Implementation of RSA Cryptosystem for Next Generation RFID Tags

By

Ashish Arun Dighe

A Thesis presented to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

Ottawa, Ontario, Canada, 2011

I hereby declare that I am the sole author of this document. I authorize the University of Ottawa to lend this document to other institutions for the purpose of scholarly research.

Ashish Dighe

I further authorize the University of Ottawa to reproduce this document by photocopying or by any other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Ashish Dighe

The University of Ottawa requires the signatures of all persons using or photocopying this document. Please sign below, and give address and date.

ABSTRACT

This thesis addresses concepts of implementing a RSA cryptosystem on a passive RFID tag. With a limited number of public key cryptosystems on passive RFID platforms, the proposed algorithm makes use of Montgomery multiplication primitives to reduce the amount of computation required on the power constrained tag therefore making the proposition viable. Public key cryptography is being suggested for next generation RFID systems to reduce the number of possible attack vectors native to this type of technology. By estimating the area, power and time constraints of the RFID platform, it was determined that the area constraint was the critical variable in determining the maximum implementable security variable. Although the application of this algorithm has been targeted for passive HF RFID platforms, the algorithm could be used in other low power, sized constrained applications.

ACKNOWLEDGMENTS

This thesis would not have been completed without the help and support I received. I would like to acknowledge all those who have provided me with the opportunity to complete this thesis and my graduate studies.

I would like to thank my supervisor at the School of Information Technology and Engineering at the University of Ottawa, Dr.Tet Yeap. I grateful for both the guidance and support he has provided me during the course of my research.

I would also like to express my gratitude and appreciation to all my friends and family for their understanding and encouragements during the period of my graduate studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 Overview of Field

Increasingly the world has been adopting wireless technologies to meet the needs of an ever growing need to manage the world of objects. Contactless radio frequency identification (RFID) tags make up a portion of the wireless segment with the goal of cataloguing, securing and modifying the manner in which we interact with objects in the world around us. RFIDs are currently being used, but are not limited to, inventory control mechanisms and identification. As these devices become more ubiquitous, concern is rising regarding the security of the information that is broadcast through the wireless medium. This thesis attempts to identify a means of securing the technology using an asymmetric public key encryption, which can be used to perform authentication routines and secure the communications medium.

Inventory control and management is a common use for RFID tags. In the commercial sector, there has been great interest in the promulgation of this technology to identify merchandise from production to consumer purchase. The US government has implemented the RFID tags in their new passport cards for travel between the US, Canada, and Mexico, with many other countries following their stead. The common concern amongst these implementations is the likelihood of private information being transmitted over the air which could possibly be intercepted by an unintended recipient actively reading the tags without appropriate permission. In the commercial case, information regarding inventory levels could lead to a competitive edge from

competitors. The passport cards being implemented by the United States make available all printed information in the passport, including a grayscale image of the individual, on the embedded RFID chip. A breach in security, or capability to read this information from a distance without proper encryption techniques, could lead to an increase in personal identity theft and an invasion of privacy. Prior to describing how the system security may be augmented it is necessary to describe the RFID system itself.

RFID systems generally operate using three discrete sub-systems: the identification tag, a reader and a backend database system. These systems operate together to transmit the data contained on the RFID tag to the backend system where statistics and other metrics can be collated and measured.

The tag generally contains a bit sequence in memory that the reader is interested in. This bit sequence can be thought of as a unique identification number which corresponds to a database entry in the backend system containing associated metadata regarding the object. More advanced RFID tags can be programmed to store all of the information on the chip itself. The RFID tag operates without contact and depends on the reader to extract information from the tag at a given distance. This is achieved through radio frequency communication protocols across various spectrums which differ depending on the region of the world and the operative standard.

The reader can be a mobile unit or stationary unit depending on the application. It is meant to wirelessly communicate with the RFID tags to extract pertinent information from the device. The readers work in various frequency ranges dependent on the

technology which is used. In general, once the data is received it will append metadata regarding the approximate position of the tag, the time read, and other application specific metrics and send the data along to the database backend. The applications that are run on the reader are generally called middleware and are an important step to ensure that data is properly formatted for ingestion into the main database.

The database backend is an important analytical tool which is used to collate the collected data from the RFID tags. It has many advanced functions that enable it to track and trend the RFID tags and depending on the specific application allow large volumes of data regarding the object to be stored and accessed. It is generally at this level that analysis of the data is performed through a graphical user interface which is tied to the database.

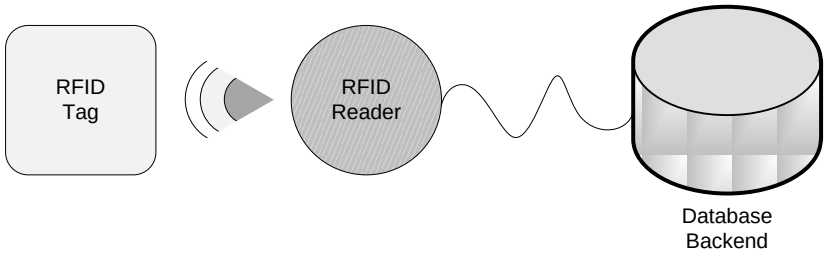

Figure 1.1 RFID System Overview

1.1.1 RFID Tag Security

Security for passive RFIDs is generally performed through the use of symmetric keys. Some public key algorithms have been implemented on expensive active RFID units which require batteries to operate but none to the author's knowledge have been implemented on a passive RFID unit (which operates using power transmitted

wirelessly by the reader). Ideally, the public key infrastructure would be extended to the passive RFID tags to enable authentication and authorization protocols to enhance interoperability between various client systems.

There are several key differences between symmetric key applications and public key applications. In a symmetric key implementation, a secret key is shared between two parties. The secret key must first be shared in a secure location, and then can be subsequently used for all communications. Symmetric key algorithms are not flexible and can be permanently compromised due to the reliance on a single secret key. The inflexibility is due to the fact that once a key is programmed into the device, the key is generally not changed. All communications use the same key, and once it is broken the communications become available to the adversary who has discovered the key.

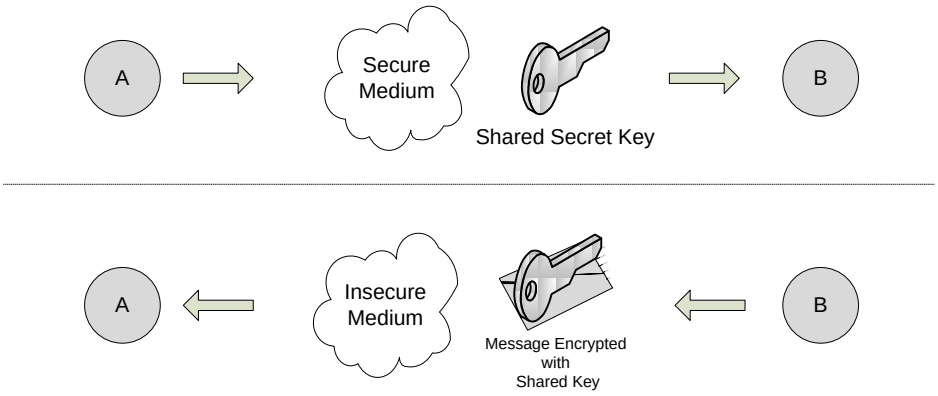

Figure 1.2 Private Key Exchange and Encryption

The most popular method of securing passive RFIDs is the use of symmetric key algorithms. This is due to the area, size and power limitations are present in passive RFID implementations. Symmetric key algorithms share a secret key which is used to encrypt and decrypt information between parties. The most basic form of a symmetric

key algorithm is the use of an XOR operation between the message and the secret key. This type of operation is small to implement, takes up little power, and does not use many transistors or logic elements. This is ideal for simple read operations, but once the key is compromised, the information is always accessible. Using symmetric keys does not allow the same level of authentication of the reader by the tag or vice versa. This is a limitation in the use of symmetric key algorithms since digital certificates for authentication purposes are not possible in a symmetric key system. Although the primary purpose of a symmetric key system is to encrypt data, with an asymmetric key system there are additional features that can be implemented such as authentication and authorization protocols. When using a symmetric key, a trusted entity is based on possession of the symmetric key. In an asymmetric key system a result is computed given publically available information and hash strings are compared to identify a trusted entity. The difference is subtle but important: possession of the key versus possession of a key verified through a computational result. In addition the a symmetric key cannot be changed unless there is a secure channel. The secret key that is on a tag is usually burned into it memory upon manufacture. This is especially important when dealing with sensitive personal information in the case of passports or other government issued identification as repeated use could lead to cloning.

Public key algorithms provide a greater amount of flexibility when dealing with securing the communications medium. Public key algorithms can be used in a manner that facilitates the distribution of private keys for a given session. If there are numerous interactions with a tag over the course of a transaction that need to be encrypted a private key exchange can be performed in which the private key is issued

under a public key encryption. Since public key encryption and decryption generally take longer and are more computationally intense, it is advantageous to perform such an exchange.

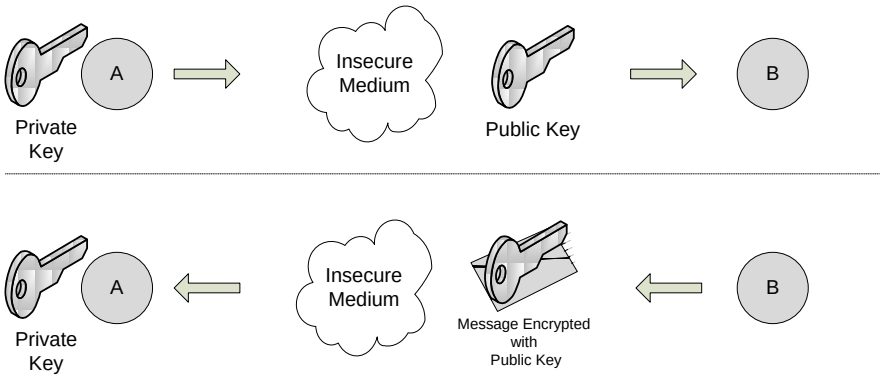

Figure 1.3 Public Key Exchange and Encryption

Public key algorithms are able to facilitate the use of public key infrastructure segments. As part of the public key infrastructure, authentication and authorization play a large role. Authentication of a given reader would enable the tag to ensure it is not responding to a malicious reader. By authenticating against a known set of readers, a tag would be able to selectively reply to queries. Additional work would need to be done to implement such a system and ensure that the processing could be performed at either end to verify the identity of the reader. Implementing a public key encryption system on an RFID tag would be the first step to allowing authentication and authorization protocols on to an RFID tag chipset.

1.1.2 RFID Industrial Standards

The common standards that are referred to when dealing with RFID tags are ISO 14443 and ISO 15693. These standards are used to provide interoperability between various manufacturers. ISO 14443 refers to proximity cards and their mode of operation. ISO

15693 refers to vicinity cards which have a greater distance of operation than the cards that are identified in ISO 14443. In this thesis we examine the possibility of implementing an RSA public key cryptosystem using an ISO 15693 compatible RFID system.

1.2 Problem Statement

The objective of this thesis is to determine the feasibility of implementing an RSA cryptographic system using a passive RFID tag system.

It has been stated that an RSA algorithm implemented on a passive RFID tag would be difficult to implement with current technology [**1**]. The main reason that this technology is thought to be infeasible in the form factor presented is due to the size of a general RSA cryptographic implementation. It is stated in [**1**], that there is only room for 4,383 gates on a passive UHF RFID, and that an RSA implementation would take 34,000 gates. From the information that is gleaned from the research performed in the prior paper it indicates that it is not possible to perform RSA cryptography on an RFID tag. This conclusion does not take into account varying the distance of the tag nor does it analyze possible algorithmic variations that could lead to an implementation which would reduce the gate requirements for an RSA cryptosystem on an RFID tag. The suggested research will attempt to vary some of these variables to ascertain the possibility of implementing an RSA cryptosystem on an RFID tag.

There are many methods of implementing an RSA cryptosystem that are described in literature primarily focused on the FPGA medium. Using this information, a considered effort is used to assess the suitability of various algorithms for the RFID application.

Proper examination of the algorithms will be undertaken to identify the most probable candidate for application to the RFID platform. This will be done through the consideration of power, timing constraints, and the size of implementation. Through this examination a proper algorithm will be identified to meet the requirements of implementing an RSA based encryption on the RFID tag platform.

1.3 Prior Art

There has been a significant amount of work which identifies optimal implementations of cryptographic encryption algorithms for an RSA implementation [**2**], [**3**]. This work generally focuses on implementations for FPGA technology. There has been little work on the implementation of optimal public key encryption processors for the passive RFID platform.

RFID tags are generally constructed out of the following constituent parts: antenna, rectifier, voltage pump, decoder, controller, memory, cryptographic unit, and encoder. This thesis focuses on the cryptographic unit. The cryptographic unit will be characterized through an area, power and timing analysis. The RFID system integration will be considered in terms of the constraints on inputs that the system would impose and the required output format.

The power that is available to the tag is of primary interest for this type of implementation. The power that is delivered to the tag is constrained by the output power of the reader which is regulated by regional spectrum licensing authorities. This constraint defines the amount of power that is available to the tag at various distances.

Regulation on the output imposes a maximum distance of operation and defines the amount of power that is available to the tag for both operation and response to query.

The RSA algorithm can be implemented in various ways. An examination of the more probable implementations is examined and discussed. The most realistic implementation deals with the Montgomery method which in itself has various algorithmic variances. A thorough analysis of the implementation options is presented with a discussion of the most suitable algorithm meeting the constraints defined below.

The security of the proposed algorithm is presented to demonstrate the ability of the proposed method to secure the channel and maintain privacy between the reader and the tag. This security is shown to through commonly accepted definitions of security to be secure. The main consideration is for the algorithm to produce a ciphertext which is indistinguishable from a randomly chosen plaintext and the desired message.

1.4 Contributions

Various algorithms have been developed which are capable of performing the necessary steps for the RSA algorithm. Most of these comparisons are implemented on an FPGA architecture where the space and power constraints are not the primary goal of the design. The proposed method investigates the feasibility of implementing an RSA cryptographic system on an RFID tag, considering the security, and technology limitations.

With the RFID application in mind a review of the available to the tag will be performed. Currently, there are few publications that identify the power available to public key cryptographic systems. Using the 0.35um technology and the ISO 15693

standard as the baseline identifying the constraints on the tag, a power analysis will be performed to identify the available power to the tag and will constrain the size and type of implementation chosen for the realized circuit.

The thesis also identifies an algorithm would be able to securely transmit the data from the tag to the reader. This algorithm will be proven secure under the premises of probabilistic encryption set out by Goldwasser and Micali [**4**]. By showing that this type of encryption is suitable through formal methods it will reduce the footprint of the required circuit requirements.

A final analysis will be performed to show that the algorithm proposed will meet the requirements of the encryption schema and that the overall algorithm will fit within the constraints that were identified in the previous analysis.

The flow of the analysis will be the following. First an examination of the available methods of encryption will be performed using the known techniques and identifying the required space at the transistor level. Second an analysis of the power requirements of the tag circuitry will be performed. This will identify the constraints on the algorithm in terms of available power. Third an analysis of the proposed algorithm will be performed with an analysis of the security that it is able to provide. Fourth, an examination of the proposed algorithm and how it will fit on to tag will be done. This will conclude the desired scope of the thesis.

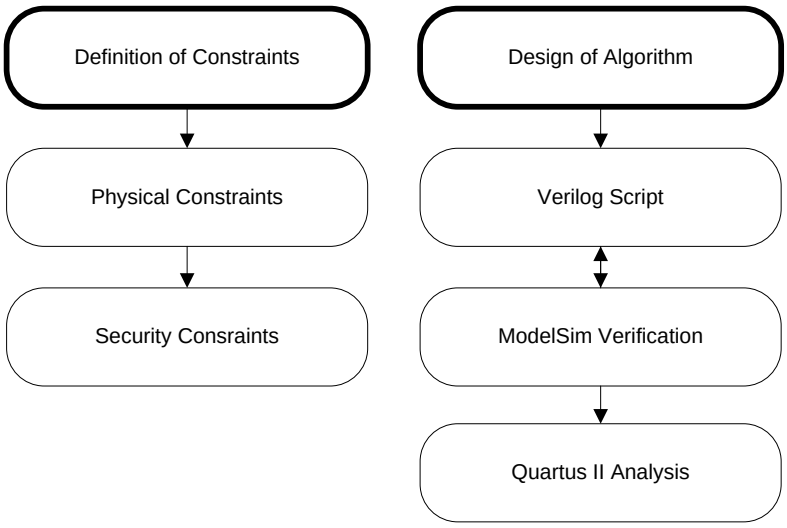

Figure 1.4 Thesis Methodology

1.5 Assumptions and Limitations

The thesis is limited to the scope that it defined in the problem statement. The main purpose of the paper is to examine the possibility of a public key encryption system to be ported to an RFID tag platform within the constraints of a 0.35 μm transistor technology. The paper will examine the constraints that are imposed by the technology and what the possible implementation sizes are after doing the appropriate link analysis and power analysis. It does not purport to design an entire tag nor does it investigate the reader or backend portion of the system. This would be considered out of scope for the thesis at hand.

The thesis topic is constrained by various international and industrial standards. The tag design will be assumed to conform to the ISO 15693 standard. This constrains the power that is delivered by the reader, and the type of communications protocols that

are used in the transference of information from the reader to the tag, and from the tag to the reader. It also defines some of the components that need to be present for the operation of the tag. The industry standards that have already been defined are used to constrain the scope of the thesis. These standards are not limiting so much as they define best practices for RFID communication. Deviating from the standards significantly reduces the industrial application of the device. ISO 15693, and others that define the manner in which the RFID system must conform constrain the design.

The paper will advise on the type of algorithm that should be implemented and perform the analysis to determine the feasibility of the design, but does not strive to design the encryption engine itself on an ASIC.

The thesis will, through the use of FPGA technology, identify the size, and power requirements of the recommended implementation. This is done because of the inherent flexibility of the FPGA platform for algorithmic development. In addition, the FPGA platform was the only available option at the university which allows for the analysis of the variables under consideration without a significant investment in resources (ASIC design, or custom integrated circuit fabrication). The FPGA platform can provide insight into the size, power, and timing parameters that would be characteristic of a given implementation. Using the information that is gained through this method, approximations on the relative size, timing, and power of the proposed algorithm will be made.

1.6 Thesis Overview

The structure of this thesis is organized with a general progression of understanding culminating to the results and discussion chapter 5. Chapter 2 describes background information of RFIDs, security present on current iterations of RFID tags, future requirements of RFID tags and a general discussion of complexity theory and how it relates to security. Chapter 3 provides a more in depth discussion of the RSA algorithm, methods of implementation algorithmically, and the constraints on the design of the RFID tag. Chapter 4 provides a discussion on the proposed algorithm, and the methodology used in the design. Chapter 5 shows results and discussion. Chapter 6 provides a conclusion and the contributions of the thesis.

CHAPTER 2- RFID SECURITY

This chapter examines Radio Frequency Identification (RFID) systems and security. A general explanation of the current state of the art is provided along with the standards that govern their use. A brief look at the security protocols currently being implemented to mitigate security threats to RFID technology is provided. Discussion of the issues that face current security protocols is presented. Finally a discussion of what security means in the modern context is examined to better understand the requirements that are being implied for a truly secure protocol implementation.

2.1 Radio Frequency Identification

"RFID tags are an automatic data capture technology that comprise of small data-carrying transponders (tags), and fixed or mobile scanners (readers)," [**5**]. Tags are generally attached to objects or people to allow them to be automatically identified by a reader. The tags can be used to localize or track goods. Traditionally barcodes were used for this application but they have several disadvantages that are addressed through the use of RFID technology.

Some of the advantages that RFIDs pose over their predecessor include non-line of sight operation, orientation independence, memory capabilities, and the ability to read and write to numerous tags at the same time [**6**]. Since RFID tags operate using radio frequencies, line of sight is not a requirement. Limitations to the communication link are related to the radio frequency propagation characteristics in the environment. In general, the tags can communicate with the reader as long as the reader is sensitive enough to intercept a tag's response. Another benefit of RFID tags is that they are less orientation sensitive than

barcodes. This means that whether the tag is on its side or upright, the tag should be able to respond. This is not the case with barcodes as the pattern must be read in a certain orientation. The read and write capability of a tag augments the number of applications possible compared to previous mediums. Traditional two dimensional barcodes are able to store up to 2Kb of static data. RFID tags can store an increasing number of bits depending on the type of tag (passive tags can store more than 8KB) [**5**]. Barcodes are only able to present data in a static format whereas RFIDs can change the data that they transmit dynamically. Last, many RFID tags can be read at a time, whereas barcodes are traditionally only able to read one at a time. These advantages are of significant benefit for various industries.

Table 2.1 Barcode and RFID comparison [5**]**

Property	Barcodes	RFID
Line of Sight Required	Yes	No
Cost In Volume	~Free (printed)	10-12 cents (2006)
Processing function options	No	Yes
Additional Memory	No	Yes
Read/Write capability	Read only	Read/Write
Multi-tag arbitration	No	Yes

2.1.1 Classes of RFID Tag

RFID tags are generally split amongst their constituent groups through the definition of how the tag is powered. There are two separate parts that are usually mentioned: the communications channel, and the logic on chip. The communications channel refers to the means in which the tag communicates to the reader. Two varieties exist: transmitting a signal using the power from the tag, and reflecting the incident energy incident to the tag.

The logic on chip is defined as everything past the antenna circuit. With these parts, the types of tags can be explained.

There are four classes of RFID tag: semi-active, active, semi-passive, and passive [7]. Semi-active tags harvest energy from their environment to power the logic and communications link. These tags can use solar energy, vibration energy (piezoelectric rectification) or another means to power the logic on the tag. They are also known as energy harvesting tags. Active tags are those tags which use battery power to power their communications with the reader and the logic on the tag. These tags usually have a greater range than semi-passive or passive tags and can be comparable to semi-active tags depending on the specific implementation. Active tags have the disadvantage of relying on a battery source, which once depleted must be replaced. Semi-passive tags use batteries but only to power the logic portion of the tag once activated through incident energy from a reader. The semi-passive tag uses modulates the incident signal to communicate to the reader. The process of reflecting the energy back to the reader as a means of communication is usually called back-scattering. Passive tags do not have any source of power on their own and rely solely on the power that can be rectified by an interrogating reader to power the tag. These tags, like the semi-passive variety, use back-scattering to communicate back to the reader.

Table 2.2 RFID Tag Types

Active Tag • Communication and logic powered by onboard battery • Increased range	Semi-Active Tag • Communication powered by backscattering incident signal • Logic powered by onboard battery
Semi-Passive Tag • Logic powered by energy harvesting methods (solar, vibration, etc) • Communications performed through backscatter link	Passive Tag • Communications performed through backscatter • Logic powered through reader transmitted power.

Semi-passive and passive tags can be distinguished through their coupling mechanism: near-field and far field operation [**7**]. Depending on the frequency of operation and the intended distance of operation the tags can harvest either the magnetic field (near-field) or the electromagnetic field (far-field) energy. In terms of power that is delivered to the tag, near-field operations usually have the benefit of an increased power density compared to far-field tag operation in close proximity. In general, the lower the frequency the more likely it will use near-field coupling due to radiation propagation.

2.1.2 Standards Associated with RFID Tag Operation

RFID technology has been under study since the 1970's [**8**]. Although the technology was initially put to the national standards committee to determine the viability of creating a standard for operation, this was not followed through. Consequently, this decision had a twofold effect: the technology was able to become more mature through heuristic trials; there was no means of maintaining control over the protocol stack and the various implementations of the technology. The number of devices spiraled and there were few if any standards that were imposed on the technology. In 2003 EPC (Electronic Product Code) Global put forward an outline governing the usage of the RFID tags for the use of electronic product identification. At this time ISO had already defined portions of its RFID centric standards. This led to confusion in the marketplace as to which standard to follow and which one has greater reach and applicability.

The International Organization for Standardization deemed that the technology needed to be standardized to ensure interoperability between the various manufacturers' and require a common protocol stack from which the RFID tag and reader could communicate. The standards that were developed operate in the LF, HF, and UHF frequency ranges. These

standards can be cited as ISO 14223 (LF) [**9**] [**10**], ISO 14443 (HF) [**11**] [**12**] [**13**] [**14**], ISO 15693 (HF) [**15**] [**16**] [**17**], and ISO 18000 (UHF) [**18**] [**19**] [**20**] [**21**] [**22**] [**23**].

During the period of dispute, with regard to the applicability standards to the RFID market, EPC continued to vie for industry preference for their standards. EPC Global has three specifications one for the HF frequency range [**24**]and two for the UHF frequency range [**25**] [**26**]. It can be said that EPC Global was first to enact an RFID standard, but their standard focuses heavily on a specific application of RFID technology: inventory. The ISO standards allowed more flexibility in its applicability and did not prescribe the use of the technology to the same extent as the EPC standard. Thus, initially EPC Global was the assumed standard until ISO was able to define a more general RFID standard. Although there are a few discrepancies between the two standards there has been an effort by both organizations to ensure minimal interference with regard to the operation of either tag infrastructure [**24**].

For the application of a public key cryptosystem the HF RFID was found to be a more suitable option for further exploration. In industry, HF tags have found preference among consumers due to their relatively low cost, their operability with various mediums, and the power delivered to the tag [**7**]. The factor of greatest importance in the realization of an RFID cryptosystem is the available power to energize the additional transistors required to enable and run a cryptosystem. In the ISO 15693 standard the best compromise between power delivered to the tag and operational distance was found. ISO 15693 defines a tag should be operable up to one meter. This is important when performing tasks such as inventory control. The HF spectrum, due to its propagation characteristics, can be placed on and in more locations compared to the UHF tags.

2.1.3 Standards for Security

Through consultation of the standards mentioned above, it is distressing to note that security is not mentioned in any of the standards. In the EPC Global standard there is the mention of a password, but this is meant to lock the memory so that no additional writes occur once the tag has been programmed unintentionally [**24**]. There is no standardization that is proposed to ensure that the information that is passed to the reader is secure. It is assumed that the standard's associations have left the securitization of the tag-reader communications up to the discretion of the manufacturer due to its nascent state. Working under this assumption there have been several iterations of "secure" implementations of RFID technology by industry. All of these implementations have been performed through the use of symmetric keyed algorithms to the author's knowledge.

MIFARE developed by NXP Semiconductors of Austria is leading the way regarding securing RFID tags, although Texas Instruments has a similar secure implementation. To date, MIFARE has implemented cryptosystems which can be used with triple-DES (Data Encryption Standard) and AES (Advanced Encryption Standard) [**27**] [**27**]. These standards are both implementations of symmetric key block ciphers.

Triple-DES is a block cipher that uses three keys to encrypt and decrypt a plaintext [**28**]. The plaintext is first encrypted with key 1, subsequently decrypted using key 2 and then encrypted again with key 3. Since each key defines a separate transformation space, alternating between encryption and decryption creates another subset transformation space in which commutation is not applicable. Thus, to decrypt the plaintext from the ciphertext the ciphertext must be decrypted using key 3, encrypted using key2 and finally decrypted using key1. Triple-DES was implemented to take advantage of the already implemented DES

block cipher method without varying the hardware. The 56-bit key size of the original DES system was found to be inadequate, and could be easily broken. By performing the three steps as mentioned above the overall system effectively used a 168 bit key. A mathematical representation of the above system is shown in equations (2.1) and (2.2)

$$C_1 = \left(D_{K2}\left(E_{K1}(P_1)\right)\right) \tag{2.1}$$

$$P_1 = D_{K3}\left(E_{K2}\left(D_{K1}(C_1)\right)\right) \tag{2.2}$$

The AES standard that is being implemented uses a variation on the block cipher theme. The algorithm uses a series of steps to ensure that the enciphered plaintext is permuted in such a way as to disguise the symmetric key and to provide diffusion of the plaintext [**29**]. It operates using a matrix of bytes which it uses to perform a block cipher. Three matrices, one that contains the plaintext (up to 256 bits), a similar sized key matrix and a third lookup table matrix. Initially the key is XORed with the plaintext. The resultant is then transformed using the byte transformation lookup matrix. The matrix entries are then shifted by a known quantity. This matrix is then linearly transformed through row and column shifting. Finally the matrix is XORed once again with the key to produce the final enciphered message. The process is reversed to obtain the original plaintext.

Both of these solutions are sufficient but do not allow the flexibility of a public key encryption system. As will be described later, public key encryption ensures that the information that is being sent is encrypted, but does not have the additional key management overhead. Public key infrastructures also enable the recipient to verify that the tag is valid and not a forfeit. This verification can be achieved through digital signature verification, in

essence authenticating the tag as being genuine. This type of flexibility is not currently available to the author's knowledge using the same hardware in symmetric key algorithms.

2.2 Security Concerns

2.2.1 Security Vulnerabilities

There are currently a number of security vulnerabilities that are present in RFID tags and in the RFID system architecture. Network attacks as well as RFID specific attacks can be used to: prevent data access, spoof the identity of an RFID, clone or track the tag among others. Some of the risks are outlined in Figure 2.1.

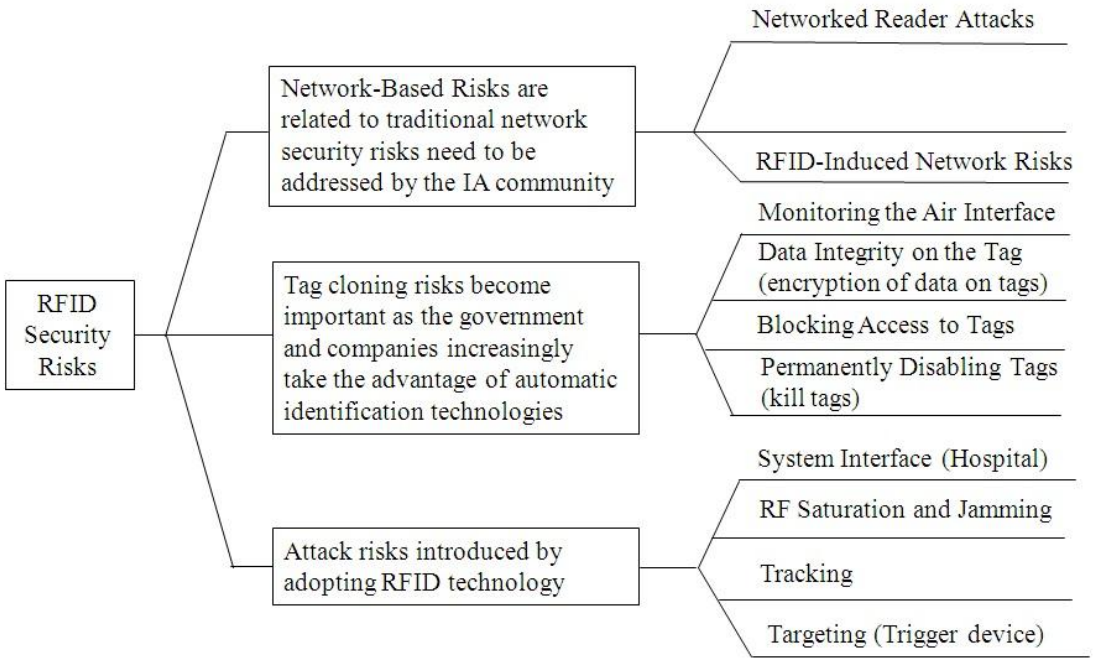

Figure 2.1 Diagram of Security Risks [30**]**

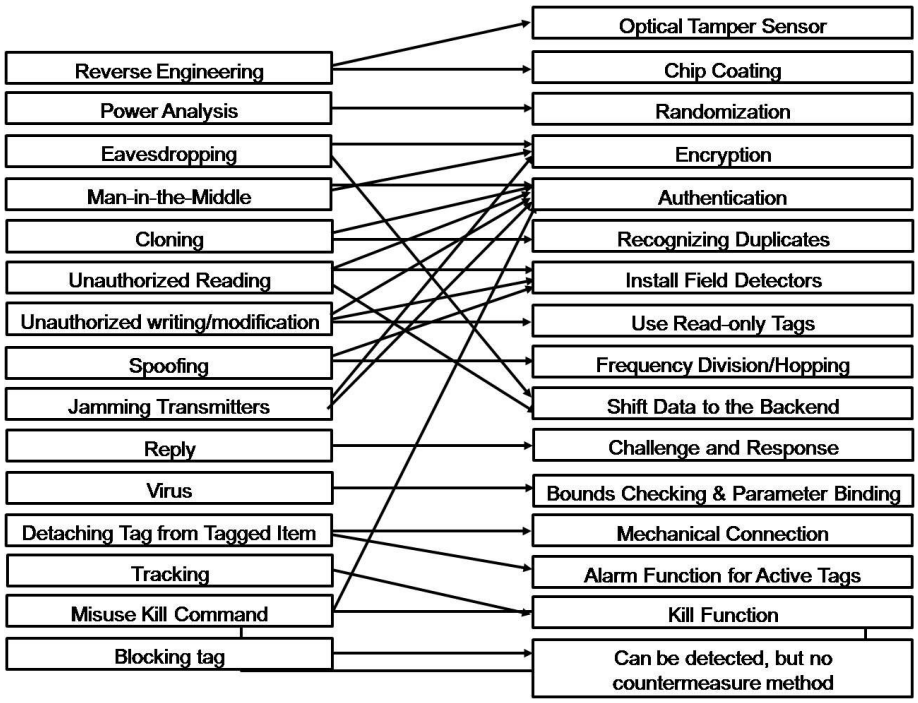

Figure 2.2 RFID Attacks and Countermeasures [30**]**

A number of the attack vectors have known countermeasures indicated by Figure 2.2, with many of them already being addressed through other research initiatives. Many of the countermeasures are able to mitigate more than one threat. The attacks and their countermeasures are discussed in greater detail in the following sections.

2.2.1.1 Eaves Dropping

Eavesdropping is a passive attack strategy that enables an unintended recipient to identify messages that are being passed from both tag and reader. This is performed by placing a secondary reader within the vicinity of the primary reader so that messages that are transmitted can be picked up by both the primary reader and the unintended recipient perpetrating the eavesdropping [**31**]. This type of attack compromises the intended confidentiality between the tag and reader. This type of information could be used to identify materials, to indicate the location of specific items, or to simply act as a trigger

indicating that an RFID is within proximity of a given location. Positioning of the eavesdropping equipment is important for this type of attack since tag responses are typically 60dB lower compared to the interrogating reader signal [**5**].

2.2.1.1.1 Countermeasures

For eavesdropping, the main method of mitigating this type of attack is to encrypt the channel. Encryption is a main countermeasure to many attacks, and eavesdropping is just one of many attacks which can be mitigated through its judicious application. Given a secure encryption scheme through which the plaintext has been transformed, if the eavesdropper intercepts a message the contents will be concealed by the use of encryption.

2.2.1.2 Man-in-the-Middle Attacks

Man-in-the-middle attacks occur when someone actively eavesdrops and changes the content of the information being passed through the communication channel. The key difference from passive eavesdropping is that the adversary is able to change the contents of the message in a controllable manner. In a man-in-the-middle attack, the adversary intercepts a message, reconfigures the contents, and then transmits the modified message to the reader, posing as the originator. This form of attack is of extreme concern as we move toward NFC (Near-Field Communications) devices whose primary application is in performing financial transactions [**32**]. Using a man-in-the-middle attack in a financial transaction could allow the adversary to change the amount deducted from an account and allow them to reroute funds without the user's knowledge. This type of attack poses a significant risk profile which encompasses identity theft, personal credit card theft, and unauthorized access to sensitive materials.

2.2.1.2.1 Countermeasures

Authentication and encryption are both methods of preventing man-in-the-middle attacks. By authenticating the tag with the reader, this ensures rogue readers are not interacting with the tag. This type of authentication would be required to have a small computation time and area footprint to be realizable on an RFID tag. Such an authentication protocol is outlined in Lee's examination of pseudo-random stream generators for use in authentication [**33**]. Since authentication can be compromised if the handshake is performed in plain text, encryption is also required. The strength of the encryption algorithm employed should vary depending on the sensitivity of the data stored on the RFID tag. There are various means of encrypting the information from simple LFSR (Linear Feedback Shift Register) configurations, to public key RSA computations.

2.2.2 Generalized Countermeasure

For both the attack vectors described above, encryption is the preferred method to counter the attack. Through encryption it is possible to thwart attempts of unauthorized eavesdropping and man-in-the-middle attacks. These attacks could possibly reveal information about the tags or to a greater extent the system that it is part of. Man-in-the-middle attacks are of greater concern since the information that is passed back to the reader can be altered in such a manner so as to prevent proper operation, or provide information that is inaccurate.

2.4 Next Generation RFID Tags

2.4.1 Capabilities of next-generation RFID tags

Next generation RFID tags will likely have a myriad of new and useful features that will enable them to perform inventory functions better and faster than current iterations. It is

posited that they will become ubiquitous and cheaper through printing technologies as suggested in [**34**]. They will provide more computing power to run mini-applications, and share memory in a distributed fashion as suggested by [**35**]. Many of these ideas are being worked on and will likely come to fruition in the near future as transistor sizes reduce and new ways of harnessing energy in micro environments are implemented. A major theme of future RFID tags is the need to securely communicate with each device to ensure that passing data is not compromised along the way or is subject to undesired surveillance.

Security is of utmost concern when dealing with sensitive information. As more government and commercial agencies realize the benefits of RFID technology, there is an ever increasing awareness that the information that is kept on these units needs to be secure through manageable means. This is ever more crucial when storing personal information through the use of RFID augmented passports and other government issued identification [**36**].

A global requirement to use RFID technology has been mandated through the use of e-Passports, posing a security risk due to its symmetric key architecture. Symmetric key e-Passports require a vast distribution of private key information to a variety of domestic and international agencies so that records may be kept with regard to the movement of individuals in and out of a country. This in turn provides many agencies with the private and confidential information that may not be necessary for them to perform their function, yet they are able to access it due to the all or nothing approach of a symmetric key cryptosystem. In addition, the distribution of symmetric keys also poses a security concern due to the lack of commonality in the management of keys and information security once distributed to the outside agencies. As more agencies are allowed access to the private key information, the ability to manage the distribution diminishes.

Many of these issues are mitigated through the implementation of asymmetric key algorithms. In fact as shown by [**37**], digital credentials could be created using the public key system to restrict pieces of information from being revealed through the use of digital signatures and certification. This ensures that unnecessary information is not divulged to parties who do not have the need to know. The management of key information could also be provided by a backend certification authority, implementing a proper PKI architecture.

A brief example of the symmetric key distribution matter is provided below. It illustrates the symmetric key management issues and how distribution can be compounding problem in the industrial supply chain deployment of private key RFID enabled platforms.

2.4.2 Issues Arising from use of Symmetric Key Security Algorithms

Company A is a producer of goods and uses RFID tags to track their production from receipt of goods to final assembly. Company A's produces products of a sensitive nature and they has been mandated to use symmetric key accessible RFID tags by the Department of Defense. Each product that Company A produces contains at least 10 distinct parts all of which are fashioned with an RFID for ease of quality control and repair at a later date. Company A produces 1 million of such units every year. In this case the company must keep 10 million key records and augment this each year.

Company B is a systems integrator and purchases Company A's products. Its system consists of 10 from Company A, 10 units from Company X, 10 units from Company Y, and 10 RFID tags for its own purposes. For each product that Company B produces, 40 RFID tags are present per unit and if Company B wishes to keep track of these units for life cycle management and failure rate analysis, it must now inventory 40 symmetric keys for each

unit. Company B produces 1 million units a year. This means that Company B must request and store symmetric keys from Company A, X, and Y for each of its units. A loss of any one key would mean that the tag would no longer be communicable. A security breach of the symmetric key database would compromise the system. If the units are resold, the information would need to be distributed, leading to an increased dilution of the security of the product, and increasing the probability of the information being released into the open market.

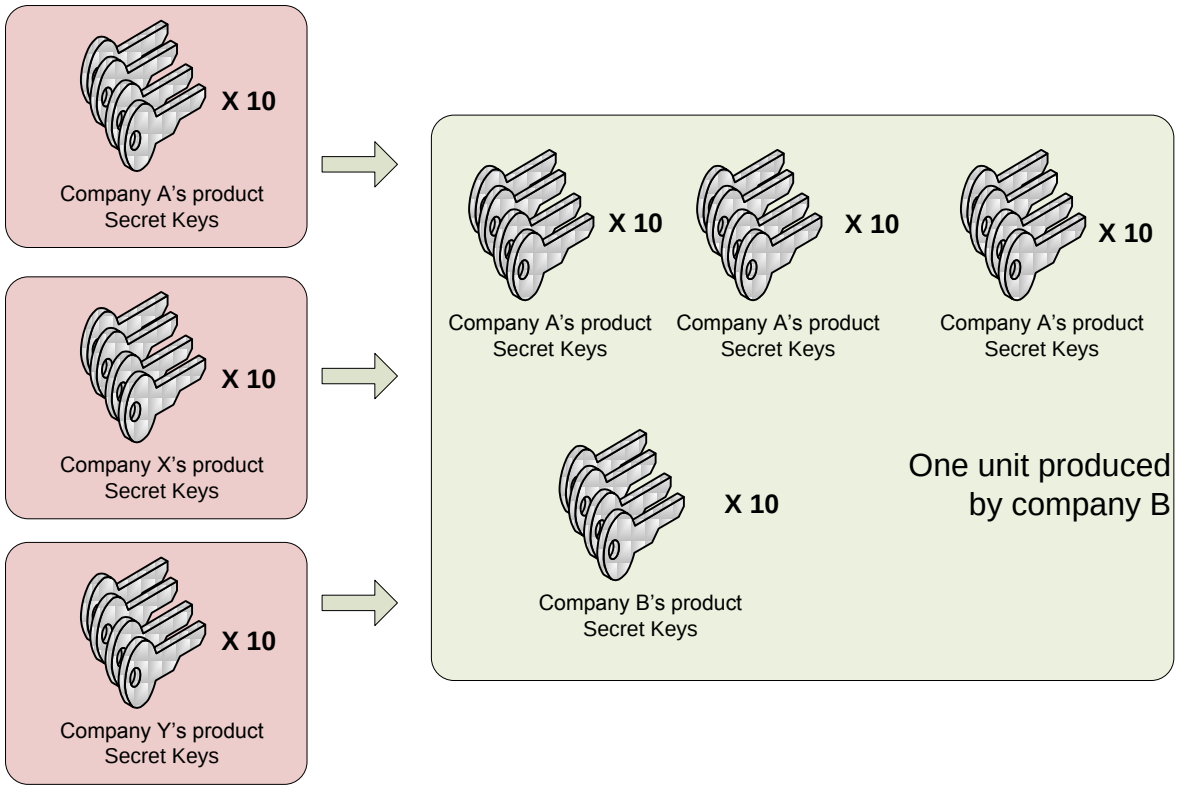

Figure 2.3 Illustration of Symmetric Key Distribution Problem

2.4.3 Public Key Algorithms: A Solution to Distributing Key Information

The answer to the problems addressed above can be found through the use of public key cryptosystems. Through judicious use, it is possible for each company to have several

public/private key pairs that enable them to securely interact with the tag without being prone to eavesdropping or man-in-the-middle attacks. This has the great advantage that each company only needs to keep only a few public/private key pair, and not numerous keys to communicate with each unique tag. This reduces the overall key management burden. In addition, the information regarding key management can be upgraded to be of more use through the integration of a PKI system in the future. By sharing the public key, the tag can respond securely by encrypting the information with the public key and replying with the enciphered message. Since the public key is available to anyone there is no fear of transmitting the public key information over the air to the tag.

In a cryptographic handshake using public key algorithms there are two primary goals: to disguise a given message, and to verify the identity of a specific identity. In the first case, this would be the condition where the tag has a message (assumed to be the tag identifier or similar bit length item) to the reader. The second case is the use of digital signatures used to authenticate in which case there is an exchange between the tag and reader of public key and private key generated information.

The first case assumes that there is always a transaction between the reader and tag flowing from the tag to the reader, not the other way around. In other words, the tag contains the information that the reader needs. In this case we would like to protect the information that is traversing the medium (air). To properly protect the information, the reader transmits its public key information to the tag, the tag uses it to encrypt the message without translating back from Montgomery space, and the reader would then translate the information into normal form where it would then use the resultant to decrypt the message using its private

key. This exchange is well defined under the premises of the proposed distributed cryptographic computation.

The second case, verifying the identity of the tag, could be performed but would need to be defined more succinctly. In this case, a set of attributes are digitally hashed to the private key of the tag and then separately encrypted and then multiplied together to compute the digital signature. The cryptographic unit that is being proposed could be used to support this purpose, but a significant amount of additional hardware would be required to enable this functionality which is beyond the scope of this thesis to analyze. The possibilities for this type of implementation would lend itself to being able to identify fraudulent tags in the network, and to authenticate the results that are being passed. Other methods have been proposed that are able to perform this capability, but generally with less flexibility than public key digital signature architectures. In future, when transistors can be fashioned in smaller packages requiring less space and power this could be an additional option to consider. The cryptographic system that is being suggested could be used to augment a public digital signature scheme.

2.4.4 Level of Security

For the next generation of RFID tags, there needs to be a reasonable assurance that the algorithms that are going to lead our next iteration of this technology are secure, and flexible to meet our ever growing needs. The key attribute is security. This needs to be properly quantified to identify the level of security that would be offered by the considered RSA implementation. To do this a proper examination of the definition of modern security is established in the next section. The required key size for a given length plaintext is defined through an exploration of complexity theory and security as it is currently understood.

2.5 Complexity Theory and Security

"In complexity theory the goal is to prove that difficult problems cannot be solved with modest resources," [**38**]. This theory is pivotal to the implementation of secure algorithms in the context of cryptographic security within the modern context. This definition was promulgated through the seminal work of Goldwasser and Micali, who reduced the requirements of security from what was initially defined as a perfect cryptosystem by Claude Shannon. The following section attempts to identify what security means in a public key cryptosystem and how a modified RSA algorithm will be able to achieve this level of security.

2.5.1 Perfectly Secure Systems

Shannon described a perfectly secure system as being a "one-time pad". This is described as being a rolling key that does not repeat its past patterns (is non-periodic). Other criteria that make a perfectly secure system include:

1. A key chosen completely at random
2. A key as long as the plaintext
3. A key that is used only once.

This one-time pad is the description of a system that provides information-theoretic security. This type of security is extremely difficult to implement since it requires a completely random key generation scheme. True randomness is extremely difficult to produce. As of yet there has yet to be a method of generating random numbers that can be used en masse for a communications channel where both ends are aware of the random number algorithmically. One time pads are also an unrealistic method of providing security due to

the vast quantities of information that are passed through electronic means. Since perfect security is impractical, there have been several definitions which have relaxed these information theoretic requirements.

2.5.2 Modern Cryptographic Definitions

In their seminal paper [**4**], a description of a new more flexible encryption scheme is suggested, and is the basis of many modern cryptographic encryption schemes. The major contribution of this paper is what stems from its initial statement "Whatever is efficiently computable about the clear text given the ciphertext, is also efficiently computable without the ciphertext," [**4**] . This postulation brings about a new dimension of security, namely semantic security and indistinguishability of encryptions, which is later reduced to being the same postulate under chosen plaintext attack models in [**39**]. Loosely speaking, this suggests that security can be maintained through the use of hard computational problems. The definition of what is hard is described in [**4**].

Problems that have been identified in computational theory as being difficult make up a class of problems called nondeterministic polynomial time problems or simply NP. This class of problems encompasses the set of polynomial time (P) and nondeterministic polynomial time complete (NP-complete) problems. Problems defined as being from the NP-complete set are used primarily in security applications. It should be noted that although it is generally thought that NP-complete problems are difficult, measuring security against the NP-complete benchmark only ensures that the algorithm is as secure as any other NP-complete problem, and not an absolute assurance of security [**40**].

NP-complete sets were first defined in 1971 [**41**], wherein a problem is required to have two basic properties to be considered part of the set. The first property, given a problem L, is that it must be part of the larger set NP. The second property is that every problem in NP must be reducible to L in polynomial time. What this means in more common terms is that the problem L can be verified quickly in polynomial time, and through transformation any NP problem can be reduced to a problem L in polynomial time.

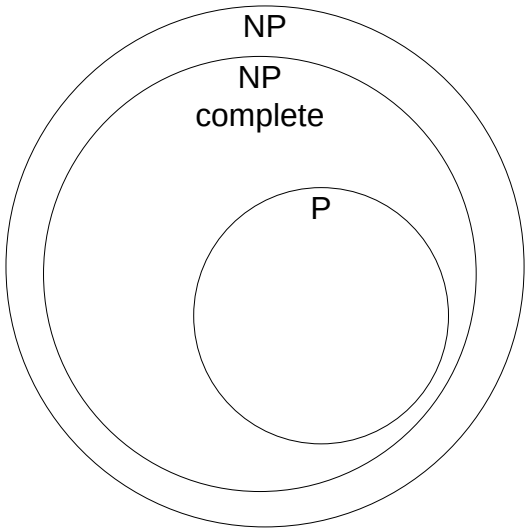

Figure 2.4 NP, NP-complete, P sets

2.5.3 Integer Factoring

The RSA algorithm, which this thesis is based, has its foundations in the factoring problem. At this point it has not been properly determined where the factoring problems lies in the classes mentioned above. After several attempts it is assumed to lie outside of the P class of problems. The factoring problem is found to be in the intersection of the NP and co-NP class but it has still yet to be proved to be in P. This is reassuring due to the reliance of prime factorization in the implementation of many public key cryptosystems. If the prime factorization problem is found to be in the class P, this would indicate that there is a

polynomial time factorization algorithm that can generate the decryption prime from the public key. This discovery would be a great leap forward in computational analysis, but it would also mean that many of our encryption systems today would suffer gravely.

Assuming that prime factorization is an NP-hard problem many public key cryptosystems have been developed. One public key cryptographic algorithm is the RSA algorithm. The RSA algorithm can be shown to be secure under the security constraints that were defined under [**4**]. Semantic security is provided through the use of pseudo-randomly generated bits concatenated to the message to modulate the result of the modular exponentiation that is performed as the basis of the encryption and decryption algorithm.

2.5.4 Semantic Security

Semantic security, loosely speaking, defines that whatever can be efficiently learned about the plaintext from the ciphertext, can be learned without the ciphertext [**42**]. A technical definition of the previous statement would be the following:

"An encryption scheme, (G,E,D), is semantically secure (in the public-key model) if for every probabilistic polynomial-time algorithm A, there exists a probabilistic polynomial-time algorithm A' such that for every $\{X_n\}_{n\in\mathbb{N}}, f, h, p$ and n as in Definition 5.2.1," [**42**].

$$\Pr\left[A\left(1^n, G_1(1^n), E_{G_1(1^n)}(X_n), 1^{|X_n|}, h(1^n, X_n)\right) = f(1^n, X_n)\right] - \Pr\left[A'\left(1^n, 1^{|X_n|}, h(1^n, X_n)\right) = f(1^n, X_n)\right] < \frac{1}{p(n)} \tag{2.3}$$

What was described in definition 5.2.1 in the quotation above were the quantities for $\{X_n\}_{n\in\mathbb{N}}, f, h, p$ and n. $\{X_n\}_{n\in\mathbb{N}}$ is the set of probability ensembles with $|X_n| \leq$ poly(n), where n is the security parameter. The functions f, h are a pair of polynomially bounded

functions such that $f, h: \{0,1\}^* \rightarrow \{0,1\}^*$, for every positive polynomial p and sufficiently large n. Essentially, equation (2.3) requires that there be a negligible quantity (given polynomial p(n) is large) in which an adversary could correctly guess the message.

CHAPTER 3 – IMPLEMENTATIONS AND CONSTRAINTS

3.1 RSA Algorithm

The RSA algorithm was developed by Rivest, Shamir, and Adleman, at MIT in 1977 [**43**]. Since its conception, the algorithm has steadily gained acceptance as a secure method of encrypting information when properly implemented.

In cryptography there are essentially two overarching architectures to encrypting messages across a medium, namely public key (asymmetric) or secret key (symmetric) algorithms. In the case of a symmetric key implementations, the channel is encrypted using a predefined key. This method requires that the key be shared initially by the parties through a secure medium to ensure communication is not compromised. In symmetric keys the two parties are able to use the key to encrypt and decrypt any message from the other party. If another party is able to obtain the key used to encrypt the channel, the other party will be able to decipher the plaintext of all messages that are being sent and received. In the public key scenario, two keys are used. One key is made public, while the other key is kept private. The public key enables a party to encrypt a message and send it to the holder of the private key. The private key holder is able to decrypt any messages sent using the public key. Given only the public key, it is difficult to determine an inverse of a public key encrypted message, such that the plain text message may be revealed. In this way, only the originator and the receiver of the message are able to communicate while other parties will find it difficult to determine the message.

The type of function that is being exploited in the production of a public/private key pair is a "trap-door one-way function". This type of function was first described in [**44**]. The term "trap-door one-way" function was coined due to the ease in which a computation could be completed one way and the difficulty of performing its inverse.

The RSA algorithm uses a number of specific variables to compute a result that is cryptographically transformed to satisfy the requirement of a trap-door one-way function. The algorithm is based on the large number factor problem which has been said to be computationally difficult. By creating a space defined within the constraints of a modular group composed of two large prime numbers and the hardness of the factor problem, it is possible to define a relation between the public key and the private key such that they will reveal the message.

3.1.1 Cyclic Groups

Cyclic groups form an important foundation in group theory. A cyclic group is an abelian group that can be generated by a single element. Given a specific order, *k*, it has the property that the generator returns to the identity unit when raised to the exponent *k*. In the additive case the identity is zero, wherein the multiplicative case the identity is one.

In RSA, the multiplicative cyclic groups are the area of interest. In the multiplicative case, there are pairs of elements which are capable of generating the identity when multiplied together (these pairs are the inverse of one another in the defined cyclic group). These pairs are integral to the RSA implementation, since they are used to encrypt and decrypt the message.

When dealing with prime based cyclic groups all integer elements up to *p* are represented in the group. This property allows all message up to the magnitude of the modulus (positive integers) to be properly represented in the group created by the bi-prime factor modulus. This is further explained through the use of Euler's totient function.

3.1.2 Euler's Totient Function

Leonard Euler came up with a method of determining the number of elements less than or equal to the value in the function that are co-prime with the number being evaluated. The totient is usually represented by the Greek letter phi (φ). By identifying the number of elements in the ring, it equivalently defines the order of the group.

Euler was able to show the following is true for all values *a* relatively prime to *n* (this is a generalization of Fermat's little theorem).

$$a^{\varphi(n)} \equiv 1 \; mod \; n \tag{3.1}$$

3.1.3 RSA Algorithm Applied

The RSA Algorithm makes use of the properties that were outlined above. Euler's totient function is the basis on which the encryption and decryption process is performed.

For this section some variables are defined which have the following properties:

$$p \; and \; q \; are \; large \; prime \; integers$$

$$n = \{n | n = pq, n \in \mathbb{Z}\} \tag{3.2}$$

$$e = \left\{e \middle| \; 0 < e < n, \gcd\left(e, \varphi(n)\right) = 1, e \in \mathbb{Z}\right\} \tag{3.3}$$

$$d = \{d |\; ed \equiv (1 \; mod \; \varphi(n)), 0 < d < n, d \in \mathbb{Z}\} \tag{3.4}$$

$$m = \{m | m < |n|, m \in \mathbb{Z}\} \tag{3.5}$$

Initially, two large prime numbers are chosen randomly such that when they are multiplied together they result in a value which is equal to the security parameter in magnitude. A large number is said to be a positive integer which has a magnitude greater than or equal to half the magnitude of the security parameter in the case above. The security parameter is represented in bits and is defined as the number of bits that are used to encrypt the message. For example, if the algorithm is to encrypt the message into a 128-bit message, then the security parameter is 128 bits. When indicating that a 128-bit encryption is being used, this also refers to the size of the modulus. The modulus is defined as being the product of two primes *p* and *q*:

$$n = pq \tag{3.6}$$

The length in bits of p and q dictate the length of the modulus n through the following relation:

$$|n| = |p| \cdot |q \tag{3.7}$$

Once n has been calculated, the values of the public and private keys can be determined through the evaluation of Euler's Totient function.

$$\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q)\left(\frac{d}{\varphi(d)}\right) = (p-1)(q-1), \tag{3.8}$$

$$where\ d = \gcd(p, q)$$

Consequently, two co-prime numbers to phi must be chosen such that they are prime relative to Euler's totient. In general, the exponent of the public key is chosen randomly among the set of co-prime values with respect to phi. Once chosen, the decryption exponent is found such that it is the cyclic inverse of the encryption exponent in the group phi.

$$e \cdot d = 1 \; mod \; \varphi(n) = S + 1, \text{where S is a multiple of } \varphi(n) \tag{3.9}$$

The following demonstrates how it is possible to reconstruct a message which is first encrypted with the encryption exponent *e* and then subsequently decrypted using the exponent *d*.

$$m^{ed} mod \; n = \; m^{(S+1)} mod \; n = m^S m^1 mod \; n = (m^S mod \; n)(m^1) mod \; n \tag{3.10}$$

$$= (1)(m^1) \; mod \; n$$

To encrypt a message the pair (e, n) is transmitted publicly to those who wish to send a message. The following is computed:

$$c = m^e \; mod \; n \tag{3.11}$$

The message originator would then transfer the value of *c* to the destined recipient. Once received the recipient makes use of their private key pair (d,n) to decrypt the message.

$$m = c^d \; mod \; n \tag{3.12}$$

Without the decryption prime it is deemed computationally infeasible to determine the message. The algorithm can be compromised through the use of a factoring algorithm which is able to determine the prime factors of the modulus and subsequently determine the values of the encryption and decryption exponents through the use of Euler's totient. Factoring large integers is a computationally intensive process and as long as the security parameter is chosen to be greater than the current computational power it is infeasible to compromise the encryption. It is therefore important to size the correct length of the modulus as computational power increases. The benefit of this algorithm is that adding a bit to the size of the modulus, doubles the complexity, thereby insuring an effective deterrent to an ever increasing adversarial threat.

3.1.4 RSA Algorithm Semantic Security

To satisfy the condition for semantic security a degree of randomness needs to be introduced into the RSA algorithm. This is achieved in the RSA cryptosystem [**45**] through the use of a trapdoor one-way permutation as first described in [**46**] and the use of random message padding.

A trapdoor one-way function describes a bijective function which is capable of performing a transformation from one space onto itself ($f: X \rightarrow X$), and for which it is computationally difficult to reverse the operation without a separate unique key.

Random bits are added to the original message to increase the number of possible encryptions for a given set {X}. To ensure that maximum number of encryptions which are available for a given input message a system called OAEP (Optimal Asymmetric Encryption Padding) has been developed to ensure that there are just as many encryptions as possibilities for the message. This method was first introduced by [**47**]. OAEP dictates that the minimum amount of padding that is required to make an encryption of the message is quire is equal to the length of the message itself, effectively doubling the size of a given message. Doing so ensures that there are as many possibilities for a ciphertext as there are possibilities for each message given a particular plaintext. This requires a message of length *n* to be encrypted into a ciphertext of length *2n*, with *n* bits of randomly or pseudo-randomly generated padding. In this manner we can ensure that the message is properly secured through the use of padding since it is now computationally infeasible to determine the original message as long as factoring of the primes used in the modulus is not feasible.

The message is computationally infeasible to determine due to the number of possibilities introduced by padding. It is assumed that the message can be represented through the use of bits of length n ($X \in \{1,0\}^n$). With this assumption, the ciphertext of a modular exponentiation would result in a single output. Given that an adversary, *A*, is performing a chosen plaintext attack as defined in [**4**], it is reasonable to assume that *A* is able to identify the difference between two plaintexts that are a small Euclidean distance apart less than the modulus. To prevent an information leak, padding of randomly generated bits is required to ensure that messages that are close are not detected in this manner. As shown in [**47**], the optimal magnitude of padding is found to be the magnitude of the message. A more formal proof of the Optimally Asymmetric Encryption Padding is given in [**40**].

3.2 Translating RSA Algorithm Realizable Hardware Implementation

The RSA algorithm is based on modular exponentiation. In hardware this is most readily performed through a series of modular multiplications. The modular multiplications are performed through a varied approach, generally using a specialized multiplication technique.

Exponentiation is performed by scanning each bit of the exponent and subsequently squaring the result or squaring and then multiplying the result. In this way the exponent acts as a control mechanism for the main arithmetic operations. Figure 3.1 shows a high level overview of how the algorithm would be implemented. The key to the implementation for the RFID case is the reuse of the modular multiplication component and the use of the exponent to act as a control mechanism.

Multiplication is at the heart of performing the RSA algorithm. Specifically, a means of efficiently performing a modular multiplication is required. Traditional binary

multiplication methods of length *n* and generate a result with length *2n*. Modular multiplication is required due to the exponentiation operation. Without a reduction, the multiplication would result in a n^e answer, which would surpass the memory available in most data centers in the world. This is why it is critical to perform the reduction during the multiplication process. There are various methods of performing this reduction which are described fully in the proceeding sections.

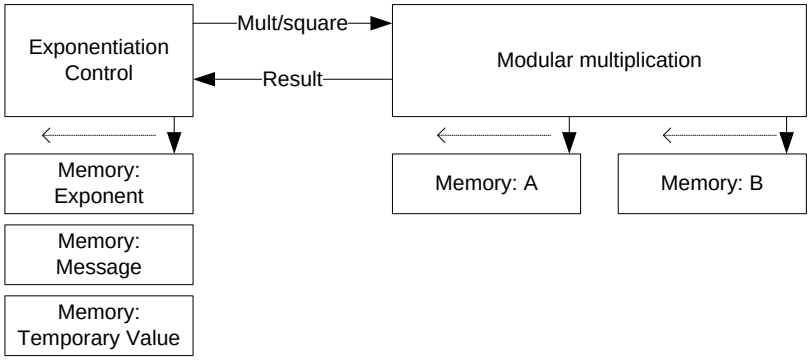

Figure 3.1 High Level Description of Algorithm

3.2.1 Time and area trade-off

There has been a lot of work which has examined the time area trade-off on FPGA implementations of various multiplier structures. It is generally agreed that the more area dedicated to the circuit the faster in which the algorithm can be performed. This is due to an increased number of resources which can be dedicated to perform the required calculations in parallel in a single clock cycle several. This advantage is offset by a larger area and power consumption on the target technology. A parallel architecture would be considered too large for the desired RFID tag implementation since space and power are at a premium. The implementations that will be considered will need to be sequential in nature as indicated by [**48**]. This is due to the area and power constraint that is being forced on the design. A

more in depth examination of this constraint is provided in the 3.3 Design Constraints on current RFID tag section.

There are three general architectures that have been described in past implementations. These include sequential, parallel, and systolic. It has been found that each architecture increases the throughput of the final answer. It has been found that depending on the security parameter an optimum time area product can be achieved. In [**48**], it was found for operand sizes that are less than 256 the optimum architecture is a serial multiplication algorithm.

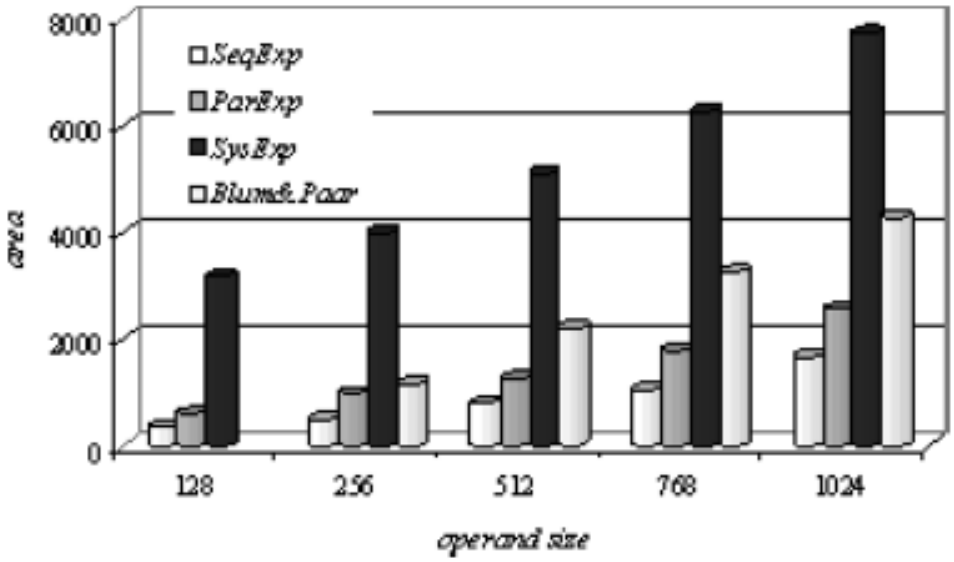

Figure 3.2 Comparison of the hardware area required by sequential, parallel, systolic and Blum & Paar Implementations [48**]**

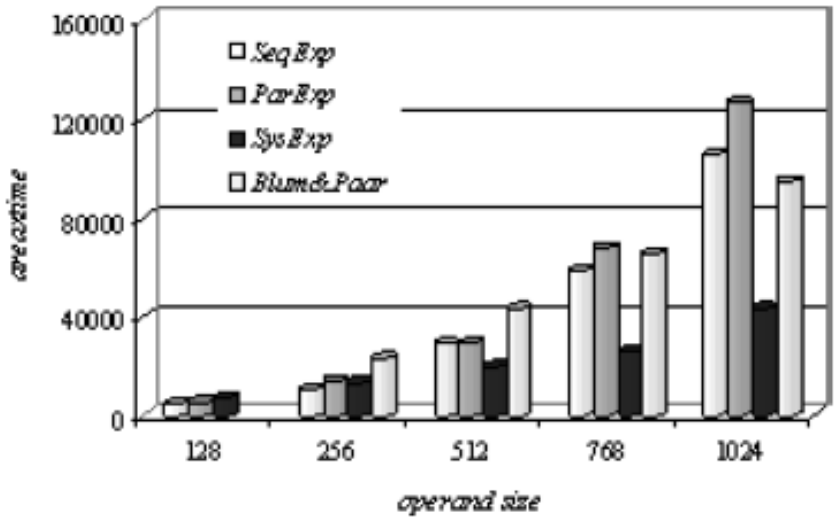

Figure 3.3 Comparison of the area x time factor for sequential, parallel, systolic and Blum & Paar Implementations [48**]**

3.2.2 Parts of the RSA algorithm

3.2.2.1 Modular Multiplication

Multiplication is an integral part of performing the RSA algorithm. The method of performing multiplication is critical due to it being the central iterative operation when performing an exponentiation. There are several methods of performing multiplication in an embedded application. The type of multiplication that is required is unique since it needs to be multiplication modulo *n*. This type of multiplication can be performed in many ways. There are algorithms that perform the modular division after calculating the result of a multiplied set of values, and there are those that perform the modular division as an intermediate step when performing each addition.

In [**49**], it was found that multiplication on an embedded system when performing exponentiation could be more efficiently performed by translating the multiplicand and

multiplier into a representation that would restrict the use of trivial divisions. Although initially there is an upfront cost to perform the initial transformation of the results into a residue format, the algorithm makes it more efficient to perform multiple modular multiplications for the purpose for exponentiation. When considering exponentiation, the suggested algorithm is a perfect fit for the desired result, although this an efficient means of performing a single modular multiplication. The algorithm is faster than traditional modular multiplication, and it could potentially use less space during implementation.

3.2.2.2 Montgomery Multiplication

Montgomery modifies the elements that are used as standard inputs into an RSA cryptosystem and modifies them so that trial divisions are not required. It also generally requires the pre-computation of specific values such as n' and r^{-1}.

$$function\ MonPro$$

$$step1.\ t \coloneqq \bar{a} \cdot \bar{b}$$

$$step2.\ u \coloneqq (t + t \cdot n' mod(r)) \cdot n)/r$$

$$step3.\ if\ u \geq n\ then\ return\ u - n\ esle\ return\ u$$

[**50**]

The first step performs a multiplication of two transformed values $\bar{a}$ and $\bar{b}$, where $\bar{x} = x \cdot r\ mod\ n$. The second step performs a modular *n'* residue calculation to transform step 1 into a result which can be readily divided by the value of r. The last step ensures that the value is not greater than the modulus and subtracts if required.

The algorithm above is the general Montgomery multiplication algorithm although there have been several minor variations on the procedure to gain efficiencies in speed and area

utilization. The different implementations of the function can be attributed to the methods and order used to perform the operations. Broadly, the algorithms can be split up into two categories: Separated and Integrated. Separated algorithms first multiply and then perform the modular division to determine the result. Integrated algorithms perform the division at intermediary points of the multiplication, prior to the calculation of the final result. There are some algorithms that will wait until the end of an array of words to perform the reduction (coarse), while others perform the reduction at each word (fine). There are also algorithms that will straddle the two extremes which are called hybrids [**50**]. This section will investigate the method of Montgomery multiplication and its different methods to determine whether one of them will be considered as a plausible algorithm for the RSA implementation.

3.2.2.2.1 Seperated Operand Scanning (SOS)Method

In the SOS method the product "a*b" is computed and then divided it by the modulus to obtain the result. As shown in [**50**] this requires $2s^2+s$ multiplications, $4s^2+4s+2$ additions, $6s^2+7s+3$ reads, and $2s^2+6s+2$ writes. The method also requires a total of 2s+2 words for temporary results, which are used to store the $(2s_1)$-word array *t* and the one-word variable *m*. There are also numerous pre-calculations that are required to obtain the result, which include the value r' and the value of n'. For the calculation another value,n'_o, is required which is derived from the relation shown below. This method would not be suitable for a single modular multiplication due to the amount of overhead required to perform it, but does lend itself to use in modular exponentiation. The intermediary values can be determined using the following equations:

$$m \coloneqq t \cdot n' mod(r) \tag{3.13}$$

$$r^{-1} \cdot r \coloneqq 1 \; mod \; (n) \tag{3.14}$$

$$r^{-1} \cdot r - nn' = \; 1 \tag{3.15}$$

In [**51**] it was found that for an FPGA implementation of the SOS method where the word size was flexible, a savings of 58.5% can be achieved by using a word size of 32 bits instead of 64 bits. The data throughput was increased accordingly. For a word size of 16 bits, the clock was 135.14 MHz, with 1022 slices, a single 18x18 multiplier and took 449 clock cycles to complete one multiplication.

3.2.2.2.2 Coarsely Integrated Operand Scanning (CIOS) Method

In this algorithm for every intermediary result of the multiplication algorithm, a reduction is performed to improve the number of logic memory units needed to compare to the SOS method. This can be done since the temporary values are finished computing at the end of each intermediary step, so reduction instructions can be implemented right after the initial computation. The CIOS method (with slight improvements as indicated by [**50**]) requires $2s^2+s$ multiplications, $4s^2+4s+2$ additions, $6s^2+7s+s$ reads, and $2s^2+5s+1$ writes, and uses only s+3 words of memory space [**50**]. This is called CIOS because the inner multiplication (intermediate step) is performed and is then reduced in the overarching loop.

The FPGA implementation as described by McIvor [**51**] found that there was a significant savings in terms of silicon area when using a CIOS method for multiplication. Using 717 slices with a 163.53MHz clock and one 18x18 bit multiplier, his team was able to achieve a result in 310 clock cycles producing a throughput of 67.52Mb/s when multiplying 128 bit operands split into 16 bit words. This is a significant result, and will be used to determine the best method of implementing an RSA algorithm onto an RFID chip.

3.2.2.2.3 Finely Integrated Operand Scanning (FIOS) Method

The FIOS method differs from the CIOS method in that it only has one inner loop that performs the multiplication. This results in both the multiplication and addition being performed in the same loop. The CIOS method has two inner loops whereas the FIOS method only has one. This is an advantage in terms of computation time, but this advantage is balanced by the requirement to call an addition function for iterations of the inner loop [**50**]. This method requires $2s^2+s$ multiplications, $5s^2+3s+2$ additions, $7s^2+5s+2$ reads, and $3s^2+4s+1$ writes. The total temporary space required is s+3 words.

In an FPGA implementation, similar results to the CIOS implementation were found. With 128 bit operands (8 words with a length of 16 bits), a result was obtained in 318 clock cycles with 836 slices and a single 18x18 bit multiplier (fast look-ahead carry save adders). The clock speed was also significantly less at 111 MHz.

3.2.2.2.4 Fast method for Montgomery's modular multiplication

The paper by McLaughlin [**52**] describes a method which could be used to quicken the Montgomery Method. It should be noted that it does require additional calculations prior to performing the multiplication. In the paper, a table describes the algorithm with these overhead calculations and shows that it is slower than the Karatsuba-Ofman multiplication algorithm.

3.2.2.2.5 Variations on the Montgomery Multiplication Algorithm

For many of the multiplication algorithms methods of improving the algorithm can usually be derived. In the case of Celebi, [**53**], they show that it is possible to perform additional parallel computations in some cases. They also show that deeper pipeline stages can improve the speed at which the process is executed without increasing the amount of

occupied silicon area. As suggested by Rijnders [**54**] [**54**], there are methods to optimize the addition chain that can be used to optimize the silicon area. In the case of implementing a systolic array for multiplication, Fournaris' solution [**55**] [**55**] provides a scalable systolic array which is generally thought to be greater in silicon area than a traditional Montgomery implementation. Although the implementation that is used seems to be quite large for a 128 bit implementation, using smaller word sizes and the same architecture might yield a useable construct to perform a CIOS Montgomery multiplication.

3.2.2.2.6 Serial Multiplication

"Serial multiplication, sometimes referred to as "MSB-First multiplier," is a polynomial basis multiplier that uses k processors (or slices) and computes GF(2^k) multiplication in k cycles," [**56**] [**56**] . In serial implementations, the multiplication is performed on a bit by bit basis. This is generally the smallest implementation that is possible due to the low requirement of transforming the input into a useable form. It is also one of the slower implementations due to the bit scanning required to perform the multiplication. Some improvements have been suggested, but they generally increase the amount of implemented hardware required. One method of increasing the speed of implementation was suggested by [**56**] [**56**], who suggests using columns of serial multipliers and implementing them in a fashion which would ensure that an intermediate operation would be performed in each "slice" of the multiplication architecture. The author was able to achieve serial multiplication for a 167 bit multiplicand with 420 function generators, 349 flip flops and 422 combinational logic blocks, with an average of 3.8μs per multiplication. As previously mentioned, this type of implementation increases the speed of the operation, but introduces additional hardware which take both power and implementation space. A serial

implementation would likely be the best method used to ensure a small size. Timing will need to be properly measured to ensure that performing the calculation can be done within the timing constraints of the protocols enforced.

3.2.2.2.7 Multiplication as a Series of Modular Additions

It is possible to perform multiplication as a series of additions. The advantage of breaking the multiplication down to modular additions is that it is now possible to reduce the number of memory elements required for each multiplication. Instead of a result that is 2N bits long it should be N-1 bits, where N is the modulus. This with the reduction of memory elements, a reduction of logic is also achieved. This type of multiplication is shown in [**57**].

3.2.2.2.8 Modifications to Montgomery Multiplication Algorithm

A study was performed on the benefit of reducing the subtraction from a multiplication algorithm [**58**]. It was found that both the area of implementation was reduced and the overall time for execution was reduced by simply removing the subtraction step from the Montgomery multiplication [**58**]. The suggestion was to perform a pre-computation to relay pertinent information including a derived unit from the modulus to perform the calculation without the need to perform any intermediary subtractions.

Ideally, no pre-computed reader translations would be necessary at the reader since this would be required prior to each communication with a tag. What would be more suitable would be a post computation if one is required at the reader end so that the reader could interrogate a tag and store the information for processing at a later time.

In traditional Montgomery multiplication there is a requirement to have the value of $r^2 mod\ n$ calculated prior to transmitting and multiplying the value of the message and the

temporary value. This is not a feasible translation on the RFID chip as it would require additional resources and time which are at a minimum. The suggestion is that the result be transmitted back to the RFID reader where it can perform a single translation for the computation of a final result since it will have additional computational power readily available to it.

3.2.3 Exponentiation

There are several ways in which exponentiation can be performed. The first method is to multiply the base of the power by itself iteratively exponent times. The algorithm for this is shown below:

Algorithm 1:

$$For\ (i = 0, i = i + 1, i < exponent)\{$$

$$T \leq T * base$$

$$\}$$

The number of multiplications in this algorithm would be equal to exponent. This method is not as efficient as other proposed algorithms which make use of a serialized method for exponentiation. Ideally, the number of multiplications required would be less than the magnitude of the exponent. Since we are working with binary systems another representation of the exponent can be utilized to minimize the number of multiplications. In binary format the exponent is represented as a series of bits with a power of two being the base. It is possible to exploit this through the use of squaring and multiplying a temporary result. By performing bit scanning it is possible to square the value and subsequently choose to multiply the result by the base if the exponent bit being scanned is a 1. In this way the

number of multiplications required reduces substantially. In the example below, instead of having to perform forty-two multiplications, the calculation can be performed in 7 multiplications, a substantial savings. This property is shown in the mathematical expression below. The algorithmic implementation is also provided below:

$$m^{42} = m^{(101010)_2} = (((((m^{2^2})m)^2)^2)m)^2 \tag{3.16}$$

Algorithm 2:

$$\begin{aligned}
&For\ (i = 0, i = i + 1, i < |exponent|)\\
&\{\\
&T \leq T * T;\\
&If\ e_i = 1\ then\\
&\quad T \leq T * base;\\
&\}
\end{aligned}$$

This algorithm allows us to skip unnecessary multiplications resulting in an overall improvement in the speed and size on chip of the algorithm. There are different variations to this algorithm. The first way is to read the bits of the exponent from left to right. If this is done, the overall architecture is much simpler than if a right to left method is used (assuming little-endian) [**59**]. The opposite would hold true if this was implemented using a big-endian scheme.

3.3 Design Constraints on current RFID tag

There are numerous constraints that affect the design of an RSA cryptosystem on an RFID. In the following analysis these have been roughly separated into physical and logical constraints. The physical constraints identify the size of the tag and the derived constraints including antenna size and power. The logical constraints consider the security of the algorithm and the requirements dictated by need to maintain a certain level of security.

3.3.1 Physical Constraints

RFIDs are physically constrained by industrial standards that dictate their size. This affects the size of the antenna and the power that can be transferred to device. This affects the number of transistors that can be implemented on a given technological standard, which directly influences the size of the RSA cryptosystem that could be feasibly implemented on a given architecture. This section defines the power that is delivered to the tag at a given distance and indicates the number of gates that can implemented given the available power.

3.3.1.1 Assumptions

When defining constraints of the RSA RFID design problem it is important to confine the problem to a specific technology and industry standards as some variables significantly affect the final analysis. The assumptions are listed below:

1. ISO 15693 prevails
2. 0.35um CMOS technology
3. Size of tag cannot exceed the dimensions allocated in ISO 7816

3.3.1.2 Maximum Distance of Operation

The laws of physics dictate the amount of energy that can be used by the tag at a given distance. Specifically, we are bound by the limitations of the energy to be transferred inductively to the tag. To be inductively charged there is necessarily a reactive magnetic field which induces a current in the tag's coil; the magnitude of which determines the potential that can be derived from it. This characteristic is based on the equations of Maxwell and the work of Henri Hertz. Since the magnetic field is a dominant term when close to the antenna, this is where the greatest magnitude of energy transfer to the tag will occur. As distance is increased from the antenna the magnetic field is replaced by

dominance in the electric field. Taking this into account, it has been shown that the point at which the magnetic field loses its dominance is at the distance shown in the equation (3.17).

$$d \geq \frac{\lambda}{2\pi} \tag{3.17}$$

Given the frequency of operation is 13.56MHz (±7kHz) [**60**], the wavelength is 22.12m giving 3.5m as a maximum. This is a rough theoretical maximum distance at which the tag will be inductively charged and does not consider any other characteristics of the tag or the reader.

It should be noted that this approximation assumes that the tag is normal to the incident magnetic field. Orientation plays a significant role in the power induced by the tag. As the tag moves from a normal position, the resultant incident power reduces in a sinusoidal manner. Since the orientation of the tag is assumed to be orthogonal, the magnetic field intensity can be expressed as described in equation 3.18 ([**1**], [**61**]), having a direct relation to the current, distance and the radius of the loop.

$$\vec{H} = \frac{N_1 I a^2}{2(a^2 + z^2)^{\frac{3}{2}}}\vec{z} \tag{3.18}$$

According to the ISO/IEC standard for vicinity cards the magnetic field for RFID readers requires that the field intensity be between 150 mA/m and 5A/m RMS [**60**]. The figure below shows the magnetic flux produced by various reader antennae dimensions assuming a loop configuration. The radius for each of the antennae varies from 6cm to 18cm. As the radius of the loop increases, the initial magnetic flux decreases. The trade off, apparent in the figure below, is that as the radius of the antenna increases, the initial magnetic flux decreases but slopes more gradually toward zero than those antennae with smaller radii.

Larger antennas generally are able to activate tags at a greater distance than those with small radii.

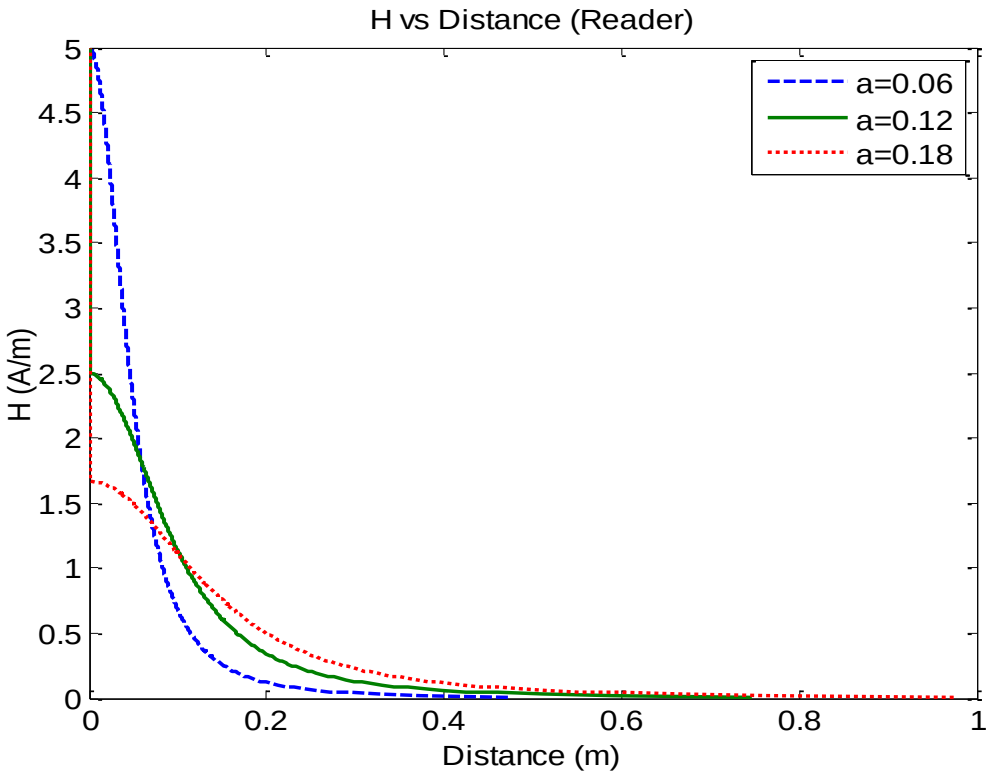

Figure 3.4 Magnetic Field Intensity Induced vs. Distance

Understanding antenna characteristics in terms of magnetic flux yield at a given distance is important to define the minimum magnetic flux required to activate the tag. It is possible to determine this value by assuming that the characteristics for the 13.56MHz near field magnetic coupling approximates a loosely coupled transformer. Since we are dealing with the near field propagation of the magnetic field, well known transformer equations derived from the use of Maxwell's equations as well as the use of Faraday's Law can be used. Equation 3.19 [**62**] approximates the voltage potential at the terminal of the antenna with a given magnetic field intensity.

$$V_{tag} = 2\pi f NSQBcos(\alpha) \tag{3.19}$$

Where the variable quantities are defined as follows:

$$f - frequency\ \ (MHz)$$

$$N - number\ of\ turns\ in\ the\ antenna\ coil$$

$$S - surface\ area\ of\ the\ tag\ (m^2)$$

$$B - magnetic\ field\ intensity\ \left(T \rightarrow \frac{Ns}{Cm}\right)$$

$$Q - Quality\ factor\ \left(Q = R\sqrt{\frac{C}{L}} = \frac{f_o}{BW}\right)$$

$$\alpha - angle\ of\ tag\ to\ the\ incident\ magnetic\ field$$

From the application note [**62**], the quality factor is of primary importance when attempting to determine the voltage and resonance characteristics of the antenna. The higher value Q achieves the greater the resonance of the circuit. This directly affects the bandwidth of the antenna and the voltage available at the tag. This is apparent in the equation above. In general, it can be said that the tags will have a high Q value (40), since they do not require a wide bandwidth. Also a higher Q value will positively impact the read range of the tag.

In [**63**], the standard defines the maximum size of an identification card to be 85.6 mm W x 53.97 mm H x 0.78 mm D. The maximum area of the card with this restriction is 4619 mm^2.

For many of the RFID operations that are performed, a minimum voltage potential is required to activate the tag. This minimum potential is dependent on the type of architecture implemented and also the type of operation that is desired. To read memory a tag requires

2.4V DC power [**62**], [**64**]. This corresponds to $4V_{rms}$ produced by the tag. The figure below provides a relationship between the desired voltage and the required magnetic field intensity.

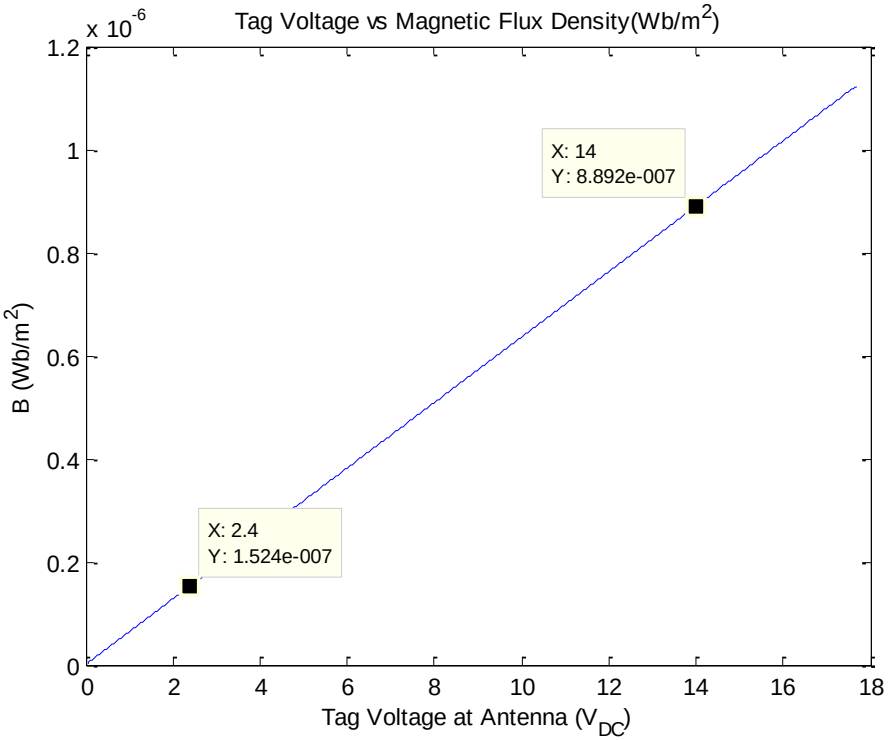

Figure 3.5 Tag voltage versus Magnetic Flux Density

The maximum distance and read range is defined as being the distance at which a $1.5422 * 10^{-7} Wb/m^2$ field can be induced on the tag antenna. This parameter depends on the following parameters as described by [**62**]:

- Reader and tag antenna performance
- Q of the antenna and tuning
- Antenna Orientation

- Excitation current
- Sensitivity of receiver
- Operating Environment

The other data point in the graph represents the average voltage required to write to an EEPROM (non-volatile memory). The implemented algorithm will not use non-volatile memory for its implementation due to the requirement to power the chip with 14V to perform a write operation to EEPROM [**62**], [**64**].

From previous experimental results with common tag architectures it was found that the power varied as a function of distance in following manner.

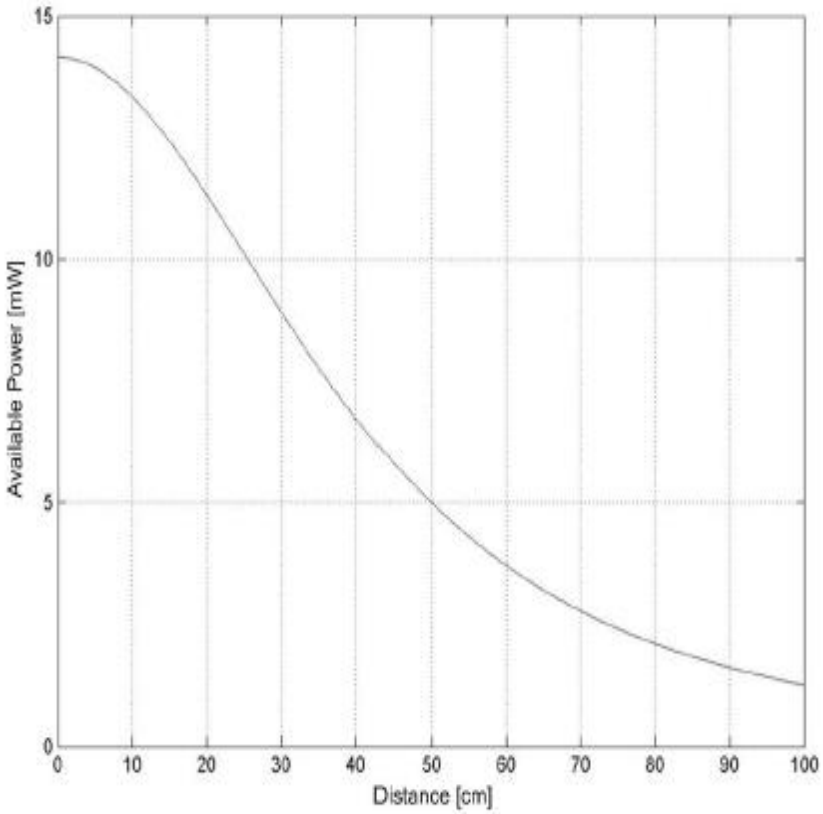

Figure 3.6 Available Power at the Tag vs Distance [1**]**

3.3.1.3 Transistor Power

To determine the number of transistors that can be placed onto an RFID tag for passive operation it is important to take into account the power characteristics of transistors. Although there are numerous types of transistors that can be used, MOSFETs are the most common due to their good isolation characteristics and their ease to manufacture in small sizes. The smallest transistor width built to date in a commercial fashion is the 0.18um transistor. For most RFID applications the 0.35um transistor is used. MOSFETs when used in CMOS applications are configured in a fashion that utilizes the least amount of power. This is accomplished through the use of negatively doped and positively doped regions inserted into the dielectric. Although the ideal case would be when there is little to no current which is used by the transistor, this is not the case. The transistors do consume power when they switch between states. The table below was taken from an analysis of power consumption of transistors placed into common CMOS logic configurations.

Table 3.1 Power Usage for Common Logic Building Blocks [1]

Type	Voltage	Power
Shift Registers	3.3	140.18 fWs
Xor gates	3.3	90.03 fWs
Nand gate	3.3	42.57 fWs

3.3.1.3.1 Maximum number of gates

Given a distance of 50 centimetres, the available power at the tag would be approximately 5 mW.

If the logic is clocked with 6.78 MHz (half the rate of the reader's transmit frequency) equation (3.20) can be used to approximate the energy that is available per clock cycle.

$$E_{clock} = \frac{P(d)}{f_{clock}} \tag{3.20}$$

It was determined that the energy per clock cycle is 737.46 pWs. This energy is then divided by the consumption of a single 0.35μm NAND gate to determine the maximum number of NAND gates which is given in equation (3.21).

$$N_{gates} = \frac{E_{clock}}{E_{NAND}} \tag{3.21}$$

With this calculation we arrive at the number of possible gates for this implementation as being 17,323 gates.

3.3.1.4 Power and gate count

From the work that has been described above, it can be shown that a limit on the number of transistors that can be used for a given RFID implementation is highly dependent on numerous criteria. Of greatest import is the available power delivered to the tag. This power is highly dependent on the link budget defined by the magnetic coupling of the reader and the tag antennas. As the radius between the reader and the tag increases, this reduces the number of transistors that can be implemented. Distance away, antenna configuration, matching and other factors beyond the scope of this thesis are extremely important when dealing with constraining the number of logic gates which are available for use. That being said, the calculations above indicate that at a given distance and assuming perfect matching between the antennas, it is possible to have an RFID which can power 17,323 NAND gates during a single clock cycle. This will be taken as the upper limit for the number of gates possible for any given implementation.

3.3.2 Timing Constraints

Timing is an issue when dealing with RFID tags. The benefits of RFID tags are the number of tags that can be read in a given period of time. When dealing with encryption, a write and read cycle is required. The process of selecting and reading a tag is governed by a tree based search algorithm, making use of bit masks to identify RFID tags available in the environment.

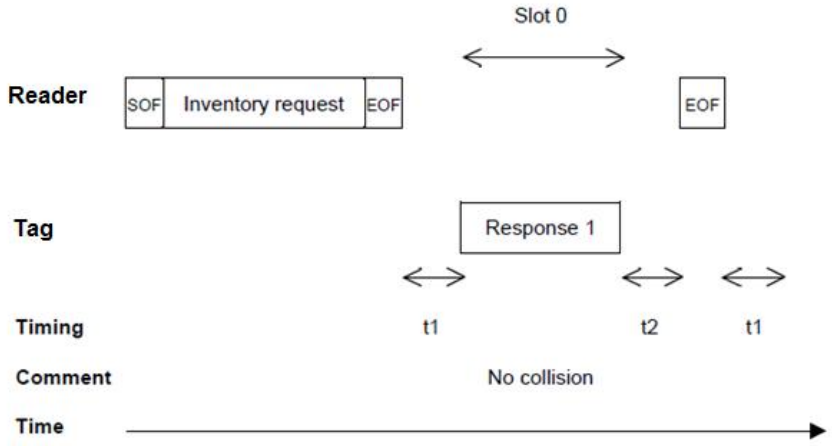

Figure 3.7 Read Protocol Diagram for Successful Read with No Collision [13]

For a given read request from the RFID reader, there may be a number of tags that respond causing a collision. This will increase the time that is required to read a tag. Figure 3.7 shows the case without collision with the associated times reported in Table 3.2.

Table 3.2 Time allocated for reading a tag under ISO15693 [13]

Sequence	Minimum	Nominal	Maximum
T1	309.2µs	311.5µs	313.9µs
T2	309.2µs		

During the writing process, it is defined in [**13**] that the write process cannot exceed 20 ms. The public key algorithm requires two values to be written to the tag: the exponent and modulus. It is critical to ensure that the public key variables that are transmitted to the tag are accurate. A prior study determined that to ensure accuracy of transmission (which is required for proper encryption), an increase from 10ms to 100ms is required to write to the tag on average [**65**]. This is due to increased retransmission time required when the tag does not receive the correct bit sequence due to environmental noise or propagation limitations. This reduces the number of tags that may be read in one second, but does not increase the time that is allotted to the cryptographic unit. Since there is still a 20 ms limit on the write process, and in [**65**] it was nominally determined that a write required 10 ms, the cryptographic unit has 10.9 ms to complete its computation and respond to the tag (10ms with additional nominal overhead between transmissions as indicated in Table 3.2).

3.3.3 Logical Constraints

Given the information in the previous sections it is possible to identify some of the requirements that are forced on the cryptographic engine from a logical point of view. To ensure that there is adequate security for a given message an upper limit for the message size must be given. A random generator needs to be created to perform the random bit padding required for public key security.

EPC (Electronic Product Codes) are generally used in conjunction with RFIDs to provide a means of identifying products. In this standard [**66**], the maximum bit length is given as 96 bits although smaller implementations do exist. This clearly determines the requirement for random padding that will need to be implemented in a secure implementation of the RSA cryptosystem on an RFID. Since the number of bits of the EPC is 96, an additional 96 bits

are required to pad the message if OAEP is to be considered. This makes the total ciphertext length 192 bits at minimum. This would ensure that there is negligible probability of the adversary divining the correct message from a set {X}.

With regard to the randomized values required for the padding, this can be implemented in a number of different ways. Although a full description of random number generator implementations is beyond the scope of this thesis, a few examples are provided in [**67**]. It can be seen that random number generators vary in their randomness and their size. The "true" random number generator that is suggested in the paper would be recommended due to its small size and the ability to closely approximate a random sequence.

It can be seen that these requirements place certain constraints on the design of the RFID cryptographic system. Of direct import is the requirement for additional bits to furnish the need for security of the enciphered message. Since the implementation is physically restricted by the power transmitted to the tag, every bit that is added to the computation uses up additional resources. These additional resources then reduce the operating range of the tags due the additional power that is required for the circuitry and the time that is required for computation.

CHAPTER 4 – PROPOSED ALGORITHM AND METHODOLOGY

4.1 Problem Description

To date, there has been little work to implement an RSA cryptosystem on an RFID tag. From the work that has been done, the majority of it is primarily concerned with increasing the overall speed of a system on an FPGA, with little consideration to the increase of area and power. The converse consideration is considered, one that weighs area and power above speed, when designing an RSA cryptosystem for an RFID implementation.

Most of the work that has been done in this field looks at increasing the output speed in computing the cryptographic results in on the FPGA platform. Size is rarely a main concern, being only limited by the specific chipset technology on which the cryptographic solution is implemented. There has been a concerted effort by a select few that try to explore the size, area, time relationship, but not on the RFID platform. This uniquely positions this research in the field to provide a pointed solution to a problem that faces next generation RFID solutions in the future.

4.2 Considerations

From the work that has been accomplished using the FPGA platform implementations, there are generally three distinct tradeoffs that must be taken into account when determining the design of a cryptographic system: time to compute the result, area of implementation and power. Attempts at defining a relationship between area, time, and power have been pursued, but these relationships are all technology specific. Since there is no one formula

that can be applied to the generic case, a heuristic approach must be undertaken to properly characterize these three tradeoffs for a given implementation.

Within the area of possible implementations of cryptographic processing units there are various algorithmic options, of which one was chosen for further study and implementation. The main operation that is required in the RSA algorithm is modular exponentiation. This requires an iterative modular multiplication to arrive at a final result. In the realm of efficient multiplication algorithms there are a few options, of specific merit are: traditional multiplication, Booth recoding Multiplication, and Montgomery multiplication. Through a review of literature it has generally been accepted that the Montgomery multiplication method is one of the more efficient methods of performing iterative multiplications for the purposes of exponentiation with potentially one of the smallest implementation sizes.

4.3 Montgomery Multiplication Application

Montgomery multiplication proves to be advantageous when constructing efficient multiplier architectures. When performing the Montgomery multiplication method, the multiplicand and multiplier are transformed into a common base, so that trial division does not need to be performed. Trial division is the process of examining integer divisors less than the dividend and steadily increasing them until the greatest common divisor is found. By performing the Montgomery transformation this process is avoided and it is possible to select a base that is easily divisible for a given architecture (in the binary case this is a base to the power of two). This transformation is required only once the computation is completed, and iterative multiplications can be performed without the need to perform any transformation of bases at each step. The result of a Montgomery multiplication is expressed as shown in equation 4.1 [**49**].

$$T = xyr^{-1} mod\ n \tag{4.1}$$

Apparent in the formulation, the result is reduced by a factor of r, where r is an arbitrary base to a given power. As previously mentioned, this base is reduced iteratively through each multiplication. To transform the result back into the normal domain, the result must be multiplied by a factor of $mod\ n$. The more efficient approach suggested by [**49**], is to perform the conversion at the end of the computation. The conversion process can be a resource intensive operation. Each modular multiplication operation exponentially increases the value of r that needs to be multiplied to return the resultant to the normal space representation. This predicates an efficient method of calculating the value of $r\ mod\ n$.

The value can be pre-computed or computed in tandem with modular multiplication. Considering these two options increases the number of computations that need to be performed by the tag significantly. Since the value of *r* increases at an exponential rate, the calculation of its result requires additional hardware to calculate the result in parallel. Although a serial implementation could be considered it would require approximately double the total time to compute the final result.

4.4 Montgomery Multiplication Implementation Methods

In the realm of Montgomery multiplication implementations there are generally three methods of performing the calculation. These three methods include: serial, parallel and systolic implementations. In order, they generally increase in the area consumed while reducing the overall time that is necessary to compute the result. This finding has been corroborated through the results that were presented by [**48**], where the area, time product for a modulus equal to or less than 256 bits was found to be most efficient in the serial Montgomery multiplication case when compared to the other architectures. With the RFID

implementation in mind it is proposed that a serial architecture would best suit the implementation of a next generation RFID cryptosystem.

Constraints on the system are not only with respect to the area that is utilized on chip, but also has a time dimension that needs to be taken into account. The algorithm will be tested with a set of test vectors. These vectors represent the mean time of execution. This was deemed to be an acceptable means of testing the result as the encryption exponent can be chosen as an arbitrary positive integer which is co-prime with Euler's totient. Given this, the encryption exponent can be a range of number with a normal at half the length of the modulus. This provides the average time to compute the exponent. It may be likely that smaller exponent sizes are chosen such that the execution time is reduced, but this will be left up to the system designer as an option.

Since there is a time constraint that is rigorous and an additional requirement to ensure that the cryptographic unit be as small as possible, it is desirous to reduce the size of the cryptographic footprint even further than what was described through the use of a modified Montgomery multiplication. Through the analysis of the system, it was determined that additional resources on the RFID chip could be reduced by relocating the Montgomery multiplication transformation unit. It was found through analysis that this procedure does not affect the security of the proposed system. In addition it could be used to reduce the overall complexity of the requesting system.

A comparison of the proposed algorithm with a serial Montgomery multiplication algorithm without modification is performed. The algorithm that is proposed may not be the most efficient in literature, but it is one that to the author's knowledge is the most suitable

comparison. The comparison will take into account the area of implementation, and the time for computation. An examination will also reveal the number of clock cycles that are required for a given exponentiation with varied security parameter sizes (16, 32, 64, 128, 256, 512, and 1024 bits).

4.5 Modified Montgomery Multiplication System for RFID Implementation

To be in line with the recommendations proposed above and in previous sections, a new RFID RSA security algorithm is proposed to reduce the overall burden of computation on the RFID tag and distribute the cryptographic computation. The proposal is to transmit a partial result of the Montgomery exponentiation result over the air instead of a transformed cryptographic response. A further description of the protocol for performing this and how it fits into the already established public key cryptosystem protocols is examined below.

The proposed algorithm will make use of the core Montgomery multiplication components and will redistribute the transformation from Montgomery residue format back to normal modulo n format at the reader. It was suggested that to perform the Montgomery multiplication there was pre-transformation that was required, namely the multiplication of the message by r and the subsequent reduction modulo n of each multiplied factor. Instead of performing this, the proposed algorithm will take the public key integer pair and message natively without transformation and perform the Montgomery multiplication technique in absentia of the transformation. This has the effect of creating a deficit in the computed number. Each time a Montgomery multiplication is performed the value is reduced by a factor of r. When compounded by the exponentiation algorithm, the value computed is equivalent to (4.2).

$$c = m^e r^{-(k*(e-1))} \; mod \; n \tag{4.2}$$

The advantage of what is being suggested is that there is no conversion of the inputs or the outputs on the RFID tag, which frees up either clock cycles, area or a combination of both (depending on specific implementation). Instead this conversion is placed on the reading device where it there is more power and computational resources available. This is of significant benefit for the specific RFID implementation and other area, power, and time constrained implementations.

This puts additional computational power at the reader end. The pre-computation that is required at the reader needs to be computed only once for each public key pair. The computation that is required at the reader is shown below.

$$x = r^{(k*(e-1))} \; mod \; n \tag{4.3}$$

This computation is required to transform the computed result from the tag using the public key information. Computing the value of *x* can be performed once and then stored in memory for iterative use. Once the tag's response is received, the reader only needs to perform a single multiplication of the result from the RFID tag with the value of *x* (modulo *n*) to arrive at the ciphertext in the normal representation, for further computation.

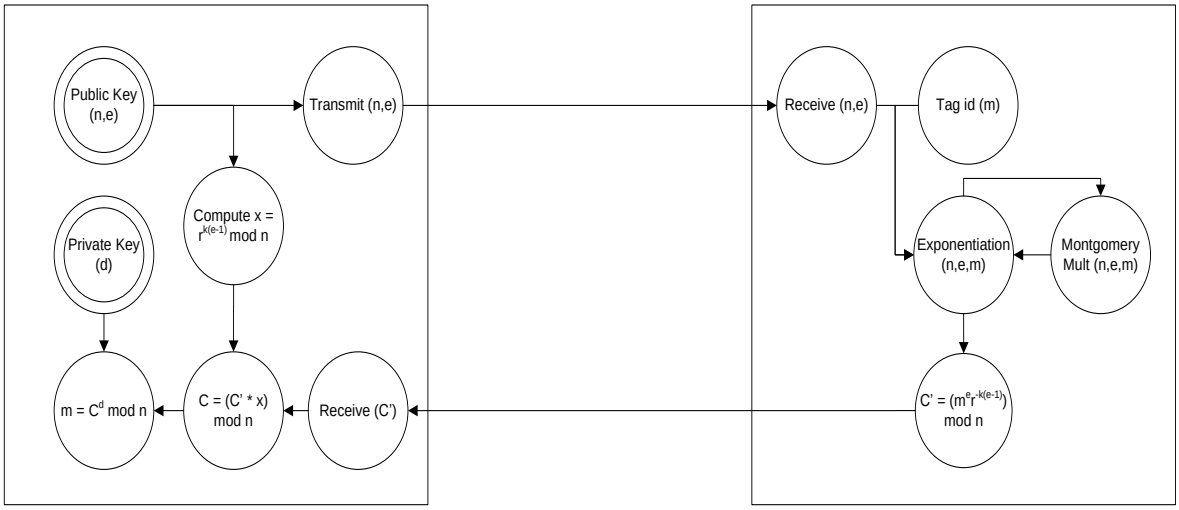

Figure 4.1 Data Flow from Reader to Tag and the Reverse Transmission

Examining the problem from a security aspect, it is important to note that the fidelity of the RSA algorithm is not lost. Transmitting the partial result does not affect the overall security of the system as can be shown analytically in the following example:

$$c' = m^e r^{-(k*(e-1))} \, mod \, n \tag{4.4}$$

$$c = (m^e mod \, n) \cdot (r^{-k*(e-1)} \, mod \, n) \cdot (r^{k*(e-1)} \, mod \, n) \tag{4.5}$$

Since the result that is transmitted is a multiplicative factor of the value of r, and r is composed of public key information, whatever is obtainable through as a result of the RSA algorithm is what is available to the adversary through the proposed algorithm. No additional information is revealed than what is revealed by the public key parameters themselves.

4.6 Considerations for Modified Montgomery Multiplication System for RFID Implementation

Instead it is being suggested that the result be kept in Montgomery space and converted once received by the reader. This proposal addresses three main concerns. First, the overall size

of implementation is drastically reduced. Second, the number of clock cycles that are required is reduced, decreasing the power requirements of the tag. Third, security is maintained through the transaction which is one of the main goals of the work.

Reducing the overall system size is an important means of ensuring the smallest form factor for the implementation. The suggested implementation is far smaller than other comparable algorithms due to the lack of conversion step modules. With that in mind, this method is in fact performing this calculation at another location where it can be performed at a higher rate than on the RFID hardware. The distributed nature of the solution lends itself to optimizations that would not otherwise be obtainable. The hardware usually dedicated to this is relocated to the reader which can better handle additional hardware and timing delays. This enables the tag to free up some hardware space for other applications in the future and distribute some of the computational requirements to other parts of the system. The hardware savings consist of a divider circuit to find the residue, the additional memory required for the storage of temporary values, and a regular multiplier circuit required for the computation of the final result. The final result at the reader will require the multiplication of the result of the Montgomery multiplication and the residue of the $r \; mod \; n$ operation. The result of the multiplication will need to be reduced modulo n once more to arrive at the final encrypted result. These additional components are not required if the result is transmitted as a Montgomery multiplication result.

With regard to timing, no additional clock cycles dedicated to the conversion of the result from Montgomery space to normal space on the tag is required. This has a significant benefit since the exponentiation computation requires a significant number of clock cycles to complete as shown in prior works. The additional benefit of the reader performing the

computation post tag interaction is that once the result is recorded by the reader it can then compute the result at a later time (assuming the message contains solely the tag id). The benefit is apparent when there is a requirement to read numerous tags at an increased speed, and the time allotment is not sufficient for the tag to produce the final translated cryptographic result. By removing the requirement for additional computation the tag is now able to respond in a reduced period of time. It is also reasonable to assume that the reader will have additional computational power to finish the computation in a period of time that far out paces a tag's capabilities.

4.7 Proposed Turing Machine

The proposed state machine for the RSA cryptosystem is described below in Figure 4.2. There are two state machines that are present in the algorithm that control the data transformation process on the RFID tag. The one depicted on the left side of figure represents the exponent finite state machine while the diagram on the right depicts the modular multiplier finite state machine.

The exponent Turing machine consists of 12 states. The first state is the activation state which initializes the variables. This state is triggered when all the data has been loaded into the memory banks for the variables of interest (the message, exponent and modulus). Once this information is available the temporary values are cleared and the machine progresses to scan through the exponent deciding to multiply or to square the temporary result.

The modular multiplier Turing machine is much smaller and performs the modified Montgomery multiplication algorithm. The machine consists of only 6 states and is activated once the start command is received at its port. This is usually actuated by the

exponent control machine when the variable “upsun” is set to a high state. The Montgomery multiplication is carried out by performing a series of additions, subtractions, and binary block shifts. The method progress through the state machine iteratively until the variable “size1” has reached zero. “size1” keeps track of which bit is being examined and multiplied at any given time. Its size is one less than the number of bits in the operand. The machine is executed 2 or 3 times for each bit of the exponent depending on the value of the exponent bit being examined.

Exponentiation State Machine

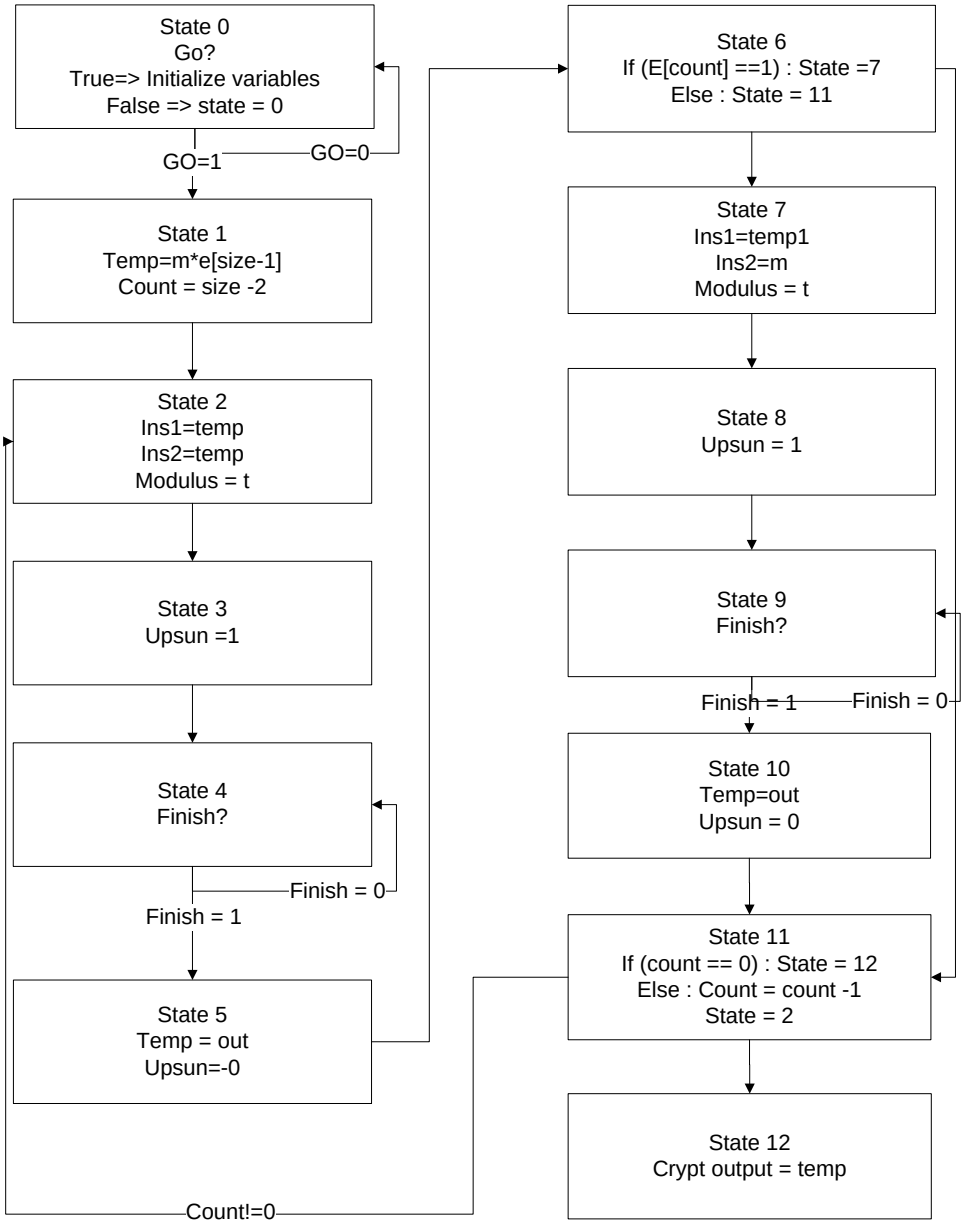

Figure 4.2 Turing Machine for Exponentiation

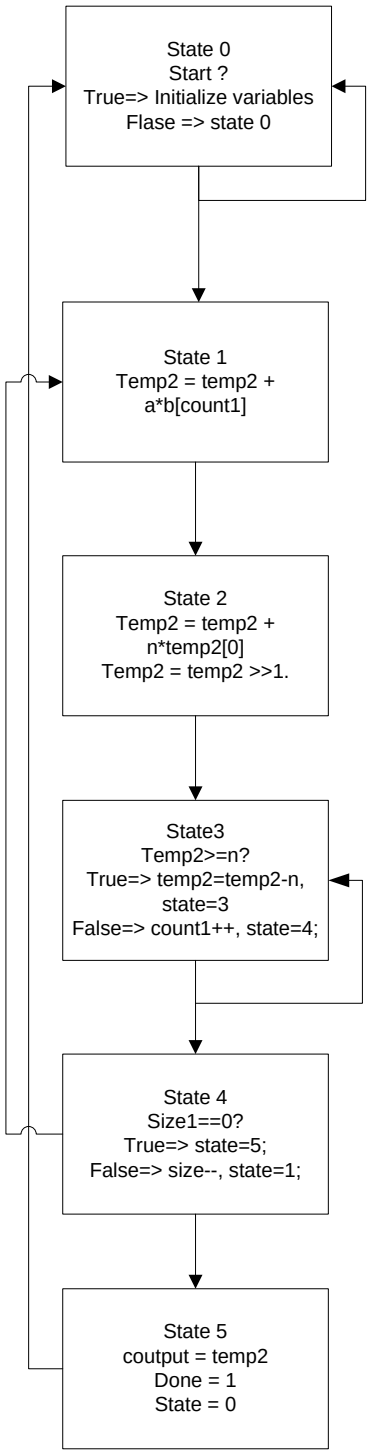

Figure 4.3 Turing Machine for Multiplication

4.8 Memory resources

The figure below describes the memory allocations that would be required in the proposed algorithm. The RSA cryptosystem would have several pins that would be dedicated to the input of the public key information which would be serially transmitted over the air and then buffered into memory. This buffered memory allocations would then be used to feed the controller with the public key information for processing. The message (assumed to the EPC code) would be available in memory and would be accessed in the appropriate memory address allocation. The total memory required for the input of the controller cryptosystem would be $6(n-1)+n+\log_2(n)$, where n is the security parameter. This is due to the registers that are required to keep the values of the exponent, message, and modulus, all $n-1$ in size. The other memory requirement are for the registers used to transfer values into the multiplier architecture. The value of n is from the temporary value and the value $\log_2(n)$ is from a counter used to keep track of the exponent.

Temporary memory is used to ensure that intermediary results determined in the state machine are stored for additional operations. Since the exponentiation is split into modular multiplications one variable is used to store the temporary value of the multiplication and another temporary value is used to store the output of the intermediary exponent results. The size that is used for these temporary memory allocations is sized to be 1 bit greater than the required bit length due to the addition that is performed at the modular multiplication stage. The maximum value that the addition can take is two times the original value which is represented by a single bit.

Ancillary memory resources are needed to serve as control flags and state space identification. The memory resources required for the exponentiation finite state machine is

small, on the order of four bits, since there are only 12 states needed to perform all the operations. In the performance of the multiplication there are six states that are possible which can be adequately represented in 3 bits. The start and stop conditions for the exponentiation control require additional 1 bit inputs/outputs (one for each start and stop command).

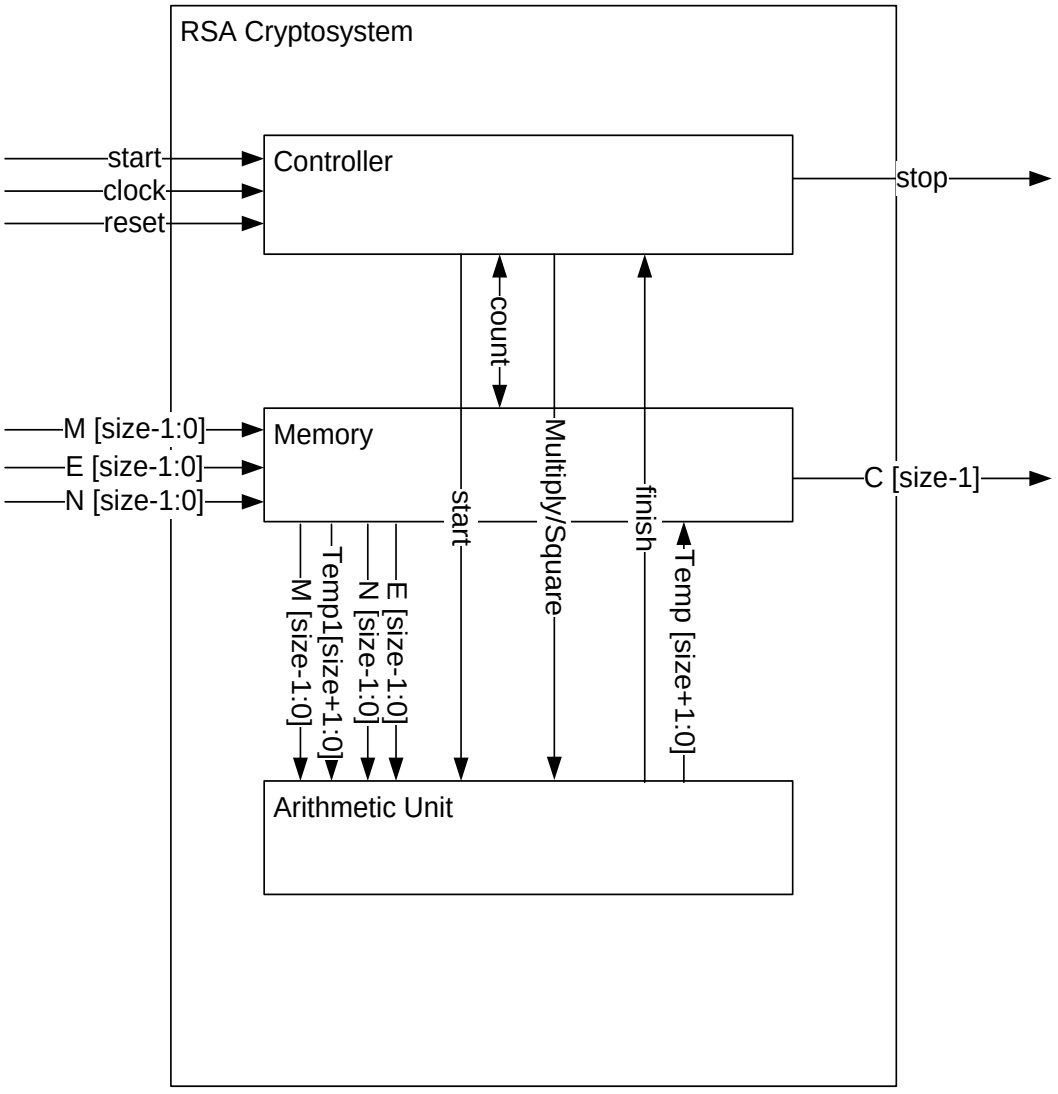

Figure 4.4 Input, Output and Memory Utilization across Logical Systems

4.9 Modules of the modular multiplication

There are three distinct modules that comprise of the proposed system. The first module is considered the exponentiation/control unit. The second module is the addition module which takes on the role of performing the modular multiplication that required by the processor. The third module is the modular reduction piece which ensures that the result of the modular multiplication algorithm is within the confines of the modular arithmetic.

The exponentiation/control unit controls the number of modular multiplications that occur and the number of squaring operations that occur depending on the bit in the exponent. This is the high level coordinator used for the circuit. It controls the number of modular multiplications and modular squaring operations that occur. It also maintains the memory modules for the intermediary exponentiation result as the algorithm progresses.

The addition module performs makes up the major operation for the modular multiplication. Every multiplication of k bits requires k steps to compute the result. The addition module requires a temporary register to store the multiplicand and the multiplier. These are k bits long since they have been reduced modulo n. The result that is produced by this operation is $k+2$ bits long since there is a possibility that the addition also requires the modulus to be added for Montgomery division.

The modular division is performed as part of the modular multiplication algorithm. It takes the intermediary result of the addition and division and ensures that the result is less than the modulus. Since there are two additions that are performed there is a possibility that the result may be twice as great as the modulus, the division (subtraction) is performed iteratively to ensure that the final result is a result modulo n. If only one modular reduction

is performed there is a chance that the result would still be larger than the modulus which could cause overflow issues in subsequent computations. The modular division that is performed ensures that the result is tractable.

4.10 Design Tools

Selecting appropriate design tools for the comparison and development of RTL logic algorithms is of prime importance. These tools enable to quantifiable measurement of key parameters when designing the algorithm. At each phase of the design there are several tools that are required to implement the algorithms. The phases of design include: concept, program verification tool, program Verilog implementation, waveform simulation, verification of functionality, and fitting of design onto a specific device.

During the concept phase, an examination of the mathematical algorithm is performed through traditional paper and pencil methods. This consists of defining the Turing machine that will compute the result of a given algorithm. From this phase a state machine is generated with an approximation for the number of steps that would be required for a given exponentiation. This phase is performed iteratively until a suitable solution is derived which meets the constraints.

The program verification phase uses a C program compiler to mimic the steps that are used in the Turing machine. In C it is very easy to create a tool that is less rigorous than programming in Verilog or other HDL languages. The main output of this step is to provide a means to verify the results that are provided through HDL construction. This is invaluable

when comparing the results that are produced through Verilog when performing code troubleshooting.

A Verilog program is used to provide a low level construct for the implementation of the Turing machine. Elaborating the design in Verilog allows subsequent analysis of resource allocation and overall quantifiable measurement of the algorithm. Efficiently programming these parameters is crucial to a proper comparison of the algorithms.

Once a Verilog file has been produced, stimulating the inputs should result in known values for the outputs. This verification is performed through the use of the waveform generator that is available through ModelSim Altera 6.5e. A cross reference with the desired results from the C generated program is produced to ensure that both the intermediary and final results conform to the final desired result. In addition to providing a means of verification it also furnished the results regarding the number of clock cycles that would be required for execution.

The final design tool that is used is the Altera Quartus II 10.0 Web Edition. This tool was used to compile the code into a simple Cyclone III (EP3C25F324C8) unit to identify the number of CLB (combinational logic blocks) that would be utilized by the design. This is critical for determining the size of the algorithm and the overall applicability to be considered for implementation in an RFID cryptosystem for the next generation tags.

CHAPTER 5 - RESULTS

5.1 Introduction

The following chapter presents the results that were found through analysis of the proposed algorithm. The first section provides a review of the proposed algorithm and how is to operate under normal conditions. The next section discusses the constraints that are present on the development of the algorithm. The final sections provide an analysis of the algorithm when compared to a fully implemented cryptographic encryption on the RFID tag.

5.2 Proposed Algorithm

The proposed algorithm is meant to reduce the overall computational requirements of the RFID tag and to reduce the amount of area that is required on the tag itself. This is critical due to the area constraints on the implementation of the algorithm. The proposed method of ensuring a small footprint while reducing the overall time is to perform a Montgomery multiplication without performing any prior transformation of the input nor transformation at the output. This reduces the number of computational units required and the time required to perform the calculation.

Instead of performing the transformation on the tag as many have suggested, the transformation to the final result can be performed on the reader once the information has been transmitted. This will ensure that minimal resources are being used on the RFID and that the reader, which has more computational and energy resources available to it, can perform any necessary conversions.

5.3 Constraints

As mentioned in previous sections the algorithm is constrained by two major parameters: area (power), and time.

The area problem can also be described as an issue of power available to the circuit. The area can increase so long as the overall power characteristics do not. Since 0.35 um transistor technology has been chosen as the standard for examination this will be the benchmark and all power constraints can be transformed into equivalent area constraints. This is shown more in the analysis. In terms of the time constraint, when dealing with circuits it is best to consider the number of cycles. This measure is appropriate since it is not clearly defined what future RFID tags may have as their clock speeds. Both cycles and time (assuming a 6.78MHz clock) are presented for clarity.

The area has been constrained through the identification of the number of active transistors during a single clock cycle. This restricts the number of transistors that can be active at any given time and generally constrains the number of total transistors on the chip. Due to the serial nature of the algorithm that is being proposed, it is safe to assume that not all transistors will be powered at a given time and that the power will cycle through the circuit as different portions are activated through the state machine. The means of estimating this was performed by indicating that only half of the transistors in a given logic element are active for given clock cycle.

When calculating the time required to compute the final result, a standard clock is used with the clock cycles found in the result to derive an approximate execution time. The timings have been derived from the ModelSim waveform generator. In addition to the clock cycles

that are provided a translation into a nominal clock speed for an RFID is presented. This has been done to provide a close approximation of the time required for an ISO 15693 RFID compliant tag to perform the computation. The maximum clock frequency of the devices is also provided to allow cross platform application of the technology.

5.4 Area

Figure 5.1 shows the number of logic elements, combinational units, and registers utilized for the proposed algorithm. The horizontal line indicates the constrained limit in terms of area. This limit was derived from the number of transistors identified as being the limit for a given distance. The line crosses at just under 129 bits. The limit was derived from Altera estimates of the number of transistors in a given logic element. Their report found that 12 transistors are utilized in any given logic element on average over 100 designed FPGA implementations. The approximate depends on the number of combinational and look up tables utilized per logic element. For this implementation, an approximation of 12 was taken as a good measure of the number of transistors that are implemented in a logic element. When calculating the power consumed by a logic element it is assumed that half of the logic implemented would not consume power at a given point in time. This would vary as transistors are activated to perform arithmetic and writing sequences, but for analysis purposes 50% was deemed to be a high number of active transistors.

In terms of progression, the use of logic elements increases as a positive quadratic function of the number of bits used in the design. The combinational functions exhibit the same quadratic increase as the logic element relation. This is due to the additional overhead required for the management of data. This overhead takes the form of additional logic gates for the representation of stages in the carry propagate chains performing the addition. The

number of registers (memory units) increases at a linear rate as the security parameter increases. This is due to the linear increase in memory requirements for the variables defined in the Verilog script. The major contributing factor to the quadratic nature of the logic element count is the initial control logic which is required for the state machine. The logic elements that are dedicated to control are large in smaller bit size implementations and do not contribute much to the size of the overall construct beyond 256 bits.

Table 5.1 Results of Proposed Algorithm - Number of LE, Combinational Functions, and Registers

N bits	Total Logic Elements	Total combinational functions	Total Registers
16	324	194	238
32	598	340	447
64	1141	627	864
128	2231	1205	1697
256	4721	2928	3362
512	9273	6712	6691
1024	20524	15403	13348

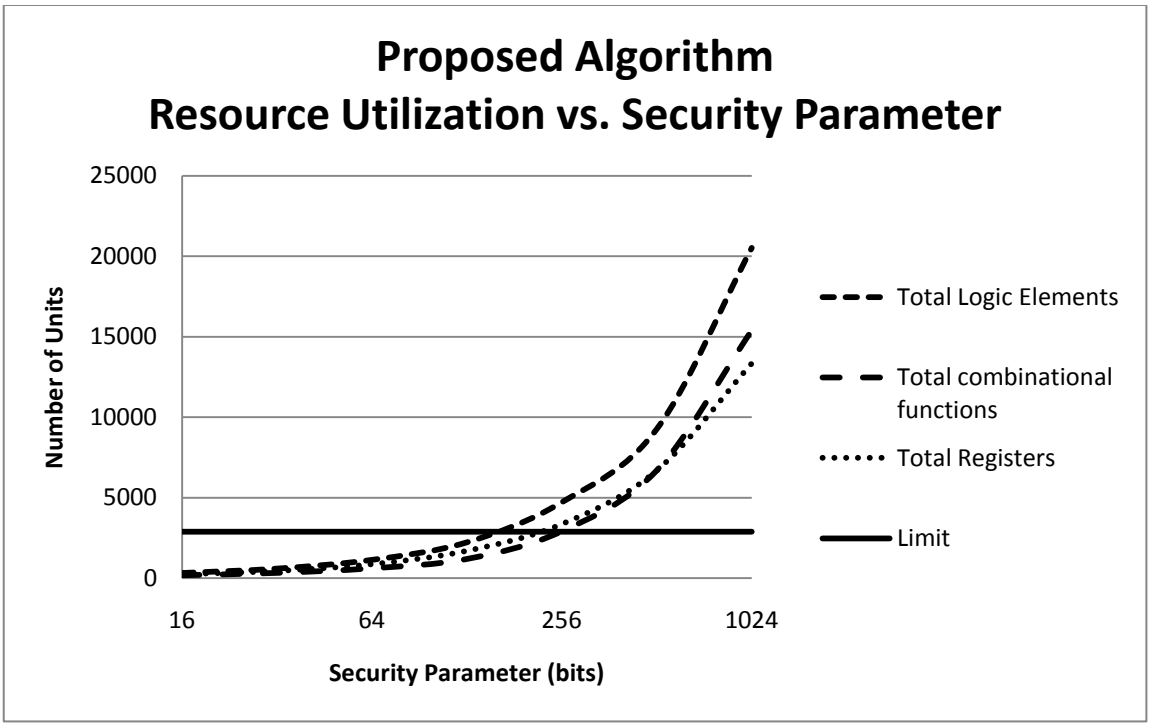

Figure 5.1 Proposed Algorithm: Resource Utilization vs. Security Parameter

The next figure demonstrates the difference between the bit serial implementation that is proposed and a bit serial implementation that performs the translation from Montgomery representation. There is a significant difference in the amount of area that is consumed by the translation hardware. In this case division and multiplication units are required for a full implementation. In larger implementations the multiplication and dividing units require ever increasing area to perform the translation, growing faster than the proposed algorithm.

For the full translation, a full multiplier and dividing unit was implemented through the use of Quartus II packaged modules. These modules can perform a multiplication in a single clock cycle and were deemed to be one end of the spectrum of viable methods of implementing the translation circuit. It was found that although the number of clock cycles did not increase dramatically (as is shown in the next section), the area for implementation increased exponentially. This type of characteristic would have been present if a more area

sensitive process was chosen, and effectively the time to process would have seen the increase.

Table 5.2 Comparison of Logic Elements for Proposed Algorithm and Full Algorithm

Bits	Proposed Algorithm	Full Algorithm
16	324	2266
32	598	7986
64	1141	30121
128	2231	117177
256	4721	462423

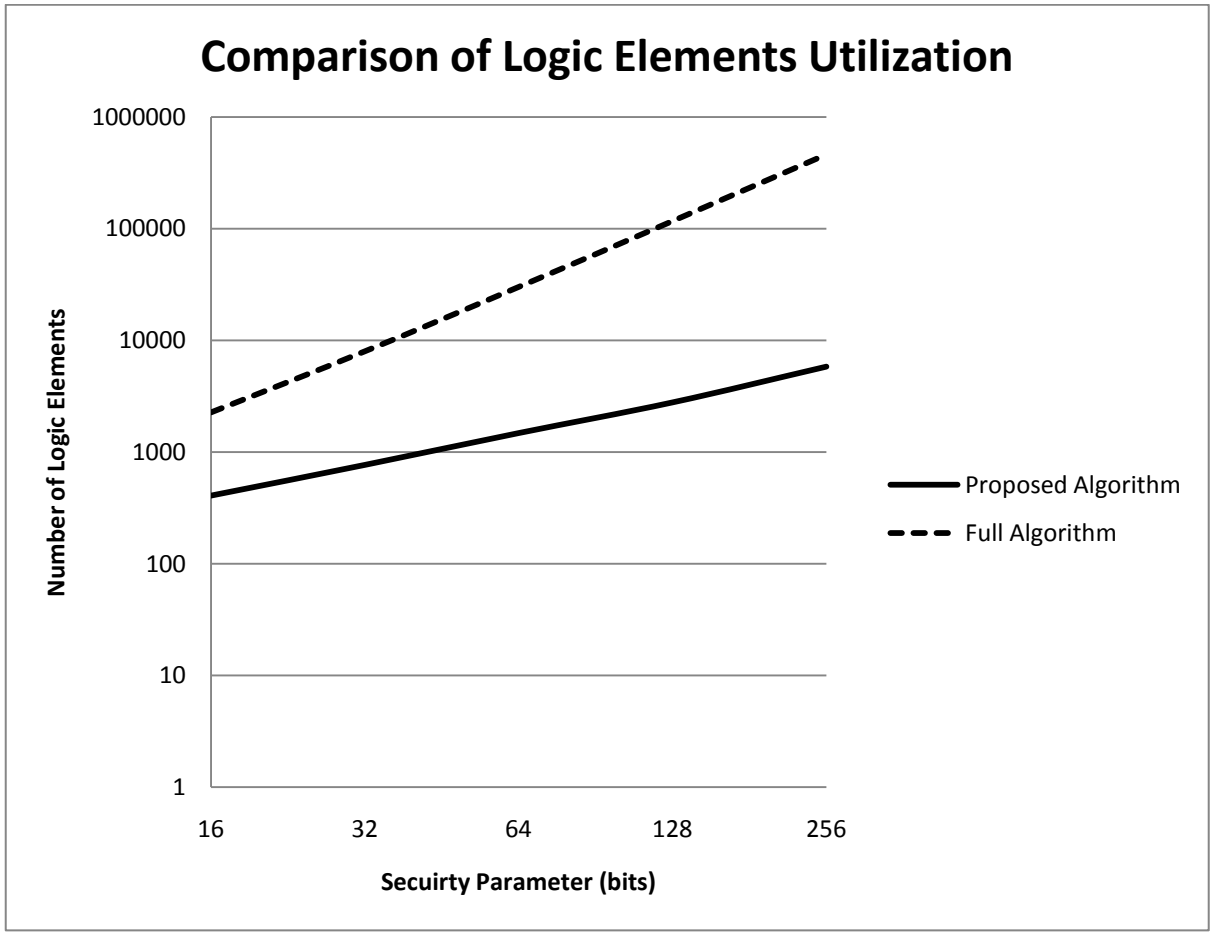

Figure 5.2 Comparison of Logic Element Utilization vs. Security Parameter for Both Algorithms

5.5 Time

The most accurate method of determining the amount of time each algorithm requires is closely related to the number of cycles required to compute the result. It is clear to see that as the number of bits is increased the number of cycles required for the computation approximates a quadratic function. This is attributable to the near doubling in the number of computations required for each bit that is added to the security parameter.

Table 5.3 Comparison of Algorithms: Cycles vs. Security Parameter

N bits	Cycles (Proposed Method)	Cycles (Full Method)
16	1547	845
32	2951	3013
64	13531	13977
128	50352	50562
256	194410	194814
512	821450	822240
1024	3235770	3237330

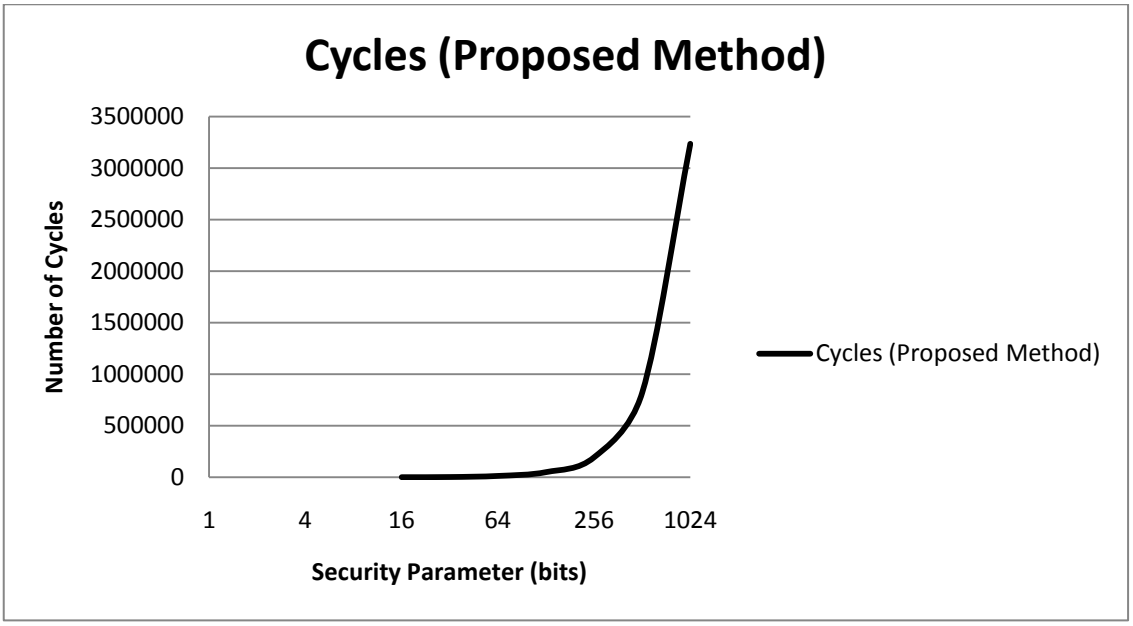

Figure 5.3 Proposed Method Cycles vs. Security Parameter

As indicated in previous discussion, the two algorithms performed nearly at the same level in terms of the number of cycles. A better indication of the difference is provided in the next figure which subtracts the proposed algorithm's cycles from the number of cycles required for the full algorithm.

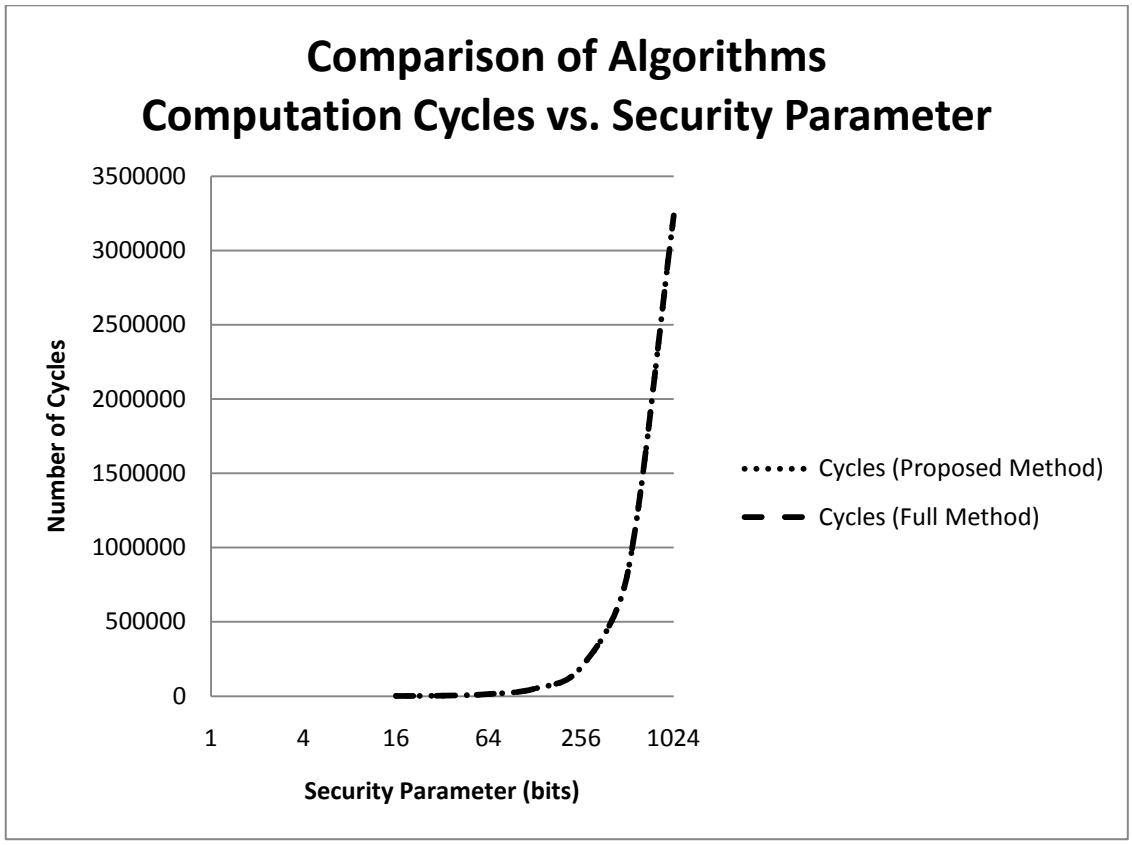

Figure 5.4 Comparison of Algorithms: Computation Cycles vs. Security Parameter

It is interesting to note that for a small number of bits the proposed algorithm fairs worse than the full algorithm with respect to the number of cycles required to complete. The full algorithm makes use of dedicated multiplier logic that is available on the FPGA. This is not

the case when dealing with the proposed algorithm, as it assumes no dedicated logic constructs. What is interesting is that although the number of clock cycles initially required for low modulus size on the full algorithm is faster than the proposed method, after increasing the size of the modulus, an advantage in the proposed algorithms in terms of the number of clock cycles is apparent.

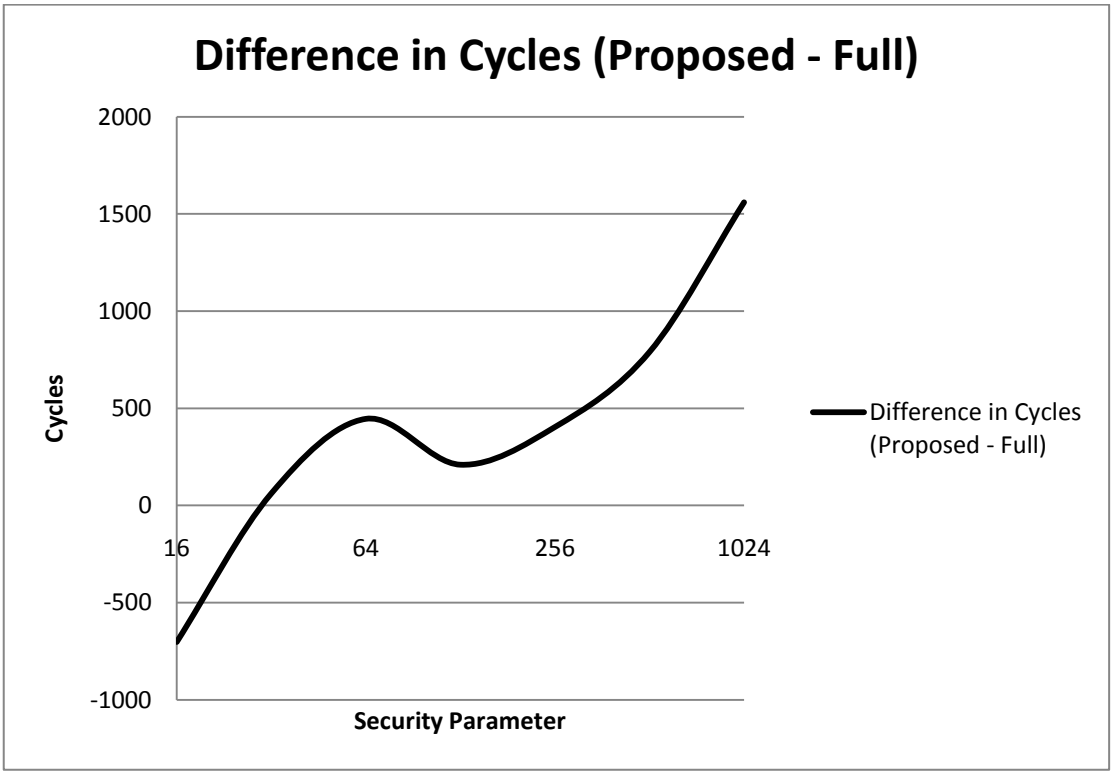

Figure 5.5 Difference in Cycles between Algorithms

For RFID implementations using the ISO 15693 standard, the clock that is used for logic in the circuit is derived from the carrier frequency. The carrier frequency is 13.54 MHz and the derived clock is 6.78 MHz, or simply half of the original. With this information it is possible to interpolate the data and translate it into the time that is required for a specific implementation. The results of this are shown in the figures below.

Table 5.4 Comparison of Algorithms: Time for Computation

N bits	Proposed Method (ms)	Full Method (ms)
16	0.228	0.124631268
32	0.435	0.44439528
64	1.996	2.062
128	7.427	7.458
256	28.674	28.734
512	121.158	121.274
1024	477.252	477.482

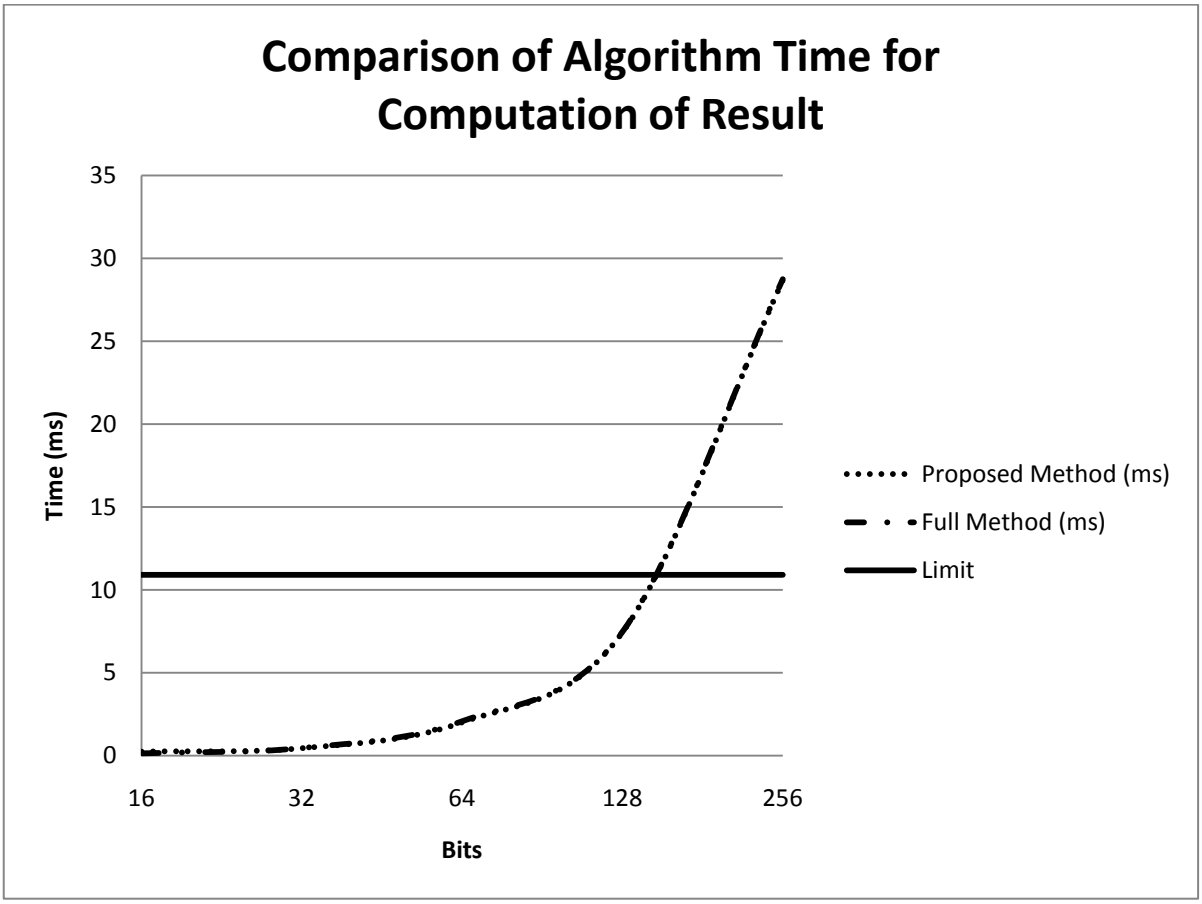

Figure 5.6 Comparison of Algorithms: Time for Computation of Result

From the above analysis it can be shown that for an implementation of152 bits requires 10.9ms which was the identified limit between write and read operations.ms. This is a sufficient amount of security for a 76 bit message if OAEP padding is implemented to ensure semantic security, assuming computational infeasibility for determining the prime factors.

The measured times are given for a common vector set for the message, modulus and exponent. This time will change as the exponent value changes. The smaller the exponent that is used the faster the time of execution for the algorithm. In this case exponent sizes between the smallest and largest were measured as an appropriate mean at which to create a benchmark. For example, if the modulus size was 128 bits long, an exponent was chosen such that it would be 64 bits long constrained by the predicates of choosing the exponent. The specific vectors chosen are available in the appendix.

5.6 Power

The amount of power that each design consumes is derived from the number of logic elements, combinational functions, and registers each algorithm uses. This information was made available from the Altera Quartus Power Play application. After compiling the Verilog algorithms it was able to determine the power consumed on the FPGA platform. As this was the only platform that was examined, it was deemed that this would provide a good insight into possible power consumption on an ASIC, and at least provide an above average estimate of the power required for a given implementation. The estimates for power consumption are higher than anticipated on an ASIC implementation due to the additional overhead power that is required on an FPGA platform.

The following shows power estimates for the two algorithms. There is a significant difference between the two implementations which can be associated with the multiplication and division units that are used in the full algorithm which are larger dedicated logic units than what is in the proposed algorithm.

Table 5.5 Comparison of Algorithms: Power Consumption

N bits	Power Proposed Algorithm (mW)	Power Full Algorithm (mW)
16	0.24	0.886
32	0.455	2.887
64	0.88	10.378
128	1.692	38.533
256	3.456	152.067

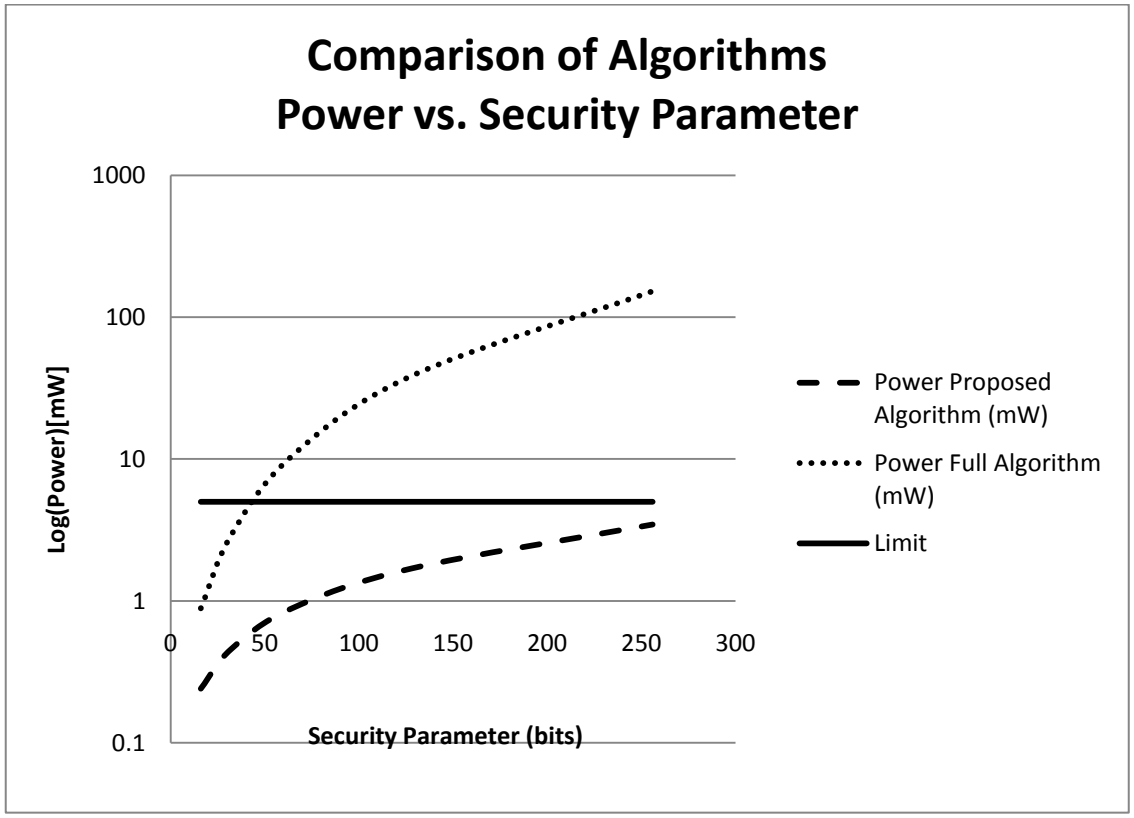

Figure 5.7 Comparison of Algorithms: Power Consumption vs. Security Parameter

From the information above, it is clear that the proposed algorithm is much better suited for an RFID implementation. Using the full algorithm, it would not be possible to implement a high order RSA algorithm without hitting the power limitation to the tag. In this case, it is clear that even at 256 bits, the proposed algorithm does not hit the 5 mW ceiling mandated by the transmitted power at 50 cm as explored in chapter 3.

CHAPTER 6 – CONCLUSIONS

6.1 Summary

A variety of examinations have been done to determine an efficient means of performing a fast modular exponentiation algorithm. The concentration of work has focused on the speed of the algorithm and whether it there is a possibility of increasing the throughput of the algorithms.

An examination of the current state of the art with respect to RFID security was performed. Current technology and technical implementations for securing RFID tags was covered. The limitations of the current security protocols were examined and the future requirements of RFID tags explored.

An examination into security and its requirements was undertaken. A discussion about what makes an algorithm secure was performed. Current symmetric and asymmetric algorithms were discussed and compared. The RSA algorithm was explored and shown to be secure under OAEP as long as the modulus size (security parameter) is twice as large as the message that is to be encrypted..

Timing analysis was dictated by the standards that are currently in place. It was found that the maximum time between a write cycle and a read cycle would be 10.9 ms. This was then taken as the time in which the algorithm must be able to compute the result of the encryption.

This study examined the methods being implemented and determined whether it would be feasible to implement an RSA cryptosystem onto an RFID tag. Through examination it was determined that a novel method of performing encryption could be achieved by leveraging the capabilities of the RFID tag and reader. Using this new method of performing the RSA encryption, the results indicate that on 0.35μm CMOS technology it would be feasible to perform RSA cryptography on an RFID tag when used in conjunction with the reader.

From the results it can be shown that the limiting factor to the implementation RSA cryptosystem on an RFID tag is the power required to drive logic circuit. This can be seen by the limitation on the area of the RFID chip presented in Figure 5.1, which indicates a 129 bit modulus as the limit for implementation. This limitation is closely followed by the time constraint which shows 152 bits modulus as a maximum. Although, the overall power delivered to the tag indicates that a modulus of 256 bit can be computed, the division of power at a given clock frequency (6.78 MHz nominally) and the number of transistors required to be powered at a given time almost halves the number of bits which can be used in the RSA cryptosystem from more than 256 to 129 bits. By distributing the encryption, it is possible to fit a RSA cryptosystem onto a passive RFID chip with a marginal security parameter of 128 bits, sufficient for a 64 bit message on 0.35um transistor technology.

6.2 Contributions

The contributions of this work are as follows:

- Survey of existing Montgomery multiplication methods
- Assess the feasibility of an RSA cryptosystem on an RFID chip

- Examine the level of security required to ensure that the algorithm is semantically secure
- Suggest a new method of public key encryption involving both reader and tag to reduce area, time and power requirements on the tag
- Demonstrate that the new algorithm is secure under the definitions of semantic security.
- Investigate the performance improvement of the suggested algorithm with a traditional algorithm and demonstrate significant savings for low powered devices.

6.3 Future Work

The work was commissioned as a feasibility study in the realm of performing RSA cryptographic computations on an RFID. The result of which was a clear indication that under certain constraints this is indeed feasible.

Future work in along the lines of this work would be the implementation of this architecture on an ASIC design. Layout and power management have very interesting problems associated with them in which this project would have to surmount. This would be the natural progression to the work that was done in this thesis.

Analysis could be performed to evaluate the use of 0.18 μm CMOS technology using the same parameters. As this technology gains traction amount manufacturers, significant benefits can be realized for area restricted applications. The power required for a 0.18 μm CMOS implementation would likely be half of that required for a 0.35 μm implementation. The implementation area would reduce by a quarter due to the dimensions of the new transistors. This would mean that additional power and space is available for a given RSA

cryptosystem based implementation significantly increasing the security parameter that could be successfully implemented.

Additional work would be to find even more efficient algorithms to perform RSA encryption on an inductive power harvesting platform. The work that was performed here would be a great spring board into this topic and further investigations into performance enhancement methods for a low power application.

References

[1] L. Tobias, M. Schneider, and C. Ruland, "Analysis of Power Constraints for Cryptographic Algorithms in Mid-Cost RFID Tags," *Lecture Notes in Computer Science*, vol. 3928, pp. 278-288, 2006.

[2] S. Moon, "Design of a Scalable RSA Cryptoprocessor Embedded with an Efficient MAC Unit," in *International Conference on Future Generation Communication and Networking*, 2008, pp. 74-77.

[3] S. Wu, Y. Zhu, and Q. Pu, "Resource Efficient Hardware Design for RSA," in *Proceedings of the First Interational Multi-Symposiums on Computer and Computational Sciences*, 2006.

[4] S. Goldwasser and S. Macali, "Probabilistic Encryption ," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, Apr. 1984.

[5] M. Bolic, "Radio Frequency Identification Technology," University of Ottawa Course Notes, Sep. 2009.

[6] N. C. Wu, M. A. Nystrom, T. R. Lin, and H. C. Yu, "Challenges to global RFID adoption," *Technovation*, vol. 26, no. 12, pp. 1317-1323, Dec. 2006.

[7] D. Dobkin and T. Wandinger, "A Radio-Oriented Introduction to Radio Frequency Identification," *High Frequency Electronics*, pp. 46-51, 2005.

[8] J. Landt, "The History of RFID," *IEEE Potentials*, vol. 24, no. 4, pp. 8-11, Dec. 2005.

[9] International Standards Organization, "Radiofrequency identification of animals -- Advanced transponders -- Part 1: Air interface," International Standards Organization Standard 14223-1:2003, 2003.

[10] International Standards Organization, "Radiofrequency identification of animals -- Advanced transponders -- Part 2: Code and command structure," International Standards Organization Standard 14223-2:2010, 2010.

[11] International Standards Organization, "Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 1: Physical characteristics," International Standards Organization Standard 14443-1:2000, 2000.

[12] International Standards Organization, "Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 2: Radio frequency power and signal interface," International

Standards Organization Standard 14443-2:2001, 2001.

[13] International Standards Organization, "Identification cards -- Contactless integrated circuit(s) cards -- Proximity cards -- Part 3: Initialization and anticollision," International Standards Organization Standard 1444-3:2001, 2001.

[14] International Standards Organization, "Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 4: Transmission protocol," International Standards Organization Standard 14443-4:2001, 2001.

[15] International Standards Organization, "Identification cards -- Contactless integrated circuit cards -- Vicinity cards -- Part 2: Air interface and initalization," International Standards Organization Standard 15693-2:2000, 2000.

[16] International Standards Organization, "Identification cards -- Contactless integrated circuit cards -- Vicinity cards -- Part 1: Physical characteristics," International Standards Organization Standard 15693-1:2000, 2000.

[17] International Standards Organization, "Identification cards - Contactless integrated circuit(s) cards - Vicinity cards -- Part 3: Anticollision and transmission protocol," Interational Standards Organization Standard 15693-3, 2006.

[18] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 1: Reference architecture and definition of parameters to be standardized," International Standards Organization Standard 18000-1:2004, 2008.

[19] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 2: Parameters for air interface communications below 135 kHz," International Standards Organization Standard 18000-2:2004, 2004.

[20] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 3: Parameters for air interface communications at 13,56 MHz," International Standards Organization Standard 18000-3:2008, 2008.

[21] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 4: Parameters for air interface communications at 2,45 GHz," International Standards Organization Standard 18000-4:2008, 2008.

[22] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 6: Parameters for air interface communications at 860 MHz to 960 MHz," International Standards Organization Standard 18000-6:2004, 2004.

[23] International Standards Organization, "Information technology -- Radio frequency identification for item management -- Part 7: Parameters for active air interface communications at 433 MHz," International Standards Organization Standard 18000-7:2008, 2008.

[24] EPC Global, "13.56MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification Candidate Recommendation," MIT Auto-ID Center, Cambridge, Technical Report, 2003.

[25] EPC Global, "Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag," MIT Auto-ID Center Specification, 2003.

[26] EPC Global, "860MHz–930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical COmmunication Interface Specification Candidate Recommendation," MIT Auto-ID Center, Cambridge, Technical Report, 2002.

[27] NXP Semiconductors, "NXP MIFARE PLUS - benchmark security for mainstream applications," Phillips Brochure 9397 750 16722, Oct. 2009. [Online]. http://mifare.net/technology/security/mifare-plus/

[28] National Institute of Standards and Technology, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher," Special Publication 800-67, 2004.

[29] J. Daemen and V. Rijmen, "The Block Cipher Rijndael," in *Smart Card. Research and Applications*. Springer Berlin Heidelberg, 2000, pp. 277-284.

[30] Q. Xiao, "RFID Security and Privacy," in *class "RFID Technology", ELG7177*, Ottawa, Nov. 2009.

[31] E. Haselsteiner and K. Breitfus, "Security in Near Field Communications: Strengths and Weaknesses," Philips Semiconductors, Gratkorn, Technical Report, 2006.

[32] R. Anderson, "RFID and the Middleman," Cambridge University, 2007.

[33] J. Lee and Y. Yeom, "Efficient RFID authentication protocols based on pseudorandom sequence generators," *Designs, Codes and Cryptography*, vol. 51, no. 2, pp. 195-210, 2008.

[34] P. H. Cole, L. Turner, Z. Hu, and D. C. Ranasinghe, "The Next Generation of RFID Technology," in *Unique Radio Innovation for the 21st Century*, D. C. Ranasinghe, Q. Z. Sheng, and S. Zeadally, Eds. Adelaide, Australia: Springer Berlin Heidelberg, 2010, pp. 3-23.

[35] M. Simatic, "RFID-based Distributed Memory for Mobile Applications," in *Mobile Computing, Applications, and Services*, O. Akan, et al., Eds. Paris, France: Springer Berlin Heidelberg, 2010, pp. 172-189.

[36] E. Kosta, M. Meints, M. Hansen, and M. Gasson, "An analysis of security and privacy issues relating to RFID enabled ePassports," in *New Approaches for Security, Privacy and Trust in Complex Environments*, H. Venter, et al., Eds. Springer Boston, 2007, pp. 467-472.

[37] S. Brands, "A Technical Overview of Digital Credentials," Credentica, Redmond, Technical Report, 2002.

[38] I. Wegener, *Complexity Theory: Exploring the Limits of Efficent Algorithms*. Berlin, Germany: Springer Berline Heidelberg, 2005.

[39] Y. Watanabe, J. Shikata, and H. Imai, "Equivalence between semantic security and indistinguishability agains chosen cipertext attacks," *Lecture Notes in Computer Science*, vol. 2567, pp. 71-84, Feb. 2002.

[40] D. R. Stinson, *Cryptography: Theory and Practice*, 2nd ed., K. H. Rosen, Ed. Boca Raton, FL, USA: Chapman & Hall/CRC, 2002.

[41] S. Cook, "The comoplexity of theorem-proving procedures," *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pp. 151-158, 1971.

[42] O. Goldreich, *Foundations of Cryptography*. Cambridge , UK: Cambridge University Press, 2004.

[43] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-jkey cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[44] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transacations on Information theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976.

[45] R. L. Rivest, A. Shamir, and L. Adleman, "A mehtod for obtaining digital signatures nad public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[46] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions Information Theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976.

[47] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption - How to Encrypt with RSA," in *Lecture Notes in Computer Science: Advances in Cryptology - EUROCRYPT '94*, Berlin, 1995, pp. 92-111.

[48] N. Nedjah and L. de Macedo Mourelle, "Three Hardware Architectures for the Binary Exponentiation: Sequential, Parallel and Systolic," *IEEE Transactions on Circuits and Systems*, vol. 53, no. 3, pp. 627-633, Mar. 2006.

[49] P. Montgomery, "Modular Multiplication Without Trial Divsion," *Mathematics of Computation*,

vol. 44, no. 170, pp. 519-521, Apr. 1985.

[50] T. A. Cetin Kaya Koc, "Analyzing and Comparing Montgomery Multiplication Algorithms," *IEEE Micro*, pp. 23-33, 1996.

[51] M. M. J. V. M. Ciaran McIvor, "FPGA Montgomery Multiplier Architectures - A Comparison," *IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004.

[52] P. B. McLaughlin, "New Frameworks for Montgomery's Modular Multiplication Method," *Mathematics of Computation*, pp. 899-906, 2003.

[53] E. C. M. G. Levant Ertaul, "Implementations of Montgomery Multiplication Algorithms in Machine Languages," California State University, East Bay Conference Paper, 2009.

[54] Z. S. P. S. H. D. M. Luc Rijnders, "Timing Optimization by Bit-Level Arithmetic Transoformations," in *Design Automation Conference*, Brighton, 1995, pp. 48-53.

[55] O. K. A. P. Fournaris, "Montgomery Modular Multiplier Architectures and Hardware Implementations for an RSA Cryptosystem," in *Micro-NanoMechatronics and Human Science, 2003 IEEE International Symposium on*, Cairo, 2003, p. 778.

[56] C. P. Gerardo Orlando, "A Super-Serial Galois Fields Multiplier for FPGAs and it Application to Public-Key Algorithms," in *Field-Programmable Custon Computing Machines, FCCM '99*, Napa Valley, 1999, p. 232.

[57] M. B. I. R. K. A. S. J. Muhammad I. Ibrahimy, "FPGA Implementation of RSA Encryption Engine with Flexible Key Size," *International Journal of Communications*, vol. 1, no. 3, pp. 107-113, 2007.

[58] M. McLoone, C. McIvor, and J. McCanny, "Coarsely integrated operand scanning (CIOS) architecture for high-speed Montgomery modular multiplication," in *2004 IEEE International Conference on Field-Programmabled Technology*, 2004, pp. 185-191.

[59] A. Daly and W. Marnane, "Efficient Architectures for Implementing Montgomery Modular Multiplication and RSA Modular Exponentiation on Reconfigurable Logic," in *ACM FPGA 2002*, Monterey, Feb. 2002, pp. 40-49.

[60] International Organization for Standardization; International Electortechnical Commission, "Identification Cards - Contactless integrated circuit(s) cards - Vicinity cards," ISO/IEC International Standard, 1999.

[61] F. T. Ulaby, *Fundamentals of Applied Electromagnetics*, 5th ed., M. J. Horton, Ed. Upper Saddle River, U.S.A: Pearson, Prentice Hall, 2007.

[62] Y. Lee, "Microchip: Antenna Circuit Design for RFID Applications," Microchip Technology Inc. Application Note AN710, 2003.

[63] International Organisation for Standardization, "Identification cards -- Integrated circuit(s) cards with contacts -- Part 1: Physical characteristics," International Organization for Standardization Standard ISO/IEC 7816-1:1998, 1998.

[64] G. F. Derbenwick, "Embedded Ferroelectric Memory for RFID Tag Applications," *IEEE International Symposium on the Applications of Ferroelectrics*, pp. 1-2, Feb. 2008.

[65] C.-C. O. Yang, B. S. Prabhu, C. Qu, C.-C. Chu, and R. Gadh, "Read / Write Performance for low memory passive HF RFID tag-reader system," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 4, no. 3, pp. 1-16, Dec. 2009.

[66] Auto-ID Center, "13.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification: Candidate Recommendation, Version 1.0.0," Massachusets Institute of Technology, Cambridge, Technical Report, 2003.

[67] P. Martin, "An Analysis fo Random Number Generators for a Hardware Implementation of Genetic Programming Using FPGAs and Handel-C," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 837-844, 2002.

[68] T. Lohmann, M. Schneider, and C. Ruland, "Analysis of Power Constraints for Cryptograhic Algortihms in Mid-Cost RFID Tags," *Lecture Notes in Computer Science*, vol. 3928, pp. 278-288, 2006.

Appendix – Test Vectors

Variable – 16bits	Vector
P (8 bits)	211
Q (8 bits)	199
N (16 bits)	41989
Φ	41580
E	181
D	24121
R	23198
R^X	180
Message (16 bits)	41641
Montgomery Multiplication Result	21340
Ciphertext	36999
Time	30940 ns

Variable – 32bits	Vector
P (16 bits)	46273
Q (16 bits)	52837
N (32 bits)	2444926501
Φ	2444827392
E	47111
D	1385441975
R	1438688511
R^X	47110
Message (31 bits)	1116044077
MM Result	2004831824
Ciphertext	840815138
Time	59020 ns

Variable – 64 bits	Vector
P (32 bits)	3246329599
Q (32 bits)	4076493089
N (64 bits)	13233640174939641311
Φ	13233640167616818624
E (32 bits)	3737986037
D (64 bits)	11072773610182757981
R	1562982242371278665
R^X	3737986036

Message (64 bits)	10089650546822379504
MM Result	7108760683879842223
Ciphertext	490404142599986699

Variable – 128 bits	Vector
P (64 bits)	13554591584659775893
Q (64 bits)	15241383738513467519
N (128 bits)	206590731760625001070328161620874719467
Φ (128 bits)	206590731760625001041532186297701476056
E (64 bits)	16893218476843552673
D (128 bits)	186930165870785561221301683181018183801
R	175335756296588514942749441307596910257
R^X	16893218476843552672
Message (124)	167259222007945783907266721387226447 91
MM Result	5444316555795436838471871470621522 5317
Ciphertext	936922216921864649566487109745196 16

Variable – 256 bits	Vector
P (128 bits)	240018241272736891191044375511572914567
Q (128 bits)	291462585031024916232493960151625815387
N (256 bits)	69956337055952130161162709097744032210028433575686565137465973746125465042429
Φ (256 bits)	69956337055952130161162709097744032209496952749382803330042435410462266312476
E (128 bits)	212961103047037086582846502739494333223
D (255 bits)	33317820785936861501516805037510602532328847913304338585894799118052930440011
R	28730985351666980243403965994801347138234409873528072421657711500440335957554
R^X	212961103047037086582846502739494333222
Message (255 bits)	50246235181034361464091463499312280777008577771566203327009115257224541100589
MM Result	97881380434732016604784254381736421779799173173362346946277126126208950978 95
Ciphertext	62372707327102876877555157079764083779661489721854248154288226730885083043634

Variable – 512 bits	Vector
P (256 bits)	7943566540803721228696935284103369636805550378258000150937007328449998717250 1
Q (256 bits)	8972357467898167960757962041376144684813370254216979564618130020969625488080 7
N (512 bits)	7127251857412628530878421894380288343871130830188264527523123191477657065623937046975593011836490331714075881278557479645162025333870143922091712703088307
Φ (512 bits)	7127251857412628530878421894380288343871130830188264527523123191477657065623767887735505992944595782740821086135341290438837275536714592548597516461035000
E (256 bits)	7959580613759923707955486320034770418731434888034931439259274532939356272096 7
D (512 bits)	6723513903524736817761059042394880863458011336459672259100221694024505365712409197737597637455164791509290397337804084253422199948792594594950230392446303
R	3139399258435286215277916201496772740831208305829443699996152634350907979449464778310083010639195697114968045154136848767514215339758610095069224080019794
R^X	7959580613759923707955486320034770418731434888034931439259274532939356272096 6
Message (511 bits)	4025088839751686338815588152813564451115530825469459225316384763088278457788527252278264633513715696630984183838557815503626719725551184866230515611507238
MM Result	4264610443704747291768438084429313062782111486561426772090666453130924009376736335616347753475609978056255435117085456513883790108043128019238882735489565
Ciphertext	5469634031240325872625422011316393161809151548780016074279553946305239412662609746068893716453836856996969933899374862794863035717896039064130967999947737

Variable – 1024 bits	Vector
P (512 bits)	1334137566586190098447099391647268133677202408846401860698730433084889062402789286836277992558068253249872004421206610803044004260851525790932213872922793 7
Q (512 bits)	7372826716721926416162052020054319661900523648968251141803097015195303294204808687519330521506078082241973997785175339533926681893958098315158260216629 61
N (1024 bits)	9836365094709040426598173768006573884901530382693131044932911283454739115059596709317484085235834824171373171807093499051707579770108480671574092869735095652200712693942382043434306764565497450345641254607828391480497832353899276757604195629478894969267750522995398380385731273790865903117840455881635934145 7

Φ (1024 bits)	98363650947090404265981737680065738849015303826931310449329112834547391150595967093174840852358348241713731718070934990517075797701084806715740928697350935807804744355596419801297131118653975830908675113808535124403632279345074534874486074181811218402336782312509993220215328905197861120110663720851608450560
E (512 bits)	12259197491215679662969539558343972444103903672508819147186732497089771986378383185644435950976280618461258205411101783620952894537013024533413176255317781
D (1021 bits)	11664196452297712425592075852734639000666221143342368856133989796162627963970333651091683311113073807120875156596648042288661808544317127375177460136118794291149590605082183282549874788224339272687355833805218783404065811410891951127117015567680389648912314204085508403993983934736490170324985419163345300541
R	48882517217828885094518299476842540758367623874437634126926014288629824206372639461627926782639692638221361928840741232487232720160156939181402021560550498717056805335683851684920709927747612652897723272436538862328243451415370181386365169774570664475053040101195442257629901047348953411868262089104275623331
R^X	12259197491215679662969539558343972444103903672508819147186732497089771986378383185644435950976280618461258205411101783620952894537013024533413176255317780
Message (1022 bits)	36154808791957472345886288496444447124738278151160777595729779993958934180775224666389505849042470155404210163242623276611759488832746801877529347340107649869237096426338419668172267818414913153271783834220283416665186547033624243452295283144385758255172936658139931740380106687298674190086299134317968511294
MM Result	72695877608748555065717268070922056379271109263593270941872501621880213402039215447850789379847776085863020759482851920146621800491176435620539035448647730490581937477422931959714130126201081599130237093583883277905494631685183823669902997129107768702412655896635163336858973010891372469105814002588045808757
Ciphertext	47773262907832226423205291768256538433770910357272869989272121735488528071726908025950351811461187772494744663134973223275986069038915405539679250046449536571074189950177922395419416000617182366097769983347792678912733748791390013858355279511335697481712786390696943538661226242255105024261640341118682626824

Note: The following prime number s were calculated with Arnold Reinhold's Big Number Calculator Applet: http://world.std.com/~reinhold/BigNumCalc.html.