Lightweight Cryptography for Passive RFID Tags

David, Mathieu

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
David, M. (2012). *Lightweight Cryptography for Passive RFID Tags*.

Lightweight Cryptography for Passive RFID Tags

Ph.D Thesis

by

MATHIEU DAVID

December 2011

Supervisor

Prof., Dr. Techn. Torben Larsen

Assessment Committee

Prof. Gildas Avoine, Université Catholique de Louvain (UCL), Belgium

Postdoc Pedro Peris-Lopez, Delft University of Technology (TUD), The Netherlands

Assoc. Prof. Petar Popovski, Aalborg Universitet (AAU), Denmark

Moderator

Assoc. Prof. Ole Kiel Jensen, Aalborg Universitet (AAU), Denmark

by Mathieu David

Aalborg University

Niels Jernes Vej 12

Aalborg Øst

Denmark

e-mail: md@es.aau.dk, mr.mathieu@gmail.com

Printed in Aalborg, Denmark

I dedicate this Thesis

to my closest family relatives for their continuous support...
(... even without understanding what I was doing).

AES
achieved
achieve
able
additional
around
author
authentication
A2U2
A2U2A2U2
available
AESAES
circuit
also
block
applications
blocks
compare
algebraic
challenge
area
building
analysis
called
bogdanov2007present
attacks
broken
best
classification
based
bits
cipherblock
chosen
chapter
bit
architecture
array
ciphertext
attacker
application
approach
attack
ciphers
consider
comparison
determine
circuits
communication
common
case
computational
constrained
Chapter
challenges
complexity
details
clocked
cipher
chip
data
due
begin
designed
Boolean
described
david2011a2u2
counter
compared
cipherstream
estimate
equations
encryption
clock
electronics
cycles
cryptanalysis
Evaluation
CMOS
DES
Consequently
de2009katan
cryptography
capable
cost
design
different
estimated
four
electronicsprinted
fact
feedback
extremely
full
et
evaluate
consumption
designs
find
functions
considered
end
Cryptography
enough
device
existing
gate
even
following
cryptographic
hardware
increase
functionhash
generated
ensure
family
differential
filter
frequency
example
first
equation
gates
generate
function
Grain
far
keys
implemented
Figure
issue
guess
Furthermore
integrated
ID
IC
left
initial
issues
increasing
devices
high
implementation
ii
lower
IV
information
large
However
given
length
key
obtain
linear
kbps
literature
must
initialization
ink
may
hash
input
kHz
label
less
level
least
operation
lightweight
nonlinear
low-cost
meet
operating
paper
need
knudsen2011printcipher
presents
limited
LFSR
particular
Lightweight
mCRYPTON
metric
per
implementations
operations
minimum
new
order
know
mechanisms
one
keystream
known
parameters
memory
passive
low
number
location
main
requirement
MM
mechanism
plaintexts
PRESENT
optimized
obtained
output
plaintext
performance
possible
physical
present
non-linear
pairs
recover
privacy
optimization
numbers
random
polynomial
reader
related
PRINTcipher
platforms
power
often
previous
providing
period
registers
needs
primitive
propose
protocols
primitives
problem
rate
printed
product
recovery
probability
processes
part
presented
requirements
register
remains
performed
published
Section
PUF
proposed
required
relatively
result
round
range
resource
provide
response
require
reported
RFID
real
rounds
recent
public
size
see
shown
section
SEA
right
research
requires
simple
protocol
several
sigma
second
Tags
process
silicon
secure
since
sequence
security
state
solve
set
stored
secret
results
strong
significant
system
systems
Therefore
selected
solution
step
solutions
scheme
Thus
tag
tags
stream
suitable
specific
small
time
TEA
weaknesses
single
thesis
still
throughput
two
Table
similar
SQUASH
uses
Security
techniques
using
used
use
thus
target
μW
universal
technologies
well
value
terms
three
technology
threat
update
values
various
version
work
written
Trivium
without
variables
updated
years

Abstract

In 2011, we are entering a decade where Radio Frequency IDentification (RFID) systems will become ubiquitous, slowly but surely replacing its old ancestor: the barcode. With the RFID technology come many advantages such as faster retailing, continuous control along the supply chain, real-time monitoring and localization of items, etc. However, all these benefits come to the condition of secure systems, especially in sensitive application areas such as military, finance, pharmaceutics, etc. Additionally, the privacy aspect involved with this technology could become a major issue in the perspective of a global adoption. In the past few years, an increasing number of researchers concentrates their efforts into providing secure solutions for RFID systems.

After several attempts to integrate traditional cryptographic primitives into small, embedded, and extremely resource constrained devices, the results were mostly unsatisfactory. As a conclusion, a new branch of cryptography, commonly called *Lightweight Cryptography*, emerged to address the issues of these tiny ubiquitous devices.

This Thesis presents a comprehensive engineering to lightweight cryptography, proposes a classification and explores its various ramifications by giving key examples in each of them. We select two of these branches, ultralightweight cryptography and symmetric-key cryptography, and propose a cryptographic primitive in each of them. In the case of symmetric-key cryptography, we propose a stream cipher that has a footprint among the smallest in the published literature and aims at being implemented on printed electronics RFID tags.

Then, we compare different cryptographic primitives based on their key parameters: throughput, area, power consumption and level of security. Our main concern is the integrability of these selected primitives into real passive

RFID tags. Therefore, in order to go beyond a comparison of the different parameters, we propose a metric that combines all their characteristics into one single value. This metric also has the advantage of being customizable, depending on the requirement of an integrator for a particular application.

Finally, we conclude that the research for finding robust cryptographic primitive in the branch of lightweight cryptography still has some nice days ahead, and that providing a secure cryptosystem for printed electronics RFID tags remains an open research topic.

Keywords: Lightweight Cryptography, RFID, Security, Printed Electronics, Stream Cipher, Comparison Metric.

Dansk Abstrakt

I 2011 går vi ind i et årti, hvor RFID-systemer bliver allestedsnærværende og langsomt, men sikkert vil afløse dets gamle forgænger: stregkoden. Med RFID-teknologi er der mange fordele, såsom hurtigere detailhandel, kontinuerlig kontrol af logistikkæden, realtids monitorering og lokalisering af objekter osv. Men alle disse fordele kommer på betingelse af sikre systemer, specielt i følsomme applikations-miljøer såsom militæret, finansverdenen, medicinalindustrien osv. Desuden kan folks privatliv blive et problem med denne teknologi set i et globalt perspektiv. I de sidste år har et stigende antal forskere koncentreret deres indsats for at levere sikre løsninger til RFID systemer.

Efter adskillige forsøg på at integrere traditionel kryptografiske primitiver ind i små, indlejrede og ekstremt ressourcebegrænsede enheder var resultaterne meget utilfredsstillende. Som følge deraf startede en ny gren af kryptografi, kendt som *Letvægts-Kryptografi*, der henvender sig til disse små, allestedsnærværende enheder.

Denne afhandling præsenterer et omfattende teknisk perspektiv til letvægtskryptografi, foreslår en klassifikation og undersøger dens forskellige forgreninger ved at give eksempler på hver af dem. Vi vælger to af disse forgreninger: ultra-lightweight cryptography og symmetric-key cryptography og foreslår en kryptografisk primitiv for hver af dem. For symmetric-key cryptography foreslår vi et stream cipher med et areal, der hører til blandt de mindste i den udgivne litteratur og som sigter mod implementation i printede elektroniske RFID-mærker.

Derefter sammenligner vi forskellige kryptografiske primitiver baseret på deres nøgleparametre: produktion, areal, effektforbrug og sikkerhedsniveau. Vores hovedproblem er integrerbarheden af de valgte primitiver ind i eksisterende passive RFID-mærker. For at gå endnu længere end en sammenligning af forskellige parametre, foreslår vi derfor en måleenhed, der kombinerer alle disse karakteristika. Denne måleenhed har desuden fordelen af at være definerbar, afhængig af kravene til en integrator til en bestemt applikation.

Til sidst konkluderer vi, at forskning i at finde robuste kryptografiske primitiver inden for letvægts-kryptografi stadig har nogle gode dage foran sig og problemet med at levere et sikkert kryptografisk system til printede RFID-mærker forbliver et åbent forskningsområde.

Nøgleord: Letvægts-Kryptografi, RFID, Sikkerhed, Printet Elektronik, Stream Cipher, Sammenligningsmåleenhed.

Thanks to Karsten Fyhn for the translation
and Eva Hansen for the proof-reading.

Acknowledgements

My first thought goes to my current supervisor. While I was facing an NP-hard problem[1] for almost two years and started to lose hope, Torben Larsen gave me a hand to solve it, despite the fact that he was already overwhelmed with work. If I graduate today, this is mainly thanks to him, and I can never thank him enough for that. Torben, **THANK YOU!**

On the other side of the World – down under, as they say – stands another brilliant researcher who hosted me, inspired me, and guided me during the six months I spent with him. We had passionate discussions, he spent countless hours to help me, and he always pushed me to give the best ... and then a bit more. Damith Ranasinghe was my supervisor during the most productive period of my thesis and more than a mentor, he became a friend. Damith, **THANK YOU!**

I want to thank all the colleagues who helped me in different aspects of my thesis including Yannick Le Moullec, Mehmood-Ur-Rehman Awan, Jan Mikkelsen, Joachim Rodrigues (from Lund University), Olav Geil, and Hans Hüttel.

I want to thank all the colleagues who did not help at all with my thesis... but instead brought me so much more. Among the superheroes, you can find Jose Gutierrez, Michael Jensen, Tahir Riaz, Jens Myrup Pedersen, Carlo Galiotto, Bilge Kartal, Carrie Peterson, Alexandre Fleury, Patrick Lemmonier, Guillaume Monghal, Ruben Grigoryan, Pawel Pankiewicz, Karsten Fyhn and many more... but I promised my thesis will be shorter than 200 pages.

I do not forget to thanks Dorthe Sparre and Eva Hansen for their time, kindness and permanent support.

[1] NP-hard stands for *Non Polynomial-hard* and refers to a problem one cannot solve in a lifetime.

I also take the occasion to thank all my international friends who were the elements of my surrogate family here in Aalborg and made time fly. A special thanks to my closest friends Karsten (Germany), Ubaldo (Mexico), Luca (Italy), Jette (Denmark), Hernan (Chile), Manuel (Portugal) and the two who convinced me to start a Ph.D: João Figueiras (Portugal) and Gian Paolo Perrucci (Italy).

Last but not least, a big thanks to my girlfriend Agnieszka for her permanent support, and to my family back in France who made this study abroad possible.

MATHIEU DAVID

Preface

A large part of the content of this thesis has already been published (or submitted) in different papers. Following is the list of these publications, together with the contribution of the author to the respective papers, and how they have been reused in this thesis.

Conference papers

- **M. David** and N.R. Prasad. Providing strong security and high privacy in low-cost rfid networks. *Security and Privacy in Mobile Information and Communication Systems*, Turin, Italy, June 3-5, 2009, pages 172–179 [41].

    ✧ This paper is presented in Chapter 3. The author of this thesis is the main originator and contributor to the paper. He has conducted the major part of the analysis, has designed the security protocol, and has analyzed and concluded on the achieved results. In Chapter 3, Section 3.1.1 to Section 3.5 are directly copied from the paper. The remaining sections of that chapter are new in this thesis. Additionally, Table 3.1 has been updated in this thesis to match with the latest published literature.

- **M. David**, D.C. Ranasinghe, and T. Larsen. A2U2: A stream cipher for printed electronics RFID tags. *RFID 2011, the fifth IEEE International Conference on RFID*, Orlando, Florida, USA, April 12-14, 2011, pages 176–183 [42].

    ✧ This paper is presented in Chapter 4. The author of this thesis is the main originator and contributor to the paper. He has conducted the major part of the analysis and the design of the cipher, and has analyzed and concluded on the achieved results. He has written the original draft of the paper and has corrected it several times, based on the co-authors reviews. In Chapter 4, Section 4.1 to Section 4.5 are directly copied from the paper. Section 4.6 and Section 4.8 are new material in this thesis.

- M. A. Abdelraheem, J. Borghoff, E. Zenner and **M. David**. Cryptanalysis of the lightweight cipher A2U2. *Thirteen IMA International Conference on Cryptography and Coding*, Oxford, UK, accepted for publication in LNCS, 16 pages, December 12-15, 2011.

  ✧ This paper is included in Appendix A. The author of this thesis initiated the cooperation on the paper, and provided information on the background and details of the A2U2 cipher. He wrote the section on "Necessary changes and possible improvements" to the cipher based on the co-authors input, which identified the three main concerns. In Chapter 4, Section 4.7.2 is directly copied from the paper, while Section 4.7.1 is a summary of the main attacks presented in the paper.

Journal (submitted)

- D.C. Ranasinghe, **M. David** and Q. Z. Sheng. Lightweight Cryptography: Classification and Evaluation, *submittted in January 2011 to the Journal of Cryptology*, 50 pages.

  ✧ This paper is presented in Chapters 1, 2, 5, and 6. The author of this thesis wrote the first draft of the paper, based on a previous (unpublished) report written by Damith C. Ranasinghe, and updated the numbers in Section 2.5, based on the most recent published literature. He has conducted the power consumption analysis and written Section 5.1.4. He is the originator of the metric proposed in the paper and has written the Section 5.3.1. In Chapter 1, the Introduction and Problem Statement sections are copied from the paper. The remaining part of that chapter is new material in this thesis. In Chapter 2, the four sections are copied from the paper. In Section 2.5, some numbers have been updated since the submitted version of the paper, including some cryptanalysis results of KTANTAN[44]. The section concerning PRINTcipher [96] is new material. In Chapter 5, the four section are copied from the paper, but have been updated accordingly to the inclusion of A2U2 [42] and PRINTcipher [96] in the several comparisons. Finally, in Chapter 6, some of the paragraphs are copied from the paper, while the remaining paragraphs are written for this thesis.

Contents

Chapter 1

Introduction

"Learning **seriously** affects your brain."
– *Aalborg University*

A key goal in the development of security mechanism for extremely low cost platforms, such as passive RFID tags, is the design of low cost security primitives that consume very little power (in the order of few micro Watts) and have adequate performance (throughputs of hundreds of kilobits per second). Their aim is to support various end-user applications such as product brand protection and authentication, disposable mass transit fare card ticketing, consumer retail product promotions, and embedded product intelligence [100]. For example, in a pharmaceutical supply chain, various parties may want to authenticate a vial of medicine or only allow authorized parties to access a bottle's globally unique identifier.

Most modern cryptographic mechanisms cannot be applied directly to low cost platforms, such as RFID technology, because their design goals do not meet the platform and regulation specific limitations offered by such innovative technologies [138, 137]. Moreover, encryption standards such as AES (Advanced Encryption Standard) [36], RSA (Rivest, Shamir and Adleman) [142], and Elliptic Curve Cryptography [124, 98] use larger and larger key sizes, supported by increasing computational capabilities of the target hardware platforms to achieve higher levels of security. However, modern pervasive technologies (e.g. low cost RFID) have limited memory and no processing capability. They are instead application specific integrated circuits (ASICs), whose operational aspects are further constrained by environmental factors, such as indeterministic power loss, and regulations, which limit period of tag engagement and available bandwidth [51, 83]. Consequently, an alternative approach is demanded by low cost platforms such as passive RFID technology. The resource constraints of the technology dictate the rules of security algorithm designs that must have an hardware implementation cost as low as possible, while ensuring adequate performance and security, in order to facilitate the use of mechanisms in real applications.

The attractiveness of applications enabled by low cost computing devices and the lack of secure and suitable solutions have initiated a global research effort to develop low cost cryptographic primitives and implementations. Its objective is to meet (i) the needs of extremely resource limited platforms, (ii) the security goals of those platforms, and (iii) the end-user requirements for adequate performance. Nevertheless, the growth of mobile device platforms over the past twenty years has pushed researchers to focus their efforts on cryptographic mechanisms capable of delivering fast, low cost and low power implementations in hardware (such as A5[1] in GSM mobile phones). However, approaches to optimize existing primitives for small hardware implementations have failed to deliver practicable solution for low cost devices [55, 54, 72]. This forced researchers to re-think the design of security primitives as opposed to optimizing existing designs [20, 44, 42, 96]. This view is clearly evident in recent European projects such as New European Schemes for Signatures, Integrity, and Encryption (NESSIE) [126], and ECRYPT Stream Cipher Project (eSTREAM) [52], as well as ECRYPT I and II (European Network of Excellence for Cryptology) [1], formulated to address the design vacuum.

1.1 Problem Statement

Given the critical importance of security for low cost pervasive technologies, an emerging array of security mechanisms often categorized under *lightweight cryptography* has sought to meet the challenge of developing fast and efficient security mechanisms for resource constrained environments. Many primitives and low cost implementations have been proposed recently [158, 66, 144, 105, 53]. Although the majority are presented as lightweight cryptographic primitives, they are based on various design goals and requirements that do not

[1] A5/1 (and A5/2) is a stream cipher used to encrypt data in the GSM cellular telephone standard.

necessarily satisfy those of extremely low cost platforms (e.g. DES [105], TRIVIUM [53] and mCRYPTON [112] implementations are too large for passive RFID tags). Furthermore, the claim that a primitive is lightweight seems to be based entirely on a personal perspective rather than a quantitative analysis. Unfortunately, a clear view of what constitute a lightweight primitive or a coherent discourse in this area is lacking in the literature.

Moreover, reporting of developments in lightweight cryptography is tangled in a web of variables such as:

- levels of security,
- fabrication process parameters and technology used in the implementation (area),
- variations in power consumption measurements (power),
- clock rates at which the primitives are run (throughput).

Consequently, it is becoming increasingly difficult to compare their suitability for practical implementation in resource constrained platforms.

To the best of the authors knowledge, [50] is the only survey of lightweight cryptographic primitives that has attempted to compare various developments in this area. Unfortunately, a detailed comparison that is agnostic to numerous variations in the cipher design and implementation, as well as a careful consideration of a primitive's suitability for resource constrained devices is not presented. In addition, cryptography being heavily embedded in the discipline of pure mathematics, this survey has tended to exclude developments in other multidisciplinary areas capable of delivering security to extremely low cost devices. This is indeed the case with Physically Unclonable Functions [114, 62].

1.2 Scientific Approach

This thesis is written from an engineering perspective, where a strong emphasis is put on the integrability of a primitive into a real passive RFID tag. This approach is more practical rather than purely mathematical. It is a deliberate choice made by the author, accordingly to his educational background in telecommunication and networks. This thesis is articulated around a classical research method in three steps: analysis, proposition, and comparison, explained as follows:

Analysis The mandatory first step into any new research topic is a thorough analysis of the State-of-the-Art (SoA) within the field. Lightweight cryptography has attracted a growing number of researchers for the past ten years due to the emergence of ubiquitous mobile devices and mobile networks that require embedded security. Here, the author focuses on lightweight cryptography applied to passive RFID tags (i.e. cryptographic primitives that can be implemented in less than 2000 gates). A classification of the primitives in different categories is introduced to evaluate where some contribution would have the most impact and/or fill a gap.

Proposition Based on the analysis of the State-of-the-Art, the author identify two areas where a contribution would have some significance. First, with the design of an ultra-lightweight security primitive which, initially, appears to be the easiest entry to the world of cryptographic primitives design. Second, the author proceeds with a more elaborate security primitive by designing the smallest stream cipher possible to be implemented on an emerging and challenging technology: printed electronics RFID tags.

Comparison A strong way of evaluating a research work is most likely to compare it with the existing published literature. This is the approach taken here with both of the proposed cryptographic primitives compared to similar works. The comparison methods are different for each proposed primitives. The ultra-lightweight primitive (Chapter 3) is compared in terms of (im)possible attacks, since not many proposals provide an implementation of their primitive. The second primitive, A2U2 (Chapter 4), is compared in terms of implementation performances, since the cryptanalysis of such ciphers usually can take up to several months or years. Additionally, the author go beyond the classical comparison scheme and propose a comparison metric to evaluate the different ciphers (see section 1.4). In order to design this metric, the ciphers are evaluated from a different perspective, from the side of an integrator who needs to implement a security protocol into a given RFID application.

1.3 Scientific Challenge

The main challenge raised in this thesis is to develop a realisticly secure cryptosystem for printed electronics RFID tags. By realisticly secure, the author means that the cryptosystem should be secure relatively to any given application of printed electronics RFID tags. From a mathematician point of view, a cryptosystem is not secure if it can be broken in a polynomial amount of time. From an engineering point of view, a cryptosystem is secure if it remains unbroken for the entire lifetime of a given application. For example, if a movie ticket carrying an RFID tag is secure for five years, this can be considered as sufficiently secure (given that revealing its secret information would only impact this sole ticket).

1.4 Contributions

The key contributions of the author are summarized as the following:

- **A definition and a classification of lightweight cryptographic primitives** that incorporates recent developments in the area of cryptography for extremely resource limited platforms such as passive RFID technologies. The aim is to propose a common foundation for cryptographers working in the area of developing lightweight cryptographic primitives. The classification is comprehensive, meaning that it includes the multidisciplinary approaches taken by researchers to develop new solutions to the challenging problem of providing security to devices with limited resources. The classification is illustrated using existing cryptographic primitives published as being suitable for extremely resource limited devices such as passive RFID tags.

- **The designof an ultra-lightweight primitive** to improve on the work carried out previously by the research community. The aim of this protocol is to improve on the security issues (tracking, mutual authentication) of similar previously published works, while at the same time attempting to reduce the computational effort on the tag side.

- **The design of a tiny stream cipher** for implementation in printed electronics RFID tags, filling two open gaps in the literature: (i) a tiny stream cipher implementable on less than 1,000 gates, and (ii) a cipher small enough to meet the heavy constraints of integration in printed electronics.

- **The evaluation of the surveyed cryptographic primitives** along four dimensions, (i) cost of implementation in hardware, (ii) performance in terms of throughput, (iii) performance in terms of power consumption and, (iv) the level of security each primitive is capable of providing.

- **The development of the Weighted nOrmalised cOst Power and Throughput (WOOPT) metric** for comparing and contrasting various lightweight cryptographic primitives identified through the preceding classification. The proposed metric incorporates multiple diverged characteristics of lightweight primitives, in particular their cost, power consumption and performance to evaluate their suitability for practical applications in resource constraint devices. The usefulness of the WOOPT metric is demonstrated to collectively assess the lightweight cryptographic primitives and help the selection of a primitive to meet application specific requirements.

1.5 Thesis Organization

The aim of this thesis is to assemble the work published (or submitted) during the entire period of the PhD studies into a monograph in order to avoid the abrupt topic switches of a collection of paper and to give an improved reading experience. In this way, Chapter 2 to Chapter 5 regroup the peer-reviewed material and is slightly adapted to match the general flow of the thesis. In the introduction of these chapters, the respective publication(s) is mentioned to the reader. Besides Chapter 1, which introduces the work achieved in the thesis, the remainder of the thesis is organized as follows:

Chapter 2 briefly introduces the RFID technology and describes the platform-specific requirements for lightweight cryptographic primitives. It describes the development of the emerging area of lightweight cryptography for resource limited environments. It proposes a classification of lightweight cryptographic primitives. Finally, it expounds the research outcomes under the proposed classification reported as being capable of providing security services such as authentication, confidentiality and anonymity for resource limited platforms. **Paper used:** *Lightweight Cryptography: Classification and Evaluation.*

Chapter 3 presents an ultra-lightweight primitive design for passive RFID tags. A short introduction and background related to the specific area is given. The primitive is then compared to the existing related works. Additionally, some conclusions are drawn by taking into account the published cryptanalysis of the protocol. **Paper used:** *Providing strong security and high privacy in low-cost rfid networks.*

Chapter 4 describes a stream cipher for printed electronics RFID tags. Here again, a comprehensive introduction and related works are presented in relation to the specific area. This is followed by the design criteria, the analysis of the cipher, and the results of a cryptanalysis of the cipher. The author propose some possible improvements to fix the cipher against the attacks described in the cryptanalysis. **Paper used:** *A2U2: A stream cipher for printed electronics RFID tags* and *Cryptanalysis of the lightweight cipher A2U2*.

Chapter 5 compares cryptographic primitives and evaluates their merits, weaknesses and suitability for platforms such as low cost passive RFID tags. It extracts the primitives that can be categorized as being lightweight from those that we have evaluated based on the requirements framework presented in Chapter 2. Finally, it presents a metric for evaluating the goodness of a cryptographic primitive as a lightweight primitive based on coupling cost, throughput and power consumption. **Paper used:** *Lightweight Cryptography: Classification and Evaluation.*

Chapter 6 summarizes the work achieved in this thesis, provides concluding remarks and opens to further research perspectives. **Paper used:** *Lightweight Cryptography: Classification and Evaluation.*

1.6 Thesis Delimitation

As its title suggests, this thesis is almost exclusively focused on lightweight cryptographic primitives for passive RFID tags. It covers the various aspects and branches of lightweight cryptography, as presented in the proposed classification (see Section 2.2), and the keys requirements of integration in passive RFID tags (see Section 2.1.4).

However, this thesis does not go into much details regarding the operating process of the RFID technology itself, the RFID standards, nor the different types of active or passive attacks applied to RFID systems. The purpose is to focus principally on the contributions brought by the author to the scientific community, and avoid rewriting on topics widely covered in the literature, and where it would not add any relevant information. Furthermore, the author believes that the readers interested in this thesis already have a background in RFID, and therefore already know its main caracteristics. If necessary, it exists in the published literature a large amount of books and theses that describe all these different aspects of the technology. Among others, [57] presents a complete description of the RFID technology, [30] presents the issues and challenges of RFID cryptography, [128] presents the various type of attacks on RFID systems as well as the popular EPC standard. Additionally, [11] is an exhaustive repository of the existing published work on security and privacy for RFID systems.

Chapter 2

Background and Challenges

> "Lots of people working in cryptography have no deep concern with real application issues. They are trying to discover things clever enough to write papers about."
>
> – *Whitfield Diffie.*

Foreword

This chapter is a selected part of the journal paper *Lightweight Cryptography: Classification and Evaluation*, written in collaboration with Dr. D. Ranasinghe and Dr. Q. Sheng, and submitted in January 2011 to the Journal of Cryptology[1].

2.1 Introduction

In this section we summarize the very low cost computing platforms, more specifically Radio Frequency Identification (RFID) technology, that are set to become ubiquitous over the coming years. Then we discuss the challenges of providing a layer of security to such platforms.

2.1.1 RFID Technology

Radio frequency identification (RFID) regroups all the objects equipped with micro electronics that can process data automatically [33], but in this thesis, we consider exclusively passive low-cost RFID tags having the caracteristcs presented in Table 2.1. A simple illustration of the concept of an RFID system

[1] At the time of delivering this thesis, the paper is still under review.

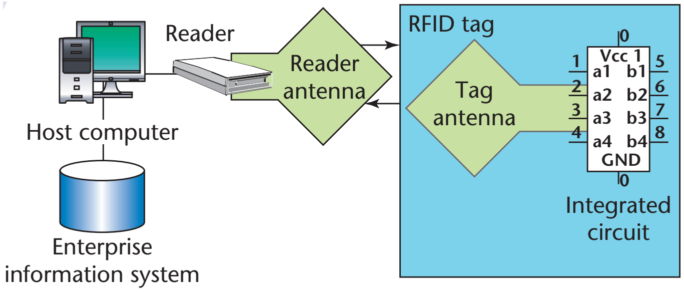


Figure 2.1: Illustration of an RFID system [133].

is provided in Figure 2.1. Here, a transmitter of interrogation signals, which is contained within an interrogator, communicates via electromagnetic waves with an electronically coded label to elicit from the label a reply signal containing useful data, characteristic of the object to which the label is attached. The reply signal is detected by a receiver in the interrogator, and made available to a control system.

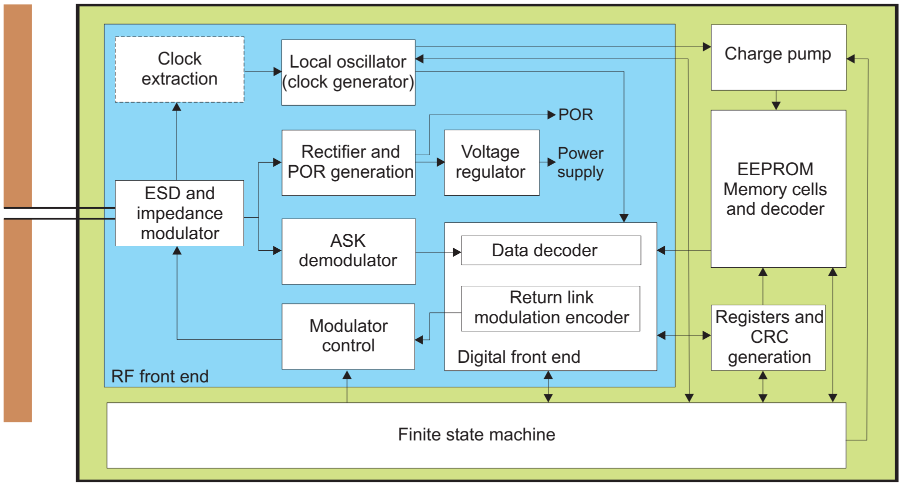

Clock extraction
Local oscillator (clock generator)
Charge pump
POR
Rectifier and POR generation
Voltage regulator
Power supply
EEPROM Memory cells and decoder
ESD and impedance modulator
ASK demodulator
Data decoder
Return link modulation encoder
Modulator control
Digital front end
Registers and CRC generation
RF front end
Finite state machine

Figure 2.2: Block diagram of a passive UHF/HF RFID label [57].

One of the inhibitors to wide-scale adoption of RFID technology is the cost of a label [78]. The primary cost of an RFID label, which includes both an Integrated Circuit (IC) or the silicon chip and the antenna, is the cost of the silicon chip. Low cost RFID refers to an RFID system based on inexpensive RFID tags with the smallest possible implementation of the label IC. Low cost RFID labels are passive transponders since having an on-board battery would add significantly to the cost of the label. The most common operating principle of labels in the category of passive technology is that of RF backscatter or load modulation [30], in which a powering signal or communication carrier supplies power or command signals via an HF or UHF link. However, the circuits within the label operate at the carrier frequency or at a lower frequency. They reply via sidebands generated by modulation, within the label, of a portion of the powering carrier. This approach combines the benefits of relatively good propagation of signals at HF and UHF, and the low power operation of microcircuits at RF or lower. Powering at UHF is employed when a longer interrogation range (several meters) is required. HF powering is employed when electromagnetic fields, which exhibit good material penetration and sharp spatial field confinement, is required or sometimes when a very low cost RFID system implementation is desired [48, 76].

Figure 2.2 is an illustration of a typical low cost transponder [174]. The block diagram of an HF and a UHF chip varies little. In a UHF chip there is a dedicated low power oscillator, while in an HF chip the clock signal is derived from the received carrier by dividing down the carrier in steps. Low cost RFID chips generally have limited memory, typically around 512 bits or less and have no computing hardware except a simple finite state machine for logical functionality [57].

2.1.2 Single Crystal Silicon Integrated Circuit based Tags

Until recently, RFID application specific integrated circuits have been fabricated on single crystal silicon. Although advanced technologies such as 0.13 micron, 0.15 micron or 0.16 micron have been used since 2008, 0.18 micron has been the popular process in widespread usage in low cost RFID manufacture since 2006 [57, 78]. The predominant reason for using older processes has been a strategy for constraining the cost of the RFID IC critical to supporting business cases for RFID enabled applications. Instead, older fabrication processes where the capital cost of the facilities have depreciated over at least 4 to 5 years – currently the 0.18 micron process – proves to be the most cost-effective choice for low cost RFID [78]. It is expected that this situation will eventually change and manufacturer will migrate to 0.15 micron and then eventually to 0.13 micron processes in years to come [40].

In 2005, it was estimated that no more than 2,000 gates were available for security in RFID tags [90]. Moore's Law meant that we are able to deliver more gates per unit area of silicon since then. However, continued end-user demands for cheaper tags to support business cases of novel RFID applications, that will result in mass utilization of RFID tags, imply that the several thousand gates limit is still a reality. Furthermore, newer and smaller feature size fabrication processes are not used for low cost RFID devices [40].

2.1.3 Printed Semiconductor Tags

In recent years, there has been a significant level of interest in printed electronics since it is believed to realize substantially lower cost electronic systems than those available from conventional single crystal integrated circuit (IC) chip based circuit fabrication [77]. Hence, printed electronics are often conceived as a feasible way to solve the high cost problem limiting widespread deployment of RFID tags through dropping the manufacturing cost per tag to the sub one cent level when processes and manufacturing plant is gradually up-scaled to high volume production [77]. Manufacturers of printed RFID are projecting early selling prices of only a few cents in the billions of RFID labels and some foresee sub one cent pricing in much higher volumes [86]. Currently two different approaches are used to develop printed semiconductor tags, namely Silicon Ink Printed Electronics and Organic Printed Electronics [31].

There are a few companies dedicating their R&D and commercialization resources to the application of organic printed RFID tags. One of the leading companies in this area is PolyIC (see *http://www.polyic.com*). In 2007, the company presented the first organic printed RFID tag working at the high frequency range of 13.56MHz with a simple circuit and certainly low functionality [65]. It is only supposed to be used for brand protection and ticketing. Organic printed tags working in the LF band was obtained before the achievement of HF band tags. However, because the LF antenna element size is relatively larger than that of HF antenna, the LF band tags are not applied as widely as the HF band tags. Philips also reported that a 64-bit tag composed of 1940 transistors is obtained based on organic printed electronics. The tag's data rate is 150 bits per second [87]. This figure is much lower than that achievable with printed silicon ink CMOS technology, wherein a tag communication data rate of 106 kbps is believed to be achievable.

There are very few commercial players in the area of silicon ink printed semiconductor, currently exemplified by Kovio Inc. (see *http://www.kovio.com*).

While silicon ink technologies have clear advantages over organic semiconductors, silicon ink technology shares with organic transistor technology the disadvantages of limited transistor numbers. Consequently, there is still an impact and limitation on the complexity of protocols. Somewhere around 2000 transistors are believed to be the comfortable upper boundary zone for printed semiconductor tag circuits today as demanded by yields from printing processes and physical feature sizes [31].

2.1.4 Why is Providing Security a Challenge?

There are several reasons explaining why many of the currently available security primitives are unsuitable for integration into RFID tags:

- The cost of implementing algorithms on hardware is too high to be implemented on a cost constrained RFID Integrated Circuit (IC) where the price of tags, in large volumes, of around 8 US cents [30] is still considered too expensive for mass deployments.
- Relative power consumption by the cryptographic hardware modules is too high for RFID labels that are passive (not self-powered) and typically exceeds that required to operate the tag or read data from a tag's memory [57].
- RFID transponders have limited logic functionality (limited to one or a few state machines) and limited memory (limited to a few kilobits) with no microprocessor for complex operations [57].
- Large key sizes and the resulting ciphertext sizes are generally unsuitable for narrow band communication systems where transmission of significant amount of data directly affects the performance of the system [48].
- The security protocol must be robust against sudden power loss, for example due to the mobility of tagged objects or multipath effects [48].
- Regulatory limitations imply that times in the order of milliseconds are available to complete a secure transaction with a low cost RFID tags [51].

In this thesis, our focus is on mechanisms for addressing the problem of providing secure primitives to extremely resource constrained environments such

Table 2.1: Security related system characteristics comparison.

	Conventional Silicon passive RFID tags	Printed ink passive RFID tags [77]
Gates	2000 available for a security primitive [90]	Less than 200 available for a security primitive
Available memory	Most likely an EPC (Electronic Product Code) of 96 – 256 (see EPCglobal's tag data specification standard [51]) and several hundred bits of user memory. Read-Write memory. Although further steps to reduce costs imply that we are likely to only see Read-Only memory.	Enough bits (96-256) to store a unique identifier. Additional bits may need to be implemented as ROM.
Power consumption	10s of microwatts, and should not exceed that required for EEPROM read operation, so the tag read range requirements can be maintained. Currently EEPROM read operations require around 20-30 μW [108].	Few microwatts.
Performance	In North America it is conceivable to allow a tag to expend around 400,000 clock cycles (based on a 1 MHz internal clock) during a 400 millisecond period (time constraint imposed by FCC regulations for UHF frequencies) for communications. In Europe under revised EN 302 208 regulations it is conceivable to allow a tag a maximum of 4 seconds for communications. Then performance appears to be mainly limited by user requirements and air interface protocols. Bit rates: 40 kbps to 640 kbps (EPCglobal C1G2 protocol) Tag read rates of 200 – 1500 (demanded by end users).	Around 100 kbit/s will not be able to support complex anti-collision techniques and thus may only support the reading of few labels a second.
Read range	3 m – 10 m for UHF and 200 – 500 mm for HF operation under FCC regulations [57, 48].	Much reduced read ranges. Currently UHF tags are not possible.
Communication Protocols used	The most prevalent standard for UHF tags is the CIG2 protocol [51]. The multi-part ISO 18000 air interface standard defines protocols for a number of different frequencies; LF, HF and UHF. ISO 18000 Part 3 Mode 1 is possibly the most prevalent standard as of yet. The most commonly used HF standard, other than the ISO 18000, is ISO 14443 (types A and B).	No standardized protocols suitable for printed ink tags. New protocols based on the keeping their implementation to around 1200 transistors leaving around 800 transistors or 200 gates for security are needed.
Tag IC footprint	Currently 18,000 – 30,000 gates for a Class I Generation 2 air interface protocol (C1G2) implementation [51]. However these are expected to be simplified in the future to reduce the cost of tag ICs.	Around 2000 transistors (500 gates)
Available Resources	32 bit random number generator (as required by the C1G2 protocol)	None.

as passive RFID tags. These technologies provide extremely resource scarce platforms (as outlined in Table 2.1) on which to implement security primitives. In the case of low-cost RFID tags, the challenges presented are threefold:

- The cost (area of silicon used or the gate count) has to be minimal in order to ensure that the security solution will be used in low cost tags. A low cost tag demand a very limited silicon area footprint from conventional tags (typically around 2000 gates) and a more severe limitation from printed ink RFID tags (typically around 200- 300 gates is considered as threshold).
- Power consumption of the security implementation has to be reduced to its minimum since passive devices do not have an on-board battery and rely on an external electromagnetic field to supply them the required energy [57, 48].
- Performance (throughput) should be reasonable to support application and end-user requirements (read rates of over 200 tags per second) to as well as to allow their use in real protocol such as EPCglobal Class 1 Generation 2 or ISO 18000- 6C [51].

One of the goals of this thesis is to develop a classification scheme and a definition for lightweight cryptography. It is done in such a way that it is inclusive of various schools of thought pursued by researchers, in order to achieve solutions to meet the challenges of providing security to the extremely resource constrained devices.

2.2 Classification

Early candidates for resource constrained devices are based on a simplification or hardware optimization (for minimum area) of well-known block ciphers: Feldhofer proposed an optimized version of AES [55] while Leander did a similar work with the DES [105]. XTEA [125] and SEA [158] are two other block ciphers suitable for embedded devices. However, none of these ciphers were designed with RFID applications as a precise target and turned out to be either slow (AES, SEA, TEA) or have a high cost (AES, DESL, TEA) as presented in Table 5.1. The first block cipher intentionally designed for resource constrained devices, PRESENT [20], is the result of the work carried out in an

EU Project called UbiSec&Sens [163]. PRESENT is based on key ideas articulated by both Shannon and Rueppel, and widely used in the design of modern stream ciphers. Inspired by the techniques used in DES and AES, PRESENT's design was derived in two different flavors: a low cost implementation (1000 gates) and a high throughput implementation (200 Kbps) [144]. A later block cipher, KATAN [44], has reached an extra step with a design tailor-made for low-cost RFID tags. A rigorous analysis of power consumption (< 1μW), area minimization (down to 480 gates) and throughput optimization (12.5 kbps) has been achieved in its design.

However, the number of mechanisms that are suitable for extremely resource limited devices is unfortunately much fewer. For instance, none of the stream ciphers resulting from the EU project NESSIE ([126]) met the requirements of passive RFID systems, leading to a new EU project called eSTREAM ([52]), to address this gap. Among the selected candidates of eSTREAM, there are only two of particular interest for low cost RFID applications; GRAIN and TRIVIUM [53, 66].

Other designers have tried to exploit characteristics that are unique to these resource limited platforms to develop less orthodox security mechanism. For example, the physical variation of signaling [161, 63] or the difference in the strength of the bi-directional link between the tag and the reader [74]. Finally, an interesting alternative to block and stream ciphers was proposed by Yüksel with a scalable universal hash function for RFID tags called WH-16 [174], at a very low cost (460 gates).

Nevertheless, the term *Lightweight Cryptography* is widely used to describe many optimizations of existing primitives as well as new designs. These often fail to meet the requirements of the target platforms described in Section 2.1, albeit a few (as demonstrated further in this thesis). Moreover, new developments in lightweight cryptography cannot be adequately and appropriately reflected in the broad field of cryptography, exemplified by recent developments such as physical one-way functions [161, 63] or distance implies distrust [58]. Furthermore, the term lightweight cryptography lacks an adequate articulation of its meaning in the literature.

From previous work such as [50], lightweight cryptography can be considered as a fusion of separate disciplines in cryptography, information technology, radio frequency engineering, and microelectronics. It can be considered as a novel direction in cryptography that aims to develop fast and efficient security mechanisms for extremely resource constrained environments such as passive RFID tags. In building primitives suitable for limited resource environments such as low cost RFID the designers not only have to consider the security strength of its algorithm, but also its computational complexity, its power consumption and its hardware integration size.

Lightweight Cryptography is the collection of cryptographic primitives, techniques and ciphers that can be implemented in highly resource-constrained mobile devices such as passive RFID tags. Such devices harvest energy for all their functions, communicates over band limited channels and every gate used for security is considered an additional cost that must be carefully utilized. In the lightweight context, a designer has to analyze the computational complexity of the algorithm, with respect to the demands on the hardware and other limitations of the device. There are both a direction and a constraining challenge in these limitations that guide the development of cryptography.

In a scheme that is built on traditional cryptography, lightweight cryptography can be classified as shown in Figure 2.3.

A distinct omission from the classification scheme is asymmetric key primitives because public key ciphers, based on the factorization of the dicrete log problems, are unsuitable for implementation in resource constrained devices because their implementation cost is too high [50]. The only public key cipher that could be considered suitable for RFID tags is NTRU [81], based on the closest vector problem [160]. It requires moderate resources and is considered to be a much faster algorithm compared to other public key ciphers such as RSA [142]. The most recent low cost implementation of NTRU presented in [9] uses 2,884 gates for encryption on a 0.13 μm process with an architecture that consumes 1.78 μW of power when clocked at 500 kHz. Although the cipher is considered to be faster than other public key ciphers, the implementation presented in [9] still requires 28,223 clock cycles to generate 1,169 ciphertext bits for a 256 bit plaintext message. Consequently, researchers have not focused on lightweight primitives based on asymmetric key ciphers.

The following sections illustrate the proposed classification using primitives published in the literature (briefly mentioned in this section) as being suit-

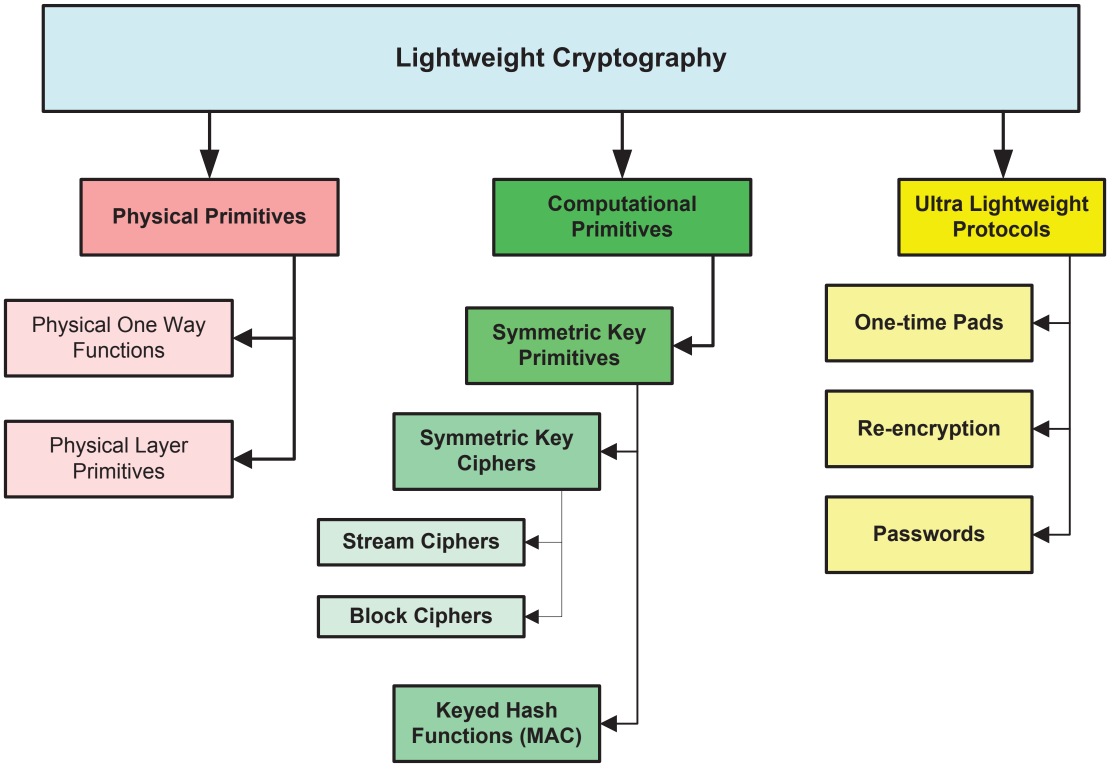


Figure 2.3: Classification of Lightweight Cryptography.

able for resource constrained devices. Furthermore, we present a brief analysis along the dimensions of (i) cost of implementation in hardware, (ii) performance in terms of throughput, (iii) power consumption and, (iv) the level of security each primitive is capable of providing, where such information is publicly available.

2.3 Physical Primitives

The general idea of physical primitives mainly lies in the use of biometrics for authentication. However, in RFID a physical primitive is a function that is inseparably integrated to the hardware platform and the physical layer of the device. Therefore, a physical primitive can measure an analogue phenomenon or variations in physical characteristics, which is inherent to physical systems but prohibitively difficult to duplicate, and convert it to a digital value for the purpose of precise quantification.

Such a function was first published in [64]. The Physical One-Way Functions

(POWF) described in [64] are based on accurately measuring scattering patterns of visible laser radiation, resulting from the 3D microstructures of a transparent optical medium, which is incorporated into the physical system. The output is dependent on the frequency, the angle of the laser beam entering the optical medium, and the optical characteristics of the medium.

POWF provide a means to assign a unique, tamper-resistant, and unclonable identifier to everyday objects at a relatively very low implementation cost. However, the cost of embedding optical structures onto electronic transponders and the added cost of scatter pattern measuring instruments on RFID readers implies that POWF are not a suitable low cost solution for RFID.

In addition to optical systems, there are several other physical systems on which POWF can be based. The main types are coating POWF [64], acoustic POWF [64] and silicon POWF. The primitives derived in silicon are the most relevant to the author's investigation.

2.3.1 Physical One-Way Functions

The ability to construct a POWF on silicon was outlined in [64, 114, 62]. These POWFs, referred to as Integrable Physically Unclonable Functions (IPUF), map a set of challenge inputs to a set of responses, utilizing some physical characteristic of integrated circuits on silicon. There are also various IPUF, most often referred to simply as PUF (Physical Unclonable Function), such as:

- architectures termed Arbiter PUFs [161],
- XOR Arbiter PUFs [161],
- Lightweight Secure PUFs [118],
- Feed Forward Arbiter PUFs [63],
- Ring Oscillator PUFs [161].

The main concept behind PUF designs is to use process variations in wires and transistors on an IC to obtain a characteristic response from each IC, when given a certain input. Manufacturers always attempt to control process variations to a great degree. However, these variations are largely beyond their control and hence it is not possible for an adversary to fabricate identical PUF circuits. The PUF circuit is able to uniquely characterize each IC due to manufacturing variations. Thus, it may be possible to identify and authenticate each IC reliably by observing the PUF response.

The particular advantage in this technique is that secret keys no longer need to be securely stored in memory. Therefore, physical attacks such as microprobing, laser cutting and reverse engineering techniques [167], which are used to reconstruct the layout of circuits to enable adversaries to extract digital keys stored in the memory of integrated circuits are no longer effective. While various tamper-proofing methods – such as the tamper sensing technology in [101] – have been developed over the years to counter such physical attacks, they are a costly solution, especially for low cost RFID applications. Therefore, PUFs provide a powerful primitive upon which to build security mechanisms.

2.3.1.1 Physically Unclonable Function Implementations

There exist a number of structures for building a PUF [82, 116, 117]. The most notable ones are the latch based structures and the ring oscillator based structures. This thesis focuses on the latch based structures because they offer architectures with the lowest power consumption as well as the lowest silicon area.

The PUF based structures are sensitive to noise, especially thermal noise, as wire latencies and gate delays depend on the operating temperature of the device [64]. This leads to reliability issues when trying to obtain consistent responses for a given input. Unreliability due to such environmental variations have been addressed in a PUF configuration given in [114], wherein a challenge response pair is created using an PUF circuit based on a differential topology. It is such a differential configuration that is considered in this thesis.

Figure 2.4 depicts the structure of a PUF circuit, which is based on the arbiter-based PUF in [114, 139]. The circuit accepts a n bit challenge b_0, b_1, b_2, $\ldots$, b_n to form two delay paths in 2^n different configurations. In order to generate a response bit, two delay paths are excited simultaneously to allow the transitions to race against each other. The arbiter block, at the end of the delay paths, determines which rising edge arrives first and sets its output to 0 or 1. The actual implementation of arbiter-based PUFs in [114, 139] uses 64 bit challenges.

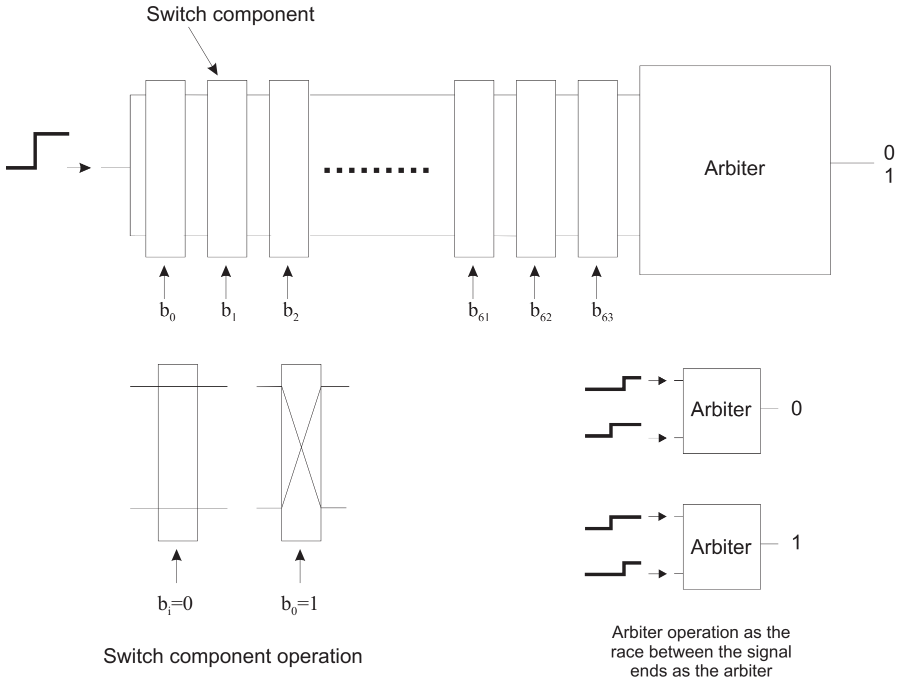

Switch component
Arbiter
0
1
b0
b1
b2
b61
b62
b63
bi=0
b0=1
Switch component operation
Arbiter
0
Arbiter
1
Arbiter operation as the race between the signal ends as the arbiter

Figure 2.4: Arbiter-based PUF circuit implementation [139].

The switch component is implemented using a pair of two-to-one multiplexers. Depending on the select bit C_i, the switch either allows the signal to travel straight through or swap the delay paths. The arbiter is constructed using a simple transparent latch with an active-low enable input.

It was estimated in [114] that there is a strong enough variation between two chips fabricated from the same silicon wafer for a sufficient number of random challenges to identify billions of chips. The probability that the first measured response bits to a given challenge (set of bits) on a chip is different from the measured response for the same set of bits (challenge) on a different chip is estimated to be 23% to 40% depending on the PUF circuit architecture [114]. It has been estimated that about 800 challenge response pairs are sufficient to distinguish 10^9 chips with an error probability $p_e < 10^{-10}$ [114].

Measurements of noise in PUF circuits have shown a bit error rate comprised between 8% and 25% as a result of challenges not producing predictable response in repeated measurements. This effect is due to various environmental factors such as fluctuation in operating voltage and temperature [172]. There are two possible approaches to overcome the issues related to unreliability in the context of RFID:

1. Using a threshold level for matching bits. For instance if 75 bits of a 100 bit response are validated, the tag is evaluated to be authentic.
2. Using helper bits to correct the errors in the PUF responses.

Although correcting PUF responses can be employed at the RFID reader, correcting responses is unnecessary for a simple authentication mechanism. Such a strategy will require the storage of error correcting information (for instance with the ability to correct 25 bits out of 100 bits), called a syndrome, based on coding schemes such as BCH (Bose-Chaudhuri-Hochquenghem) or index-based syndrome (IBS) coding [172]. Instead, it can be shown that, by selecting an appropriate threshold level, associated false negative can be minimized. It has been reported that using 128 challenges of 64 bits each to generate a response of 128 bits is adequate to achieve a false-positive or a false-negative probability of a few parts per billion [46].

2.3.1.2 Authentication: Direct Use of PUF

Simplest authentication mechanisms use PUFs responses directly in a challenge-response protocol based authentication mechanism (as illustrated in Figure 2.5) instead of using a PUF as a mechanism to obtain a secret key. In such a scheme, a trusted party such as a product manufacturer securely stores a set of challenge response pars (CRPs) from an RFID tagged object when the object is certain to be authentic. It is these responses that are compared with those obtained from an RFID tag to establish its authenticity at a later time. To defend against man-in-the-middle attacks, CRPs are never reused.

2.3.1.3 Authentication: PUF and Linear Feedback Shift Register (LFSR) Combination Design

The primary performance obstacle in the tag authentication protocol presented above is the excessive overhead of transmitting a large number of challenges. Based on using a 64 stage PUF circuit, each challenge consists of 64 bits. Thus to obtain a PUF response of 128 bits the reader must transmit 8,192 bits using a narrow-band communication channel.

An alternative method published in [139] is to transmit only one challenge C, where the challenge is used to initialize a linear feedback shift register. Then

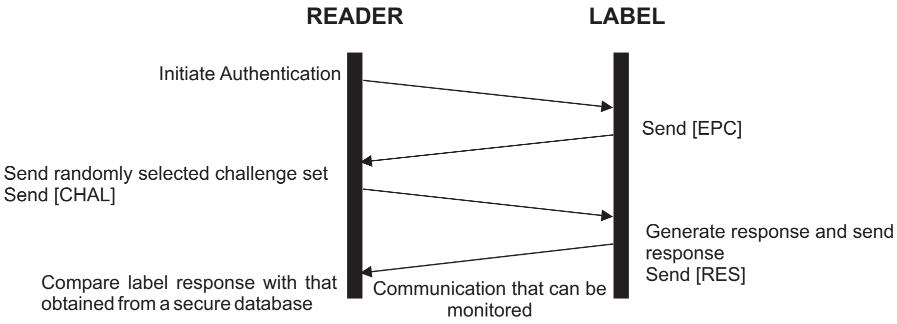

READER
LABEL
Initiate Authentication
Send [EPC]
Send randomly selected challenge set
Send [CHAL]
Generate response and send response
Send [RES]
Compare label response with that obtained from a secure database
Communication that can be monitored

Figure 2.5: Message exchange between a reader and an RFID label during an authentication process [139].

the following challenges can be generated on-tag to extract a response from the PUF circuit. This arrangement is illustrated in Figure 2.6. The additional hardware of an LFSR allows the challenge response protocol to be executed with greater efficiency.

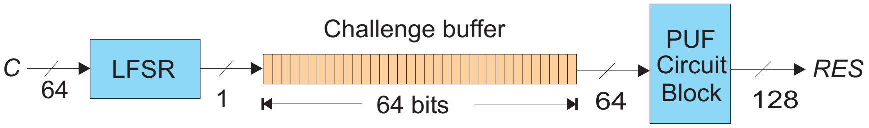


Figure 2.6: PUF-based authentication engine design to reduce overhead [139].

2.3.1.4 Evaluation

The performance of the two security schemes outlined above are presented in Table 2.2. The need to transmit a large number of challenges from the reader to the tag remains the primary obstacle with using PUFs directly, especially given that the maximum possible transmission speed from a reader to a tag in the EPC Class 1 Generation 2 (C1G2) specification is about 126 kbps given equiprobable ones and zeros [51]. As a consequence of this overhead, on average, only 15 tags can be authenticated per second.

Table 2.2: Evaluation of PUF implementations [139].

	PUF (using 128 CRPs)	**PUF (using a LFSR to generate 128 CRPs)**
PUF area (gates)	856	1720
Effective Throughput (kbps)	2.048	15.49

The schemes using a LFSR (see Figure 2.6) overcomes the overhead of transmitting large sets of challenges. Unlike the authentication schemes outlined previously, where a large number of challenges need to be sent to the tag, the current transmission requirement is that of a single challenge C, from which other challenges are derived. Using a smaller challenge set as well as the incorporation of an LFSR has allowed the lightweight primitive to be implemented in a low cost RFID tag while improving its performance.

The following are challenges and security issues related to PUF that need to be considered despite the attractiveness of the physical cryptographic technique:

- The arbiter favors the path to output zero since it is preset to zero and requires a setup time constraint to switch to a logic one. Fixing a small number of most significant challenge bits can compensate for this skew by effectively lengthening one delay path. The circuit layout must be designed carefully to ensure that both paths are symmetrical and arbiter responses are not biased to 0 or 1.
- Recently, model building attacks performed on software-based implementations of PUF instances have shown the use of machine learning techniques to generate accurate-enough delay-based models of PUF circuits [147]. The models developed can, with accuracies of over 95%, predict the response of PUF circuits. More significantly, the complexity of these attacks is linear or log-linear with respect to the parameters such as the number of stages in the PUF circuit [147]. However, the attack methods have not yet been verified using actual hardware implementations of PUF circuits.

It should be noted that addressing modeling attacks may require the use of more stages in the PUF circuits and/or using more challenges. Both approaches increase the cost of implementation and further reduce performance of the approach.

2.3.2 Physical Layer Primitives

The Distance Implies Distrust scheme [58] is a prime example of developing a simple lightweight cryptographic primitive based on the physical layer of RFID systems. The mechanism is based on the assumption that an unauthorized reader attempting to read a tag is generally more physically distant from the tags than a legitimate reader. The latter assumption is based on the realization that a closer and more visible reader draws greater investigation by tag owners or tag bearers. Thus, the measurement of distance of a reader to a tag based on link layer measurements is proposed as a measure of trust [74]. This is a simple implementation that should be considered for applications based on proximity cards where the reading distances are indeed small to prevent malicious scanning attacks on tags or mafia fraud type attacks.

2.4 Ultra-Lightweight Protocols

The term *ultra-lightweight* is used to refer to security systems that only employ simple logic operation such as exclusive-or (XOR) for its implementation. These types of security mechanisms have blossomed along with the maturity and the increasing adoption of RFID technology. Often ultra-lightweight cryptographic techniques are referred to as “XOR Cryptography” to highlight its simplicity, which is demanded by the target platform.

The formulation of mechanisms to achieve security objectives under the constraints presented by low cost RFID systems to real-world tags using a weak, but perhaps realistic, security model can form a central part of the security proof for ultra-lightweight cryptography. These mechanisms generally rely on a number of practical assumptions (the attacker has no computational facilities, the attacker can read the tag only a small number of times, etc.) to demonstrate the practical security of mechanisms. More broadly, all ultra-lightweight primitives can be classified into those based on: (i) one-time pads, (ii) re-encryption and, (iii) passwords.

2.4.1 One-Time Pads

Szewczykowski [162] demonstrates a very simplistic approach which relies on a simple one-time pad concept to prevent counterfeiting of bank notes. The scheme involves the recording of a random number, a date-time stamp, on an RFID label of a bank note when it is released. The bank note keeps a track of the number of times it has been scanned and this number is used as part of its authentication process. When a bank note is read by a bank teller, the date-time stamp and the number of scans are sent to a central computer to verify the authenticity of the note based on comparing the same information securely stored on the computer. This scheme is vulnerable to cloning by physical attacks and the possibility of desynchronization of the back-end systems with that of the data on RFID tags is a real concern [88, 12].

A different application of one-time pads can be found in [29]. A set of random numbers (authentication codes) is stored into the label along with a label ID prior to its release. A copy of the authentication codes and label IDs are securely stored in a back-end database. The reader sends a tag specific authentication code. If a match occurs, the label responds with a return authentication code known exclusively to the database and increments a counter to select a set of new random numbers for the next procedure. A successful authentication results in the synchronization of the database with that of tag. In the event of an unauthorized reader, the label will not respond unless the reader knows the next random number expected by the label. In case of a counterfeit label, the interrogator fails to find a match to the tag response and thus detects the counterfeit. Nevertheless, this scheme still leaves the possibility of a physical attack where the contents of the label may be discovered. In the worst case, this information cannot be used to counterfeit labels in massive quantities as the set of authentication keys and authentication responses are all different and completely random on each individual label. This mechanism is also vulnerable to mafia fraud type attacks and resynchronization attacks.

2.4.2 One-Time Pads: Pseudonyms

A mechanism proposed in [140] uses a list of randomly generated tag identifiers on a tag. On querying a tag, a reader is able to hash the response and access tag-related data on a secure hash table. A similar version was also published in [87] with accompanying protocols for low-cost tags. The mutual authentication protocol (both tag and reader) is based on a list of pseudonyms and keys residing on tags and on a back-end server. The protocol only needs additional memory on tag and updates the tag's pseudonym list using one-time pads to resist eavesdropping. However, the communication cost is relatively high because of the tag data updates. The use of pseudonyms in [87] is based on the assumption that the intruder only comes into the scanning range of a tag on a periodic basis, as a complete analysis of the limited number of pseudonyms will allow the identification of the tag. The security model is based on the underlying assumption that the tags release their data at a limited rate [87]. The minimalist security model sets an upper limit (of half-a-dozen) on the number of times an intruder or an adversary can scan a given tag or try to spoof a valid reader.

2.4.3 Re-encryption

In [88], an unorthodox re-encryption mechanism is proposed for providing security protection to banknotes embedded with RFID labels. In a traditional setting, the entity conducting the re-encryption will not be aware of the plaintext. In the re-encryption scheme discussed in [88], the plaintext is known to the entity performing the re-encryption. The security of the mechanism is based on the ciphertext created by encrypting the digital signature stored on the RFID chip by a central bank authority, the serial number of the banknote and a random number. The authenticity of the banknote can be verified by comparing the ciphertext stored on the banknote to the ciphertext obtained by encrypting the digital signature, the serial number, and the random number using a public key. A match indicates an authentic banknote. Consequently this scheme can be performed off-line.

The significant security achievements claimed include forgery resistance, fraud detection, and tamper resistance. The primary weakness is that a banknote is still in possession of all the inputs required to create a copy. The digital signature is not verified during a transaction; hence, the fake banknotes can be created with ciphertext obtained from a collection of believable serial numbers. Other shortcomings that might be exploited by a resourceful adversary are provided in [88]. However, it is an innovative approach based on shifting the encryption engines and secret keys away from the RFID label to more secure locations, such as the readers and the central bank authority.

2.4.4 Passwords: Exploiting the KILL Password

Current generation of EPC Class-1 Generation-2 tags (ISO 18000-6C) all specify the use of passwords to protect the KILL functionality of tags (permanent disablement of an RFID tag) [51]. Furthermore, Juels [90] proposed a low-cost authentication mechanism, where the read-protected 32-bit kill passwords of tags are used to implement an ad-hoc tag authentication protocol. The central idea is based on the fact that even though the EPC of a transponder can be skimmed, the kill-password remains secret. Cloned tags can be found by testing if the kill password matches the original one stored in a database, without killing the tag. Furthermore, the protocol supports mutual authentication.

2.5 Computational Primitives

All modern cryptographic primitives based on mathematical techniques to provide secrecy (authenticity, integrity, confidentiality, availability and non-repudiation), where the security of the system relies on keeping the key secret, are computational primitives. Most modern cryptographic primitives based on the hardness of a mathematical problem are examples of computational primitives. However, lightweight primitives are a subset of these, since not all of them are suitable for resource constrained environments. In particular, public key cryptographic primitives are excluded since any primitive considered secure (e.g. RSA, ECC, NTRU [81]) have been proven difficult to be optimized for resource limited devices [50] (see Section 2.2). Thus, computational primitives are those lightweight cryptographic primitives based on mathematical techniques that achieve Shannon's ideas of confusion and diffusion [155] using simple mathematical and logical operations.

2.5.1 Keyed Hash Functions (MAC)

Usually, the characteristics of hash functions and public-key cryptography (large keys, complex and energy consuming computations) restrict their use in severely resource constrained devices [22]. However, it is worth mentioning the few primitives that have been developed, taking into consideration the resource scarcity of devices as a valid option for applications requiring strong authentication, based on a challenge-response protocol.

Universal hash functions can be thought of as collections of hash functions that map plaintext into short output strings, such that the collision probability of any given pair of messages is very small [70]. The following steps describe the use of a universal hash function to build a Message Authentication Code (MAC):

1. The communicating parties share a secret and a randomly chosen hash function from the universal hash-function family,
2. They also share a secret encryption key,
3. A MAC is generated by hashing a message with the shared secret hash function and then encrypting the resulting hash using the secret key.

A matter of distinction with universal hash functions is that their level of security is provable. Consequently, a universal hash-function family can be used to build an unconditionally secure MAC, in theory. Unfortunately, to achieve unconditional security the secret encryption key used to select the hash function and to encrypt the resulting hash must be a one-time pad. This is rarely possible in practice and a pseudo-random number generator (PRNG) inevitably needs to be used. As a result of using a PRNG the security of the hash schemes is defined by the security of the PRNG generator.

For the primitive to achieve provable security, a cryptographically secure PRNG is needed to generate new keying material. Standardized and trusted PRNGs are in turn again based on block or stream ciphers or hash functions. This implies that for RFID security mechanisms using PRNGs on top of primitives like hash functions, some substantial additional circuit is required. A recent PRNG proposed in [120] can be implemented in as few as 266 gates.

2.5.1.1 WH Hash Function

A secure hash scheme implementation is certainly beyond the limitation of the 2,000 gates available for security on a low cost RFID IC, as the primary target of hash functions is computer security [91]. Interestingly, Yüksel [174] presented implementations of low-cost universal hash functions, taking only 460 gates for block size of 64 bits with a reported power consumption of 2.95 µW when clocked at 500 kHz. His work presents three variations of a universal hash function, namely PH, PR and WH. The construction we have examined is that of WH-16 (16 bit implementation of the WH hash). The time complexity against forgery achieved with a 64-bit MAC (using a 128 bit key) is approximately 2^{64}.

Associated drawback and security issues related to the WH hash function are as follows:

- A key recovery attack based on recovering partial keys of size w (where w is the word length of the hash function implementation) requires 2^w plaintext and MAC queries as well as a similar number of MAC verification queries to a legitimate reader to find the partial key used in the MAC [75]. The probability of forgery is approximately 2^{-w} and a partial key recovery attack has time complexity 2^w.

- The above attack is a serious concern for the minimal hardware implementation of WH-16 but this is overcome with the use of increased key bits and increased hash rounds.
- If the key of the universal hash function is reused, the fact that weak keys are easy to recover can have dramatic implications, as key recovery allows for arbitrary forgeries [75].
- A substantial keying material is required. For instance, using a 16-bit architecture to generate a 64-bit hash requires 512 key bits, and a further number for encrypting the hash result to achieve a partial key recovery attack complexity of 2^{64}. As a consequence, the performance of the WH-16 hash function is poor for moderately secure applications.

2.5.1.2 SQUASH

SQUASH (which is short for SQUare-hASH) is a recent universal hash function, with an implementation based on the non-linear feedback shift register (NLFSR) of Grain-128, proposed by Shamir [154]. The basic idea of SQUASH is to mimic the operation of the Rabin encryption scheme. The hash function is based on replacing the computationally intensive modular squaring operation $m^2 (\bmod n)$ by a randomized squaring operation $m^2 + r.n$ where r is a random number, m is the plaintext and n is the key. SQUASH-128 uses a 64-bit key and a 64-bit challenge to produce a 32-bit hash value.

Significance of this new scheme can be summarized as follows:

- It is exceptionally simple, and yet it is provably, at least as secure as the Rabin scheme (which has been extensively studied over the last 30 years) [154].
- Shamir [154] reports that the best attack on SQUASH requires exponential time and grows monotonically with the size of n. Therefore breaking an extremely reduced version of SQUASH which uses $n = 2^{128}-1$ as the universal modulus is still extremely difficult even though it is very easy to factor 128.

Given that the scheme is based on 512 update rounds, SQUASH is a low throughput hash function. Two implementations of SQUASH have been reported and are presented in Table 2.3.

Table 2.3: Evaluation of SQUASH implementations clocked at 100 kHz.

	Output (bits)	Area (gates)	Throughput (kbps)	Power Consumption (μW)	Clock cycles per block
SQUASH-64 [67]	32	6,303	0.05	-	63,250
Optimized SQUASH-64 on 0.13μm UMC [175]	32	2,646	0.1	0.0357	31,800

The later implementation of SQUASH [175] has an implementation cost less than half of the first implementation proposed by Gosset [67]. With 2,646 gates, SQUASH is slightly above the requirements of RFID tags. However, given its promises of strong security, SQUASH remains a worthy candidate to evaluate in this thesis. Additionally, an alternative implementation of SQUASH using static permutations on the NLFSR sequence output is presented in [175]. This implementation requires only 1,918 gates, which is below the requirement threshold of RFID tags. However, since this implementation presents some (small) design changes, it is to be considered with precautions until a cryptanalysis of this implementation has been achieved. The main drawback of SQUASH remains its low throughput, which is a serious concern regarding RFID standard requirements.

2.5.1.3 Additional Hash Functions

The recent interest to the different branches of lightweight cryptography has also seen the development of several new hash functions in the past months. It is probably utopic to list all of them but it is worth mentioning three of them, which meet the requirements of passive RFID tags: QUARK [10], SPONGENT [19], and the PHOTON family [69]. The implementations details for these three hash functions (in their smallest implementation variant) are given in Table 2.4.

The three hash functions presented in Table 2.4 have a small footprint (lower than 1,400 GE) and a low power consumption (lower than 2.5 μm). Their only drawback is their throughput, which is low compared with other ciphers presented in this chapter. Due to their recent publication, these hash function did not yet benefited from public scrutiny to challenge their security. Therefore, they are not integrated in the comparison in Chapter 5.

Table 2.4: Evaluation of QUARK, SPONGENT, and PHOTON implementations clocked at 100 kHz.

	Output (bits)	Area (gates)	Throughput (kbps)	Power Consumption (μW)	Clock cycles per block
PHOTON-80 on 1.8V, 0.18μm UMC [69]	64	865	1.51	1.59	3,540
U-QUARK on 1.8V, 0.18μm UMC [10]	128	1,379	1.47	2.44	8,704
SPONGENT on 0.13μm UMC [19]	88	738	0.81	1.57	15,840

2.5.2 Symmetric Key Primitives: Block Ciphers

This category of cipher is the most common in cryptography. There are more than 100 block ciphers [11]. Those closely matching the platform specific requirements outlined in Table 2.1 with a hardware-oriented design are explored in this section.

Block ciphers are symmetric key ciphers operating on fixed-size blocks of bits. There are three common aspects to block ciphers:

- key mixing,
- permutations,
- substitutions.

The output of a block cipher is a ciphertext block with length similar to the block size.

2.5.2.1 Tiny Encryption Algorithm

Tiny Encryption Algorithm (TEA) is a Feistel cipher designed for simplicity and ease of implementation in 1994 [169]. TEA uses only XOR, ADD and SHIFT operations to provide Shannon's properties of diffusion and confusion necessary for a secure block cipher without the need for P-boxes and S-boxes. The encryption algorithm is based on a large number of iterations to gain security without compromising simplicity. A description of the algorithm is provided in [169].

TEA is a 64-bit block cipher and uses a 128-bit key based on a suggested 64 Feistel rounds, typically implemented in pairs termed cycles (32 cycles). There are three different architectures of the TEA algorithm, namely:

- a parallel design where two Feistel rounds are implemented as one cycle,
- a serial design,
- an 8-bit architecture design.

These implementations are summarized in Table 2.5 for a 0.35µm CMOS process [84].

TEA suffers from a few weaknesses due to the simplicity of the key schedule mechanism. A full thesis on the crytpanalysis of TEA has been proposed by Andem [8]. Its main conclusions are the following:

- TEA is insecure from equivalent keys where each key is equivalent to three others.
- TEA is vulnerable to related key attacks.
- The best known attack (a related key attack) requires 2^{23} chosen plaintexts under a related-key pair, with 2^{32} time complexity for a 64 Feistel round TEA cipher.

The above weaknesses have been demonstrated by TEA's most famed vulnerability exploited in the Xbox, where it was used directly as a hash function [159]. However, it was never intended to be used as hash function, and [56] clearly indicated that TEA must never be used as a hash, because it is insecure if used this way. This critical flaw gives direct access to the flash memory of the Xbox [159]. Therefore TEA should not be employed for applications requiring moderate levels of security beyond a few weeks or months.

TEA has undergone several iterations since its inception to address its weaknesses (equivalent keys and related-key attacks [8]). The most relevant new cipher for RFID applications is the XTEA (Extended TEA) described by Wheeler

[2] Note that silicon area has been converted to gate counts using 62.5×10^{-6} mm^2 per gate based on a 0.35μm CMOS process [85].

Table 2.5: Evaluation of TEA implementations [85].

	Parallel	**Sequential**	**8-bit**
Area (gates) [2]	3312	1984	2032
Clock cycles per 2 Feistel rounds	1	9	8
Power Consumption (in μW)	7.37	39.0	38.4

in 1997[3] [125]. It was proposed to fix the two minor weaknesses of TEA and remains an appropriate choice for low cost RFID application. In particular due to the following reasons:

- An examination of XTEA in [8] suggests that XTEA is highly resistant to differential cryptanalysis as a result of the level of diffusion achieved by the cipher.
- This research found the encryption of cipher texts with very few rounds (less than six) to be weak.
- Encryption of cipher texts with more than six rounds produced a very good mixture of intermediate values and showed high resistance to cryptanalytic attacks.
- The best attack reported on XTEA is a related-key differential attack on 26 out of 64 rounds of XTEA, requiring$2^{20.5}$ chosen plaintexts and a time complexity of $2^{115.15}$ [97].

With the few exceptions with regards to related key cryptanalysis attacks, TEA remains a suitable candidate for some RFID applications. Where a higher level of security is required XTEA remains an option. However, at the time of writing there appeared to be no hardware implementation of XTEA available for consideration.

2.5.2.2 Scalable Encryption Algorithm

Scalable Encryption Algorithm (SEA) is a parametric block cipher suitable for small embedded applications [158]. The two most important parameters being *the key* size and *the number of block cipher rounds*.

[3] A further version of the cipher, XXTEA, has been proposed by the authors of XTEA in 1998. The cipher has been broken by Yarrkov in 2010 with a differential cryptanalysis requiring 259 queries and negligible work [171].

The key factors defining SEA are the following:

- The simplicity of the cipher based on simple XOR operations and 3-bit substitution boxes.
- It is based on a variable number of block cipher rounds, which can be user defined. Thus the cipher may operate as a simple low security but fast block cipher or as one with a higher level of security [158].
- SEA does claim an advantage over AES and other ciphers when both encryption and decryption needs to be implemented in hardware. But this is only significant in high throughput SEA architectures (e.g., SEA architectures designed to run at 80 MHz with throughput of over 5 Mbps) [158].

A recent adaptation of SEA in [115] by optimizing the hardware for minimum silicon area to develop a small co-processor core shows significant potential. The minimum data path (using an 8 bit data bus) implementation published in [115] consumes approximately 3.2 µW and requires 50 clock cycles to complete a single bock cipher round. The estimated hardware for the co-processor is 917 gates but an additional 192 bits of RAM storage as well as eight 8-bit storage registers are also required to store intermediate states [115]. SEA has benefited from the opportunity to be subjected to a range of cryptanalysis attacks over the years similarly to TEA.

A guide to estimate the number of recommended rounds from a thorough analysis of the weaknesses of the cipher by its designers is given in [158]. Using the recommended number of rounds roughly corresponds to the number of rounds to resist linear and differential attacks as well as meeting the requirement for having twice the number of rounds to obtain complete diffusion (to prevent both structural attacks and outer rounds improvements of statistical attacks). The only significant attack discussed as possibly being tractable in polynomial time is an algebraic attack [158]. At the time of writing the author is not aware of such a weakness related to the SEA cipher.

2.5.2.3 mCRYPTON

The acronym stands for miniature Crypton and can be thought of as a 64-bit variant of Crypton [113] (one of the 15 candidates considered for AES in 1998, and revised in 1999 [111]) with variable key sizes based on substitution and permutation operations published in 2006 [112]. Consequently it is a 64-bit block cipher that can use 64, 96 or 128 bit keys. A hardware implementation published in [112] requires about 3,500 to 4,100 gates for both encryption and decryption, and about 2,400 to 3,000 gates for encryption alone using 0.13µm CMOS technology.

mCRYPTON is a highly secure cipher for the following reasons:

- Generally, the complexity of differential and linear cryptanalysis is completely determined by the number of active S-boxes involved and their linear approximation probabilities. mCRYPTON has a linear approximation probability upper bound of 2^{-128} for an 8-round block cipher. This completely eliminates any advantage that can be gained from commonly known linear and differential cryptanalysis techniques [112].
- The cryptographers have not considered a variety of other cryptanalysis techniques for the security of mCRYPTON (algebraic attacks, related key attacks and key schedule cryptanalysis). However, the mCRYPTON cipher, similar to the Crypton cipher, has not yet been threatened by these attacks for implementations that use more than 12 rounds, which is the case for Crypton [111].
- A recent attack using related-key impossible differential cryptanalysis performed on a 9-round reduced version of mCrypton-96 requires $2^{59.9}$ chosen plaintexts and $2^{63.9}$ bytes of memory, and has the time complexity of about $2^{74.9}$ encryptions. The attack on mCrypton-128 has data, time and memory complexities of $2^{59.7}$ chosen plaintexts, $2^{66.7}$ encryptions and $2^{55.7}$ bytes of memory, respectively [119].

The version of mCRYPTON considered in this thesis is based on using 13 rounds to compute a single block of ciphertext. The implementation presented is based on a parallel architecture and the authors in [112] expect a more compact implementation with a greater degree of serialization to deliver a further size reduction of around 30% at the expense of using 5 cycles per round architecture. Such a reduction would allow mCRYPTON to be implemented with around 1,680 gates.

2.5.2.4 The KATAN family

Based on their previous experience with Trivium (discussed later in Section 2.5.3.2), the authors of this cipher proposed KATAN, a family of small hardware-oriented block ciphers, in 2009 [44]. Their main goal was to design a robust cipher with a minimal implementation size, high throughput and low power consumption. Their family is composed of two slightly different design principles: KATAN and KTANTAN, where the difference resides in the way tags store their secret key. KATAN has a variable key while KTANTAN has its key stored on a read-only memory. They propose 3 versions of both cipher types: with 32, 48 and 64 bits block sizes. We have selected KTANTAN-32 for our analysis since it is most suitable for low cost RFID.

Table 2.6: Evaluation of KATAN variants [44].

	KATAN-32	**KTANTAN-32**
Area (gates)	802	464
Throughput (kbps)	12.5	12.5
Power Consumption (µW) [4]	0.381	0.146

To create the cipher text, an initial value (IV) is entered into 2 LFSRs (Linear Feed Back Shift Registers) of different sizes (depending on the cipher version), which are clocked for 256 rounds. The feedback function of one LFSR feeds the other LFSR and reciprocally. A non-linear parameter is introduced via a 3$^{\mathrm{rd}}$ LFSR that acts as a counter of the number of rounds. A more detailed description of the cipher is given in [44]. The Table 2.6 summarized the results obtained with the KATAN cipher family (implementation on a 0.13µm CMOS process and clocked at 100 kHz).

The results provided by the KATAN family are impressive compared with other ciphers. The area used is the second lowest of all presented ciphers in this section. Similarly, the power consumption of KTANTAN-32 is close to only 1/10 of the lowest power consumption among the other ciphers, but comparing power consumption between ciphers implemented on different process is not an easy task and these results must be considered with precaution.

[4] The power consumption results of KATAN and KTANTAN do not take into account the leakage power, which is a major factor in the 0.13μm CMOS process.

Because of its recent publication, the KATAN cipher family did not benefit from an extensive public scrutiny to challenge its security claims. However, the authors provide a self-performed security analysis as a basis for their cipher. The main conclusions for the KATAN-32 are the following [44]:

- The best differential characteristic has probability 2^{-33} for 126 rounds.
- The best linear approximation has a bias of 2^{-16} for 126 rounds.
- The best boomerang attack has probability 2^{-44} for 128 rounds.
- There is no slide property with probability greater than 2^{-32} from the 1st round (slide attack).
- The probability of a related-key attack is at most 2^{-32}.
- The cipher is expected to be secure against algebraic attack [44].
- The best side-channel attack has 2^{51} time and $2^{23.8}$ data complexity for the full 254 rounds KATAN32 [14].

Some further cryptanalysis have been published regarding the fixed-key version of the cipher, KTANTAN, with the following results:

- A meet-in-the-middle attack can be performed with time complexity $2^{72.9}$, $2^{73.8}$, and $2^{74.4}$ for the 32, 48, and 64 bits versions respectively [166].
- A related-key attack can break the cipher with time complexity $2^{28.44}$, $2^{31.77}$, and $2^{32.28}$ for the 32, 48, and 64 bits versions respectively [3].

The authors have encouraged researchers to challenge KATAN ciphers' robustness. Promises regarding area and power consumption offered by the KATAN family are attractive for large scale implementations with low-cost RFID tags.

2.5.2.5 Hardware Optimized Versions of Data Encryption Standard

Our discussion is based around the following versions of Data Encryption Standard (DES):

- Hardware optimized DES.

- Lightweight version of DES with redesigned S-boxes called DESL.
- DES supporting a larger key called DESX.
- Lightweight version of DESX called DESXL.

DESL (DES Lightweight) [105] is a compact version of DES based on using a single S-box instead of the 8 S-boxes used by DES . The authors report that DESL's S-boxes have been designed to resist both linear and differential cryptanalysis attacks as well as the Davies-Murphy attack. The DESL proposal is significant in that it uses less resources (49% less chip, 85% less clock cycles and 90% less energy) than the hardware optimized AES implementation presented by Feldhofer in 2004 [54] (discussed in Section 5.1.1.1) while providing improved security over DES.

Table 2.7: Evaluation of DES variants [105].

	Hardware Optimized DES	**DESL**	**DESX**	**DESXL**
Area (gates)	2309	1848	2629	2168
Clock cycles required to encrypt 64 bit block of plaintext	144	144	144	144
Power consumption (µW)	2.14	1.6	-	-

More significantly they have also addressed the issue of the smaller key size (56 bits) used by DES. This issue allowed brute force attacks to be completed within a few months using several hundred PCs (FPGA based parallel machine COPACOBANA can break DES in 9 days at a hardware cost of $12,000 [103]). DESX [105] is based on extending DES with a key whitening step to be able to use larger keys of 184 bits. This is an optional step that can be added to the hardware optimized version of DES if increased security is required. Results of hardware optimized implementations of DES variants from [105] are summarized in Table 2.7 (implementations on a 1.8V, 0.18 µm CMOS process and clocked at 100 kHz).

DESX is estimated to increase the time required to break the cipher using linear cryptanalysis to a period of more than 80 years based on DESX operating at 500 KHz. Its new S-box design has met eight different conditions which fulfill some requirements to be resistant against classical linear cryptanalysis and differential cryptanalysis as well as the Davies- Murphy attack.

They have shown that the differential cryptanalysis attacks published by Biham and Shamir against DES [16] are no longer feasible with DESL. Furthermore, DESL is more resistant against linear cryptanalysis than DES due to the improved S-box design.

However, there is no independent evidence published to evaluate DESL and DESXL to determine their vulnerability as a result of the changes made to their design.

2.5.2.6 PRESENT

PRESENT is a recent block cipher (first published in 2007) [20]. The cipher is a result of work carried out in an EU funded research project called UbiSec&Sens. PRESENT is a block cipher based on 32 rounds, a key size of 80 or 128 bits, and a ciphertext block size of 64 bits.

Table 2.8: Evaluation of PRESENT implementations clocked at 100 kHz.

	Area (gates)	**Throughput (kbps)**	**Power Consumption (μW)**	**Clock cycles per block**
Low power PRESENT-80 (serialized architecture) on 1.8V, 0.18μm UMC [144]	1075	11.4	2.52	563
PRESENT-80 (round architecture) on 1.8V, 0.18μm UMC [144]	1650	200	3.86	32
Low power PRESENT-80 (serialized architecture) on 3.3V, 0.35μm AMI [144]	1000	11.4	11.20	563
PRESENT-80 (round architecture) on 3.3V, 0.35μm AMI [144]	1525	200	33.40	32

PRESENT is based on key ideas articulated by both Shannon [155] and Rueppel [145], and widely used in the design of modern stream ciphers. PRESENT in particular is based on techniques used in DES and AES. PRESENT relies upon the encryption techniques of confusion and diffusion similarly to AES and DES:

- Confusion is accomplished through substitution where specially chosen sections of data are substituted for corresponding sections from the original data.
- Diffusion is accomplished through permutation. The data is permuted by rearranging the order of the various sections.

Both substitutions and permutations are based upon the key and the original plaintext. PRESENT, like any other SPN (Substitution-Permutation Network), comprises of three stages: a key-mixing step, a substitution layer, and a permutation layer.

Area optimized versions of PRESENT has been documented as capable of being implemented on 1,000 gates based on a 64 bit key. However, the power consumption of this architecture is reported to be 11.2µW and it takes 563 clock cycles to generate 64 bits of ciphertext when clocked at 100 kHz [144].

From Table 2.8 it is clear that the technology used for implementing PRESENT has a serious impact on the power consumption. Therefore, using a cheaper 0.35µm technology may not be an option for implementing the cipher on passive RFID technology using older CMOS processes.

While the inventors of the cipher have ruled out some of the most powerful cryptanalysis techniques based on attacks such as differential and algebraic attacks [21], there is limited independent proof on the robustness of the cipher given its recent appearance.

There are currently four known attacks against PRESENT but none of them are enough to threaten the full 32-round PRESENT cipher. In addition, these attacks are only tractable in the time frame of 10s of years based on moderate computing resources and a PRESENT implementation clocked using a 10 MHz clock.

- The first attack is a differential cryptanalysis attack that can recover the secret key, for a PRESENT cipher based on up to 16 rounds, using 2^{64} chosen plaintexts and 2^{65} memory accesses [165].

- The second attack is a differential attack using algebraic techniques that can recover an 80-bit key up to 16 rounds with similar complexity to [165] and a 128-bit key up to 19 rounds by 2^{113} computations [5].

- The third attack is a statistical saturation attack that is applicable up to 24 rounds using 2^{57} chosen plaintexts. This attack has a time complexity of 2^{57} under the condition that parts of the plaintext can be fixed to a constant value by an adversary [32].

- The fourth is a linear cryptanalysis attack reported for a 23-round PRESENT using around $2^{59.3}$ known plaintexts and time complexity of $2^{78.5}$. Indeed, this attack has violated the proof of impossibility of a LC attack by the designer as a consequence of the attack methodology, multidimensional linear cryptanalysis, employed in [28].

Given its robustness and its lightweight implementation of only 1,000 gates, PRESENT is currently one of the best solutions to provide security for low-cost RFID systems. Its main weakness thus far is probably its lack of public scrutiny; as such cryptanalysis remains the best security proof of a cipher.

2.5.2.7 PRINTcipher

Based on their experience in the field [20, 21, 105], Knudsen et al. proposed a new block cipher, PRINTcipher [96]. Its claimed purpose is to be small enough to be implemented in printed electronics device. The starting point of its design is the well known cipher PRESENT [20], which they customized using the properties offered by printed electronics (bits permutations), and building blocks of other ciphers: 3-bits S-boxes [158] and other features from key-dependent algorithms [152, 150, 68]. PRINTcipher comes in two different flavor: (i) PRINTcipher-48 with 48-bit blocks and a 80-bits key, and (ii) PRINTcipher-96 with 96-bit blocks and a 160-bits key. The implementation results of the several versions of PRINTcipher are presented on Table 2.9.

Table 2.9: Evaluation of PRINTcipher implementations clocked at 100 kHz.

	Area (gates)	**Throughput (kbps)**	**Power Consumption**[5] **(μW)**	**Clock cycles per block**
PRINTcipher-48 (serialized architecture) on 1.8V, 0.18μm UMC [96]	402	6.25	< 2.60	768
PRINTcipher-48 (round architecture) on 1.8V, 0.18μm UMC [96]	503	100	< 2.60	48
PRINTcipher-96 (serialized architecture) on 1.8V, 0.18μm UMC [96]	726	3.125	< 2.60	3072
PRINTcipher-96 (round architecture) on 1.8V, 0.18μm UMC [96]	967	100	< 2.60	96

[5] No detail is given regarding the power consumption for the different implementations. It is only mentioned that all implementations require less than 2.6 μW [96].

Like most of the previously presented block ciphers, PRINTcipher implementations are always a trade-off between area and throughput. However, with only 402 GE in its smaller implementation, PRINTcipher has the smallest footprint of all the presented ciphers thus far. Despite its recent publication and due to its remarquable parameters, PRINTcipher already “benefited” from public scrutiny to challenge its security claims. The main conclusions of the several cryptanalysis of the PRINTcipher are the following:

- A linear cryptanalysis has time complexity 2^{72} for a 28-rounds reduced version of PRINTcipher [4].
- A combined differential and linear cryptanalysis can be performed on a 29 or 31-rounds reduced version of PRINTcipher for respectively 4.54% and 0.036% of the keys [92].
- An invariant subspace attack can break PRINTcipher-48 for 2^{52} keys (out of 2^{80}) and PRINTcipher-96 for 2^{102} keys (out of 2^{160}) [104].

Yet, no attack succeed to fully break the 48 rounds of the cipher (or for any key), but the cipher is still recent and needs some additional time and public scrutiny to prove its robustness, if ever.

2.5.2.8 Additional Block Ciphers

Similarly to the hash functions, a few recent block ciphers are worth mentioning in this thesis. In particular, two of them present suitable characteristics for implementation in low cost RFID tags: LBlock [170], and Piccolo [156]. The block ciphers characteristics (in their smallest implementation variant) are presented in Table 2.10.

Table 2.10: Evaluation of LBlock and Piccolo implementations clocked at 100 kHz.

	Block Size	**Area**	**Throughput**	**Power**
				Consumption
	(bits)	**(gates)**	**(kbps)**	**(μW)**
LBlock on 0.18μm UMC [67]	64	1,320	200	-
Piccolo-80 on 0.13μm UMC [175]	64	683	14.81	-

The two block ciphers presented in Table 2.10 do not provide power consumption evaluation, but given the the small amount of gates and compared to similar block ciphers [144], it is likely that their power consumption is lower than

5 μW. Therefore, they could be suitable for low-cost RFID applications. However, due to their recent publication, their security has not yet been challenged. For this reason, these two ciphers are not compared with the other ciphers in Chapter 5.

2.5.3 Symmetric Key Primitives: Stream Ciphers

Stream ciphers are less common than block ciphers, mainly because they were considered weaker for a long period [153]. The low hardware cost offered by stream ciphers has reinvigorated interest in the last decade with an aim to secure resource constrained device applications. Stream ciphers are also symmetric ciphers, but contrary to block ciphers, they operate on 1 bit at a time. The cipher produces a keystream which is subsequently XORed with the plaintext to produce the ciphertext. Similar to the selection of block ciphers, the stream ciphers presented in this thesise are limited to those meeting the requirements outlined in Table 2.1. The notable difference here is that the set of suitable stream ciphers is much smaller.

An outcome at the conclusion of the EU funded project eSTREAM, organized by the ECRYPT network (European Network of Excellence for Cryptology) in late 2008 is the development and evaluation of a number of stream ciphers suitable for efficient hardware implementation in resource constrained environments. The eSTREAM project was set up as the result of the failure of all six stream ciphers proposed by the NESSIE project [126]. The surviving candidates at the end of the eSTREAM project are summarized below (for the complete list of candidates, see [49]).

- F-FCSR-H v2
- Grain version 1
- MICKEY
- Trivium

From the four finalists, both MICKEY-128 and F-FCSR-Hv2 have reported minimum implementation area and power consumption given below for a 0.13μm CMOS process.

- MICKEY-1283 [66] – 5,039 gates and 11.2µW at 100 kHz
- F-FCSR-H v2 (8-bit word size) [66] – 4,760 gates and 10.6µW at 100 kHz

Due to their significant implementation cost, compared to the best known hardware optimized implementation cost of AES, these two ciphers are not be considered any further. Designers of these four ciphers have applied well-established principles and methodologies (since perhaps 1883) for building ciphers. While these are relatively new designs and available free without patent protection, a rigorous analysis of the ciphers have not yet taken place.

2.5.3.1 Grain version 1.0

The stream cipher is based on two non-linear Boolean functions for eliminating weaknesses in the use of linear feedback shift register operations. Similarly to the shrinking generator [34], a non linear filter function selects and combines the output from two shift registers running in parallel. Grain is a synchronous stream cipher. Grain accepts an 80-bit key and a 64-bit initial value (IV).

In the key initialization phase, the goal is to scramble the contents of the shift registers before the running key is generated. The number of clock cycles is a trade-off between security and speed. Table 2.11 presents a comparison of two different implementations of Grain version 1.0: i) the first, optimized for minimum area and, ii) the second, optimized for minimum power.

Table 2.11: Evaluation of GRAIN implementations clocked at 100 kHz.

	Area (gates)	**Throughput (kbps)**	**Power consumption (µW)**	**Latency[6] (clock cycles)**
Minimum area GRAIN-80 on 1.2V, 0.13µm CMOS [66]	1294	100	3.3	321
Minimum power GRAIN-80 on 1.5V, 0.35µm CMOS [53]	3360	123	1.2	130

The low power design appears to be comparable with an AES implementation in [72] using 3,400 gates but AES is based on a 128-bit key. However, strong security of the AES implementation is achieved at greater power consumption (around 4 times more) [53]. It can be seen from Table 2.11 that latency is an

[6] Cycles taken for initialization and loading key and IV.

issue with Grain. If Grain is to be reinitialized often with a new IV, then its long initialization phase (321 clock cycles for a minimum area implementation) is a bottleneck. This is indeed the case with most protocols employed with RFID tags.

The following cryptanalysis results on Grain have been reported:

- The original Grain Version 0.0 cipher was broken by a key recovery attack [15] which required 2^{43} computations and 2^{38} keystream bits to determine the 80-bit key.
- An attack based discovering related keys and initial values of Grain 1.0 are described in [102]. This method improves a brute-force attack by allowing an adversary to test keys twice as fast as usual, yielding a worst case brute force attack complexity of 2^{79} trials. Although the method does not result in an efficient key recovery attack, it indicates a weakness in the initialization process (achieved by feeding back the keystream generated over 160 rounds).
- In 2008, [18] describes breaking Grain using TMD (time/memory/data) trade-offs to recover the state of Grain version 1.0 with time and memory complexity 2^{71} after a pre-computation phase with a complexity of $2^{106.5}$ steps and using $2^{53.5}$ bits of known keystream. Clearly the precomputation steps imply that only a very resourceful adversary is able to use this method to improve upon an attack based on an exhaustive search.
- A algebraic attack using a weak IVs can recover the key of Grain-128 with 100 keystream bits and $2^{93.8}$ operations [71].
- A dynamic cube attack on the full version of Grain-128 can recover the full key but only when it belongs to a large subset of 2^{-10} of the possible keys. This attack is faster than exhaustive search over the 2^{118} possible keys by a factor of about 2^{15}[47].

2.5.3.2 Trivium

Trivium is also a synchronous stream cipher, first published in 2006 [43]. Trivium requires an 80-bit key and an 80-bit initial value to initialize the state registers of the cipher, which then enables the generation of a keystream of up to

2^{64} bits. The basic building blocks of Trivium are NLFSRs combined with 1,152 repeated update cycles in the initialization phase of the cipher. This high overhead cost of initializing the cipher and an implementation cost of 2,599 GE are significant drawbacks for RFID applications. Table 2.12 provides a comparison of Trivium implementations for low cost and low power.

Table 2.12: Evaluation of Trivium implementations clocked at 100 kHz.

	Area (gates)	**Throughput (kbps)**	**Power consumption (μW)**	**Latency[7] (clock cycles)**
Minimum area TRIVIUM-80 on 1.2V, 0.13μm CMOS [66]	2599	100	5.6	1333
Minimum power TRIVIUM-80 on 1.5V, 0.35μm CMOS [53]	3090	72	1.02	1603

Similar to Grain the security of the cipher is not well documented due to the relatively recent publication of its details. At present, there are three significant results detailing its weaknesses published in [164], [121] and [135], summarized as the following:

- Trivium has been reported to be vulnerable to an algebraic attack called the cube attack reported in [164]. The attack requires 2^{30} steps to break a Trivium implementation using 735 initialization rounds.

- Raddum in [135] outlines a new algorithm for solving nonlinear systems of equations generated to represent the internal state of the cipher. The attack complexity found was 2^{164} for the full version of Trivium. This complexity is far higher than an exhaustive search.

- Attack described in [121] is capable of recovering the internal state (and thus the key) of the full cipher in around $2^{83.5}$ steps (where each step is roughly the cost of a single trial in exhaustive search) given that an attacker is able to obtain $2^{61.5}$ bits from the keystream (a key recovery attack).

- A significant result presented in [121] is that the security of the cipher cannot be increased by increasing the size of the key as the proposed attack method will become increasingly better than an exhaustive search.

[7] Cycles taken for initialization and loading key and IV.

> The only option then is to increase the complexity of the internal state and thus the implementation cost, performance and power consumption.

The authors of Trivium have reported the cipher to be capable of thwarting correlation attacks where detecting a correlation is estimated to require at least 2^{144} bits of keystream [43]. This is well above the security requirement given that Trivium generates 2^{64} bits of keystream once initialized. A significant component of Trivium (which is also its drawback) is the extensive key run-up phase. Although the mixing that takes place during the former phase, before any keystream is produced, essentially ensures that any system of equation generated to model the full cipher cannot be easily solved as a result of the large number of variables, the latency of 1,152 update rounds is a barrier to performance.

2.6 Conclusion

A definition and a presentation of the various techniques of lightweight cryptography was proposed in this chapter. Section 2.5 is of particular interest for the rest of this thesis, since the following two chapters describe some cryptographic primitives that belong to these categories. In this chapter, the most recent and significant (for low-cost RFID implementation) ciphers have been introduced. Since it is a fast growing field of research, this chapter does not include all the published ciphers but the ones that present a significant impact on this research field.

The lightweight primitives presented in this chapter are further compared and discussed in Chapter 5, together with the cipher presented in Chapter 4.

Chapter 3

Ultra-Lightweight Primitive for passive RFID Tags

"Research is what I'm doing when I don't know what I'm doing."
– *Werner von Braun.*

Foreword

The first five sections of this chapter is a selected part of the conference paper *Providing strong Security and high Privacy in low-cost RFID networks* [41], written in collaboration with Dr. N. Prasad, and presented during the *First International ICST Conference on Security and Privacy in Mobile Information and Communication Systems*, in Turin, Italy, in June 2009.

3.1 Introduction

As described in Chapter 2, ultra-lightweight cryptography is the part of cryptography using the most basic computational elements to provide security protocols. Yet, combining these elements to achieve a complex and robust cryptosystem is a challenge that still remains to be solved. In this chapter, we propose an ultra-lightweight protocol inspired by the work conducted in previous research [27, 94]. The aim was to address the issues of previous works, while requiring less computational effort from the tag at the same time.

3.1.1 Components of networking security

Among the different aspects of RFID security, protecting RFID systems from hackers, spies, thieves, and other unauthorized entities is a challenge, which is based on three major components, common to any kind of network security. Those three components are represented in Figure 3.1.

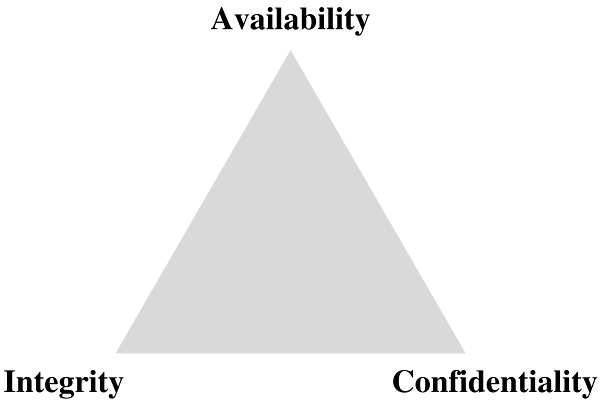


Figure 3.1: The three pillars of networking security.

Availability: It assures that the system is running at the required performance and scalability level. The most common threat to availability is the denial-of-service (DoS) attack.

Integrity: It ensures the accuracy and authenticity of information transmitted by the system, by preventing its accidental or malicious modification. Spoofing (or Man-in-the-middle) attack is a common threat to integrity.

Confidentiality: It aims at limiting information access to authorized personnel only. Consumer privacy issues fits into the confidentiality metric and are described more in details in the following paragraph.

3.1.2 Privacy threats

With the development of the RFID technology, most of the security concerns turn around the privacy issue: what, when, where and how much amount of consumer data is recorded without permission or even notification? Privacy has become one of the most sensitive topics, since it has to deal with not only the privacy and integrity of the company, but also with the privacy of the consumer, who is the end-user in the production chain. These privacy threats become a reality as the deployment of RFID tags on everyday products is propagating, but even more because the RFID readers are small, relatively inexpensive, and have sufficient processing capabilities to read most of the tags. Solving those threats is one of the most challenging issue regarding this technology, both for the research perspective and the social aspect it introduces. So far, seven serious privacy threats have been identified, compromising the privacy of the consumer [61]:

- The *action threat* concerns the behavior or intent of a user that can be inferred from the evolution of the group of tags surrounding him.
- The *association threat* focuses on the product itself. Not only the kind of products the person owns, but the precise product can be discovered (very limited edition, for example).
- The *location threat* deals with tracking of people thanks to the tags they are wearing. Since most of the readers are fixed, it can be quite easy to monitor someone's location through the whole day, by checking all the places where some of the tags he is wearing, have been read.
- The *preference threat* is related to the specific kind of product someone owns and buys, to define his consumer profile, and thus target him more specifically (target advertising, for example).
- The *constellation threat* is highly related to the location threat except that it is not a targeted individual that is tracked but a random individual without knowing its identity. The tracking of this person would be performed by tracking the constellation of RFID tags that he is wearing.
- In the *transaction threat*, the tracking of goods does not stop at the consumer step, but goes further and keep on tracking the location and ownership of the tagged object through its entire product life (until the chip is destroyed). If some product goes from one person to another, you can conclude that they know each other and draw a link between them. Step by step, you can draw a complete social network, connecting all the links between people.
- The *breadcrumb threat* is the issue that links someone to the objects he bought as long as the objects exist (i.e. the tag is working). When someone buys a product in a retail shop, the tag information is stored in the

shop database (or even a larger database), and is not updated after the consumer's purchase. It can then create some trouble to the owner in case of a misuse by a third person.

We can infer from all those threats that the consumer privacy can be highly exposed if no action is done to avoid them. They are described in more details in [61]. Many researchers have been working on this topic and their contribution to this field is detailed in the next section.

3.2 Related work

While facing an issue, two options are available. One consists in finding the solutions to solve the issue. In this particular case, the solution is to implement encryption and authentication in the process. Some research has been done to compare all the available encryption algorithms for ultra-low power devices [91]. However, this work is more focused on Wireless Sensor Networks, which embedded more computational resources than RFID Tags. A similar survey is presented in [50] and goes even more deeply in the energy consumption of each single computational operation. Many different ways have been explored to ensure security in RFID technology. Some works focus on the physical proper-ties of the device to maintain security, using the physical imperfections of the hardware to guarantee authenticity [23]. Several security algorithms have been proposed as well, some relatively energy-consuming introducing "lightweight" elliptic curve cryptography primitives [95] or trapdoor-based mutual authentication [107]. Others are using considerable memory resources either through a key-table [176] or storing additional data to preserve untraceability [173]. Some propose a secured solution limited to a single reader scenario [176, 26], which could be an unrealistic constraint in real case deployment. Finally, a few other protocols perform strong security and authenticity with simple bit-wise operations [94, 27]. These two last mentioned protocols seem to be the best alternative for low-cost RFID solutions, since they are not as energy-consuming and memory-demanding as the others, and do not present some major constraints or security gaps.

The other option consists in removing the issue itself. Authentication and encryption is a strong requirement in a wireless communication process to maintain privacy. However, this requirement becomes useless without communication. In fact, all the privacy threats are based on a simple assumption: tags and readers are able to communicate. As communications stop, the threats disappear (except the breadcrumb threat which is related to the physical product itself). So, the concept is to avoid communication between tags and reader. In this way, a possible solution was the use of a battery-powered mobile device called "RFID Guardian", supposed to create interference around the guardian to preserve the privacy of the RFID Guardian holder [141]. The same principle has been studied by the RFID expert A. Juels et al. who propose a "blocker tag" that selectively allow communications with authenticated readers [89]. Those solutions present the disadvantage to constrain the user to hold an additional device permanently with him, and open the path to security gaps in case the device fails. The ultimate alternative is to temporarily deactivate the tag to make sure it is not able to communicate anymore, without purpose.

3.3 Proposed Solution

3.3.1 A few assumptions

In the production cycle, the privacy concerns appear at the very last link of the chain, when the consumer buys the product. The concept of the proposed solution presented in Figure 3.2, is to ensure a strong security during the whole production process, and a strong privacy, once the consumer owns the product. The security protocol is inspired by the works done in [94] and [27] combining a strong authenticity and reducing the computational load of the tag, without compromising the security. While the tag is created, four numbers are shared by the tag and the server; two Pseudo ID (P_{ID} & P_{ID_2}, initially equals) and two keys, K_1 and K_2. All of them have a length equal to the tag ID. In this protocol, we assume that the link between the Reader and the Server, which are two powerful devices, is secured against all the traditional attacks related to networks. We also assume that the reader will always be able to communicate with the server, making the communication available everywhere, anytime.

3.4 The Protocol

- *1st step:* The reader requests a certificate of authenticity to the server. The server decides whether or not the reader is allowed to retrieve further information from the server, by sending back a certificate. This first step is done only once per reader, per day. If the reader wants to read multiple

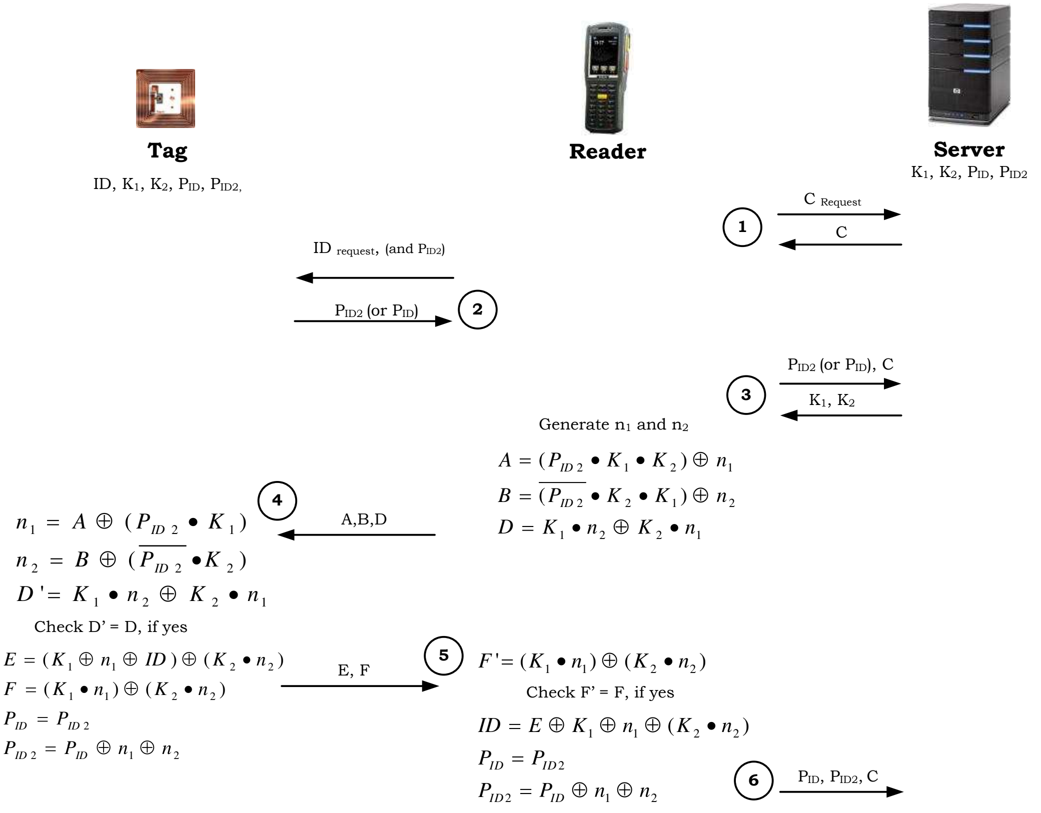

Tag
ID, K1, K2, PID, PID2,
Reader
Server
K1, K2, PID, PID2
1
C Request
C
ID request, (and PID2)
2
PID2 (or PID)
3
PID2 (or PID), C
K1, K2
Generate n1 and n2
$A = (P_{ID2} \bullet K_1 \bullet K_2) \oplus n_1$
$B = (\overline{P_{ID2}} \bullet K_2 \bullet K_1) \oplus n_2$
$D = K_1 \bullet n_2 \oplus K_2 \bullet n_1$
4
A,B,D
$n_1 = A \oplus (P_{ID2} \bullet K_1)$
$n_2 = B \oplus (\overline{P_{ID2}} \bullet K_2)$
$D' = K_1 \bullet n_2 \oplus K_2 \bullet n_1$
Check D' = D, if yes
$E = (K_1 \oplus n_1 \oplus ID) \oplus (K_2 \bullet n_2)$
$F = (K_1 \bullet n_1) \oplus (K_2 \bullet n_2)$
$P_{ID} = P_{ID2}$
$P_{ID2} = P_{ID} \oplus n_1 \oplus n_2$
E, F
5
$F' = (K_1 \bullet n_1) \oplus (K_2 \bullet n_2)$
Check F' = F, if yes
$ID = E \oplus K_1 \oplus n_1 \oplus (K_2 \bullet n_2)$
$P_{ID} = P_{ID2}$
$P_{ID2} = P_{ID} \oplus n_1 \oplus n_2$
6
PID, PID2, C

Figure 3.2: Security Protocol [41].

tags at the same time, it will just request a certificate once, and use it for all the tags. If the reader already has a valid certificate (the one of the day) the protocol starts in Step 2.

- *2nd step:* The reader sends a request to the tag, which replies with its P_{ID_2}. If the reader cannot find a match in the database, it will send another request with the P_{ID_2} to the tag, which will reply with its P_{ID}.

- *3rd step:* The reader sends the P_{ID} together with its certificate. If the certificate is authentic, the server replies with K_1 and K_2. The reader then generates 2 random numbers n_1 and n_2. It computes A, B and D with the keys shared by the tag and the random numbers.

- *4th step:* The tag computes the values of n_1, n_2 and D'. If K_1 and K_2 are genuine, D' and D will be equal. This step is necessary to authenticate the reader from the tag's point of view. Since K_1 and K_2 are not exchanged between tag and reader, a match in the D value means that the reader is legitimate.

- *5th step:* The tag computes E with its ID, F with the 4 secret values, and sends them to the reader. The reader will first check the value of F to check the authenticity of the tag. If the values match, it will be able to retrieve the ID from E using K_1, K_2, n_1 and n_2. Both reader and tag compute a new value of Pseudo ID (P_{ID_2}) that will be used for the next communication.

- *6th step:* The reader sends an updated version of the Pseudo ID (P_{ID_2}) as well as the previous version (P_{ID}) and its certificate, to maintain authenticity. Thanks to n_1 and n_2 the values A, B, D, E and F are always different. Changing the value of P_{ID} is a necessity to avoid tracking of the tag. In fact, it's rather easy to track a tag that would always reply with the same message to a simple request. Storing the previous value of the P_{ID}, while the next value to be used is P_{ID_2}, is to maintain the integrity of the network and avoid de-synchronization attacks.

3.5 Security and Privacy Evaluation

In this section, we review a little more in detail the security threat and the solutions proposed by the algorithm.

Eavesdropping happens when an attacker listens to the channel to retrieve information. Even if the attacker will receive the messages, it is not possible to determine the values of the secrets keys or the ID of the tag, since the messages are encrypted (A, B, D, E and F).

Unauthorized tag reading and **tag cloning** are solved through the use of authentication. If the reader is not allowed by the server it will never get any information related to the tag. Similarly, if the tag is not genuine, it will not be able to decrypt the values of A, B and D.

Tracking is done by simply listening to the data transmitted by the tag. Since the data sent by the tag is always different, P_{ID_2} is changing in a random way since it involves n_1 and n_2, it is not possible to track the tags over time. Though, between two successful authentications, a malicious reader will always receive either P_{ID} or P_{ID_2} from its requests and will therefore be able to track the tag. However, it assumes that the malicious reader is following the tag, which does not make sense in a realistic scenario (i.e. if the reader follows the tag, you do not need to read it in order to track it).

Replay attack occurs when a malicious tag tries to authenticate itself by repeating the authentication sequence (P_{ID}) of a genuine tag. While the reader will reply with A, B and D, the malicious tag won't be able to retrieve any information from those values.

De-synchronization attack is used by an attacker to update the values in only one part of the network, either the tag or the reader, in order to make it impossible for them to communicate further. In our scheme, before any update of data, there is a check (D and F) and if the values do not match, the intrusion is detected and the value of P_{ID_2} is not updated in the server. If it is updated by mistake, the old value (P_{ID}) will be used to recover from the attack.

Forward Security is the possibility to maintain integrity of the communication over time. It means that even if the tag is physically compromised one day, and the attacker is able to recover the secret values of the tag, it will not be able to find the previous data, since every exchange of data includes two random numbers.

Disclosure attack is used to retrieve some secret information from one entity by sending a slightly modified message to see the impact on the answer. In our scheme, any change is detected and the attacker won't receive any answer.

Regarding the privacy issues described in Section 3.1.2, at the exception of the exception of the location threat that is equivalent to the tracking security issue, all the other issues are directly linked to the secrecy of data exchanged. If the data transmitted through this authentication protocol happen to be unsecure, then all the privacy issues become potential threats.

The Table 3.1[1] is a comparison of security threats in different ultra-lightweight protocols implemented for low-cost RFID tags. Our solution is slightly lighter

[1] The table has been updated with the cryptanalysis results published in [79] since the published version of 2009.

Table 3.1: Proposed comparison of ultra-lightweight authentication protocols.

	LMAP [132]	M^{2}AP [130]	EMAP [129]	SASI [27]	*David-Prasad [41]*
Resistance to De-synchronization attacks	No	No	No	No	No
Resistance to Disclosure attacks	No	No	No	No	No
Resistance to Tracking	No	No	No	No	No
Privacy and Anonymity	No	No	No	No	No
Forward Secrecy	No	No	No	No	No
Mutual Authentication	No	No	No	No	No
Cryptanalysis	[110]	[7, 110]	[109, 7]	[38, 80, 13]	[79]
Memory size on Tag	6L[2]	6L	6L	7L	6L
Memory size on Server	6L	6L	6L	4L	5L
Operation types on Tag	$\oplus, \bullet, +, 2^m$	$\oplus, \bullet, +2^m$	$\oplus, \bullet, +$	$\oplus, \bullet, +, 2^m, Rot$	$\oplus, \bullet, +$

than SASI. As presented on the Table 3.1, all the proposed ultra-lightweight primitives have been broken.

3.6 Cryptanalysis of the proposed protocol

Despite the original belief that the protocol was secure against the most common attacks over the radio, a recent work presented by Hernandez-Castro et al. [79] fully broke the protocol. In [79], three different attacks prove that the protocol is:

- prone to traceability (traceability attack),
- leaking stored secrets bits,
- open to full disclosure of the secret values K_1, K_2 and ID (Tango attack).

The later attack may reveal itself to be very efficient against most of the ultra-lightweight cryptographic primitives based on simple triangular functions. Its principle is to find good approximations of the secret values using combinations of the messages sent in clear over the radio. By making the average of these approximations over a small number of sessions (10 for the primitive presented in this chapter), it is possible to retrieve almost all the bits of the secret values.

[2] L denotes the bit length of one pseudonym or one key.

3.7 Discussion

As presented in the cryptanalysis, the proposed protocol has been totally broken and allows full disclosure of the secret values. The conclusion in [79] advise to integrate rotation operations into the design of [41], as in SASI. However, the various cryptanalysis of the later protocol ([38, 80, 13]) suggest that it would probably not solve the issues of the proposed primitive. The author of this thesis believe that it would be more secure to design a totally new protocol, from scratch, learning from the cryptanalysis of several protocols proposed in the literature. However, despite all the attempts to propose ultra-lightweight security protocols, yet none of them have proven to be secure. The latest protocols include additional operations such as bits mixing (Gossamer, [131]), and substitutions tables (to compute logarithms, SSL-MAP [136]), which are components used in the design of block ciphers. Furthermore, these protocols cost more than the latest published block and stream ciphers ([96, 42, 44]), without guaranty of stronger security. The border between ultra-lightweight and lightweight cryptography has never been that thin.

3.8 Conclusion

In this chapter, an ulta-lightweight cryptographic primitive was presented and analyzed. This chapter also discussed the current difficulties to design a secure ultra-lightweight primitive, which have a tendency to integrate additional features that are characteristic of block and stream cipher designs. Such a stream cipher is introduced in the following chapter.

Chapter 4

A2U2: A Stream Cipher for Printed Electronics RFID Tags

> "An inventor is simply a fellow who doesn't take his education too seriously."
>
> – *Charles F. Kettering.*

Foreword

The first five sections of this chapter are based on the conference paper *A2U2: A Stream Cipher for Printed Electronics RFID Tags* [42], written in collaboration with Dr. D. Ranasinghe and Dr. T. Larsen, presented during the *Fifth Annual IEEE International Conference on RFID*, in Orlando, Florida, USA, in April 2011. Section 4.7 is based on the conference paper *Cryptanalysis of the lightweight cipher A2U2*, written in collaboration with M. A. Abdelraheem, J. Borghoff and E. Zenner, presented during the *Thirteen IMA International Conference on Cryptography and Coding*, in Oxford, UK, in December 2011.

4.1 Introduction

RFID is one of the most promising technologies of the coming decade because of its ability to automatically and uniquely identify objects wirelessly [57]. It is also a key enabling technology of the 'Internet of Things' [39]. The range of applications enabled by RFID technology is so wide that it will soon become ubiquitous. However, the multiple advantages offered by RFID are linked to numerous challenges, which need to be overcome to realize the full potential of the technology. Providing security services, such as authentication necessary for e-ticketing applications and counterfeit detection and prevention, and ensuring privacy of both consumers and corporations are among the primary concerns. These issues must be addressed to facilitate the global adoption of RFID technology with confidence. With the emergence of printed electronics technology, RFID systems will soon reach a horizon where the cost of an RFID tag is no longer an impediment to deployments. In fact, printed electronics technology is believed to realize electronic systems at a substantially lower cost, unachievable with conventional single-crystal based integrated circuit (IC) fabrication [31]. However, thus far printed ink technologies are only capable of manufacturing RFID tags that operate at 13.56 MHz (High Frequency range). Nevertheless, such low cost tags will see the use of RFID technology grow in yet underexploited areas, such as postal items, books, e-tickets and airline baggage handling [39].

However, despite recent advances in printing processes and material science, integrated circuits on printed RFID tags are limited to a few thousand transistors [31]. Consequently printed ink based RFID tags are extremely resource-limited devices. Due to previous constraint and given the more recent advances in printed electronics, implementing a cryptographic primitive on printed RFID tags is a challenge that is largely unexplored. In the case of printed electronics RFID tags, the limitations are threefold:

- Cost (area) has to be low in order to be integrated on the chip. Current printed RFID tags support approximately 2,000 transistors (500 gates). Around 200 gates or less are expected to be available for implementing a security mechanism [31].

- Power consumption must be low to overcome the lack of a battery and to allow a tag to operate at a minimum read range. Consequently, a tag's power consumption is limited to tens of μWs. For example ISO 14443 specifications for tags operating at 13.56 MHz must provide a read range of 100 mm [83].

- Throughput of a security primitive should be reasonable to allow real time interaction for a large number of tags (generally in the order of hundreds of tags). For example ISO 14443 specifications for tags must transmit at 106 kbps [83].

Over the last decade, various research efforts have achieved lightweight security primitives for resource constrained devices [125, 158, 20, 44]. However, none of the proposed primitives are suitable for printed electronic RFID tags. Although the PRINT cipher published very recently has sought to address the challenges posed by printed electronic RFID tags, the cost of the block cipher still requires using nearly all available 2000 transistors on a printed RFID tag [96].

This chapter introduces a new hardware-oriented stream cipher, A2U2, conceptualized specifically to meet the extremely resource limited environment of printed electronic RFID tags. The central design blocks of A2U2 are based on:

- learning from vulnerabilities in previously published stream and block ciphers,
- key cryptographic ideas introduced by the work of Rueppel on stream ciphers [145],
- Shannon's ideas on confusion and diffusion [155].

In particular, A2U2 overcomes significant issues such as identical initialization values resulting in predictable bit streams on power-up, specific to the use of both block and stream ciphers in RFID-related applications. Such issues have not been dealt with in previously published designs. The result is a novel stream cipher that can be implemented on a small area to provide security services on printed ink RFID tags.

The rest of this chapter is organized as follows. Section 4.2 provides an overview of recent research carried on lightweight cryptographic primitives. Section 4.3 presents the key design principles that have been employed to design A2U2. Section 4.4 details the building blocks of the stream cipher. Section 4.5 presents an analysis of A2U2 and compares it to similar ciphers. Section 4.6 presents the simulation results of the implementation of A2U2 in ASICs. Section 4.7 presents the main cryptanalysis results of A2U2 and the possible improvements to the cipher. Finally, Section 4.8 concludes the chapter and provides further research challenges.

4.2 Related Work

Early candidates for resources constrained devices were based on hardware optimizations of well-known block ciphers. Feldhofer proposed an optimized version of AES [55], while Leander developed a low-cost version of DES [105]. XTEA [125] and SEA [158] are two other block ciphers, designed specifically for embedded devices. However, none of these ciphers were designed with RFID applications as a specific target platform and, as a result, are either too slow (AES, SEA, and TEA) or too expensive in terms of implementation costs (AES, DESL and TEA).

PRESENT is the first block cipher designed specifically for resource constrained devices [20] such as RFID tags. It is the result of work carried out in the EU Project UbiSec&Sens [163]. PRESENT is based on key ideas articulated by both Shannon and Rueppel, and used in the design of two modern block ciphers: DES and AES. PRESENT's architecture was developed to support two different but competing design goals: i) a low cost implementation (1000 gates), and ii) a high throughput implementation (200 kbps) [20].

A recent block cipher, KATAN [44], has reached a further milestone in terms of area minimization with a design tailor-made for low-cost RFID tags. A rigorous analysis of power consumption (< 1µW), area minimization (down to 480 gates), and throughput optimization (12.5 kbps when clocked at 100 kHz) has been achieved in its design. Finally, PRINT Cipher [96] is a more recent block cipher design that, for the first time, has targeted printed ink RFID tags. However, PRINT still requires at least 402 gates and the cost of the cipher is still well beyond the expected number of gates (less than 200 gates) available for a security primitive.

More interestingly, there are only a limited number of stream ciphers suitable for small embedded devices. In fact, none of the six stream ciphers resulting from the EU Project NESSIE (concluded in 2003) were satisfactory [126], leading to a new EU project called eSTREAM [52], to address this gap. Among the selected candidates of eSTREAM, two are of particular interest for RFID applications. GRAIN and TRIVIUM have implementation results with either low area (1294 gates for GRAIN), or low power (1.2 µW for GRAIN, 1.02 µW for TRIVIUM) [19, 20]. Again, these stream ciphers are well beyond the cost limitations of printed ink tags.

Finally, an interesting alternative to block and stream ciphers was proposed by Yüksel, with a scalable universal hash function called WH-16 [174], for RFID tags at a very low cost (460 gates).

On one hand, it is evident that KTANTAN [44] and, more recently, the PRINT Cipher [96] have advanced block cipher designs to new levels of compactness, while WH-16 achieved a similar goal with hash functions. Further reducing cost of implementation of one of these ciphers would be at the expense of its security, which is not a desirable outcome. On the other hand, the smallest stream cipher published so far is GRAIN and, with its 1294 gates, leaves a vast margin for improvement. Consequently, our approach has been to focus on developing a hardware-efficient stream cipher. Furthermore, during a conference in 2004, Adi Shamir pointed out the slow and continuous decline of stream ciphers to the benefit of block ciphers [153]. According to him, stream ciphers will be useful for only two kinds of applications in the long term: i) applications that require extremely high encryption speed (beyond Gbps), and ii) applications in resource constrained devices such as RFID.

Like Shamir, we believe that stream ciphers have been underestimated, mainly due to the difficulty of analyzing them. However, as we have shown with A2U2, stream ciphers can offer comparable, if not superior, alternatives to the current suite of low-cost block cipher designs.. In addition, stream ciphers also have the clear advantage of being much faster than block ciphers.

4.3 Design Principles

Block ciphers have been the most widely studied symmetric encryption algorithms so far. The research in the past decades has led to a relatively good understanding of the security of block ciphers [151]. The security of stream ciphers has only recently received increasing attention and the European projects, NESSIE and ECRYPT, have played a key role to this end. In NESSIE, no stream cipher made it to the final portfolio, as weaknesses had been discovered in all candidate stream ciphers. This illustrates how the study of stream ciphers is not as mature as the study of block ciphers. For example, common building blocks of block ciphers such as S-box constructions (used in PRESENT and AES) can be easily analyzed by existing tools [106]. However, until recently, the study of stream ciphers with a nonlinear update function was little studied and there are very diverse strategies for analyzing such stream ciphers [53].

Nevertheless, there is a large body of established guidelines and design principles for building stream ciphers such as those elaborated by Rueppel [145]. We have chosen to focus on synchronous stream ciphers with a nonlinear update function as these appear to offer the best combined security and performance. The design methodology is based on a system-theoretic approach first elaborated by Ruppel in [145]. Furthermore, it should be noted that cryptanalytic attacks against stream ciphers have often been based on exploiting flaws in their design (e.g. [121], [151] and [17]). As a result, every aspect of a stream cipher needs to be carefully designed. The following sections describe: i) the methodologies we employed to achieve a high level of complexity and security; and ii) the principles we used to reduce the implementation cost of A2U2, while balancing the need for an adequate level of security.

4.3.1 Use of Primitive Polynomial Function for LFSR

The use of a primitive polynomial is perhaps the most well known guideline regarding the use of LFSR. Nonetheless, it remains a key criterion to guarantee that the LFSR is of maximal length and has a period of 2L-1, where L is the length of the LFSR. Furthermore, the primitive polynomial should not be sparse (combination of a small number of connections) to avoid correlation attacks [122].

4.3.2 Use of Good Nonlinear Boolean Function for NFSR

The Nonlinear Feedback Shift Register (NFSR) is a more secure alternative to LFSR, since its nonlinear feedback function makes it cryptographically stronger against several attacks, such as correlation attacks and algebraic attacks [35]. However, good NFSRs are difficult to design and can easily be weaker than LFSR, if their nonlinear Boolean function is not carefully selected. The design aspects of nonlinear Boolean functions are generally omitted from publications on stream and block ciphers, since no simple and precise guidelines on how to build a strong function are available. Although several papers ([93, 149, 127, 123]) present some construction guidelines, their results remain too complex and difficult to be implemented in practice for stream cipher design. Nonlinear Boolean functions are characterized by the following four properties:

- Balancedness: A Boolean function is said to be balanced if the output probabilities of '1' and '0' are equal.
- Nonlinearity: The nonlinearity of a Boolean function f with n-variables is the Hamming distance [73] of f from the set of all affine functions with n variables.
- Algebraic Degree: The algebraic degree of an n-variable Boolean function is defined by the highest degree of its terms. The maximum algebraic degree of an n-variables Boolean function is n-1.
- Correlation Immunity: A Boolean function is said to be correlation immune of order m, if the distribution of their truth table is unaltered while fixing any m inputs [157]. By definition, it is impossible to design a Boolean function that is perfectly balanced, has highest degree of non-linearity, highest algebraic degree, and highest correlation immunity [123]. Siegenthaler [157] proved the following fundamental relation between the number of variables n, the degree d, and order of correlation immunity m of a balanced Boolean function. $m + d \leq n – 1$ Therefore, our task of building a good nonlinear Boolean function was based on constructing a function with the best possible combination of the properties we have discussed above, while considering the key requirement of minimizing the cost of implementing the function.

4.3.3 Exploit the Confusion and Diffusion Concepts

Introduced by Shannon in 1949 [155], the concept of confusion and diffusion is often applied in block ciphers with the use of substitution-permutation networks. In stream ciphers, the relation between plaintext, ciphertext and the key is different since the plaintext is not an input to the cipher. Therefore, diffusion (dissipation of plaintext statistics in the ciphertext) is not obvious; however, it is achieved by the filter at the output of the NFSRs. The filter function randomly distributes the ciphertext in the transmitted message. We have used Shannon's ideas of confusion to ensure that the secret key is used in a complex manner, to modify the feedback function as well as the inclusion of other irregularities in the cipher design. In addition to the nonlinear feedback functions, we introduce four sources of nonlinearity and pseudo-random irregularities in the cipher:

- Irregular change in the feedback function

- Randomized initialization value
- Randomized number of initialization rounds
- Irregular length of the ciphertext These design features not only contribute to increase confusion, but also ensure that the ciphertext generated for a given plaintext is uncorrelated, each time the cipher is used.

4.3.4 Learning from Previous Designs

One of the advantages of public scrutiny of published ciphers is the lessons learn. In general, any specific type of attack uses and/or reveals a particular flaw or weakness in a cipher design. Experience has proven that trying to fix a broken cipher will ultimately result in a new cipher that can still be easily broken (e.g. TEA [169]). However, even if the overall design of a given cipher is weak, some concepts might be worth reusing (the S-Box concept introduced in the DES is used in most contemporary block ciphers).

In constructing A2U2, we have built on the approach taken by Coppersmith in designing the Shrinking Generator (SG) [34]. However, the Shrinking Generator in its original published version has been broken with a few thousand bits of chosen ciphertext in a recent work [34], which has also highlighted the danger of using interleaved sequences, as in the SG. The weaknesses of the SG arise due to: i) the connection polynomials of the LFSRs being known, ii) the use of LFSRs, and iii) it uses identical initial values in the LFSRs at each session. In A2U2, we solve these issues by: i) modifying the feedback values irregularly, ii) using NFSRs, and iii) using pseudo-random numbers to initialize the registers, instead of fixed initial values. It is important to note here that A2U2 does not rely on the security of SG, since we only expound upon clock controlled generator architectural design concepts of the SG. From the clock controlled generators such as the SG, we have found a simple mechanism for achieving a nonlinear keystream. However, the practical use of shrinking generators designs is limited because they are unable to generate a bit stream at a constant rate. We have addressed this issue without compromising security as discussed in Section IV. E.

Furthermore, the study of previous cipher design principles is also useful to extract key concepts regarding area optimization. These ideas are presented in the following subsection.

4.3.5 Area Optimization

Since our goal is to develop a primitive for printed ink based RFID tags, reducing the number of gates is a critical design criterion. We employ two techniques to achieve a compact cipher: i) a design with short-length registers, supported by compact functional blocks, to increase nonlinearity in the design, and ii) the reuse of existing capabilities, such as the 32 random bits a tag is required to generate, partly for use as the random handle. The random handle uses 16 bits of the pseudo-random number generated by the tag, as defined in the EPC-global standards for Class 1 Generation 2 (C1G2) RFID tags [51], to identify itself to a reader during a communication session. Note that we assume such a feature is available in tags, or such a generator can be implemented with a modest cost.

Area optimization is often achieved through: i) processing shorter word lengths (8 bits as opposed to 32 bits at a time), ii) repeated use of hardware components (such as S-boxes) or increasing the number of rounds before ciphertext is generated to maintain security at the compromise of speed and iii) using shorter keys, block sizes (in block ciphers), feedback functions, register lengths, and permutations and substitutions boxes (P-box and S-box, see Section V. A).

Reusing design concepts of previous ciphers is a common practice, illustrated by block ciphers such as PRESENT [20] and PRINT Cipher [96]. In fact, to achieve the estimated 402 gates, the PRINT Cipher reuses a number of optimization and design principles. For example, the 3-bit S-box layer from SEA [158], hardwired permutations layer from PRESENT [20] (thus no logic gates are needed), and the counter, as used in KATAN [44], has been replaced by an LFSR to save additional gates.

A2U2 reuses an LFSR-based counter design as in KATAN, since it has been designed to minimize the number of gates. A2U2 also uses short-length registers, hardware-efficient nonlinear Boolean functions, and irregularities coupled with nonlinear functional blocks, to overcome the weaknesses posed by shorter register lengths. This is significant, since each shift register adds 6.25 to 12 additional gates to a design. The following section describes the detailed design of the cipher, based on the concepts we have discussed above.

4.4 Cipher Design

The A2U2 stream cipher is composed of four distinct building blocks. The four elements include: i) a counter, ii) a combination of two nonlinear registers, iii) an irregular change in the feedback function through a key-bit mixing mechanism, and iv) a filter function. An overview of the cipher design is illustrated in Figure 4.1.

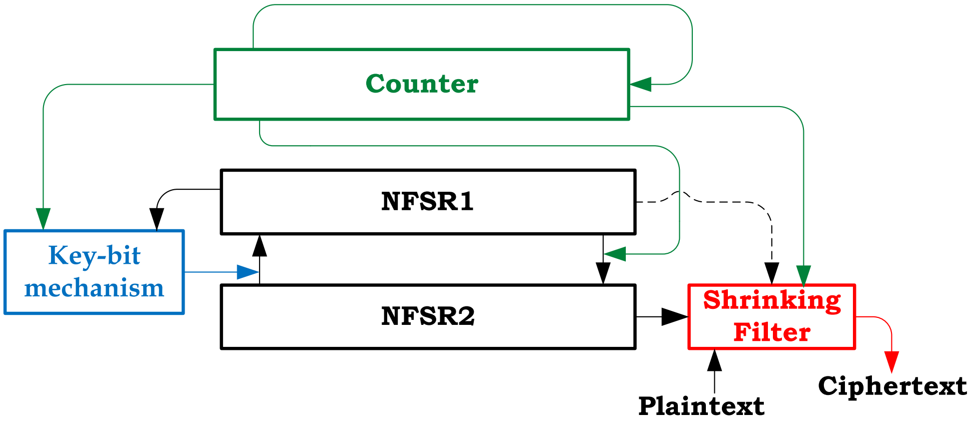


Figure 4.1: Overview of the A2U2 architecture [42].

4.4.1 The Counter

The counter is a 7-bit Linear Feedback Shift Register (LFSR), as represented in Figure 4.2. We denote the state of the counter LFSR at time t by $C^t = (C_t, C_{t-1}, ..., C_{t-6})$,for $t = 0, 1, ...$ The starting state after initialization (see below) is an all-ones state: $C^0 = (1, 1, ..., 1)$, . The LFSR uses the feedback recurrence

$$C_{t+1} = C_{t-6} + C_{t-3} \tag{4.1}$$

for updating the state. It is a standard LFSR with maximal period (i.e. $2^7 - 1$).

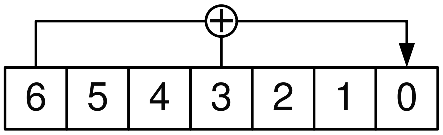


Figure 4.2: The Counter used in A2U2 [42].

The counter is initialized with an XOR operation of three strings:

- The 5 Least Significant Bits (LSBs) of a 32-bit (two 16- bit) pseudo-random numbers generated by the tag.
- The 5 LSBs of a 32-bit random number generated by the reader.
- The 5 last bits of the secret key.

The 5 bits resulting from this operation are input to the 5 Most Significant Bits (MSBs) of the LFSR (positions 6 to 2 in Figure 4.2). The second LSB of the LFSR (position 1 in Figure 4.2) is set to 1 to avoid an all-zeros string. Finally, the LSB of the LFSR is set to 0 in order to avoid an all-ones string, which is the condition to end the initialization process. During the initialization process, each clock cycle updates the counter until it reaches the all-ones state. The counter is clocked an irregular and secret number of times (ranging from 9 to 126), depending on the randomly selected initialization value of the counter.

After initialization, the counter simply works as an LFSR, where the bits are shifted clockwise, and the feedback is input in the LSB position. The counter also plays a role in the other building blocks of the cipher, as described in the following subsections.

4.4.2 The Two Nonlinear Registers

This part of the cipher (as well as the counter) has been freely inspired by the block cipher KATAN [44], which introduces a new combination of two NFSRs, where the feedback function of each NFSR provides the feedback to the other NFSR, as shown in Figure 4.3. We denote the state of the NFSRs by $A^t = (A_t, ..., A_{t-16})$ and by $B^t = (B_t, ..., B_{t-8})$. The update uses an auxiliary variable h_t(defined in Section 4.4.4) and the following non-linear feedback recurrences:

$$B_{t+1} = A_{t-16}+A_{t-14}A_{t-13}+A_{t-11}+A_{t-9}C_{t-6}+A_{t-6}A_{t-5}A_{t-4}+A_{t-3}A_{t-1}+1 \quad (4.2)$$

$$A_{t+1} = B_{t-8} + B_{t-7}B_{t-6} + B_{t-5} + B_{t-2} + h_t + 1 \quad (4.3)$$

Both NFSRs are initialized using the same process as the counter, with the following 26 bits of the random numbers (two 16 bit numbers generated by the tag) and the secret key. In the unlikely event (probability of 2^{-26}) of an all-zeros initialization value (IV), the bits introduced by the counter and the key-bit mechanism prevent the stream cipher from generating a series of zeros. Once initialized with IVs, the NFSRs are updated at each clock cycle, and the bits are shifted clockwise.

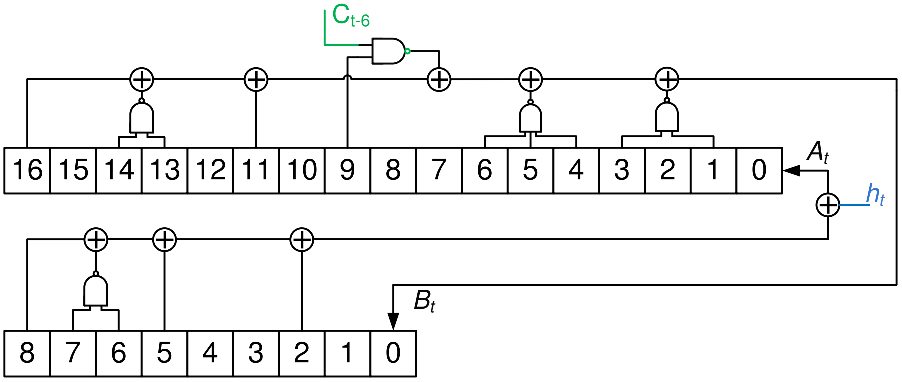


Figure 4.3: The 2 NFSRs of the A2U2 cipher [42].

4.4.3 Nonlinear Boolean Function Design

As discussed in Section 4.3, there is not a best design for a nonlinear Boolean function, but a good enough design based on a combination of the different properties outlined in Section 4.3. Further complicating their design is the requirement to ensure that the implementation of the functions in hardware results in a small circuit footprint, to ensure that costs are contained. Hence, the number of terms of the equation needs to remain small. Functions with high algebraic degree and high correlation-immunity are composed of a large number of terms, and a perfect nonlinear function is not balanced [123]. The upper bound on nonlinearity of balanced Boolean functions with n variables is theoretically 2^{n-1}–$2^{n/2}$ [25]. An example is a function of 41 terms with 7 variables [127]. It is clear that we cannot implement such a large function in A2U2. Our approach to construct a nonlinear Boolean Functions was to start with the function $f(x) = x_1x_2 + x_3x_4 + ... + x_{n-1}x_n$, which is a perfect nonlinear function, and then to remove terms to achieve a balanced function. Finally, we increased the algebraic degree to three, by combining the variables A_{t-6}, $A_{t-5}$and A_{t-4} in B_t.

4.4.4 The Irregular Key-bit Mechanism

The third component of A2U2 is a function that utilizes the tag's key. An extra bit (h_t) is XORed in the nonlinear Boolean function A_t given by (4.3), as shown in Figure 4.3. It increases the complexity of the cipher and modifies its feedback function, using the securely stored 56-bit private key. This extra bit generated is obtained with the nonlinear functional block, presented in

Figure 4.4.

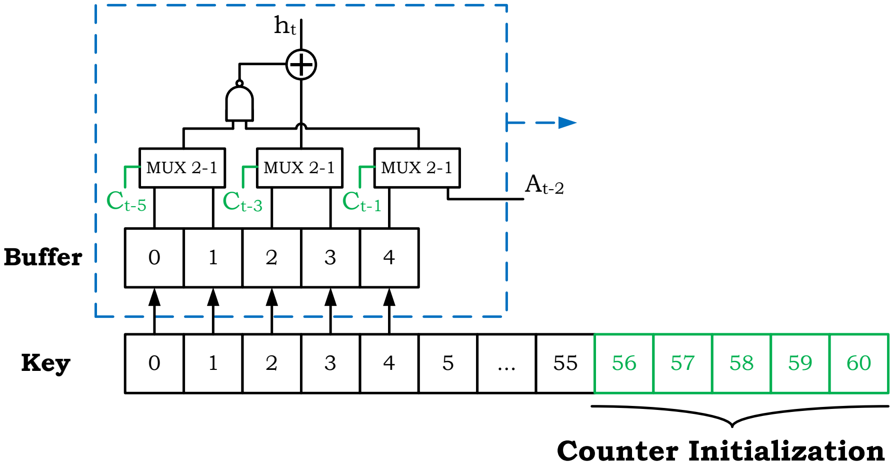


Figure 4.4: The key-bit mechanism [42].

As shown in Figure 4.4, the last five bits of the key are reserved for the counter initialization; they are not used in the process of key-bit generation. The key can be described as a rotation register, i.e. the state in time t is a rotated version of the initial state. If we denote the key bits by $(k_0, ..., k_{55})$, then each state of the register is defined as:

$$K^t = (k_{5t}, k_{5t+1}, ..., k_{5t+55}) \tag{4.4}$$

where all indices are computed modulo 56.

Each round, the first five bits of the register are stored in a buffer $S^t = (S_0^t, ..., S_4^t) = (k_{5t}, ..., k_{5t+4})$, and are used to compute the auxiliary variable h_t as follows:

$$h_t = \text{MUX}_{C_{t-5}}(S_0^t, S_1^t) \cdot \text{MUX}_{C_{t-1}}(S_4^t, A_{t-2}) + \text{MUX}_{C_{t-3}}(S_2^t, S_3^t) + 1 \tag{4.5}$$

where $\text{MUX}_z(x, y)$ is the multiplexer function that selects x if $z = 0$ and y otherwise.

The use of A_{t-2} considerably increases the period of the h_t sequence. Overall, the key-bit mechanism exploits the confusion principle described in Section 4.3.

4.4.5 The Filter Function

The final building block of A2U2 is the filter function, named the "shrinking filter" in reference to the clock controlled generator design of the Shrinking Generator [34]. The filter is represented in Figure 4.5.

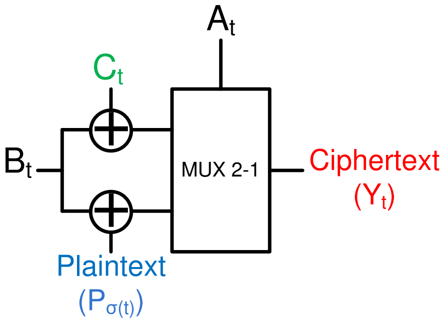


Figure 4.5: The "shrinking" filter [42].

If we denote the plaintext string by $P = (P_{0,}P_1, ...)$ and if we define $\sigma(t) = \sum_{i=0}^{t-1} A_t$ with $\sigma(0) = 0$, then the output of the cipher in round t is:

$$Y_t = \text{MUX}_{A_t}(B_t + C_t, B_t + P_{\sigma(t)}) \tag{4.6}$$

This filter ensures that only part of the input string (B_t) will be XORed with the plaintext, depending on a selector string A_t. We have overcome a significant drawback in the SG resulting from the irregular key stream (as a result of discarding the bits of the input string when not selected by the selector bit) by XORing the otherwise discarded bits of the input string with the LSB of the counter (C_t). This particular process presents several advantages:

- The “buffer” problem[1] at the output of the filter is solved, without additional hardware.
- For a series of plaintexts with a given fixed length, the resulting ciphertexts will likely have different lengths.
- The bits of ciphertext actually containing plaintext information are uniformly and randomly distributed within the ciphertext.

The output of the shrinking filter is the ciphertext of A2U2. The filter function starts operating once the initialization phase is complete. Then, A2U2 has a throughput of 1 bit of ciphertext per clock cycle.

[1] Due to the selectivity of the shrinking generator it may not output one bit per clock cycle, since some of the input bits are discarded. To solve this issue, the solution proposed in [33] adds a buffer of a few bits (16 to 24) at the output of the filter. Then the LFSRs are clocked twice as fast as the required output of the generator.

4.5 Cipher Analysis

4.5.1 Cost Evaluation

In lightweight cryptography, every additional gate is an added cost that must be carefully considered. The additional cost of a gate is even more significant for printed ink RFID tags. Block cipher designers have reduced implementation costs by reducing the key and block sizes, altering P-box and S-box designs and using serial architectures. In the latest hardware-optimized implementation of AES [72], substitutions and permutations represent more than 48% of the 3100 gates. To reduce this area, PRESENT [20] and PRINT [96] used shorter keys (80 bits compared to 128 bits), and block sizes (64 bits and 48 bits respectively, compared to 128 bits). Recent stream cipher designers have achieved area reductions by reducing the size of registers and the complexity of Boolean functions and filter function. Grain [53] uses two 80-bit registers, and hardware-expensive nonlinear Boolean and filter functions that requires 315 gates. As a comparison, the functions of A2U2 require only 42 gates, while its registers are only 17-bit and 9-bit long. In general, reducing the implementation cost of stream ciphers is relatively more difficult than block ciphers. Block cipher designers have the advantage of maintaining a sufficient level of security by increasing the number of rounds (547 for PRESENT and 768 for PRINT, compared to 160 for AES). However, increasing the number of rounds results in lower throughput (see Figure 4.6). A thorough analysis and design of each individual building block of A2U2 has ensured that it can be implemented in less than 300 gates. At the time of writing this section, the authors had not yet implemented the cipher in hardware. Therefore, all the evaluation details given are estimates. The details of implementation costs are presented in Table 4.1.

Table 4.1: Gate Estimate of A2U2 Implementation [42].

	Sequential Logic	Combinational Logic	Unit size (in GE) [59]	
Counter	43,75	2,25	2-input MUX	2,5
Register 1	106,25	15,75	2-input XOR	2,25
Register 2	56,25	10	2-input NAND	1
Key-bit mechanism	31,25	10,75	3-input NAND	1,5
Shrinking filter	-	7	D Flip-Flop [44]	6,25
Sub-total	237,5	45,75		
	TOTAL	**283,25**		

To measure the impact of this result, we compare A2U2 to some of the most well-known or most recent lightweight cryptographic primitives, as well as the most hardware optimized version of AES in Figure 4.6. To better reflect the difficulty of reducing the gate count, values in Figure 4.6 are represented in a logarithmic scale. In fact, the effort to reduce the cost of a cipher implementation by 50% from 1000 to 500 gates is as difficult (if not more) as down-scaling by 50% from 2000 to 1000 gates, while maintaining the same level of security. Such optimization often comes at the cost of trading-off throughput or power, or both. Furthermore, the number of published block and stream ciphers increase more rapidly with increasing cost thresholds, and rapidly diminish in number below the 1000 gate boundary. An overview of the most well-known ciphers can be found in [49]. To the best of the authors' knowledge, A2U2 is the only stream cipher below the cost threshold of 900 gates[2].

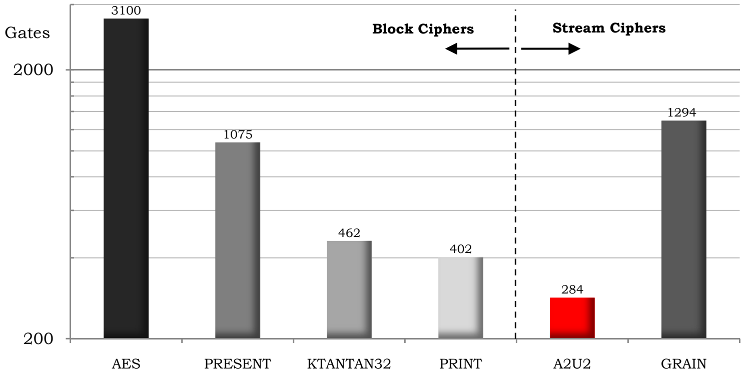


Figure 4.6: Comparison of lightweight cryptographic primitives' implementation costs in their area-optimized versions [42].

The current area available for security in printed ink tags being close to 200 gates, we believe that A2U2 is thus far the most suitable cipher for printed ink RFID technologies. However, printed ink technology is still within its early years of production, and the 200 gates constraint may be increased with further developments in printing techniques and research into material science.

[2] A5/1 stream cipher used in GSM networks can be implemented with less than 1000 gates (932 gates). This cipher relied on security through obscurity, but its mechanisms have been discovered and the cipher has been broken [17]. Therefore we did not include it in our comparison charts.

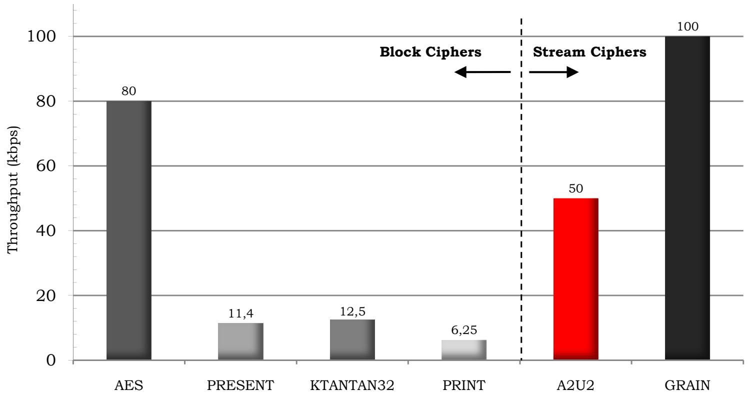


Figure 4.7: Comparison of effective throughput when clocked at 100 kHz [42].

4.5.2 Throughput Evaluation

As a stream cipher, once the initialization process is complete, one bit of ciphertext is produced per clock cycle. With the particular design of A2U2, based on the use of a shrinking filter, the effective encryption throughput will be approximately half the ciphertext generation throughput. Therefore, it is estimated to be close to 50 kbps when clocked at 100 kHz. We compare the throughput of A2U2 with other ciphers in Figure 4.6. The first observation we can formulate, based on Figure 4.6, is the clear advantage of stream ciphers over block ciphers in terms of throughput. Despite its effective throughput, which is half its actual throughput, A2U2 outperforms all the compact block ciphers compared by at least 400%. Note that in Figure 4.6 and Figure 4.7, we use AES as a benchmark but, because of its high cost, we do not consider it as a potential candidate cipher for resource constrained devices. We also remind the reader that we compare the same cipher implementations for cost and throughput. Throughput optimized versions of the block ciphers we compared do exist (see [20, 44, 96]); however, these designs are achieved at the expense of higher implementation costs. Therefore, we have decided to only consider hardware-optimized versions.

The throughput of A2U2 meets the highest transmit requirements of the ISO 14443 Standard [51], and is expected to be more than sufficient for applications envisioned with printed electronics RFID tags, which operate in HF (High Frequency) ISM (Industrial-Scientific-Medical) band.

4.5.3 Security Evaluation

The security of a stream cipher relies on two different aspects: i) the randomness and complexity of its bit stream sequence (its cryptographic strength), and ii) the design criteria followed. A2U2 has been implemented in software, using the C language. In order to evaluate its output, we used the Statistical Test Suite developed by the National Institute of Standards and Technology (NIST) [148]. This suite consists of a series of 15 tests that evaluate the randomness of a sequence. The various tests include basic tests such as the frequency test, which calculates the number of 0's and 1's, and more elaborate tests such as a linear complexity test, which evaluates if a sequence is complex enough to be considered as random. All the tests return a so called *p-value*, which is a condition of passing or failing the test. The *p-values* are numbers ranging from 0 to 1, while $p > 0.01$ is the condition of success of any given test. The resulting *p-value* itself does not present much interest, since different series generated will have completely different results for the various tests. The important result in these tests is the "pass or fail" condition. It is recommended to provide as input a sequence of at least 1 million bits to test its randomness. We ran the tests with sequences of 10 million bits, and all the tests returned a "success" value.

The ultimate goal of any cipher design is to be secure (e.g. one-time pads). However, in general, the aim is to provide a cipher for which no attack is better than a brute force attack.

It was conjectured by Rueppel in [145] and [146], and confirmed by Dai and Yang in [37] that the linear complexity of a periodic random sequence is close to the period length [60]. Empirical test results on the two registers of A2U2 (tested separately) demonstrate a period in the order of $2^{25.3}$. With A2U2's particular design, we estimate the overall period length of A2U2 (including the counter, the key-bit mechanism and the filter) to be 2^{70}. Therefore, we estimate the linear complexity of A2U2 to be close to 2^{70}, which currently guarantees lifetime secrecy (a brute force attack on RC5-72 would take an estimate 200 years with a few thousand computers [99]).

Furthermore, a shrunken sequence of a SG has a period of $(2^{N_2}-1) \times (2^{N_1}-1)$ [34], which would be slightly below 2^{25} if applied to the A2U2 cipher. Therefore our period length of 2^{70} has significantly improved on the period of a comparable length SG.

We expect that it will not be possible to improve significantly on a brute force attack, as a result of the nonlinear function blocks, the filter function, and the large linear complexity of the cipher.

Some of the design choices in A2U2 have been made to avoid possible attacks. On one hand, the variable number of rounds during initialization and the shrinking filter ensure that the cipher outputs a different ciphertext for an identical plaintext message each time the tag is powered. This feature prevents attacks such as replay and tag impersonation attacks. On the other hand, the filter function diffuses the plaintext bits within the ciphertext in a pseudo-random manner, making it impossible for an attacker to know which bits of the ciphertext contain relevant information.

4.6 A2U2 implementation simulation

Following the publication of A2U2 [42], we coded the cipher in Very-High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) in order to obtain a more accurate estimation of its implementation cost and its power consumption. While the entire cipher is digital, we anticipate that the estimated power consumption will actually be very close to a real (on-chip) implementation. We first draw a first draft using Simulink, then optimized it using Quartus. The optimal design of the cipher is presented on Figure 4.8.

We synthesized the design of the cipher using the software *Synthesis*, with a 130nm low-leakage standard cell library, and at two different frequencies 13,56MHz (RFID using the HF Band) and at 100kHz (for comparison purpose with other ciphers). We obtained the results presented in Table 4.2.

Table 4.2: Results of A2U2 implementation synthesis.

Cost parameters	
Number of nets	143
Number of leaf cells	49
Gate Equivalent	**226**

Power parameters		
	13.56 MHz	**100 kHz**
Net Switching Power	$1.56 \cdot 10^{-6}$ (8.54 %)	$1.16 \cdot 10^{-8}$ (8.53 %)
Cell Internal Power	$1.67 \cdot 10^{-5}$ (91.46 %)	$1.24 \cdot 10^{-7}$ (91.41 %)
Cell Leakage Power	$8.52 \cdot 10^{-11}$ (0.00 %)	$8.52 \cdot 10^{-11}$ (0.06 %)
Total Power	$1.83 \cdot 10^{-5}$	$1.35 \cdot 10^{-7}$

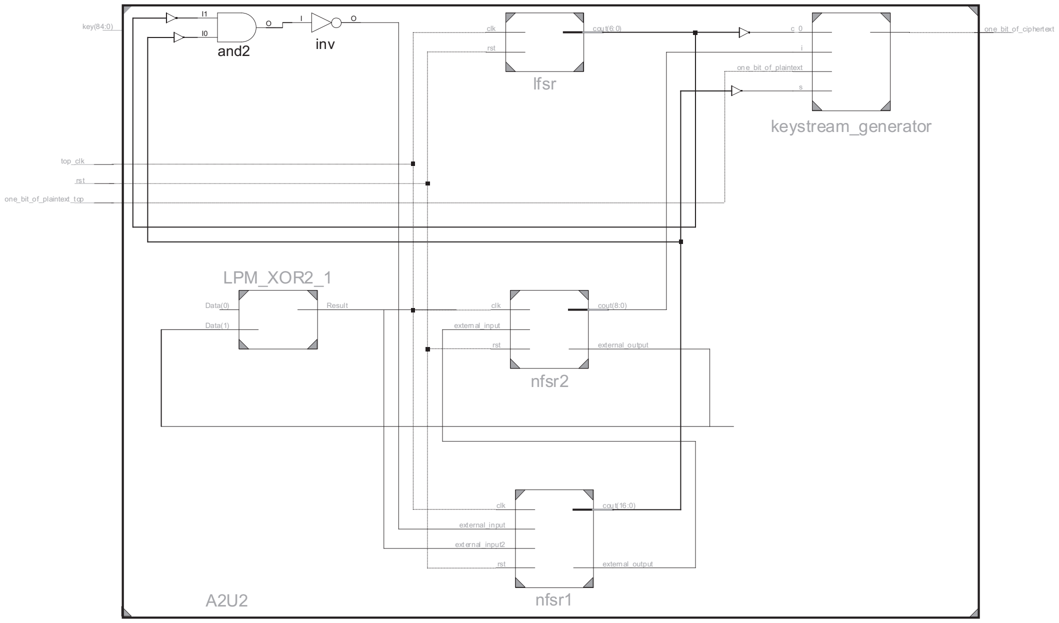


Figure 4.8: VHDL representation of A2U2.

We can formulate several interesting observations using the Table 4.2:

- The implementation cost is only 226 Gates Equivalent, which is a reduction of more than 20% with regards to the published estimated cost in [42] (see Table 4.1). This is of particular interest in this case, where the claimed target is to reach the implementation threshold of printed electronics RFID Tags (200 GEs, see Table 2.1). With such a small cost, A2U2 is one of the smallest cipher proposed in the published literature upon submission of this thesis.
- The cipher is using 18.3 μW of power in the HF frequency band (13.56 MHz), which is less than half of the power consumption limitation of passive RFID tags standard (see Table 2.1).
- The decrease of power consumption is linear with regards to the decrease of frequency from 13,56 MHz at 100kHz. This confirm the validity of equation 5.1 (see section 5.1.4) when the leakage power in neglectable.

By looking at the implementation results, it appears that A2U2 could almost be integrated in printed electronics RFID tags. However, section 4.7 reminds the limitations of such a tiny cipher.

4.7 Cryptanalysis of A2U2

Following the publication of A2U2 [42], a collaboration with a group of cryptanalysts at Danmarks Tekniske Universitet (DTU) has been engaged to evaluate the weaknesses of the cipher. Whereas the details of the cryptanalysis are attached in appendix of this thesis, this section integrates the main results of the cryptanalysis as well as the possible improvements to be applied to the cipher in order to protect it against the proposed attacks.

4.7.1 Attacks on A2U2

Several attacks have been performed on the cipher, which lead to the following conclusions:

- A *key recovery* attack requires two chosen plaintexts of length around 60 bits which are encrypted using the same key and initialization vector in order to recover secret key bits (excluding the 5 bits which are used to initialize the counter). Note that a 60 bit plaintext corresponds to a ciphertext of length 120 bit on average. The main computational effort consists in solving a linear Boolean equation system which can be done in well under a 1 second. This attack was inspired by the attack presented in [134].

- A *guess-and-determine* attack combined with *algebraic cryptanalysis* requires to collect at least 56 equations to determine a unique solution in 56 key variables. In the first two parts of the attack we guess 17 bits and obtain 8 equations. We expect that we have to guess every second value for A_t in the third part of the attack. Thus, we expect that we have to guess at least 24 further bits in the third part of the attack in order to obtain a fully determined equation system. The estimated complexity for this approach is 2^{49} bit guesses.

- Two attacks target the lowest number of initialiszation round of the cipher (9), (i) a *counter key bits recovery* attack recovers the 5-bit counter key using 2^{14} different state pairs with a specified difference, and (ii) a *master key bits recovery* attack recovers 32 secret key bits and 6 subkey bits using 8 plaintext/ciphertext pairs with a time complexity 2^{38}. Both of the attacks use known plaintexts and chosen IVs.

All these attacks exploit flaws in the design of the cipher. In the following paragraph, we summarize the weaknesses of the published version of A2U2 [42], and propose some possible improvements that would prevent the attacks described above.

4.7.2 Necessary changes & Possible improvements

It is important to mention that the following possible improvements may be prone to other types of attacks, and a full re-evaluation of the cipher would need to be performed in order to assess the strenghts of these potential changes. The following weaknesses make the above attacks possible:

- The fact that the adversary knows the counter state at every instant facilitates the cryptanalysis. In particular, it signicantly simplies all algebraic expressions in the cipher, bringing the algebraic degree down to 2 or even 1. Note that to fix this problem, a completely new mechanism for key/IV setup has to be developed. In addition, the size of the counter register has to be increased signicantly, since with the current register size, the adversary can just guess the counter starting state which only contributes a factor of 2^7 to the attack complexity. Furthermore, the potentially low number of 9 clock cycles to initialize the cipher is not sufficient to ensure a full mixing of the IV bits inside the registers, giving the attacker some valuable information regarding the untouched bits. There are several possible and complementary ways to fix these weaknesses:

    1. In order to increase the number of rounds during initialization, the counter can be set to an all-ones sequence and run for 127 clock cycles (2^7-1) until it reaches an all-ones sequence again. This would ensure a proper mixing of both registers bits.
    2. At the end of the initialization phase, the counter value could be replaced by the 7 LSB of the secret key, reserved for this sole purpose. The 2 LSB that are fixed in the original cipher description for initialization would no longer be required since this new operation would occur after the 127 initialization rounds. This would prevent an attacker from knowing the exact sequence of bits generated by the counter.
    3. However, this second measure does not solve the problem where the attacker only guess the initial 7 bits of the counter. As mentioned

above, increasing the length of the register would solve this issue, but it would be at the expense of numerous additional gates (which contradicts the design principles of this cipher). A more reasonable solution, in terms of gates, would be to XOR one (or several) external bit(s) to the feedback function of the register. These bits could for example be the output of the key scheduling mechanism or a bit from one of the nonlinear registers. In this way, the length of the register period would be signicantly increased at a very low cost.

- The original design mixes random values both from the reader and the tag to avoid replay and man-in-the-middle attacks [42], which are of common concern in RFID systems. However, the fact that the adversary has an influence (up to total control) on the IV is a serious weakness for a stream cipher. In particular, it must not be possible for the attacker to repeat the IV (nonce-respecting adversary [143]). Thus, it is common practice for stream ciphers that the sender chooses the IV, e.g. as a counter or random value. This design criterion should be adhered to. The registers' IVs may be set as an XOR operation of the tag's pseudo-random number and the secret key.

- The key size itself should be increased, since a total key length of 56 bit is not strong enough for modern standards. Due to the structure of the key scheduling mechanism, the key size needs to be relatively prime with 5 in order to obtain the longest rotation possible before reusing the same 5 key-bits. In addition, 7 bits need to be reserved for the counter (after initialization). A minimum of 88 bits (81 + 7) would seem to be a more reasonable choice. Given that the primary target of the printed tag encryption protocol is to encrypt an EPC of 96 bits [51], having a key size longer than this value would be at the compromise of the area available for the cryptosystem.

As mentioned earlier, when these weaknesses are fixed, a full re-evaluation of the cipher is necessary. Note that in particular, the output unit leaks information (in form of a correlation) about the inner state. Whether or not this information can be used for an attack depends on the inner workings of the cipher. No simple answers can be given here without a full specification of the new design.

Even though having short-size registers and building blocks will necessarily affect the strengh of a cipher, design decisions have to stick to the need of designing a cipher small enough to be integrated in printed electronics, where every additional gate counts[3]. This particular target may have some practical sides regarding the amount of data one can collect. According to the specications of the ISO 14443 Standard [83], with a communication session of 400ms between tag and reader, it would take more than 5 years to collect 2^{36} bits of ciphertext from a single tag, and approximately 6 months to collect 2^{39} bits of ciphertext from 100 tags using the same key. This number could be a limiting factor to break an improved version of A2U2, where no practical attack could be performed with less than 2^{39} bits of known plaintext/ciphertext.

4.8 Conclusions

In this section, we have presented a novel stream cipher, A2U2, which can be implemented in less than 230 gates. The approach taken was based on following a path less traveled, which is the design of a stream cipher as opposed to the various developments in lightweight block ciphers such as KATAN and PRINTcipher. The cryptanalysis of A2U2 revealed several serious flaws in its design and highlighted the difficulties of combining a compact design and a high security in the same cipher. Given that the attacks presented earlier can be fixed, A2U2 could be a worthy candidate for implementation in printed electronics RFID tags due to its compactness, simple computational operations, and a throughput of 1 bit per clock cycle (after initialization).

Note regarding the cipher name: A2U2 is the result of merging the two university acronyms, AAU and AUU, that participate in this paper. It is also a wink to the book *H2G2* [2], which was a source of inspiration and distraction during the author's research.

[3] Every additional register in the design implies an increase of (at least) 6.25 GE. Increasing the size of a static key (which is the case for passive RFID tags) comes at a lower cost of 1GE/bit. Moreover, it is often assumed that some space is reserevd for the key inside the device, therefore coming at practically no additional cost

Chapter 5

Comparison of Lightweight Cryptosystems

> "Success is the ability to go from one failure to another with no loss of enthusiasm."
>
> – *Winston Churchill.*

Foreword

This chapter is a selected part of the journal paper *Lightweight Cryptography: Classification and Evaluation*, written in collaboration with Dr. D. Ranasinghe and Dr. Q. Sheng, and submitted in January 2011 to the Journal of Cryptology[1].

5.1 Primitives Comparisons

Ideally, the most suitable lightweight cryptographic primitive would be highly secure, inexpensive, and consume negligible power. However, the reality is that providing security to resource limited platforms is a compromise between divergent parameters. A security primitive needs to balance between cost, level of security, performance and usability of the solution [50]. These trade-offs explain the challenges behind research on lightweight cryptographic primitives explored in Section 2.1.

Competing factors illustrated in Figure 5.1 (cost, performance and security) implies that ciphers achieve a good enough compromise between each factor for a

[1] At the time of delivering this thesis, the paper is still under review.

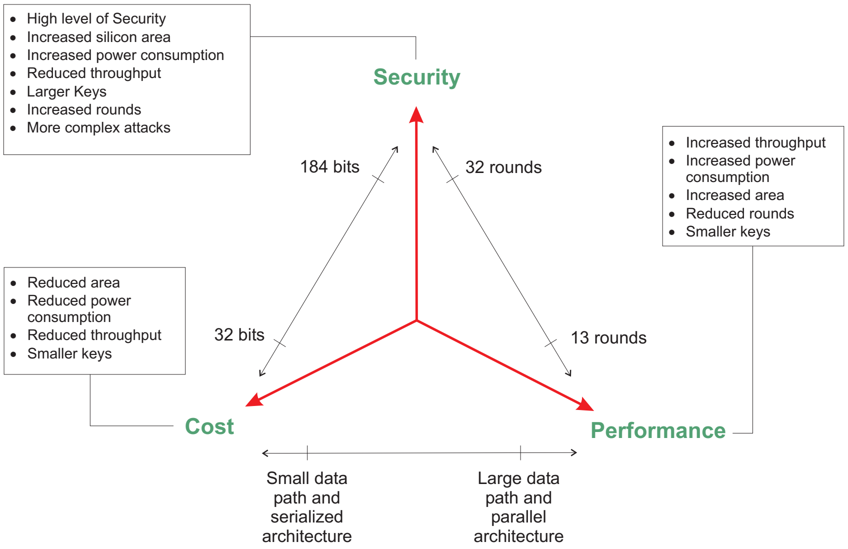


Figure 5.1: Cost, Performances and Security trade-off.

given application. All the alternative methods we have examined are based on managing the competing objectives outlined in Figure 5.1. Therefore, the following sections will compare and evaluate the primitives we have surveyed along the dimensions of security, cost, and performance. Since throughput and power consumption are closely related we have considered performance in terms of both power consumption and throughput of the cipher. Achieving higher throughput through increased clock rates will result in increased power consumption but power available to a low cost and passive device is severely restricted. Therefore power consumption should be part of the performance evaluation.

5.1.1 Comparison Overview

A significant goal with lightweight primitives is to provide an adequate level of security while being practically implementable in resources limited platforms such as passive low cost RFID. It is then worthwhile to consider the most optimized version of AES (Advanced Encryption Standard), being the standard for providing the highest level of security in modern communication systems, currently available as a benchmark for evaluating the newly developed cryptographic primitives. Clearly the novel designs must satisfy the requirements of the target platforms considerably better than the AES implementations in terms of power, cost and performance while providing an acceptable level of security to a target application.

5.1.1.1 A Benchmark: Hardware Optimized AES

Feldhofer et al. [54] presented an implementation of an AES encryption function based on a 128 bit key requiring only 3,400 gates (and 256 bits of RAM) and thus bringing cost-efficient strong authentication closer to reality for RFID tags. More recent optimization of AES has been published in [72]. The architectures therein are based on 8 bit data paths and have shown to require 3,100-3,200 gates. Implemented on a 0.13 µm process and run at 80 MHz [72] has shown to have a throughput of 100 -120 Mbps. However due to the high frequency of operation the power consumption of these circuits are over 2 mW. A low power implementation of AES, published by Feldhofer et al. in [55], presents an optimization requiring only 4.5µW at 100 kHz.

The best known attacks against AES are based on side-channel attacks on specific implementations. Side-channel attacks do not attack the underlying cipher and have nothing to do with the security of the cipher itself. But such attacks are based on weak implementations of the cipher on systems which inadvertently leak data. An example of such an attack is a power analysis attack. Thus any low cost implementation of AES on RFID should be based on using techniques to avoid leaking information to adversaries who can monitor the powering channel from great distances. Consequently any remedies implemented will add to the overall cost of the cipher while reducing performance and increasing power consumption (published literature on the vulnerability of hardware optimized AES to side-channel attacks are currently not available).

AES is certainly the most robust cipher. For this particular reason, AES is useful as a representation of an upper limit for area size, since any primitive with a larger implementation size would not present any advantage over AES. Yet, its hardware optimized version is still too greedy in terms of resources (both area and power consumption) to be implemented in low-cost RFID tags.

Table 5.1: Comparison of security primitives.

Cipher	**Key Size (bits)**	**Block Size (bits)**	**Area (GE)**	**Power Consumption (μW)**	**Throughput @ 100 kHz (KBit/s)**	**Tech. (μm)**	**Level of Security**
Benchmark cipher							
AES-optimized [55]	128	128	3400	4.5 @ 100 kHz	12.4	0.35	Very High
Symmetric Key Primitives: Block Ciphers							
TEA [84]	128	64	1984	39 @ 230 kHz	22	0.35	Low
SEA (93 rounds) [115]	96	96	1333	3.22 @ 100 kHz	16	0.13	Low-Moderate
DESL [105]	56	64	1848	1.6 @ 100 kHz	44.44	0.18	Low-Moderate
DES [105]	56	64	2309	2.14 @ 100 kHz	44.44	0.18	Low
DESXL [105]	184	64	2168	N/A	44.44	0.18	Moderate-High
mCRYPTON-64[2] (13 rounds)[112]	64	64	2420	N/A	492	0.13	High
PRESENT-80 (4 bits) [144]	80	64	1650	3.86 @ 100 kHz	200	0.18	High (High Risk)
PRESENT-80 (4 bits) [144]	80	64	1075	2.52 @ 100 kHz	11.4	0.18	High (High Risk)
KTANTAN-32 [44]	80	32	464	0.15 @ 100 kHz	12.5	0.13	Low
PRINTcipher-48[96]	80	48	402	2.6 @ 100 kHz	6.25	0.18	Moderate-High

table continues next page

[2] The implementation of mCRYPTON is not optimized for minimum area or minimum power.

Cipher	Key Size (bits)	Block Size (bits)	Area (GE)	Power Consumption (μW)	Throughput @ 100 kHz (KBit/s)	Tech. (μm)	Level of Security
Symmetric Key Primitives: Stream Ciphers							
GRAIN (16 bit word size)[3][53]/[66]	80	-	3360 (Low power) 1294 (Min Area)	1.2 @ 100 kHz (Low power) 3.3 @ 100 kHz (Min Area)	123 (low power) 100 (Min area)	0.35 (low P) 0.13 (min A)	Moderate-High (High Risk)
TRIVIUM (16 bit word size)[4][53]/[66]	80	-	3090 (Low power) 2599 (Min Area)	1.02 @ 100 kHz (Low power) 5.6 @ 100 kHz (Min Area)	72 (low power) 100 (Min area)	0.35 (low P) 0.13 (min A)	Moderate-High (High Risk)
A2U2 [42]	56+5	-	226	0.135 @ 100 kHz	50	0.13	Low
Keyed Hash Functions							
WH-16 [174]	512	64	460	2.95 @ 500 kHz	8.3	0.13	Low
SQUASH [175]	128	32	2,646	0.036 @ 100 kHz	0.1	0.13	High
Physical Primitives: Physical One Way Functions							
PUF-64 (Using 128 CRPs)	-	128*0.4 = 52 bits	856	N/A	2.048	0.18	Low-Moderate
PUF+LFSR-64	-	128*0.4 = 52 bits	2392	N/A	15.488	0.18	Low-Moderate

5.1.1.2 Summary

Table 5.1 outlines a comparison of the lightweight methods discussed previously based on relevant published information. Where not explicitly mentioned, equivalent gate counts are for minimum area implementations with the minimum size data paths. First, we would like to draw the reader's attention to the following information:

- The data presented for PUFs is based on using 200 CRP pairs where the underlying assumption is that in the best case scenario, given a 40% inter-chip variability, a PUF will produce 80 bits of information. This amount of keying material is comparable with PRESENT-80.

[3] The data is obtained from an independent source and not from the report by ECRYPT I project.

[4] The data is obtained from an independent source and not from the report by ECRYPT I project.

- The throughput rates for the direct application of PUF circuits only takes into account the number of clock cycles to generate a 200 bit response provided that a readily available set of two hundred 64-bit challenges are available to the chip.
- Two versions of PUF are considered, one with and one without LFSR. The LFSR is used as a PRNG for generating two hundred 64-bit challenges to improve on the overall performance of the cipher.

The AES standard primitive is provided as both a benchmark for evaluating other ciphers as well as an option that should seriously be considered for applications that require security beyond a few months or years.

A fair comparison is rather difficult due to various requirements of ciphers being different from each other and the different technologies used to implement the ciphers. All the power consumption estimations are based on simulation tools used by the various authors. It is important to note that all the hardware implementations considered for block ciphers are only capable of encryption. Providing both encryption and decryption capabilities will almost double the required gate count given in Table 5.1. Consequently, where both encryption and decryption are required, block ciphers are not as attractive as reported in Table 5.1.

5.1.2 Security

Given the reported attacks on ciphers, all primitives can be implemented on an RFID tag to provide various levels of security as none of the attacks, with the exception of TEA, WH-16, A2U2, and KTANTAN-32 are easy to perform in a reasonable amount of time (see Figure 5.2). It is important to ensure that the recommended security parameters and the number of rounds are utilized in their implementations. The most significant concern with regards to security is related to new designs of ciphers (PRESENT, Trivium, Grain, SQUASH and KTANTAN) and to some extent DESL and DESXL since the period of public scrutiny of these ciphers are relatively short. Hence there is an inherent risk associated with using these ciphers.

Figure 5.2 quantifies the level of security provided by the lightweight cryptographic primitives we have considered. Security analysis of the ciphers PRESENT, SEA, DESL and mCRYPTON which are based on implementations that

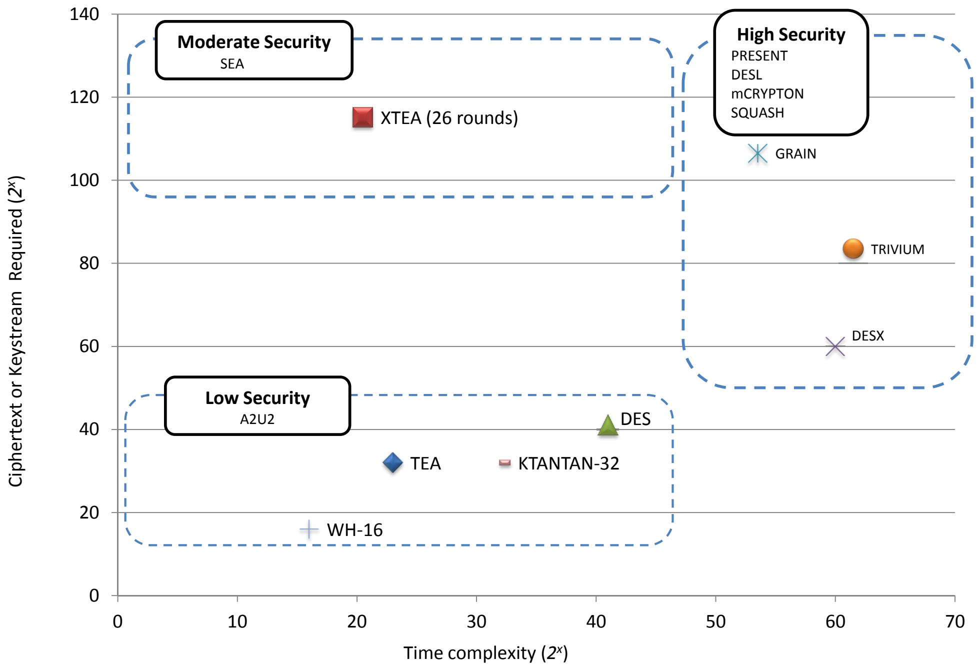


Figure 5.2: Security Matrix.

use the recommended number of rounds are not available. For example, under the assumption that a cryptanalyst needs only an approximate 28 of the 31 rounds in PRESENT to mount a key recovery attack, linear cryptanalysis of the cipher would require of the order of 2^{84} known plaintexts and ciphertexts but none of the attacks are possible for the full 31 round cipher. Thus it is assumed at the time of writing that there are no known attacks better than brute force attacks for a complete implementation of these primitives. Therefore SEA is related as providing a moderate level of security while PRESENT, DESL SQUASH and mCRYPTON are regarded as providing a high level of security. The level of security provided by A2U2 in its original version being too low, it does not appear on this figure.

5.1.3 Cost (Area) and Performance

The cost of various ciphers ranges from 226 gates for A2U2 to 2,599 gates needed for TRIVIUM implementations. In the following discussion, we will evaluate the primitives based on hardware cost, performance (throughput), and performance obtained per unit cost of hardware in order to compare their usefulness to low-cost RFID applications. Although latency is not directly considered, it is mostly reflected in the throughput data (except for stream ciphers, universal hash functions and PUF based approaches). On Figure 5.3, the most efficient primitive would be on the top left corner (i.e., minimal gate area with maximal throughput) while the least efficient would lie on the bottom right corner (large gate area and low throughput). Since the level of security is not taken into account in Figure 5.3, AES is placed at the bottom right. The throughput values for all of the ciphers are calculated based on each primitive being clocked at 100 kHz so that they can be meaningfully compared. From Figure 5.3 we can see that only A2U2, PRESENT, GRAIN, KTANTAN-32, and to some extent mCRYPTON achieve the best compromise in terms of low cost and performance (represented by the thick dashed line).

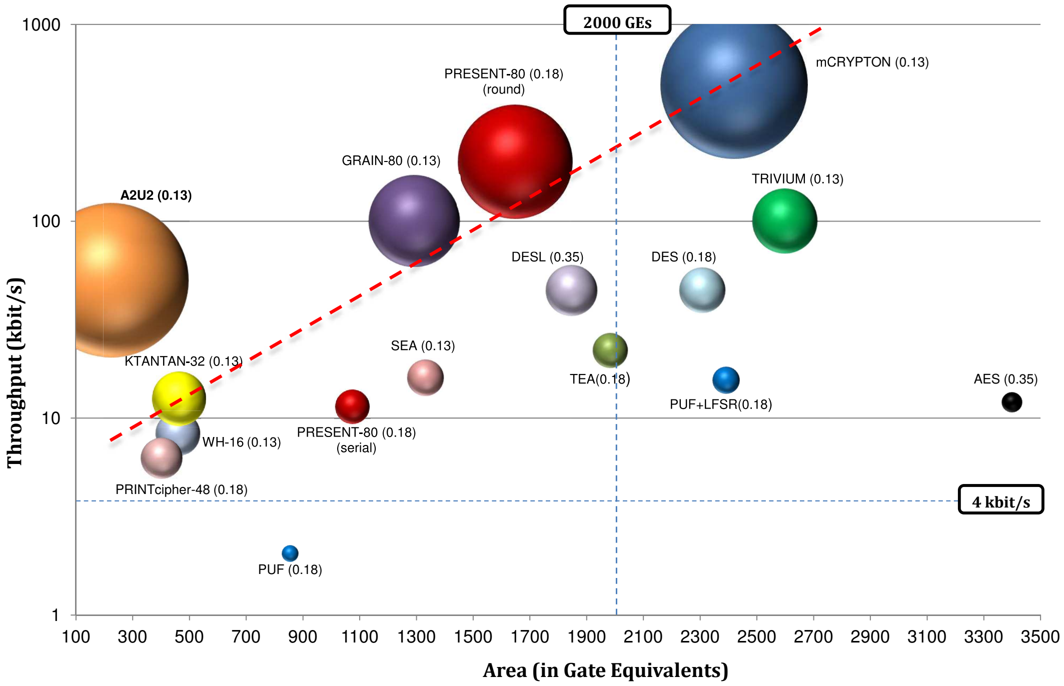


Figure 5.3: Comparison of throughput, cost and throughput per unit gate cost (bubble size).

5.1.4 Power Consumption

Different power consumption estimates from various fabrication processes, some not from standard CMOS processes, have often made it difficult to compare lightweight cryptographic primitives from a holistic perspective that not only considers the cost of implementation. In this section, we will consider transforming power consumption of various security primitives to a single technology, to allow a fairer evaluation of the various approaches and their suitability as a lightweight cryptographic primitive.

The power dissipation in CMOS devices can be characterized by the following equation [168]:

$$P = \underbrace{\left(\alpha C_g V_{DD}^2 f\right)}_{P_{Dynamic}} + \underbrace{\left(\alpha Q_{SC} V_{DD} f\right)}_{P_{short-circuit}} + \underbrace{\left(I_{leak} V_{DD}\right)}_{P_{leakage}} \tag{5.1}$$

In equation 5.1, α is the switching probability of a gate, C_g is the gate capacitance of the circuit, V_{DD} is the input voltage, *f* is the operating frequency, Q_{SC} is the quantity of charge carried by the short circuit current during transitions and I_{leak} is the total leakage current. C_g, Q_{SC}, V_{DD} and I_{leak} are technology dependent, and α is dependent on the primitive design.

In processes with feature size above 180 nm, leakage power is typically insignificant. Similarly, short-circuit current represents less than 10% of the dynamic power and can be neglected in nanometer technologies where the threshold voltage (V_t) is low [168]. Therefore the main source of power dissipation is the dynamic power, which we refer to as "power" in the rest of this chapter unless specifically mentioned otherwise.

The disparity in power results claimed by the various primitives in the literature is mainly due to different CMOS process-specific parameters such as V_{DD} and varying operating frequencies employed by the designers. This lack of coherence between results makes it almost impossible to compare primitives on a fair basis, for the same reason as one cannot compare apples and pears.

Our approach aims at converting the power consumption of a primitive to an estimate of the same design in another technology where all the primitives are clocked at 100 kHz. We draw particular attention to the fact that we are providing an estimate, not a precise value of power consumption, as it would require implementing each primitive in the target technology.

From equation 5.1, we only consider the dynamic power. V_{DD} is CMOS process specific while f (operating frequency of the ciphers) is a published value with regards to a primitive. However, we need to determine C_g and α. Since we are not modifying the design of a given primitive, we assume that the switching probability of the circuit α would not change from one technology to another, therefore α is assumed to be a constant. Then it remains to estimate the gate capacitance C_g for various CMOS processes used by the ciphers. We have used published values of C_g obtained from [168] for different CMOS processes (2.20 fF/µm for 0.35µm process, 2.06 fF/µm for 0.18µm process and 1.34 fF/µm for 0.13µm process). These values are then converted to get an approximation of a gate capacitance for a 2-input NAND gate (or gate equivalent - GE) as reported for the area in the various primitives.

The most significant source of error in the power estimation is a result of the simplistic assumption that the circuit is exclusively using 2-input NAND gates. Although our technique is not extremely precise, it is adequate for obtaining reasonable estimates when comparing results from different technologies. Furthermore, there appears to be no work closely related in the literature to provide a better estimation.

We decided to convert the implementations to a 0.18µm CMOS technology as it is currently the technology used for large-scale production of low-cost RFID ICs. The conversion from 0.35µm to 0.18µm is performed through the technique described above. However, we also need to take into account the leakage power in the conversion from 0.13µm to 0.18µm. Trivium, Grain, A2U2, SQUASH and WH-16 provide values for leakage power. Since leakage power is mainly proportional to the area, an estimate is calculated for the remaining primitives. The leakage power is reduced from the total power consumption and the remaining dynamic power is then converted. The Table 5.2 shows the power estimates for the various primitives in the 0.18µm CMOS technology where the operating frequency of the cipher is 100 kHz.

Based on the values obtained in Table 5.2, we can draw a comparison of power consumption related to the size of the primitive as shown in Figure 5.4. Since PRESENT-80 has reported implementations in both 0.35µm and 0.18µm CMOS processes the dashed line represents an estimate of the error in the power conversion technique (around ± 1 µW). The bubbles' size represents the area per µW of power. A few primitives such as mCRYPTON and the PUF are not listed because their power consumption values have not been reported in literature thus far. SQUASH is also missing because its specifications do not meet the requirements of low cost RFID platforms.

Table 5.2: Estimation of power consumption in a 0.18μm CMOS process.

Primitive	Area (GE)	Tech.	Published Power Consumption (μW)	Estimated Power Consumption (μW)
PRESENT-80	1000	0,35	11,2	1,60
TEA	1984	0,35	39 (230 kHz)	2,43
AES	3400	0,35	4,5	0,64
PRESENT-80 (round)	1650	0,18	3,86	3,86
PRESENT-80 (serial)	1075	0,18	2,52	2,52
DESL	1848	0,18	1,6	1,60
DES	2309	0,18	2,14	2,14
PRINTcipher-48	402	0,18	2,6	2,60
GRAIN	1294	0,13	3,3	5,17
TRIVIUM	2599	0,13	5,6	8,43
WH-16	460	0,13	2,95 (500 kHz)	0,60
SEA	1333	0,13	3,22	4,65
NTRU	2884	0,13	1,74	1,67
SQUASH	2646	0,13	0,036	0,12
KTANTAN-32	464	0,13	0,15	0,72
A2U2	226	0,13	0,135	0,65

Again, Figure 5.4 indicates that the block ciphers A2U2, KTANTAN-32, PRINTcipher-48 and PRESENT-80 along with the hash function WH-16 present the best lightweight primitives in terms of power consumption and implementation costs. However, the stream ciphers, at the noticeable exception of A2U2, seem to consume the highest levels of power as well as have a low number of gates per microwatt indicating a high level of activity in the shift register based designs. In contrast, AES performs relatively well given its large area and low power consumption as shown on Figure 5.4, where its large area per microwatt is possibly a result of using the SP-networks, as opposed to shift registers.

5.2 Discussions

We have surveyed and evaluated low cost security primitive designs. These designs are either constructed with platform specific constraint parameters (see Table 2.1) limiting their design, or they were published as being suitable for extremely resource constraint devices, currently exemplified by passive RFID

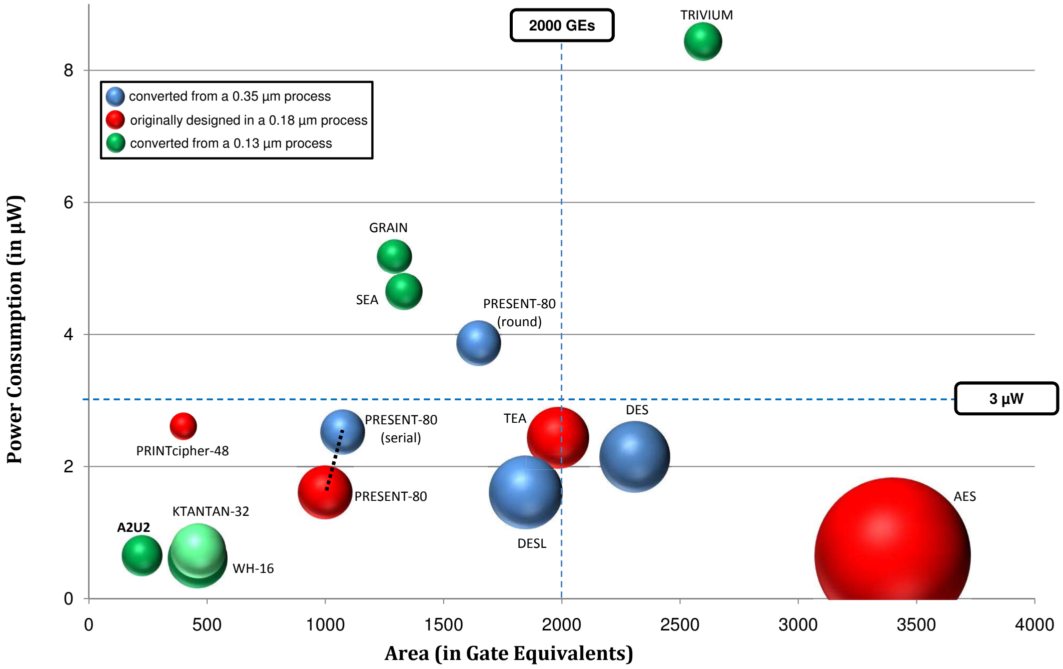


Figure 5.4: Estimate of Power Consumption vs. Area.

tags based on both single crystal silicon integrated circuits and printed semiconductor tags. In Table 2.1, we highlighted the following platform specific design constraints that must be met by cryptographic primitives to be considered suitable for practical implementations: (i) capable of being implemented using 2000 gates or less, (ii) consume about 20-30 µW of power, and (iii) have a throughput of at least 40 kbps.

The throughput and power consumption of the ciphers in this thesis were evaluated using 100 kHz clock rates while in reality the clock rates are likely to be at least 1 MHz. Given that the highest speed at which (provided both ones and zeros are equiprobable in the transmitted bit stream) a tag implementing the ISO 18000-6C or EPCglobal C1G2 air interface protocol can transmit is 640 kbps, a throughput of 40 kbps can be achieved by a cipher capable of 4 kbps running at 100 kHz. However a 10 fold increase in the clock rate is likely to produce a 10 fold increase in power consumption (see equation 5.1). Therefore, power consumption values should also be scaled by a factor of 10 as the operating frequency is scaled. As a result, only primitives reported as consuming 2-3 µW are suitable for single crystal silicon IC based tags which corresponds to 20-30 µW in a practical implementation (see Table 2.1) [108].

Hence lightweight cryptographic primitives evaluated in this thesis needs to satisfy: (i) throughput rates of 4 kbps, and (ii) power consumption limits of 2-3 µW.

It appears that the level of security of a majority of the ciphers is adequate for most applications enabled by low cost RFID technology. Throughput rates of all the ciphers, with the exception of PUF are above the 4 kbps minimum throughput rate (see Figure 5.3). Furthermore, the minimum throughput requirement is air interface protocol and electromagnetic compatibility regulation dependent but, more significantly, an end-user requirement based on a particular application. Then, hardware cost and power consumption become the differentiating factors.

Although power consumption varies significantly (see Figure 5.4), it is only critical in far field operations requiring tags to operate at great distances. From Figure 5.4, it is clear that most primitives, with the exception of Trivium, consume less than 5 µW of power. Furthermore power consumption is a limiting factor only for tags that operate at UHF frequencies in the far field, where power consumptions in excess of that required for reading EEPROM memories significantly reduce the operating range of the tag.

Therefore, the most significant factor allowing for the broadest classification is cost (i.e., the implementation area or the equivalent gates) when deciding suitability of a primitive for classification as a lightweight primitive. Figure 5.3 illustrates the selection of primitives that are indeed suitable lightweight cryptographic primitives based only on their cost of implementation (noted by the dashed line indicating ciphers that can be implemented with no more than 2,000 gates). It would be appropriate to consider all those primitives, that can be implemented using less than 2,000 gates, as being a lightweight cryptographic primitive, as defined in Section 2.2. Then Trivium, mCRYPTON, DES and PUF implementations using a LFSR cannot be considered lightweight. Furthermore, Trivium's power consumption is significantly outside the required 2-3 µW range.

However, what is not reflected in Figure 5.3, Figure 5.4 and Table 5.1 is that in the case of mCRYPTON, the encryption-only mode with 128-bit keys may be implemented with about 2,000 gates and the full mode with about 2,500 gates using a serialized architecture based on utilizing 5 clock cycles per round [112]. However, the published architecture is based on achieving high throughput, and there remains the possibility to optimize the design further for a minimal hardware footprint. The optimization of mCRYPTON to about 2,000 gates would match the cost of the cipher to a PRESENT-80 implementation. Therefore, mCRYPTON is still considered as a lightweight cryptographic primitive.

5.2.1 Single Crystal Silicon integrated Circuit based Tags

PRESENT-80 appears to be the best choice for meeting the requirements of single crystal IC based RFID tags considering the lightweight cryptographic primitives in Figure 5.3 because of its high throughput, cost of 1000 gates, and a power consumption of only 1.6 µW.

Considering throughput and throughput per unit gate cost mCRYPTON, A2U2 and PRESENT-80 are the best candidates. However, taking into account the high risk of using ciphers that have not been scrutinized adequately by public as well as the level of security offered into consideration, mCRYPTON and A2U2 are the standout selection for consideration. mCRYPTON was one of the 15 candidate proposals considered for AES, with its outstanding throughput, area as well as security metrics. Therefore, if one is to consider the level of security as an additional factor, then mCRYPTON appears to be the best candidate at the present time albeit requiring the development of a serialized architecture implementation. Unfortunately, a power consumption estimate for this cipher has not been found in the public domain.

The ciphers considered clearly stand apart from the hardware optimized version of AES. As a result, it is clear that research efforts into developing new lightweight cryptographic primitives have yielded results far better than that expected from optimizing existing cipher designs.

5.2.2 Printed Semiconductor Tags

While there are a number of possible primitives suitable for single crystal silicon IC based RFID tags, there appears to be no (secure) candidate ciphers yet capable of being implemented with 200-300 gates as required by printed semiconductor tags (see Table 2.1). The most likely candidates are:

- A2U2,
- PUF (in its direct implementation without the use of an LFSR),

- KTANTAN-32,
- WH-16.

The small cost of A2U2 makes it a serious candidate for implementation in printed electronics RFID tags, given that the attacks presented in 4.7 can be fixed.

Thus at the present time, semiconductor tags will need to be based on ultra-lightweight primitives or minimalist primitives. With such methods, hardware requirements are limited to simple operations (such as XOR, AND, etc.) and additional read-write memory.

5.3 Performance Coupling Measure

Being able to evaluate primitives based on multiple characteristics is essential in the successful adoption and selection of lightweight primitives for real applications. Comparing lightweight primitives based on the results available in literature is a complicated task. Furthermore, RFID systems are mostly application dependent, therefore the lightweight security primitive implemented should match the requirements of that particular application performed by the tag. The three main characteristics of a primitive (area, throughput, power) will determine the appropriate choice for a given application.

In the following section, a single measure is introduced. It takes into account the parameters critical to a primitive's classification as being lightweight to enable the evaluation of primitives considering cost, throughput and power. Furthermore, the single measure can empower the user to compare and select primitives based on specific user requirements, as well as evaluate future developments.

5.3.1 Weighted Normalized Cost Power and Throughput (WOOPT) Metric

In order to aggregate the three parameters (cost, power consumption and throughput) into one measure, these measures need to be normalized. The transformation cannot be linear for the reason that reducing area and power or increasing throughput is not a linear process. For an equivalent level of security, reducing the area from 2,500 to 2,000 gates does not require the same effort as reducing

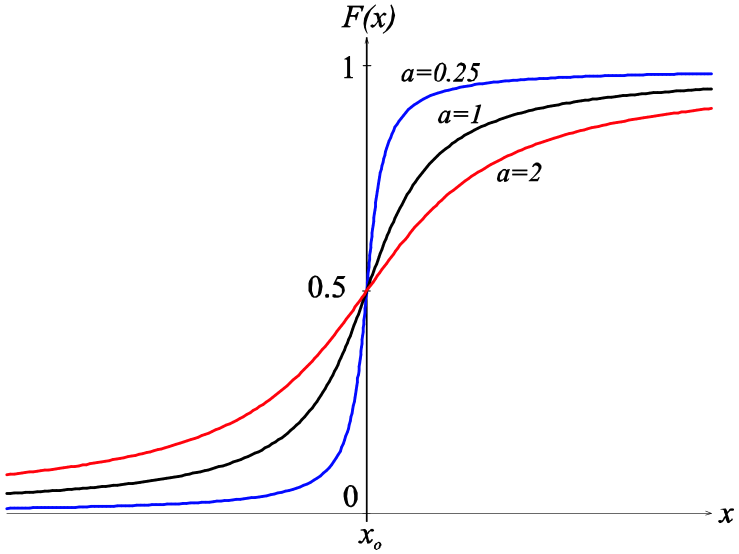


Figure 5.5: Cauchy distribution function.

it from 1,000 to 500 gates. Therefore, we use a Cauchy distribution with the cumulative distribution function given in the following equation and illustrated in Figure 5.5 as our normalizing function.

$$F = \frac{1}{\pi} \arctan\left(\frac{x - x_0}{a}\right) + \frac{1}{2} \tag{5.2}$$

In equation 5.2:

- x is the parameter value (area, throughput, power),
- x_o is the location parameter (the average or targeted value of the parameter)
- a is the scale parameter (the acceptable margin around the targeted value).

When the scale parameter a is reduced, the slope (gradient) of the curve increases and the function becomes more selective by increasing the distance between results that are not close to the target value. The latter is a very useful feature for being able to select a cipher on its ability to meet strict design goals of area, power and throughput.

The three parameters are evaluated using equation 5.2, and then coupled together to form the metric WOOPT given in the following equation where n_A, n_T and n_P are the respective weighting factors for area (F_{Area}), throughput ($F_{Throughput}$) and power (F_{Power}).

$$WOOPT = \frac{\left[n_A \left(1 - F_{Area}\right) + n_T F_{Throughput} + n_P \left(1 - F_{Power}\right)\right]}{3} \tag{5.3}$$

Where:

$$n_A = \begin{cases} \frac{x_{0(A)} - a_A}{x_0} & for \;\; x_{0(A)} \neq a_A \\ 1 & otherwise \end{cases},$$

$$n_T = \begin{cases} \frac{x_{0(T)} - a_T}{x_0} & for \;\; x_{0(T)} \neq a_T \\ 1 & otherwise \end{cases},$$

$$n_P = \begin{cases} \frac{x_{0(P)} - a_P}{x_0} & for \;\; x_{0(P)} \neq a_P \\ 1 & otherwise \end{cases},$$

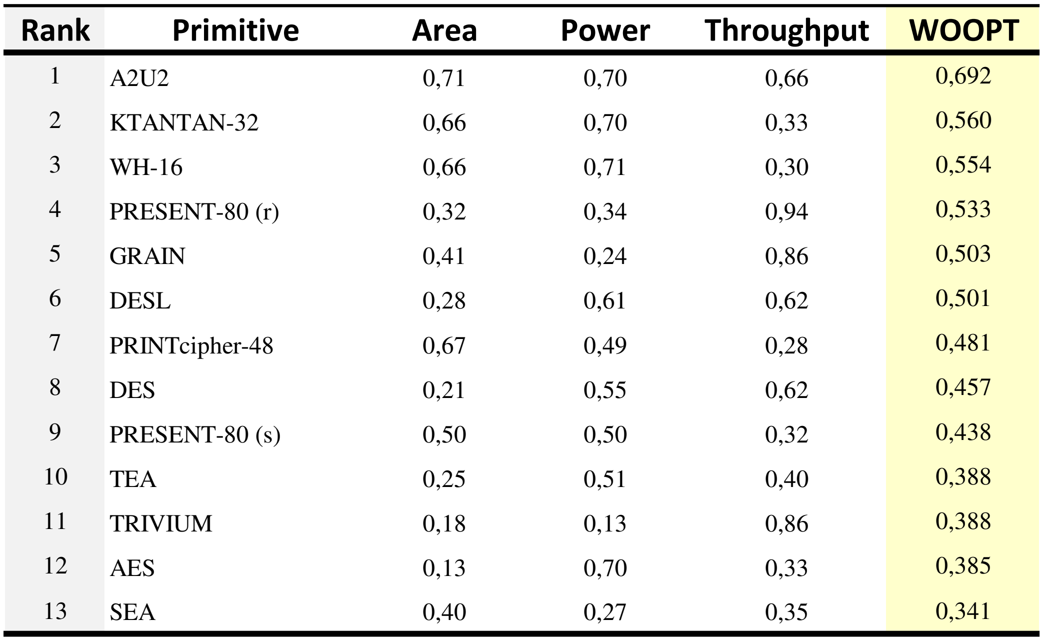
When evaluating lightweight primitives the weights n_A, n_T and n_P are all set to '1' to provide a uniform level of significance to each primitive. The values of x_o and a for the three distribution functions are reported in Table 5.3. More specifically the location parameters were chosen to be a central point along the range of useful gate values (being 0 to 2,000) and useful power consumption values being (0 to 5 µW). However, the location parameter for throughput was set to 32 kbps since the throughput values of the ciphers were evaluated using 100 kHz clock rates and in reality the clock rates are likely to be at least 1 MHz (a 10 fold increase). Given that the highest speed at which a tag transmits to a reader is 640 kbps (based on EPCglobal C1G2 air interface protocol) a 10 fold increase in the clock rate will increase a 32 kbps throughput to 320 kbps. The scale parameter is chosen to cover the range of acceptable values.

Table 5.4 details the WOOPT metric values for the lightweight primitives we have considered. The results of the metric evaluation determined that A2U2 presents the best compromise between cost, power consumption and area from the lightweight primitives we have considered.

Table 5.3: Parameters used in the distribution functions.

	F_{Area} (in GEs)	$F_{Throughput}$ (in Kbps)	F_{Power} (in µW)
Location Parameter (x_0)	1000	32	2.5
Scale Parameter (a)	1000	32	2.5

Table 5.4: Rank based on the WOOPT value for lightweight cryptographic primitives.



Rank	Primitive	Area	Power	Throughput	WOOPT
1	A2U2	0,71	0,70	0,66	0,692
2	KTANTAN-32	0,66	0,70	0,33	0,560
3	WH-16	0,66	0,71	0,30	0,554
4	PRESENT-80 (r)	0,32	0,34	0,94	0,533
5	GRAIN	0,41	0,24	0,86	0,503
6	DESL	0,28	0,61	0,62	0,501
7	PRINTcipher-48	0,67	0,49	0,28	0,481
8	DES	0,21	0,55	0,62	0,457
9	PRESENT-80 (s)	0,50	0,50	0,32	0,438
10	TEA	0,25	0,51	0,40	0,388
11	TRIVIUM	0,18	0,13	0,86	0,388
12	AES	0,13	0,70	0,33	0,385
13	SEA	0,40	0,27	0,35	0,341

Using the design parameters it is now possible to evaluate any future developments in the area of lightweight cryptographic primitives. To the best of the author knowledge, WOOPT measure is the first attempt to classify as well as evaluate lightweight cryptographic primitives (although the security robustness is not taken into account, it is impractical to compare different attacks).

The scale and location parameters defining the weighting factors can be altered according to design needs. This allows RFID practitioners to easily establish the most appropriate selection based on their application specific needs. In order to achieve this goal, we have defined n_A, n_T and n_P as that given in equation 5.3 based on area (A), throughput (T) and power (P) specific location and scale parameters. Here, increasing the scale parameter achieves two goals. First, it controls the gradient around the target value of the function in equation 5.2, and thereby reduces the variability in the goodness value assigned to primitives falling on either side of the target value (in other words control the inter-quartile range). Second, large scale factors ensure that a reduced weighting is assigned to F_A, or F_T or F_P, effectively controlling the influence of that factor on the overall metric result. We illustrate the usefulness of the metric employing the following example.

5.3.2 Primitive Selection for Single Crystal Silicon IC Tags: An Example

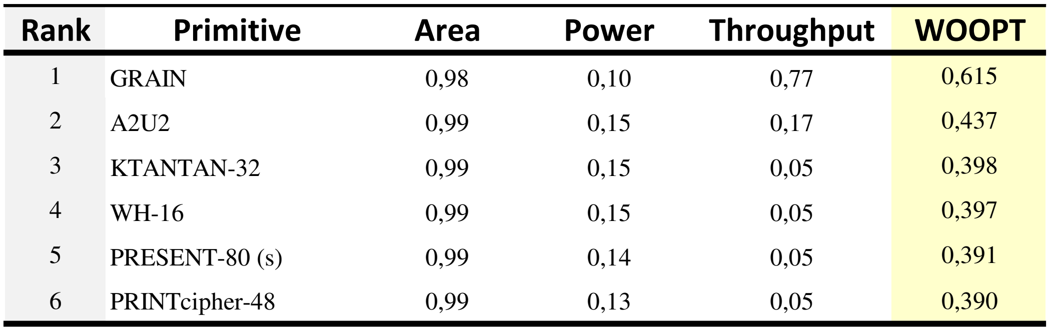
A security primitive is required for single crystal silicon based tags to be used in a contactless card ticketing system using passive RFID technology. The cost of tags requires a primitive that can be implemented with strictly less that 1,500 GE. The tags will be used in near field operation and therefore does not have strict power consumption limitations. However, the throughput must be at the maximum possible rate of 640 kbps. The tags will be clocked using a 1 MHz clock.

Table 5.5: WOOPT metric parameters selected.

	F_{Area} (in GEs)	$F_{Throughput}$ (in Kbps)	F_{Power} (in µW)
Location Parameter (x_0)	1500	64	5.0
Scale Parameter (a)	10	10	4.0

The location parameters can be based on the design requirement limits while the scale parameters are selected to indicate the accepted tolerances within these upper and lower bounds (see Table 5.5). We have selected 64 kbps as the throughput location since this corresponds to a bit rate of 640 kbps when the clock used is 1 MHz. Similarly we have selected 5 µW as the power location parameter and this corresponds to 50 µW. Then, from Table 5.6 it is clear that GRAIN provides the best compromise between power, area and throughput for meeting the desired design goals. The result can be explained by considering the assessment of cost, power consumption and throughput from Table 5.1. Most specifically, the reason that GRAIN has been evaluated as the best parameter is based on the strict requirement of needing 640 kbps throughput. A2U2 appears to be the next best choice with its low implementation cost, decent throughput and low power consumption values. An RFID practitioner can also select a primitive based on the level of security required and offered by the ciphers, by considering the ciphers from the most suitable (ranked 1) to the least suitable (ranked 6).

Table 5.6: WOOPT metric evaluation results.



Rank	Primitive	Area	Power	Throughput	WOOPT
1	GRAIN	0,98	0,10	0,77	0,615
2	A2U2	0,99	0,15	0,17	0,437
3	KTANTAN-32	0,99	0,15	0,05	0,398
4	WH-16	0,99	0,15	0,05	0,397
5	PRESENT-80 (s)	0,99	0,14	0,05	0,391
6	PRINTcipher-48	0,99	0,13	0,05	0,390

5.4 Conclusion

In this chapter, we have developed a classification scheme that is capable of reflecting the developments in the areas of lightweight cryptographic primitive design to support future research directions as well as create a common ground for future discussions. Furthermore, we have illustrated our classification using security primitives from a comprehensive survey of existing methods purported as being suitable for resource constraint devices. We have also compared and contrasted the existing array of solutions to show that novel developments are indeed needed to meet the security challenges posed by low-cost and pervasive computing devices such as RFID tags.

We have normalized the implementation of various cryptographic primitives to provide a common basis of comparison. Furthermore, we have proposed a new metric based on three key parameters (Cost, Power, and Throughput) of lightweight cryptosystems, WOOPT, to ease the comparison between them in the perspective of real life implementation.

The last decade has seen the emergence of printed electronics as a new development that will have a significant impact on the future. The future of RFID will see the traditional silicon integrated circuit used in low cost passive RFID tags with its complex and costly manufacture and antenna interconnect challenges gradually relegated to high-end tags requiring only the utmost operational performance in terms of achievable reading distance, memory features and data capacity, speed of tag anti-collision's arbitration, and overall sophistication. We can expect the resources available on tag to diminish further with the advancement of printed ink technologies for manufacturing RFID tags.

Chapter 6

Conclusions

"Learn from yesterday, live for today, hope for tomorrow."

– *Albert Einstein.*

Security comes in many different flavors. *Low-cost* implies that we find mechanisms that are "good enough" and are deterrents, rather than mechanisms that are impossible to break. For example, ticketing applications will present adversaries greater incentives (learning from the MIFARE experience [45]) and therefore such an application should consider the highest level of security possible for a given cost of a tag, ensure careful implementation and use publicly scrutinized primitives where the weaknesses are well understood.

We have proposed an ultra-lightweight security protocol based on the most simple operations a tag can perform. The design of this protocol was a first attempt to understand the insights of ligthweight cryptography. The experience acquired in its process helped to gain more confidence in designing a more complex cryptosystem, A2U2, based on the combination of several theories on stream ciphers, block ciphers, and (non)linear feedback shift registers. The aim to provide a secure cryptosystem for printed electronics RFID tags was, if not irrealistic, at least very ambitious. By looking at the figures presented in Chapter 4, it looks like we almost succeeded. The cryptanalysis of both proposed cryptographic primitives proves once more the difficulties of providing a secure solution for devices with extreme resource constraints. However, the author do believe that A2U2, by dividing the smallest cost of previous proposed ciphers by (at least) two (see Table 3.1), opens up to new perspectives on the edge of lightweight cryptography.

We have shown that the levels of security provided by (some) lightweight primitives are adequate for RFID applications. However, using these primitives require a careful understanding of their weaknesses and a robust implementation. It is clear from the investigations that attempts to optimize existing standard cryptographic mechanisms (e.g., optimized versions of AES, elliptic curve cryptographic processors, NTRU, DES) have not yielded solutions that will be practically implemented in modern resource constraint platforms such as RFID systems because of their inability to meet the needs of low-cost platforms. Clearly, research efforts in a direction to develop novel primitives have since made considerable progress (e.g., PRESENT, KTANTAN, GRAIN, A2U2) to address the challenges of meeting the new constraints discussed in Chapter 2.

It is evident that there is no universal solution but a collection of solutions suited to different applications based on compromises between level of security, power consumption, cost (area), and performance (throughput). Therefore, the author has proposed a single measure to evaluate lightweight cryptographic primitives called the Weighted nOrmalized cOst, Power and Throughput (WOOPT) metric and demonstrated its usefulness in evaluating lightweight cryptographic primitives in Chapter 5. Furthermore the author has shown how WOOPT can be utilized by practitioners to evaluate lightweight cryptographic primitives that provide the best compromise between the competing design goals of cost, throughput and power consumption.

Existing security mechanisms that can be implemented in 200 gates almost always rely on passwords or a variation on the concept of one-time pads. The author has shown that more robust lightweight primitives, predominantly due to their cost of implementation, are unsuitable for printed ink semiconductor tags and remain a research challenge to be addressed in the future.

Bibliography

[1] European network of excellence for cryptology, ecrypt. http://www.ecrypt.eu.org, 2007-2011.

[2] D. Adams. *H2G2: The Hitchhicker's Guide to the Galaxy*. Pan Books, 1979.

[3] M. Ågren. Some instant-and practical-time related-key attacks on ktantan32/48/64. 2011. `http://eprint.iacr.org/2011/140.pdf`.

[4] M. Ågren and T. Johansson. Linear cryptanalysis of printcipher—trails and samples everywhere. Cryptology ePrint Archive, 2011. `http://eprint.iacr.org/2011/423.pdf`.

[5] M. Albrecht and C. Cid. Algebraic techniques in differential cryptanalysis. In *Fast Software Encryption*, pages 193–208. Springer, 2009.

[6] Martin Albrecht and Carlos Cid. Cold boot key recovery by solving polynomial systems with noise. Cryptology ePrint Archive, Report 2011/038, 2011. `http://eprint.iacr.org/`.

[7] B. Alomair, L. Lazos, and R. Poovendran. Passive attacks on a class of authentication protocols for rfid. In *Proceedings of the 10th international conference on Information security and cryptology*, pages 102–115. Springer-Verlag, 2007.

[8] V.R. Andem. *A cryptanalysis of the tiny encryption algorithm*. PhD thesis, Citeseer, 2003.

[9] A.C. Atici, L. Batina, J. Fan, I. Verbauwhede, and S.B.O. Yalcin. Low-cost implementations of ntru for pervasive security. In *Application-Specific Systems, Architectures and Processors, 2008. ASAP 2008. International Conference on*, pages 79–84. IEEE, 2008.

[10] J.P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia. Quark: a lightweight hash. *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 1–15, 2011.

[11] G. Avoine. Rfid security and privacy lounge. http://www.avoine.net/rfid/index.php.

[12] G. Avoine. Privacy issues in rfid banknote protection schemes. *Smart Card Research and Advanced Applications VI*, pages 33–48, 2004.

[13] G. Avoine, X. Carpent, and B. Martin. Strong authentication and strong integrity (sasi) is not that strong. *Radio Frequency Identification: Security and Privacy Issues*, pages 50–64, 2010.

[14] 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 2010*, pages 176–196, 2010.

[15] C. Berbain, H. Gilbert, and A. Maximov. Cryptanalysis of grain. In *Fast Software Encryption*, pages 15–29. Springer, 2006.

[16] E. Biham and A. Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

[17] A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of a5/1 on a pc. In *Fast Software Encryption*, pages 37–44. Springer, 2001.

[18] T.E. Bjørstad. Cryptanalysis of grain using time/memory/data tradeoffs. eSTREAM submitted papers, 2008. `http://www.ecrypt.eu.org/stream/papersdir/2008/012.pdf`.

[19] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede. Spongent: a lightweight hash function. *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 312–325, 2011.

[20] 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 2007*, pages 450–466, 2007.

[21] A. Bogdanov and LR Knudsen. Small-footprint block cipher design-how far can you go? 2008.

[22] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. Robshaw, and Y. Seurin. Hash functions and rfid tags: Mind the gap. *Cryptographic Hardware and Embedded Systems–CHES 2008*, pages 283–299, 2008.

[23] L. Bolotnyy and G. Robins. Physically unclonable function-based security and privacy in rfid systems. In *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*, pages 211–220. IEEE, 2007.

[24] J. Borghoff, L. Knudsen, and M. Stolpe. Bivium as a mixed-integer linear programming problem. *Cryptography and Coding*, pages 133–152, 2009.

[25] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine. Propagation characteristics and correlation-immunity of highly nonlinear boolean functions. In *Advances in Cryptology, EUROCRYPT 2000*, pages 507–522. Springer, 2000.

[26] Y.C. Chen, W.L. Wang, and M.S. Hwang. Rfid authentication protocol for anti-counterfeiting and privacy protection. In *Advanced Communication Technology, The 9th International Conference on*, volume 1, pages 255–259. IEEE, 2007.

[27] H.Y. Chien. Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. *Dependable and Secure Computing, IEEE Transactions on*, 4(4):337–340, 2007.

[28] Joo Yeon Cho. Linear cryptanalysis of reduced-round present. Cryptology ePrint Archive, Report 2009/397, 2009. `http://eprint.iacr.org/`.

[29] P.H. Cole. Secure data tagging systems, June 8 2004. US Patent App. 20,050/017,844.

[30] P.H. Cole and D.C. Ranasinghe. *Networked RFID systems and lightweight cryptography: raising barriers to product counterfeiting*. Springer-Verlag New York Inc, 2008.

[31] P.H. Cole, L.H. Turner, Z. Hu, and D.C. Ranasinghe. The next generation of rfid technology. *Unique Radio Innovation for the 21st Century*, pages 3–23, 2010.

[32] B. Collard and F.X. Standaert. A statistical saturation attack against the block cipher present. *Topics in Cryptology–CT-RSA 2009*, pages 195–210, 2009.

[33] European Commission. Commission recommendation on the implementation of privacy and data protection principles in applications supported by radio-frequency identification. http://ec.europa.eu/information_society/policy/rfid/documents/recommendationonrfid2009.pdf, 2009.

[34] D. Coppersmith, H. Krawczyk, and Y. Mansour. The shrinking generator. In *Advances in Cryptology, Crypto 93*, pages 22–39. Springer, 1994.

[35] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. *Advances in Cryptology, EUROCRYPT 2003*, pages 644–644, 2003.

[36] J. Daemen and V. Rijmen. *The design of Rijndael: AES–the advanced encryption standard*. Springer Verlag, 2002.

[37] Z.D. Dai and J.H. Yang. Linear complexity of periodically repeated random sequences. In *Advances in Cryptology, EUROCRYPT 91*, pages 168–175. Springer, 1991.

[38] Paolo D'Arco and Alfredo De Santis. From weaknesses to secret disclosure in a recent ultra-lightweight rfid authentication protocol. Cryptology ePrint Archive, Report 2008/470, 2008. `http://eprint.iacr.org/`.

[39] R. Das and P. Harrop. *Printed, organic & flexible electronics forecasts, players & opportunities 2009-2029*. IdTechEx, 2009.

[40] R. Das and P. Harrop. Rfid forecasts, players and opportunities 2011-2021. *IDTechEx report*, 2010.

[41] M. David and N.R. Prasad. Providing strong security and high privacy in low-cost rfid networks. *Security and Privacy in Mobile Information and Communication Systems*, pages 172–179, 2009.

[42] M. David, D.C. Ranasinghe, and T. Larsen. A2u2: A stream cipher for printed electronics rfid tags. In *RFID (RFID), 2011 IEEE International Conference on*, pages 176–183. IEEE, 2011.

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

[44] C. De Canniere, O. Dunkelman, and M. Knežević. Katan and ktantan: a family of small and efficient hardware-oriented block ciphers. *Cryptographic Hardware and Embedded Systems, CHES 2009*, pages 272–288, 2009.

[45] G. de Koning Gans, J.H. Hoepman, and F. Garcia. A practical attack on the mifare classic. *Smart Card Research and Advanced Applications*, pages 267–282, 2008.

[46] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal. Design and implementation of puf-based unclonable rfid ics for anti-counterfeiting and security applications. In *RFID, 2008 IEEE International Conference on*, pages 58–64. IEEE, 2008.

[47] I. Dinur and A. Shamir. Breaking grain-128 with dynamic cube attacks. In *Fast Software Encryption*, pages 167–187. Springer, 2011.

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

[49] Selected block cipher listing, ecrypt website. http://www.ecrypt.eu.org/lightweight/index.php/Block_ciphers.

[50] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Design and Test*, pages 522–533, 2007.

[51] EPC Global. *EPC Class 1 Generation 2 UHF Air Interface Protocol Standard "Gen 2"*, 2008.

[52] The estream project. http://www.ecrypt.eu.org/stream/endofphase3.html, 2008.

[53] M. Feldhofer. Comparison of low-power implementations of trivium and grain. In *State of the Art of Stream Ciphers Workshop (SASC 2007), eSTREAM, ECRYPT Stream Cipher Project, Report*, volume 27, page 2007, 2007.

[54] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for rfid systems using the aes algorithm. *Cryptographic Hardware and Embedded Systems, CHES 2004*, pages 85–140, 2004.

[55] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. Aes implementation on a grain of sand. 2005.

[56] N. Ferguson and B. Schneier. *Practical cryptography*, volume 141. Wiley New York, 2003.

[57] K. Finkenzeller and D. Müller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. Wiley, 2010.

[58] K.P. Fishkin and S. Roy. Enhancing rfid privacy via antenna energy analysis. In *RFID Privacy Workshop*, 2003.

[59] B. Gammel, R. Göttfert, and O. Kniffler. The achterbahn stream cipher. *Submission to eSTREAM*, 2005.

[60] B.M. Gammel and R. Göttfert. Combining certain nonlinear feedback shift registers. In *Workshop Record of SASC–The State of the Art of Stream Ciphers*, pages 234–248. Citeseer.

[61] S.L. Garfinkel, A. Juels, and R. Pappu. Rfid privacy: An overview of problems and proposed solutions. *Security & Privacy, IEEE*, 3(3):34–43, 2005.

[62] 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*, pages 148–160. ACM, 2002.

[63] B. Gassend, D. Lim, D. Clarke, M. Van Dijk, and S. Devadas. Identification and authentication of integrated circuits. *Concurrency and Computation: Practice & Experience*, 16(11):1077–1098, 2004.

[64] B.L.P. Gassend. *Physical random functions*. PhD thesis, Citeseer, 2003.

[65] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. *Advances in Cryptology, CRYPTO'97*, pages 112–131, 1997.

[66] T. Good and M. Benaissa. Hardware results for selected stream cipher candidates. *State of the Art of Stream Ciphers*, pages 191–204, 2007.

[67] F. Gosset, F.X. Standaert, and J.J. Quisquater. Fpga implementation of squash. In *Proceedings of the 29th Symposium on Information Theory in the Benelux*, 2008.

[68] G.S. GOST. 28147-89. *Cryptographic protection for data processing systems, Government Committee of the USSR for Standards*, 1989.

[69] J. Guo, T. Peyrin, and A. Poschmann. The photon family of lightweight hash functions. 2011.

[70] P. Gutmann, D. Naccache, and C.C. Palmer. When hashes collide [applied cryptography]. *Security & Privacy, IEEE*, 3(3):68–71, 2005.

[71] Z. Haina and W. Xiaoyun. Cryptanalysis of stream cipher grain family. Cryptology ePrint Archive, Report 2009/109, 2009. `http://eprint.iacr.org/`.

[72] P. Hamalainen, T. Alho, M. Hannikainen, and T.D. Hamalainen. Design and implementation of low-area and low-power aes encryption hardware core. In *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on*, pages 577–583. IEEE, 2006.

[73] R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[74] G.P. Hancke and M.G. Kuhn. An rfid distance bounding protocol. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 67–73. IEEE, 2005.

[75] H. Handschuh and B. Preneel. Key-recovery attacks on universal hash function based mac algorithms. *Advances in Cryptology–CRYPTO 2008*, pages 144–161, 2008.

[76] P. Harrop. Near field uhf vs. hf for item level tagging. *IDTechEx article, available at http://www.eurotag.org*.

[77] P. Harrop and R. Das. Printed and chipless rfid forecasts, technologies & players 2011-2021. *IDTechEx report*, 2010.

[78] P. Harrop and G. Holland. Item level rfid 2008-2018. *IDTechEx report*, 2008.

[79] J. Hernandez-Castro, P. Peris-Lopez, R. Phan, and J. Tapiador. Cryptanalysis of the david-prasad rfid ultralightweight authentication protocol. *Radio Frequency Identification: Security and Privacy Issues*, pages 22–34, 2010.

[80] J.C. Hernandez-Castro, J.M.E. Tapiador, P. Peris-Lopez, and J.J. Quisquater. Cryptanalysis of the sasi ultralightweight rfid authentication protocol with modular rotations. *Arxiv preprint arXiv:0811.4257*, 2008.

[81] J. Hoffstein, J. Pipher, and J. Silverman. Ntru: A ring-based public key cryptosystem. *Algorithmic number theory*, pages 267–288, 1998.

[82] 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*. Citeseer, 2007.

[83] ISO/IEC 14443-2 Standard. *Identification cards - Contactless integrated circuit cards - Proximity cards - Part 2: Radio frequency power and signal interface*, 2010.

[84] P. Israsena. Securing ubiquitous and low-cost rfid using tiny encryption algorithm. In *Wireless Pervasive Computing, 2006 1st International Symposium on*, pages 4–pp. IEEE, 2006.

[85] P. Israsena and S. Wongnamkum. Hardware implementation of a tea-based lightweight encryption for rfid security. *RFID Security*, pages 417–433, 2009.

[86] RFID Journal. Frequently asked questions, 2011. `http://www.rfidjournal.com/faq/20`.

[87] A. Juels. Minimalist cryptography for low-cost rfid tags. *Security in Communication Networks*, pages 149–164, 2005.

[88] A. Juels and R. Pappu. Squealing euros: Privacy protection in rfid-enabled banknotes. In *Financial Cryptography*, pages 103–121. Springer, 2003.

[89] 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 communications security*, pages 103–111. ACM, 2003.

[90] A. Juels and S.A. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology–CRYPTO 2005*, pages 293–308. Springer, 2005.

[91] J.P. Kaps, G. Gaubatz, and B. Sunar. Cryptography on a speck of dust. *Computer*, 40(2):38–44, 2007.

[92] F. Karakoç, H. Demirci, and A.E. Harmancı. Combined differential and linear cryptanalysis of reduced-round printcipher. In *Selected Areas in Cryptography SAC*, 2011.

[93] K. Khoo and G. Gong. New constructions for resilient and highly nonlinear boolean functions. In *Information Security and Privacy*, pages 218–218. Springer, 2003.

[94] I.J. Kim, E.Y. Choi, and D.H. Lee. Secure mobile rfid system against privacy and security problems. 2007.

[95] S.J. Kim, Y.S. Kim, and S.C. Park. Rfid security protocol by lightweight ecc algorithm. In *Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on*, pages 323–328. IEEE, 2007.

[96] L. Knudsen, G. Leander, A. Poschmann, and M. Robshaw. Printcipher: a block cipher for ic-printing. *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 16–32, 2011.

[97] Y. Ko, S. Hong, W. Lee, S. Lee, and J.S. Kang. Related key differential attacks on 27 rounds of xtea and full-round gost. In *Fast Software Encryption*, pages 299–316. Springer, 2004.

[98] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[99] D. Koch, M. Körber, and J. Teich. Searching rc5-keys with distributed reconfigurable computing. In *Proceedings of International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA 06)*, pages 42–48. Citeseer, 2006.

[100] T. Kohno. An interview with rfid security expert ari juels. *Pervasive Computing, IEEE*, 7(1):10–11, 2008.

[101] O. Kömmerling and M.G. Kuhn. Design principles for tamper-resistant smartcard processors. In *Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*, pages 2–2. USENIX Association, 1999.

[102] O. Kucuk. Slide resynchronization attack on the initialization of grain 1.0, estream report 2006/044 (2006). *URL: http://www.ecrypt.eu.org/stream/papers.html*, 4.

[103] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, A. Rupp, and M. Schimmler. How to break des for 8,980 euro. In *2nd Workshop on Special-purpose Hardware for Attacking Cryptographic Systems-SHARCS*, pages 3–4, 2006.

[104] G. Leander, M.A. Abdelraheem, H. AlKhzaimi, and E. Zenner. A cryptanalysis of printcipher: The invariant subspace attack.

[105] G. Leander, C. Paar, A. Poschmann, and K. Schramm. New lightweight des variants. In *Fast Software Encryption*, pages 196–210. Springer, 2007.

[106] G. Leander and A. Poschmann. On the classification of 4 bit s-boxes. *Arithmetic of Finite Fields*, pages 159–176, 2007.

[107] H. Lee, E.Y. Choi, S.M. Lee, and D.H. Lee. Trapdoor-based mutual authentication scheme without cryptographic primitives in rfid tags. 2007.

[108] K.S. Lee, J.H. Chun, and K.W. Kwon. A low power cmos compatible embedded eeprom for passive rfid tag. *Microelectronics Journal*, 41(10):662–668, 2010.

[109] T. Li and R. Deng. Vulnerability analysis of emap-an efficient rfid mutual authentication protocol. 2007.

[110] T. Li and G. Wang. Security analysis of two ultra-lightweight rfid authentication protocols. *New Approaches for Security, Privacy and Trust in Complex Environments*, pages 109–120, 2007.

[111] C. Lim. A revised version of crypton: Crypton v1. 0. In *Fast Software Encryption*, pages 31–45. Springer, 1999.

[112] C. Lim and T. Korkishko. mcrypton–a lightweight block cipher for security of low-cost rfid tags and sensors. *Information Security Applications*, pages 243–258, 2006.

[113] C.H. Lim. Crypton: A new 128-bit block cipher-specification and analysis. 1998.

[114] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 13(10):1200–1205, 2005.

[115] F. Mace, F.X. Standaert, and J.J. Quisquater. Asic implementations of the block cipher sea for constrained applications. In *Proceedings of the Third International Conference on RFID Security-RFIDSec*, pages 103–114. Citeseer, 2007.

[116] R. Maes, P. Tuyls, and I. Verbauwhede. Intrinsic pufs from flip-flops on reconfigurable devices. In *3rd Benelux Workshop on Information and System Security (WISSec 2008)*, 2008.

[117] R. Maes, P. Tuyls, and I. Verbauwhede. Low-overhead implementation of a soft decision helper data algorithm for sram pufs. *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 332–347, 2009.

[118] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Lightweight secure pufs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 670–673. IEEE Press, 2008.

[119] Dakhilalian M. Mala, H. and M. Shakiba. Cryptanalysis of mcrypton - a lightweight block cipher for security of rfid tags and sensors. *International Journal of Communication Systems*, 2011.

[120] H. Martin, E.S. Millan, L. Entrena, J.C.H. Castro, and P.P. Lopez. Akari-x: A pseudorandom number generator for secure lightweight systems. In *On-Line Testing Symposium (IOLTS), 2011 IEEE 17th International*, pages 228–233. IEEE.

[121] A. Maximov and A. Biryukov. Two trivial attacks on trivium. In *Proceedings of the 14th international conference on Selected areas in cryptography*, pages 36–55. Springer-Verlag, 2007.

[122] W. Meier and O. Staffelbach. Fast correlation attacks on stream ciphers. In *Advances in Cryptology, EUROCRYPT 88*, pages 301–314. Springer, 1988.

[123] W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology, EUROCRYPT 89*, pages 549–562. Springer, 1990.

[124] V. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology-CRYPTO'85 Proceedings*, pages 417–426. Springer, 1986.

[125] R.M. Needham and D.J. Wheeler. Tea extensions. *University of Cambridge, Cambridge, UK, Tech. Rep*, 1997.

[126] New european schemes for signatures, integrity, and encryption (nessie) project website. https://www.cosic.esat.kuleuven.be/nessie, 2004.

[127] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar. New constructions of resilient and correlation immune boolean functions achieving upper bound on nonlinearity. *Electronic Notes in Discrete Mathematics*, 6:158–167, 2001.

[128] P. Peris López. Lightweight cryptography in radio frequency identification (rfid) systems. 2008.

[129] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. Emap: An efficient mutual-authentication protocol for low-cost rfid tags. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 352–361. Springer, 2006.

[130] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. M 2 ap: A minimalist mutual-authentication protocol for low-cost rfid tags. *Ubiquitous Intelligence and Computing*, pages 912–923, 2006.

[131] P. Peris-Lopez, J. Hernandez-Castro, J. Tapiador, and A. Ribagorda. Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol. *Information Security Applications*, pages 56–68, 2009.

[132] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda. Lmap: A real lightweight mutual authentication protocol for low-cost rfid tags. In *Proc. of 2nd Workshop on RFID Security*. Citeseer, 2006.

[133] T. Phillips, T. Karygiannis, and R. Kuhn. Security standards for the rfid market. *Security & Privacy, IEEE*, 3(6):85–89, 2005.

[134] Guang Gong Qi Chai, Xinxin Fan. An ultra-efficient key recovery attack on the lightweight stream cipher a2u2. Cryptology ePrint Archive, Report 2011/247, 2011. `http://eprint.iacr.org/`.

[135] H. Raddum. Cryptanalytic results on trivium. *eSTREAM, ECRYPT Stream Cipher Project, Report*, 39:2006, 2006.

[136] N. Rama and R. Suganya. Ssl-map: A more secure gossamer-based mutual authentication protocol for passive rfid tags. *International Journal on Computer Science and Engineering*, 2:363–367, 2010.

[137] D. Ranasinghe, D. Engels, and P. Cole. Low-cost rfid systems: Confronting security and privacy. In *Auto-ID labs research workshop*, pages 54–77. Citeseer, 2004.

[138] D.C. Ranasinghe and P.H. Cole. Confronting security and privacy threats in modern rfid systems. In *Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on*, pages 2058–2064. IEEE, 2006.

[139] D.C. Ranasinghe, S. Devadas, and P.H. Cole. A low cost solution to cloning and authentication based on a lightweight primitive. *Networked RFID Systems and Lightweight Cryptography*, pages 289–309, 2008.

[140] D.C. Ranasinghe, D.W. Engels, and P.H. Cole. Security and privacy solutions for low-cost rfid systems. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004*, pages 337–342. IEEE, 2004.

[141] M.R. Rieback, B. Crispo, and A.S. Tanenbaum. Rfid guardian: A battery-powered mobile device for rfid privacy management. In *Information Security and Privacy*, pages 184–194. Springer, 2005.

[142] Shamir A. Rivest, R.L. and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[143] P. Rogaway. Nonce-based symmetric encryption. In *Fast Software Encryption*, pages 348–359. Springer, 2004.

[144] C. Rolfes, A. Poschmann, and C. Paar. Security for 1000 gate equivalents. *Secure Component and System Identification (SECSI)*, 2008.

[145] R. Rueppel. Linear complexity and random sequences. In *Advances in Cryptology, EUROCRYPT 85*, pages 167–188. Springer, 1986.

[146] R.A. Rueppel. New approaches to stream ciphers. *Diss. ETH*.

[147] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber. Modeling attacks on physical unclonable functions. In *ACM CCS*, volume 2010, 2010.

[148] A. Rukhin. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, DTIC Document, 2001.

[149] P. Sarkar and S. Maitra. Construction of nonlinear boolean functions with important cryptographic properties. In *Advances in Cryptology, EUROCRYPT 2000*, pages 485–506. Springer, 2000.

[150] B. Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *Fast Software Encryption*, pages 191–204. Springer, 1994.

[151] B. Schneier. A self-study course in block-cipher cryptanalysis. *Cryptologia*, 24(1):18–33, 2000.

[152] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. *The Twofish encryption algorithm: a 128-bit block cipher*. John Wiley & Sons, Inc., 1999.

[153] A. Shamir. Stream ciphers: Dead or alive? In *Advances in Cryptology, ASIACRYPT*, volume 1, 2004.

[154] A. Shamir. Squash–a new mac with provable security properties for highly constrained devices such as rfid tags. In *Fast Software Encryption*, pages 144–157. Springer, 2008.

[155] C.E. Shannon. *Communication theory of secrecy systems*. AT & T, 1949.

[156] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai. Piccolo: an ultra-lightweight blockcipher. *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 342–357, 2011.

[157] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.). *Information Theory, IEEE Transactions on*, 30(5):776–780, 1984.

[158] F.X. Standaert, G. Piret, N. Gershenfeld, and J.J. Quisquater. Sea: A scalable encryption algorithm for small embedded applications. *Smart Card Research and Advanced Applications*, pages 222–236, 2006.

[159] M. Steil. 17 mistakes microsoft made in the xbox security system. In *22nd Chaos Communication Congress*, pages 378–390, 2005.

[160] D.R. Stinson. *Cryptography: theory and practice*. CRC press, 2006.

[161] G.E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference*, pages 9–14. ACM, 2007.

[162] J. Szewczykowski. Method for identifying counterfeit negotiable instruments, October 6 1998. US Patent 5,818,021.

[163] Ubisec&sens project website. http://www.ist-ubisecsens.org, 2006.

[164] M. Vielhaber. Breaking one. fivium by aida an algebraic iv differential attack. *eprint. iacr. org/2007/413. pdf*.

[165] M. Wang. Differential cryptanalysis of reduced-round present. In *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, pages 40–49. Springer-Verlag, 2008.

[166] L. Wei, C. Rechberger, J. Guo, H. Wu, H. Wang, and S. Ling. Improved meet-in-the-middle cryptanalysis of ktantan. Cryptology ePrint Archive, 2011. `http://eprint.iacr.org/2011/201.pdf`.

[167] S. Weingart. Physical security devices for computer subsystems: A survey of attacks and defenses. In *Cryptographic Hardware and Embedded Systems, CHES 2000*, pages 45–68. Springer, 2000.

[168] N.H.E. Weste and D. Harris. Cmos vlsi design: a circuits and systems perspective| macquarie university researchonline. 2005.

[169] D. Wheeler and R. Needham. Tea, a tiny encryption algorithm. In *Fast Software Encryption*, pages 363–366. Springer, 1995.

[170] W. Wu and L. Zhang. Lblock: A lightweight block cipher. In *Applied Cryptography and Network Security*, pages 327–344. Springer, 2011.

[171] E. Yarrkov. Cryptanalysis of xxtea. 2010.

[172] M.D. Yu and S. Devadas. Secure and robust error correction for physical unclonable functions. *Design & Test of Computers, IEEE*, 27(1):48–65, 2010.

[173] S. Yu, K. Ren, and W. Lou. A privacy-preserving lightweight authentication protocol for low-cost rfid tags. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7. IEEE, 2007.

[174] K. Yüksel. *Universal hashing for ultra-low-power cryptographic hardware applications*. PhD thesis, Citeseer, 2004.

[175] S. Zhilyaev. Evaluating a new mac for current and next generation rfid, 2010.

[176] H. Zhu and F. Bao. Securing rfid tags: Authentication protocols with completeness, soundness, and non-traceability. In *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pages 2698–2702. IEEE, 2007.

List of Figures

List of Tables

Index

Appendix A

Detailed Cryptanalysis of A2U2

Foreword

This chapter is based on the conference paper *Cryptanalysis of the lightweight cipher A2U2*, written in collaboration with M. A. Abdelraheem, J. Borghoff and E. Zenner, presented during the *Thirteen IMA International Conference on Cryptography and Coding*, in Oxford, UK, in December 2011. It presents the details of the cryptanalysis of A2U2, which is summarized in Section 4.7. The cryptanalysis has not been performed by the author of this thesis, therefore this material is presented in the appendix for completness purpose only.

A.1 A Chosen Plaintext Attack

A.1.1 Disproving the Chai/Fan/Gong Attack

In [134], a very efficient chosen plaintext attack against A2U2 is proposed. However, as we are going to show in this section, the attack contains a flaw that makes it unapplicable against the real A2U2 cipher.

Attack Idea: If the attacker could freely choose one plaintext bit for each clock, then he can write the output equation as follows:

$$\begin{aligned} Y_t &= \mathrm{MUX}_{A_t}(B_t + C_t, B_t + p_t) \\ &= \mathrm{MUX}_{A_t}(C_t, p_t) + B_t. \end{aligned}$$

Depending on the amount of knowledge the attacker has about the plaintext, he can now learn more about the inner state. If the attacker can choose the plaintext, he can start by encrypting a plaintext that is identical to the counter sequence. In this case, the above equation simplies to

$$\begin{aligned} Y_t &= \mathrm{MUX}_{A_t}(C_t, p_t) + B_t \\ &= p_t + B_t, \end{aligned}$$

meaning that the attacker can learn the whole sequence $(B_t)_{t\geq 0}$. Next, he encrypts a plaintext that is the bitwise inverse of the counter sequence. This allows him to distinguish for every ciphertext bit whether C_t or p_t was encrypted, providing the attacker with the full sequence $(A_t)_{t\geq 0}$. Now he has the sequences produced by the LFSR and by both NFSRs. All that remains is to test for each round which key bit gives the correct NFSR update. This can be done in unit time, yielding an extremely efficient attack.

The Catch: However, the initial assumption of the above attack is wrong, invalidating the whole cryptanalysis. The problem is that plaintext is not used at a rate of 1 bit per round. It is not possible to choose a plaintext bit for each round, because (1) some plaintext bits are used in several rounds and (2) without knowledge of the sequence $(A_t)_{t\geq 0}$, it is impossible to say in which rounds a plaintext bit will be used. Since the first version of this paper, this problem was acknowledged by Chai et al., and their paper was updated accordingly. Their new attack recovers first the sequence $(A_t)_{t\geq 0}$ by choosing two complementary plaintexts (p_t and its complement $\bar{p_t}$) and xoring their corresponding ciphertexts ($c \oplus \bar{c} = \Delta c$). Now, if $\Delta c_t = 0$ then $A_t = 0$, otherwise $A_t = 1$. This recovers the sequence $(A_t)_{t\geq 0}$ and consequently the sequence $(B_t)_{t\geq 0}$ can then be simply recovered.

A.1.2 A Leak in the Output Function

However, as it turns out, this attack can be repaired. In the following, we will demonstrate a leak in the output function that can even be used for general known-plaintext attacks and will then expand this weakness into a chosen plaintext attack that reminds of the one described above but is actually functional.

*Known plaintext:*Assume that the inner sequences A_t and B_t are statistically close to random. Then in particular, $\Pr(A_t = 0) = 1/2$ for all t. We can now consider two cases for the output function:

- If $A_t = 0$, then $Y_t = B_t + C_t$. Since we know C_t, we can rewrite this as $B_t = Y_t + C_t$. For $A_t = 0$, this is always true.
- If $A_t = 1$, we have $Y_t = B_t + P_{\sigma(t)}$, with $P_{\sigma(t)}$ unknown. If we assume that $P_{\sigma(t)} = C_t$ with probability 1/2,[1] then the equation $B_t = Y_t + C_t$ is also true with probability 1/2.

In total, the equation $B_t = Y_t + C_t$ is thus met with probability $1/2 + (1/2)^2 = 3/4$, i.e. by observing the keystream and knowing the behaviour of the counter LFSR, we can predict the inner stream $(B_0, B_1 \ldots)$ with probability $3/4$ per bit.

*Chosen plaintext:*Note that when we can choose the plaintext, we can increase the probability of $P_{\sigma(t)} = C_t$ and thus the probability of the equation $B_t = Y_t{+}C_t$ being correct. As an example, consider the first 5 output bits of the LFSR, which are $(1, 0, 0, 0, 0)$. Thus, if we choose a plaintext $(1, 0, 0, 0, 0)$, then $P_{\sigma(t)} = C_t$ is true with probability 1 for the first bit, $1-1/2$ for the second, $1-1/4$ for the third, $1-1/8$ for the fourth, and $1-1/16$ for the fifth bit. Thus, we can predict the inner state bits $(B_0, \ldots, B_4)$ with probabilities $(1, 3/4, 7/8, 15/16, 31/32)$.

A.1.3 The Attack

The most useful plaintexts for this kind of analysis seem to be $(0, 0, ...)$ and $(1, 1, ...)$, since for them, the attacker knows exactly the bit $P_{\sigma(t)}$ for every time slot t. Let us start with the all-zero sequence. The attacker knows that the plaintext sequence $(P_{\sigma(t)})_{t\geq 0}$ consists only of zeros. He now looks at all time slots t with $C_t = 0$. For those time slots, it holds that $B_t = Y_t$, independent of the choice of A_t. Thus, he learns about half of the bits of the sequence B. The remaining bits can be learned using the all-one sequence. In this case, in all positions where $C_t = 1$, the attacker learns $B_t = Y_t + 1$, also independent of A_t. Thus, he has fully reconstructed the sequence B. What is more, he can also use this new information to learn the sequence A as well. For every time slot, he picks the ciphertext bit Y_t corresponding to the plaintext bit $P_{\sigma(t)} \neq C_t$.

[1] Note that if the probability for $P_{\sigma(t)} = C_t$ significantly differs from 1/2, then the success probabilities for the rest of the attack are even better than claimed.

If it holds that $B_t = Y_t + C_t$, then $A_t = 0$, otherwise $A_t = 1$. After this step, the attacker knows the sequences generated by all three registers A, B, and C. The remaining attack proceeds as follows. Knowing the sequences A, B and C the attacker can determine the values h_t because

$$h_t = B_{t-9} + B_{t-8}B_{t-7} + B_{t-6} + B_{t-3} + A_t + 1.$$

Furthermore, it holds that

$$h_t = \mathrm{MUX}_{C_{t-5}}(S_0^t, S_1^t) \cdot \mathrm{MUX}_{C_{t-1}}(S_4^t, A_{t-2}) + \mathrm{MUX}_{C_{t-3}}(S_2^t, S_3^t) + 1,$$

where C_{t-i} for $i = 1, 3, 5$ and A_{t-2} are known. This equation is at most quadratic and in about half of the cases (when $C_{t-1} = 1$) it is linear. Determining 56 values of h_t yields a fully determined quadratic Boolean equation system, which can be solved by e.g., using Gröbner basis techniques. As about half of the equation are linear, a linear equation system can be obtained after determining 112 values of h_t. After 11 clockings of the algorithm the key register is rotated once and the key bits are reused, thus it can happen that the same equation is generated twice. However, experiments showed that this does not happen frequently, thus we expect that observing around 120 values of h_t is sufficient to generate a fully determined linear equation systems in 56 unknowns.

The Effort: The attack requires two chosen plaintexts of length around 60 bits which are encrypted using the same key and initialization vector in order to recover secret key bits (excluding the 5 bits which are used to initialize the counter). Note that a 60 bit plaintext corresponds to a ciphertext of length 120 bit on average. The main computational effort consists in solving a linear Boolean equation system which can be done in well under a 1 second. Thus, in the chosen plaintext scenario, the cipher must be considered as completely broken.

A.2 Guess-and-Determine attack

In this section we discuss a known plaintext attack which is in general a more likely scenario than a chosen plaintext attack. When we know but are not allowed to choose the plaintext we cannot use the same trick as in Section A.1 to determine the sequence $(B_t)_{t\geq 0}$. We cannot simply calculate this sequence for a given plaintext/ciphertext pair because we do not know which bits of the ciphertext correspond to the plaintext bits. This is controlled by register A.

The idea of this attack is to guess the sequence of $(A_t)_{t\geq 0}$, meaning we guess at which positions of the ciphertext a plaintext bit was used. These guesses are used to determine additional bits of register B and then later on the value of h_t. As we know the value of the counter at any time during the encryption process, given h_t and A_{t-2} we obtain a Boolean equation in the key bits which is at most quadratic and contains at most three variables. If we are able to collect sufficiently many of those equations we will be able to recover the key bits by solving the equation system.

We denote by A_0 the content of the last cell of the first NFSR at the time when the ciphertext generation starts.The attack is divided into three parts of guessing bits.

In the first part we guess the value A_t for $t = 0, \ldots, 8$ for 9 consecutive clockings of the algorithms. Depending on our guess we know if the counter bit or a plaintext bit was used to generate the corresponding ciphertext bit and we can determine the value of B_t for $t = 0, \ldots, 8$. After guessing 9 bits we know the full second NFSR and about the lower half of the first NFSR.

In the next part we continue guessing the value of A_t for $t = 9, \ldots, 16$ and determine the value of B_t for $t = 9, \ldots, 16$. Additionally, we obtain the value of h_t for $t = 9, \ldots, 16$ and the corresponding Boolean equation in the key bits, because it holds that

$$h_t := A_t + B_{t-9} + B_{t-8}B_{t-7} + B_{t-6} + B_{t-3} + 1, \qquad \text{(A.1)}$$

the full register B is known and we guessed the value of A_t. After the second part of the attack both registers are known and we have already obtained 8 equations.

In the third part we want to determine the value of h_t for further clockings of the cipher. The full register A is known and in order to update register B only bits of register A are used. This means we can update register B and know the value which was use to encrypt next ciphertext bit (bit 17, 18 etc). Furthermore, we know the counter value C_t and the plaintext bit p that might have been used (according to our guess). As mentioned before we want to determine the value of h_t and obtain the corresponding equation. Using equation (A.1) we need to determine A_t in order to obtain h_t. Depending on the values of counter C_t and the value of the plaintext bit $P_{\sigma(t)}$ we can either calculate the value of A_t

or we have to guess it. The output generation can be presented as a quadratic equation

$$Y_t + B_t + C_t = A_t(C_t + P_{\sigma(t)}). \tag{A.2}$$

This means if $C_t \neq P_{\sigma(t)}$ and thus $C_t + P_{\sigma(t)} = 1$ we can simply use Equation (A.2) to determine the value of A_t. However, if $C_t = P_{\sigma(t)}$ and thus $C_t + P_{\sigma(t)} = 0$, Equation (A.2) does not yield any information about A_t and we have to guess the value of A_t as before.

The Effort: We need to collect at least 56 equations to determine a unique solution in 56 key variables. In the first two parts of the attack we guess 17 bits and obtain 8 equations. We expect that we have to guess every second value for A_t in the third part of the attack. Thus, we expect that we have to guess at least 24 further bits in the third part of the attack in order to obtain a fully determined equation system. This leads to a complexity of at least 2^{41}.

There are two factors that increase the attack complexity. Firstly, the equation system is non-linear and therefore it might not have a unique solution even though it is fully determined. However, it is often sufficient to add a few extra equations to get a unique solution. This will slightly raise the complexity of the attack. Secondly, the key register is rotated once after 11 clockings of the algorithm and thus key bits are reused. This property will on the one hand increase the complexity of the attack for the correct guess but on the other hand enable us to discard wrong guesses in an early stage. After producing 11 ciphertext bits the key register has been rotated once, that means the key bits will be reused when generating more equations. This leads to rounds where we guess or determine the value of A_t but do not get a new equation, thus do not gain extra information about the key. This is especially true for the correct guess and means that it is necessary to guess extra bits in order to obtain a fully determined system. For a wrong guess however this might be to our advantage because it is very likely that when the same polynomial is generated the RHS differs. Thus, we get contradicting equations and can abort the guess.

In general, for a wrong guess the equation system will not have a solution. The inconsistency might be very obvious as mentioned above, but it might also be necessary to solve a non-linear Boolean equation system. Therefore, we have to make a trade-off how often we want to check if the system is still solvable.

An implementation of the attack is necessary in order to provide a better estimate of the attack complexity. Simulations showed that after guessing 47 bits we obtain a set with 57 equations on average. When testing these equation systems for solvability around 5% have a non-empty solution set. This means 5% of our guesses survive. Guessing 6 additional bits yields equation systems containing 70.7 equations on average and we expect that only the correct guess or very few guesses survive and that the equation system corresponding to the right guess will have a unique solution. The estimated complexity for this approach is 2^{49} bit guesses. As we in the worst case have to solve an non-linear equation system for each guess we cannot ignore the costs for this step. The costs for solving a non-linear equation system by for example using Gröbner bases are hard to estimate as the problem of solving a random non-linear equation system is NP-hard and thus the running time is equivalent to a brute force search in the worst case. However, we deal with fairly small equation systems and experiments indicate that these equation systems are solvable in a fraction of seconds using Gröbner basis algorithms and that the costs are comparable to about four encryptions. Furthermore, the number of Gröbner basis applications can be reduced by implementing techniques for checking for inconsistencies in the equation system such as checking if the subsystem of linear equations is solvable etc.

A.3 Targeting the low number of initialisation rounds

A2U2 has a secret number of initialisation rounds. There are 32 possible choices for the number of initialisation rounds that varies from 9 to 126 where each choice is specified by the 5 LSB of the tag's random number, the reader's random number and the 5-bit counter key. In this section, we propose two attacks on A2U2 when 9 initialisation rounds are used. The first attack recovers the 5-bit counter key using 2^{14} different state pairs with a specified difference, while the second attack recovers 32 secret key bits and 6 subkey bits using 8 plaintext/ciphertext pairs with a time complexity 2^{38}. Both of the attacks use known plaintexts and chosen IVs.

A.3.1 Recovering the 5-bit counter key

The following attack requires for each of the possible 32 initialisations, a certain number of state pairs (chosen IVs) with a good difference (sparse characteristic). Under these states we use the ciphertext of a single bit of plaintext that is equal to the first bit of the counter, C_0, at the time when the encryption starts (known plaintext). Then we can distinguish the state pairs corresponding to 9 rounds of initialisation by observing a bias in the difference of the first bit of the corresponding ciphertext pairs, $\Delta(Y_0)$, which is equal to the difference $\Delta(B_0)$.

Experiments show that $\Pr(\Delta(B_0) = 0) > 0.7$ for 9 initialisation rounds when 2^9 state pairs with differences $\Delta(A) = 10000000000000000$ and $\Delta(B) = 100000100$ are used. The bias is smaller when more initialisation rounds are applied. We observe the strong bias in B_0 for 9 rounds because in only 9 rounds the difference cannot propagate through state and does not spread out sufficiently. After having distinguished the 2^9 state pairs corresponding to 9 rounds of initialisation, we can consequently find the 5-bit counter key.

A.3.2 Recovering the master key bits

The following attack targets plaintext/ciphertext generated using 9 rounds of initialisation (can be obtained using chosen IVs) and exploits the key scheduling used to generate the subkey bits, h_t. The attacker starts by guessing the 26 master key bits used in initialising registers A and B and then at each round he guesses one subkey bit if $C_{t-1} = 0$, or two master key bits if $C_{t-1} = 1$ (since A_{t-2} will then be used to generate h_t) or no bits when all the key bits involved in the generation of the round subkey bit are from the 26 master key bits used in initialising registers A and B.

The cipher is initialised using 9 rounds and then a 5-bit plaintext is encrypted, so in total the cipher runs for 14 rounds. Without loss of generality we assume that the starting key position is 0, so the key bits at positions 0 to 25 are used in initialising registers A and B.Table A.1 shows the key bits positions used from round 0 to round 13, the value of C_{t-1} and the number of guessed subkey/key bits. The table also shows that we have to guess 12 subkey/key bits plus the 26 master key bits used in initialising the registers.

To recover 32 master key bits and 6 subkey bits using plaintexts of length 5-bit, we need only $\lceil \frac{38}{5} \rceil = 8$ plaintext/ciphertext pairs of length 5-bit to find the right key guess. The remaining 24 master key bits can be recovered using a brute force search. Thus, the total complexity of the attack is in the order of 2^{38}. In order for the above attack to work we need to find only 8 plaintext/ciphertext pairs whose initial state pairs are initialised with position 0 as a starting key position and are generated using 9 initialisation rounds which we can easily find using the 5-bit counter key recovered in the previous attack.

Table A.1: List of the masterkey bits used in the subkey generation of each round, together with the counter value and the number of required guesses. $r \equiv$ round no, $G \equiv$ number of subkey/key bits that are guessed.

r	S_0^t	S_1^t	S_2^t	S_3^t	S_4^t	C_{t-1}	G	r	S_0^t	S_1^t	S_2^t	S_3^t	S_4^t	C_{t-1}	G
0	26	27	28	29	30	0	1	7	5	6	7	8	9	1	0
1	31	32	33	34	35	0	1	8	10	11	12	13	14	1	0
2	36	37	38	39	40	0	1	9	15	16	17	18	19	1	0
3	41	42	43	44	45	1	2	10	20	21	22	23	24	0	0
4	46	47	48	49	50	1	2	11	25	26	27	28	29	0	1
5	51	52	53	54	55	1	2	12	30	31	32	33	34	0	1
6	0	1	2	3	4	1	0	13	35	36	37	38	39	0	1

A.4 Exploiting the noisy keystream

As mentioned earlier, the equations $B_t = Y_t + C_t$ holds with probability $\frac{3}{4}$. One possibility to use this kind of information is polynomial system solving with noise. We can describe the inner state of the cipher and the key bits as a polynomial system by introducing variables for the initial state bits and the key bits. Additionally, in each clocking of the cipher we introduce a new variable for each updated register bit and one variable for the auxiliary value h_t. Moreover, we obtain three non-linear Boolean equations; two coming from the update of the registers and one from determining h_t.

In each clocking we obtain an additional equation from the correlation between the sequence B_t and the noisy keystream. Let $E_t = Y_t + C_t$ be the noisy keystream. This yields the additional equation $E_t + B_t = 0$ that holds with probability $\frac{3}{4}$.

Given the description of the internal state and the key bits as polynomial system we can formulate the key recovery problem as *Partial Max-PoSSo*[**?**]. This means that we split the equations in two sets, a hard set $\mathcal{H}$ and a soft set $\mathcal{S}$, where all equations in $\mathcal{H}$ have to be satisfied, will we try to maximise the number of equations which are satisfied in set $\mathcal{S}$. An approach to solve Partial Max-PoSSo problems is mixed-integer linear optimization, where the problem of solving Boolean equations has first to be converted into a integer/real valued optimization problem [24, 6]

When using mixed-integer linear programming to solve a Boolean equation system the approach is to convert the Boolean equations into linear real-valued equalities and inequalities at the cost of introducing additional variables and equalities/inequalities. This set of equalities and inequalities describes then the set of constraints in the corresponding mixed-integer program. Additionally, some variables can be restricted to binary or integer. A solver for MIPs will then try to find an element in the feasible set, which is the set of elements that satisfies all constraints and restrictions, that minimises a given function, the so-called objective function.

We use the integer adapted standard conversion method [24] to convert the Boolean equations into integer-valued equations. Let $f = 0$ be a Boolean equation. We interpret the Boolean polynomial f as a polynomial g over the integers by replacing AND by multiplication and XOR by addition. Evaluating the integer polynomial g for a solution of the Boolean equation will yield a multiple of 2. Let u be the maximum outcome of g for a solution of $f = 0$. The integer equation $g - 2y = 0$ where $0 \leq y \leq \frac{u}{2}$ is an integer holds for all solutions of $f = 0$.

These equations over the integers are still non-linear. We replace non-linear terms by new variables and introduce inequalities that will force the newly introduced variable to take on the correct value. As an example consider the quadratic term $q = x_1x_2$ where $x_1, x_2, q \in \{0, 1\}$. The inequalities $q \leq x_1$ and $q \leq x_2$ force q to be zero if either x_1 or x_2 are zero, while the inequality $x_1+x_2-q \leq 1$ ensures that q is one if x_1 and x_2 are both one.

Given the the conversion and linearization method it is straight forward to transform the equations in the hard set $\mathcal{H}$, meaning the equations coming from the register updates and generation of the value h_t, into constraints for the MIP. When modeling the soft set we introduce slack variables s_t. We know that $B_t + E_t = 0$ with a probability of 3/4. This translates into the following integer equations

$$B_t + s_t = 1 \quad \text{if } E_t = 1 \tag{A.3}$$

$$B_t - s_t = 0 \quad \text{if } E_t = 0 \tag{A.4}$$

where $s_t \in \{0, 1\}$. In order to maximise the number of equations of $\mathcal{S}$ that are satisfied we minimise over the sum of the slack variables.

First, we consider a chosen-plaintext attack where the attacker may choose one plaintext of length around 50 bit and knows a second plaintext. As in Section A.1 the attacker chooses the plaintext to be the all zero plaintext. That means that the attacker knows that the equation $B_t + E_t = 0$ holds with probability 1 if $C_t = 0$, meaning that those equations can be moved to the hard set $\mathcal{H}$ and the number of slack variables can be reduced. Furthermore, we simplify the MIP by guessing the values of h_t for the first 35 clockings. In our experiments we used the solver CPLEX which would yield between one to three solutions. In more 90% of the test cases the right key and initial state was one of the solutions. In the remaining cases the solver found a solution for the MIP yielding a smaller objective value than the desired solution without considering this solution before. Solving the mixed-integer program took 116 seconds on average which leads to a total complexity of $116 \cdot 2^{35} \approx 2^{42}$ seconds on a single CPU.

As future research it remains to investigate the behaviour of the attack when reducing the number of guesses auxiliary values. Furthermore, the attack should be considered in a known-plaintext scenario.

A.5 Conclusion

In this paper we presented several attacks on the lightweight stream cipher A2U2 which all constitute practical breaks of the cipher.

A2U2 is designed for IC printing. To keep the area small, short registers for the inner state are used. A new output generator has been developed to increase the security of the cipher. This output function works similar to the shrinking generator but it overcomes the problem of the irregular output of ciphertext bits by outputting 'dummy' ciphertext bits such that attacker does not know at which positions of the ciphertext plaintext bits were used.

We show that using only two chosen plaintexts, the cipher can be broken in a second by first recovering the sequence which is used to produce the ciphertext and afterwards determine the sequence which controls when plaintext bits are used.

Furthermore, we propose a guess-and-determine attack. We guess the positions where the plaintext bits are used and set up a non-linear equation system which can be solved, e.g., using Gröbner basis techniques. With this approach we can determine the key with a complexity of 2^{49} guesses using a known plaintext of length less than a hundred bits.

We also investigated the possibilities of a chosen-IV attack. Choosing the IV allows us to introduce a difference in the initial state of the register and keeping the counter constant at the same time. Using a differential-style attack, we can identify a bias in the difference of the first ciphertext bit when only 9 initialization rounds are used. Thus, we can recover the counter.

When only 9 rounds of initialization are used, we can recover the master key using only 2^{38} computational steps. We find 32 master key bits by guessing depending on the counter either master key bits or the auxiliary value h_t. The remaining master key bits can be found by exhaustive search.

Moreover, we applied a rather new attack technique: polynomial system solving with noise. We make use of the fact that we know approximately 3/4 of the sequence B_t and set up an equation system where the equations containing information of this noisy sequence do not necessary hold while all other equations in the system have to be satisfied. The problem of solving noisy polynomial system can be modeled as a mixed-integer linear programming problem and we have been able to recover the secret key with a success rate of more than 90% in a single-chosen-plaintext scenario.

We conclude our analysis by pointing out some of the most severe weaknesses of the cipher. The biggest and most obvious weakness is that the counter value at the beginning of the encryption is known. However, any change made to the cipher demands a re-analysis of the security. In the current state the cipher can be considered as broken.