Design and Analysis of Security Schemes for Low-cost RFID Systems

by

Qi Chai

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the remarkable progress in microelectronics and low-power semiconductor technologies, Radio Frequency IDentification technology (RFID) has moved from obscurity into mainstream applications, which essentially provides an indispensable foundation to realize ubiquitous computing and machine perception. However, the catching and exclusive characteristics of RFID systems introduce growing security and privacy concerns. To address these issues are particularly challenging for low-cost RFID systems, where tags are extremely constrained in resources, power and cost. The primary reasons are: (1) the security requirements of low-cost RFID systems are even more rigorous due to large operation range and mass deployment; and (2) the passive tags' modest capabilities and the necessity to keep their prices low present a novel problem that goes beyond the well-studied problems of traditional cryptography. This thesis presents our research results on the design and the analysis of security schemes for low-cost RFID systems.

Motivated by the recent attention on exploiting physical layer resources in the design of security schemes, we investigate how to solve the eavesdropping, modification and one particular type of relay attacks toward the tag-to-reader communication in passive RFID systems without requiring lightweight ciphers. To this end, we propose a novel physical layer scheme, called *Backscatter modulation- and Uncoordinated frequency hopping-assisted Physical Layer Enhancement* (BUPLE). The idea behind it is to use the amplitude of the carrier to transmit messages as normal, while to utilize its periodically varied frequency to hide the transmission from the eavesdropper/relayer and to exploit a random sequence modulated to the carrier's phase to defeat malicious modifications. We further improve its eavesdropping resistance through the coding in the physical layer, since BUPLE ensures that the tag-to-eavesdropper channel is strictly noisier than the tag-to-reader channel. Three practical *Wiretap Channel Codes* (WCCs) for passive tags are then proposed: two of them are constructed from linear error correcting codes, and the other one is constructed from a resilient vector Boolean function. The security and usability of BUPLE in conjunction with WCCs are further confirmed by our proof-of-concept implementation and testing.

Eavesdropping the communication between a legitimate reader and a victim tag to obtain raw data is a basic tool for the adversary. However, given the fundamentality of eavesdropping attacks, there are limited prior work investigating its intension and extension for passive RFID systems. To this end, we firstly identified a brand-new attack, working at physical layer, against backscattered RFID communications, called *unidirectional active eavesdropping*, which defeats the customary impression that eavesdropping is a "passive" attack. To launch this attack, the adversary transmits an un-modulated carrier (called *blank carrier*) at a certain frequency $f_{\mathcal{E}}$ while a valid reader and a tag interacts at another frequency channel $f, f \neq f_{\mathcal{E}}$. Once the tag modulates the amplitude of reader's signal, it causes fluctuations on the blank carrier as well. By carefully examining the amplitude of the backscattered versions of the blank carrier and the reader's carrier, the adversary could intercept the ongoing reader-tag communication with either significantly lower bit error rate or from a significantly greater distance away. Our concept is demonstrated and empirically analyzed towards a popular low-cost RFID system, i.e., EPC Gen2. Although active eavesdropping in general is not trivial to be prohibited, for a particular type of active eavesdropper, namely a *greedy proactive eavesdropper*, we propose a simple countermeasure without introducing extra cost to current RFID systems.

The needs of cryptographic primitives on constraint devices keep increasing with the growing pervasiveness of these devices. One recent design of the lightweight block cipher is Hummingbird-2. We study its cryptographic strength under a novel technique we developed, called *Differential Sequence Attack* (DSA), and present the first cryptanalytic result on this cipher. In particular, our full attack can be divided into two phases: *preparation phase* and *key recovery phase*. During the key recovery phase, we exploit the fact that the differential sequence for the last round of Hummingbird-2 can be retrieved by querying the full cipher, due to which, the search space of the secret key can be significantly reduced. Thus, by attacking the encryption (decryption resp.) of Hummingbird-2, our algorithm recovers 36-bit (another 28-bit resp.) out of 128-bit key with 2^{68} (2^{60} resp.) time complexity if particular differential conditions of the internal states and of the keys at one round can be imposed. Additionally, the rest 64-bit of the key can be exhaustively searched and the overall time complexity is dominated by 2^{68}. During the preparation phase, by investing 2^{81} effort in time, the adversary is able to create the differential conditions required in the key recovery phase with at least 0.5 probability.

As an additional effort, we examine the cryptanalytic strength of another lightweight candidate known as A2U2, which is the most lightweight cryptographic primitive proposed so far for low-cost tags. Our chosen-plaintext-attack fully breaks this cipher by recovering its secret key with only querying the encryption twice on the victim tag and solving 32 sparse systems of linear equations (where each system has 56 unknowns and around 28 unknowns can be directly obtained without computation) in the worst case, which takes around 0.16 second on a Thinkpad T410 laptop.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Professor Guang Gong for her excellent supervision and inspiring support during the past years. Her extensive expertise and rich experience combining with her passion and enthusiasm for research fill me with courage, lead and motivate me to explore an unforeseen world in the science, which is so fascinating. The critical thinking and positive, optimistic attitude towards life and work offered by her will be rewarding for me forever.

Besides, I would like to express my appreciation to Professor Guan Yong at the Iowa State University for serving as my external examiner and giving me many valuable suggestions. My deepest gratitude also goes to my thesis committee members, Professor Mark Aagaard, Professor Mohamed Oussama Damen and Professor Doug Stinson, for their instrumental comments, their time and energy on this thesis. I am truly honored and blessed to have been guided by these individuals of highest professional and personal character. This dissertation would not have been possible without the assistance of them.

I am particularly grateful for the encouragement and help I received from Dr. Daniel Engels from Revere Security Corporation, especially for all the inspiring discussions on the active eavesdropping attack in passive RFID systems as well as lightweight cryptographic primitives and their applications in securing RFID systems.

I am deeply obliged to many of my friends and colleagues in the University of Waterloo for their support and help during and beyond my PhD program: I thank Dr. Xinxin Fan, Dr. Honggang Hu, Dr. Zhijun Li, Dr. Anuchart Tassanaviboon, and Bo Zhu for their friendship and exchanging ideas. I also thank Yiyuan Luo, Kalikinkar Mandal, Fei Huo, Yang Yang, Teng Wu and Shasha Zhu for their supportive suggestions and for having much fun with them. I want to thank all the wonderful members of the Communication Security (ComSec) Lab of the University of Waterloo for always contributing to create a pleasant and stimulating group atmosphere.

Last but not least, I am deeply indebted to my family for their unflagging care, trust and support. My father's commitment to lifelong learning, growth and hard work has nurtured and inspired me throughout my life. My mother's endless love and faith in me has always been a beacon of confidence for me. Most of all, my lovely wife deserves special acknowledgement. Her perseverance, devotion and sacrifices enabled me and motivated me to focus on the research I am interested. I am eternally grateful and wonderfully blessed to have her as my wife.

Dedication

This dissertation is dedicated to my wife, Ms. Weibei Li, and my parents, Mr. Ke Chai and Ms. Chunrong Liang. I love you forever.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the remarkable progress in microelectronics and low-power semiconductor technologies, Radio Frequency IDentification technology (RFID) has moved from obscurity into mainstream applications. Unlike earlier bar-code technology, RFID technology enables automatic identification from a distance without requiring a line of sight and can incorporate additional context data such as manufacturer, product type, and even environmental factors. In a broad sense, RFID technology is believed to be an indispensable foundation to realize ubiquitous computing and machine perception as long as RFID principle is more thoroughly understood, cheap components are more available, and RFID security and privacy are guaranteed. Unfortunately, security and privacy risks associated are not easy to address for low-cost RFID systems, e.g., threats such as tracking, counterfeiting and denial of service are instant doom for some people.

This thesis is concerned with the design and analysis of security and privacy solutions for low-cost RFID systems, ranging from developing and implementing schemes in light of the physical layer of RFID communication to analyzing the cryptographic primitives proposed for the constrained devices. This chapter starts with an introduction to basic principles, applications and active standards of passive RFID systems in Section 1.1. In Section 1.2, we highlight the security and privacy threats towards passive RFID systems, followed by presenting the security requirements and goals. Our research motivation and methodology are given in Section 1.3. Finally, the outline of this thesis and a summary of our research contributions are displayed in Section 1.4.

1.1 RFID – Principle, Applications and Standards

1.1.1 The Start of Something Big

Basic Gradients: An RFID system usually consists of considerable *transponders* or *tags*, couples of *readers* or *interrogators*, and one *backend database*. A general view of an RFID system is given in Figure 1.1.

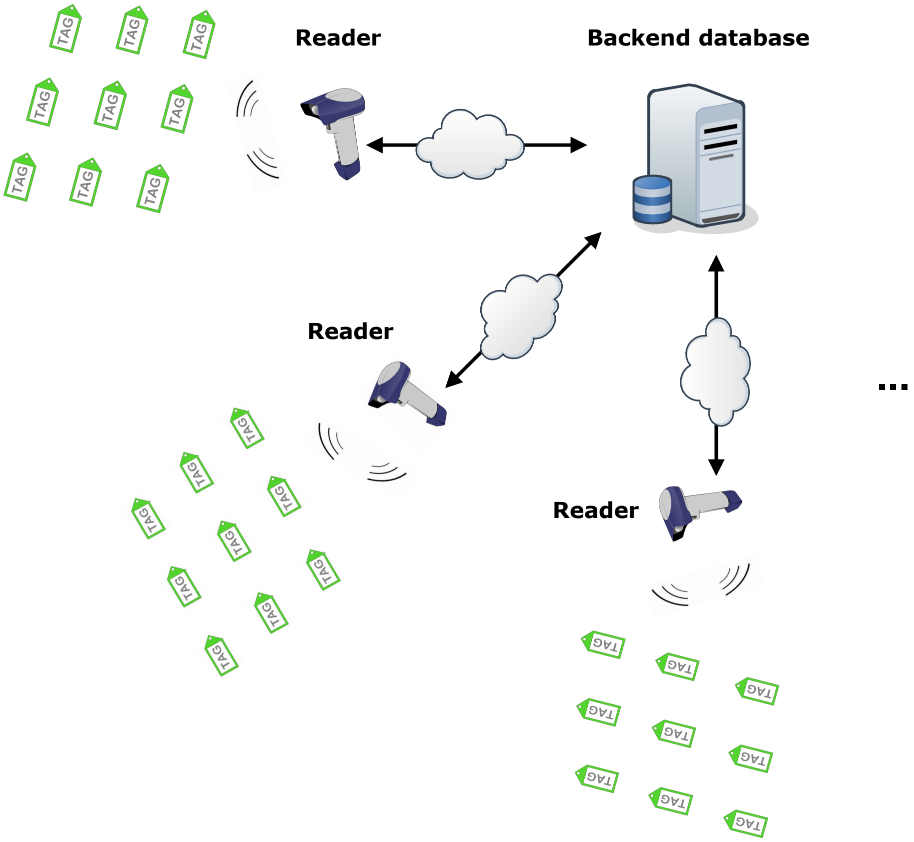

Figure 1.1: A General View of an RFID System

- *Transponders* or *Tags*: Each tag contains an IC chip with certain computation and storage capabilities, and an antenna coil for communication. The computation and storage capabilities entirely depend on the type of the tag, i.e., active, passive and semi-passive.

  - *A passive tag* does not have an internal source of power. Instead, it harvests power from the electromagnetic field created by the readers nearby. Tags fall into this category are either weak in computation but can be operated in a large range, or, capable of performing intensive computation just in near proximity, because the number of operations can be performed rests on the amount of power available, which diminishes at the second order of the distance between the tag and the reader.
  - *An active tag* is more a conventional wireless communication device in terms of its processing and storage capabilities, e.g., cell phone or wireless sensor node, which possesses a power source, e.g., battery, that is used to support on-tag computation as well as signal transmitting and receiving.
  - *A semi-passive tag* is a hybrid of the above two, which uses battery to run the chip's circuitry but communicates by harvesting power from the reader signal.

- *Readers* or *Interrogators*: An RFID reader is a transmitter and receiver that work together to communicate with the tag. There are two types of readers, namely *portable readers* and *regular readers*. A portable reader, e.g., handheld PDA, smart phone, may still have some limitations in its processing and storage capabilities, while a regular reader could be as powerful as today's computer. Both types of the readers connect to backend databases through a wireless/wired link, which is generally secured by SSL/TLS, and is usually not considered in the context of RFID security and privacy.

- *Backend database*: To reduce the cost per tag and to manage data in a more reliable and secure way, the information carried by the tag is an index to a backend database, e.g., pointers, IDs, cursors, etc.. The information stored for each tag can be further classified as: (1) public information such as tags' IDs; (2) private information such as tags' secret keys; and (3) context information such as physical properties of objects/items associated with tags, which can be either public or private, depending on the concrete applications. It is worth to mention that, albeit the backend database is logically unique in one RFID system, it could be geometrically distributed to provide extra reliability.

At a higher level, an *RFID infrastructure* considers not only ownership transferring of physical objects, but also enriching, e.g., by filtering and aggregating, raw RFID-data, marketing the data and exchanging with other vendors. In this thesis, we focus more on the interaction between readers and tags regardless of the upper layer structures.

Reader-Tag Interaction:

One of the main purposes of the reader-tag interaction is for the reader to "wake up" a passive tag by providing it with enough impinging energy. There are two various methods for transferring energy wirelessly to a passive tag:

- *Radiation-based energy transfer*: The reader transmits a continuous RF signal. When a tag appears in the area, it receives the signal and converts it to direct current by a rectifier circuit and stores the energy by charging a capacitor.
- *Induction-based energy transfer*: By exploiting inductive coupling, the reader's antenna uses current to generate a magnetic field, which produces a current flow in the tag's antenna that powers the tag's IC chip.

After power harvesting, the tag starts to parse the command issued by the reader usually by amplitude-demodulating the reader's carrier wave and decoding the baseband signal. To respond, the tag encodes the target message to a longer binary sequence and switches the impedance load of its antenna according to the data stream, causing modulation of either: (1) the amplitude of the signal which is later reflected (in radiation-based energy transfer); or (2) magnetic field joining the reader and tag (in induction-based energy transfer).

The coding scheme used by RFID tags is usually Manchester-like codes, where the encoding of each data bit has at least one transition and occupies the same time, e.g., $1 \mapsto 10$ and $0 \mapsto 01$. One benefit of using the Manchester code or its variants is that if two tags responds approximately at the same time, a collided bit would lost the transition, e.g., $10 + 01 = 11$, which turns to be an illegal codeword. Therefore, this abnormal codeword informs the reader of the occurrence of the collision. In fact, *anti-collision mechanism* is an important research topic for the RFID community and the most popular solution is based on the *time division multiple access*, where a time interval is split into slots, in one of which, the reader interacts with only one tag.

After establishing an error-free physical layer and an error-free data link layer, at the logic layer, the reader interacts with the tag in a question-and-answer manner, e.g., the reader issues a command to a tag, and the tag, according to its knowledge, computes an answer, responds with the answer and puts itself into the next state. Next, if the reader already obtains the information desired, the current session is suspended and the tag resets its state; otherwise, the reader continues to ask and the tag continues to answer and transit its state. Note that a *protocol* is often used to describe this interaction in an abstract way.

1.1.2 RFID Applications

RFID technology is one of the most discussed auto-identification and data capture technologies nowadays, and the range of the applications is broadening rapidly. Some of the applications are shown in Figure 1.2: an RFID-enabled smart poster promoting the concert of a popular singer – fans may get details about the event and book the ticket by reading the poster using their iPhones; a healthcare tag widely used in hospital scenarios helps doctors to locate a patient and access the medical history of the patient in real time; an *Oyster card* is deployed in London for using the public transportation facilities; a reusable smart coffee mug embeds an RFID chip and enables people fill the mugs and walk out of the shop without caring about the paying; a person is able to enter others' social networks by touching their *pokens* – an RFID-enabled social networking card; Wal-Mart is trying to keep better track of its inventory by adding RFID tags to individual items in its stores; *Google Wallet* allows its users to store credit cards, loyalty cards, and gift cards among other things, as well as redeeming sales promotions on their RFID-enabled mobile phone.

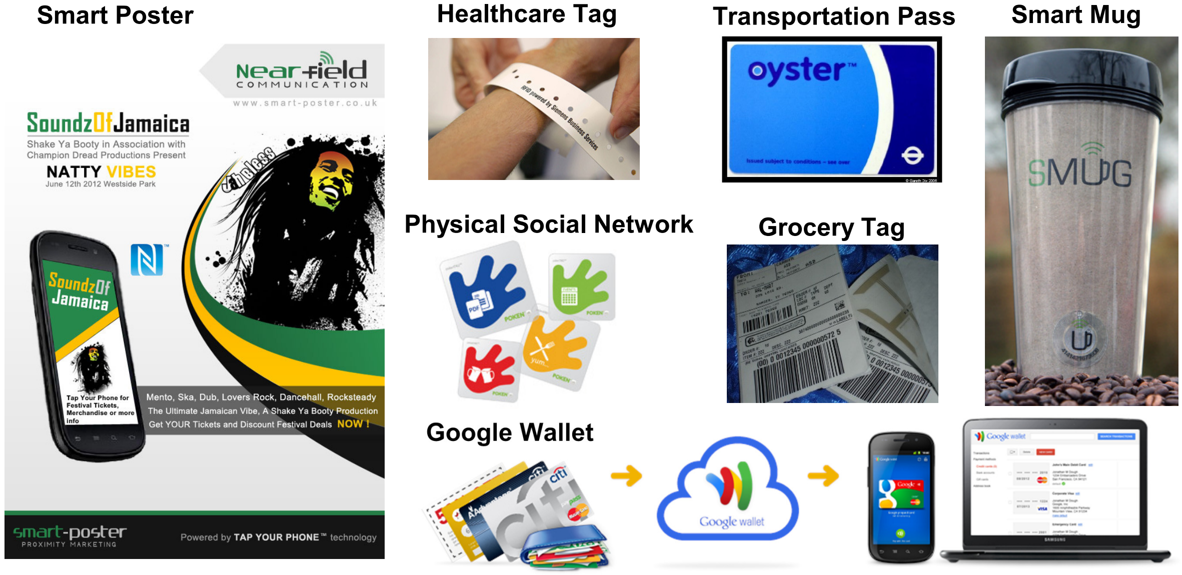
Smart Poster
Healthcare Tag
Transportation Pass
Smart Mug
Near-field COMMUNICATION
www.smart-poster.co.uk
SoundzOfJamaica
Shake Ya Booty in Association with Champion Dread Productions Present
NATTY VIBES
June 12th 2012 Westside Park
Tap Your Phone for Festival Tickets, Merchandise or more info
Mento, Ska, Dub, Lovers Rock, Dancehall, Rocksteady
The Ultimate Jamaican Vibe, A Shake Ya Booty Production
Get YOUR Tickets and Discount Festival Deals NOW !
smart-poster PROXIMITY MARKETING
Powered by TAP YOUR PHONE™ technology
oyster
SMUG
Physical Social Network
Grocery Tag
Google Wallet
Google wallet

Figure 1.2: Novel Applications of RFID Technology

Generally speaking, the applications of RFID technology can roughly fall into the following categories.

- *Enterprise Supply Chain Management and Asset Management*: such as logistics management, inventory control/audit, item tracking/management and retail check-out.

For instance, IBM developed an RFID-based solution for chemical and petroleum manufacturing, which tracks production processes and monitor feedstock, raw materials, finished products, logistics and transportation, as well as fixed assets throughout the supply chain.

- *Contactless Payment*: These applications are especially widely deployed in the transportation systems as a convenient way for payments, e.g., *SpeedPass* from Mobil Oil Corporation, *Electronic Road Pricing systems* in Toronto and *oyster cards* in London as mentioned. High-value payment, in conjunction with RFID-enabled smart phones, began to receive attention. Pioneering applications are *Google Wallet*, *Erply* and *Square* in United States, *SWIFT* in Sweden, *iBOXpay* in China.

- *Access Control*: RFID technologies naturally allow: (1) users to enjoy complete hands free access control; and (2) electronically produced records to be easily audited. For example, *keyless entry* permitting or denying access to premises or automobiles is broadly deployed; *theft control systems*, mostly known as *electronic article surveillance systems*, are widely accepted by the public and are applied to protect assets in libraries and super markets. In a broad sense, as RFID bridges the physical society with the digital world, the concept of intrusion detection is also extended to detect anomalous behavior of physical objects.

- *Identification and Tracking*: By making the digital ID of a tag unique and unalterable, RFID potentially offers solutions for human/animal/device identification/authentication as well as tracking. Given an example, *Agility Healthcare* is used to track mobile medical equipment in Virginia hospitals. Most recent, *smart soccer ball* is proposed to be used in the World Cup, which is a system placing up to 12 interrogators around a stadium that can detect an RFID tag inside a ball and track its exact position in real time to help the references make more accurate decisions.

- *Others*: Some of the innovative applications go beyond the conventional use of RFID by integrating other technologies, e.g., environmental monitoring by employing sensor-enabled tags that can detect rainfall, water level and weather conditions; RFID positioning such as *Photosensing RFID for Location Aware Services* developed by MITSUBISH engineers; and *EpixMix* that tracks states of the skier via the lift ticket that is RFID-equipped, such that the skier would be able to marks off the done runs and be socially connected to nearby skiers or friends on the Facebook and Twitter.

Table 1.1: A Quick Reference of RFID Standards

Name	Frequency	Content	Targeted Application
ISO11784/5	125/134.2kHz	full duplex and half duplex protocols between tags and readers are defined	animal identification
ISO14223	125/134.2kHz	air interface, code and command structure	animal identification
ISO/IEC14443 A/B	13.56MHz	modulation methods, coding schemes and protocol initialization procedures	near field communication (NFC)
ISO/IEC15693	13.56 MHz	physical characteristics, air interface, initialization, protocols and registration	vicinity applications, human identification
ISO/IEC18000-6C or EPC Gen2	860-960 MHz	air interface	product-tracking in the supply chain
ISO/IEC18092	13.56 MHz	communication modes, interface and protocol using inductive coupled devices	a simple extension of the ISO/IEC14443
ISO/IEC18047	All	air interface	test methods for conformance with ISO/IEC18000

1.1.3 State Of The Art RFID Standards

The term, “RFID”, in its broadest sense, can refer not just to next-generation barcodes, but to a compact class of wireless communication/computing devices that are standards-compliant. To clarify the context of this thesis, we summarize in Table 1.1 the state-of-the-art RFID standards promulgated by International Organization for Standardization (ISO). These standards typically describe the physical and data link layers, covering aspects such as the air interface, anti-collision mechanisms and communication protocols, whereas security functions are rarely mentioned.

In the pool of various RFID standards, the most promising two are ISO14443 A/B, mostly known as NFC, and ISO18000-6C, mostly known as EPCglobal UHF Class-1 Gen-2 (EPC Gen2) [78], the former of which is capable of conducting intensive computations in a near field of the reader and is thus widely used in applications such as biometric passports, electronic ticketing, *paypass* credit cards, while the latter of which is capable to be operated through a long distance with lightweight operations as mentioned. We provide an interesting comparison between these two different types of RFID tags in Table 1.2, which discloses the fact that NFC tags are powerful enough such that research outcomes from sensor network security and even computer security can be applied.

Table 1.2: Comparison Between EPC Gen2 Tags and NFC Tags

	EPC Gen2 Tag	ISO14443 A/B Tag
Has energy source	no	no
Reading range	up to $\approx$ 10m	less than $\approx$ 5cm
Frequency	$902 - 928$MHz in North America	13.15MHz worldwide
Chip area	1mm^2	$15 - 20$mm^2
Price	several cents	several dollars
Security functions	none (at least now)	crypto coprocessors to perform 3DES, AES, RSA and ECC

1.2 Risks and Threats in RFID Systems

Like many technologies, while yielding great productivity gains, RFID systems may create new threats to the security and privacy of individuals or organizations. In what follows, we consider a computation-bounded adversary $\mathcal{A}$ attacking an RFID system composing of a reader R and bunch of tags $T_i, i = 1, 2, 3, ...$ As mentioned earlier, the interaction between the readers and the database is usually performed on a TLS/SSL-secured link, thus is not considered here. In what follows, we classify the identified risks and threats into two classes, namely *privacy concerns* and *security concerns*.

Privacy Concerns:

- **Inventorying or Rogue Scanning**: $\mathcal{A}$ wishes to learn the contents of the tag T_i without the owner's knowledge or consent by placing a compliant reader, denoted as $\hat{R}$, at a certain location which identifies RFID-labeled items passing by. $\mathcal{A}$ succeeds if any on-tag content is learnt. This threat becomes extraordinary dangerous when a tag's ID is combined with personal information, e.g., a credit card contains not only the serial number but also the holder's name, date of birth, and so on. The consequence of this attack is privacy invasion.

- **Tracing or Tracking**: Utilizing the linkability between the tag T_i and the bearer, $\mathcal{A}$ correlates data collected by multiple malicious readers placed at different locations to track the bearer if fixed or predictable patterns exist in the protocol used by T_i and the legitimate reader R. For instance, a person carrying an RFID tag effectively broadcasts a fixed serial number to nearby readers provides a striking sign for clandestine stalkers. The consequence of this attack is privacy invasion.

- **Backward/Forward Tracing or Tracking**: Tracing or tracking can be further extended if $\mathcal{A}$ is able to compromise the tag before tracing or tracking: (1) backward tracing – given all the secret credentials of a target tag at time t, $\mathcal{A}$ is able to identify target tag interactions that occurred at time before t; (2) forward tracing – given all the secret credentials of a target tag at time t, $\mathcal{A}$ is able to identify target tag interactions that occurred at time after t.

Security Concerns:

- **Replay Attack**: $\mathcal{A}$ eavesdrops the communication between R and T_i for several sessions and $\mathcal{A}$, with the forged reader $\hat{R}$, interrogates T_i for another several sessions, $\mathcal{A}$ then writes the recorded flow(s) to a fake tag, denoted as $\hat{T_i}$, and utilizes $\hat{T_i}$ to cheat R. $\mathcal{A}$ succeeds if R believes that $\hat{T_i}$ is T_i during the authentication. The consequence of this attack is tag impersonation, meaning that an illegitimate electric device is able to pretend to be the legitimate tag.

- **Tag Counterfeiting or Cloning**: $\mathcal{A}$ eavesdrops the communication between R and T_i for several sessions and $\mathcal{A}$, with $\hat{R}$, interrogates T_i for another several sessions. $\mathcal{A}$ then tries to recover the tags' secrets by breaking the underline cryptographic functions or protocols. At last, $\mathcal{A}$ creates a fake tag $\hat{T_i}$ based on the learnt information. $\mathcal{A}$ succeeds if R believes that $\hat{T_i}$ is T_i during the authentication. The consequence of this attack is tag impersonation.

- **Relay Attack**: To pass identification and/or authentication without the fully control of a valid tag, $\mathcal{A}$ relays messages exchanged between R and T_i without modification. No matter how well-designed cryptographic protocols are and how strong the cryptographic primitives are, this attack is unavoidable [125] since the attacker in the middle essentially plays the role of a communication medium or channel. The relay attack has serious security implications as the attacker is able to bypass any application layer (cryptographically sound) security protocols, and the consequence of this attack is tag impersonation in the current setting. A recent instance of such an attack is to use two NFC-enable mobile phones to relay a contactless transactions between a reader and a credit card, as demonstrated in [87, 88], i.e.,

$$\text{card reader} \xleftrightarrow{\text{NFC Channel}} \text{phone A} \xleftrightarrow{\text{WiFi Channel}} \text{phone B} \xleftrightarrow{\text{NFC Channel}} \text{credit card},$$

which breaks the assumption made to almost every credit card payment system that the legitimate card holder is always in the close physical proximity of where the transaction happens (thus is aware of and agrees the transaction). A possible countermeasure of this attack is *distance bounding technology* as described in Chapter 2.

- **Man-In-The-Middle Attack**: This is a variant of the relay attack. $\mathcal{A}$ is not merely interested in passing identification and/or authentication but also keen on recovering the tag's secret credentials, $\mathcal{A}$ may modify the relayed messages and analyze responses. The consequence of this attack is tag impersonation.

- **Denial of Service**: $\mathcal{A}$ tries to insert/block/modify messages transmitted between R and T_i to cause de-synchronization or misunderstanding between the communicating entities. For example, R might update its shared secrets after a session, while T_i does not. As a result, they would no longer be able to authenticate each other.

- **Reverse Engineer of Tags**: $\mathcal{A}$ obtains a sample of the victim tag and reverse engineers it to reveal all its secret credentials, private design of protocols and algorithms. After that, $\mathcal{A}$ tries to mount the above attacks to another similar tag. The fact that tags for low-cost RFID systems are generally easy-accessible makes this attack quite a practical threat, e.g., [106].

Clearly, an ideal scheme for RFID security and privacy is able to resist to all the threats listed above. However, given the constraint that a possible scheme should be efficient and lightweight enough to be implemented, a silver-bullet solution for RFID security and privacy does unlikely exist. Therefore, we expect a security scheme to achieve resistance to some of the attacks listed, which are application-dependent. For example, the fundamental goal and requirement of an authentication protocol is the resistance to *inventorying or rogue scanning*, *tracing or tracking*, *replay attack* and *tag counterfeiting or cloning.*

1.3 Motivation

As discussed, the spread of RFID technology gives rise to significant user privacy and security issues. To secure the upper layer interactions, an confidentiality-, integrity- and availability-preserving channel is expected to be constructed between the reader and the tag. There are two various ways to realize this goal.

- *Cryptographic Approach* (more conventional): One could employ encryptions, in conjunction with message authentication codes (MACs) and error correcting codes, to reach the goal, if and only if robust and lightweight cryptographic primitives are available for RFID tags.
- *Physical Layer Approach* (less conventional): In a conventional communication system, the physical layer is solely responsible for transmission, reception and error correction, whereas data security has been taken care of at the upper layers of the protocol stack. As opposed to this tradition, there has been a considerable recent attention on exploiting physical layer resources, i.e., channel noise, multi-path propagation, space diversity, etc., in the design of security schemes, which result in more compact implementations and/or stronger security in terms of confidentiality, integrity and availability.

For the purpose of this thesis, we follow these two approaches. We start with investigating the design and analysis of the physical layer schemes as we believe: (1) low layer designs are more costless; (2) the asymmetry between the reader and the tag provides interesting characteristics (which do not exist in other communication systems), some of which can be exploited for security purpose. On the other hand, we study the lightweight cryptographic algorithms and primitives as well, with the cross-layer optimization in our mind.

Unless otherwise stated, we exclusively target the security and privacy for low-cost, passive RFID systems in this thesis. The reasons are: (1) low-cost and passive RFID systems will most likely have the biggest impact on consumer security and privacy, due to their potentially large numbers, pervasive deployment and large operating range; (2) unlike ISO 14443-compliant tags, e.g., [170], which are able to perform standard cryptographic functions like AES, DES and even RSA and ECC, low-cost and passive tags, e.g., EPC Gen2, are designed to strike a balance between cost and functionality, with less attention paid to security. Henceforth, the designing of security mechanisms and cryptographic functions is quite challenging under this resources-limited (i.e., 200-4000 gates), power-limited (i.e., harvest power from the electromagnetic field) and cost-limited (i.e., ≤ 5 cents) platform.

1.4 Outline and Main Contributions

The outline and the main contributions of this thesis are the following:

- **Chapter 1** provides a general description of RFID systems, including the basic concepts, underlying principles, applications and active standards. In addition, the security and privacy issues are closely examined. The motivation and the context of this thesis are also clarified.

- **Chapter 2** describes the related work as well as a couple of research topics on which this thesis will focus. We begin with the design and analysis of the lightweight cryptographic symmetric ciphers, e.g., PRESENT, PRINTCipher, GOST, WG-7 and etc.. We then categorize and summarize the recent progress on the design of lightweight authentication protocols, which are expected to provide not only authenticity and computation/storage efficiency but also anonymity, untraceability and other properties desired in the current setting. Moreover, solutions leveraging the electronic characteristics of the physical devices and the randomness in the channels between the communicating devices are scrutinized as well, including: distance bounding protocols, channel impairment for good, fingerprinting technologies, physical unclonable functions. Finally, solutions along a nontechnical way are briefly reviewed.

- **Chapter 3** investigates how to solve the eavesdropping, modification and one particular type of relay attacks toward the tag-to-reader communication in passive RFID systems without requiring lightweight ciphers or secret credentials shared by legitimate parties using a physical layer approach. To this end, we propose a novel physical layer scheme, called *Backscatter modulation- and Uncoordinated frequency hopping-assisted Physical Layer Enhancement* (BUPLE). We further improve its eavesdropping resistance through the coding in the physical layer as BUPLE ensures that the tag-to-eavesdropper channel is strictly noisier than the tag-to-reader channel. Three practical *Wiretap Channel Codes* (WCCs) for passive tags are then proposed: two of them are constructed from linear error correcting codes, and the other one is constructed from a resilient vector Boolean function. The security and usability of BUPLE in conjunction with WCCs are further confirmed by our proof-of-concept implementation and testing on the software-defined radio platform with a programmable WISP tag.

- **Chapter 4** identifies a brand-new and quite powerful family of attacks, called *uni-directional active eavesdropping*, which defeats the customary impression that eavesdropping is a "passive" attack. Besides the formalization and the theoretic analysis of this attack, we set out to fill the literature's gap by demonstrating and empirically evaluation of this new attack towards an EPC Gen2-compliant passive RFID system, using software-defined radio devices working at 860-960MHz and a programmable passive tag, i.e. WISP v4.1. Our experimental results show that the active eavesdropping achieves a significant improvement in the bit error rate of the intercepted communication. Finally, although active eavesdropping in general is not trivial to be prohibited, for a particular type of active eavesdropper, namely a *greedy proactive eavesdropper*, we propose a simple countermeasure without introducing any computation/storage overhead to the current system.

- **Chapter 5** presents a novel attack called *Differential Sequence Attack* (DSA), in conjunction with guessing and determining the internal states, to attack the Hummingbird-2 lightweight block cipher, as we discover that the differential sequences for the last round can be computed by the full cipher and the search space of the key can be reduced due to the property of the differential sequences. Using those observations, our full attack can be divided into two phases: *preparation phase* and *key recovery phase*. In the key recovery phase, by attacking the encryption (decryption resp.) of HB-2, our algorithm recovers 36-bit (another 28-bit resp.) out of 128-bit key with 2^{68} (2^{60} resp.) time complexity if particular differentials of the internal states and of the keys at one round can be maintained to the next round of encryption/decryption. Furthermore, the rest 64-bit of the key can be exhaustively searched and the overall time complexity is dominated by 2^{68}. During the preparation phase, our second algorithm creates the conditions required by the key recovery phase with at least 0.5 probability and 2^{81} effort in time.

- **Chapter 6** reports an ultra-efficient key recovery attack under the chosen-plaintext-attack model against the stream cipher A2U2, which is the most lightweight cryptographic primitive proposed so far. Our attack can fully recover the secret key of the A2U2 cipher by only querying the A2U2 encryption twice on the victim tag and solving 32 sparse systems of linear equations in the worst case, which takes around 0.16 second on a laptop. Our cryptanalysis implies that A2U2 has been completely broken and is not eligible to provide confidentiality and authenticity.

- **Chapter 7** summarizes and concludes our work and suggests possible directions for future research.

Chapter 2

Proposed Solutions for Securing RFID Technology in the Literature

In this chapter, we introduce various design and analysis of security solutions proposed in the literature, with a particular focus on their practicality for low-cost RFID systems. We start with the recent process in the lightweight cryptographic primitives in Section 2.1. Based on the given primitives, design of lightweight protocols targeting privacy-preserving authentication are summarized in Section 2.2. Besides, in Section 2.3, low-cost solutions leveraging the electronic characteristics of the physical devices or the randomness in the channels between communicating devices are examined and surveyed. Finally, non-technical and less-technical solutions are treated in Section 2.4.

2.1 Design and Cryptanalysis of Lightweight Ciphers

The intensive studies toward design, implementation and analysis of lightweight cryptographic primitives for low-cost passive tags lead to the born of a new sub-field of cryptography – *lightweight cryptography*, which is considered as the intersection of electrical engineering, computer science and mathematics. The major tasks of lightweight cryptography are as follows.

- *Design* of new cryptographic primitives and protocols, e.g., stream cipher, block cipher, hash function, identification/authentication methods, etc.. To be specific, research toward this topic can be divided into the three categories:
  - Optimizing implementations for standardized and trusted algorithms/protocols on low-cost or constrained devices, e.g., compact ASIC encryption cores for 128-bit AES [85, 89].
  - Tailoring well-investigated and trusted algorithms/protocols to made them more hardware-efficient, e.g., DESXL [154], a lightweight DES variant, in which the eight original S-boxes in DES is replaced by a single new one.
  - Designing brand new algorithms/protocols taking advantage of characteristics of the low-cost hardware, e.g., PRINTCipher achieves a quite small chip area by embedding the key-dependent part in their design, which origins from the observation that, with an IC printer, there is essentially no cost in changing the circuit that is printed at each run.
- *Analysis* of the primitives and protocols proposed to ensure their cryptanalytic strength, which is even a more active topic. This is because, in the ongoing competition to design the most efficient primitives, aggressive designs are used for hardware/power efficiency, e.g., (1) innovative techniques are less well-understood and may potentially introduce vulnerabilities; (2) the security margins that cryptographic primitives are traditionally equipped with are reduced a lot in order to optimize the performance.

Table 2.1: Recent Design/Implementation of Lightweight Ciphers

	Key size	Block size	Area	Throughput	Logic Process
	[bits]	[bits]	[GE]	[Kb/s]	[μm]
Stream Ciphers					
Trivum [163]	80	N/A	749	100	0.35
Grain [93]	80	N/A	1,294	100	0.13
KASUMI [80]	128	N/A	9,000	850×10^3	90×10^{-3}
ZUC [82]	128	N/A	10,000	1.5×10^6	65×10^{-3}
SNOW 3G [81]	128	N/A	34,000	1.9×10^6	90×10^{-3}
Block Ciphers					
PRINTCipher-48 [137]	80	48	402	6.25	0.18
PRINTCipher-48 [137]	80	48	503	100	0.18
PRINTCipher-96 [137]	160	96	726	3.13	0.18
PRINTCipher-96 [137]	160	96	967	100	0.18
KTANTAN-32 [64]	80	32	462	12.5	0.13
KTANTAN-48 [64]	80	48	571	9.4	0.13
KTANTAN-64 [64]	80	64	684	8.4	0.13
GOST [183]	256	64	651	24.24	0.18
LED-64 [99]	64	64	688	5.1	0.18
LED-128 [99]	128	64	700	3.4	0.18
Piccolo-80 [194]	80	64	683	14.8	0.13
Piccolo-128 [194]	128	64	758	12.1	0.13
KATAN-32 [64]	80	32	802	12.5	0.13
KATAN-48 [64]	80	48	916	9.4	0.13
KATAN-64 [64]	80	64	1,027	8.4	0.13
PRESENT-80 [188]	80	64	1,075	11.4	0.18
KLEIN-64 [97]	64	64	1,981	N/A	0.18
KLEIN-80 [97]	80	64	2,097	N/A	0.18
KLEIN-96 [97]	96	64	2,213	N/A	0.18
DESXL [158]	184	64	2,168	44.4	0.18
mCrypton-128 [154]	128	64	2,500	492.3	0.13
CLEFIA-128 [5]	128	128	2,678	73	0.13
HIGHT [121]	128	64	3,048	150.6	0.25
XTEA [129]	128	64	3,490	57.1	0.13
AES [89]	128	128	3,400	12.4	0.35
HummingBird-2 [84]	128	16	2,159	N/A	0.13

We summarize the lightweight ciphers recently proposed in Table 2.1, where GE is the acronym of *Gate Equivalent*. As can be seen, although, compared with block ciphers, stream ciphers inherits lower hardware complexity and typically operate at a higher speed, the majority candidates proposed are actually block ciphers. In particular, the competitive candidates in the family of lightweight stream cipher, except WG-7, are actually Trivium [60] designed in 2006 and Grain [115] designed in 2007, which are not specifically targeted low-cost devices such as passive tags. This phenomenon may imply that, to be secure enough, the current linear feedback shift register (LFSR)-based design of stream ciphers is hard to be further tailored/optimized. In parallel to this, the nonlinear linear feedback shift register (NFSR)-based design, although has a great potentiality due to the high linear complexity and long period of NFSR sequences, e.g., a recent design of lightweight pseudorandom number generator [162] achieves $1,242$ GE, remain in the infancy and are yet to be fully understood.

Additionally, the structures of the proposed lightweight block ciphers follow the same structure as a general block cipher design, i.e., they can be categorized into: Substitution Permutation Network (SPN) structure, e.g., PRESENT, PRINTCipher, and Feistel-type structure, e.g., GOST, HIGHT, Piccolo. SPN is widely accepted due to its success application to AES, while Feistel-type structures, besides its successful application to DES, generally require a larger number of rounds than an SPN-based construction because of its slow diffusion. However, a nice property of Feistel-type structures, as pointed out in [193, 194], is that it can support a decryption function without much implementation cost.

2.1.1 PRINTCipher

PRINTCipher [137] is a novel lightweight block cipher proposed by Knudsen, Leander, Poschmann and Robshaw in CHES'10, which is the first design that takes the IC printing into consideration. The authors observed that: (1) a key is unlikely to be changed in a tag's life cycle; (2) IC printing does not require all versions of the cipher to be identical and a specific tag can be personalized with a unique key without extra cost.

PRINTCipher-48 (PRINTCipher-96 resp.) is a block cipher with $n = 48$-bit (96-bit resp.) block size and a key length of $l = 80$-bit (160-bit resp.), which adopts SPN structure with $r = 48$ ($r = 96$ resp.) rounds. One round of PRINTCipher-48 is shown in Figure 2.1, where S represents a 3-bit S-box and P represents a 3-bit permutation. Specifically, let the input of one P block be (c_2, c_1, c_0), the output can be described as

$$\begin{array}{lll} (c_2, c_1, c_0) & \text{if} & (a_1, a_0) = 0 \\ (c_1, c_2, c_0) & \text{if} & (a_1, a_0) = 1 \\ (c_2, c_0, c_1) & \text{if} & (a_1, a_0) = 2 \\ (c_0, c_1, c_2) & \text{if} & (a_1, a_0) = 3 \end{array}$$

where (a_1, a_0) are two key bits embedded to each P block during printing.

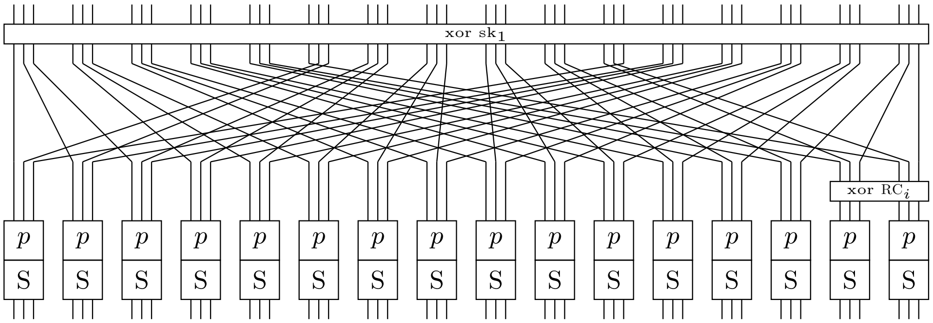

Figure 2.1: One Round of PRINTCipher-48.

The key is split into two parts: the first n bits are used as the whitening key for each round, the remaining $l - n$ bits, as mentioned before, are embedded into the permutation blocks. A round counter RC_i is used which is generated by an LFSR to avoid self-similarity.

Cryptanalytic Results: Although the designers of PRINTCipher claimed its security with respect to the main cryptanalytic methods, the first attack, discovered by Abdelraheem, Leander and Zenner in [7], appeared very soon. This attack, exploiting the fact that the differential characteristics are key-dependent, successfully breaks 22 rounds of PRINTcipher-48 requiring the full code book and about 2^{48} computational steps. Their attack begins with noticing that, in S, all occurring differences are equally probable, e.g., with probability 1/4, and that for every 1-bit input difference, there exists exactly one 1-bit output difference. From this, it follows that starting with a 1-bit input difference, a 1-bit differential trail through r rounds of the cipher occurs with probability $(1/4)^r$. Additionally, if the 1-bit differential occurs, the S-box does not permute the active bit on a differential trail, which is only influenced by the fixed round permutation and the key-dependent permutation P. Therefore, knowing the best differential, one is able to deduce the key.

In CRPYTO'11, Leander, Abdelraheem, AlKhzaimi and Zenner presented a so-called *invariant subspace attack* [150] that breaks the full cipher for a significant fraction of its keys, e.g., 2^{52} keys out of 2^{80} for PRINTCipher-48 and 2^{102} keys out of 2^{160} for PRINTCipher-96. The general idea of the invariant subspace is that the round function, including an SP-layer and a key addition layer, maps the affine subspace (out of the entire key space) onto itself. This property is preserved for an arbitrary number of rounds as long as all the round keys are in this subspace, which results in an efficient distinguisher for this fraction of the keys. Note that the invariant subspace attack displays interesting relationships to other well-established attack techniques, e.g., truncated differential cryptanalysis [130], statistical saturation attack [57, 147, 58], conditional differential cryptanalysis [138, 139], dynamic cube attack [75].

2.1.2 KATAN-32/48/64 and KTANTAN-32/48/64

KATAN/KTANTAN [64] is a family of hardware oriented lightweight block ciphers proposed by Canniere, Dunkelman and Knezevic. Both KATAN and KTANTAN have three variants each, of 32-bit, 48-bit, or 64-bit block size. All ciphers share the same key length of 80 bits, where the only difference between KATAN and KTANTAN is the key schedule. Here we provide a brief description of KATAN-32/KTANTAN-32 as an example.

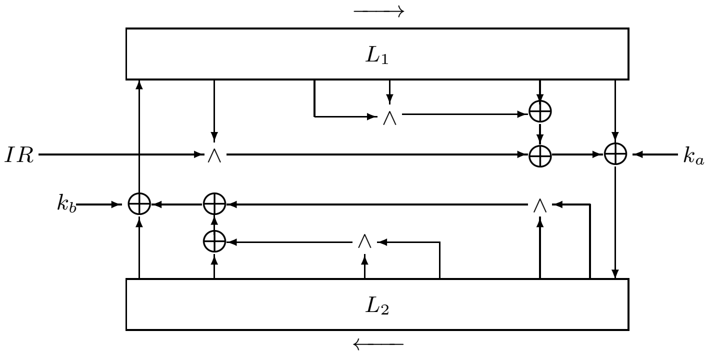

Figure 2.2: Round Function of KATAN-32 (note that, in each clocking, one shift is made and two key bits are added to the state. IR is a round constant which decides whether or not the 9th state of S is used in the state update)

As shown in Figure 2.2, the plaintext is loaded into two NFSRs denoted as L_1 and L_2 (of lengths of 13-bit and 19-bit), where the least significant bit of the plaintext is loaded to the 0th stage of L_2, and the most significant bit of the plaintext is loaded to 12th stage of L_1. Each round, out of the 254 rounds, L_1 and L_2 are shifted to the left by one position, where the new bits produced are loaded in the least significant bits of L_1 and L_2. Let us denote the states of L_1 and L_2 as $(s_0, ..., s_{266})$ and $(l_0, ..., l_{272})$ through the whole encryption process. Thus, the nonlinear recursive relations can be written as

$$\begin{aligned} s_{13+i} &= l_i + l_{6+i} \cdot l_{8+i} + l_{11+i} + l_{10+i} \cdot l_{15+i} + K_i^s \\ l_{19+i} &= s_i + s_{5+i} + s_{4+i} \cdot s_{7+i} + IR_i \cdot s_{9+i} + K_i^l \end{aligned}$$

where $IR = (IR_0, ..., IR_{253})$ is an m-sequence of period 255 (generated by an 8-stage LFSR in $\mathbb{F}_2$) with the initial state "11111111", and K_i^l and K_i^s are the two subkey bits generated by the key schedule with the secret key K.

The key schedule of KATAN family is actually an LFSR of 80 stages defined over $\mathbb{F}_2$. Let the key be K, then the subkey of round i is $K_i^l || K_i^s = k_{2\cdot i} || k_{2\cdot i+1}$, where

$$k_i = \begin{cases} K_i, & i = 0, ..., 79 \\ k_{i-80} + k_{i-61} + k_{i-50} + k_{i-13}, & \text{otherwise.} \end{cases}$$

The key schedule of KTANTAN family is designed under the consideration that the key is burnt (i.e., fixed) to the device. Therefore, it utilizes MUX, AND gate and XOR gate that go with the ASIC to schedule the key bits in a nonlinear and low-cost way. Details can be found in [64].

Cryptanalytic Results: In SAC'11, Bogdanov *et al.* in [37] found a vulnerability in the key scheduling of KTANTAN: as tabulated in Table 2.2, some key bits are not used until very late in the cipher, while some others are never used after some surprisingly small number of rounds, which results in a *3-subset meet-in-the-middle attack*. This reported attack is of time complexity $2^{75.170}$ on the full KTANTAN-32, $2^{75.044}$ on the full KTANTAN-48 and $2^{75.584}$ on the full KTANTAN-64. Compared to the 80-bit security KTANTAN targets, it is only slightly better than the exhaustive search, and, it is not extendable to KATAN, which has a more robust key schedule. Recently, in [2], this vulnerability is further exploited to mount a related-key attack towards KTANTAN. However, related-key attack is not universally accepted as a valid attack model, especially for KTANTAN, where the key is burnt into the device and fixed for its life time. Most recent, Zhu and Gong presented in [223] a novel extension – guessing the intermediate state of KTANTAN before launching the meet-in-the-middle attack, and obtained the best cryptanalytic results on KTANTAN so far, e.g., KTANTAN32/48/64 can be broken with the time complexities of $2^{68.06}$, $2^{70.92}$ and $2^{73.09}$.

Table 2.2: Vulnerability in KTANTAN's Key Schedule [37, 2]

Key variable	First used in round	Key variable	Last used in round
k_{13}	109	k_{38}	164
k_{27}	110	k_{46}	158
k_{59}	110	k_{15}	157
k_{39}	111	k_{20}	131
k_{66}	123	k_{74}	130
k_{75}	127	k_{41}	122
k_{44}	136	k_{3}	106
k_{61}	140	k_{47}	80
k_{32}	218	k_{63}	79

In Indocrypt'10, Bard *et al.* [21] presented some experimental results on AIDA/cube, algebraic and side channel attacks on the reduced rounds of the KATAN family, i.e., 60/40/30 rounds of KATAN-32/48/64. However, these attacks are marginal as they are effective toward a small number of rounds in KATAN. Knellwolf *et al.* proposed to use *conditional differential cryptanalysis* to attack KATAN/KTANTAN in [138] and extended this idea later in [139]. Unlike the conventional differential cryptanalysis that the input pairs are selected uniformly at random, conditional differential cryptanalysis asks for particular pairs of inputs which satisfy some conditions. In fact, the imposed conditions control the propagation of the difference up to a certain round, and therefore may be potentially better to distinguish the cipher from an ideal primitive. Since the round function of KATAN/KTANTAN has the slow diffusion, this method work for a small number of rounds, e.g., the best results given in [138, 139] are the recovery of the 4 bits of the key of 78 rounds of KATAN-32 under a single-key scenario and recovery of 10 bits of the key of 120 rounds of KATAN-32 under a related-key scenario. As can be seen, this method produces some marginal results and does not have practical impact on the security of KATAN family. Note that there is no published attack toward full round KATAN-32/48/64 to the best of our knowledge.

2.1.3 GOST

To be away from other dedicated designs/implementations of symmetric ciphers targeting lightweightness, GOST was developed in the Soviet Union during 1970's as an alternative to the DES developed by US and revisited recently by Poschmann *et al.* in [183] as a competitive candidate for low-cost passive RFID-tags due to its highly efficient ASIC implementation, e.g., 651 GE.

GOST has a block size of 64 bits and a keysize of 256 bits (to offer an extra margin of security). The overall structure, similar to that of DES, is a two branch Feistel network with 32 rounds as shown in Figure 2.3, in which the right half of the block R_i is processed, XORed to the left half L_i, and swapped with the left half, i.e., for $i = 1, 2, ..., 32$,

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus (S((K_i + R_i) \mod 2^{32}) \lll 11) \end{aligned}$$

Note that S represents a set of eight S-boxes, which could be selected at the user's will. For example, to minimize the hardware footprint, one S-box is used eight times in parallel in [183]. In addition, GOST has a simple key schedule: the 256-bit key is divided into eight 32-bit words, i.e., $K_1, K_2, ..., K_8$. Each round, GOST uses one of them according to the array given below, e.g., K_1 is used in rounds $1, 9, 17$ and 31,

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7,$$
$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

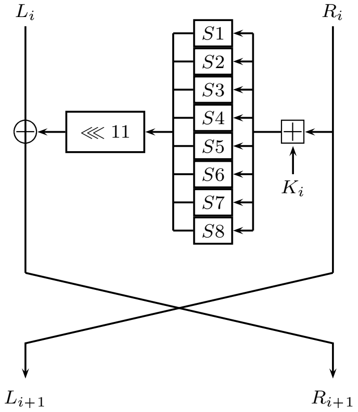

Figure 2.3: Round Function of GOST

Cryptanalytic Results: Although in the past 20 year, GOST has been intensively studied and several related-key attacks and signal-key attacks targeting round-reduced version of GOST have been published, the first single key attack on the full 32-round version of GOST was published recently by Takanori Isobe in [122], which leverages the known property that, providing $R_{24} = L_{24}$ (which in fact happens with probability 2^{-32}), the last 16 rounds become an identity mapping, and thus the effective number of rounds of GOST is reduced to 16. Based on this property, Isobe combined the 3-subset meet-in-the-middle attack as mentioned to extract the entire secret key with 2^{32} plaintext/ciphertext pairs, 2^{224} time and 2^{64} memory. This idea is further extended by Dinur, Dunkelman and Shamir in [65] such that given the same amount of known plaintext/ciphertext pairs, the memory complexity can be further reduced to 2^{36}. To achieve this improvement, the number of effective rounds is further reduced to 8 by first applying the aforementioned reflection property followed by a guessing of the internal state at the start of 9th round. After that, the basic meet-in-the-middle technology or the so-called *2 dimensional meet-in-the-middle* technology is used. Although the recent progress in analyzing this cipher shows the attacks much faster than exhaustive search, neither the time complexity nor the memory complexity are even close to being practical. Without exaggeration, GOST, as a classical design, is the most promising candidate in the family of lightweight cryptography, which integrates durable security and compactness in an elegant way.

2.1.4 Piccolo

In CHES'11, Shibutani *et al.* proposed in [194] a new 64-bit block cipher, called Piccolo, optimized for passive RFID tags. Piccolo has an iterative structure which is a variant of the Feistel network and supports 64-bit block with 80 or 128-bit keys, which are referred as Piccolo-80 and Piccolo-128, respectively. The differences between Piccolo-80 and Piccolo-128 are the number of rounds for encryption/decryption and the key scheduling.

As shown in Figure 2.4, the encryption/decryption of Piccolo, consisting of r rounds (e.g., $r = 25$ for Piccolo-80 and $r = 31$ for Piccolo-128) , takes $X = (X_0, X_1, X_2, X_3) \in \mathbb{F}_2^{64}$, four whitening keys $wk_i \in \mathbb{F}_2^{16}, i = 0, 1, 2, 3$, and $2r$ round keys $rk_i \in \mathbb{F}_2^{16}, 0 \leq i < 2r$, as the inputs, and outputs $Y \in \mathbb{F}_2^{64}$. In each of the r rounds, following is performed

$$\begin{aligned} X_1 &= X_1 + F(X_0) + rk_{2i} \\ X_3 &= X_3 + F(X_2) + rk_{2i+1} \\ (X_0, X_1, X_2, X_3) &= RP(X_0, X_1, X_2, X_3). \end{aligned}$$

The whitening keys are XORed to X_0 and X_2 before applying F during the first and the last round. Moreover, $F : \mathbb{F}_2^{16} \mapsto \mathbb{F}_2^{16}$ consists of four parallel 4-bit S-boxes, whose canonical representatives is $E4B238091A7F6C5D$, followed by multiplying, over $\mathbb{F}_2^4$ defined by an irreducible polynomial $x^4 + x + 1$, with a diffusion matrix

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}.$$

The round permutation $RP : \mathbb{F}_2^{64} \mapsto \mathbb{F}_2^{64}$ transforms $(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ to $(x_2, x_7, x_4, x_1, x_6, x_3, x_0, x_5)$, where $x_i \in \mathbb{F}_2^8$ for $i = 0, ..., 7$.

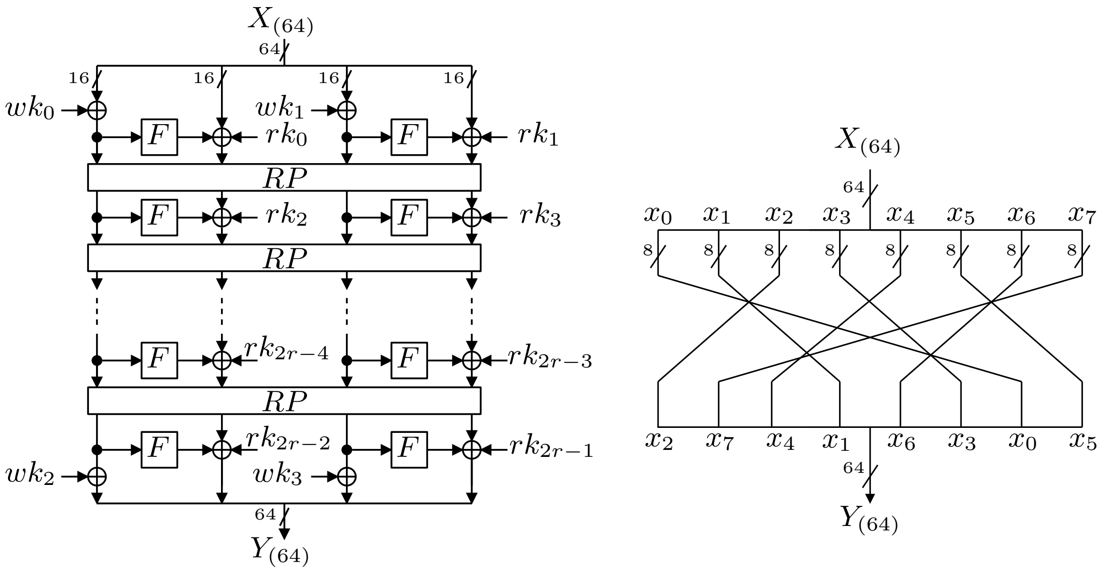

Figure 2.4: Encryption in Piccolo (left) and Round Permutation RP (right)

The key schedule of Piccolo is designed to achieve: (1) hardware efficiency as the registers for storing keys are not required and it leads the almost same gate requirement for each key size; (2) enough immunity against attacks exploiting weakness of the key schedule such as related-key differential and MITM attacks. The key scheduling function for Piccolo-80, divides an 80-bit key K into five 16-bit subkeys $k_i \in \mathbb{F}_2^{16}, i = 0, 1, 2, 3, 4$, and produces round keys $rk_j \in F_2^{16}, j = 0, 1, ..., 2r - 1$ as shown in Table 2.3, where the hex values are the public round constants.

Table 2.3: Key Schedule of Piccolo-80

Rd.	Rd. key	Rd.	Rd. key
0	$(0\mathrm{x}0071c293d, 0\mathrm{x}01f1a253e) \oplus (k_2, k_3)$	1	$(0\mathrm{x}01718213f, 0\mathrm{x}02f163d38) \oplus (k_0, k_1)$
2	$(0\mathrm{x}027143939, 0\mathrm{x}03f12353a) \oplus (k_2, k_3)$	3	$(0\mathrm{x}03710313b, 0\mathrm{x}04f0e0d34) \oplus (k_4, k_4)$
4	$(0\mathrm{x}0470c0935, 0\mathrm{x}05f0a0536) \oplus (k_0, k_1)$	5	$(0\mathrm{x}057080137, 0\mathrm{x}06f061d30) \oplus (k_2, k_3)$
6	$(0\mathrm{x}067041931, 0\mathrm{x}07f021532) \oplus (k_0, k_1)$	7	$(0\mathrm{x}077001133, 0\mathrm{x}08f3e6d2c) \oplus (k_2, k_3)$
8	$(0\mathrm{x}0873c692d, 0\mathrm{x}09f3a652e) \oplus (k_4, k_4)$	9	$(0\mathrm{x}09738612f, 0\mathrm{x}0af367d28) \oplus (k_0, k_1)$
10	$(0\mathrm{x}0a7347929, 0\mathrm{x}0bf32752a) \oplus (k_2, k_3)$	11	$(0\mathrm{x}0b730712b, 0\mathrm{x}0cf2e4d24) \oplus (k_0, k_1)$
12	$(0\mathrm{x}0c72c4925, 0\mathrm{x}0df2a4526) \oplus (k_2, k_3)$	13	$(0\mathrm{x}0d7284127, 0\mathrm{x}0ef265d20) \oplus (k_4, k_4)$
14	$(0\mathrm{x}0e7245921, 0\mathrm{x}0ff225522) \oplus (k_0, k_1)$	15	$(0\mathrm{x}0f7205123, 0\mathrm{x}10f5ead1c) \oplus (k_2, k_3)$
16	$(0\mathrm{x}1075ca91d, 0\mathrm{x}11f5aa51e) \oplus (k_0, k_1)$	17	$(0\mathrm{x}11758a11f, 0\mathrm{x}12f56bd18) \oplus (k_2, k_3)$
18	$(0\mathrm{x}12754b919, 0\mathrm{x}13f52b51a) \oplus (k_4, k_4)$	19	$(0\mathrm{x}13750b11b, 0\mathrm{x}14f4e8d14) \oplus (k_0, k_1)$
20	$(0\mathrm{x}1474c8915, 0\mathrm{x}15f4a8516) \oplus (k_2, k_3)$	21	$(0\mathrm{x}157488117, 0\mathrm{x}16f469d10) \oplus (k_0, k_1)$
22	$(0\mathrm{x}167449911, 0\mathrm{x}17f429512) \oplus (k_2, k_3)$	23	$(0\mathrm{x}177409113, 0\mathrm{x}18f7eed0c) \oplus (k_4, k_4)$
24	$(0\mathrm{x}1877ce90d, 0\mathrm{x}19f7ae50e) \oplus (k_0, k_1)$		

In addition, the key schedule also generates $wk_i \in F_2^{16}, i = 0, 1, 2, 3,$ through

$$\begin{aligned} wk_0 &= (k_0^L, k_1^R) \\ wk_1 &= (k_1^L, k_0^R) \\ wk_2 &= (k_4^L, k_3^R) \\ wk_3 &= (k_3^L, k_4^R), \end{aligned}$$

where k^L and k^R are left and right half 8 bits of k, respectively. The key schedule for Piccolo-128 is similar and can be found in [194].

Cryptanalytic Results: There is no published cryptanalytic results on this new primitive besides its designers' self-evaluations – Piccolo has enough immunity against differential-type, linear attacks, related-key attacks, MITM/slide/saturation attacks.

2.1.5 WG-7

Observing the phenomenon that only a tiny amount of data, e.g., 16 bits, needs to be processed and transmitted each pass between the reader and tags, where the use of block ciphers (which usually have the block size of 64-bit or 128-bit) seems to be overkill, Luo, Chai, Gong and Lai designed a lightweight stream cipher WG-7 [152] for passive RFID tags, which is a variant of the WG stream cipher [172] as submitted to the eSTREAM project. WG-7 includes a 23-stage LFSR with each stage over the finite field $\mathbb{F}_2^7$ and a nonlinear filtering function which is realized by Welch-Gong (WG) transformation [105]. The key of WG-7 is of size 80-bit, while the IV is of size 81-bit. The microcontroller implementation [152] and FPGA implementation [149] demonstrated that the performance of WG-7 beats Trivium and Grain.

Keystream Generation: The keystream generation is shown in Figure 2.5, where the 23-stage LFSR is defined by a primitive polynomial $f(x) = x^{23}+x^{11}+\beta$ over $\mathbb{F}_2^7$ (defined by $g(x) = x^7+x+1$) and β is a root of $g(x)$. The nonlinear WG transformation $WG : \mathbb{F}_{2^7} \to \mathbb{F}_2$, is applied to generate the keystream from the LFSR, which is the cubic decimation of the original WG transform for better security (while preserving the ideal two-level autocorrelation property as the original WG transform), i.e.,

$$WG(x) = f(x^3) = Tr(x^3 + x^9 + x^{21} + x^{57} + x^{87}), x \in \mathbb{F}_{2^7}.$$

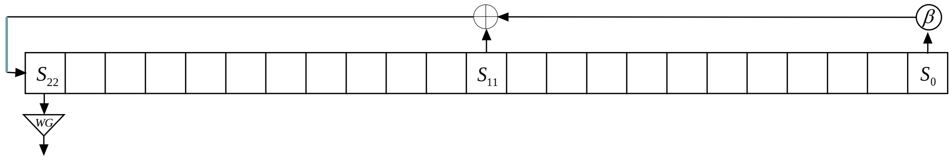

Figure 2.5: Keystream Generation of WG-7

Resynchronizatio/Initialization: Like any other stream cipher, WG-7 is initialized before outputting the keystream. Let the states of the LFSR be represented as $S_0, S_1, \ldots, S_{22}$, $S_i \in \mathbb{F}_2^7$, the key bits be $K_{0,\ldots,80}$ and the IV bits be $IV_{0,\ldots,81}$. The key and the IV are loaded into the LFSR according to the following rules, i.e., for $0 \leq i \leq 10$,

$$\begin{aligned} S_{2i} &= (K_{7i,7i+1,7i+2,7i+3}, IV_{7i,7i+1,7i+2}) \\ S_{2i+1} &= (K_{7i+4,7i+5,7i+6}, IV_{7i+3,7i+4,7i+5,7i+6}) \\ S_{22} &= (K_{77,78,79}, IV_{77,78,79,80}). \end{aligned}$$

After that, it runs for 46 clock cycles with a nonlinear permutation feedback called $WP : \mathbb{F}_2^7 \mapsto \mathbb{F}_2^7$, which is defined as

$$WP(x) = t^3 + t^{58} + t^{99} + t^{117} + t^{123} + 1, \text{ where } t = x + 1, x \in \mathbb{F}_2^7.$$

Cryptanalytic Results: It is claimed in the same paper that WG-7 provides the ideal two-level autocorrelation because of the underlying filtering function. Besides, the authors also show that WG-7 is secure against time/memory/data trade off attack, differential attack, algebraic attack, correlation attack and *discrete fourier transform* (DFT) attack as described in [91]. On the other hand, the attacks against the original WG stream cipher (11-stage LFSR defined over $\mathbb{F}_2^{29}$), e.g., [217], seems un-transplantable to WG-7 as the number of rounds in resynchronization/initilization has been increased to avoid predictable differentials when IV can be freely chosen by the attacker.

2.1.6 Other Lightweight Primitives

Besides the ones summarized in Table 2.1, there are many other lightweight design/analysis of cryptographic primitives.

Message Authentication Code: Shamir in [191] proposed SQUASH, which, although based on the Rabin public-key cryptosystem, performs very well on benchmarks. By denoting tag's response, tag's secret key, reader's challenge, and truncation function as R, K, C, and T respectively, SQUASH can be simply represented as

$$R = T((\sum_{i=0}^{l-1} f_i(K, C))^2 \bmod N),$$

where the f_i's are the nonlinear mixing functions realized by one NFSR. Note that N is a composite Mersenne number, e.g., $N = 2^{1277} - 1$, which is not only easy to store (since its binary representation is a sequence of 1277 ones), but also makes the modular computation particularly simple. Khaled and Serge later in [179] analyzed and attacked an early version of SQUASH, which uses an LFSR as the mixing function expanding the XOR of the key and the challenge. However, the security of SQUASH in general remains open.

Hash Function: As early as in 2003, Weis in his thesis [215] discussed using NFSRs to build a low-cost hash function. However, this idea has not been elaborated so far. Rather recently, lightweight hash functions began to receive attention. Bogdanov *et al.* in [32] firstly described ways of using the PRESENT block cipher in hashing modes of operation and which achieves 64-bit collision resistance with 1600 GE. Badel *et al.* in [24] presented a lightweight hash-function family ARMADILLO, which has recently been attacked in [35]. Aumasson *et al.* in [6] designed a dedicated lightweight hash function Quark (1379 GE for

64-bit collision resistance) using sponge functions as domain extension algorithm and an internal permutation inspired from the GRAIN and the KATAN; Guo *et al.* in [98] proposed a family of lightweight hash functions uses a sponge-like construction as domain extension algorithm and an AES-like primitive as internal unkeyed permutation and achieves 1120 GE for 64-bit collision resistance. The latest proposal is SPONGENT in [30], which has the smallest footprint among all hash functions published so far at all security levels it attains.

Public-key Schemes: Lightweight public-key schemes represent another promising avenue of research, even though its implementation remains too heavyweight at the current stage: WIPR in [175] is a full-fledged public key identification scheme following the idea of randomized Rabin function [190], which is secure, e.g., 1024-bit, yet highly efficient, e.g., 5705 GE. The security of reduced WIPR is investigated in [218], where, as an additional contribution, two variants are proposed to improve its security and to further reduce its hardware cost; Pendl, Pelnar and Hutter in [184], presented their results of an implementation of *elliptic curve cryptography* (ECC) running on the Wireless Identification and Sensing Platform (WISP), which is operated by a low-resource microcontroller MSP430 [205] from Texas Instrument. Their best implementation performs a scalar multiplication using the Montgomery powering ladder within 1.6 seconds at a frequency of 6.7MHz, which cannot meet the practical requirements.

2.2 Lightweight Authentication Protocols

One fundamental function of RFID systems is identification and authentication because RFID technology has its roots in the “identify friend or foe” (IFF) for fighter planes in the Second World War. In this section, we survey the design and analysis of the lightweight authentication protocols for RFID systems.

2.2.1 Security Requirements

Albeit RFID systems are principally simple at first glance, design of the identification/authentication protocols is quite challenging, e.g., such a protocol should ensure both the anonymity and the untraceability of a legitimate tag during execution[1]. Generally speaking, security requirements of such a protocol can be characterized from security, privacy

[1] A classical solution to handle such a problem is zero knowledge proofs, which prevent any leakage of the secret information of the prover. However, zero knowledge proofs is not applicable due to the tight budget on the on-tag computation/storage.

and performance, which are detailed below. Note that theoretic models, which try to generalize and unify these requirements, are studied in literature, e.g., [128, 173, 212, 71].

- Fundamental Requirements:
  - **Authenticity** (security): After execution of the protocol, the reader can identify a legitimate tag with certainty, e.g., an adversary cannot impersonate any legitimate tag (or reader) at the reader (or tag).
  - **Anonymity** (privacy): A protocol between the reader and tag does not leak any fixed or predictable patterns related to a tag's ID or pseudo-ID.
  - **Untraceability** (privacy): The adversary is not able to tell whether a transaction after time $t+\delta$, $\delta > 0$ involves the tag, after eavesdropping on the reader-tag communication before t for a sufficient number of rounds.
- Additional Requirements (or Bonus):
  - **DoS Resistance** (security): Blocking of arbitrary number of sessions of the reader-tag communication before time t does not affect the success probability of the execution of the protocol after t.
  - **Backward Untraceability** (privacy): If the adversary reveals the internal state, e.g., the secret key, of a tag at time t, the adversary is not able to tell whether a transaction before time t involves the tag (note that this property is always obtained by updating the internal state of the tag).
  - **Forward Untraceability** (privacy): If the adversary reveals the internal state of a tag at time t, the adversary is not able to tell whether a transaction after time $t+\delta$, $\delta > 0$, involves the tag, provided that the adversary does not eavesdrop on the reader-tag communication continuously after time t.
- Practicality:
  - **Computational/Storage Efficiency** (performance): A tag has very limited resources in computation and storage as mentioned before. Hence, a suitable protocol must be efficient at least on the tag side.
  - **Scalability** (performance): A protocol must be scalable to allow the reader to deal with such a large tag population. Performing an exhaustive search to identify/authenticate individual tags is difficult when the number of tags is large.

Roughly speaking, a "high quality" protocol should have: (1) all of the fundamental requirements satisfied; (2) as many as possible additional requirements satisfied according to its use; (3) feasibility to be implemented and deployed to real-world low-cost RFID systems. Another observation one may obtain from the list above are the inherit contradictions in several pairs of these requirements, which is the root cause that the design of such a protocol is quite challenging. To name few,

- Security v.s. Performance: This is the most obvious contradiction as shown also in the previous section – efficiency is always obtained at the cost of losing certain security margin.
- Privacy v.s. Scalability: A tag must encrypt its identity with a secret key so that only authorized readers can extract the identity, while authorized readers, in order to authenticate the tag, need to know the identity of the tag to determine the key associated with it. Unsurprisingly, the reader has to try every key in its database until the valid key is found. This is characterized as the *key search problem* in [125] and remains unsolved if the underlying cryptographic primitives are symmetric.
- Privacy v.s. Scalability v.s. Computational-efficiency: The above problem can be perfectly solved by introducing public-key primitives to the tags. For instance, the tag could use a valid public-key and a nonce to encrypt its identity and responses. However, the requirement of computational/storage-efficiency makes this solution invalid at least at the current stage.

In what follows, we summarize the recent progress in this topic. Although protocols based on public-key primitives shows interesting properties, e.g., [151], we constrained ourselves to the symmetric-key-based protocols, as practicality and implementation cost are more concerned in this thesis. For the large body of literature focusing on the design and analysis of RFID protocols, we roughly classify related works according to the characteristics that they achieve best. Note that the classes we present is not mutual exclusive, e.g., [28] can be seen as a scalability-oriented protocol as well as a performance-oriented protocol.

2.2.2 Scalability-oriented Protocols

Molnar and Wagner [169], by extending Weis's early work in [215], proposed a tree-based scheme for library RFID applications. Similar as the Merkle tree, they consider N tags as leaves in a binary tree and each edge in the tree is associated with a secret. Each tag stores the $\log N$ secrets corresponding to the path from the root to the tag. During the authentication, the reader starts at the root and uses the secret to check whether the tag uses the "left" secret or the "right" secret. If the reader successfully authenticates the tag using one of these two secrets, both of them continue to the next level of the tree. If the reader passes all secrets in the path, the tag accepts the reader. Although this protocol is scalable, it needs $O(\log N)$ rounds of interaction and $O(\log N)$ storage on the tag. Also, the more tags an adversary tampers with, the more secrets in the tree are exposed.

Burmester *et al.* in [23] described an authentication scheme with constant key-lookup, which is, to the best of our knowledge, one of the most scalable solutions that preserve privacy as claimed. However, a subtle flaw is found in [156], exploiting the fact that an attacker can launch a three-run interleaving attack to trace and identify a tag. An improved version of this protocol is also presented. Moreover, Cheon *et al.* in [51] exhibits an interesting idea: use the meet-in-the-middle strategy (usually used to attack symmetric ciphers with poor key schedules) to reduce the reader computation to $O(\sqrt{N} \log N)$.

2.2.3 Backward Untraceability-oriented Protocols

Ohkubo, Suzki and Kinoshita in [178] proposed an one-way authentication protocol, known as the OSK protocol, to realize the backward traceability. To be specific, their protocol works as below:

1. Reader: query to wake up the tag.
2. Tag: respond $M = g(s)$ and update s to be $h(s)$, where s is the secret shared by the reader and the tag, g and h are hash functions.
3. Reader: compute $g(h^j(s))$ for s of each tag in the database until it finds a match with the received value M, where h^j is the jth composition of h, e.g., $h^2(s) = h((h(s))$.

Although it is obvious that this protocol is subject to replay attacks, e.g., an eavesdropper can impersonate a tag without knowing the tag's secret, and poor scalability, e.g., the reader needs to perform $O(n)$ work to identity a tag amongst a population of n tags, it introduced an innovative concept, i.e., refreshing the state of the tag each time it is queried by a reader, which inspires considerable following works. In [11], Avoine and Oechslin reduced the reader's search complexity by using a specific time-memory trade-off. Besides, Avoine and Oechslin in [11] as well as Vaudenay in [212] noticed that when the two hash functions are modeled as random oracles, the security of this scheme against a strong model of attackers can be proven.

YA-TRAP [206] is designed to achieve untraceability even when the tag is compromised. In this scheme: a tag pre-shares a time interval $[T_t, T_{max}]$, where T_t denotes the last time it was interrogated, and a secret key with the reader. The reader challenges the tag by sending the current time T_r; if T_r is within the interval $(T_t, T_{max}]$, the tag responds with a keyed hash value of T_r which can be verified by the reader, and updates T_t with T_r; otherwise, the tag outputs a random number that an adversary is unable to distinguish. However, YA-TRAP is vulnerable to both database-side DoS attack and tag-side DoS attack. For the former, an adversary can incapacitate a tag by sending a wildly inaccurate "current times". To solve this, O-TRAP [39], a hash-chain-like scheme, is introduced with a resynchronization mechanism. However, the resynchronization causes $O(l \times n)$ search burdens to the backend database, where n is the number of tags and l is the steps required to ensure synchronization across the hash chain (the adversary could make l a huge number). Hence, it is not practical. Aiming at the same goal, RIPP-FS [56] claims to offer more security properties than its predecessors. However, a complicated tracking technique is later found in [177].

As one may expect, all previous protocols offering backward untraceability requires on-tag hash functions, which are prohibitively expensive for RFID tags and have an unnecessary security property, namely, the collision resistance. In FSE'10, Billet, Etrog and Gilbert proposed a privacy-preserving mutual authentication protocol using a stream cipher [25] (a minor revision from their previous work [15]) and proved that this protocol achieves security, efficiency and a strong privacy close to the backward untraceability. The detail of this protocol is listed below:

1. Reader: query the tag with a nonce n_r.
2. Tag: respond (n_t, G_t), where n_t is a nonce contributed by the tag, $G_t||G_r||G_s = G(n_t||n_r, K)$ is a sequence produced by the stream cipher G with $IV = n_t||n_r$ and the key K (note that, in order to avoid any de-synchronization attack, the backend database keeps one of the current key and one of the most recent used-key for each tag).
3. Reader: search a potential K such that the produced G_t matches the received one; if so, respond with G_r and update the key to be G_s.
4. Tag: if the received G_r matches the G_r produced locally, update the key to be G_s.

In all, every protocol in this category demands a stateful tag, or, implicitly requires the tag has the nonvolatile memory and has enough power to write to or read from this memory, which may result in an increasing in the tag's cost.

2.2.4 Forward-untraceability-oriented Protocols

Lim and Kwon in [155] described a complicated authentication scheme satisfying both forward and backward untraceability, where two hash key chains are used: a forward key chain for updating of tag's secret, and a backward key chain for validating of the server. In addition, if an authentication completes successfully, the tag and the reader both update their secrets using exchanged random numbers. If an authentication fails, the tag updates its secrets using a deterministic algorithm. The protocol provides forward untraceability from the moment that an adversary misses one successful authentication session after time t. It is also shown in [155] that this protocol is secure against server impersonation and DoS attacks. However, it is not scalable, since the reader needs to perform significant computations to update tags' secrets in each session, and two key chains of each tag data required to be stored. Furthermore, [192] presented an attack that breaks its untraceability and backward untraceability.

Song and Mitchell in [196] demonstrated an improved version of this protocol, a simplified description of which is listed below:

1. Manufacture: before the start of this protocol, assign the reader a pair $(u \in \mathbb{F}_2^l, t = h(u))$ and assign the tag a value $t = h(u)$, where h is a hash function and l is a positive integer.

2. Reader: query the tag with a nonce $n_r \in \mathbb{F}_2^l$.

3. Tag: respond (M_1, M_2), where $M_1 = t + n_t$, $M_2 = f_t(n_r + n_t)$ and $n_t \in \mathbb{F}_2^l$ is the nonce contributed by the tag.

4. Reader: search for a potential t such that $n_t = M_1 + t$ and $M_2 = f_t(n_r + n_t)$ and generate and respond with $M_3 = u + (n_t \ggg l/2)$, where $\ggg$ denotes the right circular shift.

5. Tag: compute $u = M_3 + (n_t \ggg l/2)$ and update t with $h((u \lll l/4) + (t \ggg l/4) + n_t + n_r)$ if $h(u)$ equals t, where $\lll$ denotes the left circular shift.

6. Reader: update u with $((u \lll l/4) + (t \ggg l/4) + n_t + n_r)$ and t with $h(u)$.

This protocol is forward-untraceable because as long as the adversary misses M_3, he cannot have u and the corresponding new secret t. Besides, this protocol is secure against tag/reader impersonation, replay, DoS attacks (in their full description, the previous pair of (u, t) is actually stored by the reader for the purpose of re-synchronization) and backward untraceability. The scalability is the only remaining problem for this design. Another example protocol that supports forward and backward untraceability is [33].

2.2.5 Performance-oriented Protocols

HB Family: An interesting avenue of research was initiated by Juels and Weis' HB$^+$ protocol [127] (inherited from Hopper and Blum's early work in [112]), which is a probabilistic algorithm that can be used to authenticate a tag to a reader while hiding the tag's identity from an eavesdropper using extremely simple algebraic operations. The security is reduced to the difficulty of the *Learning Parity with Noise* (LPN) problem, which has been proven to be NP-hard in [34]. In this protocol, the tag and the reader share a secret vector $(\mathbf{x}, \mathbf{y})$. In each round,

1. Reader: wake up the tag.
2. Tag: respond with a blinding-factor vector $\mathbf{b}$, which is randomly generated.
3. Reader: randomly select $\mathbf{a}$ as the response.
4. Tag: generate a noise bit u which takes "1" with probability η, i.e., Prob$[u = 1] = \eta$, compute and respond with $z = (\mathbf{a} \cdot \mathbf{x}^T) + (\mathbf{b} \cdot \mathbf{y}^T) + u$.
5. Reader: independently compute $z' = (\mathbf{a} \cdot \mathbf{x}^T) + (\mathbf{b} \cdot \mathbf{y}^T)$, and validate the tag's response if $z = z'$.

After n rounds, the authentication succeeds if and only if there is no more than $\lceil \eta n \rceil$ mismatched responses. Assuming the intractability of LPN problem, the HB$^+$ protocol is provably secure against passive eavesdroppers. In EUROCRYPT'06, Katz and Shin in [142] extended the security proof of the HB$^+$ protocol, i.e., it remains secure under arbitrary concurrent interactions of the adversary with the honest prover/tag, and, as a consequence, the iterations of the HB$^+$ protocol can be parallelized.

However, an active adversary can easily break this protocol. Gilbert, Robshaw, and Sibert [103] showed a simple-and-effective man-in-the-middle attack, known as the GRS attack. The attacker first modifies one bit of $\mathbf{a}$ to be $\mathbf{a} + \alpha$ for the second pass of every round of HB$^+$ and observes the authentication result, e.g., acceptance or rejection, to learn $\alpha \cdot \mathbf{x}$. By repeating this simply process sufficient number of times, the attack is able to learn $\mathbf{x}$. The same GRS manipulation can be applied to blinding vectors $\mathbf{b}$ to recover $\mathbf{y}$. To thwart the GRS attack, a variety of protocols built upon HB$^+$, such as HB^{++} [20], HB* [69], etc., have been designed. However, Gilbert, Robshaw and Sibert in [101] showed again that these variants are vulnerable to GRS-like attacks as well.

In EUROCRYPT'08, Gilbert, Robshaw and Seurin [102] presented another two interesting variants, known as Random-HB$^{\#}$ and HB$^{\#}$, which are proved to be resistant to the GRS attack in the sense that the adversary is only allowed to manipulate the challenges from the reader to the tag. The core idea is to use two secret matrices $\mathbf{x}$ and $\mathbf{y}$ to replace the secret row vectors in HB$^+$. The difference between these two versions lies in the structure of the secret matrices: while in Random-HB$^{\#}$ these two are completely random, e.g., $\mathbf{x} \in \mathbb{F}_2^{p\times m}$, $\mathbf{y} \in \mathbb{F}_2^{q\times m}$, thus $(p+q)m$ bits of storage is needed, HB$^{\#}$ reduces this amount to $p+q+2m-2$ by using Toeplitz matrices. At AsiaCrypt'08, Ouafi, Overbeck, and Vaudenay presented a general man-in-the-middle attack, known as OOV attack, in [176], where the adversary is given the ability to modify all messages, against all HB-like protocols, especially it recovers the shared secret in 2^{25} or 2^{20} authentication rounds for HB$^{\#}$ and 2^{34} or 2^{28} for Random-HB$^{\#}$, depending on the parameter set. The crucial observation exploited by the OOV attack is that, providing an adversary modifies the messages going in both directions in a smart way, he can compute the hamming weight of the vector $\bar{\mathbf{a}}\mathbf{x} + \bar{\mathbf{b}}\mathbf{y} + \bar{\mathbf{z}}$, where $\bar{\mathbf{a}}, \bar{\mathbf{b}}, \bar{\mathbf{z}}$ are the modifications applied to $\mathbf{a}, \mathbf{b}, \mathbf{z}$ in the victim protocol. Additionally, there is a recent trend to replace the linear encoding operation or vector/matrix multiplication in HB-like protocols by nonlinear operations, e.g., [167], which is expected to have a higher security margin. However, the security of this nonlinear variant remains doubtful [1]. Most recent, Li, Gong and Qin in [148] produced a novel derivative named LCMQ protocol (standing for the combination of learning parity with noise, circulant matrix, and multivariate quadratic), which uses a lightweight and secure (against ciphertext-only attack) encryption/decryption based on the circulant matrix multiplication/inversion to replace the underlying linear encoding in HB$^+$. The security of this protocol is also proved under a generalized man-in-the-middle model in [148].

In all, HB$^+$ is still the most elegant design in this family, given its lightweightness and proved security under the passive attacks, which seems to be suffice for low-cost RFID systems. Its variants, although provide extra security under active attacking models, are unnecessarily complicated and away from the original design goal. It is worth to mention, as a conventional challenge and response protocol, HB family still has the scalability problem as the reader has to go through every possible key to identity thus authenticate the current tag. Besides, a public denunciation of protocols in this family is that the success of the authentication is only guaranteed within a certain probability less than one.

EPC Gen2 Family: EPC Gen2 tag is designed to strike the balance between cost and functionality, with little attention paid to security. To address this problem, particular protocols, e.g., [48, 59, 202], are proposed exclusively for this standard.

The protocols designed by Chen *et al.* [48] and Qing *et al.* [59] are expected to provide mutual authentication. However, two design flaws make them vulnerable to tracking and replay attack: (1) the reader contributes the randomness to the protocol only; and (2) *Cyclic Redundancy Check* (CRC) is inappropriately treated as a hash function. In fact, CRC inherits strong linearity and is designed to support error detection particularly with respect to burst errors, not security. For any CRC and for any input a, b, c, $d \in \mathbb{F}_2^l$, $CRC(a \parallel b) + CRC(c \parallel d) = CRC((a + c) \parallel (b + d))$ always hold.

In Gen2$^+$ [202], instead of transmitting identity, PIN etc., the tag responds with two 7-bit random numbers, say a and b, which serve as an index of a random l-word string k, denoting the key pool, which is shared with the database, as well as creating a so-called check c by XORing the two least significant bit (lsb) of ath word and bth word. The database first removes tuples of tag which do not satisfy c and next computes ck' which is the majority vote of the CRC for the interval $[a, b]$ of the remaining tags. Let us denote the substring of k from ath word to bth word as $k[a : b]$. The tag then makes use of its local k to compute $ck := CRC(k[a : b])$ and compares ck with ck'. The tag does not respond if the Hamming distance between ck and ck' is greater than some threshold. Otherwise, the tag sends the locally stored EPC. This protocol is clearly subject to replay attacks because only the tag contributes to the randomness of protocol flows. Besides, it is possible for an adversary to gradually build up sufficient information about the CRC of k and then recover the tag's keypool k. As can be concluded, to design a secure protocol targeting even only fundamental requirements without using cryptographic primitives is unlikely to be success.

After the recognition of this fact, Blass *et al.* in [28] proposed F_f-family of protocols which is a cross-layer design of cryptographic primitive, e.g., an HMAC-like function F_f, as well as the protocol that relies on it. The benefit of this design philosophy is that the cryptographic function can be customized to achieve minimalism [124]. Besides, the "key search" is faster because the protocol works in a different way: (1) the tag provides the reader with series of one-way results computed over its key; and (2) the reader compares these one-way results with the entries of its database using the key included in each entry, the reader identifies the entry in its database, whose series of one-way results matches all the one-way results received. However, [26] found the connections between the F_f protocol and the LPN problem, and showed a key-recovery attack with time complexity of about 2^{38} against the instance has a 512-bit secret key.

2.2.6 Others

Yoking proof allows a verifier or a reader to collect the proof of the simultaneous presence of two tags in a specified communication range, e.g., a product and its safety cap must be stayed together. Two-party yoking proof is introduced by Juels in [123]. Saito and Sakurai in [201] found a replay attack on this protocol and improved it by using time stamps. Furthermore, they generalized the yoking protocol to be used for a group of tags. Piramuthu in [181] found another replay attacks on Saito and Sakurai's protocol and proposed an modified yoking protocol.

In [4], an interesting time-released-based pairing protocol for passive RFID systems is proposed, which exploited an advantage shared by the legitimate tag and the legitimate reader – the amount of uninterrupted time spent by the two legitimate devices in the proximity of each-other. This idea, originating from the timed-release cryptography [161], exhibits a great resource of secret credentials for identification and authentication and deserves future research.

In some applications, the bearer of a tag might change. An *ownership transfer protocol* allows transferring the ownership over a tag from the current owner to the new owner in a secure and private way. Here "ownership" means having authorization to identity/authenticate a tag and read/write all of the related information. Typical designs can be found in [166, 141, 197].

2.3 Physical Layer Approaches

In a conventional sense, security and privacy is viewed as an independent feature addressed above the physical layer, and all cryptographic protocols as mentioned are designed and implemented with the assumption that the physical layer has already been established and provides an error-free link. Motivated by Wyner's early work [216, 180] and advances in the communication technologies, there has been a considerable recent attention on studying the fundamental ability of the physical layer to provide security for upper layers. Compared to the prevalent cryptographic approaches, the physical layer approach presents embedded security properties by utilizing random processes from physical world. In this section, we survey the previous designs leverage the physical characteristics to provide security and privacy for RFID applications.

2.3.1 Distance Bounding Protocols

Desmedt *et al.* presented at CRYPTO'87 a new method, known as the *mafia fraud* [61], to defeat any cryptographic authentication protocol. Based on the chess grandmaster problem [43], this attack allows the adversary to successfully pass the authentication by relaying the messages between a verifier and a legitimate prover. As one of the countermeasures, the *distance bounding protocol* was proposed in [14, 22], which leveraging the facts: (1) nothing travels faster than light [45]; (2) any relaying operation takes time, which may cause observable delays and thus informs the verifier that the responses received might be illegible, even they are algorithmically correct.

A distance bounding implicitly accomplishes, as pointed out by [3], *authentication* and *distance checking*. Authentication, in its conventional sense, is a process whereby one party is assured of the identity of another party involved, while distance checking in fact refers to a process whereby one party is assured, through acquisition of corroborative evidence, that a given property on its distance to another party involved is satisfied. It is intuitive that any authentication protocol is able to provide authenticity but may not be suitable for distance checking. This is because the distance between two parties is measured by the *round trip time* (RTT) of a message. As a consequence, any authentication protocol that works for distance checking must enable at least for one party, e.g., the resource-constrained prover, a quick way to respond, or stated in another way, should guarantee the operations performed by at least one party is ultra-lightweight. The existed designs follow this principle.

Although the distance bounding protocol was created to fight with the mafia fraud, its uses are not limited to this, e.g., besides the mafia fraud, it also deals with the following:

- *Distance Fraud*: The prover is fraudulent and tries to convince the verifier that he is closer than is actually the case.
- *Terrorist Attack*: The prover collaborates with an attacker, who wants to convince the verifier that the real prover is in the neighboring.

Note that, in most cases, the distance bounding protocols aim to prevent mafia fraud and distance fraud.

Hancke and Kuhn's Protocol: Brands and Chaum [17] designed the first distance bounding protocol, which is published in EUROCRYPT'93. This protocol works as follows:

- Slow Phase: Both the verifier and the prover generate random binary sequence $C = c_1, c_2, ..., c_n)$ and $R = (r_1, r_2, ..., r_n)$, respectively.

- Fast Phase: The verifier transmits one challenge bit c_i at the ith time slot, $i = 1, ..., n$, to which the prover responds immediately with r_i. The verifier times the delay between sending c_i and receiving r_i.
- Slow Phase: After all n bits have been exchanged, the prover completes the protocol by transmitting a message authentication code or digital signature for the two binary sequences of C and R.

Hancke and Kuhn's observed in [117] that the last phase of Brands and Chaum's protocol can be removed, thus proposed the first distance bounding protocol for RFID applications, which is shown pictorially in Figure 2.6. Note that in the scenarios of RFID, the prover is always the tag while the verifier is always the reader. In USENIX'07, Drimer and Murdoch in [72] implemented this protocol on the Chip & PIN payment system in compliance with EMV standard [79] as a practical solution for the relay attacks identified in the same paper.

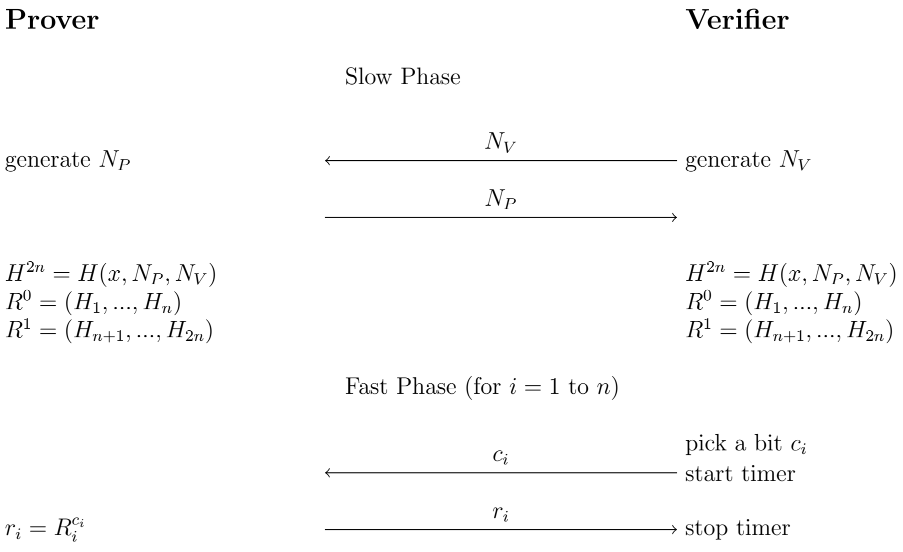

Figure 2.6: Hancke and Kuhns Protocol

Multistate Enhancement of Hancke and Kuhn's Protocol: A potential vulnerability in this protocol, as expected by sharp readers, is that after eavesdropping the slow phase, the attacker could start to query the victim tag in prior to the start of the fast phase. In this interaction with the tag, the attacker randomly selects a $\hat{c}_i$ (and unsurprisingly he has 0.5 chance to get $\hat{c}_i = c_i$) and stores the tag's corresponding response. On the other hand, to interact with the reader in the fast phase, he responds using the tag's answer if $\hat{c}_i = c_i$; responds using a random bit otherwise. Therefore, he is expected to fool the reader with $(3/4)^n$ chance, after n bits of c_i are committed. Munilla, Ortiz and Peinado [164, 165] modified the Hancke and Kuhn's protocol by introducing another states, called *void challenges*, in order to reduce the success probability of the adversary. However, their protocol structured differently in the sense that it performs an additional slow phase in which the tag signs the exchanged bits. As stated in [13], if the last phase is interrupted, the whole authentication process is lost. Therefore, protocols without this final slow phase are usually more appealing. Avoine, Floerkemeier and Martin further extended this 3-state approach (state 0, state 1, state void) in [8], by showing that the number of rounds of [164, 165] can be reduced while maintaining the same security level, as well as generalizing this approach to be multistate that improves all existing distance bounding protocols. Their protocol is shown in Figure 2.7, where the attacker has a success probability of $(5/9)^n$.

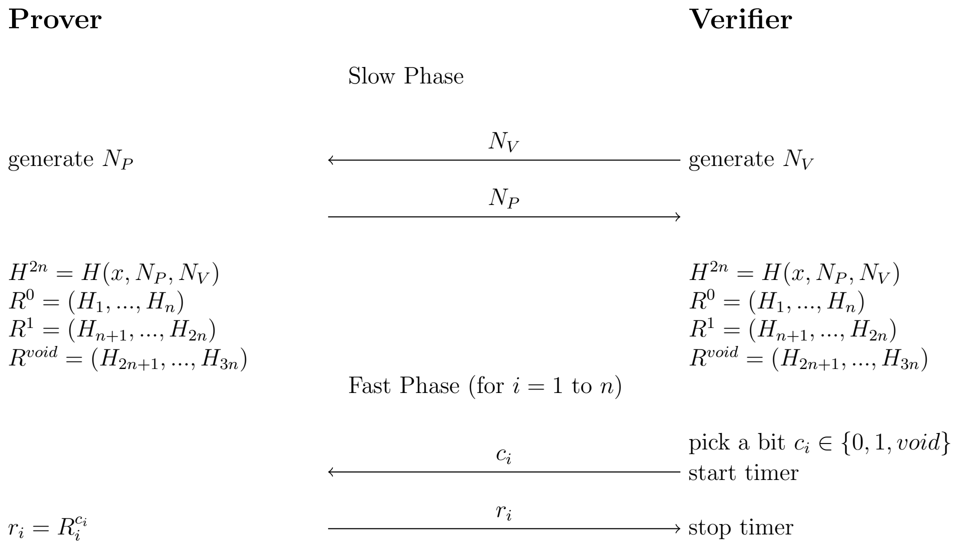
Prover
Verifier
Slow Phase
generate N_P
N_V
generate N_V
N_P
$H^{2n} = H(x, N_P, N_V)$
$R^0 = (H_1, ..., H_n)$
$R^1 = (H_{n+1}, ..., H_{2n})$
$R^{void} = (H_{2n+1}, ..., H_{3n})$
$H^{2n} = H(x, N_P, N_V)$
$R^0 = (H_1, ..., H_n)$
$R^1 = (H_{n+1}, ..., H_{2n})$
$R^{void} = (H_{2n+1}, ..., H_{3n})$
Fast Phase (for $i = 1$ to n)
pick a bit $c_i \in \{0, 1, void\}$
c_i
start timer
$r_i = R_i^{c_i}$
r_i
stop timer

Figure 2.7: Avoine, Floerkemeier and Martin's Protocol

Kim and Avoine's Protocol: In CANS'09, Kim and Avoine introduced an enhanced version in [132] of Hancke and Kuhn's protocol, as shown in Figure 2.8, based on binary mixed challenges, that converges toward the expected and optimal $(1/2)^n$ bound in case of both noisy and error-free channels. Similar idea also appears in [13].

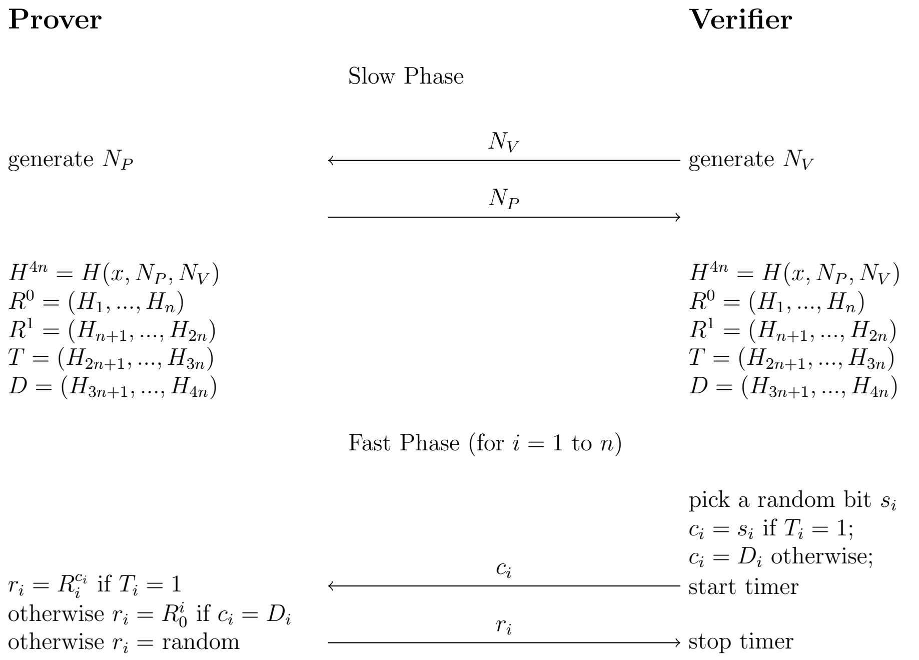
Prover
Verifier
Slow Phase
generate N_P
N_V
generate N_V
N_P
$H^{4n} = H(x, N_P, N_V)$
$R^0 = (H_1, ..., H_n)$
$R^1 = (H_{n+1}, ..., H_{2n})$
$T = (H_{2n+1}, ..., H_{3n})$
$D = (H_{3n+1}, ..., H_{4n})$
$H^{4n} = H(x, N_P, N_V)$
$R^0 = (H_1, ..., H_n)$
$R^1 = (H_{n+1}, ..., H_{2n})$
$T = (H_{2n+1}, ..., H_{3n})$
$D = (H_{3n+1}, ..., H_{4n})$
Fast Phase (for $i = 1$ to n)
pick a random bit s_i
$c_i = s_i$ if $T_i = 1$;
$c_i = D_i$ otherwise;
$r_i = R_i^{c_i}$ if $T_i = 1$
c_i
start timer
otherwise $r_i = R_0^i$ if $c_i = D_i$
otherwise $r_i =$ random
r_i
stop timer

Figure 2.8: Kim and Avoines Protocol

Implementation Issues of Distance Bounding Protocols: The theoretic aspects of distance bounding protocols seem mature enough and the attention began to shift to the practical implementations of this technology, which is, counter-intuitively, even more challenging. As pointed out in [52, 116], the security of such time-of-flight protocols depends not only on the cryptographic protocol itself, but also on the signal design at the physical layer, e.g., short bit duration, signal formats that enable the recipient's instantly reaction on the reception, a communication medium with a propagation speed that approaches the physical limit, and a low-cost transceiver that is able to receive, process and transmit signals in negligible time, are generally needed.

To be specific, at the physical layer, if an attacker could start a response within the allowable time window but still change the value at a later stage, once he knows the correct response, the protocol's security would be compromised. For example, if the reader's receiver integrates the signal amplitude over an entire bit period for demodulation, the attacker could send no energy for the initial $(m - 1)/m$ of the time interval and then send an m-times stronger-than-normal signal during the final $1/m$ of the time interval reserved for the bit. By this method, known as *deferred bit signaling* in [116], the attacker can delay committing to a bit's value by $(m-1)/m$ of the bit period. To mitigate this, an UWB (ultra-wideband) transceiver is designed and analyzed in [117, 143]. Hancke presented another UWB-based physical layer design for a near-field, bit-exchange channel that allowing for a resource-constrained prover in [110]. However, the UWB transmission occupies a large portion of the radio spectrum and effective, e.g., resilient to noise and multi-path effects, in a short-range.

On the other hand, Rasmussen and Capkun further investigated the design of the transceiver under such a scenario in [186], where a prototype prover, that is able to receive, process and transmit signals in less than 1ns is built. Their implementation leverages a quite interesting point – the time needed for signal conversion and demodulation can be saved if the protocol is designed in a proper way.

Others: Trujillo-Rasua, Martin and Avoine in [208] proposed an instance of the graph-based protocol that resists to both mafia and distance frauds without sacrificing memory. Capkun, Defrawy and Tsudik exhibited in [49] another direction of development of the distance bounding technologies by considering a brand new scenario that a set of provers interact with a set of verifiers, which is motivated by applications such as group device pairing and location-based access control. Their key idea is to let the passive verifier obtains a distance bound, when the active verifier executes distance bounding with the prover. Avoin *et al.* gave an unified framework for analyzing RFID distance bounding protocols in [3]. Rasmussen and Capkun in [185] analyzed location privacy problem in the distance bounding protocols by showing location and distance between communicating partners can be leaked to even passive attackers. Peris-Lopez *et al.* in [182] scrutinized the combined use of cryptographic puzzles and distance-bounding protocols. A mutual distance bounding protocol is proposed in [221], which uses an additional binary sequence to determine, for the two participants, who plays the role of prover/verifier.

2.3.2 Channel Impairment for Good

There has been a considerable recent attention on investigating the security implications of the physical layer, known as *wireless physical layer security.* The breakthrough concept behind is to exploit the characteristics of the wireless channel, such as fading or noise, which are traditionally considered as impairments. Following the original ideas [216, 180, 53] proposed in 1970s, the theorists intend to create a clear and clean framework by evaluating secret capacity for different channel models, e.g., [145, 96, 55, 159], and constructing algebraic codes to achieve this optimum goal, e.g., [207, 168], while the practitioners tries to apply this idea to the following topics, which are conventionally covered by cryptography:

- Key generation/extraction using channel reciprocity, e.g., [38, 10, 12]. Note that an implicit requirement for key generation/extraction is that the participants are equally powerful.

- Confidentiality- or integrity- or availability-preserving channel construction under eavesdropping, modification, jamming attacks: e.g., [47, 220, 9].

The broadcast nature of an RFID communication implies that some ideas of the physical layer security are applicable to the RFID scenarios as well. However, the potentiality of the physical layer approach has been far away from been fully explored and there is very limited schemes following this direction.

To construct an unidirectional confidentiality-preserving channel from the tag to the reader, cooperative-jamming methods are introduced in [126, 44]. To protect the unwanted scanning of tags, Jules in [126] proposed a conceptual scheme that the common tag and the blocker tag (both worn by the customer) transmit two identifiers at the same time, where the latter could simulate the full set of all possible k-bit identifiers of tags, which are arranged in a binary tree of depth k. This tree is then traversed by the reader, who queries tags in a bit-by-bit manner. Once the blocker tag is functional, when the reader queries the bit that lies in the "privacy zone" of the binary tree, the blocker tag simultaneously broadcasts both a '0' bit and a '1' bit (no matter what the response is from the real tag). The net effect is that the blocker tag "blocks" the reading of wanted tags. It is worth mentioning that the blocker tag could be recognized as a source of artificial noise. However, bitwise synchronization and pre-shared secrets required between the reader and the friendly jammer may be problematic in real-world applications. Melanie *et al.* proposed a battery powered device, the *RFID Guardian*, in [187], which not only produces a randomly modulated jamming signal, but also allows the user to upload access control lists indicating which party can perform what operation on which population of tags.

Later in [44], this cooperative-jamming method has been generalized and refined for the key distribution, where the noisy tag (different names, but behaves similarly as the blocker tag) is owned by the RFID system instead. By assuming that all tags reply simultaneously, there is another observation: if both the noisy tag and the conventional tag transmit a "1" ("0" resp.), the reader as well as the eavesdropper get symbol S_{11} (resp. S_{00}). If both of them transmit a different bit, then S_{01} or S_{10} will be received. Assuming the channel is additive, the binary sequence of S_0, S_1 becomes a ternary sequence of S_{00}, S_{01}, S_{10}, S_{11} by superimposing, where S_{01} and S_{10} are indistinguishable to the attacker. Next, the reader discards S_{00} and S_{11}, and collects only S_{01} and S_{10}, referred to as *protected bit.*

Based on this straightforward principle, in a protocol proposed in the same paper, the tag tries to send back a pseudo random sequence and the noisy tag emits another pseudo random sequence (shared with the reader in advance) to jam it. From the mixed signal, the reader chooses some of the protected bits as the secret key for the future encrypted communication. As one can see, this cooperative jamming method provides a good solution towards eavesdropping. However, the jamming signal, though makes the reader's channel far better than the attacker's channel, causes bad Signal to Noise Ratio (SNR) at the RF frontend of its neighboring readers, which may reduce the reliability and efficiency of the whole RFID system. In addition, Bringer and Chabanne in [18] observed that the proposed schemes can be envisioned as an application of the wiretap channel model [216]. Bringer *et al.* in [19] improved [44] by encoding the messages exchanged in the protocol using the *Integrity-code* as proposed in [47]. Therefore, the resultant channel is not only confidentiality-preserving but also integrity-preserving under active attacks.

In CHES'07, Savry *et al.* in [200, 50] constructed a noisy reader by exploiting the fact that a passive tag is able to modulate a noisy carrier generated by a reader during its reply. Similar idea also appears in [108, 107]. Note that noise is in the same bandwidth as the message, by which both the amplitude and phase of the reflected signal are blurred. Meanwhile, the reader, by knowing the noise that it sent, is able to subtract this noise and to retrieve the tag's answer, while the eavesdropper presumably cannot. However, this approach may be illegal at least if the broadcast power is too high and it could cause severe disruption of all nearby RFID systems as well. Additionally, the idea proposed in [220] to solve the jamming problem in the wireless sensor network seems transferrable to the case of RFID communication to create an availability-preserving channel.

Recently, UWB's implications on the physical layer security began to receive attention as it can "hide" the signal in the time-domain, e.g., [119, 133]. Considering the fact that the passive UWB tag and the corresponding reader are commercially available, e.g., [204], UWB-driven physical layer security for RFID communication deserves future research.

2.3.3 On the Fingerprinting of RFID Tags

The proliferation of wireless technologies has triggered a number of research initiatives to detect illegally operated radio transmitters and identify wireless devices by using physical characteristics of the transmitted signals, e.g., [209, 219]. These characteristics are usually introduced by imperfections of transceivers caused by manufacturing deviations.

In 2003, Weis already noted in [215] that non-unique IDs can uniquely identify a tag by observing the particular signal constellation they carries. Danev *et al.* in [67] officially introduced this concept to the cases of identifying ISO14443 RFID tags in a controlled environment, by exploiting the modulation shape and burst and sweep spectral features of the signals emitted by tag. Their experiments show that a set of 50 tags of the same manufacturer and type can be identified with an error rate of around 4% based on stable fingerprints in a measurement environment that requires close proximity and fixed positioning of the tag with respect to the acquisition antennas. In WiSec'10, the same group of researchers further investigated this idea and experimentally verified a potential impersonation attack in [70], such that the modulation-based identification, e.g., the proposal in [67], can be impersonated with an accuracy close to 100% by simply replaying the used features, while the transient-based features, e.g., the schemes in [209], are much harder to be reproduced since these features can be channel- and antenna-dependent. By "transient-based features", we mean that extracting unique features from the radio signal transient shape at the start of each new packet transmission.

Similar as [67], the physical layer identification of UHF RFID tags in compliance with the EPC Gen2 standard is studied in [222] by primarily leveraging the time interval error, i.e., how far each active edge of the clock varies from its ideal position, and the average baseband power, i.e., the average power of an acquired RN16 preamble. The gained entropy is enough to uniquely identify at most of 2^6 tags independently of the population size.

Every blade has two edges – although this result seems positive in prevention of device cloning, e.g., fake ePassports, counterfeit products, it causes serious privacy issues as pointed out in [225]. To be specific, although the tag's digital identify is usually invisible to the rogue scanners thanks to the privacy-preserving protocols as mentioned, unique and fixed physical identities leak the bearer's location information. Zanetti *et al.* in [225] built a fingerprint for clandestine people tracking in a shopping mall, using which the mobility traces of people wearing EPC Gen2 tags can be reconstructed with a high accuracy. Removing or reducing the effect of the random hardware impairments in the analog circuitry components is the only solution besides killing/blocking the tag. However, there seems no interest for the manufacturers to produce tags that producing the same radio fingerprint.

There are other ways to construct the physical fingerprints. For instance, the scheme proposed in [113, 114] harvests static identity from existing volatile CMOS memory without requiring any dedicated circuitry. However, as noticed in [203], RAM is subject to data remanence, which means that after a portion of memory has been used for entropy collection once, it will require a relatively extended period of time without power before it can be reused. In [68], NFC tags are created from a collection of randomly bent, thin conductive wires with lengths within 3-7cm to serve the role of certificates (a particular kind of fingerprint) to provide authenticity. The underlying idea is that the random distributed wires within the tag manifest special dielectric and/or conductive properties, which is further sampled and digitized as the tag's fingerprint.

2.3.4 Physical Unclonable Function

A Physical(ly) Unclonable Functions (PUF) is a function that is embodied in a physical structure, which is easy to evaluate with low-power consumption, but hard to characterize and duplicate. Generally speaking, PUF is not (or should not be) a purely mathematical function, but the use of PUF can be understood mathematically, i.e., the whole process can be seen as a hash function or pseudo random function $R = f(C, K)$, where C is a physical stimulus, R is the reaction or response from the PUF f and K can be understood as the secret key embedded into the device when it is manufactured. Note that here K, inherited from manufacture deviations, cannot be simply measured or represented, which is thus, as it name implies, unclonable – given an instance of a particular PUF, it is hard to (physically) reproduce it such that the exact functionality f is preserved.

With this interpretation, PUF can be used obviously in three ways:

- By treating $f(C, K)$ as an ID or a fingerprint of a device under a fixed stimulus C, PUF can be used for identification, e.g., [113, 114].
- By treating K as a cryptographic key, PUF can be used to perform challenge-and-response authentications, e.g., [120, 134, 118], ownership transferring, e.g., [144, 131].
- By treating K as a random seed, PUF can be used to produces randomness, i.e., PUF mixes and expends the C and K to a long sequence, e.g., [113, 114].

Considerable applications of PUFs are proposed based on the assumption that PUF, as a primitive, is both efficient and secure. Unfortunately, this assumption may not be true always since there is no unified framework or rigorous metric to evaluate and analysis the cryptanalytic strength provided by each of the designs. Especially, the lesson people learnt from LFSR, which provides optimum randomness according to [95] and is extremely efficient in hardware, tells that such a function may have a cryptographically simple representation. Therefore, the cryptanalysis of PUFs, as a missing part in the area, is expected to be done in the future. In prior to that, each design of PUF should ship with a reasonable number of input/output pairs as a preparation for cryptanalysts.

On the contrary, the actual constructions of PUFs can be very different based on their setting, e.g., optical PUF, coating PUF, arbiter PUF, ring oscillator PUF, SRAM PUF, butterfly PUF and flip-flop PUF. The most classic design is a silicon PUF in [94], which exploits the random variations in delays of wires and gates introduced during the circuit fabrication process. Given an input challenge, a race condition is set up in the circuit, and two transitions that propagate along different paths are compared to see which comes first. An arbiter, typically implemented as a latch, produces a '1' or a '0', depending on which transition comes first. The reader is refer to [160] for a full treatment of designs of PUF.

2.4 Non-technical and Less-technical Ways

Faraday Cage: Inspired by the characteristics of electromagnetic fields, *Faraday Cage*, an enclosure formed by conducting material or by a mesh of such material, is used to prevent the penetration of the exterior radio signals, thus protecting the cage's interior, say RFID tags. This idea has been commercialized, e.g., DIFRwear provides stylish clothing/wallet/accessories that block the reading of RFID. In fact, clever thieves are already known to use foil-lined bags in retail shops to circumvent shoplifting detection mechanisms.

Disabling and Killing: Karjoth and Moskowitz [140] proposed to physically clip tags at checkout, using perforated tear-off antennas. Tags remain functional, yet their range is effectively reduced to few centimeters. However, the applicability of this technology is limited to items with non-embedded tags. Another straightforward way for the protection of holder's privacy is to kill the tags before they are placed in the hands of consumers, which is also a standard function of EPC Gen2 tags. Each EPC Gen2 tag has a unique 16-bit password (obviously, it is too short to provide any resiliency against brute force attack), which is programmed at the time of manufacture. On receiving the correct password, the tag will deactivate itself forever. *Moisture-Dependent Contact* proposed in [104] is a special tag, which operates normally prior to sale. At the point of sale, a ROM component or wire of the tag is burnt by applying a large amount of power to the tag. Note that the tag is not completely killed but its RF interface is disabled. As a consequence, the tag cannot be skimmed in an uncontrolled environment as long as it stays dry, and can be finally re-enabled when the washing machine pumps water onto it.

However, there are many scenarios, for which simply kill or disable tags are unworkable or undesirable for privacy enforcement, such as effortless physical access control, theft-protection of belongings, and wireless cash cards. In addition, the deactivation of tags not only gives trouble to the shop-lifting of tagged items, but also destroys the item identification for the after sales services which are appreciated.

The Regulation: Garfinkel [90] proposed a voluntary framework for commercial deployment of RFID tags which may be helpful to improve the security and privacy of RFID systems. This frame work includes: (1) the right of the consumer to know what items possess RFID tags; (2) the right to have tags removed or deactivated upon purchase of these items; (3) the right of the consumer to access the data associated with an RFID tag; (4) the right to access of services without mandatory use of RFID tags; (5) the right to know when, where, and why the data in RFID tags is accessed.

Chapter 3

Physical Layer Enhancement of Passive RFID Communication

For a low-cost RFID system, a typical risk is the reader-tag communication via a radio channel is susceptible to *eavesdropping*, *modification* and *relay*. To mitigate, our work in this chapter presents a marked departure from the existing paradigm such as lightweight cryptography as we focus on defeating eavesdropping, modification and one particular type of relay attacks toward the tag-to-reader communication in passive RFID systems without requiring on-tag ciphers or secret credentials to be shared by legitimate parties. Our solution exploits the physical layer resources of passive RFID systems, i.e., backscatter modulation, uncoordinated frequency hopping and the coding for the wiretap channel, exhibiting a promising way to provide security functions while keeping the hardware cost of the reader and the tag almost unchanged, as expected in many RFID applications. To be specific, we present the following contributions:

1. We propose a novel physical layer scheme, called *Backscatter modulation- and Uncoordinated frequency hopping-assisted Physical Layer Enhancement* (BUPLE), for passive RF communication. The idea is to use the amplitude of the carrier wave to transmit messages as normal, while utilizing its periodically varied frequency to hide the transmission from the eavesdropper/relayer and exploiting a random sequence modulated to the carrier's phase to defeat malicious modifications. Our rigorous security analysis shows that BUPLE achieves desired security goals without affecting the cost of the reader and the passive tag.

2. BUPLE ensures that $\mathcal{A}$ receives a noisier signal than that of the legitimate reader, which presents a potential opportunity to further improve its eavesdropping resistance

through the coding in the physical layer. Three Wiretap Channel Codes (WCCs) with practical parameters for passive tags and with tradeoffs in the *information rate* (the proportion of the data-stream that is non-redundant), the *equivocation rate* (the degree to which the eavesdropper is confused) and the cost of implementation, are given – two of them are constructed from linear error correcting codes, and the third one is constructed from a resilient vector Boolean function.

3. BUPLE and the proposed WCCs are implemented on the software-defined radio platform (served as an RFID reader) and a programmable WISP tag. Results from our experimental data well support our theoretic hypothesis and security analysis. Additionally, performance comparison of the proposed WCC encoders with four lightweight ciphers from literature suggests that WCCs consume much less resource and have much higher throughput.

This chapter is organized as follows: Section 3.1 introduces adversary model, basic concepts and definitions. In Section 3.2, we present BUPLE and its security analysis. In Section 3.3, we give our constructions of the wiretap channel codes for passive tags. A prototype implementation and experimental results are shown in Section 3.4. We conclude the chapter in Section 3.5.

3.1 Preliminaries and Background

In this section, we briefly introduce the adversary model we rest on throughout of this chapter, gradients of passive RFID communication such as the backscatter modulation, uncoordinated frequency hopping. At last, we roughly introduce Wyner's wiretap channel.

3.1.1 Problem Statement and Security Model

Assuming that a powerful RFID reader shares a common RF channel with a passive tag which is computation- and storage-constrained, no secrets or authentication materials are shared by these two entities. We address the following problem: *how could confidentiality, authenticity and integrity of the tag-to-reader communication be preserved in the presence of a budget-limited adversary $\mathcal{A}$?* Here, by "confidentiality", we mean that given an eavesdropped version of the raw signal, to $\mathcal{A}$, the entropy of the message from the tag does not decrease. By "authenticity", we mean that the reader should be clear about who the sender of the message is. By "integrity", we mean that malicious modifications to the message can be detected by the reader. By "budget-limited", we mean that $\mathcal{A}$'s RF devices are effective in a narrow frequency band.

We assume that the two communicating entities are legitimate and are not compromised; otherwise, little can be done from the physical layer (issues caused by a malicious reader or an impersonated tag are beyond the scope of this chapter). We adopt a Dolev-Yao-alike model that $\mathcal{A}$ controls the communication which allows him to conduct the following actions:

- **Eavesdropping**: $\mathcal{A}$ intercepts tag-to-reader signals, demodulates and decodes to get communicated messages.

- **Modification**: $\mathcal{A}$ either adds to the channel a signal, which converts bit "0" into "1" (called *bit flipping* [47]), or adds to the channel a signal representing a bit string different from the one sent by the tag with a significantly higher power than that of the original signal (called *signal overshadowing* [47]). However, $\mathcal{A}$ is unable to eliminate energy from any channel.

- **Active Relay**: $\mathcal{A}$ places an active radio device in between a valid reader and a victim tag, e.g., [87], which generates new signals in a narrow frequency band to answer the valid reader according to the format of backscatter modulation after querying the victim tag.

Remark 1 *Note that, besides the category of active relay attack that $\mathcal{A}$ is able to cope with, there is another category of relay attack in the scenario of passive RFID systems, i.e., passive relay attack, such that a malicious passive tag wired with a malicious reader is placed in between the valid reader and the victim tag to relay the communication. Technically, this attack can be considered as a particular kind of tag impersonation, which violates our assumptions made to physical layer schemes thus is not considered here.*

3.1.2 Backscattering for Passive RF Communication

Radar principles tell us that the amount of energy reflected by an object is proportional to the reflective area of the object. A passive RFID system is principally a radar system in which the reader provides an RF signal for communication in both directions, i.e., from the reader to the tag and the tag to the reader. To be specific, we consider a passive tag composed of an antenna with impedance Z_a and a load with impedance Z_l. The impedance is often a complex quantity, where the real part is the resistance (i.e., R_a, R_l), and the imaginary part is the reactance (i.e., X_a, X_l). According to the *maximum power theorem* in RLC circuit theory [62], if the antenna's impedance is matched to that of the load (i.e., $R_a = R_l$), no reflection occurs at the interface. On the contrary, if the load is shorted, total reflection occurs and the power is re-radiated by the antenna. Thus by switching between the two states, a backscattered signal is in fact modulated by the Amplitude Shift Keying (ASK).

3.1.3 Availability of Uncoordinated Frequency Hopping in Passive RFID Systems

Frequency Hopping (FH) communication [199], in which the carrier frequency of a transmitted signal constantly changes according to a pre-shared pseudorandom sequence, was developed to defeat unintended listeners. *Uncoordinated Frequency Hopping* (UFH) indicates that two entities establish FH communication without sharing any secret. Strasser *et al.* in [195] considered applying UFH for fighting against a hostile jammer and proposed a hash-chain based pre-authentication scheme. However, implementing this probabilistic scheme is challenging, because: (1) the sender and receiver have less chance to “meet” in a particular channel at a certain time especially when the hopping set is large; (2) synchronization of the sender and the receiver is non-trivial when the hop rate is high, e.g., synchronization signals are vulnerable to jamming.

Nevertheless, we observed that UFH can be practically realized in passive RFID systems due to the following property: the reader changes the carrier frequency, while the tag only has to modulate responses on the carrier and reflect it without concerning which carrier frequency it uses. The reader is then able to center at the right frequency to capture the backscattered signal. Besides, the imperfect time synchronization, which is the main issue in a FH system, can be trivially solved, since the returned signal from the tag is strictly Δt second later than the emitted signal, where Δt is the sum of the tag's processing time and the signal's propagation delay in a small distance ($<$ 10m). Finally, FH mechanism is standardized in EPC Gen2 as an optional strategy to eliminate interference in dense reader scenarios and implemented in commercial products. In the light of UFH, our scheme brings confidentiality, authenticity and integrity to the tag-to-reader communication for free.

3.1.4 Wiretap Channel

The wiretap channel model, as shown in Figure 3.1, is introduced by Wyner [216] and extended in [180, 53]. In this model, when the main channel is better than the wiretap channel, i.e., $p_0 < p_w$, where p_0 and p_w are the error probabilities of the main channel and the wiretap channel respectively, it is possible through a particular coding to establish an (almost) perfectly secure source-destination link without relying on any pre-shared keys.

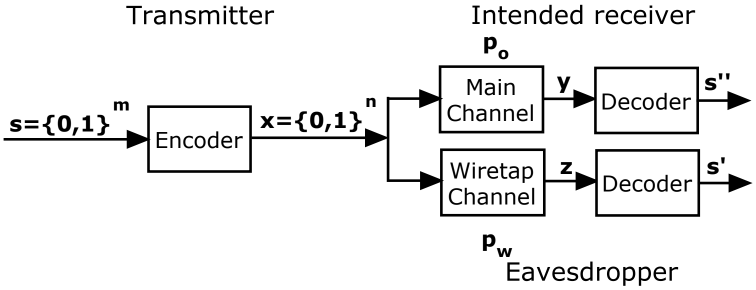

Figure 3.1: Wiretap Channel Model [216, 53]

As shown in Figure 3.1, to send an m-bit message $\mathbf{s} = (s_1, ..., s_m) \in \mathbb{F}_2^m$, the sender first encodes it into an n-bit codeword $\mathbf{x}$, which is then propagated through the main channel and wiretap channel simultaneously. The legitimate receiver, e.g., RFID reader, received a corrupted version $\mathbf{y} \in \mathbb{F}_2^n$ of $\mathbf{x}$ while the eavesdropper receives an even more strongly corrupted binary stream $\mathbf{z} \in \mathbb{F}_2^n$. After decoding, all information of $\mathbf{s}$ is expected to be leant by the legitimate receiver at a code rate as high as possible, while no information about $\mathbf{s}$ is leaked to the eavesdropper. Stated in another way, a wiretap channel has an *achievable secrecy* (R, L), $0 \leq R, L \leq 1$, if there is an encoder-decoder pair such that the following is true, for any $\eta > 0$,

$$\frac{1}{m}\text{Prob}[\mathbf{s} \neq \mathbf{s}''] \leq \eta, \quad \frac{m}{n} \geq R - \eta, \quad \Delta = \frac{H(\mathbf{s}|\mathbf{z})}{m} \geq L - \eta, \tag{3.1}$$

where Δ is the *equivocation rate* and $H(\mathbf{s}|\mathbf{z})$ is the *conditional entropy* of $\mathbf{s}$ given $\mathbf{z}$. Wyner exhibited the set of achievable (R, L) pairs always forms a region $\{(R, L) : 0 \leq R, L \leq 1, R \times L \leq h(p_w) - h(p_o)\}$, where $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the *binary entropy function* of p, and, $h(p_w) - h(p_o)$ is the *secrecy capacity* meaning the maximum product of the rate and equivocation rate of a code under which perfect secrecy can be achieved.

Although this model offers a potential opportunity to achieve Shannon's perfect secrecy (i.e., $\Delta = 1$) without a pre-shared key, two strong assumptions make it less appealing to practitioners: (1) two channels are distinct such that the main channel should be apparently better or less-noisy. This is difficult to realize in reality; (2) given p_o and p_w, there must exist a code satisfying Eq. (3.1) (we call such a code *Wiretap Channel Code* or (n, m)-WCC hereafter). Note that general constructions of WCCs, especially those with satisfactory information rate, equivocation rate and finite codeword length, remain an open problem [207].

As shown in Section 3.4 and Section 3.5, our work firstly closes the gap between this theoretic model and practice as: (1) UFH is exploited to significantly degrade the tag-to-eavesdropper channel by increasing p_w; and (2) three WCCs with small codeword length, targeting practical security, are given which can be implemented in tags with modest computation/storage capabilities.

3.2 BUPLE and Its Security

For the rest of the chapter, we keep the following notations.

- $\{f_1, ..., f_M\}$ represents a *hop set* with M possible frequencies.
- $W = \max(\{f_1, ..., f_M\}) - \min(\{f_1, ..., f_M\})$ is the *hopping band*.
- In one hop, t_h is the signal duration, called *hop duration*, (we ignore the transient *switching time* here for simplicity) and W_h is the bandwidth for each frequency channel.

- v_T is the tag's data rate, while v, $v \gg v_T$, is the rate of a random binary sequence generated by the reader, and v_{cmd} is the data rate of reader's commands.

- τ_0 is the power-up time in second for a tag.

3.2.1 BUPLE Scheme

The BUPLE scheme works as follows:

1. During the time interval $[it_h, (i+1)t_h)$, $i = 1, ..., n$, the reader emits a carrier wave CW_i modulated by Minimum Shift Keying (MSK)[1], i.e.,

$$CW_i = \sqrt{2E_b v_T} \cos\left(2\pi f_i t + b_{i,j} \frac{\pi v t}{2}\right),$$

where $f_i \in \{f_1, ..., f_M\}$ is randomly selected by the reader, $\sqrt{2E_b v_T}$ is a positive constant indicating the carrier's amplitude, and $b_{i,j} \in \{+1, -1\}$, $j = 1, ..., \lfloor t_h v \rfloor$, is randomly selected by the reader at the information rate v.

2. On this MSK-modulated carrier, the reader further amplitude-modulates its commands at rate v_{cmd} if necessary, e.g., QUERY as specified in EPC C1G2.

3. Once the tag powers up, it starts to amplitude-demodulate the double-modulated carrier to get the commands issued by the reader if there is any. The tag next computes a K-bit response $(r_1, ..., r_K)$ and backscatters "10" if $r_j = 1$ and "01" otherwise, at rate v_T, for $j = 1, .., K$.

4. The reader, with the receiver centered at f_i, receives the backscattered signal, which is denoted as $\widehat{CW}_i$. By amplitude-demodulation of $\widehat{CW}_i$ and further decoding "10" ("01" resp.) to "1" ("0" resp.), r_j is transmitted.

5. Above steps are repeated until the completion of the communication.

To exemplify our scheme, we present a toy instance in Figure 3.2 with $\tau_0 = 2/v$, $v = 3v_T$ and $v = 4v_{cmd}$, during the ith time slot. As shown, a random sequence "10101111...1101" is MSK-modulated to the carrier wave centered at f_i. Next, the reader's command "101" is amplitude-modulated on the carrier wave (thus on the random sequence). After receiving

[1] MSK is chosen because of its spectrum efficiency – power spectrum drops as the fourth power of frequency – and it provides constant energy to the tag.

the signal from the reader, the tag takes τ_0 second to power up and to process the reader's command. To respond with "10", the tag encodes "10" to "1001" and backscatters it. Note that the tag-to-reader message, i.e., "10", is now protected by BUPLE.

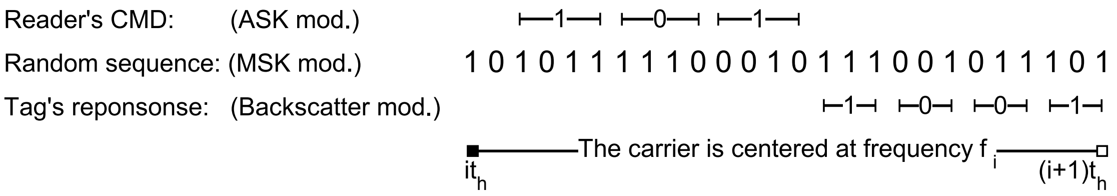
Reader's CMD: (ASK mod.)
1
0
1
Random sequence: (MSK mod.)
1 0 1 0 1 1 1 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 1 1 0 1
Tag's reponsonse: (Backscatter mod.)
1
0
0
1
The carrier is centered at frequency f i
it h
(i+1)t h

Figure 3.2: An Example of BUPLE with $\tau_0 = 2/v$, $v = 3v_T$ and $v = 4v_{cmd}$ (note that the message sent from the tag is actually "10")

Choose of Parameters: Choosing appropriate parameters for our scheme is crucial to realize the expected security properties. One typical configuration of BUPLE satisfying Part 15 of Title 47 of the Federal Communications Commission (FCC) regarding the spread spectrum system is:

- Total bandwidth $W = 100$MHz.
- Size of hop set $M = 200$.
- Bandwidth for each frequency channel $W_h = 500$kHz.
- Hop duration $t_h = 20\mu$s.

BUPLE-S vs. BUPLE-W: As a result of spreading the power of the signal to hide the transmission, a technical challenge arises: FH signals are usually unable to power up a passive tag – providing the power of FH signals is strong enough to power up a tag, it is also detectable by $\mathcal{A}$'s envelope detector (even $\mathcal{A}$ is unaware of the carrier's frequency). To address this problem, BUPLE takes different values of E_b, which leads to the following *two sub-schemes.*

- BUPLE-S ("S" for strong): E_b is a great positive float to the extent that CW_i provides enough power for passive tags to operate, i.e., $\int_0^{\tau_0} \sqrt{2E_b v_T} dt > V_{in}$, where V_{in} is the tag's minimum operating voltage, e.g., $V_{in} = 1.8$v for WISP v4.1 tags.

- BUPLE-W (“W” for weak): E_b has small numerical values such that CW_i is not detectable by the eavesdropper.

These two sub-schemes differ in several aspects as listed in Table 3.1: BUPLE-S provides more functionalities while BUPLE-W offers more security properties. For example, although BUPLE-W can neither power up tags nor issue commands, it has full resistance to eavesdropping in tag-to-reader communication when executed right after BUPLE-S. As confirmed by our experiments, few rounds of BUPLE-W could be executed immediately following one round execution of BUPLE-S. This is because the passive tag's capacitor stores constraint energy, which supplies the tag's circuit for a short while even without (enough) power supply from the reader. Depending on the design of upper protocols, BUPLE-S can be used independently, or with BUPLE-W alternatively.

Table 3.1: Functionalities v.s. Security Properties of BUPLE-S and BUPLE-W

	power-up tags	issue cmd	anti-modification	anti-eavesdropping	anti-relay
BUPLE-S	✓	✓	✓	limited	✓
BUPLE-W[a]	×	×	✓	✓	✓

[a] when BUPLE-W is executed right after BUPLE-S.

3.2.2 Security Analysis

Using the adversary model introduced, we have the following analytical results. Note that, while we utilize the bit error rate as the main metric to evaluate the attacker's performance in the analysis of the eavesdropping resistance of BUPLE-S, we adopt a computation-aided justification, as used in [111], in the analysis of the eavesdropping resistance of BUPLE-W. This is because it would be more meaningful, in the latter case, to investigate the possibility of eavesdropping for the attacker (in terms of its required SNR) as the BER becomes very close to 0.5 due to the low probability of interception of the signals in BUPLE-W.

Eavesdropping BUPLE-W: Generally speaking, the detection of FH signals is hard and all existed detectors exploit the known structure of signals [111], e.g., the hopping sequence is repeated after a short while. With the specified parameters, here we estimate the required *Signal-to-Noise Ratio* (SNR) to detect the presence of signals in BUPLE-W in terms of different types of FH detectors. Following the calculations in [199], given the probability of detection $P_D = 0.7$ and the probability of false alarm $P_{FA} = 10^{-6}$, we have:

(1) for a *wideband radiometer*, the required SNR at $\mathcal{A}$'s receiver is $SNR_{req} \approx 132$dB; (2) for a *partial-band filter bank combiner* (PB-FBC) with 50 branches, the required SNR for each channel $SNR_{req,I} \approx 128$dB; and (3) for an *optimum detector* with exact M branches, e.g., the legitimate reader, $SNR_{req} \approx 123$dB. This data suggests that $\mathcal{A}$'s wideband radiometer (PB-FBC resp.) has 9dB (4dB resp.) disadvantage relative to the optimum receiver owned by a legitimate reader. Thus, given the noise power spectrum in a specific environment, if E_b is carefully chosen, only the intended reader is able to receive messages backscattered by tags. Note that to enable a tractable computation, we actually assume: (1) the tag-to-eavesdropper channel is *Additive White Gaussian Noise* (AWGN); (2) $\{f_1, f_2, ..., f_M\}$, W, t_{msg}, M, t_h and W_h are public; and (3) $\mathcal{A}$ has exact knowledge of both the time at which a transmission originates and stops; otherwise, $\mathcal{A}$ has 1dB extra disadvantage [199].

Eavesdropping BUPLE-S: Although BUPLE-S offers a poor eavesdropping resistance, it does differentiate the tag-to-reader channel and the tag-to-eavesdropper channel in the sense that the error probability of the latter is enlarged. Let a backscattered signal be $\widehat{CW}_i = \sqrt{2E_{b,k}V_t}\cos(2\pi f_i t)$, if $k = 0$ or 1 is sent by the tag (ignore the MSK-modulated sequence for the time being). According to the minimum distance detection, the bit error probability for the tag-to-reader channel is:

$$p_o = Q\left(\frac{\sqrt{E_{b,1}} - \sqrt{E_{b,0}}}{\sqrt{N_o}}\right). \tag{3.2}$$

where Q is one minus the Gaussian cumulative distribution function.

Providing the eavesdropper listens at a wrong frequency, the received signal is passed through a band-pass filter, which leads a degradation, denoted as δ in dB, $\delta \leq 0$, to both $E_{b,0}$ and $E_{b,1}$, i.e., $E'_{b,0} = 10^{\delta/10}E_{b,0}$, $E'_{b,1} = 10^{\delta/10}E_{b,1}$. Thus the bit error probability for the tag-to-eavesdropper channel is:

$$p_w = Q\left(\frac{10^{\delta/20}(\sqrt{E_{b,1}} - \sqrt{E_{b,0}})}{\sqrt{N_o}}\right), \tag{3.3}$$

which is greater than p_o as Q is a decreasing function. Given an numerical example, let $E_{b,0} = 4$, $E_{b,1} = 25$, $\delta = -20$ and $N_o = 1$, we have $p_o = 0.0013$ for the intended receiver while $p_w = 0.3821$ for the eavesdropper.

Message Modification: First of all, the *signal overshadowing* is prevented: to inject a high amplitude signal to the channel, $\mathcal{A}$ has to know at which frequency the reader's receiver is working at; otherwise, the inserted signal will be filtered. In BUPLE, the attacker has $\frac{1}{M}$ chance to hit the right frequency. Transmitting the same message N times in different hops further decreases this probability to $\frac{1}{M^N}$, which is negligible when N is large[2]. Secondly, the *bit flipping* could be eliminated: in order to change "$r_j = 1$" to "$r_j = 0$", $\mathcal{A}$ needs to modify "10" to "01" in the channel (note that "00" or "11" are illegal codewords that help the reader to detect modification). To change the first bit in "10", $\mathcal{A}$ has to predict the shape of its carrier and sends the inverted signal to cancel it out. However, this is impossible since, besides the carrier frequency is unknown, the phase of the backscattered carrier is randomized by the MSK-modulated sequence and the channel condition is unpredictable as analyzed in [47].

Relay: In this case, $\mathcal{A}$ produces a well-formatted signal centered at f'_i carrying the relayed information to respond to the reader. The reader ignores this signal generated by the relayer with probability $1 - \frac{1}{M}$ since the reader's receiver always listens at f_i and filters out signals happening in other bands, where the probability, for $\mathcal{A}$, to have $f'_i = f_i$ is $\frac{1}{M}$. Multiple rounds of executions, say N, further decrease this probability to be negligible, i.e., $\frac{1}{M^N}$.

3.3 Enhanced BUPLE through Wiretap Channel Codes

As indicated by Eq. (3.2) and (3.3), if $\mathcal{A}$'s receiver tunes to a wrong frequency, a portion of energy of the backscattered signal is filtered and the demodulated and decoded bit streams are apparently noisier than those received by the intended receiver. Therefore, the wiretap channel model is realized by BUPLE. In this section, we further enhance BUPLE by considering *how could BUPLE-S achieve immunity to eavesdropping to the practical maximum extent possible?*

Our solution relies on the wiretap channel code. As shown in Figure 3.3, the tag's message is WCC-encoded before transmission and WCC-decoded by the reader launching BUPLE. Considering the moderate processing/storage capability of passive tags, we require a candidate WCC to have a equivocation rate close to 1 (rather than perfect secrecy), a relatively high information rate and a small codeword length n. In what follows, we assume both channels are *Binary Symmetric Channel* (BSC) with $p_o = 0$ and $p_w > 0$ for simplicity,

[2] There is a confliction that repeated transmissions impair the eavesdropping resistance. In reality, which security property is more important depends on upper layer protocols, e.g., modification resistance is more imperative to protocols in HB$^+$ family [176, 102].

otherwise a suitable error correction code can be employed to make $p_o = 0$ while keeping $p_w > 0$ (remember $p_w > p_o$). All "$\oplus$"s are addition operations in $\mathbb{F}_2$ unless otherwise stated and superscript T is the transpose of a vector.

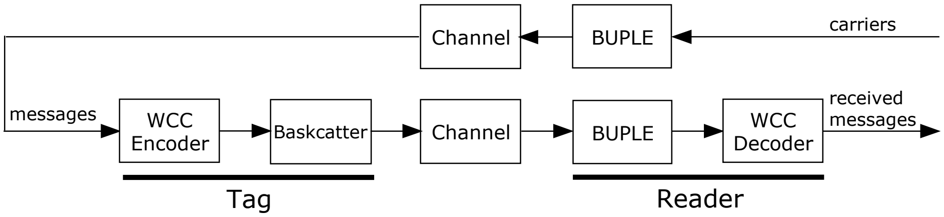

Figure 3.3: Enhanced BUPLE through Wiretap Channel Codes

3.3.1 Parameterized WCCs from Linear Error Correcting Codes

The *coset coding* based on linear error correcting codes with infinite codeword length was first used in Wyner's proof [216] of the existence of a secrecy-capacity-achieving WCC. Along this line, our first two constructions concentrate more on: (1) carefully selecting the underlying linear code to maximize the desired security with small n; and (2) designing of a storage efficient encoding algorithm, i.e., reducing the storage complex from $O(2^{2m})$ to $O(2^m)$. We thus have the following constructions.

Construction I: (8, 1)-WCC The encoder works as follows: to transmit $\mathbf{s} \in \{0,1\}$, the encoder outputs a random vector $\mathbf{x} = (x_1, ..., x_8) \in \mathbb{F}_2^8$ satisfying $x_1 \oplus x_2 \oplus ... \oplus x_8 = \mathbf{s}$. The decoder at the receiver's side evaluates $x_1 \oplus x_2 \oplus ... \oplus x_8$ (or $z_1 \oplus z_2 \oplus ... \oplus z_8$ for $\mathcal{A}$) to obtain $\mathbf{s}$ (or $\mathbf{s} \oplus \Sigma_{i=1}^{8} e_i$ resp.), where, as received by $\mathcal{A}$, $z_i = x_i \oplus e_i$ and e_i is an error bit introduced by the channel, i.e., Prob$\{e_i = 1\} = p_w$. Its rate, equivocation rate and $R \times L$ for different p_w are calculated and listed in Table 3.2. Similarly, we could construct a $(16, 1)$-WCC.

Construction II: (8, 4)-WCC Let $g(.) : \{0,1\}^4 \mapsto i$, $0 \leq i \leq 15$, be a public injective function and H be the parity check matrix of an $(8,4)$-extended hamming code $\mathcal{C}$, i.e.,

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Moreover, the *cosets* of $\mathcal{C}$ is denoted as C_i, $0 \leq i \leq 15$.

To transmit a 4-bit message $\mathbf{s}$, the encoder randomly selects a code $\mathbf{c} \in \mathcal{C}$ and XOR it with the coset leader $\mathbf{a}$ of $C_{g(\mathbf{s})}$ to produce $\mathbf{x}$. The decoder at the receiver's side evaluates $H\mathbf{x}^T$ (or $H\mathbf{z}^T = H(\mathbf{x} \oplus \mathbf{e})^T$ for $\mathcal{A}$) to obtain $H\mathbf{a}^T = \mathbf{s}$ (or $H(\mathbf{a} \oplus \mathbf{e})^T$ resp.). Here $H\mathbf{a}^T$ is called the *syndrome* of $\mathcal{C}$. In terms of implementation, this tag needs to store: (1) g of 64-bit; (2) $\mathcal{C}$ of (8×16)-bit; (3) coset leaders of (8×16)-bit; and (4) the syndromes of (16×4)-bit in the tag's memory. That is 384 bits in all. Its rate, equivocation rate and $R \times L$ for different p_w are calculate and listed in Table 3.2.

Security Analysis: It is intuitive that after decoding the noise-corrupted codeword $\mathbf{z} = (z_1, ..., z_n)$, where each z_i can be seen as a random binary variable, $\mathcal{A}$ is ignorant of $\mathbf{s} = (s_1, ..., s_m)$ if and only if the output of the decoder appears (almost) equally likely ranging from "$\underbrace{0...0}_{m}$" to "$\underbrace{1...1}_{m}$". This is achieved by the above WCCs because of the following reasoning.

Let $\mathbf{s} = (s_1, ..., s_m) \in \mathbb{F}_2^m$ be the message to be sent through a binary symmetric channel and let codewords in the dual of a linear code $\mathcal{C}$ have minimum distance d and let $wt(H_i)$ be the hamming weight of the ith row of the parity check matrix H of $\mathcal{C}$ (thus $wt(H_i) \geq d$). By leveraging the fact that: when the channel makes even number of errors, the adversary receives the correct parity check; when the channel makes odd number of errors, the adversary receives the wrong parity check. We have the following derivations:

$$\begin{aligned}
\text{Prob}\{s_1 = 0|\mathbf{z}\} &= \Sigma_{j \text{ even}}^{wt(H_1)} \binom{wt(H_1)}{j} p_w^j (1-p_w)^{wt(H_1)-j} = \frac{1}{2} + \frac{1}{2}(1-2p_w)^{wt(H_1)} \\
\text{Prob}\{s_1 = 1|\mathbf{z}\} &= \Sigma_{j \text{ odd}}^{wt(H_1)} \binom{wt(H_1)}{j} p_w^j (1-p_w)^{wt(H_1)-j} = \frac{1}{2} - \frac{1}{2}(1-2p_w)^{wt(H_1)} \\
\text{Prob}\{s_i = 0|s_1, ..., s_{i-1}, \mathbf{z}\} &= \frac{1}{2} \pm \frac{1}{2}(1-2p_w)^{wt(H_1 \oplus ... \oplus H_{i-1})} = \frac{1}{2} \pm \frac{1}{2}(1-2p_w)^d, i > 1 \\
\text{Prob}\{s_i = 1|s_1, ..., s_{i-1}, \mathbf{z}\} &= \frac{1}{2} \mp \frac{1}{2}(1-2p_w)^{wt(H_1 \oplus ... \oplus H_{i-1})} = \frac{1}{2} \mp \frac{1}{2}(1-2p_w)^d, i > 1
\end{aligned}$$

Furthermore, since $\text{Prob}\{\mathbf{s}|\mathbf{z}\} = \text{Prob}\{s_1|\mathbf{z}\} \times \prod_{i=2}^m \text{Prob}\{s_i|s_1, ..., s_{i-1}, \mathbf{z}\}$, the above WCCs achieve:

$$(\frac{1}{2} - \frac{1}{2}(1-2p_w)^d)^m \leq \text{Prob}\{\mathbf{s}|\mathbf{z}\} \leq (\frac{1}{2} + \frac{1}{2}(1-2p_w)^d)^m.$$

Therefore, the above WCCs have an achievable secrecy (R, L), as defined by Eq. (3.1), such that

$$R = \frac{m}{n}, \quad -\log_2(\frac{1}{2} + \frac{1}{2}(1 - 2p_w)^d) \leq L \leq 1.$$

3.3.2 WCCs Constructed from Resilient Boolean Functions

As we observed, the decoding process (e.g., $H(\mathbf{x} \oplus \mathbf{e})^T : \{0,1\}^n \mapsto \{0,1\}^m$ in Construction II) can be generalized as passing the noise-corrupted codeword through a well-designed S-box as shown below: when $(\mathbf{x} \oplus \mathbf{e})^T$ is not random as $p_w < 0.5$, the output of the S-box can be sufficiently random such that each output bit appears to be "0" and "1" (almost) equally likely. The tool of design for such an S-box is the *vector resilient Boolean functions*. A Boolean function with n-bit input, i.e., $x = (x_1, ...x_n)$, and m-bit output is said to be t-th order correlation immune if its output distribution does not change when at most t coordinates of x are kept constant [40]. It is called t-resilient if it is balanced and t-th order correlation-immune, that is the output distribution is uniform when at most t coordinates x_i of x are kept constant while others are chosen uniformly at random. From the information theoretic point of view, the resiliency ensures that, given $f(x)$, the information obtained about the values of t coordinates arbitrarily chosen, e.g., $x_{i_1}, ..., x_{i_t}$, is zero, that is $I(x_{i_1}, ..., x_{i_t} | f(x)) = 0$, where $I(X|Y)$ is the mutual information between random variables X and Y. Due the symmetry of mutual information, we also have

$$I(f(x)|x_{i_1}, ..., x_{i_t}) = 0.$$

Let us re-interpret the above equation in the context of the wiretap channel model: providing $x_{i_1}, ..., x_{i_t}$ are error-free bits and $x_{i_{t+1}}, ..., x_{i_m}$ are uniformly selected due to the channel noise, the eavesdropper is expected to learn nothing regarding $f(x)$ given x.

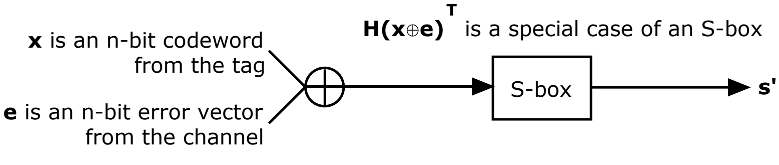
x is an n-bit codeword from the tag
e is an n-bit error vector from the channel
H(x⊕e)^T is a special case of an S-box
S-box
s'

Figure 3.4: Generalizing WCC Decoder as an S-box

In the following, we use Kerdock code studied in [198] as an example to construct a WCC.

Construction III: (16, 8)-WCC Let $\mathbf{x} = (x_1, ..., x_{16}) \in \mathbb{F}_2^{16}$, where $f(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_8(\mathbf{x})) =$

$$\begin{aligned}
&(x_9 \oplus (x_1 \oplus x_2 \oplus x_4 \oplus x_7 \oplus x_8 \oplus (x_1 \oplus x_5)(x_2 \oplus x_3 \oplus x_4 \oplus x_6) \oplus (x_2 \oplus x_3)(x_4 \oplus x_6)),\\
&x_{10} \oplus (x_2 \oplus x_3 \oplus x_5 \oplus x_1 \oplus x_8 \oplus (x_2 \oplus x_6)(x_3 \oplus x_4 \oplus x_5 \oplus x_7) \oplus (x_3 \oplus x_4)(x_5 \oplus x_7)),\\
&x_{11} \oplus (x_3 \oplus x_4 \oplus x_6 \oplus x_2 \oplus x_8 \oplus (x_3 \oplus x_7)(x_4 \oplus x_5 \oplus x_6 \oplus x_1) \oplus (x_4 \oplus x_5)(x_6 \oplus x_1)),\\
&x_{12} \oplus (x_4 \oplus x_5 \oplus x_7 \oplus x_3 \oplus x_8 \oplus (x_4 \oplus x_1)(x_5 \oplus x_6 \oplus x_7 \oplus x_2) \oplus (x_5 \oplus x_6)(x_7 \oplus x_2)),\\
&x_{13} \oplus (x_5 \oplus x_6 \oplus x_1 \oplus x_4 \oplus x_8 \oplus (x_5 \oplus x_2)(x_6 \oplus x_7 \oplus x_1 \oplus x_3) \oplus (x_6 \oplus x_7)(x_1 \oplus x_3)),\\
&x_{14} \oplus (x_6 \oplus x_7 \oplus x_2 \oplus x_5 \oplus x_8 \oplus (x_6 \oplus x_3)(x_7 \oplus x_1 \oplus x_2 \oplus x_4) \oplus (x_7 \oplus x_1)(x_2 \oplus x_4)),\\
&x_{15} \oplus (x_7 \oplus x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus (x_7 \oplus x_4)(x_1 \oplus x_2 \oplus x_3 \oplus x_5) \oplus (x_1 \oplus x_2)(x_3 \oplus x_5)),\\
&\Sigma_{i=1}^{16} x_i).
\end{aligned} \tag{3.4}$$

Let the encoder be $f^{-1}(\mathbf{x})$ and the decoder be $f(\mathbf{x})$. To transmit an 8-bit message $\mathbf{s}$, the encoder outputs a 16-bit random binary vector $\mathbf{x}$ such that $f(\mathbf{x}) = \mathbf{s}$. The decoder at the receiver's side simply evaluates $f(\mathbf{x})$ (or $f(\mathbf{x} \oplus \mathbf{e})$ for $\mathcal{A}$) given $\mathbf{x}$ (or $\mathbf{x} \oplus \mathbf{e}$ resp.) is received. This construction is optimum (among the three proposed WCCs) as its $R \times L$ is closest to the secrecy capacity as shown in Table 3.2 (see Appendix A for the brute force algorithm/program we developed as there is no known formula to compute the equivocation rate for WCC constructed from the nonlinear code).

Generally, an (n, m, t)-resilient Boolean function $f(.)$ can be used to construct an (n, m)-WCCs by letting the encoder be $f^{-1}(.)$ and the decoder be $f(.)$. However, due to the nonlinearity of the codes in the binary field, the method to derive the results for the achievable secrecy of linear WCCs, may not be applicable. The exploration of the analytic representation of the equivocation rate of WCCs from nonlinear codes constitutes a challenging work, which is part of our future research.

3.3.3 Visualize the Security of Proposed WCCs

We calculate the information rate, the exact equivocation rate and $R \times L$ of each WCC with different p_w, which are listed in Table 3.2. As seen, there is no one-size-fits-all WCC: Construction I is an extreme case when confidentiality is to be taken care of, with an imperative shortcoming in its lowest transmission rate; Construction II and Construction III are rate-efficient codes at the cost of lower equivocation rates.

To observe the real-world effects of the proposed WCCs, with Simulink, we built a digital communication system composing of a random message generator, a WCC encoder/decoder, an ASK modulator with 915MHz carrier, a BSC or AWGN channel and an

Table 3.2: Comparison of Performances of Proposed WCCs.

(n, m)	underlying code	rate	equivocation rate	$R \times L$
$p_w = 0.20$, secrecy capacity $= h(p_w) = 0.721928094887$				
(8, 1)	parity check	0.1250	0.99979649036	0.12497456129
(8, 4)	ext. hamming	0.5000	0.96977096204	0.48488548102
(16, 8)	Kerdock	0.5000	0.98711512719	0.49355756360
$p_w = 0.10$, secrecy capacity $= h(p_w) = 0.468995593589$				
(8, 1)	parity check	0.1250	0.97959953172	0.12244994146
(8, 4)	ext. hamming	0.5000	0.78495689709	0.39247844855
(16, 8)	Kerdock	0.5000	0.82311413681	0.41155706840
$p_w = 0.05$, secrecy capacity $= h(p_w) = 0.286396957116$				
(8, 1)	parity check	0.1250	0.86186434726	0.10773304341
(8, 4)	ext. hamming	0.5000	0.53233802320	0.26616901160
(16, 8)	Kerdock	0.5000	0.55356866398	0.27678433199

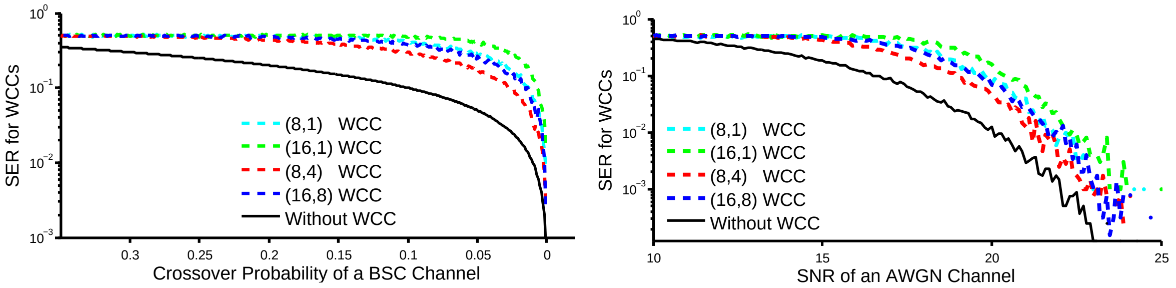

Figure 3.5: Simunlink Simulation for RFID Systems

envelope detector. The Symbol Error Rate (SER) is simulated and calculated to validate that WCCs further improves the eavesdropping resistance. As shown in Figure 3.5, the SER in BSC increases with p_w if no coding is involved (the given plots use a logarithmic scale for the y-axis). An interesting result is that the distance or resiliency of each WCC can be visualized as its maximum geometric distance away from the solid line. Besides, the plot of SER in AWGN on the right shows that the intended receiver has approximately 5dB advantage of SNR (relative to the eavesdropper) to achieve the same SER.

Table 3.3: Actual Measures of Output Voltages at Port `TX/RX` of `RFX900` w.r.t. Scale Factor

scale factor	output voltage	scale factor	output voltage
10	0.00mv	5000	2.124v
500	144mv	10000	2.880v
1000	396mv	25000	3.208v
2000	864mv	32767	3.312v

3.4 Proof-of-concept Implementation and Testing

In the following, we present our proof-of-concept implementation and testing of BUPLE and proposed WCCs.

3.4.1 Experiment Setup

We built a physical-layer programmable reader using the Universal Software Radio Peripherals (USRP-1) [76] together with two `RFX900` daughter boards (with the filters bypassed to get a 500mW peak output power): we use one `RFX900` with a `VERT900` antenna [211] to serve as the frontend of the transmitter (call them `RFX900-Tx` hereafter) and another `RFX900` with a circular polarity panel antenna [42] to be the frontend of a narrowband receiver (call them `RFX900-Rx` hereafter). In the receiving path, `RFX900-Rx` samples raw UHF signals by an ADC and then converts them to baseband signals by a digital down converter (DDC). The baseband digital signals out of USRP are sent via USB 2.0 interface to the Thinkpad T410 laptop running GNU Radio [92], a free software toolkit for signal processing from the physical layer, under the 32-bit Ubuntu 10.04. The transmission path is similar, but consists of digital up converters (DUC) and a DAC. In parallel to this, a `DPO7104` digital phosphor oscilloscope is used for measurements.

To observe behaviors of a passive tag, a WISP v4.1 tag [214], was employed. The reasons for the selection are: (1) it is programmable due to its 16-bit general purpose MSP430F2132 microcontroller. Programs for MSP430F2132 are written in embedded C and compiled, debugged and profiled with IAR Embedded Workbench 5.10.4, in conjunction with `TI FET430UIF` debugger; (2) it simulates every aspect of a passive tag in terms of limited and ephemeral energy storage and backscatter communication; (3) it implements a significant portion of EPC Gen2 commands, e.g., QUERY and QUERYREP.

In what follows, we use an integer called *scale factor* in $[-2^{16}+1, 2^{16}-1]$ to represent the amplitude of a signal without unit. The actual measures of the output voltages at port `TX/RX` of `RFX900` (without antenna) with respect to this scale factor is provided shown in Table 3.3.

3.4.2 Our Implementation

In our implementation, BUPLE-W and BUPLE-S are executed alternatively. We first developed a signal processing block for GNU Radio, in conjunction with our customized FPGA firmware, to generate a two leveled carrier signal with period 0.5s, where the high level of the amplitude 25000 represents BUPLE-S while the low level of the amplitude 3000 represents BUPLE-W (this amount, as we tested in an independent session, cannot drive the tag). In addition, our block randomly tunes the frequency of both `RFX900-Tx` and `RFX900-Rx` every 0.5s. Finally, we wrote a Python script to create and control *signal flow graphs*, in which, the gain of the receiver's antenna is set to 20dB, and the received signal is decimated by USRP with a factor of 32; right before demodulation, the decimated signals are again filtered by an 8th order low-pass filter with gain 2, cutoff frequency of 400KHz. Therefore a narrowband receiver is realized. Note that the specified hop rate cannot be implemented as there are many delays along the digitization path of USRP such as RF frontend settling time, FPGA FIFO filling time, USB transferring time, etc..

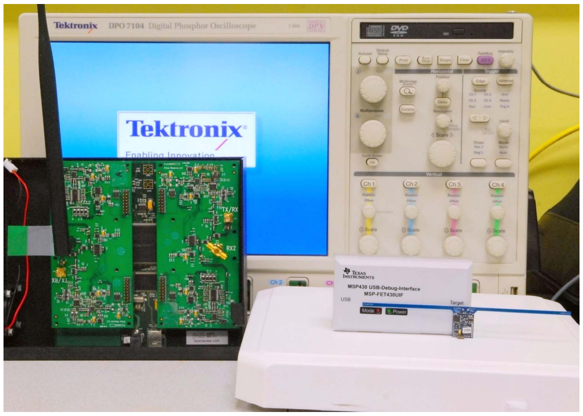
Tektronix
DPO7104 Digital Phosphor Oscilloscope
Tektronix
Enabling Innovation
MSP430 USB-Debug-Interface
MSP-FET430UIF

Figure 3.6: Devices Employed in Our Implementation and Testing: one `DPO7104` oscilloscope, one USRP (v1.0), two `RFX900` daughterboards, one `VERT900` antenna, one circular polarity panel antenna, one WISP tag (v4.1) and one `TI FET430UIF` debugger

For the tag side, we slightly modified the firmware of the WISP tag to let it intermittently answers "1010101010101010" at $v_T = 250$KHz followed each time by a sleep, when

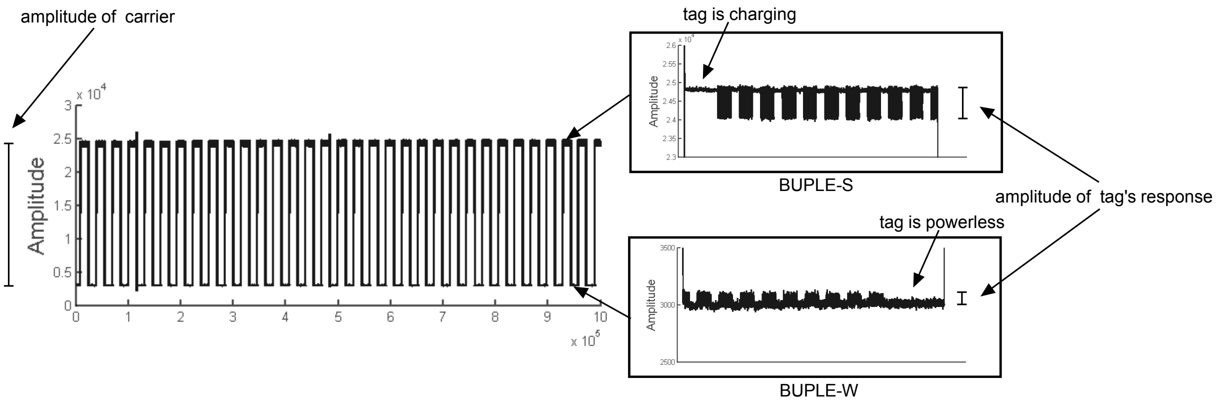

Figure 3.7: Time Domain Measurements when BUPLE Works with a WISP Tag

it has enough power, rather than implementing the command-based reader-tag interaction. This is because our physical layer scheme is essentially independent from upper layer protocols. By transmitting “1010101010101010”, we actually transmit a “1”, i.e., the tag encodes “1” as “11111111” with the $(8, 1)$-WCC, and each “1” in “11111111” is mapped to “10” as specified by BUPLE.

Figure 3.7 exhibits how our scheme works in a standard office setting with the tag placed in between the transceiver and receiver – it is 9.8cm away from `RFX900-Tx`’s antenna and 131cm away from `RFX900-Rx`’s antenna. As we can see, the backscatter communication carries out normally in BUPLE-S while it can only last for a while in BUPLE-W before the tag uses up its power. As long as the execution time of BUPLE-W is reduced, it is possible to keep the tag always alive.

Eavesdropping BUPLE Enhanced Communication: To further investigate the eavesdropper’s performance while BUPLE is running, we conducted the following tests in the same physical environment: centering `RFX900-Tx` at 915MHz while centering `RFX900-Rx` at frequencies ranging from 915MHz to 918MHz, we measured the amplitudes of the backscattered signals on BUPLE-S and BUPLE-W respectively, which are expected to exhibit the loss of communication reliability if the eavesdropper works at a wrong frequency.

We tabulated the results in Table 3.4. In both BUPLE-S and BUPLE-W, the carrier’s amplitudes as well as those of the tag’s responses drop quickly if the eavesdropper’s receiver is not centered at the right frequency. By “N/A”, we mean the signal is submerged in noise and cannot be observed. The experimental evidences support the theoretic hypothesis that to detect the presence of frequency hopped signals in BUPLE-W is non-trivial, let alone demodulate and decode them. We conducted this experiment for reader/tag/eavesdropper

Table 3.4: Amplitudes of Signals Captured by Eavesdropper Working at 915MHz to 918MHz

Rx's Freq.	BUPLE-S		BUPLE-W	
	amp. of carrier	amp. of tag's response	amp. of carrier	amp. of tag's response
915MHz	24700	564	2980	91
916MHz	6000	389	600	N/A
917MHz	4300	210	270	N/A
918MHz	300	N/A	200	N/A

with varying distances/angles and get the similar results. Note that, although, theoretically, the data in Table 3.4 should be able to fit onto the curve described by Eq. (3.3), it is practically hard to show as the degrading factor, i.e., δ, is a function of many variables and some of which are embedded in the hardware design of the USRP and the `RFX900-Rx` that are unknown to us.

Implementing On-tag WCC Encoders: To evaluate the cost of WCC encoders, we implemented them on the MSP430F2132 [205] of a WISP tag without WISP's firmware (since the firmware itself consumes a considerable portion of SRAM) and tested memory consumption and throughput. We employ a 23-stage LFSR with each stage in $\mathbb{F}_2^8$ as the random source for each WCC. To be mentioned, the encoding processes of $(8,1)$-WCC and $(8,4)$-WCC are implemented using pre-computed lookup tables while that of $(16,8)$-WCCs is computed on-the-fly by the underlying Boolean calculations. This is because when $n = 16$, the desired lookup table (of size 128KB) is far greater than the memory provided. To generate the code with maximal speed, we set the optimization level to be "high-speed" for the compiler. We then record the cycle counts through the FET debugger by letting the encoders execute at 8MHz on MSP430F2132 for 1000 times with random messages as inputs.

Table 3.5 summarizes the performance of WCC encoders, together with that of four lightweight ciphers implemented on the same or similar microcontroller platforms. Thanks to the simple operations, WCCs consume less resource and have higher throughput. The $(16,8)$-WCC encoder is resource-hungry because the pure embedded C code, as we used, is inefficient to process Boolean functions such as Eq. (3.4). Appropriate mixing of inline assembly code will allow the consumed resource be further decreased. Another noteworthy merit is that WCCs are more survivable in a frequent-loss-of-power environment since (1) they have the zero initialization time; and (2) they have a very small computation granularity, e.g., the only operation needed is a simple mapping from $\{0,1\}^m$ to $\{0,1\}^n$. On the contrary, an on-tag cipher, composing of many operations in series, is more likely to

Table 3.5: Performance Comparison of Proposed WCC Encoders and Lightweight Ciphers (note that PRESENT is implemented on a different-but-similar microcontroller platform – Atmel AVR ATmega163)

	SRAM [byte]	Flash [byte]	Initialization [cycle]	Throughput [bits/sec]
$(8, 1)$-WCC	690	0	0	740,936
$(8, 4)$-WCC	732	0	0	621,346
$(16, 8)$-WCC	1,348	0	0	86,776
Hummingbird [83]	$1,064$	0	$9,667$	$53,024$
AES [153]	13,448	92	1,745	199,377
KASUMI [153]	9,541	64	1,381	90,395
PRESENT [29]	$2,398$	528	–	$53,361$

be interrupted. In all, together with Table 3.2, we found that $(8, 4)$-WCC makes the information rate, the security and the implementation cost well-balanced, which is a favorable choice for practitioners.

3.5 Conclusion

Given the likely importance of RFID technology in practice, security and privacy problems should be solved before worldwide deployment. In this chapter, we propose to enhance the physical layer of the passive RFID communication. The security and usability are further confirmed by our implementations and testing results. Through the BUPLE scheme and proposed WCCs, a confidentiality-, authenticity- and integrity-preserving channel is created for tag-to-reader communication. It is also worth emphasizing that our solutions are designed for, but not limited to passive RFID systems, e.g., it is applicable to the backscatter wireless sensor network, e.g., [213], for establishing secret communication.

Chapter 4

Active Eavesdropping Attacks and Countermeasures

Passive RFID technology enables automatically and contactlessly identification of physical objects around ten meters away without requiring line-of-sight. This catching and exclusive characteristic not only satisfies the needs of considerable applications, e.g., logistic/asset management, human/animal identification, but also gives birth to new forms of applications such as RFID-based localization [224, 174]. Nevertheless, the large operation range of passive RFID systems and the ubiquitous deployment of passive tags introduce growing security and privacy concerns regarding the possible release of the bearer's information. Some of the attacks identified in [151, 125, 135] are *tag skimming*, *tag tracking*, *tag cloning* and *mafia-fraud* to bypass tag authentication/anti-counterfeiting mechanisms. Thus far, to launch these attacks, *eavesdropping* the communication between the legitimate reader and the victim tag to obtain raw data is a basic tool for the adversary. As an example, to clone an RFID passport, the adversary always: (1) eavesdrops several legitimate communication sessions and interrogates the victim tag in another several sessions; (2) recovers the tag's secret credentials by breaking the underlying cryptographic protocol/encryption algorithm if there is any; and, (3) burns the victim tag's identities and the learnt secret credentials to a counterfeited tag.

However, given the fundamentality of eavesdropping attacks, there are limited prior work investigating its intension and extension for passive RFID systems. One possible reason is that the stereotyped thinking patterns lead people to believe that eavesdropping can be simply solved by encryption and decryption. However, while they are indeed sound paradigms to protect majority wireless communication systems from eavesdroppers, they may fail to work for passive RFID systems because of the following considerations:

- Due to the modest computation/storage capabilities and the necessity to keep its prices low, passive tags are unlikely to perform even symmetric encryptions, especially when they are distant away from the reader's electromagnetic field. Although dedicated designs of lightweight block/stream ciphers are under development, the practical security they can offer is not very clear to date, e.g., [46, 100, 106, 171].

- Encrypting plaintext before transmission introduces the key management – key generation/distribution/storage/revoking/updating, which seems overkill for low-cost tags, e.g., once a tag is disposed, the related key information is unnecessarily stored/maintained.

- Providing a tag responds a reader with symmetric-encrypted data, the reader has to go through the entire key set to find a valid key for authentication and decryption, which, known as *key search problem* [125], results in a poor scalability. Considering the facts above, the de facto standard of passive RFID systems, EPC Gen2, does not include encryption/decryption as a part.

Therefore, *eavesdropping attack in this specific scenario seems like an animal with no natural enemies*, and better understanding of how a smart eavesdropper works and how powerful he could be is quite necessary.

In this chapter, we investigate eavesdropping attacks for passive RFID systems where encryption/decryption is unavailable. We constrain ourselves to the tag-to-reader communication (or the *backward channel*), since a reader-to-tag communication (or the *forward channel*) can be viewed as a conventional broadcast channel which is more well-understood. To be specific,

1. We identified a brand-new and quite powerful family of attacks, called *Unidirectional Active Eavesdropping*, which defeats the customary impression that eavesdropping is a "passive" attack. During the active eavesdropping attack, the adversary transmits an un-modulated carrier (call it *blank carrier* hereafter) at a certain frequency $f_{\mathcal{E}}$, while a valid reader and a tag are talking at another frequency channel $f, f \neq f_{\mathcal{E}}$. When the tag modulates the amplitude of reader's signal, it causes fluctuations on the blank carrier as well. By carefully examining the amplitude of the backscattered version of both blank carrier and reader's carrier, the eavesdropper is able to recognize tag's responses more clearly, relative to that of a conventional passive eavesdropping attack.

2. We set out to fill the literature's gap by demonstrating and empirically evaluation of the active eavesdropping towards an EPC Gen2-compliant system through software-

defined radio working at 860-960MHz and a programmable passive tag. Our experimental results show a significant improvement in the bit error rate (BER) of the intercepted communication as long as active eavesdropping is utilized. As an additional effort, we also present several strategies for the eavesdropper to make this attack even more powerful and undetectable.

3. The active eavesdropping attack is not trivial to be prohibited as it arises from the nature of backscatter communication. For a particular type of active eavesdropper, namely a *greedy proactive eavesdropper*, we propose a simple countermeasure without introducing any computation/storage overhead to the current system. The basic idea is: during a normal interaction, the reader stops emitting the carrier for a short while, and, if the tag always stays awake, the reader may suspect the existence of a greedy proactive eavesdropper nearby. This simple scheme could be helpful in the sense that, to avoid been detected, the adversary has to: (1) carefully control emitted power to be weak enough; and/or (2) adopt other strategies in the time/frequency domain; and/or (3) switch back to be passive eavesdropping, and lose all the gains in the active eavesdropping. At last, we also empirically verified this scheme using the same platform as presented.

The rest of this chapter is structured as follows. Section 4.1 introduces preliminaries and background regarding passive RFID communication and briefly summarizes the related work along this line. In Section 4.2, we formalize the active eavesdropping attack and provide theoretic analysis. In Section 4.3, we exhibit the experimental verification of this attack. A partial countermeasure is presented in Section 4.4. Section 4.5 concludes this chapter.

4.1 Preliminaries and Background

Before an in-depth discussion of the eavesdropping problem, we present the system model and attacker model used throughout rest of this chapter, and introduce the backscattered communication for passive RFID systems in more detail, which form the basis of the active eavesdropping attack. At last, we review the related work.

4.1.1 System Model and Adversary Model

System Model: Our system encompasses three roles: a legitimate reader $\mathcal{R}$, a victim tag $\mathcal{T}$ and a computationally-bounded adversary $\mathcal{E}$. The mutual distance between the tag and the reader (adversary resp.) is $D_{\mathcal{T},\mathcal{R}}$ ($D_{\mathcal{T},\mathcal{E}}$ resp.). We assume that $\mathcal{R}$ and $\mathcal{T}$ with public configuration parameters involved in the RF communication do trust each other and are not compromised. In addition, $\mathcal{R}$ and $\mathcal{T}$ communicate over a frequency f choosing from a set $\{f_1, ..., f_K\}$, e.g., 902MHz $\leq f_1 < f_2 < ... < f_K \leq$ 928MHz for EPC Gen2 in North America. Since there is only a single tag in our system, the signal collision caused by multiple and simultaneous responses from different tags are ignored.

Adversary Model: The goal of $\mathcal{E}$ is to acquire or intercept tag-to-reader communication, i.e., $\mathcal{E}$ receives, demodulates and decodes analog RF signals backscattered by $\mathcal{T}$ by using its RF receiver (call it $Rx_{\mathcal{E}}$ hereafter) working at a proper frequency. Additionally, $\mathcal{E}$ is free to "actively" transmit any well-designed signals (if necessary) by using its RF transmitter (call it $Tx_{\mathcal{E}}$ hereafter). Note that there could be more than one $Rx_{\mathcal{E}}$ and more than one $Tx_{\mathcal{E}}$, which are not necessary to situate in one physical location. That is saying, the adversary deploys/distributes (a set of) $Rx_{\mathcal{E}}$ and (a set of) $Tx_{\mathcal{E}}$ at his will, while all of these devices can be centrally coordinated.

However, unlike a Dolev-Yao attacker, $\mathcal{E}$ cannot control the communication channel between $\mathcal{R}$ and $\mathcal{T}$, i.e., $\mathcal{E}$ cannot insert and remove messages to/from the tag-to-reader communication channel and $\mathcal{E}$ cannot relay the tag-to-reader communication. Moreover, we assume that, $\mathcal{E}$ desires to keep the entire eavesdropping procedure undetected by both $\mathcal{T}$ and $\mathcal{R}$.

4.1.2 Backscattering Communication

As mentioned in the previous chapter, passive RFID system is principally a radar system in which the reader provides the RF signal for communications in both directions. To be formal, the reader at first broadcasts an amplitude-modulated carrier, denoted as,

$$CW(t) = A(t)\cos(2\pi f t + \theta),$$

where f is the carrier frequency in $\{f_1, ..., f_K\}$ and θ is a constant phase of the carrier. $A(t)$, constituting a high level and a low level, carries the binary command to be issued to the tags, e.g., "1000001000000111110110" for the QUERY command as specified in [78]. Once a command is propagated, the reader keeps $A(t)$ at the high level, e.g., "111....", expecting the responses from tags.

The tag, after receiving the operating energy from $CW(t)$, uses an envelope detector to obtain and decode the command in $A(t)$ if there is any. To respond, the tag maps the source message bits into baseband codewords using a Manchester-alike-code to enable the collision detection on the reader's side. For example, FM-0 code, as specified in [78], uses two bits $(0, 1)$ and $(1, 0)$ alternatively to represent a "0" bit, and uses $(0, 0)$ and $(1, 1)$ alternatively to represent an "1" bit. Instead, if Miller-4 code is employed, the tag utilizes $(0, 1, 0, 1, 0, 1, 0, 1)$ and $(1, 0, 1, 0, 1, 0, 1, 0)$ alternatively for the transmission of "0", $(1, 0, 1, 0, 0, 1, 0, 1)$ and $(0, 1, 0, 1, 1, 0, 1, 0)$ alternatively for the transmission of "1". It is intuitive that the latter code provides better error-tolerance at the cost of lower code rate.

In order to backscatter the encoded response, the tag next switches the reflection coefficients by changing its antenna impedance within two states (Δ_0, Δ_1), $0.5 < \Delta_0 < \Delta_1 < 1$, at a given rate. The backscattered signal can be represented as:

$$BCW(t) = \Delta_i \sqrt{2E_{\mathcal{T}} v_{\mathcal{T}}}\cos(2\pi f t + \theta), \quad i = 0, 1,$$

where $E_{\mathcal{T}}$ is the energy per bit presented at the tag's antenna when $CW(t)$ with $A(t)$ at the high level is transmitted by the reader, $v_{\mathcal{T}}$ is the tag's data rate and $\Delta_1 > \Delta_0$ implying more power is backscattered when transmitting a "1" of the codeword (thus, less is absorbed by the tag) and less is reflected otherwise. The reader, centering its receiver at f, coherently detects the responses by correlating the amplitude of $BCW(t)$ to potential codewords and comparing the resulting correlations with a particular threshold, which allows the establish of the backscatter communication.

4.1.3 Related Work

Related Attacks: Hancke in [109] experimentally confirmed that two NFC standards, namely, ISO 14443A/B and ISO 15693, where the designed operational range is less than

10cm, are eavesdroppable even the attacker is 3.5m away. Dobkin in [63] reported testing results of intercepting an EPC Gen2 tag's reply approximately 7m away in an office environment, which indicates that, although it is more difficult to intercept and interpret the tag's backscattered signal as it is weaker than the reader's signal, e.g., around 0 to −20dBm, it is by no means impossible. Our work follows Dobkin's preliminary research and discloses more. Recently, Koscher *et al.* in [136] particularly examined the security, under skimming attack, towards the United States Passport Card and Washington State enhanced drivers license, both of which incorporate EPC Gen2 tags. In their demonstrated attacks, the maximum distance a tag can be read by a rogue reader radiating 36dBm power is measured. However, skimming is different from eavesdropping in the sense that the skimmer does provide energy to the tag through the carriers it propagates. Thus, testing results on the skimming does not provide convincible results on the eavesdropping.

Related Countermeasures: In terms of cryptographic countermeasures, the passive tags' modest capabilities drive much research focused on the design and implementations of lightweight ciphers as summarized in Chapter 2. To construct an unidirectional confidentiality-preserving channel using a physical layer approach, cooperative-jamming methods are proposed in [126, 44, 18, 200, 50, 19] as summarized in Chapter 2.3.2.

In addition, since an EPC Gen2 tag is crypto-free and only supports simple operations, NIST recommends in [135] to use a simple mechanism, called *cover coding*, to hide the bidirectional transmission by assuming that the tag-to-reader channel is always much better than the tag-to-adversary channel. Cover coding works as follows: the tag generates a random number serving as the "keystream" and returns it to the reader. Since the tag-to-adversary channel is quite noisy, the adversary may be ignorant of this keystream; the reader produces "ciphertext" by XORing the received keystream and the plaintext to be sent. As can be seen, even the assumption is hold, the effectiveness of cover-coding completely depends on the security of the tag's PRNG, which is not prepared by EPC Gen2 tags; and this scheme, even it works for the conventional passive eavesdropping attack, can be completely broken by our new technology, as the tag's response can be amplified at the attacker's will.

It is worth to mention that our physical layer enhancement in Chapter 3 is designed to fight against a passive eavesdropper, and, unsurprisingly, it is partially suffers from this attack – the frequency hopping in BUPLE becomes less powerful since the active eavesdropping works irrespective of the frequency that a tag and a reader rest on.

4.2 Unidirectional Eavesdropping: from Passive to Active

We introduce our novel concepts in eavesdropping by providing theoretic analysis of both passive eavesdropping and active eavesdropping in this section. To produce meaningful and quantitative results, we make use of BER as the main metric to evaluate the reliability of the communication channel.

4.2.1 Passive Eavesdropping

Let us first consider the BER of an RF receiver in general. In the current setting, this RF receiver is a passive eavesdropper. However, the results derived here are also applicable to the case where the RF receiver is the legitimate RFID reader itself.

Assume the energy per bit at $\mathcal{R}$'s antenna is $E_{\mathcal{R}}$, the backscattered signal received by the eavesdropper is, if a bit $i \in \{0,1\}$ is sent,

$$\begin{aligned} BCW_{\mathcal{E}}(t) &= \Delta_i \sqrt{2E_{\mathcal{R}} \upsilon_{\mathcal{T}}} \cos(2\pi f t + \theta) \\ &= \Delta_i \sqrt{2E_{\mathcal{T}} \upsilon_{\mathcal{T}} \times \frac{\eta}{D^2_{\mathcal{T},\mathcal{E}}}} \cos(2\pi f t + \theta) \end{aligned}$$

The last equality is from the Friis transmission equation, i.e.,

$$\frac{E_{\mathcal{R}}}{E_{\mathcal{T}}} = \frac{\eta}{D^2_{\mathcal{T},\mathcal{E}}},$$

where η is a constant proportional to the antenna gains of both parties and the square of the wavelength. Unsurprisingly, the attacker could keep employing antennas with higher gains to compensate for the loss of η because of the increase of $D_{\mathcal{T},\mathcal{E}}$. However, we treat η as a constant to simply the analysis.

Hence, according to the minimum distance detection method, the BER of the backscatter modulation, the main metric for the reliability of the eavesdropper, is given by

$$\begin{aligned} p_{\mathcal{E}} &= Q\left(\frac{\Delta_1\sqrt{E_{\mathcal{T}} \times \frac{\eta}{D^2_{\mathcal{T},\mathcal{E}}}} - \Delta_0\sqrt{E_{\mathcal{T}} \times \frac{\eta}{D^2_{\mathcal{T},\mathcal{E}}}}}{\sqrt{N_o}}\right) \qquad (4.1) \\ &= Q\left(\frac{(\Delta_1 - \Delta_0)\sqrt{\eta E_{\mathcal{T}}/N_o}}{D_{\mathcal{T},E}}\right), \end{aligned}$$

where Q is the one minus the cumulative distribution function of the standardized normal random variable and N_o is the noise power density in the channel between the tag and the eavesdropper's receiver. As can be seen, as long as Δ_0, Δ_1, η, $E_{\mathcal{T}}$ and N_o are given and fixed, the errors in the intercepted messages grows rapidly when $D_{\mathcal{T},\mathcal{E}}$ increases. This observation suggests that the greater the distance between the tag and the passive eavesdropper, the less reliable the intercepted communication is, which is quite nature.

4.2.2 Active Eavesdropping

To combat the loss of reliability, a strategic adversary may consider *active eavesdropping*, which is effective towards the backscatter communication. Through the sequel, we formalize this attack and demonstrate how the eavesdropper obtains a better BER performance relative to the conventional eavesdropping, when the victim tag is far away.

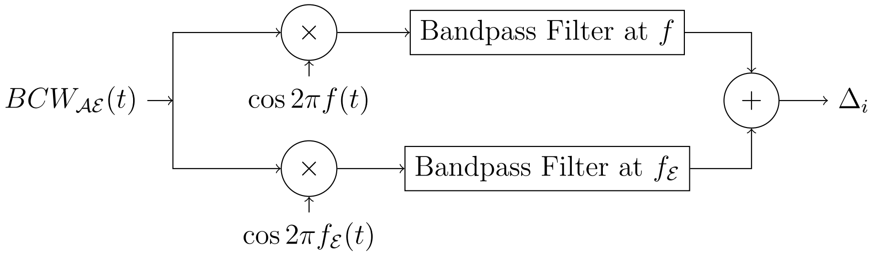

Figure 4.1: Designed Receiver for Active Eavesdropping

In addition to the aforementioned RF receiver, the eavesdropper, denoted as $\mathcal{AE}$, has an RF transmitter working at $f_{\mathcal{E}}$, $f_{\mathcal{E}} \neq f$[1], which produces a *blank carrier* while a valid reader-tag communication is ongoing, i.e.,

$$CW_{\mathcal{AE}} = \sqrt{2E_{\mathcal{AE}}v_T}\cos(2\pi f_{\mathcal{E}}t + \theta')$$

where $E_{\mathcal{AE}}$ is a constant to make the amplitude of the blank carrier (presented at the tag's antenna), i.e., $\sqrt{2E'_{\mathcal{T}}v_{\mathcal{T}}}$, suitable. When the tag amplitude-modulates reader's signal as aforementioned, it causes fluctuations on the blank carrier as well, which can be written

[1] It is discussed in Section 4 that $f_{\mathcal{E}} = f$ results in jamming of a legitimate reader-tag communication, which is in general not wanted by the eavesdroppers.

as:

$$\begin{aligned} BCW_{\mathcal{AE}}(t) &= \Delta_i(\sqrt{2E_{\mathcal{T}}v_{\mathcal{T}} \times \frac{\eta}{D^2_{\mathcal{T},\mathcal{AE}}}} \cos(2\pi ft + \theta) \\ &\quad + \sqrt{2E'_{\mathcal{T}}v_{\mathcal{T}} \times \frac{\eta(f_{\mathcal{E}})}{D^2_{\mathcal{T},\mathcal{AE}}}} \cos(2\pi f_{\mathcal{E}}t + \theta')), \end{aligned}$$

where $E'_{\mathcal{T}}$ is the energy per bit at $\mathcal{T}$'s antenna resulted by the blank carrier. It is worth to mention that $\eta(f_{\mathcal{E}})$ is no longer a constant as, even the eavesdropper could keep his gain at a constant level (by switching to a proper antenna), the tag's antenna is unlikely to maintain the same gain under different $f_{\mathcal{E}}$. For example, if $f_{\mathcal{E}}$ is totally out of the effective region of the tag's antenna, e.g., $f_{\mathcal{E}} = 10$KHz, $\eta(f_{\mathcal{E}})$ decreases to 0. Different tags may result in different $\eta(f_{\mathcal{E}})$, which can be obtained through the VSWR graph[2] of its antenna. A crucial observation here is that, for most of the antennas, although optimized for a particular frequency band, e.g., 902-928MHz, the gain degrades slowly if $f_{\mathcal{E}}$ is not too far away from this designed band, e.g., $860 \leq f_{\mathcal{E}} \leq 960$MHz. Principally, if the tag has a quadband UHF antenna, we could even actively eavesdrop it with $f_{\mathcal{E}} = 1.8$GHz. In the rest, we say $f_{\mathcal{E}}$ is *effective* iff $\eta(f_{\mathcal{E}})$ is comparable with $\eta(f), f \in \{f_1, f_2, ..., f_K\}$.

After $BCW_{\mathcal{AE}}(t)$ is intercepted, it is passed through two filters centered at f and $f_{\mathcal{E}}$ respectively as shown in Figure 4.1. The resulting baseband signals are then added up. From the signal constellation's point of view, the eavesdropper obtains a constellation of two points (representing signals for bit "1" and bit "0" respectively), that are separated by a minimum distance of

$$(\Delta_1 - \Delta_0)(\sqrt{E_{\mathcal{T}} \times \frac{\eta}{D^2_{\mathcal{T},\mathcal{AE}}}} + \sqrt{E'_{\mathcal{T}} \times \frac{\eta(f_{\mathcal{E}})}{D^2_{\mathcal{T},\mathcal{AE}}}}).$$

4.2.3 Reliability of Active Eavesdropping

In parallel to Eq. (4.1), the active eavesdropper's BER performance is

$$p_{\mathcal{AE}} = Q\left(\frac{(\Delta_1 - \Delta_0)(\sqrt{\eta E_{\mathcal{T}}/N_o} + \sqrt{\eta(f_{\mathcal{E}})E'_{\mathcal{T}}/N_o})}{D_{\mathcal{T},\mathcal{AE}}}\right), \tag{4.2}$$

[2] VSWR stands for Voltage Standing Wave Ratio. It is the ratio of the maximum/minimum values of standing wave pattern along a transmission line to which a load is connected. VSWR value ranges from 1 for a matched load to infinity for a short or an open load.

which suggests that, for an effective $f_{\mathcal{E}}$, the eavesdropper is always able to tune $E'_{\mathcal{T}}$ to get a suitable $p_{\mathcal{AE}}$. Here an numeric example is plotted in Figure 4.2 for the purpose of illustration, where $\Delta_1 - \Delta_0 = 0.2$, $\sqrt{\eta} = \sqrt{\eta(f_{\mathcal{E}})} = 10$, $E_{\mathcal{T}}/N_o = 1$dB, $E'_{\mathcal{T}}/N_o = 3$dB and $D_{\mathcal{T},\mathcal{AE}}$ varies. As shown, emitting a blank carrier with an SNR = 3dB is more than enough to compensate for the loss of reliability when the eavesdropper stands additionally 5m away from the tag.

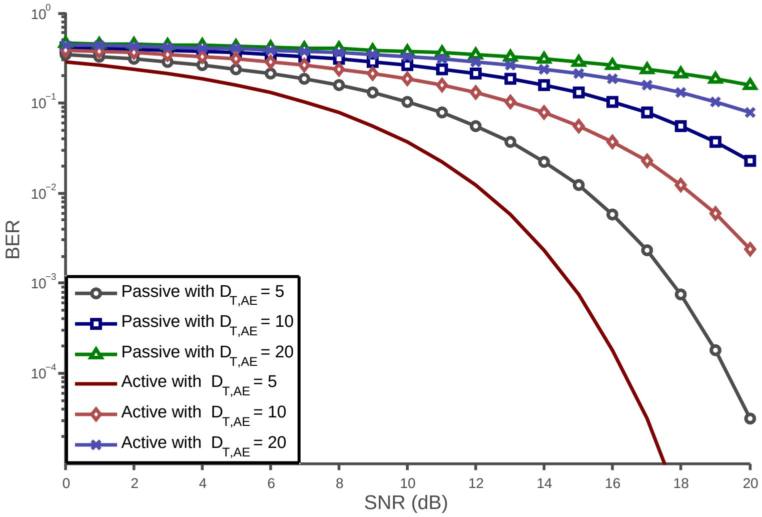

Figure 4.2: An Numeric Example Comparing BER Performances of Passive and Active Eavesdropping Attacks (note that $\Delta_1 - \Delta_0 = 0.2$, $\sqrt{\eta} = \sqrt{\eta(f_{\mathcal{E}})} = 10$, $E_{\mathcal{T}}/N_o = 1$dB, $E_{\mathcal{AE}}/N_o = 3$dB and $D_{\mathcal{T},\mathcal{AE}}$ varies from 5m to 20m)

4.3 Implementation and Testing of Active Eavesdropping Attack

In this section, we report on the design and implementation of our prototype system and the testing results to validate our concept of unidirectional active eavesdropping.

4.3.1 System Design

We developed a prototype system, as shown in Figure 4.3, that encompasses four roles: one legitimate reader, one legitimate tag, one $Rx_{\mathcal{E}}$ for the eavesdropper and one $Tx_{\mathcal{E}}$ for the eavesdropper. Note that our system evaluates two basic topologies as shown – in scenario I, all components are located in a straight line; in scenario II, the reader and the tag reside on the y-axis while $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$ are located on the x-axis. In what follows, we detail the implementation of each component of this system.

Gradients from Software Defined Radio: To enable the flexibility, the concept of Software-Defined Radio (SDR) is employed to build up programmable transmitters/receivers working at UHF band on an inexpensive-but-flexible platform, namely, the *Universal Software Radio Peripheral* (USRP). Roughly speaking, the USRP, in conjunction with the daughterboard `RFX900`[3] it carries on, constitutes a low-cost RF transceiver working at $800-1000$MHz with freely available schematics and drivers. To be specific, in the receiving path, `RFX900` receives raw UHF signals and converts them to the intermediate frequency (IF) band and passes them to the USRP board, which further samples and converts them to baseband signals by an analog-to-digital converter (ADC) and a digital-down-converter (DDC). The baseband digital signals out of USRP are sent via USB 2.0 (for USRP-1 [76]) or ethernet cable (for USRP-2 [77]) to our laptop running GNU Radio [92], a free software toolkit that provides the signal processing runtime and processing blocks for digital communications. Similarly, when transmitting, discrete signals are produced by GNU Radio and delivered to the USRP. The later up-converts them to the IF and UHF band and propagates them via `RFX900` and its antenna. In our implementations, two models of USRP were employed, i.e., USRP-1 and USRP-N210.

RFID Reader: We used an USRP-N210, in conjunction with one `RFX900` carrying two VERT900 dipole antennas [211], to play the role of the legitimate reader. We developed a simple Python script to create and control a signal flow graph to enable this reader: (1) querying a tag by propagating a constant sine wave working at 915MHz with maximum possible power that does not result in RF clipping, i.e., 23dBm, through one dipole antenna; and (2) collecting the reflected signal through another dipole antennas. The script runs on a MacBook MC516LL with OS X 10.4 and communicates with the USRP-N210 using the *Universal Software Radio Peripheral hardware driver* [210] (UHD v003.20110217015719) provided by Ettus[4]. Note that, we intentionally keep this reader as functionally simple as possible, since, in this work, we do not care much about how the reader interacts with a tag at the logic layer.

Eavesdropper's Transmitter: This transmitting part is realized by another USRP-N210

[3] We replaced the ISM band filter on `RFX900` by a capacitor of 100pF to get an extra 2dB transmission power.

[4] UHD is designed to provide a host driver and API for Ettus products. Users are able to use the UHD driver standalone or with third party applications such as Gnu Radio and Labview.

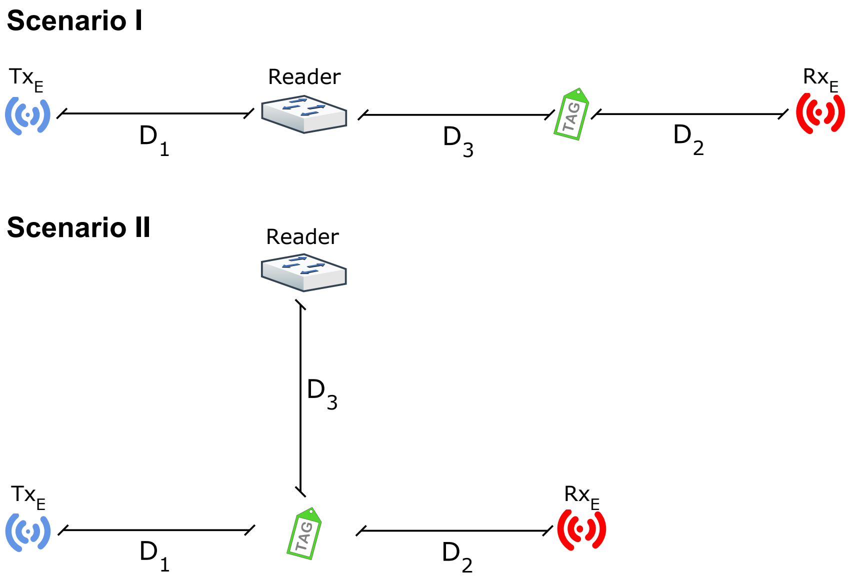

Figure 4.3: Our Prototyping System to Evaluate Active/passive Eavesdropping Attacks.

in conjunction with one `RFX900`, where the later carries a circular polarity panel antenna [42]. Similarly, in our Python script, we enable this $Tx_{\mathcal{E}}$ propagating a sine wave with a tunable center frequency $f_{\mathcal{E}}$ and a tunable amplitude amp. Furthermore, $Tx_{\mathcal{E}}$ is connected to a Thinkpad T410 laptop using the UHD driver. Note that USRPs use scale factors to represent the amplitude of a signal without unit. The actual measures of the output voltages at port `TX/RX` of `RFX900` (without antenna) with respect to this scale factor is provided shown in Table 4.1.

Table 4.1: Actual Measures of Output Voltages at Port `TX/RX` of `RFX900` w.r.t. Scale Factor

scale factor for USRP-1	scale factor for USRP-N210	output voltage
1000	0.01	396mv
2000	0.05	864mv
3000	0.1	1.120v
5000	0.2	2.124v
6500	0.5	2.400v
10000	0.8	2.880v

Eavesdropper's Receiver: To build a powerful $Rx_{\mathcal{E}}$, we make use of an USRP-1 together with one RFX900 carrying a dual polarization horn antenna effective from 700MHz-6GHz. A Python script is created to parse and process the received signals according to the two-branch receiver as shown in Figure 4.1. For each branch, the gain of the receiver's antenna is set to 20dB, and the received signal is decimated by the USRP-1 with a factor of 32. In addition, the decimated signals are again filtered by an 8-th order low-pass filter with gain 2 and cutoff frequency 400KHz, respectively. Finally, the two baseband signal are added and stored for the later analysis. Note that, to enable the collaboration between $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$, we connect $Rx_{\mathcal{E}}$ to the same Thinkpad T410.

Passive RFID Tag: The unique properties owned by a passive tag are the key points for our experiments. To this end, we made use of WISP v4.1 tag [214] from Intel Seattle Research – a full-fledged passive tag not only supports energy harvesting, ephemeral energy storage and backscatter communication, but also provides programmability. Each WISP is operated by a 16-bit general purpose microcontroller, MSP430F2132, the programs for which were written in embedded C and compiled, debugged and profiled using IAR Embedded Workbench 5.10.4, in conjunction with TI FET430UIF debugger. As an additional benefit, WISP is shipped with the firmware, hw41_D41.c-r65, which implements a significant portion of EPC Gen2 commands. For our purpose, we tweaked this firmware and let the tag transmit a random binary sequence (further encoded by Miller-4 code) at 64kbps as long as there is available power. Again, this automatic-replay policy helps us to remove the unnecessary reader-tag interaction in EPC Gen2 and lead us to be more concentrated on the backscattered signal.

Environment: In reality, multi-path effects and interferences from other RF transmitters nearby is an enemy for our proof-of-concept experiments. We primarily conducted the experiments in a microwave anechoic chamber of size 2.7m(L) × 1.5m(W) × 1.9m(H), which is a shielded room insulated from exterior sources of noise, and whose walls have been covered with a material that scatters or absorbs so much of the incident energy to simulate free space. In addition, we placed the devices on a horizontal surface in the chamber and let $D_1 = 65\text{cm}, D_2 = 83\text{cm}, D_3 = 48\text{cm}$ for scenario I; $D_1 = 89\text{cm}, D_2 = 107\text{cm}, D_3 = 48\text{cm}$ for scenario II.

4.3.2 Testing Results

In our testings, we varied the center frequency $f_{\mathcal{E}}$ of $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$ around 915MHz and tuned the transmitting power of $Tx_{\mathcal{E}}$, e.g., by setting the scale factor from 0.1 to 0.5. The results for scenario I are recorded in Figure 4.5, while the results for scenario II are recorded in Figure 4.6. Note that, to enable clear analysis, we independently plot the signals go through the two branches in our receiver: (1) the signals go through the upper branch (that mixing $\cos 2\pi f(t)$ and filtering at f) are exhibited in Figure 4.4, which can be seen as the results of a conventional passive eavesdropping; (2) the signals go through the bottom branch are exhibited in Figure 4.5 and Figure 4.6, which are the results of eavesdropping using the blank carrier only. Note that the results obtained by the attacker should be the addition of signals in both branches. Moreover, each short column represents a backscattered signal from the WISP tag that contains a random binary sequence of 16 bits.

As can be seen in these figures, active eavesdropping is surprisingly effective as, for example, in scenario I, the eavesdropper could get a view of the tag's response at $f_{\mathcal{E}} = 875$MHz with $amp = 0.5$ as clear as the ones he gets by passively eavesdropping at $f_{\mathcal{E}} = 915$MHz, which implies, when adding up these two signals, the minimum distance in his signal constellation regarding the tag's responses is doubled, which results in a significant decrease (increase resp.) in his BER (reliability resp.). A similar phenomenon happens in scenario II, which further confirms our theoretic analysis in the previous section.

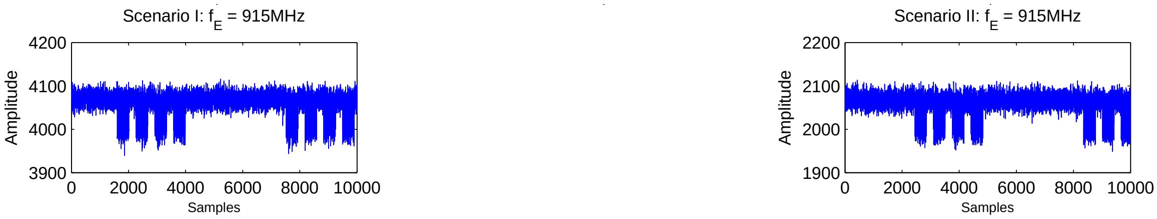

Figure 4.4: Testing Results of Passive Eavesdropping (Scenario I/II) in an Anechoic Chamber

By a vertical comparison of these plots, we can see clearly that the antenna of the WISP performs best when $f_{\mathcal{E}} = 875$MHz, 900MHz and 915MHz, and it can work at $f_{\mathcal{E}} = 960$MHz if a strong stimuli is given. This observation ensures our previous definition and discussion about the effective region of UHF tags. By a horizontal comparison of these plots, we can see clearly that the amplitude of the backscattered signals do not have a simple linear relation with respect to the amplitude of the blank carrier. For example, the second row in Figure 4.5 indicates that, when $amp = 0.5, 0.2, 0.1$, the amplitude of the backscattered signals are almost the same, albeit the amplitude of the blank carrier as received by $Rx_{\mathcal{E}}$ show significant difference. In addition, if the signal emitted by $Tx_{\mathcal{E}}$ is to weak, e.g., $amp \leq 0.1$, it gets lost during the propagating and hardly results in meaningful responses in $Rx_{\mathcal{E}}$. By scenario-wise comparison, we found, although it is intuitive that attacker in scenario I could achieve better benchmarks, there is essentially no much difference in

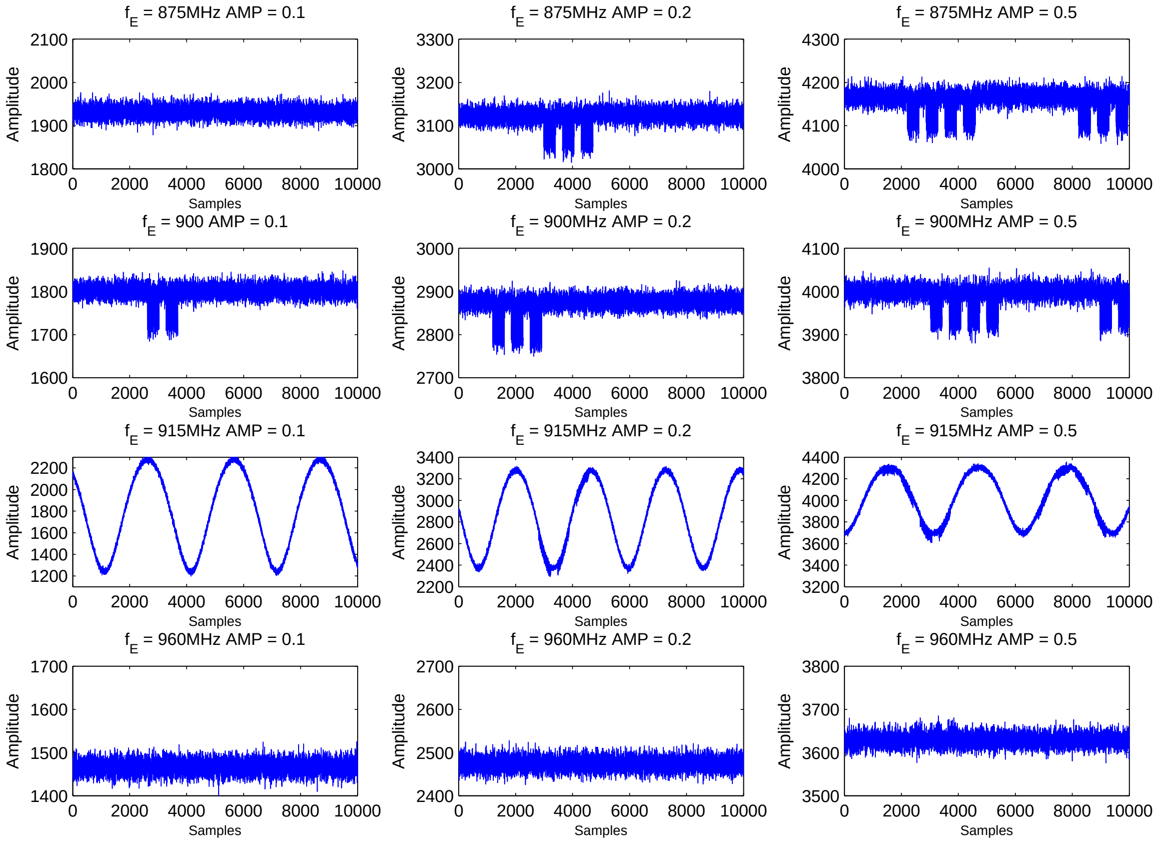

Figure 4.5: Testing Results of Active Eavesdropping (Scenario I) in an Anechoic Chamber

terms of the amplitude of the backscattered signal that the attacker obtains from these two scenarios. A possible explanation is that: we can consider the reader and the tag as one unit – an active tag, which is interrogated by $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$. Because $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$ are always placed in one line and both of them use directional antennas, thus two scenarios can be essentially reduced to one more basic scenario.

Moreover, most catching plots among many are the third rows of Figure 4.5 and Figure 4.6, in which, when $f_{\mathcal{E}} = 915$MHz, the received backscatter signal contains not only the tag's responses but also a low-frequency sine wave. After careful examination, we found this phenomenon is generated by using different USRPs (e.g., the $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$), i.e., even both of them are set to work at the same 915MHz, a slight different in their carrier and mixer, e.g., introduced by manufacturing deviations in the oscillators, produces this sine curve. Furthermore, we also noticed that this sine curve prevents the legitimate reader from

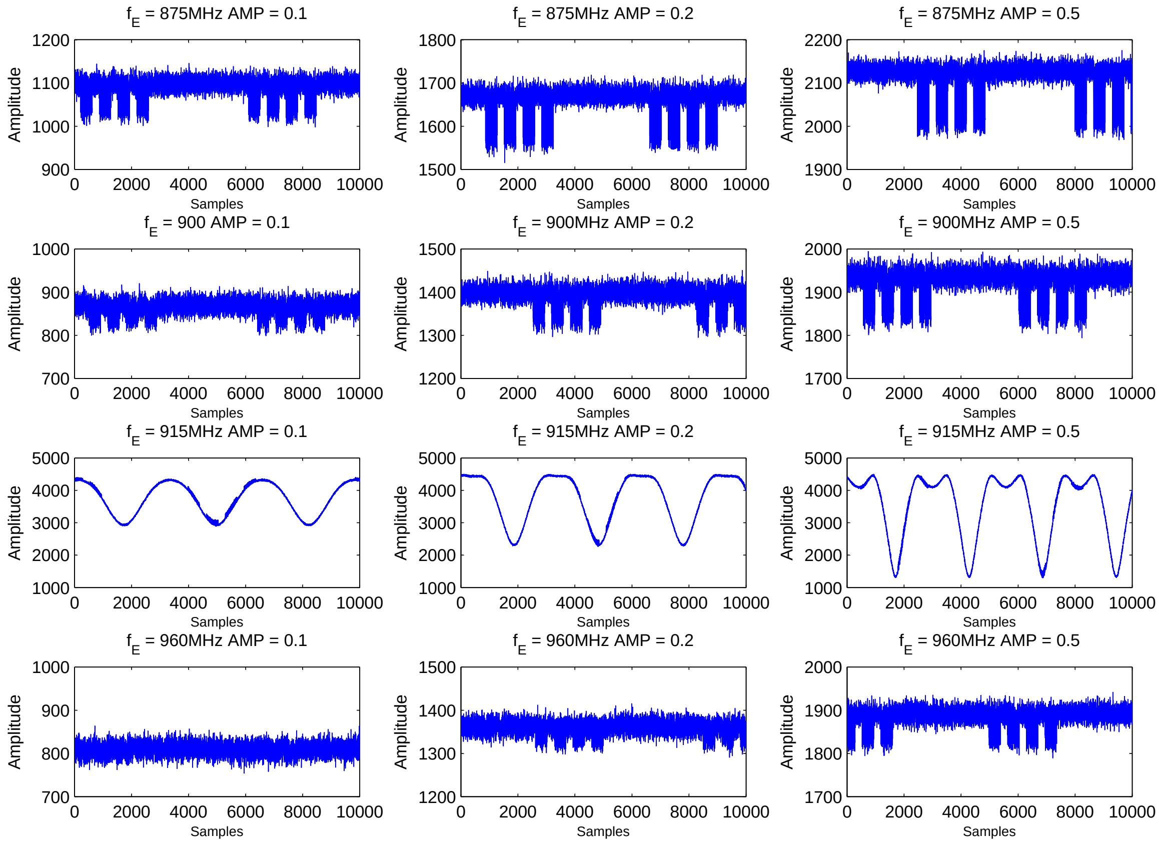

Figure 4.6: Testing Results of Active Eavesdropping (Scenario II) in an Anechoic Chamber

reading the tag's responses, which is possible because it could pass through the low-pass filter and could cause distortion at the receiver's decision logic.

4.3.3 Discussion

Active eavesdropping is a powerful tool for the attacker not only because it offers extra reliability to the eavesdropper's the channel, but also because the attacker has a rich choice of strategies which either further enhance this attack or lead this attack undetectable as in the case of passive eavesdropping. We classify these strategies as below. Note that, the strategies in different categories can work together in a composed way.

Strategies in Frequency Domain: It is suffice that the choice of $f_{\mathcal{E}}$ entirely depends on the effective region of the tag's antenna. The attacker may desire to select $f_{\mathcal{E}}$ out of $902 - 928$MHz to avoid the jamming of ongoing communications and also to avoid been detected. Fortunately, as we noticed, almost all the passive tags nowadays are designed and manufactured for global use, which are operational between 860-960MHz, e.g., Squiggle from Alien, Frog 3D from UPM, Cargo from Motorola, etc.. Therefore, it seems easy to pick up $f_{\mathcal{E}}$ in 800-1000MHz to launch the attack. In addition, since all the attacking devices are centrally coordinated, the attacker could periodically or aperiodically change $f_{\mathcal{E}}$ for both $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$ following the idea of the frequency hopping, which renders him even harder to be detected.

Another vision we have is that since there are many devices working at 800-1000MHz, e.g., cell phones (902-928MHz), pagers (929-932MHz), two-way radios (935-941MHz) and so on. The attacker could emit a camouflage signal – a blank carrier according to the signal format of these devices. Moreover, he could even treat nearby active radio devices (working within this band) as $Tx_{\mathcal{E}}$ and only launches $Rx_{\mathcal{E}}$ to listen. By employing this "camouflage" strategy, the attacker could remain undetected and decrease the budget needed to launch the attack.

Strategies in Time Domain: There are two strategies in the time domain, namely *proactive eavesdropping* and *reactive eavesdropping*. The former refers that both $Tx_{\mathcal{E}}$ and $Rx_{\mathcal{E}}$ keep working all the time, whereas, in the latter case, $Tx_{\mathcal{E}}$ works only if necessary, e.g., $Tx_{\mathcal{E}}$ starts to transmit when $Rx_{\mathcal{E}}$ is informed by a query command from the legitimate reader, and stops to work at the beginning of the next command from the legitimate reader.

After applying the reactive eavesdropping strategy, together with a frequency hopping, the detection of the active eavesdropper can be understood as the *spectrum sensing problem* in cognitive radio, which is generally considered as a hard problem. As pointed out in [111], the only reliable detectors exploit the known structure of signals, which may not be available in this scenario.

Strategy in Space Domain: In our proof-of-concept experiments, only one $Tx_{\mathcal{E}}$ and one $Rx_{\mathcal{E}}$ are deployed. In fact, the attacker could launch *distributed active eavesdropping* in the sense that multiple $Tx_{\mathcal{E}}$s and multiple $Rx_{\mathcal{E}}$s located around the target reader and the tag and work collaboratively. Under this strategy, the attacker is able to collect multiple observations of the tag's response to make a decision about the bit actually transmitted by the tag. The signal detection theory tells us that, in a centralized detection scenario, to obtain the optimal performance, the attacker can simply perform *maximum likelihood test* based on the multiple observations.

4.4 Low-cost Detection of Greedy Proactive Eavesdroppers

The active eavesdropping attack, as formalized and demonstrated, is not trivial to be prohibited in general as it arises from the nature of backscatter communication. The degree of success that the attacker will achieve depends on the strategy he selects and resources he has, e.g., an attacker with a clever strategy and an expensive, specialized RF measurement equipment is almost certain to launch this attack without being detected. In this section, we investigate one possible countermeasure to prevent a particular type of greedy eavesdropper, who uses the proactive eavesdropping strategy. By "a greedy proactive eavesdropper", we mean that the attacker keeps emitting a strong blank carrier using an effective frequency.

As one may expect, a straightforward solution for this problem is to equip the reader with a spectrum sensor that could detect existence of nearby RF sources. However, this will increase the system complexity dramatically because: (1) the spectrum sensor itself may be expensive; (2) the reader has to monitor effective regions of tags of different models or from different manufactures, which may be different and may not be known by the reader in prior.

In what follows, we introduce a simple but effective mechanism built on the tag, which leverages the greediness of the proactive eavesdropper – the attacker is always charging a victim tag even when the legitimate reader stops to work. Hence, in our scheme, the reader stops emitting for a short while, call it *voluntary pause*, and expects the tag to be discharged and go to sleep mode if there is no presence of such an eavesdropper. On the contrary, if the tag always stays awake, the reader may suspect the existence of a greedy proactive eavesdropper nearby. Note that this scheme is ineffective for passive eavesdropping, reactive eavesdropping, and even proactive eavesdropping with well-controlled blank carriers (by a non-greedy attacker). Additionally, the essential function of this simple scheme is to complicate the adversaries's transmitter (and henceforth to increase his attacking budget), which has to carefully control the emitted power or shipped with advance strategies.

4.4.1 Discharging of Passive Tags

It is widely known that passive tags can neither work nor retain its current status (without writing the data to non-volatile memory) without the power supply from the reader. To be more specific, passive tags always harvest energy when possible and store it in a capacitor, which can be modeled by charging a capacitor in an RLC circuit. Similarly, without the continuous supply of energy, the capacitor discharges.

The RLC circuit theory tells us that

$$V(t_{dchg}) = V_0(1 - e^{\frac{-t_{dchg}}{\tau_0}}),$$

where $e = 2.71828$, τ_0 is the time constant decided by the RLC circuit and t_{dchg} is the discharging time. To be intuitive, the charging and discharging processes are given in Figure 4.7.

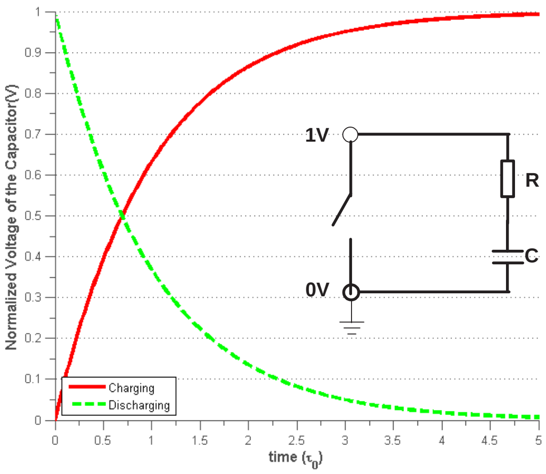

Figure 4.7: Charging and Discharging of Passive Tag's Capacitor

The formula above shows that before the capacitor reaches equilibrium, the voltage across the resistor (and actually the current through the entire circuit) decays exponentially. When $t_{dchg} = \tau_0$, 67% of the stored energy is dispensed. For example, assume $\tau_0 = 2ms$ and the normalized minimum voltage under which the tag works is 0.9, we have $t_{dchg} = 0.11\tau_0 \approx 220\mu$s, which implies that, as long as the reader stops working for around 220μs, a passive tag uses up all its power and losses all its current status. Nevertheless, if other power source is available, the voltage of the capacitor is unlikely to decay or to decay as fast as in the current case. For example, with the RF energy from the eavesdropper, we can safely assume $\tau_0 = 20ms$, which results in a discharge time $t_{dchg} \approx 2200\mu$s. Unsurprisingly, if the reader is switched off for $t\mu$s, $220 < t \leq 2200$, the tag: (1) is powered off when there is no RF source nearby; (2) stays awakes otherwise.

4.4.2 Counter-based Interaction

Based on this phenomenon and the observation that the tag is out of the control for the attacker, we designed the following *counter-based interaction* between the reader and the tag, which could easily be integrated into other RFID protocols. The basic idea is that the tag marks each reply with a counter number, which starts from 0 and keep increasing every round until it has been reset. Our scheme is shown in Table 4.2.

Table 4.2: Counter-based Interaction to Detect Greedy Proactive Eavesdroppers

Reader R		Tag	
carrier on	$\longrightarrow$	wake up and initialize $cnt = 0$	Communication
command	$\longrightarrow$	compute responses R	
	$\xleftarrow{(R,cnt)}$	$cnt = cnt + 1$	
...	...	...	
carrier off	$\longrightarrow$	may turn to sleep	Detection
carrier on	$\longrightarrow$	wake up and initialize $cnt = 0$	
command	$\longrightarrow$	compute responses R	
if $cnt \neq 0$, alarm!	$\xleftarrow{(R,cnt)}$	$cnt = cnt + 1$	

This detection scheme can be launched by the reader at variable times, under which a greedy proactive eavesdropper will be exposed with a zero false negative and a non-zero false positive. This is because any other proper RF sources nearby, e.g., cell phones, two-way radios, may result in the charging of the tags during the reader's voluntary pause. In addition, this simple scheme is helpful in the sense that, to avoid been detected, the adversary would either: (1) carefully control emitted power to be weak enough, which would introduce reliability penalty as indicated by Eq. (4.2); or (2) adopt other strategies, e.g., reactive eavesdropping, which are more expensive; or (3) switch back to be passive eavesdropping, and lose all the gains in the active eavesdropping.

4.4.3 Proof-of-concept Implementation

We successfully implemented this interaction on a WISP tag and an USRP-1 (serving as the reader). The tag-side implementation is extremely simple, e.g., by adding a self-increasing variable to the current firmware, which is returned with each of the tag's responses. This modification results no changes in its resource consumption. For the reader-side implementation, a key parameter to be determined is the length of the voluntary pause.

To choose a proper voluntary pause for a WISP v4.1 tag, which operates at 1.8V while the capacitor can be charged at most to 5.5V, we carried out the following test: we used a regular carrier to query a WISP tag 20cm away from the USRP's antenna using a regular carrier of 915MHz with maximum power[5] for 2 second to ensure the WISP's capacitor is fully charged. We then switched off the USRP and measured the unregulated and regulated voltage (relative to GRN) of the capacitor by connecting probes to CAP_CHARGE as well as VOLTAGE_SV_PIN on the WISP tag, and observing the output through the DPO-7104 oscilloscope. The actual measurements are presented in Figure 4.8. The time that the unregulated voltage drops from 5.5V to 1.8V is 280ms. Therefore, we set voluntary pause to be 280ms in our implementation. In reality, the commercial EPC Gen2 tags have smaller capacitors relative to that of the WISP tag, which may present much shorter discharging times and the choose of voluntary pause should be followed.

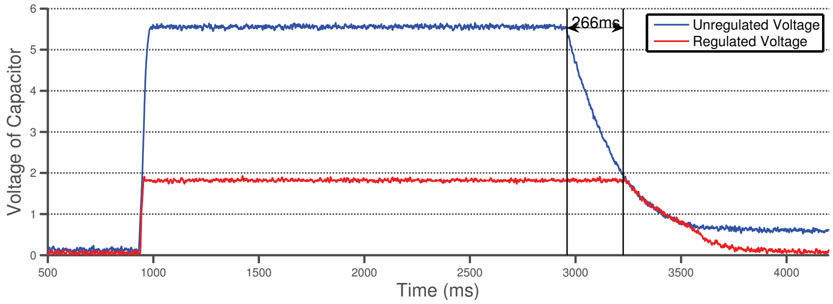

Figure 4.8: Unregulated (top) and Regulated (down) Voltage of WISP's Capacitor (note that raw data is obtained by DPO-7104 oscilloscope, sampled per 1ms)

[5] The distance between the reader and the tag and the power level from the reader do not affect the discharging process. They do have impact on the charging process.

4.5 Conclusion

In this chapter, we introduce, formalize, analyze and demonstrate a novel family of attack called unidirectional active eavesdropping, which takes advantage of the unique characteristics of the passive RFID systems and brings the eavesdropper to his full potential. As an additional effort, we propose a simple mechanism to thwart a particular type of the active eavesdropping attack, namely a greedy proactive eavesdropper, although a bullet-silver countermeasure in general may not exist.

Although, we did not observed the tag's response when actively eavesdropping on a WISP with $f_{\mathcal{E}} = 1.8$GHz, 2.2GHz and 2.4GHz, the dipole antenna is able to pick up odd-numbered multiples of the base frequency in principle. We will continue to investigate on this point. Moreover, we plan to get more quantitative results based on our current settings and investigate the possibility to actively eavesdrop NFC systems as well.

Chapter 5

Differential Sequence Attack on HummingBird-2

Hummingbird-2 is a novel lightweight cryptographic algorithm designed for passive RFID tags and other resource-constrained devices, which not only enables a compact hardware implementation and ultra-low power consumption but also meets the stringent response time as specified in ISO18000-6C. In this chapter, we present an innovative cryptanalytic method called *Differential Sequence Attack* (DSA), in conjunction with guessing and determining the internal states of the cipher, that successfully breaks the full HB-2 using two related keys, thereby giving the first cryptanalytic result on this cipher. To be specific, we exhibit the following results:

1. By attacking the encryption of HB-2, DSA recovers 36-bit out of 128-bit key with $2^{36} \times 2^{16} \times 2^{16} = 2^{68}$ time complexity and negligible memory complexity, if one particular condition regarding HB-2's internal states holds.

2. By attacking the decryption of HB-2, DSA recovers another 28-bit out of 128-bit key with $2^{28} \times 2^{16} \times 2^{16} = 2^{60}$ time complexity and negligible memory complexity, if another particular condition regarding HB-2's internal states holds.

3. The rest 64-bit of the key can be exhaustively searched. The overall time complexity for these steps is approximately 2^{68}.

4. To realize the two particular conditions needed by step 1 and step 2 respectively:

- The attacker could mount side-channel attack, e.g., to inject the difference to the victim register any time before the execution of the last round of encryption/decryption, without time/memory penalty, i.e., the overall time/memory complexity of the attack is dominated by 2^{68}.
- Without bothering side-channel models, the attacker could also makes use of guess-and-determine strategy – in each trial, the attacker produces two instances of HB-2 with random internal states and determines the occasion of the desired condition using a proposed algorithm with 2^{17} time complexity. In order to succeed with 0.5 probability, 2^{64} such trials have to be made, which results in an overall time complexity of $2^{64} \times 2^{17} = 2^{81}$. This probability can be increased if the attacker is willing to pay more on computation and vice versa.

The chapter is organized as follows. In Section 5.1, the brief history of HB-2 and its specification are presented. Section 5.2 describes, at a high level, the principle of our attack. In Section 5.3, we devise our tool, discuss its properties and how to use it to attack the last round of HB-2. In Section 5.4, we show how to achieve the desired conditions. We conclude this chapter in Section 5.5. Throughout this chapter, we use the following notation for illustration:

- "+" denotes addition in $\mathbb{F}_2$, which can also be vector-wise, e.g., $(a, b) + (c, d) = (a + c, b + d)$, where $a, b, c, d \in \mathbb{F}_2^m$.
- An hexadecimal number is indicated by a prefix "0x", e.g., 0x10 = 16.
- $\boxplus$ operator denotes addition modulo 2^{16} and $\boxminus$ operator denotes subtraction modulo 2^{16}.
- The high-bit XOR differential is defined as H =0x8000, a nice property of which is, given $x, x', y \in F_2^{16}$ and $x + x' = H$, the following holds

  $$(x \boxplus y) + (x' \boxplus y) = H; \quad (x \boxminus y) + (x' \boxminus y) = H; \quad (y \boxminus x) + (y \boxminus x') = H.$$

  That is to say, as also pointed out in [189], the differential H behaves the same under $+$ and $\boxplus/\boxminus$.

5.1 The Hummingbird-2 Cryptographic Algorithm

5.1.1 A Brief History of Hummingbird Family

Motivated by the design of the well-known Enigma machine, the first generation of Hummingbird (call it HB-1) was proposed by the engineers in Revere Security and was further analyzed and published in [83] as an ultra-lightweight cryptographic algorithm targeted for low-cost RFID tags, smart cards, and wireless sensor nodes to meet the stringent response time and power consumption requirements. Although HB-1, with an innovative hybrid structure of block cipher and stream cipher, was designed to provide 256-bit security, Saarinen, in FSE'11, showed a chosen-IV and chosen-message attack in [189] that can recover the full secret key with at most 2^{64} off-line computational effort under two related IVs. Recently, Reverse Security published the second generation of Hummingbird (call it Hummingbird-2 or HB-2) in [84], which inherits the design philosophy from HB-1, e.g., it has a small block size of 16-bit to adapt the needs of encrypting short messages in RFID applications and it retains the hybrid structure as a security compensation for the small block size. High level differences between HB-1 and HB-2 are:

- Key size has been reduced to 128 bits to satisfy the actual need for constrained devices.
- Size of the internal state has been increased from 80 bits to 128 bits.
- The nonlinear keyed transformation in HB-2 has four invocations of the S-boxes, compared to five in HB-1, to further increase the throughput.

In addition, it is claimed by the designers that HB-2 can withstand differential, linear and algebraic attacks and the four 4-bit S-Boxes in HB-2 belong to the optimal classes discussed in [157]. Its resistance to the side-channel cube attack is recently investigated in [86], where the author applied cube attack [74] to recover 48 bits of the secret key providing the attacker could access the internal states of HB-2 during an early stage in the initialization. However, this attack is marginal and does not jeopardize the security of HB-2, since: (1) it only threats HB-2 before the finishing of its initialization; (2) obtaining particular intermediate values in a cipher is less-universally accepted as a valid attack model.

5.1.2 Specification of Hummingbird-2

Hummingbird-2 is a 16-bit block cipher with a 128-bit secret key $K = (K_1, ..., K_8) \in (\mathbb{F}_2^{16}, ..., \mathbb{F}_2^{16}) = \mathbb{F}_2^{128}$ and a 64-bit public initialization vector $IV = (IV_1, ..., IV_4) \in (\mathbb{F}_2^{16}, ..., \mathbb{F}_2^{16}) = \mathbb{F}_2^{64}$. As opposed to conventional block ciphers, it has an 128-bit internal state $R = (R_1, ..., R_8) \in (\mathbb{F}_2^{16}, ..., \mathbb{F}_2^{16}) = \mathbb{F}_2^{128}$, which participates in each encryption/decryption and is updated after that.

Building Block: $WD16 : \{0,1\}^{16} \mapsto \{0,1\}^{16}$ is the fundamental block of HB-2 encryption, i.e.,

$$WD16(x, K_a, K_b, K_c, K_d) = f(f(f(f(x + K_a) + K_b) + K_c) + K_d),$$

where x is the varying input, e.g., plaintext, intermediate state, K_a, K_b, K_c, K_d are four 16-bit secret keys and the nonlinear function f is

$$\begin{aligned} S(x) &= S_1(x_1)||S_2(x_2)||S_3(x_3)||S_4(x_4), x = (x_1, x_2, x_3, x_4) \\ L(x) &= x + (x <<< 6) + (x <<< 10) \\ f(x) &= L(S(x)). \end{aligned}$$

Note that the four S-boxes, i.e., $S_1(x_i)$ to $S_4(x_i)$, are given in Table 5.1.

Table 5.1: S-boxes in HummingBird-2

$x_i \in \mathbb{F}_2^4$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_1(x_i)$	7	12	14	9	2	1	5	15	11	6	13	0	4	8	10	3
$S_2(x_i)$	4	10	1	6	8	15	7	12	3	0	14	13	5	9	11	2
$S_3(x_i)$	2	15	12	1	5	6	10	13	14	8	3	4	0	11	9	7
$S_4(x_i)$	15	4	5	8	9	7	2	1	10	3	0	14	6	12	13	1
$S_1^{-1}(x_i)$	11	5	4	15	12	6	9	0	13	3	14	8	1	10	2	7
$S_2^{-1}(x_i)$	9	2	15	8	0	12	3	6	4	13	1	14	7	11	10	5
$S_3^{-1}(x_i)$	12	3	0	10	11	4	5	15	9	14	6	13	2	7	8	1
$S_4^{-1}(x_i)$	10	7	6	9	1	2	12	5	3	4	8	15	13	14	11	0

Besides, the inverse of $WD16$ is employed in the decryption, which is defined as

$$WD16^{-1}(y, K_d, K_c, K_b, K_a) = f^{-1}(f^{-1}(f^{-1}(f^{-1}(y) + K_d) + K_c) + K_b) + K_a,$$

where $y = WD16(x, K_a, K_b, K_c, K_d)$ and f^{-1} is the inverse of f. The four S-boxes used in f^{-1} are also listed in Table 5.1.

Initialization: Hummingbird-2 is initialized before use. Let $(R_1^{(r)}, ...R_8^{(r)}) \in \{0,1\}^{128}$ denote the internal state at the rth *iteration* in the initialization. The initialization can thus be formulated as, for $r = 0, 1, 2, 3$,

$$\begin{aligned}
t_1 &= WD16(R_1^{(r)} \boxplus r, K_1, K_2, K_3, K_4) && (5.1)\\
t_2 &= WD16(R_2^{(r)} \boxplus t_1, K_5, K_6, K_7, K_8) && (5.2)\\
t_3 &= WD16(R_3^{(r)} \boxplus t_2, K_1, K_2, K_3, K_4) && (5.3)\\
t_4 &= WD16(R_4^{(r)} \boxplus t_3, K_5, K_6, K_7, K_8) && (5.4)\\
R_1^{(r+1)} &= (R_1^{(r)} \boxplus t_4) \lll 3 && (5.5)\\
R_2^{(r+1)} &= (R_2^{(r)} \boxplus t_1) \lll 1 && (5.6)\\
R_3^{(r+1)} &= (R_3^{(r)} \boxplus t_2) \lll 8 && (5.7)\\
R_4^{(r+1)} &= (R_4^{(r)} \boxplus t_3) \lll 1 && (5.8)\\
R_5^{(r+1)} &= R_5^{(r)} + R_1^{(r+1)} && (5.9)\\
R_6^{(r+1)} &= R_6^{(r)} + R_2^{(r+1)} && (5.10)\\
R_7^{(r+1)} &= R_7^{(r)} + R_3^{(r+1)} && (5.11)\\
R_8^{(r+1)} &= R_8^{(r)} + R_4^{(r+1)}, && (5.12)
\end{aligned}$$

where

$$(R_1^{(0)}, ..., R_8^{(0)}) = (IV_1, IV_2, IV_3, IV_4, IV_1, IV_2, IV_3, IV_4).$$

Note that R_5, R_6, R_7, R_8 do not participate in the randomization, i.e., Eq. (5.1)-(5.4), but simply XOR the historical statuses of R_1, R_2, R_3, R_4 respectively (behaving like an XOR-MAC). This fact may nullify their contribution to the overall cryptanalytic strength of HB-2 under a side-channel attack – following steps allow a side-channel attacker, who is able to inject "1" to a certain bit of the register storing R_j, $5 \leq j \leq 8$, to recover (R_5, R_6, R_7, R_8):

1. The attacker encrypts with a known IV and the target key to get a plaintext/cipher pair (P, C), where $P \in \mathbb{F}_2^{16}, C \in \mathbb{F}_2^{16}$.

2. He resets HB-2 and initializes HB-2 with the same IV and key. At any time during this initialization, he injects "1" to the qth bit, $0 \leq q \leq 15$, of the register which stores R_5. He then encrypts P and gets C'. If $C = C'$ (which implies the injection does not change the internal states of HB-2), the attacker in fact learns that the qth bit of R_5 is 1; otherwise it is 0. He repeats this step for every q in $\{0, 1, ..., 15\}$ to recover R_5.

3. Step 2 can be repeated to recover R_6, R_7 and R_8.

This injection attack to recover (R_5, R_6, R_7, R_8) only requires 64 injections and 64 invocations of HB-2 encryption. In addition, since the attacker has a large time window to perform the injection to the qth bit of R_j (any time during the rth iteration of the initialization), this side-channel attack seems quite possible.

Encryption: After the initialization, each *round* of encryption transforms a single plaintext word $P_i \in \mathbb{F}_2^{16}, i = 1, 2, ...,$ to a corresponding ciphertext word C_i, i.e.,

$$\begin{aligned}
t_1 &= WD16(R_1^{(i)} \boxplus P_i, K_1, K_2, K_3, K_4) && (5.13)\\
t_2 &= WD16(R_2^{(i)} \boxplus t_1, K_5 + R_5^{(i)}, K_6 + R_6^{(i)}, K_7 + R_7^{(i)}, K_8 + R_8^{(i)}) && (5.14)\\
t_3 &= WD16(R_3^{(i)} \boxplus t_2, K_1 + R_5^{(i)}, K_2 + R_6^{(i)}, K_3 + R_7^{(i)}, K_4 + R_8^{(i)}) && (5.15)\\
C_i &= WD16(R_4^{(i)} \boxplus t_3, K_5, K_6, K_7, K_8) \boxplus R_1^{(i)}, && (5.16)
\end{aligned}$$

where $(R_1^{(i)}, ..., R_8^{(i)}) \in \mathbb{F}_2^{128}$ is the internal state at round i and it is updated, at the end of each round, as follows:

$$\begin{aligned}
R_1^{(i+1)} &= R_1^{(i)} \boxplus t_3 && (5.17)\\
R_2^{(i+1)} &= R_2^{(i)} \boxplus t_1 && (5.18)\\
R_3^{(i+1)} &= R_3^{(i)} \boxplus t_2 && (5.19)\\
R_4^{(i+1)} &= R_4^{(i)} \boxplus t_1 \boxplus R_1^{(i+1)} && (5.20)\\
R_5^{(i+1)} &= R_5^{(i)} + R_1^{(i+1)} && (5.21)\\
R_6^{(i+1)} &= R_6^{(i)} + R_2^{(i+1)} && (5.22)\\
R_7^{(i+1)} &= R_7^{(i)} + R_3^{(i+1)} && (5.23)\\
R_8^{(i+1)} &= R_8^{(i)} + R_4^{(i+1)} && (5.24)
\end{aligned}$$

A shorthand of Eq. (5.13)-(5.24) is $C_i = E(P_i, K) = E(P_i, (K_1, ..., K_8))$.

Decryption: Decryption of a single word $C_i \in \mathbb{F}_2^{16}, i = 1, 2, ...,$ followed by the same initialization, is

$$\begin{aligned}
u_3 &= WD16^{-1}(C_i \boxminus R_1^{(i)}, K_8, K_7, K_6, K_5) && (5.25)\\
u_2 &= WD16^{-1}(R_4^{(i)} \boxminus u_3, K_4 + R_8^{(i)}, K_3 + R_7^{(i)}, K_2 + R_6^{(i)}, K_1 + R_5^{(i)}) && (5.26)\\
u_1 &= WD16^{-1}(R_3^{(i)} \boxminus u_2, K_8 + R_8^{(i)}, K_7 + R_7^{(i)}, K_6 + R_6^{(i)}, K_5 + R_5^{(i)}) && (5.27)\\
P_i &= R_1^{(i)} \boxminus WD16^{-1}(R_2^{(i)} \boxminus u_1, K_4, K_3, K_2, K_1). && (5.28)
\end{aligned}$$

After this, the internal states are updated as in the encryption, i.e., using Eq. (5.17)-(5.24), where $t_3 = R_4^{(i)} \boxminus u_3$, $t_2 = R_3^{(i)} \boxminus u_2$ and $t_1 = R_2^{(i)} \boxminus u_1$.

5.2 Overview of Our New Attack on the Full HB-2

Adversary Model: We consider a scenario that two paralleled-and-independent executions of encryptions are $C_i = E(P_i, (K_1, ..., K_8))$ and $C'_{i'} = E(P'_{i'}, (K'_1, ..., K'_8))$, whose internal states are $(R_1^{(i)}, ..., R_8^{(i)})$ and $(R'^{(i')}_1, ..., R'^{(i')}_8)$ respectively and whose intermediate values are (t_1, t_2, t_3) and (t'_1, t'_2, t'_3) respectively. (Notations for the decryptions are treated similarly). The attacker follows a conventional chosen plaintext/ciphertext model – he is free to choose plaintext $P_i \in \mathbb{F}_2^{16}$ and $P'_{i'} \in \mathbb{F}_2^{16}$, launch encryption without knowing the key, and observe the corresponding $C_i \in \mathbb{F}_2^{16}$ and $C'_{i'} \in \mathbb{F}_2^{16}$; or he chooses $C_i \in \mathbb{F}_2^{16}$ and $C'_{i'} \in \mathbb{F}_2^{16}$, launches decryption without knowing the key, and observes the corresponding $P_i \in \mathbb{F}_2^{16}$ and $P'_{i'} \in \ F_2^{16}$.

Our Attack in a Nutshell: Block ciphers are usually based on iterating a cryptographically weak function sufficient number of times without disturbing, e.g., modifying, the outputs of intermediate rounds except whitening them with round-keys. Our new attack on the full HB-2 exploits the fact that R_4 in the encryption (R_2 in the decryption resp.) is modulo added to t_3 (modulo subtracts u_1, resp.), which, instead of enhancing the overall cryptanalytic strength, gives the attacker an opportunity to create an input differential for the last round and to retrieve the corresponding output differentials caused by the last round. Note that we consider the collection of all such output differentials and call it a *differential sequence*, which is defined in the next section, and which are information-rich in the keys $(K_5, ..., K_8)$ $((K_4, ..., K_1)$ resp.). Our full attack can be divided into two phases: **preparation phase** as described in Section 5.4 and **key recovery phase** as described in Section 5.3.

Key Recovery Phase: In the *key recovery phase*, to remove the undesired interference introduced by the varied internal states when consecutive words of input is encrypted/decrypted, our attack here targets only the encryption/decryption at a particular round after the *preparation*, i.e., ith round for one HB-2 instance and i'th round for the other one. This is because given the key, IV, and the plaintext chain fed are fixed, the internal states at the ith and i'th round are fixed as well, although they are unknown to the attacker. Note that since only ith round and i'th round are considered, we omit the superscript/subscript

i and i' of HB-2 variables for convenience when describing operations in the key recovery phase.

In the *key recovery phase*, we make use of the divide-and-conquer strategy to make our attack substantially faster than exhaustive search. During this phase, the attacker accomplishes the following:

- Step 1. 36 bits of $(K_5, ..., K_8) \in \mathbb{F}_2^{64}$ are recovered using the differential sequence obtained from the last round of HB-2 encryption if a particular condition meets.
- Step 2. 28 bits of $(K_4, ..., K_1) \in \mathbb{F}_2^{64}$ are recovered using the differential sequence obtained from the last round of HB-2 decryption if another particular condition meets.
- Step 3. the rest 64-bit key are exhaustively searched using either encryption or decryption.

To be specific, the condition needed to launch Step 1 in *key recovery phase* is:

Condition (A):

$$R_4 + R'_4 = H,\ (t_1, t_2, t_3) = (t'_1, t'_2, t'_3)\ \textit{and}\ (K_5, ..., K_8) = (K'_5, ..., K'_8).$$

The condition needed to launch Step 2 in *key recovery phase* is:

Condition (B):

$$f^{-1}(R_2 \boxminus u_1) + f^{-1}(R'_2 \boxminus u'_1) = H,\ (u_1, u_2, u_3) = (u'_1, u'_2, u'_3)\ \textit{and}\\ (K_1, ..., K_4) = (K'_1, ..., K'_4).$$

Preparation Phase: As one may expected, *preparation phase* of our attack copes with the realization of the above conditions one at a time. To create the difference between R_4 and R'_4 (or between $f^{-1}(R_2 \boxminus u_1)$ and $f^{-1}(R'_2 \boxminus u'_1)$), one obvious way is to start with two instances initialized with the same IVs and keys and then mount side-channel injection attack, where the attacker simply injects H to the victim register, e.g., R_4 or $f^{-1}(R_2 \boxminus u_1)$, of one instance any time before the execution of the last round of encryption/decryption. Note that the *preparation through injection* gives the attacker no time/memory penalty, i.e., the overall time/memory complexity of the attack is dominated by that of the *key recovery phase*.

However, side-channel injection attack is not considered much in this work since the possibility to be success totally depends on the dedicated implementations of the cipher, which limits the use of our attack. To be away from this strong attacking model and to be more practical, we make use of the guess-and-determine strategy to realize both conditions in a probabilistic manner, i.e.,

- Guess: $(R_1^{(i)}, ..., R_8^{(i)})$ and $(R_1'^{(i')}, ..., R_8'^{(i')})$ can be "randomized" by feeding both HB-2 instances with either different IVs and/or chains of random plaintext words. According to the birthday paradox, there is at least 0.5 chance that the randomized $(R_1^{(i)}, ..., R_8^{(i)}) \in \mathbb{F}_2^{128}$ and the randomized $(R_1'^{(i')}, ..., R_8'^{(i')}) \in \mathbb{F}_2^{128}$ satisfies condition (A) (or condition (B)) providing 2^{64} attempts are made.
- Determine: Note that, in the previous step, even if condition (A) (or condition (B)) happens, the attacker cannot be aware of that since he is ignorant of $(R_1^{(i)}, ..., R_8^{(i)})$ and $(R_1'^{(i')}, ..., R_8'^{(i')})$. To determine, we develop an algorithm in light of another characteristic of HB-2, i.e., if one sufficient condition of condition (A) (or that of condition (B)) holds at the current round, it also holds for the next round. Hence, the differential sequences produced at the current round by $((R_1^{(i)}, ..., R_8^{(i)}), (R_1'^{(i')}, ..., R_8'^{(i')}))$ is exactly the same as that produced at the next round by $((R_1^{(i+1)}, ..., R_8^{(i+1)}), (R_1'^{(i'+1)}, ..., R_8'^{(i'+1)}))$. Two equivalent differential sequences produced by two consecutive rounds inform the attacker of the occasion of the desired condition.
- If the above step succeeds, the attacker performs the steps in the key recovery phase to attack.

In what follows, we detail each of the above phases and steps.

5.3 Differentials Sequence Attack (DSA)

In this section, we present the *key recovery phase* of our attack in detail by devising a tool called differential sequence and exhibiting its properties and applications to attack the last round of HB-2 encryption/decryption.

Differential cryptanalysis: *Differential cryptanalysis* is a method analyzing the effect of particular differences in plaintext pairs on the differences of the resultant ciphertext pairs, which is based on a crucial observation that for any particular input differential, not all the output differential are possible, and the possible ones may not appear uniformly. In the original version of differential cryptanalysis [193], a unique differential is exploited. The basic procedure of a differential cryptanalysis attack on an r-round iterated cipher can be summarized as follows:

1. Given the design of a cipher, find an $(r-1)$-round differential (α, β) such that given the input plaintext pairs have a difference α, the probability that the output pairs of the $(r-1)$th round have a difference β is maximum or nearly maximum.
2. Choose a plaintext P uniformly at random and compute $P' = P + \alpha$. Submit P and P' for encryption under the actual key. From the resultant ciphertext pairs, find every possible value (if any) of the subkey used in the last round corresponding to the anticipated difference β. Increase by one the counter of appearances of each such value of the subkey of the last round.
3. The above step is repeated a couple of times and the most suggested value is taken to be the subkey of the last round.

This idea has been extended in several ways: Biham and Shamir themselves further considered in [193] to use a trail of differentials to attack; Lai in [146] connected differential cryptanalysis with derivative of polynomials and presented a fine definition of higher order differentials; Knudsen [130] considered to use part of the input and output that have differential characteristics for the analysis; Biham, Biryukov and Shamir proposed in [16] to use differentials that happens with probability 0 as distinguishers; and recently, Blondeau and Gérard demonstrated the multiple differential cryptanalysis in [27], where a set of input/output differentials are considered together.

(First-order) Differential Sequence: Assume we have a keyed permutation $h(w, K)$ mapping $w \in F_2^m$ to $h(w, K) \in F_2^m$ bijectively with respect to the secret key $K \in F_2^n$, where m and n are positive integers. Given a fixed $\theta \in \mathbb{F}_2^m$, the first-order differential is

$$\Delta_{\theta,K}(w) = h(w, K) + h(w + \theta, K). \tag{5.29}$$

The *differentials sequence* of h at θ is basically one row in the differential distribution table of h with respect to the input differential θ. To discuss its properties, we define it in a more formal way.

Definition 1 *The first-order differential sequence (DS) of h at θ is a sequence of 2^m entries, i.e.,*

$$\Delta_{\theta,K} = [z_0, z_1, ..., z_{2^m-1}], \tag{5.30}$$

where z_i denotes the multiplicity (that is, number of occurrences) of i in the set $\{w \in \mathbb{F}_2^m | \Delta_{\theta,K}(w)\}$, i.e.,

$$z_i = |\{w \in \mathbb{F}_2^m |\ \Delta_{\theta,K}(w) = i\}|.$$

For example, the differential sequence is $\{0,0,4,0,2,0,0,2,0,0,0,0,4,0,2,2,0\}$ providing $h(w,K) = S_1(w)$ in HB-2, $\{w = 0,1,...,2^4-1|\Delta_{\theta,K}(w)\} = \{E,2,2,E,D,4,4,D,B,B,B,B,7,2,2,7\}$ and $\theta = 0\text{x}03$. The length of the differential sequence is the sum of all its multiplicities (16 in this example).

Note that this definition can be extended to higher orders. In this attack, we constrained ourselves to the first-order case.

5.3.1 Properties of Differential Sequence

A differential sequence of h at θ has the following properties.

Property 1 *$\Delta_{\theta,K}$ is constructed by evaluating and counting $(h(w,K) + h(w+\theta,K))$ with a fixed θ and every w in $\mathbb{F}_2^m$, regardless of the order of $w \in \{0,1,...,2^m-1\}$ been accessed.*

This property follows immediately from Definition 1 and is useful in the sense that even it is impossible to directly control or know w in $h(w,K)$, e.g., w is an intermediate value in a cipher, we can still generate $\Delta_{\theta,K}$ given that θ can be fixed and w can exhaust the whole space of $\mathbb{F}_2^m$.

Property 2 *Let $h(w,K)$ be $h(w,(K_l,K_{nl}))$, where $K = K_l \bigcap K_{nl}$, $K_l \bigcup K_{nl} = \emptyset$ and*

$$\begin{aligned} h(w,(a,K_{nl})) + h(w,(b,K_{nl})) &= h(w,(a+b,K_{nl})), \quad a,b \in F_2^{|K_l|} \\ h(w,(K_l,d)) + h(w,(K_l,d)) &\neq h(w,(K_l,c+d)), \quad c,d \in F_2^{|K_{nl}|}. \end{aligned}$$

Thus, $\Delta_{\theta,K}$ is correlated with (a subset of) K_{nl}.

This property can be visualized by viewing $h(w,K)$ as a vector boolean function, i.e., similarly as in the scenario of the cube attack [74], of public boolean variables from w and secret boolean variables from K. The first derivative at θ in fact filters out: (1) terms in $h(w,K)$ which have no certain public boolean variables; and (2) linear terms of the boolean variables in K_l. The construction of $\Delta_{\theta,K}$ is actually computing the final column of the truth table of the remaining polynomial and re-ordering it, which is information-rich in (at least a subset of) boolean variables from K_{nl}. Here we omit the information-theoretical derivation of the (bounds of) entropy of K_{nl} in $\Delta_{\theta,K}$, as we are more interested in its implication on the security of HB-2.

It is worth to emphasize that this property in fact implies that the obtained differential sequence of h at θ can be used to search for (a subset of) the key K nonlinearly associated. However, if h is the entire cipher, using the obtained differential sequence to search for the secret key is equivalent to using plaintext/ciphertext pairs to search for the secret key, which seems not appealing. As opposed, the differential sequence for a particular part of a cipher is useful as it reflects particular subkeys.

5.3.2 Differential Sequence Attack against Last Round of HB-2

In this subsection, we attack the last invocation of $WD16$ (or $WD16^{-1}$) in the encryption (or decryption) of HB-2 by exploiting the tool presented. Since the HB-2 has a 16-bit block size, we have $m = 16$ for the rest.

Attacking $WD16$ in Encryption: To show our idea in a concise way, we assume for the time being that R_1 and R'_1 are known. In addition, let h in Definition 1 be the last invocation of $WD16$ in the encryption, and recall the condition (A) is $R_4 + R'_4 = H$ while $(t_1, t_2, t_3) = (t'_1, t'_2, t'_3)$ and $(K_5, ..., K_8) = (K'_5, ..., K'_8)$. We thus have the following theorems for our attack.

Theorem 1 *With condition (A), the differential sequence of*

$$h(t_3, K) = WD16(R_4 \boxplus t_3, K_5, K_6, K_7, K_8)$$

at $\theta = H$ can be computed without knowing the key K from the following

$$\begin{aligned}
\Delta_{H,(K_5,...,K_8)} &= [z_0, z_1, ..., z_{2^{16}-1}], \text{ where} \\
z_i &= |\{t_3 \in \mathbb{F}_2^{16}|\ (WD16(R_4 \boxplus t_3, K_5, K_6, K_7, K_8) \\
&+ \ WD16(R'_4 \boxplus t'_3, K'_5, K'_6, K'_7, K'_8)) = i\}| \\
&= |\{t_3 \in \mathbb{F}_2^{16}|\ (WD16(R_4 \boxplus t_3, K_5, K_6, K_7, K_8) \\
&+ \ WD16((R_4 + H) \boxplus t_3, K_5, K_6, K_7, K_8)) = i\}| \\
&= |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus R_1) + (C' \boxminus R'_1) = i\}|.
\end{aligned}$$

Proof: Recall that the internal states and the key are fixed and condition (A) is imposed as shown in Figure 5.1. From Eq. (5.13) to Eq. (5.16), it is clear that $\{R_4 \boxplus t_3 | P \in \mathbb{F}_2^{16}\} \in \mathbb{F}_2^{16}$, e.g., $(R_4 \boxplus t_3)$ is a permutation t_3, which is a permutation of t_2, which is a permutation of t_1, which is a permutation of P. From Property 1, the above theorem follows. □

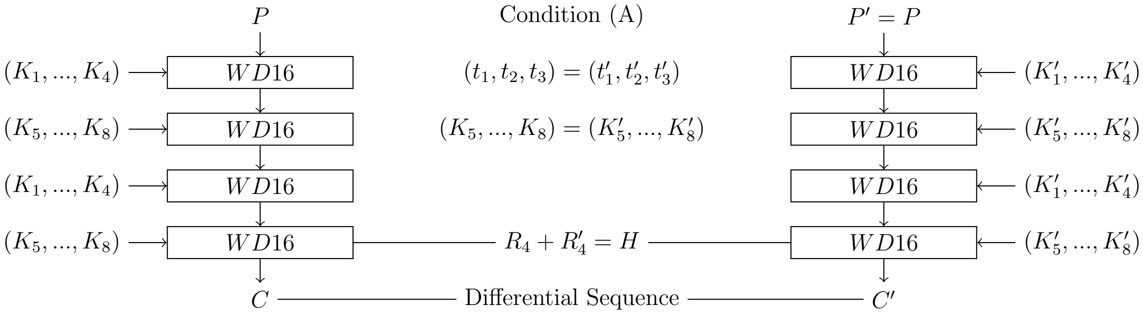

Figure 5.1: Constructing Differential Sequence from Encryption with Condition (A)

This theorem suggests that, after querying the encryption with every $P = P' \in \mathbb{F}_2^{16}$ and obtaining the resultant output differentials, the attacker has $\Delta_{H,(K_5,K_6,K_7,K_8)}$, which could be used to search for (part of) (K_5, K_6, K_7, K_8). The next theorem discloses the correspondence of $\Delta_{H,(K_5,K_6,K_7,K_8)}$ and (K_5, K_6, K_7, K_8).

Theorem 2 *Let $\Delta_{H,(K_5,K_6,K_7,K_8)}$ be obtained from Theorem 1. For $\kappa_6 \in \mathbb{F}_2^4$ and $(K_7, K_8) \in \mathbb{F}_2^{32}$, we have*

$$\Delta_{H,(K_5,K_6,K_7,K_8)} = \Delta_{H,(\kappa_6,K_7,K_8)},$$

where, let $K_6[i]$ represent the ith ($0 \leq i < 16$) bit of K_6,

$$\begin{aligned}\kappa_6 = (\quad & K_6[10] + K_6[12],\\ & K_6[11] + K_6[13],\\ & K_6[0] + K_6[2] + K_6[8] + K_6[10] + K_6[14],\\ & K_6[1] + K_6[3] + K_6[9] + K_6[11] + K_6[15] \ \ ).\end{aligned}$$

Proof: Here we provide an experimental verification of this theorem.

- In our first experiment, we make use of Eq. (5.16) to generate differential sequences at $\theta = H$ (thus we could control the input of Eq. (5.16) easily without bothering the entire encryption). Each of the following tests are repeated reasonable times to avoid loss of generality.

- *Test 1*: We use two instances with a key (K_5, K_6, K_7, K_8) and a key $(K'_5, K'_6, K'_7, K'_8) = (K_5, K_6, K_7, K_8)$ respectively to generate a differential sequence. Let us call the above process $\mathsf{DSGen}(K_5, K_6, K_7, K_8)$ hereafter. Thus, $\mathsf{DSGen}(K_5, K_6, K_7, K_8)$ was launched 2^{32} times with different chooses of (K_5, K_6, K_7, K_8) to produce differential sequences, each of which was either inserted into a hash table if it did not exist; or triggered a collision (here "collision" refers to the phenomena that two executions of DSGen with different (K_5, K_6, K_7, K_8) produces exactly the same $\Delta_{H,(K_5,K_6,K_7,K_8)}$), which was reported. In all the collisions we found, (K_7, K_8) are the same while (K_5, K_6) varies. Partial results are tabulated in Table 5.2.
- *Test 2*: We first use $\mathsf{DSGen}(K_5, K_6, K_7, K_8)$ to generate a template differential sequence with the secret key (K_5, K_6, K_7, K_8) which is randomly selected. Next, DSGen is called with a guessed key $(\hat{K}_5, \hat{K}_6, \hat{K}_7, \hat{K}_8)$, i.e., $\mathsf{DSGen}(\hat{K}_5, \hat{K}_6, \hat{K}_7, \hat{K}_8)$. Note that here we set $(\hat{K}_6, \hat{K}_7, \hat{K}_8) = (K_6, K_7, K_8)$, tried each $\hat{K}_5 \in \mathbb{F}_2^{16}$ and recorded the produced $\Delta_{H,(\hat{K}_5,\hat{K}_6,\hat{K}_7,\hat{K}_8)}$ if it matches our template sequence. As a result, $(2^{16} - 1)$ collisions happens, which suggests

  $$\forall \hat{K}_5 \in \mathbb{F}_2^{16}, \Delta_{H,(\hat{K}_5,K_6,K_7,K_8)} = \Delta_{H,(K_5,K_6,K_7,K_8)}.$$

  Thus, we say that K_5 is *uncorrelated* to the obtained template DS.
- *Test 3*: Similar as *Test 2* except that we set $(\hat{K}_5, \hat{K}_7, \hat{K}_8) = (K_5, K_7, K_8)$, tried each $\hat{K}_6 \in \mathbb{F}_2^{16}$ and recorded the produced $\Delta_{H,(\hat{K}_5,\hat{K}_6,\hat{K}_7,\hat{K}_8)}$ if it equals to our template sequence. As a result, $(2^{12} - 1)$ collisions happens, which suggests that K_6 is *partially correlated* to the obtained template DS.

  To further investigate which part of K_6 is correlated, we fixed parts $\hat{K}_6$ to be correct and varied the rest to observe the collisions found. As a result, we noticed that the second nibble is uncorrelated while the rests are partially correlated. That is, by letting the first, third and the forth nibbles take every possible value in $\mathbb{F}_2^{12}$, $(2^8 - 1)$ collisions were produced, which indicates that only four bit in $(\hat{K}_6[0], ..., \hat{K}_6[3], \hat{K}_6[8], ..., \hat{K}_6[15])$ or the linear combination of these boolean variables can be uniquely determined by the given DS. To verify, we searched and found 15 such linear combinations and exact 4 out of them are linear independent, i.e.,

  $$K_6[10] + K_6[12], K_6[0] + K_6[2] + K_6[8] + K_6[10] + K_6[14],$$
  $$K_6[11] + K_6[13], K_6[1] + K_6[3] + K_6[9] + K_6[11] + K_6[15]$$
- *Test 4*: Similar as *Test 2* except that we fixed $(\hat{K}_5, \hat{K}_6) = (K_5, K_6)$, tried each $(\hat{K}_7, \hat{K}_8) \in \mathbb{F}_2^{32}$ and recorded the produced differential sequences if it equals to

our template sequence. As a result, no collisions happen, which suggests that there is a bijective mapping between (K_7, K_8) and the template sequence.

- In our second experiment, we generated the template sequence using the entire HB-2 encryption where we manually set $R_4 + R'_4 = H$ and $R_1 = R'_1 = 0$ after initialization and repeated *Test 2*, *Test 3* and *Test 4* above, in which the same results were obtained.

□

Table 5.2: Some Collisions Found in *Test 1* (note that each two lines in one cell produce the same differential sequence with $\theta = H$)

K_5	K_6	K_7	K_8	K_5	K_6	K_7	K_8	K_5	K_6	K_7	K_8
56196	25591	31776	28608	5259	58450	1453	4613	11931	63343	42578	25131
14891	33718	31776	28608	54154	82	1453	4613	33166	10093	42578	25131
51427	52439	10383	5331	5258	3316	4909	32243	53472	43303	49735	58385
23521	58583	10383	5331	48136	5207	4909	32243	33527	55605	49735	58385
10513	23720	56266	56842	21555	37335	23804	39219	49708	46282	3833	21975
6945	18616	56266	56842	24209	32835	23804	39219	46153	46298	3833	21975
1947	12346	8764	37884	49115	38176	16609	58858	48786	4037	39048	46879
22460	37114	8764	37884	52169	13584	16609	58858	47248	56834	39048	46879
30286	45969	7858	39360	18687	51858	1745	62104	13278	564	27789	61251
55846	12209	7858	39360	25518	20129	1745	62104	16682	564	27789	61251
27611	21643	30999	59904	2273	52313	24422	24654	62681	20745	30138	18495
28496	17679	30999	59904	31137	58553	24422	24654	45269	12714	30138	18495
7789	33156	61918	11461	12411	58171	61105	32750	48030	19892	18970	2504
29301	6012	61918	11461	14459	35088	61105	32750	11738	34197	18970	2504
27117	24491	13908	1472	7520	3834	64704	42225	14156	38680	26568	47195
8623	32649	13908	1472	3552	3706	64704	42225	62750	38680	26568	47195
52790	32165	7784	55526	21946	63491	64374	57055	63324	9832	3806	11746
56139	14124	7784	55526	19353	21616	64374	57055	41204	1381	3806	11746
57427	56037	36359	7924	37371	25813	58822	38579	53894	55496	48025	29509
48306	58289	36359	7924	6120	5960	58822	38579	56300	49825	48025	29509

It is worth mentioning that the construction of the template differential sequence begins with the assumption that R_1 and R'_1 are known, which may be realized by guessing during the attacking. Therefore, to confirm that whether the differential sequence produced with wrong guesses of R_1 and R'_1 suggests candidate keys is a necessary task.

Table 5.3: Correlations Between the Key and the Differential Sequence

	Set to be correct	Try every of	Collisions	Result
Test 2	$\hat{K}_6, \hat{K}_7, \hat{K}_8$	$\hat{K}_5$	$2^{16} - 1$	$\hat{K}_5$ is uncorrelated
Test 3	$\hat{K}_5, \hat{K}_7, \hat{K}_8$	$\hat{K}_6$	$2^{12} - 1$	part of $\hat{K}_6$ is correlated
	rest of $\hat{K}_6$	$\hat{K}_6[0], ..., \hat{K}_6[3]$	0	
	rest of $\hat{K}_6$	$\hat{K}_6[4], ..., \hat{K}_6[7]$	$2^4 - 1$	$(\hat{K}_6[4], ..., \hat{K}_6[7])$ is uncorrelated
	rest of $\hat{K}_6$	$\hat{K}_6[8], ..., \hat{K}_6[11]$	0	
	rest of $\hat{K}_6$	$\hat{K}_6[12], ..., \hat{K}_6[15]$	$2^2 - 1$	
	$\hat{K}_6[4], ..., \hat{K}_6[7]$	rest of $\hat{K}_6$	$2^8 - 1$	$(\hat{K}_6[0], ..., \hat{K}_6[3], \hat{K}_6[8], ..., \hat{K}_6[15])$ is partially correlated
Test 4	$\hat{K}_5, \hat{K}_6$	$\hat{K}_7, \hat{K}_8$	0	$(\hat{K}_7, \hat{K}_8)$ is correlated

Observation 1 *Given* $\Delta_{H,(K_5,...,K_8)} = [z_0, z_1, ..., z_{2^{16}-1}]$*, where* $z_i = |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus R_1) + (C' \boxminus R_1') = i\}|$ *and* $\hat{\Delta}_{H,(K_5,...,K_8)} = [z_0, z_1, ..., z_{2^{16}-1}]$*, where* $z_i = |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus \hat{R}_1) + (C' \boxminus \hat{R}_1') = i\}|$*, we have, if* $R_1 \neq \hat{R}_1$ *and* $R_1' \neq \hat{R}_1'$*, or, if* $R_1 \neq \hat{R}_1 + H$ *and* $R_1' \neq \hat{R}_1' + H$*,*

$$\Delta_{H,(K_5,...,K_8)} \neq \hat{\Delta}_{H,(K_5,...,K_8)}.$$

This indicates that the wrong guess of R_1 and R_1' actually randomize $\Delta_{H,(K_5,...,K_8)}$. For example, let $C = $ 0x2174 and $C' = $ 0x2176, whose XOR difference is 0x2 while $R_1 = 0$ and $R_1' = 0$. However, when $R_1 = $ 0x3245 and $R_1' = $ 0x3245, we have $C \boxminus R_1 = 0\text{x}ef2f$ and $C' \boxminus R_1' = 0\text{x}ef31$, whose XOR difference is $0\text{x}1e$. Another example is $C = 0\text{x}9e7c$ and $C' = 0\text{x}9e7e$, whose XOR difference is 0x2 while $R_1 = 0$ and $R_1' = 0$. However, when $R_1 = $ 0x3245 and $R_1' = $ 0x3245, $C \boxminus R_1 = 0\text{x}6c37$ and $C' \boxminus R_1' = 0\text{x}6c39$, whose XOR difference is $0\text{x}e$. In addition, $\Delta_{H,(K_5,...,K_8)} = \hat{\Delta}_{H,(K_5,...,K_8)}$ when $R_1 = \hat{R}_1 + H$ and $R_1' = \hat{R}_1' + H$, since

$$\begin{aligned}
\Delta_{H,(K_5,...,K_8)} &= |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus R_1) + (C' \boxminus R_1') = i\}| \\
&= |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus R_1 + H) + (C' \boxminus R_1' + H) = i\}| \\
&= |\{P \in \mathbb{F}_2^{16}|\ (C \boxminus \hat{R}_1) + (C' \boxminus \hat{R}_1') = i\}| \\
&= \hat{\Delta}_{H,(K_5,...,K_8)}.
\end{aligned}$$

Attacking $WD16^{-1}$ **in Decryption**: Similar attack can be mounted to the decryption: we still assume that R_1 and R_1' are known. In addition, let h in Definition 1 be the last invocation of $WD16^{-1}$, i.e., Eq. (5.28), in the decryption, and recall that the condition (B) is $f^{-1}(R_2 \boxminus u_1) + f^{-1}(R'_2 \boxminus u'_1) = H$, $(u_1, u_2, u_3) = (u'_1, u'_2, u'_3)$ and $(K_1, ..., K_4) = (K'_1, ..., K'_4)$. We thus have the following theorems for our attack.

Theorem 3 *With the condition (B), the differential sequence of $h(w, K) = WD16^{-1}(R_2 \boxminus u_1, K_4, K_3, K_2, K_1)$ can be computed without knowing the key K, as shown in Figure 5.2, by the following*

$$\begin{aligned}
\Delta_{H,(K_4,...,K_1)} &= [z_0, z_1, ..., z_{2^{16}-1}], \ \textit{where} \\
z_i &= |\{u_1 \in \mathbb{F}_2^{16}|\ (WD16^{-1}(R_2 \boxminus u_1, K_4, K_3, K_2, K_1) \\
&+ \ WD16^{-1}(R'_2 \boxminus u'_1, K'_4, K'_3, K'_2, K'_1)) = i\}| \\
&= |\{u_1 \in \mathbb{F}_2^{16}|\ (WD16^{-1}(R_2 \boxminus u_1, K_4, K_3, K_2, K_1) \\
&+ \ WD16^{-1}(R'_2 \boxminus u'_1, K_4, K_3, K_2, K_1)) = i\}| \\
&= |\{u_1 \in \mathbb{F}_2^{16}|\ (f^{-1}(f^{-1}(f^{-1}(f^{-1}(R_2 \boxminus u_1) + K_4) + K_3) + K_2) + K_1 \\
&+ \ f^{-1}(f^{-1}(f^{-1}(f^{-1}(R_2 \boxminus u_1) + H + K_4) + K_3) + K_2) + K_1) = i\}| \\
&= |\{C \in \mathbb{F}_2^{16}|\ (R_1 \boxminus P) + (R'_1 \boxminus P') = i\}|.
\end{aligned}$$

Proof: u_1 takes every value in $\mathbb{F}_2^{16}$ as long as C takes every value in $\mathbb{F}_2^{16}$. □

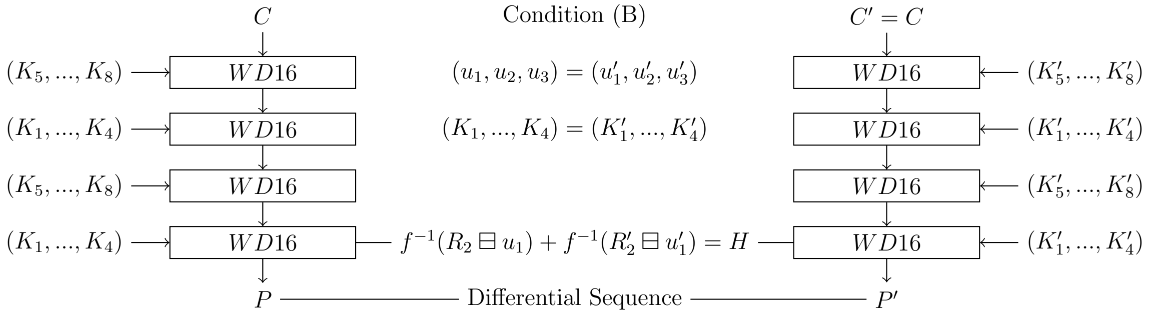

Figure 5.2: Constructing Differential Sequence from Decryption with Condition (B)

A similar theorem describes the correspondence between $\Delta_{H,(K_4,K_3,K_2,K_1)}$ and (K_4, K_3, K_2, K_1).

Theorem 4 *Let $\Delta_{H,(K_4,K_3,K_2,K_1)}$ be obtained from Theorem 3. For $K_2 \in \mathbb{F}_2^{16}$ and $\kappa_3 \in \mathbb{F}_2^{12}$,*

$$\Delta_{H,(K_4,K_3,K_2,K_1)} = \Delta_{H,(\kappa_3,K_2)},$$

where $\kappa_3 = (K_3[0], ..., K_3[3], K_3[8], ..., K_3[15])$.

This is also experimentally verified similarly as Theorem 2.

Visualization of Differential Sequences From HB-2: Here we provide several examples of the differential sequences used in our experiments. Figure 5.3 to Figure 5.7 are the ones obtained from the last invocation of $WD16$ in the encryption with $IV = (0,0,0,0)$ and different keys randomly selected. Figure 5.8 to Figure 5.12 are the ones obtained from the last invocation of $WD16^{-1}$ in the decryption with $IV = (0,0,0,0)$ and different keys randomly selected. All of the sequences looks substantially different from each other, which exhibits their correlations to the underlying keys in an intuitive way.

5.3.3 Key Recovery Phase

After exhibiting the properties of differential sequence in HB-2, we are ready to show the steps to be performed by the attacker during the key recovery phase, which compromise the entire 128-bit key.

1. When condition (A) holds, for every $R_1 \in \mathbb{F}_2^{16}$, the attacker uses $(C \boxminus R_1) + (C' \boxminus R_1')$ to construct one template differential sequence $\Delta_{H,(K_5,K_6,K_7,K_8)}$, where C and C' can be obtained by querying the encryption with P and $P' = P$. After this, the attacker has 2^{16} template sequences[1].

2. For each candidate key $(\hat{\kappa}_6, \hat{K}_7, \hat{K}_8) \in \mathbb{F}_2^{36}$, the attacker computes, in an off-line environment, $\Delta_{H,(\hat{\kappa}_6,\hat{K}_7,\hat{K}_8)}$ using $WD16(w, K_5, K_6, K_7, K_8), w \in \mathbb{F}_2^{16}$, and check if it matches one of the template sequences. If so, a key candidate is found due to Theorem 2 and Observation 1. At the end, the attacker would have the correct (κ_6, K_7, K_8).

3. Similarly, utilizing the decryption, when condition (B) holds, for every $R_1 \in \mathbb{F}_2^{16}$, the attacker uses $(R_1 \boxminus P) + (R_1' \boxminus P')$ to construct another template sequence $\Delta_{H,(K_4,K_3,K_2,K_1)}$, and guesses to determine (K_2, κ_3) using $WD16^{-1}$ in an off-line environment.

4. After that, the attacker searches for the rest of the key using 2^{64} trial encryptions.

[1] Note that $R_1 = R_1'$ is implicitly assumed in the condition (A) and achieved during the preparation phase.

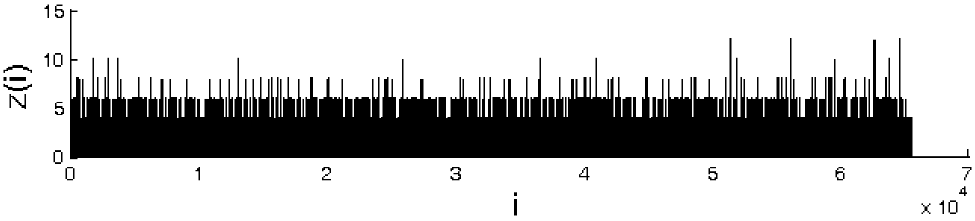

Figure 5.3: DS from Enc. using $(K_5, K_6, K_7, K_8) = (0\mathrm{x}f1e3, 0\mathrm{x}524a, 0\mathrm{x}b28a, 0\mathrm{x}c987)$

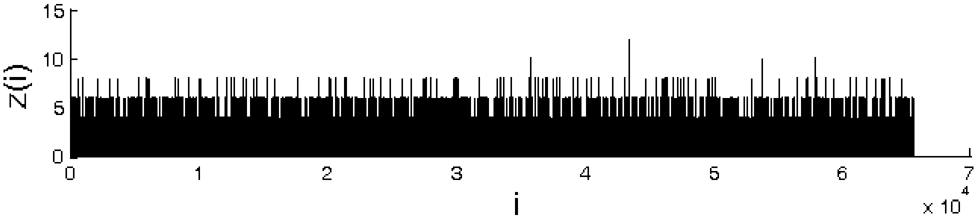

Figure 5.4: DS from Enc. using $(K_5, K_6, K_7, K_8) = (0\mathrm{x}7c9f, 0\mathrm{x}0784, 0\mathrm{x}1c96, 0\mathrm{x}bcb4)$

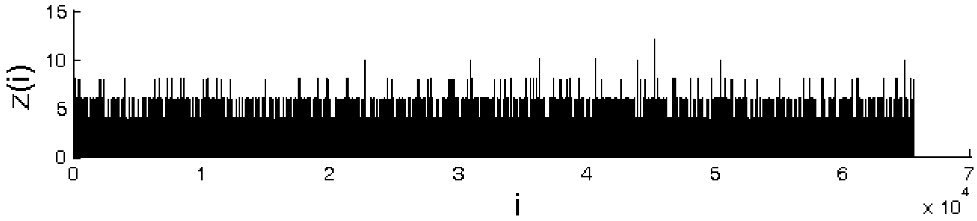

Figure 5.5: DS from Enc. using $(K_5, K_6, K_7, K_8) = (0\mathrm{x}6b03, 0\mathrm{x}cf0c, 0\mathrm{x}1ba2, 0\mathrm{x}dc27)$

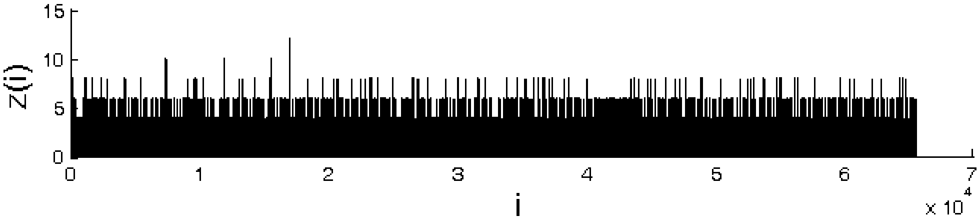

Figure 5.6: DS from Enc. using $(K_5, K_6, K_7, K_8) = (0\mathrm{x}2602, 0\mathrm{x}cb5a, 0\mathrm{x}ab7c, 0\mathrm{x}f56b)$

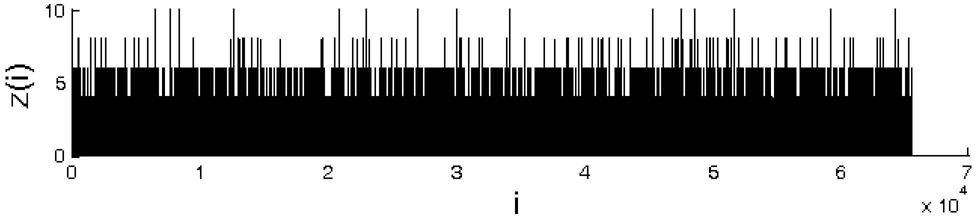

Figure 5.7: DS from Enc. using $(K_5, K_6, K_7, K_8) = (0\mathrm{x}f4c7, 0\mathrm{x}920f, 0\mathrm{x}1fbf, 0\mathrm{x}0c38)$

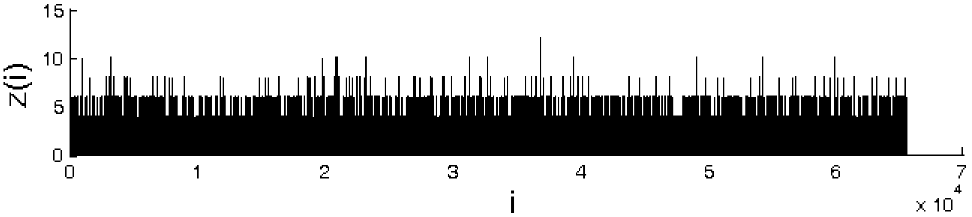

Figure 5.8: DS from Dec. using $(K_1, K_2, K_3, K_4) = (0\text{x}5d67, 0\text{x}d0ef, 0\text{x}8cec, 0\text{x}a33a)$

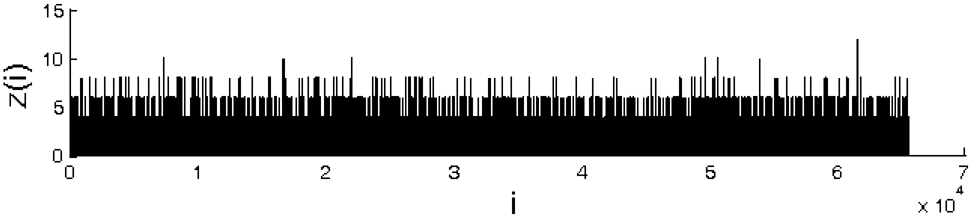

Figure 5.9: DS from Dec. using $(K_1, K_2, K_3, K_4) = (0\text{x}6601, 0\text{x}0bd8, 0\text{x}a6fa, 0\text{x}cede)$

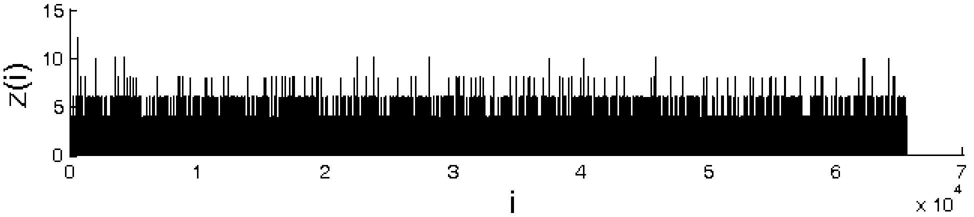

Figure 5.10: DS from Dec. using $(K_1, K_2, K_3, K_4) = (0\text{x}28dc, 0\text{x}bde1, 0\text{x}6e3d, 0\text{x}a56d)$

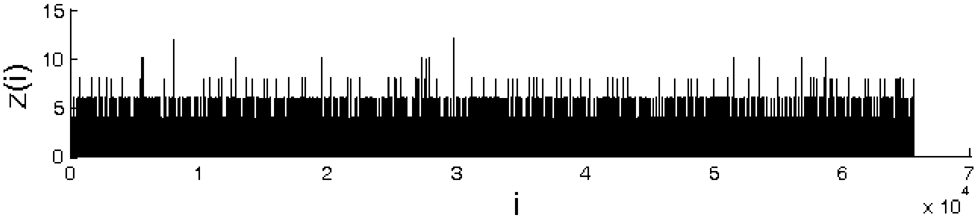

Figure 5.11: DS from Dec. using $(K_1, K_2, K_3, K_4) = (0\text{x}1927, 0\text{x}8f7d, 0\text{x}a928, 0\text{x}27e3)$

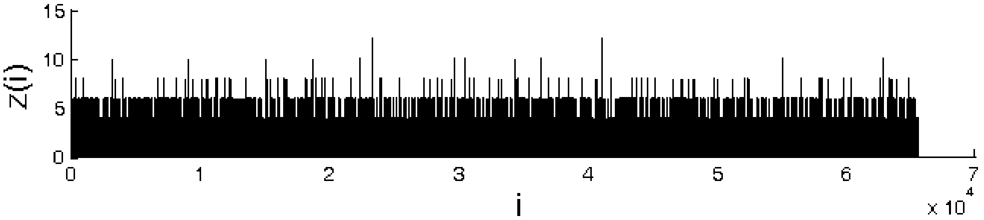

Figure 5.12: DS from Dec. using $(K_1, K_2, K_3, K_4) = (0\text{x}c30e, 0\text{x}aa4f, 0\text{x}5f89, 0\text{x}eb9f)$

Complexity Analysis: The overall complexity of the above process is

$$\underbrace{2^{36} \times 2^{16} \times 2^{16}}_{\text{determine } \kappa_6, K_7, K_8} + \underbrace{2^{28} \times 2^{16} \times 2^{16}}_{\text{determine } K_2, \kappa_3} + \underbrace{2^{64}}_{\text{determine the rest}} \approx 2^{68},$$

where negligible memory is required by each of the steps.

5.4 Preparation: Guess and Determine of the Conditions

The attack shown in the last section is based on condition (A) and condition (B), which sounds unpractical at the first glance as the initialization of HB-2 leads the internal states to be unpredictable. In this section, we realize these two conditions through a probabilistic approach – when the internal states of two HB-2 instances are respectively random, there is a certain chance that the attacker could get the desired differentials in the internal states. To this end, we study: (1) how to randomize the internal states of HB-2; (2) how to transfer condition (A) and condition (B) to be necessary to another two conditions; (3) how to determine whether those sufficient conditions happen in the two instances of HB-2. The reason that we study (2) is that we want to develop a better algorithm for (3).

5.4.1 Randomize the Internal States

There are two ways for the adversary to affect the internal states of HB-2:

- Providing the key is fixed, it is suffice, from Eq. (5.1)-(5.12), that $(IV_1, ..., IV_4) \mapsto (R_1, ..., R_4)$ is a one-to-one mapping as well as $(IV_1, ..., IV_4) \mapsto (R_5, ..., R_8)$. Therefore, the attacker could easily generate 2^{64} (out of 2^{128}) different internal states by choosing different IVs and launching the initialization.
- For a fixed key and a particular IV, the attacker could choose plaintext P_1 to feed HB-2 at the first round. As verified by our testing, 2^{16} different internal states are generated by inputting each $P_1 \in \mathbb{F}_2^{16}$. If a state transition graph is drawn, we can see that the starting state, i.e., $R^{(1)}$ (there could be 2^{64} such a starting state by choosing IVs), transits to 2^{16} neighboring states equally likely. Next, if another round of encryption is performed, e.g., encrypting P_2, each of these "neighboring states" again transits to another 2^{16} states providing P_2 takes every value in $\mathbb{F}_2^{16}$. By continuing this process,

we would have all 2^{128} states covered in this graph. Therefore, to produce a set of random internal states, i.e., $\{R^{(1)}, R^{(2)}, ...\}$, we could, as shown in Figure 5.13, feed the encryptions with a plaintext chain where P_i is selected uniformly at random in $\mathbb{F}_2^{16}$ for $i = 1, 2, ...$ Similarly, a ciphertext chain could be fed to the decryption oracle to generate a set of random internal states as well. Note that feeding HB-2 encryption with a chain of N random inputs is equivalent to perform an N-step 2^{16}-dimensional random walk in its state transition graph, and $|\{R^{(1)}, R^{(2)}, ...\}| \approx N$, given that $N \ll 2^{128}$; otherwise, when N approaches infinity, $(|\{R^{(1)}, R^{(2)}, ...\}|/N)$ converges in measure to a small constant due to [66] since the random walk here is transient.

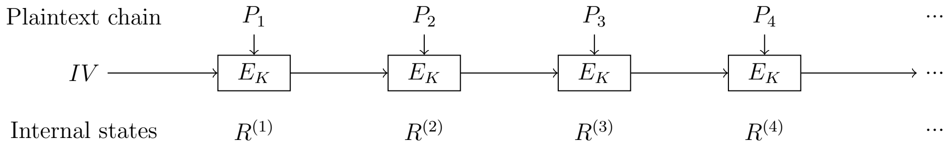

Figure 5.13: Feeding HB-2 Encryption with a Plaintext Chain

Furthermore, let $\Delta R \in \mathbb{F}_2^{128}$ be a given difference in the internal states, and let $R^{(i)} \Leftarrow E(P_i, K)$ represent that the internal state after encrypting the plaintext chain $(P_1, ..., P_i)$ is $R^{(i)}$, it thus follows from the birthday paradox that:

Property 3 *Given the following algorithm, a certain ΔR happens with* 0.5 *probability when $N = 2^{64}$.*

```
Randomly choose $IV'$ and $P'_1$, $R'^{(1)} \Leftarrow E(P'_1, K)$
Randomly choose $IV$
for $i$ from 1 to $N$ do
    Randomly choose $P_i$, $R^{(i)} \Leftarrow E(P_i, K)$
    if $R'^{(1)} + R^{(i)} = \Delta R$ then
        return "$\Delta R$ happens"
    end if
end for
```

The above algorithm provides, to the later process, the randomized differences in the internal states of two running HB-2 instances through an effort-saving way – one instance initializes a random IV and encrypts one plaintext, while the other one, besides initializes a random IV, encrypts N plaintexts in a chain. Since $\{R^{(1)}, R^{(2)}, ..., R^{(N)}\}$ is a set of random variables as analyzed, $\{R^{(1)}+R'^{(1)}, R^{(2)}+R'^{(1)}, ..., R^{(N)}+R'^{(1)}\}$ must also be a set of random variables.

5.4.2 Transferring of the Conditions

At the first glance, the above algorithm will automatically lead to the occasion of condition (A) or condition (B). Unfortunately, the adversary in fact is unable to make the decision whether $R'^{(1)} + R^{(i)} = \Delta R$ is true. Therefore, we need to find the sufficient conditions for condition (A) and condition (B) respectively, which are, by taking advantage of another design flaw of HB-2, detectable by the adversary.

Theorem 5 *Condition (A) is satisfied if the following meets,*

$$\begin{array}{lrcl} \textit{Condition } (A^+): & \Delta K & = & (K'_1, ..., K'_8) + (K_1, ..., K_8) = (H, 0, 0, 0, H, 0, 0, 0) \\ & \Delta P & = & P'_{1'} + P_i = H \\ & \Delta R & = & (R'^{(1)}_1, ..., R'^{(1)}_8) + (R^{(i)}_1, ..., R^{(i)}_8) = (0, 0, 0, 0, H, 0, 0, 0). \end{array}$$

Condition (B) is satisfied if the following meets,

$$\begin{array}{lrcl} \textit{Condition } (B^+): & \Delta K & = & (K'_1, ..., K'_8) + (K_1, ..., K_8) = (0, 0, 0, H, 0, 0, 0, H) \\ & \Delta C & = & C'_1 + C_i = H \\ & \Delta R & = & (R'^{(1)}_1, ..., R'^{(1)}_8) + (R^{(i)}_1, ..., R^{(i)}_8) = (0, 0, 0, 0, 0, 0, 0, H). \end{array}$$

Proof: This is because

$$\begin{array}{rcl} t'_1 & = & WD16(R'^{(1)}_1 \boxplus P'_1, K'_1, K'_2, K'_3, K'_4) \\ & = & WD16(R^{(i)}_1 \boxplus (P_i + H), (K_1 + H), K_2, K_3, K_4) = t_1 \\ t'_2 & = & WD16(R'^{(1)}_2 \boxplus t'_1, K'_5 + R'^{(1)}_5, K'_6 + R'^{(1)}_6, K'_7 + R'^{(1)}_7, K'_8 + R'^{(1)}_8) \\ & = & WD16(R^{(i)}_2 \boxplus t_1, (K_5 + H) + (R^{(i)}_5 + H), K_6 + R^{(i)}_6, K_7 + R^{(i)}_7, K_8 + R^{(i)}_8) = t_2 \\ t'_3 & = & WD16(R'^{(1)}_3 \boxplus t'_2, K'_1 + R'^{(1)}_5, K'_2 + R'^{(1)}_6, K'_3 + R'^{(1)}_7, K'_4 + R'^{(1)}_8) \\ & = & WD16(R^{(i')}_3 \boxplus t_2, (K_1 + H) + (R^{(i)}_5 + H), K_2 + R^{(i)}_6, K_3 + R^{(i)}_7, K_4 + R^{(i)}_8) = t_3 \\ C'_1 & = & WD16(R'^{(1)}_4 \boxplus t'_3, K'_5, K'_6, K'_7, K'_8) \boxplus R'^{(1)}_1 \\ & = & f(f(f(f(R'^{(1)}_4 \boxplus t'_3 + K'_5) + K'_6) + K'_7) + K'_8) \boxplus R'^{(1)}_1 \\ C_i & = & WD16(R^{(i)}_4 \boxplus t_3, K_5, K_6, K_7, K_8) \boxplus R^{(i)}_1 \\ & = & f(f(f(f((R'^{(1)}_4 + H) \boxplus t'_3 + K'_5) + K'_6) + K'_7) + K'_8) \boxplus R'^{(1)}_1. \end{array}$$

On the other hand,

$$\begin{array}{rcl}
u'_3 & = & WD16^{-1}(C'_1 \boxminus R'^{(1)}_1, K'_8, K'_7, K'_6, K'_5) \\
 & = & WD16^{-1}((C_i + H) \boxminus R^{(i)}_1, (K_8 + H), K_7, K_6, K_5) = u_3 \\
u'_2 & = & WD16^{-1}(R'^{(1)}_4 \boxminus u'_3, K'_4 + R'^{(1)}_8, K'_3 + R'^{(1)}_7, K'_2 + R'^{(1)}_6, K'_1 + R'^{(1)}_5) \\
 & = & WD16^{-1}(R^{(i)}_4 \boxminus u_3, (K_4 + H) + (R^{(i)}_8 + H), K_3 + R^{(i)}_7, K_2 + R^{(i)}_6, K_1 + R^{(i)}_5) = u_2 \\
u'_1 & = & WD16^{-1}(R'^{(1)}_3 \boxminus u'_2, K'_8 + R'^{(1)}_8, K'_7 + R'^{(1)}_7, K'_6 + R'^{(1)}_6, K'_5 + R'^{(1)}_5) \\
 & = & WD16^{-1}(R^{(i)}_3 \boxminus u_2, (K_8 + H) + (R^{(i)}_8 + H), K_7 + R^{(i)}_7, K_6 + R^{(i')}_6, K_5 + R^{(i)}_5) = u_1 \\
P'_1 & = & R'^{(1)}_1 \boxminus (f^{-1}(f^{-1}(f^{-1}(f^{-1}(R'^{(1)}_2 \boxminus u'_1) + K'_4) + K'_3) + K'_2) + K'_1) \\
P_i & = & R^{(i)}_1 \boxminus (f^{-1}(f^{-1}(f^{-1}(f^{-1}(R^{(i)}_2 \boxminus u_1) + K_4) + K_3) + K_2) + K_1) \\
 & = & R'^{(1)}_1 \boxminus (f^{-1}(f^{-1}(f^{-1}((f^{-1}(R^{(1)}_2 \boxminus u'_1) + H) + K'_4) + K'_3) + K'_2) + K'_1).
\end{array}$$

□

The above theorem states that as long as condition (A$^+$) in the encryption (condition (B$^+$) in the decryption resp.) is realized, condition (A) (condition (B) resp.) is essentially achieved. Note that the sufficient conditions above introduce the need of related-keys to our attack.

5.4.3 Determining while Guessing

To inform the attacker during the attempting, as long as condition (A$^+$) or condition (B$^+$) happens, we use one special differential characteristics in the encryption first pointed out by HB-2's designers, as shown in the last row of Table 5.4. The specialty of this differential is its *time-invariance*, by which we mean that the differential in the internal states/keys/inputs can be maintained and entered into the next round. In addition, we found similar differential characteristics during decryption as listed in Table 5.5, where the last one is also time-invariant. A nice observation here is that our condition (A$^+$) (condition (B$^+$) resp.) is exactly the last row of Table 5.4 (Table 5.5 resp.), which is time-invariant. Therefore, we have the following theorem.

Theorem 6 *Let $\Delta^{(i)}_{H,(K_5,K_6,K_7,K_8)}$ ($\Delta^{(i)}_{H,(K_4,K_3,K_2,K_1)}$ resp.) be the differential sequence produced by the two encryption instances (two decryption instances resp.) with internal states $(R'^{1}_1, ..., R'^{1}_8)$ and $(R^{(i)}_1, ..., R^{(i)}_8)$ and let $\Delta^{(i+1)}_{H,(K_5,K_6,K_7,K_8)}$ ($\Delta^{(i+1)}_{H,(K_4,K_3,K_2,K_1)}$ resp.) be the differential sequence produced by the two encryption instances (two decryption instances resp.) with internal states $(R'^{2}_1, ..., R'^{2}_8)$ and $(R^{(i+1)}_1, ..., R^{(i+1)}_8)$. Thus,*

Table 5.4: Differentials in Encryption (pointed out in [84])

ΔK	Current Round ΔP	 ΔR	Next Round ΔR	Time-invariant
0	0	$(0,0,0,H,0,0,0,0)$	$(0,0,0,H,0,0,0,H)$	No
0	H	$(H,0,0,H,0,0,0,0)$	$(H,0,0,0,H,0,0,0)$	No
0	H	$(H,H,H,H,H,0,0,0)$	$(H,H,H,0,0,H,H,0)$	No
$(H,0,0,0,H,0,0,0)$	H	$(0,0,0,0,H,0,0,0)$	$(0,0,0,0,H,0,0,0)$	Yes

Table 5.5: Differentials in Decryption

ΔK	Current Round ΔC	 ΔR	Next Round ΔR	Time-invariant
0	0	$(0,H,0,0,0,0,0,0)$	$(0,H,0,0,0,H,0,0)$	No
0	H	$(H,H,0,0,0,0,0,0)$	$(H,H,0,H,H,H,0,H)$	No
0	H	$(H,0,0,0,0,0,0,0)$	$(H,0,0,H,0,0,0,H)$	No
$(0,0,0,H,0,0,0,H)$	H	$(0,0,0,0,0,0,0,H)$	$(0,0,0,0,0,0,0,H)$	Yes

- *If condition (A$^+$) happens at the ith round of the encryption, the adversary observes*

$$\Delta^{(i)}_{H,(K_5,K_6,K_7,K_8)} = \Delta^{(i+1)}_{H,(K_5,K_6,K_7,K_8)};$$

otherwise, the above equation holds with negligible probability.

- *If condition (B$^+$) happens at the ith round of the decryption, the adversary observes*

$$\Delta^{(i)}_{H,(K_4,K_3,K_2,K_1)} = \Delta^{(i+1)}_{H,(K_4,K_3,K_2,K_1)};$$

otherwise, the above equation hold with negligible probability.

Proof: It follows from Property 1 and Theorem 7. □

Therefore, the above theorem can be served as an algorithm to determine the occasion of condition (A$^+$) or condition (B$^+$). Each call of this algorithm requires $2^{16} + 2^{16} = 2^{17}$ efforts, e.g., to produce two differential sequences using two consecutive rounds, and returns a boolean result.

5.4.4 Preparation Phase

We recap the whole process in the preparation phase for the encryption as shown below, which is an algorithm inherited from the one goes with Property 3, where the determination of the occasion of $R'^{(1)} + R^{(i)} = \Delta R$ is replaced by our new technology. Note that a similar preparation for the decryption is omitted here.

```
Randomly choose $IV'$ and $P'_1$, $R'^{(1)} \Leftarrow E(P'_1, K)$
Randomly choose $IV$
for $i$ from 1 to $N = 2^{64}$ do
    Randomly choose $P_i$, $R^{(i)} \Leftarrow E(P_i, K)$
    Generate $\Delta^{(i)}$ using $R'^{(1)}$ and $R^{(i)}$
    Randomly choose $P'_2$, $R'^{(2)} \Leftarrow E(P'_2, K)$
    Randomly choose $P_{i+1}$, $R^{(i+1)} \Leftarrow E(P_{i+1}, K)$
    Generate $\Delta^{(i+1)}$ using $R'^{(2)}$ and $R^{(i+1)}$
    if $\Delta^{(i)} = \Delta^{(i+1)}$ then
        return "condition (A) happens", keep current states and enter the key recovery
        phase
    end if
    Decrypt using $C'_2$ and $C_{i+1}$ to roll back HB-2's states to $R'^{(1)}$ and $R^{(i)}$
end for
```

Complexity Analysis: Using the encryption only, the attacker has at least 0.5 probability to prepare condition (A) for the DSA with $2^{64} \times 2^{17} = 2^{81}$ time complexity. Similarly, using the decryption only, the attacker has 0.5 probability to prepare condition (B) for the DSA with $2^{64} \times 2^{17} = 2^{81}$ time complexity as well. This probability can be increased if the attacker is willing to pay more on computation and vice versa.

The complexity comparison of our technique and exhaustive search with and without time/memory tradeoff are listed in Table 5.6. Since one encryption $\mathbb{F}_2^{16} \mapsto \mathbb{F}_2^{16}$ only provides 16-bit entropy of the key, $2^{128} \times 8$ calls of encryption could uniquely determine the key with probability 1. Once time/memory tradeoff is introduced, the attacker pre-computes a table of 2^{64} entries using a fixed chain of plaintexts and 2^{64} keys selected randomly in an offline environment, and, tries to check the given plaintext/ciphertext pair corresponds to one of the keys in the table. However, in the worst case, the attacker still has to investigate $2^{64} \times 8$ in time to success.

Table 5.6: Complexity Comparison

Type	Time complexity [encryption]	Data complexity [bit]	Success probability
Exhaustive search	$2^{128} \times 8$	1	1
Time/memory tradeoff	$2^{64} \times 8$	$2^{64} \times 128$	1
Time/memory tradeoff	1	$2^{127} \times 128$	0.5
Our Attack	$2^{64} \times 2^{17}$	1	0.5

5.5 Conclusion

We have present a novel attack against the lightweight block cipher Hummingbird-2 and demonstrate its validity. This attack encompasses two phases: preparation and key recovery, where the first phase, using a probabilistic algorithm, creates the conditions needed by the second phase, and has 0.5 chance to be success with 2^{81} effort. Next, the 128-bit key is recovered in the second phase using a deterministic algorithm with 2^{68} time complexity. The proposed cryptanalytic results re-emphasize the following design principles of iterative block ciphers:

- Block size should be large enough to mitigate the saturation of the inputs, e.g., AES has 128-bit block size, which means the effort to saturate the plaintext and launch DSA is equal to search for the key exhaustively.
- Do not disturb, e.g., modify, the intermediate output of round functions with other variables that can be controlled/known by the attacker.

The attack presented against HB-2 is a special case of the general DSA, to build the theoretic framework of which is part of our future work. In addition, it should be evaluated: (1) whether the generalized DSA provides even better results against HB-2 and other potentially vulnerable ciphers, especially the ones with small block size; (2) the possibility that the generalized DSA can work with other cryptanalysis technologies, e.g., meet-in-the-middle. Last but not least, the proposed attacking techniques seem transplantable to attack HB-1 that has almost the same structure, the possibility of which will be investigated in the near future.

Chapter 6

An Ultra-Efficient Key Recovery Attack on the Lightweight Stream Cipher A2U2

As shown in Table 2.1 in Chapter 2, PRESENT, GOST and KATAN are the most promising candidate in the family of lightweight ciphers because of their compact hardware implementation, satisfactory throughput and widely-accepted security after extensive cryptanalysis. More recently, David *et al.* [73] proposed a stream cipher called A2U2, which achieves, in terms of its hardware footprint of 284 GE, amazing lightweightness so far. Surprisingly, the throughput of A2U2 is approximately five times greater than that of PRESENT, KTANTAN-32 and PRINTCipher. The security analysis performed by the designers of A2U2 shows: (1) the output sequence of the A2U2 can pass the NIST's statistical tests for pseudorandom number generators; (2) the period of the output sequence is around 2^{70}; and (3) particular attacks are thwarted since variable number of clock cycles used for the initialization ensures that the cipher outputs different ciphertexts for identical plaintexts.

In this chapter, we investigate the security of the lightweight stream cipher A2U2. Our cryptanalytic results show that the A2U2 is completely broken and is insecure under a simple chosen-plaintext attack, which enables the full key recovery of the A2U2 through two encryptions with particular plaintexts of 653 bits and solving 32 sparse systems of linear equations (where each system has 56 unknowns and, among them, around 28 unknowns can be directly obtained without computation) with around 0.16 second on a Thinkpad T410 laptop.

6.1 A2U2: A Lightweight Stream Cipher

As illustrated in Figure 6.1, the stream cipher A2U2 is composed of four building blocks: a 7-stage LFSR, a combination of two NFLRs, a key schedule module, and a filter function. As always, we use + to denote an XOR operation, $\odot$ an NAND operation, and $\cdot$ an AND operation.

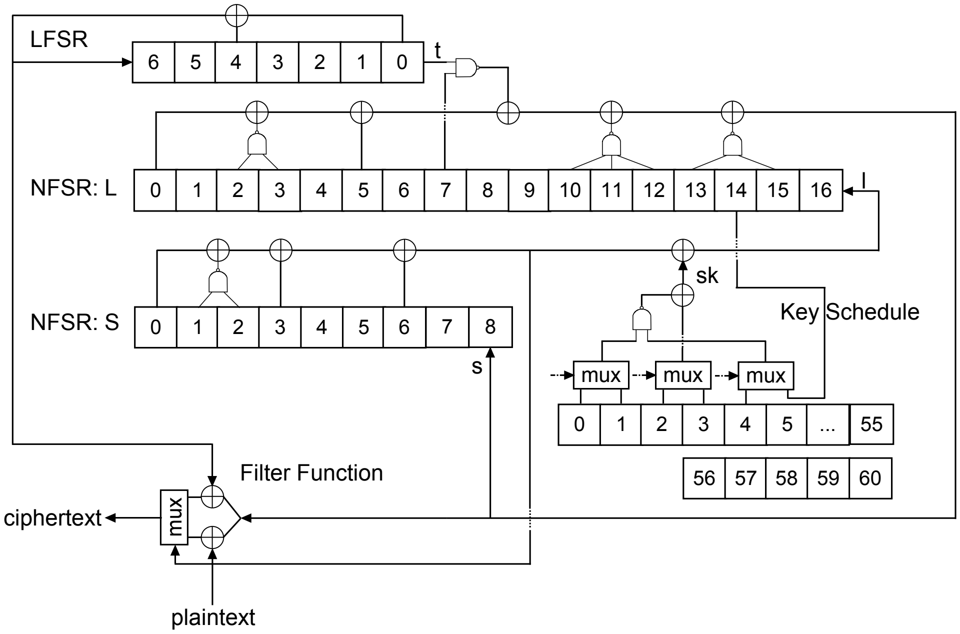

Figure 6.1: Architecture of the Stream Cipher A2U2

After a 61-bit secret key $(k_0, ..., k_{60})$ is burnt into an RFID tag, the tag is capable of encrypting plaintext bits $(p_\delta, \ldots, p_n)$ to the corresponding ciphertext bits $(c_\delta, \ldots, c_n)$, where δ is the number of clock cycles required to initialize the internal state of the A2U2 and is determined by the following steps:

- Step 1. The RFID reader and the tag generate and exchange two 32-bit random numbers $RND_R = (a_0, \ldots, a_{31}) \in \mathbb{F}_2^{32}$ and $RND_T = (b_0, \ldots, b_{31}) \in \mathbb{F}_2^{32}$, respectively.
- Step 2. The value $(RND_R + RND_T)$ is then loaded into the LFSR and two NFSRs of the A2U2 cipher.
- Step 3. The LFSR and two NFSRs run δ clock cycles for initialization without any

output until the state of the LFSR reaches all ones. From the next clock cycle, the stream cipher A2U2 outputs the ciphertext bits.

The building blocks of the stream cipher A2U2 are further detailed as below.

One LFSR: The LFSR has seven stages denoted by a binary vector $(t_{i+6}, \ldots, t_i) \in \mathbb{F}_2^7, i = 0, \ldots, n$. Moreover, the following recursive relation holds:

$$t_{i+7} = t_i + t_{i+4}, \text{ for } i = 0, \ldots, n.$$

Note that the generated sequence is an m-sequence [95] with the maximum period $2^7 - 1 = 127$. In the step 2 above, $((a_0, \ldots, a_4) + (b_0, \ldots, b_4) + (k_{56}, \ldots, k_{60}))$ is loaded into $(t_4, \ldots, t_0)$ of the LFSR, while leaving t_5 to be a constant one and t_6 to be a constant zero.

Two NFSRs: This block borrows part of the design from KATAN/KTANTAN [64]. For $i = 0, \ldots, n$, the feedback functions can be represented as follows:

$$\begin{aligned} s_{i+8} &= l_i + l_{i+2} \odot l_{i+3} + l_{i+5} + l_{i+7} \odot t_{i+6} \\ &\quad + l_{i+10} \odot l_{i+11} \odot l_{i+12} + l_{i+13} \odot l_{i+15}, \\ l_{i+16} &= s_i + s_{i+1} \odot s_{i+2} + s_{i+3} + s_{i+6} + sk_i, \end{aligned} \tag{6.1}$$

where sk_i is the subkey bit generated by the key schedule.

Key Schedule: This module derives a sequence of subkeys $(sk_0, \ldots, sk_n)$ from a portion of the secret key $(k_0, ..., k_{55})$, where sk_i is used by the NFSR during the ith clock cycle and is computed as follows:

$$\begin{aligned} sk_i &= mux(k_{mod(5i,56)}, k_{mod(5i+1,56)}, t_{i+1}) \odot mux(k_{mod(5i+4,56)}, l_{i+14}, t_{i+5}) + \\ &\quad mux(k_{mod(5i+2,56)}, k_{mod(5i+3,56)}, t_{i+3}), \end{aligned} \tag{6.2}$$

where $mod(x, y)$ returns x modulo y provided that x, y are non-negative integers and $mux()$ is a multiplexer such that, given $x, y, z \in \{0, 1\}$, $mux(x, y, z) = x$ iff $z = 0$ or $mux(x, y, z) = y$ iff $z = 1$.

Filter Function: The filter function is essentially a variant of the *shrinking generator* [54], which replaces the XOR operation of keystream bits and plaintext bits in a classical stream cipher. Regardless of its multiplexer-based implementation, the filter function can be simply written as (see Eq. (6) in [73]):

$$\textbf{Case I}: c_i = \begin{cases} s_{i+8} + t_i, & \text{if } l_{i+16} = 0, \\ s_{i+8} + p_i, & \text{if } l_{i+16} = 1, \end{cases} \quad \text{for } i = \delta, \ldots, n, \tag{6.3}$$

where p_i is the plaintext bit fed into the A2U2 cipher during the ith clock cycle, and c_i is the corresponding ciphertext bit. Let us denote the above expression of c_i as "**Case I of** $\mathbf{c}_i$".

It is worth to point out that we were recently informed by Mohamed Ahmed Abdelraheem, Julia Borghoff and Erik Zenner that the initial intention of the authors of [73] is in fact to construct a multiplexer (as shown below) behaving like a *stop-and-go sequence generator* [36], which is different from the descriptions of A2U2 in [73]. Let us call the below expression of c_i as "**Case II of** $\mathbf{c}_i$". Nevertheless, as shown later, our attack is applicable to both designs.

$$\textbf{Case II}: c_i = \begin{cases} s_{i+8} + t_i, & \text{if } l_{i+16} = 0, \\ s_{i+8} + p_{f(i)}, & \text{if } l_{i+16} = 1, \end{cases} \quad \text{for } i = \delta, \ldots, \hat{n} \tag{6.4}$$

where $f(i) = \delta + \sum_{j=\delta+16}^{i+16} l_j, \delta \leq i \leq n$, $f(\delta) = \delta$ and $\hat{n}$ equals the sum of n and the number of inserted bits from the m-sequence.

6.2 An Ultra-Efficient Key Recover Attack

6.2.1 Adversary Model

We consider the classical chosen-plaintext attack against the stream cipher A2U2 that is implemented on an RFID tag with a fixed and high-entropy 61-bit secret key burnt inside. An attacker, equipped with a programmable RFID reader, queries the victim tag with a particular random number RND_R and plaintext bits $p_\delta, ..., p_n$. The attacker's goal is to recover the secret key $(k_0, ..., k_{60})$. Note that in our attack, the reader, manipulated by the attacker, can always adaptively choose RND_R to make $(RND_R + RND_T)$ a constant, e.g., $RND_R + RND_T = 0$.

6.2.2 Step 1: Recover Sequences of s_{i+8} and l_{i+16}

A cryptographic primitive is only as strong as its weakest module, which is the general principle to break a cryptosystem. We noticed that the filter function in the A2U2 is quite weak, which enables an attacker to easily recover the internal state of the NFSRs by the following steps.

1. At the δth clock cycle[1], the LFSR reaches the all-one state and the A2U2 starts the ciphertext output. As a result, the attacker knows the sequence $(t_\delta, t_{\delta+1}, ..., t_n)$, which is just a repetition of the m-sequence with period 127 as shown in Table 6.1.

Table 6.1: One Period of $(t_\delta, t_{\delta+1}, ..., t_n)$

1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0,
0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0,
1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0.

2. **Case I of $\mathbf{c}_i$**: The attacker first chooses the plaintext bits

$$(p_\delta, \ldots, p_n) = (t_\delta, t_{\delta+1}, \ldots, t_n),$$

and sends them to the tag for encryption. Since p_i and t_i are equal for $i = \delta, \ldots, n$, Eq. (6.3) now becomes

$$c_i = s_{i+8} + t_i, \text{ for } i = \delta, \ldots, n.$$

Consequently, the variable l_i that controls the multiplexer is nullified and s_{i+8} can be recovered (i.e., both c_i and t_i are known to the attacker). Next, the attacker chooses a new set of plaintext bits

$$(p_\delta, \ldots, p_n) = (1 + t_\delta, 1 + t_{\delta+1}, \ldots, 1 + t_n),$$

and launches another encryption session with the RFID tag. In this case, Eq. (6.3) becomes

$$c_i = \begin{cases} s_{i+8} + t_i, & \text{if } l_{i+16} = 0 \\ s_{i+8} + t_i + 1, & \text{if } l_{i+16} = 1 \end{cases} \text{ for } i = \delta, \ldots, n.$$

[1] As one may expected, this cipher may also suffer from the timing attack due to the obvious relation, during the initialization, between the time used to transit to the all one state in the LFSR and δ.

Given t_i, s_{i+8} and c_i, the attacker easily distinguishes $l_i = 0$ from $l_i = 1$.

Case II of $\mathbf{c}_i$: The above procedure could be easily transferred to attack c_i in the second case, based on the observation that the output c_i of the multiplexer is a linear function, i.e., either $(s_{i+8} + t_i)$ or $(s_{i+8} + p_{f(i)})$. To launch the attack, the attacker chooses two complimentary plaintexts $(p_\delta, \ldots, p_n)$ and $(p'_\delta, \ldots, p'_n)$ for encryption, i.e., $p_i + p'_i = 1, \delta \leq i \leq n$. Let the corresponding ciphertexts be $(c_\delta, \ldots, c_{\hat{n}})$ and $(c'_\delta, \ldots, c'_{\hat{n}})$, we have

$$c_i + c'_i = \begin{cases} 0, & \text{if } l_{i+16} = 0, \\ 1, & \text{if } l_{i+16} = 1, \end{cases} \quad \text{for } i = \delta, \ldots, \hat{n},$$

which reveals the sequence $\{l_{\delta+16}, \ldots, l_{\hat{n}+16}\}$. Consequently, the sequence $\{s_{\delta+8}, \ldots, s_{\hat{n}+8}\}$ can be simply recovered from Eq. (6.4) once each l_{i+16} is known.

6.2.3 Step 2: Recover Internal States of NFSRs and Subkey sk_i

Given s_{i+8} and l_{i+16} $(i \geq \delta)$ obtained from the Step 1, the internal states of the two NFSRs are completely exposed to the attacker after $\max(\delta+9, \delta+17) = \delta+17$ clock cycles. Next, the attacker employs the following relation derived from Eq. (6.1) to recover the subkey bits sk_i for $i = \delta + 17, \ldots, n$,

$$sk_i = s_i + s_{i+1} \odot s_{i+2} + s_{i+3} + s_{i+6} + l_{i+16}.$$

As one may expect, the attacker is already capable of decrypting any ciphertext with the obtained subkeys $(sk_{\delta+17}, \ldots, sk_n)$. However, the attacker could do even better by fully recovering the secret key as described below.

6.2.4 Step 3: Fully Recover Secret Key

Without loss of generality, we fix δ to be a specific integer by guessing, e.g., assume $\delta = 88$ hereafter, and first recover the partial secret key bits $k_0, ..., k_{55}$. To this end, the attacker generates $(n - \delta - 16)$ equations (see below) from Eq. (6.2) as well as the internal state l_i

of the NFSRs, and derives the subkeys sk_i, when $i \geq \delta + 17$.

$$\begin{array}{rl}
k_{22} \cdot l_{119} + k_{23} = sk_{105} + 1 & k_{26} \cdot k_{30} + k_{29} = sk_{106} + 1 \\
k_{31} \cdot l_{121} + k_{34} = sk_{107} + 1 & k_{37} \cdot k_{40} + k_{38} = sk_{108} + 1 \\
k_{42} \cdot k_{45} + k_{44} = sk_{109} + 1 & k_{46} \cdot l_{124} + k_{48} = sk_{110} + 1 \\
k_{52} \cdot l_{125} + k_{53} = sk_{111} + 1 & k_{0} \cdot l_{126} + k_{3} = sk_{112} + 1 \\
k_{5} \cdot l_{127} + k_{8} = sk_{113} + 1 & k_{11} \cdot k_{14} + k_{13} = sk_{114} + 1 \\
\cdots & \cdots
\end{array}$$

Among these equations, approximately half of them are of the type

$$k_x \cdot l_y + k_z = sk_w + 1, \tag{6.5}$$

where x, y, z and w are integers. Since l_y is known to the attacker, he can select 56 such equations to form a sparse linear system with full rank and solve it to get $(k_0, ..., k_{55})$. Through extensive experiments, we found that when $n = 512 + \delta$, the attacker can obtain 56 linear independent equations out of 249 Eq. (6.5)-alike equations with probability 1, which implies that in a practical attack scenario the attacker should query the RFID tag with plaintexts of $n = 512 + \delta = 638$ bits (recall that $\delta \leq 126$).

To recover the rest key bits $k_{56}, \ldots, k_{60}$, we make use of the fact that δ is indeed the number of clock cycles required to transit the LFSR's state from $(k_{60}, \ldots, k_{56}, 1, 0)$ to $(1, 1, 1, 1, 1, 1, 1)$ or the reverse direction (here we assume $RND_R + RND_T = 0$ for simplicity). In our example, transiting the state $(1, 1, 1, 1, 1, 1, 1)$ reversely for $\delta = 88$ clock cycles gives us $(k_{56}, ..., k_{60}) = (1, 0, 1, 0, 0)$. Due to the uncertainty of δ, the attacker could have 32 possible keys such that the recovered $(k_0, ..., k_{55})$ is a δ-bit shifted version of the right key and $(k_{56}, ..., k_{60})$ is the state determined by δ as listed in Table 6.2. The attacker then utilizes the obtained one plaintext/ciphertext pair to test all 32 key candidates locally for retrieving the correct one.

6.2.5 Complexity Analysis of the Attack

Our attack is an ultra-efficient chosen-plaintext attack in terms of the computational overhead. Table 6.3 summarizes the computational complexity of the proposed attack.

Generally speaking, solving a system of linear equations with m variables requires $O(m^3)$ steps by the Gaussian elimination. However, as observed in our experiment, the linear system in question is quite special such that approximately 28 equations are of type

$$k_z = sk_w + 1,$$

Table 6.2: Deterministic Relation Between δ and $(k_{56}, \ldots, k_{60})$

$(k_{56}, \ldots, k_{60})$	δ	$(k_{56}, \ldots, k_{60})$	δ	$(k_{56}, \ldots, k_{60})$	δ
$(0, 0, 0, 0, 0)$	29	$(0, 0, 0, 0, 1)$	91	$(0, 0, 0, 1, 0)$	36
$(0, 0, 0, 1, 1)$	113	$(0, 0, 1, 0, 0)$	26	$(0, 0, 1, 0, 1)$	108
$(0, 0, 1, 1, 0)$	40	$(0, 0, 1, 1, 1)$	75	$(0, 1, 0, 0, 0)$	111
$(0, 1, 0, 0, 1)$	73	$(0, 1, 0, 1, 0)$	96	$(0, 1, 0, 1, 1)$	98
$(0, 1, 1, 0, 0)$	20	$(0, 1, 1, 0, 1)$	54	$(0, 1, 1, 1, 0)$	48
$(0, 1, 1, 1, 1)$	100	$(1, 0, 0, 0, 0)$	59	$(1, 0, 0, 0, 1)$	43
$(1, 0, 0, 1, 0)$	22	$(1, 0, 0, 1, 1)$	66	$(1, 0, 1, 0, 0)$	88
$(1, 0, 1, 0, 1)$	70	$(1, 0, 1, 1, 0)$	56	$(1, 0, 1, 1, 1)$	117
$(1, 1, 0, 0, 0)$	120	$(1, 1, 0, 0, 1)$	78	$(1, 1, 0, 1, 0)$	50
$(1, 1, 0, 1, 1)$	10	$(1, 1, 1, 0, 0)$	14	$(1, 1, 1, 0, 1)$	83
$(1, 1, 1, 1, 0)$	102	$(1, 1, 1, 1, 1)$	126		

Table 6.3: Computational Complexity of the Proposed Attack

Recovery Bits		Computation Cost
s_i	$i = \delta + 9, \ldots, n$	one encryption of n bits
l_i	$i = \delta + 17, \ldots, n$	one encryption of n bits
sk_i	$i = \delta + 17, \ldots, n$	negligible
k_i	$i = 0, \ldots, 60$	solve 32 sparse systems of linear equations ≈ 0.16 second on a Thinkpad T410

which immediately return the key bits. Moreover, the rest equations can be solved with around 0.005 second on a Thinkpad T410 laptop in our testing. In the worst case, the attacker has to solve 32 such systems of linear equations using around 0.16 second, which is negligible effort for the attacker.

6.3 Conclusion

In this chapter, we identified the security vulnerabilities of the A2U2 lightweight stream cipher and developed an ultra-efficient chosen-plaintext attack to fully recover the secret key of A2U2 through querying the encryption function twice on the victim tag and solving 32 sparse systems of linear equations with around 0.16 second. Our cryptanalysis implies that A2U2 has been completely broken and is not eligible to provide confidentiality and authenticity for RFID communications, which settled the concerns that are made in the conclusion of [73].

Additionally, the breaking of A2U2 manifests (again) that the lower bound of the hardware footprint (in terms of GE) of a cipher that provides enough security margin, given today's manufacturing process, is around $400 - 500$ GE. This is because the designs of lightweight cryptography must cope with the trade-offs between security, cost, and performance. It's generally easy to optimize any two of the three design goals, while it is extremely difficult to optimize all three design goals at once.

Chapter 7

Conclusions and Future Research

This chapter summarizes the contributions of the thesis and provides directions for future work.

7.1 Summary of Contributions

- **Survey of Existing Solutions**: We review various aspects of existed design and analysis for security and privacy issues. We introduce the recent advances in the lightweight symmetric ciphers, and we categorize and survey the design of lightweight authentication protocols. Moreover, solutions leveraging the physical layer resources are scrutinized as well. Finally, solutions along a nontechnical way are exhibited.

- **Physical Layer Enhancement of Passive RFID Communication**: We investigate how to solve the eavesdropping, modification and one particular type of relay attacks toward the tag-to-reader communication in passive RFID systems without requiring lightweight ciphers or secret credentials shared by legitimate parties using a physical layer approach. To this end, we propose a novel physical layer scheme, known as BUPLE. Besides, we also exploit coding in the physical layer to further improve the eavesdropping resistance. Three WCCs with ultra-lightweight constructions are exhibited. The security and usability of BUPLE in conjunction with WCCs are further confirmed by our proof-of-concept implementation and testing.

- **Active Eavesdropping Attacks against Passive RFID Systems**: We identify active eavesdropping as a brand-new and quite powerful family of attacks against passive RFID systems. In this attack, the adversary transmits an un-modulated carrier at a certain frequency, while a valid reader and a tag interacts at another frequency. By carefully examining the amplitude of the backscattered version of both blank carrier and reader's carrier, the eavesdropper is able to recognize tag's responses more reliably. Besides the formalization and the theoretic analysis of this attack, we demonstrated and empirically evaluated this new attack towards an EPC Gen2 system, through software-defined radio devices working at 860-960MHz and a WISP tag. Our experimental results show that the active eavesdropping achieves a significant improvement in the bit error rate of the intercepted communication. In addition, for a greedy proactive eavesdropper, we propose a simple countermeasure without introducing any computation/storage overhead to the current system. The experimentally grounded evaluation of this countermeasure is also presented.

- **Differential Sequence Attack on HummingBird-2**: We identify a novel cryptanalytic method, known as differential sequence attack, to attack the lightweight cipher HummingBird-2. We disclose that the differential sequences for the last round of this cipher can be computed by the full cipher and the search space for the keys in the last round can be reduced because the property of the differential sequences. Based on these observations, our full attack can be divided into two phases: (1) in the key recovery phase, we exploit the fact that in HB-2 the attacker can create an input differential for the last round and retrieve the corresponding output differentials. Thus by attacking the encryption (decryption resp.) of HB-2, our DSA algorithm recovers 36-bit (another 28-bit resp.) out of 128-bit key with 2^{68} (2^{60} resp.) time complexity and negligible memory complexity if particular differentials of the internal states and of the keys at one round can be maintained to the next round of encryption/decryption. Furthermore, the rest 64-bit of the key can be exhaustively searched and the overall time complexity is dominated by 2^{68}; (2) in the preparation phase, by investing 2^{81} effort in time, the attacker is able to create the conditions desired with at least 0.5 probability.

- **An Ultra-Efficient Key Recovery Attack on A2U2**: We report an ultra-efficient key recovery attack under the chosen-plaintext-attack model against the stream cipher A2U2, which is the most lightweight cryptographic primitive proposed so far for low-cost RFID tags. Our attack can fully recover the secret key of the A2U2 cipher by only querying the A2U2 encryption twice on the victim tag and solving 32 sparse systems of linear equations in the worst case. Our cryptanalysis implies that A2U2 has been completely broken.

7.2 Future Work

Recently, RFID technology is gaining an explosion of development in both industry and academia. It is broadly believed that, in the near future, the price of RFID will fall below a critical threshold and these tags will become commonplace – attached to almost every manufactured item. Therefore, research for security and privacy for low-cost RFID systems will become even more crucial than it is today. To this end, the following directions deserve further study.

7.2.1 Design Principle of Computation-efficient Cryptographic Primitives

Design a cryptanalytic strong and hardware efficient cryptographic primitive is presently more art than science. For the future design of computation-efficient ciphers, there is a clear need for more scientific formulation of the principles on which the security of such ciphers rests. For example, designers of the Hummingbird-2 suggests to use an 128-bit internal state in a block cipher to enable the use of smaller block size to benefit the quick response of a tag as well as the encryption/decryption of short messages in RFID protocols, while designers of KATAN suggests to use NFSRs iteratively to construct a secure block cipher. To explore and identify design principles for lightweight primitives, which are obscure at the current stage, is part of our future work.

7.2.2 Physical Layer Encryption of RFID Communication

Furthermore, the combination of the cryptographic approach and the physical-layer approach seems promising in providing low-cost security for RFID tags. To be specific, we are currently working on the *physical layer encryption of RFID communication* to further solve the eavesdropping problem – messages from a tag are first coded to longer sequence which are then encrypted using a lightweight stream cipher, e.g., WG-7. Although this approach requires longer encryption sequences, the natural randomness of the noisy tag-to-reader communication channel can be used effectively for confidentiality purpose. Our design is in light of the truth that an eavesdropper can either stay close to a victim tag for only a short time period or stay away from it in a long run. The former gives the attacker limited number of pairs of plaintext/ciphertext containing less noise, while the latter provides the attacker more pairs of noisy plaintext/ciphertext. The analysis of the security of this scheme represents a marked departure from conventional cryptanalysis of symmetric ciphers, which roots in the assumption that error-free copies of plaintext/ciphertext pairs are always available to the attacker. We plan to introduce and study the concept of *noisy cryptanalysis*, which may bring in more insights on the theoretic and practical security for low-cost RFID systems.

7.2.3 Confidentiality-preserving Bidirectional Communication though Collaboration of Multiple Readers

We are also developing a novel scheme to enable confidentiality-preserving bidirectional communication without relying on lightweight cryptographic primitives. In our scheme, multiple readers are employed to interact with a single tag as opposed to the convention that one reader interacts with multiple tags.

This research is initiated from our work in Chapter 3, where we notice that if a reader broadcasts WCC-encoded signals at a certain power level, only tags located in a certain region (call it *confidentiality-preserving region* or *C-region* hereafter), e.g., usually a circle if the reader is using a dipole antenna, is able to listen and decode the message transmitted. Moreover, the size of the C-region can be controlled by adjusting the power level of the reader's transmitter. In a multiple reader scenario, each reader creates a C-region on its own. Thus, a super C-region, i.e., the joint region of all individual C-regions, could be of any geometric shape, e.g., a point, if the readers adjust their respect C-region in a collaborative way. As long as a tag is placed within this region, a confidentiality-preserving reader-tag communication is enabled. In addition to the experimental verification of this idea we are currently working on, we will enhance this scheme to fight against distributed attackers with strategies such as space/time/frequency hopping for the readers.

7.2.4 Security and Privacy in Internet of Things (IoT)

The IoT is a vision of connectivity for anything, anytime and anywhere, which may have impact on our daily life dramatically as the Internet has done in the past 20 years. The RFID tag with sensing and positioning capabilities has been recognized as a promising enabler towards IoT. A typical configuration of IoT is that once the reader creates an electromagnetic field, all passive tags attached to physical objects in the operating range start to work and communicate indirectly with each other. One noteworthy phenomenon is the indirectness of the communication such that: without transceiver parts, tag A talks to tag B by first backscattering the message to the reader. The reader then decodes and broadcasts it to tag B. In such an interconnected world of miniaturized systems, security and privacy is even more challenging, i.e., possible threats could be: (1) communication among tags in IoT can be traffic-analyzed; (2) tags with positioning capability are more susceptible to be tracked; (3) tags with sensing capability, if unauthorized accessed, leak the information regarding the physical environment where they are placed, etc.. Our long term goal is to study the security and privacy problems in IoT, e.g., how to design multi-party lightweight authentication protocols and key exchange/establishment protocols.

APPENDICES

Appendix A

Matlab Codes for Computation Security Properties of Proposed $(16, 8)$-WCC

The following snippet of code calculates the information rate, the equivocation rate of the proposed $(16, 8)$-WCC.

```
n = 16; m = 8;  p = 0.2; pc = 1 - p;
prob_sum = zeros(1,2^m); counter = zeros(1,2^m);

for i=0:1:2^n-1
    str = dec2bin(i);str_len = size(str);
    for j=1:n-str_len(2)
        str = strcat(’0’,str);
    end
    A = zeros(1,n);
    for j=1:1:n
        if str(j)==’1’
            A(j) = 1;
        else
            A(j) = 0;
        end
    end
    Ci = A;
    f = zeros(m,1);
    f(1)=  mod(Ci(9) +(Ci(1) + Ci(2) + Ci(4) + Ci(7) + Ci(8) + (Ci(1)+Ci(5))*
           (Ci(2)+Ci(3)+Ci(4)+Ci(6)) + (Ci(2)+Ci(3))*(Ci(4)+Ci(6))),2);
    f(2) = mod(Ci(10)+(Ci(2) + Ci(3) + Ci(5) + Ci(1) + Ci(8) + (Ci(2)+Ci(6))*
           (Ci(3)+Ci(4)+Ci(5)+Ci(7)) + (Ci(3)+Ci(4))*(Ci(5)+Ci(7))),2);
    f(3) = mod(Ci(11)+(Ci(3) + Ci(4) + Ci(6) + Ci(2) + Ci(8) + (Ci(3)+Ci(7))*
           (Ci(4)+Ci(5)+Ci(6)+Ci(1)) + (Ci(4)+Ci(5))*(Ci(6)+Ci(1))),2);
    f(4) = mod(Ci(12)+(Ci(4) + Ci(5) + Ci(7) + Ci(3) + Ci(8) + (Ci(4)+Ci(1))*
           (Ci(5)+Ci(6)+Ci(7)+Ci(2)) + (Ci(5)+Ci(6))*(Ci(7)+Ci(2))),2);
    f(5) = mod(Ci(13)+(Ci(5) + Ci(6) + Ci(1) + Ci(4) + Ci(8) + (Ci(5)+Ci(2))*
           (Ci(6)+Ci(7)+Ci(1)+Ci(3)) + (Ci(6)+Ci(7))*(Ci(1)+Ci(3))),2);
    f(6) = mod(Ci(14)+(Ci(6) + Ci(7) + Ci(2) + Ci(5) + Ci(8) + (Ci(6)+Ci(3))*
           (Ci(7)+Ci(1)+Ci(2)+Ci(4)) + (Ci(7)+Ci(1))*(Ci(2)+Ci(4))),2);
    f(7) = mod(Ci(15)+(Ci(7) + Ci(1) + Ci(3) + Ci(6) + Ci(8) + (Ci(7)+Ci(4))*
           (Ci(1)+Ci(2)+Ci(3)+Ci(5)) + (Ci(1)+Ci(2))*(Ci(3)+Ci(5))),2);
    f(8) = mod(Ci(1) + Ci(2) + Ci(3) + Ci(4) + Ci(5) + Ci(6) + Ci(7) + Ci(8)+
           Ci(9) + Ci(10) + Ci(11) + Ci(12) + Ci(13) + Ci(14) + Ci(15) + Ci(16),2);

    index = 1;
    for j=1:m
        if f(j)==1
        index = index + 2^(j - 1);
        end
    end

    weight_ctr = 0;
        for j=1:1:n
            if A(j) == 1
                weight_ctr = weight_ctr + 1;
            end
        end
    prob_sum(index) = prob_sum(index) + p^weight_ctr*pc^(n-weight_ctr);
    counter(index) = counter(index) + 1;
end
entropy = 0;
for i=1:1:2^m
    if prob_sum(i)>0
    entropy = entropy - prob_sum(i)*log(prob_sum(i))/log(2);
    end
end

disp([’A [’ num2str(n) ’, ’ num2str(m) ’] code’]);
disp([’When error prob.: ’ num2str(p)])
disp([’equivocation rate: ’ num2str(entropy/m, 15)])
disp([’R*d: ’ num2str(entropy/n,15)])
disp([’secrecy capacity: ’ num2str(-(1-p)*log(1-p)/log(2)-p*log(p)/log(2),15)])

function prob_sum = prob_cal(n,m,H,S,p)
    pc = 1 - p;
    prob_sum = 0;

    for i=0:1:2^n-1
        str = dec2bin(i);str_len = size(str);
        for j=1:n-str_len(2)
            str = strcat(’0’,str);
        end
        A = zeros(1,n);
        for j=1:1:n
            if str(j)==’1’
                A(j) = 1;
            else
                A(j) = 0;
            end
        end

        if mod(H*A’,2)== S’
        weight_ctr = 0;
            for j=1:1:n
                if A(j) == 1
                    weight_ctr = weight_ctr + 1;
                end
            end
            prob_sum = prob_sum + p^weight_ctr*pc^(n-weight_ctr);
        end
    end
```

References

[1] M.R.S. Abyaneh, On the security of nonlinear HB (NLHB) protocol against passive attack, *IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, EUC'10*, pp. 523–528, 2010.

[2] M. Agren, Some instant- and practical-time related-key attacks on KTANTAN-32/48/64, to appear *In Proceedings of Selected Areas in Cryptography, SAC'11*, pp. 1–17, 2011.

[3] G. Avoine, M.A. Bingöl, S. Kardaş, C. Lauradoux, and B. Martin, A framework for analyzing RFID distance bounding protocols, *Journal of Computer Security*, vol. 19, no. 2, pp. 289–317, 2011.

[4] G.T. Amariucai, C. Bergman and Y. Guan, An automatic, time-based, secure pairing protocol for passive RFID, to appear *In Proceedings of RFIDSec'11*. pp. 1–20, 2011.

[5] T. Akishita and H. Hiwatari, Very compact hardware implementations of the blockcipher CLEFIA, *Technical Report, available at http://www.sony.net/Products /cryptography/clefia/download/data/clefia-hw-compact-20110615.pdf*, pp. 1–15, 2010.

[6] J.P. Aumasson, L. Henzen, W. Meier and M. Naya-Plasencia, Quark: a lightweight hash, *Cryptographic Hardware and Embedded Systems, CHES'10*, LNCS 6225, pp. 1–15, 2010.

[7] M. Abdelraheem, G. Leander and E. Zenner, Differential cryptanalysis of round-reduced PRINTCipher: computing roots of permutations, *Fast Software Encryption, FSE'11*, LNCS 6733, pp. 1–17, 2011.

[8] G. Avoine, C. Floerkemeier and B. Martin, RFID distance bounding multistate enhancement, *Progress in Cryptology, Indocrypt'09*, LNCS 5922, pp. 290–307, 2009.

[9] D. Alistarh, S. Gilbert, R. Guerraoui, Z. Milosevic and C. Newport, Securing every bit: authenticated broadcast in radio networks, *In Proceedings of the 22nd ACM symposium on Parallelism in algorithms and architectures*, pp. 50–59, 2010.

[10] B. Azimi-Sadjadi, A. Kiayias, A. Mercado and B. Yener, Robust key generation from signal envelopes in wireless networks, *In Proceedings of the 14th ACM conference on Computer and Communications Security, CCS'07*, pp. 401–410, 2007.

[11] G. Avoine and P. Oechslin, A scalable and provably secure hash based RFID protocol. *International Workshop on Pervasive Computing and Communication Security, PerCom'05*, pp. 110–114, 2005.

[12] H. Ahmadi and R. Safavi-Naini, Secret keys from channel noise, *Advances in Cryptology, EUROCRYPT'11*, LNCS 6632, pp. 266–283, 2011.

[13] G. Avoine and A. Tchamkerten, An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement, *Information Security Conference, ISC'09*, pp. 250–261, 2009.

[14] S. Bengio, G. Brassard, Y.G. Desmedt, C. Goutier, and J.J. Quisquater, Secure implementation of identification system, *Journal of Cryptology*, vol. 4, no. 3, pp. 175–183, 1991.

[15] C. Berbain, O. Billet, J. Etrog and H. Gilbert, An efficient forward private RFID protocol, *In Proceedings of the 16th ACM conference on Computer and Communications Security, CCS'09*, pp. 43–53, 2009.

[16] E. Biham, A. Biryukov and A. Shamir, Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials, *Journal Of Cryptology*, vol. 18, no. 4, pp. 291–311, 1999.

[17] S. Brands and D. Chaum, Distance-bounding protocols, *Advances in Cryptology, EUROCRYPT'93*, LNCS 765, pp. 344–359, 1994.

[18] J. Bringer and H. Chabanne, On the wiretap channel induced by noisy tags, *Security and Privacy in Ad-Hoc and Sensor Networks*, LNCS 4357, pp. 113–120, 2006.

[19] J. Bringer, H. Chabanne, G. Cohen and B. Kindarji, RFID key establishment against active adversaries, *First IEEE International Workshop on Information Forensics and Security, WIFS'09*, pp. 186–190, 2009.

[20] J. Bringer, H. Chabanne, E. Dottax, and S.D. Securite, HB^{++}: a lightweight authentication protocol secure against some attacks, *The 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pp. 28–33, 2006.

[21] G. Bard, N. Courtois, J. Nakahara, P. Sepehrdad and B. Zhang, Algebraic, AIDA/Cube and side channel analysis of KATAN family of block ciphers, *Progress in Cryptology, Indocrypt'10*, LNCS 6498, pp. 176–196, 2010.

[22] T. Beth and Y. Desmedt, Identification tokens – or: Solving the chess grandmaster problem, *Advances in Cryptology, CRYPT'91*, pp. 169–176, 1991.

[23] M. Burmester, B. De Medeiros and R. Motta, Robust, anonymous RFID authentication with constant key-lookup, *In Proceedings of the 2008 ACM symposium on Information, Computer and Communications Security, CCS'08*, pp. 283–291, 2008.

[24] S. Badel, N. Dağtekin, J. Nakahara, K. Ouafi, N. Reffé, P. Sepehrdad, P. Sušil and S. Vaudenay, ARMADILLO: a multi-purpose cryptographic primitive dedicated to hardware, *Cryptographic Hardware and Embedded Systems, CHES'10*, LNCS 6225, pp. 398–412, 2011.

[25] O. Billet, J. Etrog and H. Gilbert, Lightweight privacy preserving authentication for RFID using a stream cipher, *Fast Software Encryption, FSE'10*, LNCS 6147, pp. 55–74, 2010.

[26] O. Billet and K. Elkhiyaoui, Two attacks against the F_f RFID protocol, *Progress in Cryptology, Indocrypt'09*, LNCS 5922, pp. 308–320, 2009.

[27] C. Blondeau and B. Gérard, Multiple differential cryptanalysis: theory and practice, *Fast Software Encryption, FSE'11*, LNCS 6733, pp. 35–54, 2011.

[28] E.O. Blass, A. Kurmus, R. Molva, G. Noubir and A. Shikfa, The F_f-family of protocols for RFID-privacy and authentication, *Dependable and Secure Computing, IEEE Transactions on*, vol. 8, no. 3, pp. 466–480, 2011.

[29] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin and C. Vikkelsoe, PRESENT: An ultra-lightweight block cipher, *Cryptographic Hardware and Embedded Systems, CHES'07*, LNCS 4727, pp. 450-466, 2007.

[30] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı and I. Verbauwhede, SPONGENT: A lightweight hash function, *Cryptographic Hardware and Embedded Systems, CHES'11*, LNCS 6917, pp. 312–325, 2011.

[31] A. Biryukov, I. Kizhvatov and B. Zhang, Cryptanalysis of the Atmel cipher in SecureMemory, CryptoMemory and CryptoRF, *Applied Cryptography and Network Security, ACNS'11*, LNCS 6715, pp. 91–109, 2011.

[32] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw and Y. Seurin, Hash functions and RFID tags: mind the gap. *Cryptographic Hardware and Embedded Systems, CHES'08*, LNCS 5154, pp. 283–299, 2008.

[33] M. Burmester and J. Munilla, Lightweight RFID authentication with forward and backward security, *Information and System Security, ACM Transactions on*, vol. 14, no, 1, pp. 1–26, 2011.

[34] E. Berlekamp, R. McEliece and H. Van Tilborg, On the inherent intractability of certain coding problems, *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 384–386, 1978.

[35] C. Blondeau, M. Naya-Plasencia, M. Videau and E. Zenner, Cryptanalysis of ARMADILLO2, to appear *In Proceedings of AsiaCrypt'11*, pp. 1–17 , 2011.

[36] T. Beth and F. Piper, The stop-and-go generator, *Advances in Cryptology, EUROCRYPT'84*, LNCS 209, pp. 88–92, 1985.

[37] A. Bogdanov and C. Rechberger, A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN, *Selected Areas in Cryptography, SAC'10*, LNCS 6544, pp. 229–240, 2010.

[38] G. Brassard and L. Salvail, Secret-key reconciliation by public discussion, *Advances in Cryptology, EUROCRYPT'93*, LNCS 765, pp. 410–423, 1994.

[39] M. Burmester, T. Van Le and B. de Medeiros, Provably secure ubiquitous systems: universally composable RFID authentication protocols, *Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp. 1-9, 2006.

[40] C. Carlet, *Vectorial Boolean functions for cryptography*, Cambridge University Press, 2006.

[41] J. Cho, Linear cryptanalysis of reduced-round PRESENT, *The Cryptographers' Track at the RSA Conference, CT-RSA'10*, LNCS 5985, pp. 302–317, 2010.

[42] Circular Polarity Pane Antenna, *http://www.arcadianinc.com/datasheets/4123.pdf*, 2011.

[43] J.H. Conway, On numbers and games. *Number 6 in London Mathematical Society Monographs*, Academic Press, 1976.

[44] C. Castelluccia and G. Avoine, Noisy tags: a pretty good key exchange protocol for RFID tags, *International Conference on Smart Card Research and Advanced Applications*, LNCS 3928, pp. 289–299, 2006.

[45] I. Collaboration, M. Antonello, P. Aprili and others, A search for the analogue to Cherenkov radiation by high energy neutrinos at superluminal speeds in ICARUS, *Arxiv preprint arXiv:1110.3763*, pp. 1–8, 2011.

[46] N. Courtois, G. Bard and D. Wagner, Algebraic and slide attacks on KeeLoq, *Fast Software Encryption, FSE'08*, LNCS 5086, pp. 97–115, 2008.

[47] M. Cagalj, S. Capkun, R. Rengaswamy, I. Tsigkogiannis, M. Srivastava and J.P. Hubaux, Integrity (I) codes: message integrity protection and authentication over insecure channels, *29th IEEE Symposium on Security and Privacy, S&P'08*, pp. 279-294, 2006.

[48] C. Chen and Y. Deng, Conformation of EPC C1G2 standards RFID system with mutual authentication and privacy protection, *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1284–1291, 2009.

[49] S. Capkun, K. El Defrawy and G. Tsudik, Group distance bounding protocols, *In Proceedings of the 4th International Conference on Trust and Trustworthy Computing*, pp. 302–312, 2011.

[50] H. Chabanne and G. Fumaroli, Noisy cryptographic protocols for low-cost RFID tags, *Information Theory, IEEE Transactions on*, vol. 52, no. 8, pp. 3562–3566, 2006.

[51] J.H. Cheon J. Hong and G. Tsudik, Reducing RFID reader load with the meet-in-the-middle strategy, *Cryptology ePrint Archive, Report 2009/092*, pp. 1–9, 2009.

[52] J. Clulow, G. Hancke, M. Kuhn and T. Moore, So near and yet so far: distance-bounding attacks in wireless networks, *Security and Privacy in Ad-hoc and Sensor Networks*, LNCS 4357, pp. 83–97, 2006.

[53] I. Csiszar and J. Korner, Broadcast channels with confidential messages, *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 339–348, 2002.

[54] D. Coppersmith, H. Krawczyk and Y. Mansour, The shrinking generator, *Advances in Cryptology, CRYPTO'93*, LNCS 773, pp. 22–39, 1994.

[55] I. Csiszár and P. Narayan, Secrecy capacities for multiterminal channel models, *Information Theory, IEEE Transactions on*, vol. 54, no. 6, pp. 2437–2452, 2008.

[56] M. Conti, R.D. Pietro, L.V. Mancini and A. Spognardi, RIPP-FS: an RFID identification, privacy preserving protocol with forward secrecy, *International Conference on Pervasive Computing and Communications, PerCom'07*, pp. 229–234, 2007.

[57] B. Collard and F.X. Standaert, A statistical saturation attack against the block cipher PRESENT. *The Cryptographers' Track at the RSA Conference, CT-RSA'09*, LNCS 5473, pp. 195–210, 2009.

[58] B. Collard and F.X. Standaert, Multi-trail statistical saturation attacks, *Applied Cryptography and Network Security, ACNS'10*, LNCS 6123, pp. 123–138, 2010.

[59] Q. Cai, Y. Zhan and Y. Wang, A minimalist mutual authentication protocol for RFID system & BAN logic analysis, *In Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM'08*, vol. 2, pp. 449–453, 2008.

[60] C. De Canniere, Trivium: A stream cipher construction inspired by block cipher design principles, *Information Security*, LNCS 4176, pp. 171–186, 2006.

[61] Y. Desmedt, Society and group oriented cryptography: A new concept, *Advances in Cryptology, CRYPTO'87*, pp. 120–127, 2006.

[62] D.M. Dobkin, *The RF in RFID: passive UHF RFID in practice*, Newnes, 2007.

[63] D.M. Dobkin, UHF Reader Eavesdropping: intercepting a tag reply, *http://www.enigmatic-consulting.com/Communications_articles/RFID/Tag_intercept.html*, 2008.

[64] C. De Canniere, O. Dunkelman and M. Knezevic, KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers, *Cryptographic Hardware and Embedded Systems, CHES'09*, LNCS 5747, pp. 272–288, 2009.

[65] I. Dinur, O. Dunkelman and A. Shamir, Improved attacks on full GOST, *Cryptology ePrint Archive, Report 2011/558*, pp. 1–25, 2011.

[66] A. Dvoretzky and P. Erdos, Some problems on random walk in space, *In Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability*, vol. 353, pp. 353–367, 1951.

[67] B. Danev, T.S. Heydt-Benjamin and S. Capkun, Physical-layer identification of R-FID devices, *In Proceedings of 18th conference on USENIX security Symposium, USENIX'09*, pp. 199–214, 2009.

[68] G. DeJean and D. Kirovski, RF-DNA: radio-frequency certificates of authenticity, *Cryptographic Hardware and Embedded Systems, CHES'07*, LNCS 4727, pp. 346–363, 2007.

[69] D.N. Duc and K. Kim, Securing HB$^+$ against GRS man-in-the-middle attack, *Symposium on Cryptography and Information Security Information and Communication Engineers*, pp. 23–26, 2007.

[70] B. Danev, H. Luecken, S. Capkun and K. El Defrawy, Attacks on physical-layer identification, *In Proceedings of the third ACM conference on Wireless network Security, WiSec'10*, pp. 89–98, 2010.

[71] R. Deng, Y. Li, M. Yung and Y. Zhao, A new framework for RFID privacy, *In European Symposium on Research in Computer Security, ESORICS'10*, LNCS 6345, pp. 1–18, 2011.

[72] S. Drimer and S.J. Murdoch, Keep your enemies close: distance bounding against smartcard relay attacks, *In Proceedings of 16th USENIX Security Symposium, USENIX'07*, pp. 1–16, 2007.

[73] M. David, D.C. Ranasinghe and T. Larsen, A2U2: A stream cipher for printed electronics RFID tags, *IEEE International Conference on RFID, RFID'11* , pp. 176–183, 2011.

[74] I. Dinur and A. Shamir, Cube attacks on tweakable black box polynomials, *Advances in Cryptology, EUROCRYPT'09*, LNCS 5479, pp. 278–299, 2009.

[75] I. Dinur and A. Shamir, Breaking Grain-128 with dynamic cube attacks, *Fast Software Encryption, FSE'11*, LNCS 6733, pp. 167–187, 2011.

[76] Ettus Research LLC, USRP-1 and RFX900 daughter boards, *http://www.ettus.com/downloads/ettus_ds_usrp_v7.pdf*, 2011.

[77] Ettus Research LLC, USRP-N210 and RFX900 daughter boards, *http://www.ettus.com/downloads/ettus_ds_usrp_n200series_v3.pdf*, 2011.

[78] EPC Global, Class 1 Generation 2 UHF air interface protocol standard v1.2, *http://www.epcglobalinc.org*, 2008.

[79] EMV Standard, *http://www.emvco.com/specifications.aspx*, 2011.

[80] Elliptic Technologies, CLP-38: KASUMI flow through core, *http://www.elliptictech.com/products-clp-38.php*, 2011.

[81] Elliptic Technologies, CLP-41: SNOW 3G flow through core, *http://www.elliptictech.com/products-clp-41.php*, 2011.

[82] Elliptic Technologies, ZUC key stream generator, *http://www.elliptictech.com/pdf/CLP-410_ZUC_Key_Stream_Generator.pdf*, 2011.

[83] D. Engels, X. Fan, G. Gong, H. Hu and E. Smith, Hummingbird: ultra-lightweight cryptography for resource-constrained devices, *Financial Cryptography and Data Security, FC'10*, LNCS 6054, pp. 3–18, 2010.

[84] D. Engels, M.J.O. Saarinen and E. Smith, The Hummingbird-2 lightweight authenticated encryption algorithm, to appear *In Proceedings of Workshop on RFID Security, RFIDSec'11*, pp. 1–14, 2011.

[85] M. Feldhofer, S. Dominikus and J. Wolkerstorfer, Strong authentication for RFID systems using the AES algorithm, *Cryptographic Hardware and Embedded Systems, CHES'04*, LNCS 3156, pp. 357–370, 2004.

[86] X. Fan and G. Gong, On the security of Hummingbird-2 against side-channel cube attacks, *Western European Workshop on Research in Cryptology, WEWoRC'11*, pp. 100–104, 2011.

[87] L. Francis, G. Hancke, K. Mayes and K. Markantonakis, Practical NFC peer-to-peer relay attack using mobile phones, *Workshop on RFID Security, RFIDSec'10*, LNCS 6370, pp. 35–49, 2010.

[88] L. Francis, G. Hancke, K. Mayes and K. Markantonakis, Practical relay attack on contactless transactions by using NFC mobile phones, *Cryptology ePrint Archive, Report 2011/618*, pp. 1–15, 2011.

[89] M. Feldhofer, J. Wolkerstorfer and V. Rijmen, AES implementation on a grain of sand, *Information Security, IEE Proceedings*, vol. 152, no. 1, pp. 13–20, 2005.

[90] S. Garfinkel, An RFID bill of rights. *Technology Review*, vol. 10, pp. 35, 2002.

[91] G. Gong, Sequences, DFT and resistance against fast algebraic attacks, *Sequences and Their Applications, SETA'08*, LNCS 5203, pp. 197–218, 2008.

[92] GNU Radio, *http://www.gnu.org/software/gnuradio*, 2011.

[93] T. Good and M. Benaissa, Hardware results for selected stream cipher candidates, *In Proceedings of SASC 2007*, pp. 191–204, 2007.

[94] B. Gassend, D. Clarke, M. Van Dijk and S. Devadas, Silicon physical random functions, *In Proceedings of the 9th ACM conference on Computer and Communications Security, CCS'02*, pp. 148–160, 2002.

[95] S.W. Golomb and G. Gong, *Signal design with good correlation: for wireless communications, cryptography and radar applications*, Cambridge University Press, 2005.

[96] P.K. Gopala, L. Lai and H. El Gamal, On the secrecy capacity of fading channels, *Information Theory, IEEE Transactions on*, vol. 54, no. 10, pp. 4687–4698, 2008.

[97] Z. Gong, S. Nikova and Y.W. Law, Klein: a new family of lightweight block ciphers, to appear *In Proceedings of Workshop on RFID Security, RFIDSec'11*, pp. 1–18, 2011.

[98] J. Guo, T. Peyrin and A. Poschmann, The PHOTON family of lightweight hash functions, *Advances in Cryptology, CRYPTO'11*, LNCS 6841, pp. 222–239, 2011.

[99] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, The LED block cipher, *Cryptographic Hardware and Embedded Systems, CHES'11*, LNCS 6917, pp. 326–341, 2011.

[100] F.D. Garcia, P. Rossum, R. Verdult and R.W. Schreur, Wirelessly pickpocketing a Mifare classic card, *In Proceedings of the 30th IEEE Symposium on Security and Privacy, S&P'09*, pp. 3–15, 2009.

[101] H. Gilbert, M.J. Robshaw and Y. Seurin, Good variants of HB$^+$ are hard to find, *Financial Cryptography and Data Security, FC'08*, LNCS 5143, pp. 156–170, 2008.

[102] H. Gilbert, M.J.B. Robshaw and Y. Seurin, HB$^\#$: Increasing the security and efficiency of HB$^+$, *Advances in Cryptology, EUROCRYPT'08*, LNCS 4965, pp. 361–378, 2008.

[103] H. Gilbert, M. Robshaw and H. Sibert, An active attack against HB$^+$ – a provably secure lightweight authentication protocol, *IEE Electronic Letters*, vol. 41, no. 21, pp. 1169–1170, 2005.

[104] J. Guajardo, P. Tuyls, N. Bird, C. Conrado, S. Maubach, G.J. Schrijen, B. Skoric, A.M.H. Tombeur and P. Thueringer, RFID security: cryptography and physics perspectives, *RFID Security*, pp. 103–130, 2009.

[105] G. Gong and A. Youssef, Cryptographic properties of the Welch-Gong transformation sequence generators, *Information Theory, IEEE Transaction on*, vol. 48, no. 11, pp. 2837–2846, 2002.

[106] F.D. Garcia, P. van Rossum, R. Verdult and R.W. Schreur, Dismantling SecureMemory, CryptoMemory and CryptoRF. *In Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS'10*, pp. 250–259, 2010.

[107] G. Hancke, Noisy carrier modulation for HF RFID, *First International EURASIP Workshop on RFID Technology*, pp. 63–66, 2007.

[108] G. Hancke, Modulating a noisy carrier signal for eavesdropping-resistant HF RFID, *e & i Elektrotechnik und Informationstechnik*, vol. 124, no. 11, pp. 404–408, 2007.

[109] G. Hancke, Eavesdropping attacks on high-frequency RFID tokens, *In Proceedings of the 4th Workshop on RFID Security, RFIDSec'08*, pp. 1–14, 2008.

[110] G. Hancke, Design of a secure distance-bounding channel for RFID, *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 1–11, 2011.

[111] S. Haykin, Cognitive radio: brain-empowered wireless communications, *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 2, pp. 201–220, 2005.

[112] N.J. Hopper and M. Blum, A secure human-computer authentication scheme, *Technical Report of Carnegie Mellon University, CMU-CS-00-139, available at http://reports-archive.adm.cs.cmu.edu/anon/2000/abstracts/00-139.html*, pp. 1–8, 2000.

[113] D.E. Holcomb, W.P. Burleson and K. Fu, Initial SRAM state as a fingerprint and source of true random numbers for RFID tags, *In Proceedings of the Conference on RFID Security, RFIDSec'07*, pp. 11–13, 2007.

[114] D.E. Holcomb, W.P. Burleson and K. Fu, Power-up SRAM state as an identifying fingerprint and source of true random numbers, *Computers, IEEE Transactions on*, vol. 58, no. 9, pp. 1198–1210, 2009.

[115] M. Hell, T. Johansson and W. Meier, Grain: a stream cipher for constrained environments, *International Journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 86–93, 2007.

[116] G. Hancke and M.G. Kuhn, Attacks on time-of-flight distance bounding channels, *In Proceedings of the first ACM conference on Wireless network Security, WiSec'08*, pp. 194–202, 2008.

[117] G. Hancke and M.G. Kuhn, An RFID distance bounding protocol, *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm'05*, pp. 67–73, 2005.

[118] G. Hammouri, E. Öztürk, B. Birand and B. Sunar, Unclonable lightweight authentication scheme, *Information and Communications Security*, LNCS 5308, pp. 33–48, 2008.

[119] D.S. Ha and P.R. Schaumont, Replacing cryptography with ultra wideband (UWB) modulation in secure RFID, *IEEE International Conference on RFID, RFID'07*, pp. 23–29, 2007.

[120] G. Hammouri and B. Sunar, PUF-HB: A tamper-resilient HB based authentication protocol, *Applied Cryptography and Network Security, ACNS'08*, LNCS 5037, pp. 346–365, 2008.

[121] D. Hong, J. Sung, S. Hong and others, HIGHT: a new block cipher suitable for low-resource device, *Cryptographic Hardware and Embedded Systems, CHES'06*, LNCS 4249, pp. 46–59, 2006.

[122] T. Isobe, A single-key attack on the full GOST block cipher, *Fast Software Encryption, FSE'11*, LNCS 6733, pp. 290–305, 2011.

[123] A. Juels, Yoking-proofs for RFID tags, *In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, PerCom'04*, pp. 138–143, 2004.

[124] A. Juels, Minimalist cryptography for low-cost RFID tags, *Security in Communication Networks*, pp. 149–164, 2005.

[125] A. Juels, RFID security and privacy: a research survey, *Selected Areas in Communications, IEEE Journal on*, vol. 24, No. 2, pp. 381–394, 2006.

[126] A. Juels, R.L. Rivest and M. Szydlo, The blocker tag: selective blocking of RFID tags for consumer privacy, *In Proceedings of the 10th ACM Conference on Computer and Communication Security, CCS'03*, pp. 103–111, 2003.

[127] A. Juels and S.A. Weis, Authenticating pervasive devices with human protocols, *Advances in Cryptology, CRYPTO'05*, LNCS 3621, pp. 293–308, 2005.

[128] A. Juels and S.A. Weis, Defining strong privacy for RFID, *In Proceedings of Fifth IEEE Annual International Conference on Pervasive Computing and Communications Workshops, PerCom'07*, pp. 342–347, 2007.

[129] J.P. Kaps, Chai-tea, cryptographic hardware implementations of xTEA, *Progress in Cryptology, Indocrypt'08*, LNCS 5365, pp. 363–375, 2008.

[130] L. Knudsen, Truncated and higher order differentials, *Fast Software Encryption. FSE'95*, LNCS 1008, pp. 196–211, 1995.

[131] S. Kardas, M. Akgün, M.S. Kiraz and H. Demirci, Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems, *Workshop on Lightweight Security & Privacy: Devices, Protocols and Applications, LightSec'11*, pp. 20–25, 2011.

[132] C. Kim and G. Avoine, RFID distance bounding protocol with mixed challenges to prevent relay attacks, *Conference on Cryptology and Network Security, CANS'09*, LNCS 5888, pp. 119–133, 2009.

[133] M. Ko and D.L. Goeckel, Wireless physical-layer security performance of UWB systems, *Military Communications Conference, MILCOM'10*, pp. 2143–2148, 2010.

[134] S. Kardas, M.S. Kiraz, M.A. Bingöl and H. Demirci, A novel RFID distance bounding protocol based on physically unclonable functions, *Workshop on RFID Security, RFIDSec'09*, pp. 1–17, 2009.

[135] T. Karygiannis, B. Eydt, G. Barber, L. Bunn and T. Phillips, Guidelines for securing radio frequency identification (RFID) systems, *NIST Special Publication*, vol. 80, pp. 1–154, 2007.

[136] K. Koscher, A. Juels, V. Brajkovic and T. Kohno, EPC RFID tag security weaknesses and defenses: passport cards, enhanced drivers licenses, and beyond, *In Proceedings of the 16th ACM conference on Computer and Communications Security, CCS'09*, pp. 33–42, 2009.

[137] L. Knudsen, G. Leander, A. Poschmann and M. Robshaw, PRINTCipher: a block cipher for IC-printing, *Cryptographic Hardware and Embedded Systems, CHES'10*, LNCS 6225, pp. 16–32, 2010.

[138] S. Knellwolf, W. Meier and M. Naya-Plasencia, Conditional differential cryptanalysis of NLFSR-based cryptosystems, *Advances in Cryptology, AsiaCrypt'10*, LNCS 6477, pp. 130–145, 2010.

[139] S. Knellwolf, W. Meier and M. Naya-Plasencia, Conditional differential cryptanalysis of Trivium and KATAN, to appear *In Proceedings of Selected Areas in Cryptography, SAC'11*, pp. 1–14, 2011.

[140] G. Karjoth and P.A. Moskowitz, Disabling RFID tags with visible confirmation: clipped tags are silenced, *In Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES'05*, pp. 27–30, 2005.

[141] G. Kapoor and S. Piramuthu, Single RFID tag ownership transfer protocols, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, no. 99, pp. 1–10, 2011.

[142] J. Katz, J.S. Shin and A. Smith, Parallel and concurrent security of the HB and HB+ protocols, *Journal of cryptology*, vol. 23, no. 3, pp. 402–421, 2010.

[143] M. Kuhn, H. Luecken and N.O. Tippenhauer, UWB impulse radio based distance bounding, *The 7th Workshop on Positioning Navigation and Communication, WPNC'10*, pp. 28–37, 2010.

[144] L. Kulseng, Z. Yu, Y. Wei and Y. Guan, Lightweight mutual authentication and ownership transfer for RFID systems, *In Proceedings of the 29th conference on Information Communications, INFOCOM'10*, pp. 1–5, 2010.

[145] A. Khisti, G. Wornell, A. Wiesel and Y. Eldar, On the Gaussian MIMO wiretap channel, *IEEE International Symposium on Information Theory, ISIT'07*, pp. 2471–2475, 2007.

[146] X. Lai, Higher order derivatives and differential cryptanalysis, *Kluwer International Series in Engineering and Computer Science*, pp. 227–227, 1994.

[147] G. Leander, On linear hulls, statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. *Advances in Cryptology, EUROCRYPT'11*, LNCS 6632, pp. 303–322, 2011.

[148] Z. Li, G. Gong and Z. Qin, Secure and efficient LCMQ entity authentication protocol, *Technical Report, CACR 2010-21, University of Waterloo*, pp. 1–24, 2010.

[149] C.H. Lam and M. Aagaard, Hardware implementations of multi-output Welch-Gong ciphers, *Technical Report, CACR 2011-01, University of Waterloo*, pp. 1–16, 2011.

[150] G. Leander, M.A. Abdelraheem, H. AlKhzaimi and E. Zenner, A cryptanalysis of PRINTCipher: the invariant subspace attack, *Advances in Cryptology, CRYPTO'11*, LNCS 6841, pp. 206–221, 2011.

[151] Y.K. Lee, L. Batina, D. Singelée and I. Verbauwhede, Low-cost untraceable authentication protocols for RFID, *In Proceedings of the third ACM conference on Wireless network Security, WiSec'10*, pp. 55–64, 2010.

[152] Y. Luo, Q. Chai, G. Gong and X. Lai, WG-7, a lightweight stream cipher with good cryptographic properties, *IEEE Global Telecommunications Conference, GLOBECOM'10*, pp. 1–6, 2010.

[153] Y.W. Law, J. Doumen and P. Hartel, Survey and benchmark of block ciphers for wireless sensor networks, *Sensor Networks, ACM Transactions on*, vol. 2, no. 1, pp. 65–93, 2006.

[154] C. Lim and T. Korkishko, mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors, *Information Security Applications*, LNCS 3786, pp. 243–258, 2006.

[155] C. Lim and T. Kwon, Strong and robust RFID authentication enabling perfect ownership transfer, *Information and Communications Security*, LNCS 4307, pp. 1–20, 2006.

[156] B. Liang, Y. Li, C. Ma, T. Li and R. Deng, On the untraceability of anonymous RFID authentication protocol with constant key-lookup, *Information Systems Security*, pp. 71–85, 2009.

[157] G. Leander and A. Poschmann On the classification of 4 bit s-boxes, *Arithmetic of Finite Fields*, LNCS 4547, pp. 159–176, 2007.

[158] G. Leander, C. Paar, A. Poschmann and K. Schramm, New lightweight DES variants. *Fast Software Encryption, FSE'07*, LNCS 4593, pp. 196–210, 2007.

[159] T. Liu, V. Prabhakaran and S. Vishwanath, The secrecy capacity of a class of parallel Gaussian compound wiretap channels, *IEEE International Symposium on Information Theory, ISIT'08*, pp. 116–120, 2008.

[160] R. Maes, PUFs: Physical(ly) Unclonable Functions, *http://homes.esat.kuleuven.be/rmaes/puf.html*, 2011.

[161] W. Mao, Timed-release cryptography, *Selected Areas in Cryptography, SAC'01*, L-NCS 3897, pp. 342–357, 2001.

[162] K. Mandal, X. Fan and G. Gong, A lightweight pseudorandom number generator for EPC Class 1 Gen2 RFID tags, *Western European Workshop on Research in Cryptology, WEWoRC'11*, pp. 91–92, 2011.

[163] M. Nele, G. Jan, B. Preneel and I. Verbauwhede, A low-cost implementation of Trivium, *In Proceedings of SASC 2008*, pp. 197–204, 2008.

[164] J. Munilla, A. Ortiz and A. Peinado, Distance bounding protocols with void-challenges for RFID, *Workshop on RFID Security, RFIDSec'06*, 2006.

[165] J. Munilla and A. Peinado, Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels, *Wireless communications and mobile computing*, vol. 8, no. 9, pp. 1227–1232, 2008.

[166] D. Molnar, A. Soppera, D. Wagner, A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags, *Selected Areas in Cryptography, SAC'05*, LNCS 3897, pp. 276–290, 2005.

[167] M. Madhavan, A. Thangaraj, Y. Sankarasubramanian and K. Viswanathan, NLHB: a nonlinear hopper-blum protocol, *IEEE International Symposium on Information Theory Proceedings, ISIT'10*, pp. 2498–2502, 2010.

[168] H. Mahdavifar and A. Vardy, Achieving the secrecy capacity of wiretap channels using polar codes, *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6428–6443, 2011.

[169] D. Molnar and D. Wagner, Privacy and security in library RFID: issues, practices, and architectures. *In Proceedings of the 11th conference on Computer and Communications Security, CCS'04*, pp. 210–219, 2004.

[170] NXP Semiconductors, P5Cx012/02x/40/73/80/144 family, Secure dual interface and contact PKI smart card controller, *http://www.nxp.com/documents/data_sheet/P5CX012_02X_40_73_80_144_FAM_SDS.pdf*, 2008.

[171] K. Nohl, D. Evans, S. Starbug and H. Plötz, Reverse-engineering a cryptographic RFID tag, *In Proceedings of the 17th conference on USENIX Security Symposium, USENIX'08*, pp. 185–193, 2008.

[172] Y. Nawaz and G. Gong, WG: A family of stream ciphers with designed randomness properties, *Information Science*, vol. 178, no. 7, pp. 1903–1916, 2008.

[173] C. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini, RFID privacy models revisited, *In European Symposium on Research in Computer Security, ESORICS'08*, LNCS 5283, pp. 251–266, 2008.

[174] L.M. Ni, D. Zhang and M.R. Souryal, RFID-based localization and tracking technologies, *Wireless Communications, IEEE*, vol. 18, no. 2, pp. 45–51, 2011.

[175] Y. Oren and M. Feldhofer, A low-resource public-key identification scheme for RFID tags and sensor nodes, *In Proceedings of the second ACM conference on Wireless network Security, WiSec'09*, pp. 59–68, 2009.

[176] K. Ouafi, R. Overbeck and S. Vaudenay, On the security of $HB^{\#}$ against a Man-in-the-Middle attack, *Advances in Cryptology, AsiaCrypt'08*, LNCS 5350, pp. 108–124, 2008.

[177] K. Ouafi and R.C.W. Phan, Privacy of recent RFID authentication protocols, *Information Security Practice and Experience*, LNCS 4991, pp. 263–277, 2008.

[178] M. Ohkubo, K. Suzuki, S. Kinoshita and others, Cryptographic approach to privacy-friendly tags, *RFID Privacy Workshop*, vol. 82, pp. 1–9, 2003.

[179] K. Ouafi and S. Vaudenay, Smashing SQUASH-0, *Advances in Cryptology, EUROCRYPT'09*, LNCS 5479, pp. 300–312, 2009.

[180] L.H. Ozarow and A.D. Wyner, Wire-tap channel II, *Advances in Cryptology, EUROCRYPT'84*, LNCS 209, pp. 33–50, 1985.

[181] S. Piramuthu, On existence proofs for multiple RFID tags, *ACS/IEEE International Conference on Pervasive Services*, pp. 317–320, 2006.

[182] P. Peris-Lopez, J.C. Hernandez-Castro, J.M.E. Tapiador, E. Palomar and J.C.A. van der Lubbe, Cryptographic puzzles and distance-bounding protocols: practical tools for RFID security, *IEEE International Conference on RFID, RFID'10*, pp. 45–52, 2010.

[183] A. Poschmann, S. Ling and H. Wang, 256 bit standardized crypto for 650 GE–GOST revisited, *Cryptographic Hardware and Embedded Systems, CHES'10*, LNCS 6225, pp. 219–233, 2011.

[184] C. Pendl, M. Pelnar and M. Hutter, Elliptic curve cryptography on the WISP UHF RFID tag, to appear *In Proceedings of RFIDSec'11*. pp. 1–16, 2011.

[185] K.B. Rasmussen and S. Capkun, Location privacy of distance bounding protocols, *In Proceedings of the 15th ACM conference on Computer and Communications Security, CCS'08*, pp. 149–160, 2008.

[186] K.B. Rasmussen and S. Capkun, Realization of RF distance bounding, *In Proceedings of the USENIX Security Symposium, USENIX'10*, pp. 1–13, 2010.

[187] M. Rieback, B. Crispo and A. Tanenbaum, RFID guardian: A battery-powered mobile device for RFID privacy management, *Australasian Conference on Information Security and Privacy, ACISP'05*, LNCS 3574, pp. 184–194, 2005.

[188] C. Rolfes, A. Poschmann, G. Leander and C. Paar, Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents. *In Proceedings of the 8th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications*, LNCS 5189, pp. 89–103, 2008.

[189] M.J.O. Saarinen, Cryptanalysis of Hummingbird-1, *Fast Software Encryption, FSE'11*, LNCS 6733, pp. 328–341, 2011.

[190] A. Shamir, Memory efficient variants of public-key schemes for smart card applications, *Advances in Cryptology, EUROCRYPT'94*, LNCS 950, pp. 445–449, 1995.

[191] A. Shamir, SQUASH–a new MAC with provable security properties for highly constrained devices such as RFID tags, *Fast Software Encryption, FSE'08*, LNCS 5086, pp. 144–157, 2008.

[192] M. Safkhani, N. Bagheri, S.K. Sanadhya and M. Naderi, Cryptanalysis of improved Yeh *et al.* authentication Protocol: an EPC Class-1 Generation-2 standard compliant protocol, *Cryptology ePrint Archive, Report 2011/426*, pp. 1–9, 2011.

[193] A. Shamir and E. Biham, Differential cryptanalysis of DES-like cryptosystems, *Advances in Cryptology, CRYPTO'90*, LNCS 537, pp. 2–21, 1990.

[194] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, Piccolo: an ultra-lightweight blockcipher, *Cryptographic Hardware and Embedded Systems, CHES'11*, LNCS 6917, pp. 342–357, 2011.

[195] M. Strasser, S. Capkun, C. Popper and M. Cagalj, Jamming-resistant key establishment using uncoordinated frequency hopping, *29th IEEE Symposium on Security and Privacy, S&P'08*, pp. 64–78, 2008.

[196] B. Song and C.J. Mitchell, RFID authentication protocol for low-cost tags, *In Proceedings of the first ACM conference on Wireless network Security, WiSec'08*, pp. 140–147, 2008.

[197] B. Song and C.J. Mitchell, Scalable RFID security protocols supporting tag ownership transfer, *Computer Communications*, vol. 34, no. 4, pp. 556–566, 2011.

[198] D.R. Stinson and J.L. Massey, An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions, *Journal of Cryptology*, vol. 8, no. 3, pp. 167–173, 1995.

[199] M.K. Simon, J.K. Omura, R.A. Scholtz and B.K. Levitt, *Spread Spectrum Communications Handbook*, McGraw-Hill Professional Publishing, 2001.

[200] O. Savry, F. Pebay-Peyroula, F. Dehmas, G. Robert and J. Reverdy, RFID noisy reader: how to prevent from eavesdropping on the communication? *Cryptographic Hardware and Embedded Systems, CHES'07*, LNCS 4727, pp. 334–345, 2007.

[201] J. Saito and K. Sakurai, Grouping proof for RFID tags, *19th International Conference on Advanced Information Networking and Applications, AINA'05*, vol. 2, pp. 621–624, 2005.

[202] H. Sun and W. Ting, A Gen2-based RFID authentication protocol for security and privacy, *Mobile Computing, IEEE Transactions on*, vol. 8, no. 8, pp. 1052–1062, 2009.

[203] N. Saxena and J. Voris, We can remember it for you wholesale: implications of data remanence on the use of RAM for true random number generation on RFID tags, *Workshop on RFID Security, RFIDSec'09*, pp. 1–13, 2009.

[204] Tagent UWB reader and passive UWB tag, *http://www.tagent.com/?PageID=110*, 2011.

[205] Texus Instrument, MSP430F2132, 16-bit ultra low power microcontroller, *http://www.ti.com/product/msp430f2132*, 2011.

[206] G. Tsudik, YA-TRAP: yet another trivial RFID authentication protocol, *International Conference on Pervasive Computing and Communications, Percom'06*, pp. 640–643, 2006.

[207] A. Thangaraj, S. Dihidar, A.R. Calderbank, S.W. McLaughlin and J.M. Merolla, Applications of LDPC codes to the wiretap channel, *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2933–2945, 2007.

[208] R. Trujillo-Rasua, B. Martin and G. Avoine, The Poulidor distance-bounding protocol, *Workshop on RFID Security, RFIDSec'10*, LNCS 6370, pp. 239–257, 2010.

[209] J. Toonstra and W. Kinsner, Transient analysis and genetic algorithms for classification, *Communications, Power, and Computing. Conference Proceedings. IEEE WESCANEX'95*, vol. 2, pp. 432–437, 1995.

[210] Universal Software Radio Peripheral, *http://code.ettus.com/ redmine/ettus/projects/uhd/wiki*, 2011.

[211] VERT900 Antenna, *http://www.ettus.com/downloads/VERT900.pdf*, 2011.

[212] S. Vaudenay, On privacy models for RFID, *Advances in Cryptology, AsiaCrypt'07*, LNCS 4833, pp. 68–87, 2007.

[213] G. Vannucci, A. Bletsas and D. Leigh, A software-defined radio system for backscatter sensor networks, *Wireless Communications, IEEE Transactions on*, vol. 7, no. 6, pp. 2170–2179, 2008.

[214] Wireless Identification and Sensing Platform (WISP), *http://wisp.wikispaces.com*, 2011.

[215] S.A. Weis, Security and privacy in radio-frequency identification devices, *Master Thesis, Massachusetts Institute of Technology*, pp. 49–51, 2003.

[216] A.D. Wyner, The wire-tap channel, *Bell Systems Technical Journal*, vol. 54, pp. 1355–1387, 1975.

[217] H. Wu and B. Preneel, Resynchronization attacks on WG and LEX, *Fast Software Encryption, FSE'06*, LNCS 4047, pp. 422–432, 2006.

[218] J. Wu and D.R. Stinson, How to improve security and reduce hardware demands of the WIPR RFID protocol, *IEEE International Conference on RFID, RFID'09*, pp. 192–199, 2009.

[219] L. Xiao, L. Greenstein, N. Mandayam and W. Trappe, Using the physical layer for wireless authentication in time-variant channels, *Wireless Communications, IEEE Transactions on*, vol. 7, no. 7, pp. 2571–2579, 2008.

[220] W. Xu, W. Trappe and Y. Zhang, Anti-jamming timing channels for wireless networks, *In Proceedings of the first ACM Conference on Wireless network Security, WiSec'08*, pp. 203–213, 2008.

[221] D.H. Yum, J.S. Kim, S.J. Hong and P.J. Lee, Distance bounding protocol for mutual authentication, *Wireless Communications, IEEE Transactions on*, vol. 10, no. 2, pp. 592–601, 2011.

[222] D. Zanetti, B. Danev and S. Capkun, Physical-layer identification of UHF RFID tags, *In Proceedings of the sixteenth annual International Conference on Mobile Computing and Networking, MobiCom'10*, pp. 353–364, 2010.

[223] B. Zhu and G. Gong, Guess-then-meet-in-the-middle attacks on the KTANTAN family of block ciphers, *Cryptology ePrint Archive, Report 2011/619*, pp. 1–14, 2011.

[224] J. Zhou and J. Shi, RFID localization algorithms and applications – a review, *Journal of Intelligent Manufacturing*, vol. 20, no. 6, pp. 695–707, 2009.

[225] D. Zanetti, P. Sachs and S. Capkun, On the practicality of UHF RFID fingerprinting: how real is the RFID tracking problem, *Privacy Enhancing Technologies, PET'11*, LNCS 6794, pp. 97–116, 2011.