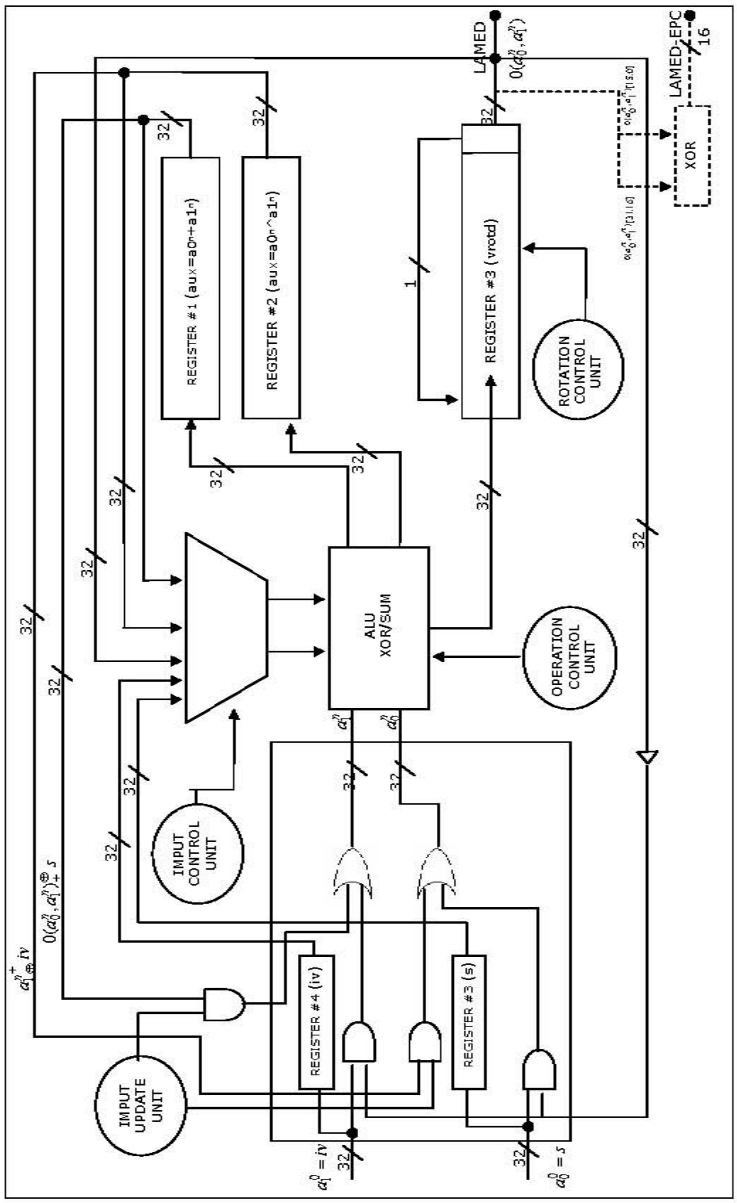


Figure 6.10: LAMED - Logic Scheme

Chapter 7

Conclusions

7.1 Introduction

Barcodes are currently the dominant identification system. They are an established technology, and this is tending to delay the implantation of RFID technology. However, RFID technology offers important advantages, although these advantages came with a cost, which is payed in, for example, various types of security drawbacks.

One of the main obstacles for the widespread adoption of RFID systems is its cost. At the moment, all costs occur in the production phase, which implies a significant limitation. Experts believe that significant cost reduction might be obtained by spreading it over the complete life cycle. Additionally, all technological change brings companies associated costs. Companies are tackling the problem in different ways. Whereas some simply wait to see what the competition does, some are looking at the Return of Inversion (ROI) associated with the introduction of RFID. Indeed, RFID penetration highly differs from country to country. In Spain, companies are mainly developing pilot projects to analyze this technology (i.e. Pascual Group, KH Lloreda Group etc.).

7.2 Attacking RFID Systems

There is some confusion about the real risks associated with RFID technology. The vast majority of articles found in the mass media exaggerate them significantly. However, this does not mean that RFID systems cannot be attacked in multiple ways, some of them new. Indeed, RFID is a wireless technology and the tags have circuitry limitations, so some of the attacks are quite similar to those well-know from other technologies such as wireless and smart cards, respectively. Therefore only a little part of the attacks associated with this technology are new.

When asking consumers about the security threats related to RFID technology, most reveal a concern for "privacy". It is an obvious response because RFID is destined to be a very pervasive technology. However, these are not the only problems that should be taken into account when designing an RFID system. As RFID systems are composed of three main components (tag, reader and back-end database), each component should be analyzed in detail. *Chapter 3* made a comprehensive analysis of each possible risk for any of the three main components.

Another important aspect is the varying security level needed for different RFID systems. From a theoretical point of view, it would be useful for all systems to be resistant to active and passive attacks. However, there are different classes of RFID tags and their field of application may not be the same. Indeed, a system's security will be a compromise between confidentiality, integrity and availability for the intended application at the objective cost.

7.3 Standards and Proposed Solutions

The introduction of any technology is accompanied by the development of standards. The EPC Class-1 Generation-2 standard can be considered the universal standard for low-cost RFID tags. In *Chapter 4*, the security of this standard is analyzed in detail. Our analysis points out important security faults. In spite of this, we see the specification as a good starting-point for constructing a secure standard for low-cost RFID tags.

Motivated by the unsatisfactory level of security of the EPC-C1G2 specification, several researchers have published slight modifications to the standard. However, all of these schemes have proved unsuccessful from the security perspective. In this thesis (see *Chapter 6*) we have presented the cryptanalysis of the two most recent proposals in this area.

Apart from the EPC-C1G2, there are other standards associated with this technology because of its heterogeneity. These standards can be classified in five main groups (contactless integrated circuit cards, animal identification, item management, near field and EPC), as detailed in *Chapter 2*. Additionally, regional regulations (ECC and ETSI in Europe) impose restrictions on the implantation of these systems. These restrictions are mainly related to the planning and use of the radio-electric spectrum.

RFID technology is not new; the first article about it was published in the fifties. However, as a research subject on security, it has received considerable attention since 2003. In 2003 and 2004, around 10 and 30 papers respectively were published in this research area. This increased to 75, 90, and 85 in the next three years (`http://www.avoine.net/rfid/download/bib/bibliography-rfid.pdf`; consulted in June 2008).

Indeed, RFID is now a topic of interest in a great number of conferences. Many of these works focus on security, and the range of the proposals is very wide. Some authors have proposed the use of non-cryptographic solutions, such as Faraday cage, active jamming, bill of rights, etc. Other authors have proposed solutions based on cryptographic techniques. These solutions are very diverse, some of them being based on block-ciphers, pseudo-random number generators, and even public-key cryptography.

However, the most commonly proposed solution is based on hash functions. All of these protocols share the common characteristic of being single round protocols. In a different approximation, the family of human based protocols are based on multiple execution of a very simple round.

Additionally, only three entities (back-end data base, reader and tag) are involved in all of the aforementioned protocols. However, there are additional applications areas beyond the mutual authentication such as the protocols that are focused on the problem of providing a proof for the simultaneous reading of two or more RFID tags.

7.4 Lightweight Protocols

The majority of proposals to make RFID tags secure make two important errors. First, they propose a protocol for RFID tags without specifying for which class of RFID tag the protocol is intended. This is a very important point, as the number of available resources (memory, circuitry, power, etc.) will highly depend on this. So not all tags will support the same kind of operations. Additionally, each RFID class should have a different security level. Secondly, the proposed protocols are not realistic about tag resources. As we have already mentioned, the most widely-adopted proposal is based on hash functions. In spite of this, many authors claim that their protocols are appropriate for low-cost RFID tags. However, a maximum of 4K gates can be devoted to security functions in this class of tag. As we saw in *Chapter 6*, considerable resources (over 9K gates) are needed to implement traditional cryptographic hash functions. On the other hand, lightweight cryptographic hash functions are not proposed. Therefore, lightweight cryptography is an imperative.

In order to avoid past errors, the requirements and the restrictions of the system were fixed at the start. In our work, we have discriminated between two classes of tags: low-cost and moderate-cost RFID tags. For each class, the following characteristics have been specified: power source, circuitry, reading distance, price, etc. (see *Chapter 6*). The main difference between these two tag classes is the number of logic gates that can be dedicated to security processes. Low-cost tags only use efficient operations and moderate-cost tags support lightweight cryptographic primitives. Due to this considerable difference, low-cost RFID are only resistant to passive attacks, but moderate-cost RFID tags are resistant to passive and active attacks.

A new ultralightweight authentication protocol for low-cost RFID tags, named Gossamer, has been proposed by the author of this thesis. Gossamer is inspired by the UMAP family of protocols and the recently proposed SASI protocol. Indeed, Gossamer attempts to avoid the errors of its predecessors. The main weak points of the UMAP family were that they were completely based on triangular functions and there was an incorrect use of the bitwise OR and AND operations. SASI was a novel proposal that

introduced rotation within the set of operations supported on the tag. This initially impeded attack against SASI because rotation is a non-triangular function. However, SASI was not designed with sufficient care and a passive attacker can disclose the secret ID, as shown in *Chapter 6*. Additionally, active attacks against it have just been published (desynchronization, identity disclosure, etc.). Our protocol Gossamer is a response to all these previous failings. Due to the severe restrictions of low-cost RFID tags, only simple operations were employed. Specifically, the protocol requires simple bitwise XOR, addition and left rotation on tags. Random number generation is demanded of the reader because of its computational cost. Finally, a specifically designed and very lightweight operation named $MixBits$ has been added to tag operations in order to introduce a diffusion effect in the messages generated by these restricted devices. Storage requirements are of the same order as SASI. From the perspective of communication cost, Gossamer is very efficient; only 4 messages (424 bits) are transmitted over the channel for each authentication phase. In conclusion, Gossamer is very efficient from a performance perspective, and the use of a dual rotation and $MixBits$ function seems to guarantee an adequate security level.

A new protocol conforming to moderate-cost tag specifications has also been proposed. It should be resistant to passive attacks but also to active attacks. Another interesting feature is that tags could be deactivated temporarily without loss of data. The protocol is inspired in a remote authentication protocol for smart cards. Smart-card technology is a mature technology that has important similarities with RFID technology. The proposed protocol is inspired specifically in Shieh et al.'s protocol, which is considered one of the most secure and efficient. The protocol has been adapted to moderate-cost RFID systems. As in Shieh et al.'s protocol, it is based on the use of a secure hash function. However, traditional cryptographic hash functions, such as SHA-256 or MD5, exceed the capabilities of this class of tag. A new lightweight hash function, named Tav-128, has been proposed. Tav-128 can be implemented with around only 2.6K gates, and 1568 clock cycles are needed. Although further security analysis of the new hash function is necessary, the preliminary analysis indicates an adequate security level for the intended application (mutual authentication of moderate-cost tags). Additionally, it is assumed that tags can generate random numbers. The use of pseudo-random number generators for

low-cost RFID tags was ratified in the EPC-C1G2 specification. The standard specifies three randomness conditions that the generators must meet. However, no algorithm is proposed and so far no public algorithm conforming to EPC-C1G2 has been published. This was the motivation for the proposal of the new PRNG conforming to the standard. A 32-bit PRNG (LAMED) and a 16-bit PRNG (LAMED-EPC) has been designed. The generators passed some very demanding batteries of randomness tests (ENT, DIEHARD, NIST, and SEXTON). In addition, LAMED-EPC's conformation to the specification was considered. Finally, the hardware complexity was analyzed, reckoning 1.6K gates for implementation. So less than 5K gates are needed to support a lightweight hash function and lightweight PRNG. In short, the proposed protocol is considered adequate for moderate-cost RFID tags.

7.5 Social Problems

Even if technological problems can eventually be solved, widespread adoption of RFID systems will not occur unless the public is educated as to their potential benefits and risks, and unless security can be guaranteed.

Gunter et al. performed an interesting empirical study about RFID and perception control from the consumer's perspective [87]. Perception control deals with the belief people have that the electronic environment will only act in ways explicitly allowed. Two main factors are considered responsible for the perception of loss of privacy. Firstly, "being accessed" reflects a need to control entry to the environment. An attacker may determine personal behavior and track movements of tag owners without being detected. Secondly, "information dissemination, use, and maintenance", refers to the vast amount of data pertaining to individuals that can be collected. Two different Privacy-Enhancing Technologies (PET) were offered to users. In the user model, users have full control over the system (i.e. authentication mechanism). In the agent model, on the other hand, access control is delegated to an agent (i.e. a privacy-preserving identity-management system). They analyzed empirically which of these two mechanisms would increase consumer acceptance of RFID technology. *Figure 7.1* shows the perceived control for each model. Additionally, the study pointed out that

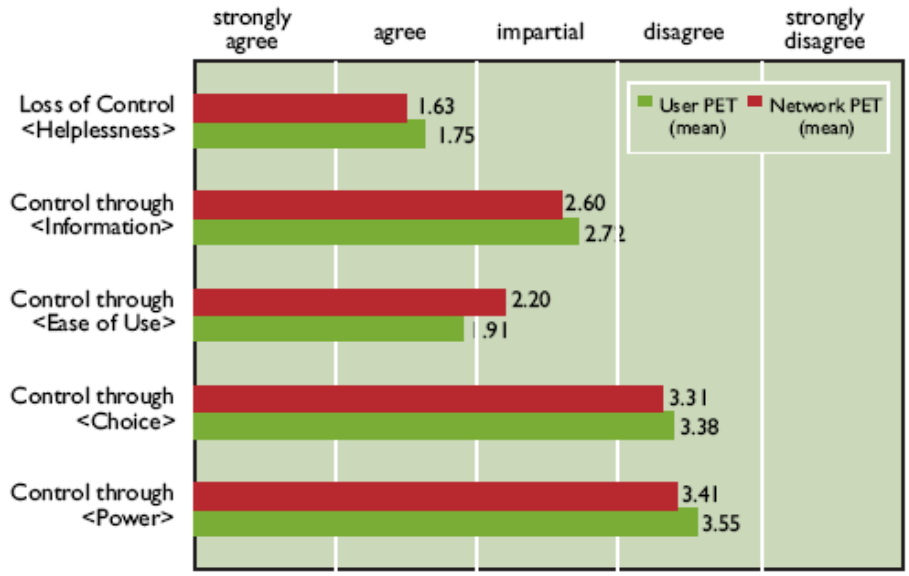


Figure 7.1: Perception of Control [87]

users preferred to deactivate tags after buying a product (73.4%). 18% of users trusted in PET (user or agent model), and a further 8.6% were undecided.

In conclusion, technological advances should be smoothly integrated into society. It is a point that the industry ought not neglect. It is, after all, consumers who decide the fate of any given technology.

7.6 Future Works

Our future work includes several research lines, as described in the following:

Lightweight Protocols An important part of our research activity is centered on the design of ultralightweight protocols for low-cost RFID tags. Different schemes were proposed, broken (often by other researchers), and later improved to make the scheme immune to the particular attacks. The Gossamer protocol constitutes our last proposal and has not been broken yet. We have analyzed its security against all the standard attacks: privacy, tracing, replay attacks, etc.

This analysis should be completed by a formal analysis. Belief logics (BAN, GNY, etc.) or formal frameworks based on other verification paradigms (i.e. Casper/FDR, OFMC, etc.) may be employed for this task.

A natural direction for further research is also the design of a new protocol based on a provably-hard problem (i.e. NP-Complete). An example of this family of protocols are the human protocols (HB and their variants). Specifically, the aforementioned protocols are based on the problem of learning parity with noise (LPN). In spite of the fact that the vas majority of the HB variants are currently broken, this research area is certainly promising and should be analyzed in depth, as other (and perhaps more appropriate) hard problems could be employed as the basis of new lightweight protocols.

Cryptographic Primitives The most common solution to the mutual authentication problem (which is central in RFID security) found in the literature is based on the use of a hash function. However, lightweight cryptographic hash functions are not suggested and standards solutions exceed by far the capabilities of low-cost RFID tags. On the other hand, the use of PRNGs has been ratified by the EPC-C1G2 specification, but no specific algorithm is proposed. Recently, the Philips Mifare cards have been broken in part due to the weaknesses in the underlying PRNG (which was a proprietary algorithm) [77, 117, 159]. This justifies too the necessity of lightweight PRNGs, and also points out that algorithms should be made as widely known as possible for general scrutiny. Summarizing, the design of lightweight primitives (particularly hash functions and PRNGs) for low-cost RFID tags is therefore imperative. We have made some interesting progress in this research area, and keep on working along these lines.

EPC-C1G2 Standard Without significantly altering the framework of the EPC-C1G2 standard, some authors have proposed new schemes to correct its unsatisfactory security level. However, all of these schemes have proved unsuccessful from a security perspective, in some cases due to our attacks. The design of protocols attempting to obtain a satisfactory security level within the framework of this specification

is a though-provoking challenge.

New Problems Two entities (a tag and a reader) are involved in the majority of RFID protocols. Some, however, consider the simultaneous reading of multiple tags (the so-called "yooking proofs" and its variants). A close inspection of these protocols should be accomplished. Additionally, more research effort should be made on distance-bounding protocols [157] (schemes mainly used to avoid relay attacks).

Noisy cryptography RFID technology uses the radio channel for communications. The inherent noise affecting the communication link may be employed to design an authentication protocol. This is an area where almost no work has been published. A good starting point could be the study of the works by Chabanne et al. [44] and by Casteluccia et al. [42].

Bibliography

[1] Universal declaration of humman rights, Article 12, 1948.

[2] EU Directive 95/46/EC - Data Protection Directive. Official Journal of the European Communities, November 23, 1995.

[3] ISO/IEC 9798 Information Technology - Security techniques - Entity authentication. http://www.iso.org, 1995.

[4] Identification cards - Contacless integrated circuits cards - Proximity cards. http://www.wg8.de/sdi.html, 2000.

[5] Anti-skimming in Japan. http://www.future.iftf.org/index.html, 2005.

[6] David Sexton's battery. http://www.geocities.com/da5id65536, 2005.

[7] FoeBuD. http://www.foebud.org/rfid, 2005.

[8] Recomendation for key management. Technical Report Special Publication 800-57 Draft, National Institute of Technology, 2005.

[9] The `lil-gp` genetic programming system. http://garage.cps.msu.edu/software/lil-gp/lilgpindex.html, 2005.

[10] Advanced security mechanisms for machine readable travel documents - Extended Access Control (EAC) version. 1.0.1. Technical guideline TR-03110, Federal Office of Information Security, 2006.

[11] Machine readable travel documments, Doc. 9303. http://www.mrtd.icao.int, 2006.

[12] Year report on algorithms and keysizes. Technical Report IST-2002-507932, ECRYPT, 2006.

[13] Easing traveling in London's congested public transport network. http://www.mifare.net/showcases/london.asp, 2007.

[14] Ecrypt stream cipher project. http://www.ecrypt.eu.org/stream/, 2007.

[15] Envelope to help you do it with your security, privacy, and discretion intact. http://www.emvelope.com, August 2007.

[16] Verichip corporation. http://www.verichipcorp.com, 2007.

[17] Michelin embeds RFID tags in tires. RFID Journal, January 2003.

[18] Radio frequency identification ready to deliver, Armed forces communications and electronics association. http://www.afcea.org, January 2005.

[19] Vatican library employs RFID tracking. RFID Gazette, July 2004.

[20] http://www.theregister.co.uk, June 27, 2003.

[21] K. Albrecht and L. McIntyre. *SPYCHIPS: How Major Corporations and Government Plan to Track your Every Move with RFID*. Nelson Communications, Inc., 2005.

[22] Anarchriz. CRC and how to reverse it. http://www.woodmann.com/fravia/crctut1.htm, 1999.

[23] D. Atkins and R. Austein. Threat analysis of the domain name system (DNS). In *Request for comments - RFC 3833*, 2004.

[24] J. Atkinson. Contactless credit card consumer report. http://www.findcreditcards.org, 2006.

[25] T. Aura. Strategies againts replay attacks. In *Proc. of CSF'97*. IEEE Computer Society, 1997.

[26] G. Avoine and P. Oechslin. A scalable and provably secure hash-based RFID protocol. In *Proc. of PERSEC'05*, pages 110–114. IEEE Computer Society, 2005.

[27] D. Bailey and A. Juels. Shoehorning security into the EPC standard. Manuscript in submission, 2006.

[28] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy. Breaking LMAP. Hand. of RFIDSec'07, 2007.

[29] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy. Passive attack against the M2AP mutual authentication protocol for RFID tags. Proc. of First International EURASIP Workshop on RFID Technology, 2007.

[30] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-key cryptography for RFID tags. Proc. of PerCom'07, 2007.

[31] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede. Low-cost elliptic curve cryptography for wireless sensor networks. In *Proc. of ESAS'06*, volume 4357 of *LNCS*, pages 6–17. Springer-Verlag, 2006.

[32] Boycott Benetton. http://www.boycottbenetton.com/, 2003.

[33] A. Beresfor and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):1536–1268, 2003.

[34] A. Biryukov and A. Shamir. Cryptanalytic time-memory-data tradeoffs for stream ciphers. In *Proc. of Advances of Cryptology-ASIACRYPT*, volume 1976 of *LNCS*, pages 1–13. Springer-Verlag, 2000.

[35] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *Proc. of CHES'07*, volume 4727 of *LNCS*, pages 450–466. Springer-Verlag, 2007.

[36] L. Bolotnyy and G. Robins. Physically unclonable function-based security and privacy in RFID systems. In *Proc. of PerCom'07*, pages 211–220. IEEE Computer Society, 2007.

[37] S. Bono, M. Greem, A. Stubblefield, A. Juels, A. Rubin, and M. Syzdlo. Security analysis of a cryptographically-enabled device. In *Proc. of SSYM'05*. Usenix Association, 2005.

[38] J. Bringer, H. Chabanne, and E. Dottax. HB^{++}: a lightweight authentication protocol secure against some attacks. In *Proc. of SecPerU'06*. IEEE Computer Society, 2006.

[39] P. Bulens, K. Kalach, F.-X. Standaert, and J.-J. Quisquater. FPGA implementations of eSTREAM phase-2 focus candidates with hardware profile. http://www.ecrypt.eu.org/stream/.

[40] D. Carluccio, K. Lemke, and C. Paar. Electromagnetic side channel analysis of a contactless smart card: first results. Hand. of RFIDSec'06, 2006.

[41] CASPIAN. http://www.nocards.org/, 2005.

[42] C. Castelluccia and G. Avoine. Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags. In *Proc. of CARDIS'06*, volume 3928 of *Lecture Notes in Computer Science*, pages 289–299. Springer-Verlag, 2006.

[43] Auto-ID Center. 900 MHz class 0 radio frequency (RF) identification tag specification. Draft, March 2003.

[44] Hervé Chabanne and Guillaume Fumaroli. Noisy cryptographic protocols for low cost rfid tags. Hand. of the Ecrypt Workshop on RFID and Lightweight Crypto, July 2005.

[45] C. Chatmon, T. Van Le, and M. Burmester. Secure anonymous RFID authentication protocols. Technical Report TR-060112, 2006.

[46] H.-Y. Chien. SASI: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, 2007.

[47] H. Y Chien and C. H Chen. Mutual authentication protocol for RFID conforming to EPC class-1 generation-2 standards. *Computer Standards and Interfaces, Elseiver Science Publishers*, 29(2):254–259, 2007.

[48] H.-Y. Chien and C.-W. Huang. Security of ultra-lightweight RFID authentication protocols and its improvements. *SIGOPS Oper. Syst. Rev.*, 41(4):83–86, 2007.

[49] E. Y. Choi, S. M. Lee, and D. H. Lee. Efficient RFID authentication protocol for ubiquitous computing environment. In *Proc. of SECUBIQ'05*. Springer-Verlag, 2005.

[50] J. Cichon, M. Klonowski, and M. Kutylowski. Privacy protection in dynamic systems based on RFID tags. In *Proc. of PerSec'07*, pages 235–240. IEEE Computer Society, 2007.

[51] J. Collins. RFID-Zapper shoots to kill. RFID Journal, 2006.

[52] Y. Cui, K. Kobara, K. Matsuura, and H. Imai. Lightweight asymmetric privacy-preserving authentication protocols secure against active attack. In *Proc. of PerSec'07*. IEEE Computer Society, 2007.

[53] Wheeler. D. and R. Needham. TEA: A tiny encryption algorithm. Proc. of FSE'04, 1994.

[54] Wheeler. D. and R. Needham. TEA extensions. Technical report, Computer Laboratory, University of Cambridge, 1997.

[55] Wheeler. D. and R. Needham. Correction to XTEA. Technical report, Computer Laboratory, University of Cambridge, 1998.

[56] G. Danezis, S. Lewis, and R. Anderson. How much is location privacy worth. In *Proc. of Workshop of Economics of IS'05*, 2005.

[57] E. *Öztü*rk, B. Sunar, and E. Savascedil. Low-power elliptic curve cryptography using scaled modular arithmetic. In *Proc. of CHES'04*, volume 3156 of *LNCS*, pages 92–106. Springer-Verlag, 2004.

[58] C. De Cannière and B. Preneel. Trivium specification. http://www.ecrypt.eu.org/stream/, 2005.

[59] T. Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Proc. of SECURECOMM'05*. IEEE Computer Society, 2005.

[60] GS1 - EAN International. http://www.ean-int.org/, June 2005.

[61] Class-1 Generation-2 UHF air interface protocol standard version 1.0.9: "Gen2". http://www.epcglobalinc.org/standards/, 2005.

[62] EPC Application level events version 1.0. http://www.epcglobalinc.org/standards/, 2005.

[63] EPC ONS standard version 1.0. http://www.epcglobalinc.org/standards/, 2005.

[64] EPCglobal. http://www.epcglobalinc.org/, 2005.

[65] EPCglobal Architecture framework. http://www.epcglobalinc.org/standards/, 2005.

[66] EPC Reader protocol standard version 1.1. http://www.epcglobalinc.org/standards/, 2006.

[67] EPC Tag data standard version 1.3. http://www.epcglobalinc.org/standards/, 2006.

[68] EPC Information services version 1.0. http://www.epcglobalinc.org/standards/, 2007.

[69] California Senate Bill 682. http://www.epic.org/privacy/rfid/, February 22, 2005.

[70] B. Fabian, G. Oliver, and S. Spiekermann. Security analysis of the object name service for RFID. In *Proc. of SecPerU'05*. IEEE Computer Society, 2005.

[71] M. Feldhofer. A proposal for an authentication protocol in a security layer for RFID smart tags. In *Proc. of MELECON'04*. IEEE Computer Society, 2004.

[72] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In *Proc. of CHES'04*, volume 3156 of *LNCS*, pages 357–370. Springer-Verlag, 2004.

[73] M. Feldhofer and C. Rechberger. A case against currently used hash functions in RFID protocols. Hand. of RFIDSec'06, 2006.

[74] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. AES implementation on a grain of sand. In *Proc. on Information Security*, volume 152, pages 13–20. IEEE Computer Society, 2005.

[75] FOEBUD. http://www.foebud.org/, 2006.

[76] R. Forré. The strict avalanche criterion: Spectral properties of boolean functions and an extended definition. In *Proc. of CRYPTO'88*, pages 450–468. Springer-Verlag, 1990.

[77] F. D. Garcia, G. de Koning Gans, R. Muijrers, P. van Rossum, R. Verdult, and R. Wichers Schreur. Dismantling MIFARE Classic. In *Proc. of ESORICS'08*, LNCS. Springer, 2008.

[78] S. Garfinkel. Bill of Rights. http://www.technologyreview.com, October 2002.

[79] G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. Proc. ESAS'04, 2004.

[80] G. Gaubatz, J.-P. Kaps, and B. Sunar. Scaling ECC hardware to a minimum. ECRYPT Workshop - Cryptographic Advances in Secure Hardware - CRASH'05 (invited talk), 2005.

[81] H. Gilbert, M. Robshaw, and H. Sibert. An active attack against HB^+ - A provably secure lightweight authentication protocol. Manuscript, 2005.

[82] B. Glover and H. Bhatt. *RFID Essentials*. O'Reilly, 2006.

[83] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal reencryption for mixnets. In *CT-RSA'04*, volume 2964 of *LNCS*, pages 163–178. Springer-Verlag, 2004.

[84] T. Good and M. Benaissa. Hardware results for selected stream cipher candidates. http://www.ecrypt.eu.org/stream/.

[85] Boycott Guillette. http://www.boycottguillette.com/, 2006.

[86] M. Guillory. Analysis: Counterfeit tags. http://www.aimglobal.org, 2005.

[87] O. Gunther and S. Spiekermann. RFID and the perception of control: the consumer's view. *Commun. ACM*, 48(9):73–76, 2005.

[88] M. Halváč and T. Rosa. A note on the relay attacks on e-passports: The case of czech e-passports. Cryptology ePrint Archive, Report 2007/244, 2007.

[89] G. Hancke. Practical attacks on proximity identification systems (Short paper). In *Proc. of SP'06*. IEEE Computer Society, 2000.

[90] M. Hell, T. Johansson, and W. Meier. Grain - a stream cipher for constrained enviroments. Hand. of the Ecrypt Workshop on RFID and Lightweight Crypto, 2005.

[91] M. Hell, T. Johansson, and W. Meier. Grain: a stream cipher for constrained environments. http://www.ecrypt.eu.org/stream/, 2005.

[92] M. Hell, T. Johansson, and W. Meier. A stream cipher proposal: Grain-128. http://www.ecrypt.eu.org/stream/, 2006.

[93] D. Henrici and P. Müller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proc. of PERSEC'04*, pages 149–153. IEEE Computer Society, 2004.

[94] J. C. Hernandez-Castro, J. M. Estevez-Tapiador, A. Ribagorda-Garnacho, and B Ramos-Alvarez. Wheedham: An automatically designed block cipher by means of genetic programming. In *Proc. of CEC'06*, pages 192–199, 2006.

[95] T. Heydt-Benjamin, H.-J. Chae, B. Defend, and K. Fu. Privacy for public transportation. In *Proc. of PET'06*, 2006.

[96] T. S Heydt-Benjamin, D. V. Bailey, K. Fu, A. Juels, and T. Ohare. Vulnerabilities in first-generation RFID-enabled credit cards. In *Proc. of FC'07*, volume 4886, pages 2–14, 2007.

[97] T. Hjorth. Supporting privacy in RFID systems (master thesis). Master thesis, Technical University of Denmark, 2004.

[98] J. H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R. Wichers Schreur. Crossing borders: Security and privacy issues of the european e-Passport. In *Proc. of IWSEC'06*, volume 4266 of *LNCS*, pages 152–167. Springer-Verlag, 2006.

[99] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and J. Chee. HIGHT: A new block cipher suitable for low-resource device. In *Proc. of CHES'06*, volume 4249 of *LNCS*, pages 46–59. Springer-Verlag, 2006.

[100] K. Hyun Kim, E. Young Choi, S. Mi Lee, and D. Hoon Lee. Secure EPCglobal Class-1 Gen-2 RFID system against security and privacy problems. In *Proc. of OTM-IS'06*, volume 4277 of *LNCS*, pages 362–371. Springer-Verlag, 2006.

[101] Texas Instruments. TI-RFid. http://www.ti.com/rfid/shtml/rfid.shtml, 2006.

[102] ITU page on definitions of ISM bands. http://www.itu.int/ITU-R/terrestrial/faq/index.html, September 2005.

[103] ISO – Internacional Organization for Standardization. http://www.iso.org/, 2005.

[104] ISO/IEC 18000-6:2004/Amd:2006. http://www.iso.org/, 2006.

[105] B. Jamali, P. H. Cole, and D. Engels. *Networked RFID Systems and Lightweight Cryptography*, chapter RFID Tag Vulnerabilities in RFID Systems, pages 147–155. Springer, 2007.

[106] RFID Journal. Behind the benetton brouhaha. http://www.rfidjournal.com, April 14, 2003.

[107] A. Juels. Minimalist cryptography for low-cost RFID tags. In *Proc. of SCN'04*, volume 3352 of *LNCS*, pages 149–164. Springer-Verlag, 2004.

[108] A Juels. RFID security and privacy: A research survey. Manuscript, 2005.

[109] A. Juels. Strengthening epc tags against cloning. Manuscript, March 2005.

[110] A. Juels and J. Brainard. Soft blocking: Flexible blocker tags on the cheap. In *WPES'04*, pages 1–7. ACM Press, October 2004.

[111] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. In *Proc. of SecureComm'05*. IEEE Computer Society, 2005.

[112] A. Juels and R. Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *Proc. of FC'03*, volume 2742 of *LNCS*, pages 103–121. Springer-Verlag, 2003.

[113] A. Juels, R. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In *ACM CCS'03*, pages 103–111. ACM Press, 2003.

[114] A. Juels and S. Weis. Authenticating pervasive devices with human protocols. In *Proc. of CRYPTO'05*, volume 3126 of *LNCS*, pages 293–308. Springer-Verlag, 2005.

[115] Ari Juels. "yoking-proofs" for RFID tags. In *Proc. of PerSec'04*, pages 138–143. IEEE Computer Society, 2004.

[116] G. Karjoth and P. A Moskowitz. Disabling RFID tags with visible confirmation: clipped tags are silenced. In *Proc. of WPES'05*. ACM Press, 2005.

[117] N. Karten and H. Plötz. Mifare little security, despite obscurity. http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html.

[118] S. Karthikeyan and M. Nesterenko. RFID security without extensive cryptography. In *Proc. of SASN'05*, 2005.

[119] A. Kerckhoffs. La cryptographie militaire. *Journal des sciencies*, 9:161–191, 1983.

[120] Z. Kfir and A. Wool. Picking virtual pockects using relay attacks on contacless smartcard systems. In *Proc. of SecureComm'05*. IEEE Computer Society, 2005.

[121] S. Kinoshita, F. Hoshino, T. Komuro, A. Fujimura, and M. Ohkubo. Low-cost RFID privacy protection scheme. In *IPS Journal 45:(8)*, pages 2007–2021, 2003.

[122] S. Kinoshita, M. Ohkubo, F. Hoshino, G. Morohashi, O. Shionoiri, and A. Kanai. Privacy enhanced active RFID tag. In *Proc. of ECHISE'05*, 2005.

[123] A Klimov and A. Shamir. Cryptographic applications of T-functions. In *Proc. of SAC'03*, volume 3006 of *LNCS*, pages 248–261. Springer-Verlag, 2003.

[124] D. M. Konidala and K. Kim. Rfid tag-reader mutual authentication scheme utilizing tag's access password. Auto-ID Labs White Paper WP-HARDWARE-033, 2007.

[125] D. M. Konidala, Z. Kim, and K. Kim. A simple and cost-effective rfid tag-reader mutual authentication scheme. Hand. of RFIDSec'07, 2007.

[126] E. Kosta, M. Meints, M. Hensen, and M. Gasson. An analysis of security and privacy issues relating to RFID enabled epassports. In *Proc. of Sec'07*, volume 232, pages 467–472. Springer-Verlag, 2007.

[127] J. R. Koza. Evolving a computer program to generate random number using the genetic programming paradigm. In *Proc. of the 4th Int. Conference on Genetic Algorithms*, pages 37–44, 1991.

[128] S. Kumar and C. Paar. Are standards compliant elliptic curve cryptosystems feasible on rfid? Hand. of RFIDSec'06, July 2006.

[129] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler. Breaking ciphers with COPACOBANA: A cost optimized parallel code breaker. In *Proc. of CHES'06*, volume 4249 of *LNCS*, pages 101–118. Springer-Verlag, 2006.

[130] RSA Laboratories. Faq on RFID and RFID privacy, 2006.

[131] A. Laurie. RFIDIOt project. http://www.rfidiot.org, 2007.

[132] G. Leander, C. Paar, A. Poschmann, and K. Schramm. New lightweight DES variants. In *Proc. of FSE'07*, volume 4593 of *LNCS*, pages 196–220. Springer-Verlag, 2007.

[133] C. Lee, Houdeau D., and Bergmann R. Evolution of the e-passport. http://www.homelandsecurityasia.com, 2007.

[134] H. Lee, E. Young Choi, S. M. Lee, and D. Dong Lee. Trapdoor-based mutual authentication scheme without cryptographic primitives in RFID tags. In *Proc. of SecPerU'07*. IEEE Computer Society, 2007.

[135] S. Lee, T. Asano, and K. Kim. RFID mutual authentication scheme based on synchronized secret information. In *Symposium on Cryptography and Information Security*, 2006.

[136] S. M. Lee, Y. J. Hwang, D. H. Lee, and J. I. L. Lim. Efficient authentication for low-cost RFID systems. In *Proc. of ICCSA'05*, volume 3480 of *LNCS*, pages 619–627. Springer-Verlag, 2005.

[137] S. Lemieux and A. Tang. Clone resistant mutual authentication for low-cost rfid technology. Cryptology ePrint Archive, Report 2007/170, 2007. `http://eprint.iacr.org/`.

[138] Stépahne Lemieux and Adrian Tang. Clone resistant mutual authentication for low-cost RFID technology. Cryptology ePrint Archive, Report 2007/170, 2007.

[139] X. Leng, K. Mayes, and K. Markantonakis. HB-MP+ protocol: An improvement on the HB-MP protocol. *IEEE International Conference on RFID*, pages 118–124, 2008.

[140] T. Li and R. Deng. Vulnerability analysis of EMAP − an efficient RFID mutual authentication protocol. Proc. of AReS'07, 2007.

[141] T. Li and G. Wang. Security analysis of two ultra-lightweight RFID authentication protocols. In *Proc. of IFIP-SEC'07*, 2007.

[142] T. Li and G. Wang. SLMAP − A secure ultra-lightweight rfid mutual authentication protocol. Proc. of Chinacrypt'07, 2007.

[143] T. L. Lim and T. Li. Addressing the weakness in a lightweight RFID tag-reader mutual authentication scheme. Proc. of GLOBECOM'07, 2007.

[144] T. Lohmmann, M. Schneider, and C. Ruland. Analysis of power constraints for cryptographic algorithms in mid-cost RFID tags. In *Proc. of CARDIS'06*, volume 3928 of *LNCS*, pages 278–288. Springer-Verlag, 2006.

[145] F. Mace, F.-X. Standaert, and J.-J. Quisquater. ASIC implementations of the block cipher SEA for constrained applications. Hand. of RFIDSec'07, 2007.

[146] F. Mace, F.-X. Standaert, and J.-J. Quisquater. FPGA implementation(s) of a scalable encryption algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 16:212–216, 2008.

[147] S. Malladi, S. Alves-Foss, and R. Heckendorn. On preventing replay attacks on security protocols. In *Proc. of SM'02*, CSREA Press, pages 77–83, 2003.

[148] G. Marsaglia. *The Marsaglia Random Number CDROM Including the DIEHARD Battery of Tests of Randomness*. http://stat.fsu.edu/pub/diehard, 1996.

[149] G. Marsaglia and and W.W. Tsang. Some difficult-to-pass tests of randomness. *Journal of Statistical Software*, 7(3), 2002.

[150] M. Matsui. Linear cryptanalysis method for DES cipher. In *Proc. of EUROCRYPT'93*, volume 1994, pages 386–397. Springer-Verlag New York, Inc.

[151] M. Matsumoto et al. Mersenne twister: A 623-dimensionally equidistributed uniform PRNG. *ACM Trans. on Modeling and Comp. Sim.*, 8(1):3–30, 1998.

[152] mCloak for RFID tags. http://www.mobilecloak.com/rfidtag/rfid.tag.html, September 2005.

[153] MiniMe and Mahajivana. RFID-Zapper project. http://www.events.ccc.de/congress/2005/static/r/f/i/RFID-Zapper(EN)_77f3.html, 2006.

[154] D. Molnar, A. Soppera, and D. Wagner. A scalable, delegatable, pseudonym protocol enabling ownership transfer of RFID tags. Hand. of Ecrypt Workshop on RFID and Lightweight Crypto, July 2005.

[155] D. Molnar and D. Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Proc. of ACM CCS'04*, pages 210–219. ACM Press, 2004.

[156] J. Munilla and A. Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51(9):2262–2267, June 2007.

[157] J. Munilla and A. Peinado. Attacks on singelee and preneel's protocol. Cryptology ePrint Archive, Report 2008/283, 2008. `http://eprint.iacr.org/`.

[158] D. Nguyen Duc, J. Park, H. Lee, and Kwangjo K. Enhancing security of epcglobal Gen-2 RFID tag against traceability and cloning. In *Proc. of Symposium on Cryptography and Information Security*, 2006.

[159] K. Nohl, D. Evans, Starbug, and H. Plotz. Reverse-Engineering a Cryptographic RFID Tag. In *USENIX Security Symposium*, 2008.

[160] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to "privacy-friendly" tags. In *Proc. of RFID Privacy Workshop*, 2003.

[161] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In *Proc. of IS'06*, volume 4277 of *LNCS*, pages 352–361. Springer-Verlag, 2006.

[162] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. Hand. of RFIDSec'06, 2006.

[163] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In *Proc. of UIC'06*, volume 4159 of *LNCS*, pages 912–923. Springer-Verlag, 2006.

[164] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. RFID systems: A survey on security threats and proposed solutions. In *Proc. of PWC'06*, volume 4217 of *LNCS*, pages 159–170. Springer-Verlag, 2006.

[165] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. An efficient authentication protocol for rfid systems resistant to active attacks. In *EUC Workshops: SecUbiq Workshop*, volume 4809 of *LNCS*, pages 781–794. Springer-Verlag, 2007.

[166] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. LAMED - a PRNG for EPC class-1 generation-2 RFID

specification. *Computer Standars & Interfaces, Elsevier Science Publishers, doi: 10.1016/j.csi.2007.11.013*, 2007.

[167] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Solving the simultaneous scanning problem anonymously: Clumping proofs. In *Proc. of SecPerU'07*. IEEE Computer Society, 2007.

[168] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard. *Computer Standars & Interfaces, Elsevier Science Publishers, doi:10.1016/j.csi.2008.05.012*, 2008.

[169] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems*, chapter RFID Specification Revisited, pages 127–156. Auerbach Publications, Taylor & Francis, 2008.

[170] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. *Security in RFID and Sensor Networks*, chapter Lightweight Cryptography for Low-Cost RFID Tags. Auerbach Publications, Taylor & Francis, 2008 (In Press).

[171] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. *Security in RFID and Sensor Networks*, chapter Attacking RFID Systems. Auerbach Publications, Taylor & Francis, 2008 (In Press).

[172] P. Peris-Lopez, T. Li, T.-L. Lim, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Vulnerability analysis of a mutual authentication scheme under the epc class-1 generation-2 standard. In *Hand. of RFIDSec'08*, 2008.

[173] E. Pinhas Barkan. Cryptanalysis of ciphers and protocols. Phd thesis, The Technion: Israel Institute of Technology, 2006.

[174] S. Piramuthu. On existence proofs for multiple RFID tags. In *Proc. of SecPerU'06*. IEEE Computer Society, 2006.

[175] Selwyn Piramuthu. HB and related lightweight authentication protocols for secure RFID tag/reader authentication. In *Proc. of CollECTeR'06*, 2006.

[176] A. Poschmann, G. Leander, K. Schramm, and C. Paar. New lightweight crypto algorithms for rfid. *Proc. of IEEE International Symposium on Circuits and Systems*, pages 1843–1846, 2007.

[177] N. Pramstaller, S. Mangard, S. Dominikus, and J. Wolkerstorfer. Efficient AES implementations on ASICs and FPGAs. In *Proc. Fourth Workshop on the Advanced Encryption Standard "AES - State of the Crypto Analysis*, volume 3373 of *LNCS*, pages 98–112. Springer-Verlag, 2004.

[178] P. Prince. United states sets date for e-passports. RFID Journal, 2005.

[179] D. Ranasinghe, D. Engels, and P. Cole. Low-cost RFID systems: Confronting security and privacy. In *Auto-ID Labs Research Workshop*, 2004.

[180] D. Ranasinghe, D. Engels, and P. Cole. Security and privacy: Modest proposals for low-cost RFID systems. In *Proc. of Auto-ID Labs Research Workshop*, 2004.

[181] D. C. Ranasinghe. *Networked RFID Systems and Lightweight Cryptography*, chapter Lightweight Cryptography for Low Cost RFID, pages 311–346. Springer-Verlag, 2007.

[182] D. C. Ranasinghe and P. H. Cole. Confronting security and privacy threats in modern rfid systems. *Proc. of ACSSC '06*, pages 2058–2064, 2006.

[183] K. Rhee, J. Kwak, S. Kim, and D. Won. Challenge-response based RFID authentication protocol for distributed database environment. In *Proc. of SPC'05*, volume 3450 of *LNCS*, pages 70–84. Springer-Verlag, 2005.

[184] M. Rieback, C. Bruno, and A. Tanenbaum. Is your car infected with a computer virus? In *Proc. of PerCom'06*. IEEE Computer Society, 2006.

[185] M. Rieback, B. Crispo, and A. Tanenbaum. Uniting legislation with RFID privacy-enhancing technologies. In *Security and Protection of Information*, 2005.

[186] M. Rieback, G. Gaydadjiev, B. Crispo, R. Hofman, and A. Tanenbaum. A platform for rfid security and privacy administration. In *Proc. of LISA'06*, 2006.

[187] C. M. Roberts. Radio frequency identification (RFID). *Computers and Security*, 25(1):18–26, 2006.

[188] M. Rogawski. Hardware evaluation of eSTREAM candidates: Grain, Lex, Mickey128, Salsa20 and Trivium. http://www.ecrypt.eu.org/stream/.

[189] What's in California's proposed Rfid Bill? http://www.rfidproductsnew.com, 2006.

[190] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST special publication 800-22, http://csrc.nist.gov/rng/, 2001.

[191] J. Saito and Sakurai Kouichi. Grouping proof for RFID tags. In *Conference on Advanced Information Networking and Applications – AINA*, volume 2, pages 621–624. IEEE Computer Society, 2005.

[192] J. Saito, J.-C. Ryou, and K. Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. In *Proc. of EUC'04*, volume 3207 of *LNCS*, pages 879–890. Springer-Verlag, 2004.

[193] S.E. Sarma, S.A. Weis, and D.W. Engels. RFID Systems and Security and Privacy Implications. In *Proc. of CHES'02*, volume 2523 of *LNCS*, pages 454–470. Springer-Verlag, 2002.

[194] A. Satoh, S. Morioka, K. Takano, and S. Munetoh. A compact Rijndael hardware architecture with s-box optimization. In *Proc. of ASIACRYPT'01*, volume 2248 of *LNCS*, pages 239–254. Springer-Verlag, 2001.

[195] W. Sean and L. Thomas. Automatic identification and data collection technologies in the transportation industry: BarCode and RFID. Technical report, 2001.

[196] W.G. Shieh and J.M. Wang. Efficient remote authentication and key agreement. *Computers and Security*, 25(1):72–77, 2006.

[197] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata. The 128-bit blockcipher CLEFIA (extended abstract). In *Proc. of FSE'07*, volume 4593 of *LNCS*, pages 181–195. Springer-Verlag, 2007.

[198] SoSGroup, ICIS, and Radbound University. JMRTD project. http://www.jmrtd.sourceforge.net/, 2007.

[199] S. Spiekermann and H. Ziekow. RFID: a 7-point plan to ensure privacy. In *Proc. of ECIS'05*, 2005.

[200] R. M. Stamp, M. Low. *Breaking Ciphers in the Real World*. John Wiley & Sons, 2007.

[201] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater. SEA: A scalable encryption algorithm for small embedded applications. In *Proc. of CARDIS'06*, pages 226–236. Springer-Verlag, 2006.

[202] F.-X. Standaert, G. Piret, and J.-J. Quisquater. Cryptanalysis of block ciphers: A survey. Technical report.

[203] M. Stigge, H. Plötz, W. Müller, and J.-P. Redlich. Reversing CRC: Theory and practice. Technical Report SAR-PR-2006-05, Humboldt-Universitat Berlin, 2006.

[204] H.-M. Sun, W.-C. Ting, and K.-H. Wang. On the security of chien's ultralightweight RFID authentication protocol. Cryptology ePrint Archive, Report 2008/083, 2008.

[205] C. Suresh, Charanjit J., J.R. Rao, and P. Rohatgi. A cautionary note regarding evaluation of AES candidates on smart-cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, 1999.

[206] C. Swedberg. Broadcom introduces secure RFID chip. RFID Journal. http://www.rfidjournal.com, 2006.

[207] C. Swenson. *Modern Cryptanalysis: techniques for advanced code breaking*. John Wiley, 2008.

[208] Symbol. RFID: A revolution in asset management. http://www.symbol.com/products/rfid-readers, 2006.

[209] P. Syverson. A taxonomy of replay attacks. In *Proc. of CSF'94*, pages 187–191, 1994.

[210] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall International, Inc, 3rd edition, 1996.

[211] Allien Technology. EPCglobal Class-1 Gen-2 RFID Specification. Whitepaper, 2006.

[212] Boycott Tesco. http://www.boycotttesco.com/, 2003.

[213] F. Thornton, B. Haines, A. Das, H. Bhargava, A. Campbell, and J. Kleinschmidt. *RFID Security*. Syngress Publishing, 2006.

[214] G. Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *Proc. of PERCOM'06*. IEEE Computer Society, 2006.

[215] I. Vajda and L. Buttyán. Lightweight authentication protocols for low-cost RFID tags. In *Proc. of UBICOMP'03*, 2003.

[216] J. Walker. *Randomness Battery*. http://www.fourmilab.ch/random/, 1998.

[217] M. Wang. Differential cryptanalysis of present. Cryptology ePrint Archive, Report 2007/408, 2007. `http://eprint.iacr.org/`.

[218] X. Wang, D. Feng, X. Lai, and H. Yu. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199, 2004.

[219] X. Wang, Y. Lisa Yin, and H. Yu. Finding collisions in the full SHA-1. In *Proc. of CRYPTO'05*, pages 17–36, 2005.

[220] E. Wasserman. Purdue pharma to run pedigree pilot. RFID Journal, 2005.

[221] S. H. Weingart. Physical security devices for computer subsystems: A survey of attacks and defenses. In *Proc. of CHES'00*, volume 1965, pages 302–317. LNCS, 2000.

[222] S. Weis. Security parallels between people and pervasive devices. In *Proc. of PERSEC'05*, pages 105–109. IEEE Computer Society, 2005.

[223] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Proc. of SPC'03*, volume 2802 of *LNCS*, pages 454–469. Springer-Verlag, 2003.

[224] S.A. Weis, S.E. Sarma, R.L. Rivest, and D.W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Proc. of Security in Pervasive Comp.*, volume 2802 of *LNCS*, pages 201–212. Springer-Verlag, 2004.

[225] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, September 1991.

[226] H. Welte. OpenMRTD project. http://www.openmrtd.org, 2007.

[227] B. Westerbaan. Reversing CRC. http://blog.w-nz.com/archives/2005/07/15/reversing-crc/, 2005.

[228] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim. Mutual authentication protocol for low-cost RFID. Hand. of Ecrypt Workshop on RFID and Lightweight Crypto, 2005.

[229] K. Yksel, J. P. Kaps, and B. Sunar. Universal hash functions for emerging ultra-low-power networks. In *Proc. of CNDS'04*, 2004.

[230] E. Zeisel and R. Sabella. *RFID+ Exam Cram*. Que, 2006.