

**NAVAL**

**POSTGRADUATE**

**SCHOOL**

**MONTEREY, CALIFORNIA**

**THESIS**

**DEVELOPMENT OF INFORMATION ASSURANCE PROTOCOL FOR LOW BANDWIDTH NANOSATELLITE COMMUNICATIONS**

by

Cervando A. Banuelos II

**September 2017**

Thesis Advisor: Marcus Stefanou

Co-Advisor: Jim Horning

**Approved for public release;Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **REPORT DOCUMENTATION PAGE** | | | | | *Form Approved OMB  No. 0704-0188* | |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | | | | |
| **1. AGENCY USE ONLY *(Leave blank)*** | | **2. REPORT DATE**  September 2017 | | **3. REPORT TYPE AND DATES COVERED**  Master’s thesis | | |
| **4. TITLE AND SUBTITLE**  DEVELOPMENT OF INFORMATION ASSURANCE PROTOCOL FOR LOW BANDWIDTH NANOSATELLITE COMMUNICATIONS | | | | | **5. FUNDING NUMBERS** | |
| **6. AUTHOR(S)** Cervando A. Banuelos II | | | | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**  Naval Postgraduate School  Monterey, CA 93943-5000 | | | | | **8. PERFORMING ORGANIZATION REPORT NUMBER** | |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**  N/A | | | | | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** | |
| **11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number \_\_\_\_N/A\_\_\_\_. | | | | | | |
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT**  Approved for public release;Distribution is unlimited. | | | | | **12b. DISTRIBUTION CODE** | |
| **13. ABSTRACT (maximum 200 words)**  Nanosatellites provide a light, efficient, and cost effective way for research institutions to carry out experiments in low Earth orbit. These satellites frequently use the ultra-high and very high frequency bands to transfer their data to the ground stations, and oftentimes will use the internet protocol and the Transmission Control Protocol as a standard for communication to ensure the arrival and integrity of the data transmitted. Due to bandwidth limitations and signal noise, these connection-based protocols end up accruing a large data bandwidth cost in headers and retransmission costs. Furthermore, due to connection unreliability, encryption and integrity checks present a challenge.  The aim of this thesis was to develop a software based low-bandwidth reliable network protocol that can support a cryptographic system for encrypted communications using commercial off-the-shelf components. This protocol would reduce the data overhead, retain the retransmission functionality and integrate support for a cryptographic system. Work consisted of developing the encryption mechanism, assessing its resilience to error propagation, and developing the protocol to work over a simulated network. The result of the study is a proof of concept that the protocol designed is feasible, applicable, and could be used as a communication standard in future projects. | | | | | | |
| **14. SUBJECT TERMS**   lowercase all terms, except proper nouns | | | | | | **15. NUMBER OF PAGES** |
| **16. PRICE CODE** |
| **17. SECURITY CLASSIFICATION OF REPORT**  Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**  Unclassified | | **19. SECURITY CLASSIFICATION OF ABSTRACT**  Unclassified | | | **20. LIMITATION OF ABSTRACT**  UU |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release;Distribution is unlimited.**

**DEVELOPMENT OF INFORMATION ASSURANCE PROTOCOL FOR LOW BANDWIDTH NANOSATELLITE COMMUNICATIONS**

Cervando A. Banuelos II

Rank, Branch of Service (spell out completely)

B.S., Texas A&M University, 2013

Submitted in partial fulfillment of the

requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL**

**September 2017**

Approved by: Marcus Stefanou, Ph.D.

Thesis Advisor

Jim Horning

Co-Advisor

Peter Denning, Ph.D.

Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Nanosatellites provide a light, efficient, and cost effective way for research institutions to carry out experiments in low Earth orbit. These satellites frequently use the ultra-high and very high frequency bands to transfer their data to the ground stations, and oftentimes will use the internet protocol and the Transmission Control Protocol as a standard for communication to ensure the arrival and integrity of the data transmitted. Due to bandwidth limitations and signal noise, these connection-based protocols end up accruing a large data bandwidth cost in headers and retransmission costs. Furthermore, due to connection unreliability, encryption and integrity checks present a challenge.

The aim of this thesis was to develop a software based low-bandwidth reliable network protocol that can support a cryptographic system for encrypted communications using commercial off-the-shelf components. This protocol would reduce the data overhead, retain the retransmission functionality and integrate support for a cryptographic system. Work consisted of developing the encryption mechanism, assessing its resilience to error propagation, and developing the protocol to work over a simulated network. The result of the study is a proof of concept that the protocol designed is feasible, applicable, and could be used as a communication standard in future projects.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1

A. rESEARCH DOMAIN 1

B. Research problem and Motivation 2

C. Research Questions 5

D. scope 5

E. Approach 6

F. Thesis Structure 6

II. Background 8

A. Problem Space: Low Bandidth in UHF and VHF bands 8

1. Common Nanosatellite Frequency and Bit Rate Ranges 8

2. Bit Error Rate and Packet Loss 10

B. Current nanosatellite communication standards 11

1. Data overhead 11

2. Connection issues 14

C. The need for cybersecruity in nanosatellites 15

1. Data usage in nanosatellites 16

2. Nanosatellite communications information security standards 16

3. Nanosatellite communications information security assessment 18

D. Encryption and One-time-Pads 22

1. Evaluating the strengths of one-time pads 22

2. Limitations of one-time pads 23

3. One-time pads in nanosatellites 24

E. Chapter Summary 25

III. ENcryption Mechanism 26

A. Goals of encryption mechanism 26

B. Development of encryption mechanism 27

1. Mechanism development and platform 28

2. Mechanism design and operation 29

C. evaluating mechanism performance 33

1. Data sizes of encrypted files 34

2. Processing and iterations 35

D. Robustness to error in transmission 35

1. Insertion and deletion of data 35

2. Replacement of data 36

E. possible solutions for error propagation 41

F. BODY TEXT STYLES 42

1. Figures 42

2. Tables 49

3. Bulleted and Numbered Lists 51

4. Block Quotes 51

G. Table of contents 52

H. zotero, refworks and the like 53

I. blank pages 53

J. Cross referencing 54

K. equations 54

IV. SAMPLE CHAPTER 57

A. THIS IS A HEADING 2 57

1. Heading 3 57

2. Heading 3 60

B. This is a heading 2 61

appendix. Optional 63

List of References 65

initial distribution list 69

There should be at least two headings per heading level, or do not use the heading level

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Do not manually type your list of figures; update this list in the same manner described in Chapter I, Section D

Figure 1. TCP time diagram for transmission of 8 packets with retransmission of packet 4. 12

Figure 2. One-time pad example on alphabetic message of length 6 and a few possible solutions 22

Figure 3. Host system specifications for hosting development platform 28

Figure 4. Hypervisor and virtual machine specifications for platform 29

Figure 5. OTP encryption utilizing logical exclusive or (XOR) function on a single byte 30

Figure 6. OTP encryption scheme on an entire message 30

Figure 7. ASCII characters used to test and analyze the encryption mechanism 31

Figure 8. Test OTP utilized for development of mechanism 31

Figure 9. Function developed to encrypt a message of arbitrary length with a corresponding OTP 32

Figure 10. Function developed to decrypt a message of arbitrary length with a corresponding OTP 33

Figure 11. Inserting or deleting a single bit in the first byte propagates throughout all subsequent data until the end of the packet. 36

Figure 12. Replacing one or more bits can have a varying degree of impact on the data, but effects do not propagate to subsequent data 37

Figure 13. Function used to simulate individual bit flips in the OTP encrypted data and compared to original data 39

Figure 14. Function used to simulate burst bit flips in OTP encrypted data and compared to original data 40

Figure 15. Heading Levels and Their Associated Styles. Adapted from Hawks (2015). 42

Figure 16. A Basic Figure 43

Figure 17. A Figure with a Title and a Citation in APA Style. Source: Doe (2017). 45

Figure 18. Placement of Optional Secondary Captions in Figure Title 46

Figure 19. Variation—Multi-Line Figure Title, with First Sentence Only in List of Figures. Adapted from Doe (2017). 47

Figure 20. Variation—Figure Title above Figure 48

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Do not manually type your list of tables; update this list in the same manner described in Chapter I, Section D

Table 1. Styles to Use and Element Placement for Figures and Tables 6

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

COTS commercial off-the-shelf

IP Internet Protocol

TCP Transmission Control Protocol

UHF ultra-high frequency

VHF very-high frequency

UDP User Datagram Protocol

NERDP Nanosatellite Encrypted Reliable Datagram Protocol

LEO low Earth orbit

CSP CubeSat Space Protocol

OSI

ISO

BER bit error rate

CRC32/CRC 32-bit cyclic redundancy check

HMAC keyed-hash message authentication code

XOR exclusive or logical function

AES 128 bit Advanced Encryption Standard

3DES Data Encryption Standard

MEROPE Montana EaRth Orbiting Pico Explorer

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Required or recommended for the departments listed in our Abstracts versus Executive Summaries guidance [here](http://my.nps.edu/web/thesisprocessing/resources), under the “Style and Grammar” section. Please consult your advisors for their preferences regarding length and style.

The executive summary should be able to stand apart from the thesis. Therefore, if you include figures or tables in your executive summary, do not apply the figure or table styles to the titles. Instead, use **NORMAL** style and manually format the title to match the titles in the body of the thesis. This will keep these titles out of the lists of figures and tables, and allow the figure and table numbering to start at “1” in the thesis body, as required.

If you include *parenthetical* citations (APA style, for example) in the executive summary (but weaving your sources directly into your sentences is preferable), include a separate reference list at the bottom of the last page of the summary. Use the same citation/reference format as in the body of the thesis.

**References**

You may include a separate reference list for the executive summary or weave your sources into the narrative.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Optional for all departments. Acknowledgments may be more informal in tone than the main thesis text but should still follow correct use of sentence structure, grammar, and punctuation.

THIS PAGE INTENTIONALLY LEFT BLANK

# INTRODUCTION

## rESEARCH DOMAIN

Nanosatellites are small low Earth orbit (LEO) devices used to undertake space-based research in a cost effective manner. Nanosatellites typically have a mass of 1-10 kilograms, have a short life time of a few weeks or months in orbit, and are often constructed using commercial off-the-shelf (COTS) components. COTS components are typically inexpensive, readily available, and can be easily repurposed for space missions. The use of these components helps keep the mission prices low and allows for a larger number of research institutions to carry out experiments and demonstrations in LEO.

Currently, nanosatellites and their COTS components rely heavily on pre-existing and well established communication protocols. These protocols are the same ones used in ground based internet communications and build on the Internet Protocol (IP) stack. Specifically, researchers will use two of the most common protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). These protocols operate at a network level on all computers on the ground, and provide a framework for communication to automate the transmission and receipt of data.

TCP is a connection-based protocol, meaning that it relies heavily on a persistent connection even if the connection is noisy or prone to errors in the data. TCP provides key services that are fundamental to the transmission of data such as retransmission of lost or deformed packets, acknowledgement of data received, integrity checks, and the ability to assemble the packets of data in the correct order. To achieve this, each TCP packet will contain anywhere between 20 to 60 bytes of data as a header containing the relevant information needed by the receiver to carry out these functions.

UDP on the other hand is a lighter protocol that does not rely on a persistent connection. UDP is a unidirectional packet sent by a transmitter to a receiver without any information for retransmission, or correct packet ordering. If an object is fragmented into discrete packets and transmitted with UDP, unlike TCP, these packets may or may not arrive, and they may or may not arrive in the right order without any mechanism to verify their order, without a mechanism to acknowledge their successful arrival to the recipient, and no way for the recipient to request the retransmission of a specific packet. UDP does provide a checksum for integrity validation of the packet, but not much more data is transmitted in its 8-byte header.

These data packets are frequently transmitted by nanosatellites over ultra-high and very-high frequency (UHF and VHF) bands. These radio frequencies allow researchers and the operators of the nanosatellites to communicate with the devices in orbit at a low financial cost as transmitting and receiving equipment is COTS. By using these bands, nanosatellite operators can also reduce the power consumption and internal space footprint of the communications components within the nanosatellite.

Nanosatellites provide an accessible opportunity for more institutions to carry out space-based research. The devices have lower expenses than other space missions, small, and the components are readily available to anyone. Since the launch of the first nanosatellites in the early 2000’s, the benefits provided rely heavily on the low cost and profile of the devices. Furthermore, the ability to transmit and receive the data from the devices is beneficial to research institutions who would otherwise have no way to extend their research projects to space exploration. To this end, it is important that a standard in data communication and transmission for nanosatellites be established to broaden the scope of the research capabilities of the nanosatellites. Said standard should take into account the technical limitations of the nanosatellites, be flexible in its application due to the various nanosatellite designs, be a software based solution, and provide an efficient mechanism for communication that improves upon existing communications protocols.

## Research problem and Motivation

The popularity of nanosatellites is due to largely to their relative simplicity and affordability. Unfortunately, these benefits come at a cost. These costs translate to signal noise, low-bandwidth, high packet drop rates, and low overall mission data transfers. These costs exacerbate the situation by limiting the range and length of experiments accessible and available, and by limiting the usage of well-established IP communication schemes and encryption methods

To make nanosatellites more accessible to multiple research institutions, and to simplify the communication schemes, researchers have designed nanosatellites to communicate over amateur radio bands in UHF and VHF using a variety of radio protocols. As mentioned above, the use of these protocols and these bands means that there is a relatively low data transmission rate accessible for space to ground communications. Surveys done by two teams, Bryan Klofas et al. in 2008, and Paul Muri and Janis McNair in 2012, show that nanosatellites, specifically CubeSats operating in the UHF band, typically have a baud rate ranging from 1200-9600 [1], [2]. Regardless of whether these satellites use transceivers either custom-built for the specific mission by the research institute, or use prefabricated COTS transceivers, if operating on the UHF or VHF band, a common communication protocol employed is the AX.25 protocol.

Klofas’ survey, dated in 2008, shows a comparison summary of the various communication transceivers, frequencies, in addition to the baud rate of 18 different satellites operating in UHF and VHF bands. This survey also specifies the data link layer protocol, Open Systems Interconnection (OSI) Layer 2 as defined by the International Organization for Standardization (ISO), used by these satellites. More specifically, the survey shows that out of the 18 satellites included in the paper, 14 devices utilize the AX.25 protocol for amateur packet radio [1]. Muri and McNair’s survey, shows a database of 30 satellites launched in the 2009-2011 timeframe; of these devices 16 utilized the AX.25 protocol [2].

The AX.25 packet radio protocol ensures the delivery of packet data encapsulated in frames and managed by the transceiver. This protocol provides a standard for the intercommunication between various ground stations and satellites in either half or full-duplex schemes. Unfortunately, this protocol does not intrinsically provide any support for the implementation of the IP protocols such as TCP or UDP, as those operate on the OSI Layer 3, the Network Layer [3]. The lack of network packet management functionality provided by TCP or UDP in the AX.25 protocol means that these protocols typically have to be added on top of the existing OSI Layer 2 much like those same IP protocols have to be used in addition to the Ethernet frames in standard internet communications.

From a security perspective, nanosatellite communication schemes lack a cryptographic method that ensures the confidentiality of the data transmitted. While there are some solutions that provide encryption of data, such as CubeSec and GndSec solutions devised by Challa et al. in [4], these solutions are hardware based. Limiting communications to specific hardware configurations places a constraint in the design and flexibility of nanosatellites. While hardware implementation of encryption may be faster for certain encryption methods as stated in [4], a low impact software encryption mechanism would be more favorable as it can be independent of specific hardware constraints. Additionally, using encryption methods such as AES means that if a large file is encrypted and transmitted, the receiver would have to wait to receive the whole object before decryption can begin which may not be in the best interests of the mission.

As mentioned above, TCP and UDP have their drawbacks in design and applicability. TCP is heavily connection based protocol that requires a persistent, connection, ideally running in full duplex mode. This allows the transmitter to receive acknowledgements while it transmits data packets. Unfortunately, due to the limitations of the AX.25 protocol in the amount of possible data transmitted per frame, the relatively higher noise rate of the UHF and VHF bands, and the size of the TCP header, TCP become unwieldly for nanosatellites with lower baud. At 9600 baud, a nanosatellite can transmit 9600 bits per second, and at half duplex this could present a large data cost to an already limited bandwidth.

An OSI Layer 4, Transport Layer, solution has been proposed by members of Aalborg University in Denmark called the CubeSat Space Protocol (CSP) [5]. This protocol was developed in C and modeled after the IP TCP standard and includes a header that is only 4 bytes long and supports eXtended Tiny Encryption Algorithm (XTEA) encryption and is designed to successfully integrate with several physical layer technologies. While this protocol does provide some additional functionality at a lower cost, it is limited to the specific physical layer drivers and is more centered towards network operations. This is reflected by looking at the packet structure and noticing that it uses 22 bits out of the available 32 just to establish a source, destination, and their corresponding ports [5]. Since most of the source and destination addressing can be done at the OSI Layer 2 for most radios, it is inefficient to use that much of the packet header in a redundant manner. Furthermore, CSP reserves several ports for buffer status, pings, and other network functions that may not be a priority for researchers or can again be derived from the radio protocol used. The use of XTEA does not allow partial decryption, as described above, and limits data validation to only after the entirety of the object has been downloaded. CSP documentation found in [5] does not readily outline the mechanism for packet receipt acknowledgement, packet retransmission, or data integrity checksums.

## Research Questions

The following questions are key for this investigation:

1. What are the processing, data overhead, and encryption costs of current communication protocols?
2. What are the processing and storage costs associated with using a one-time pad for encryption in nanosatellite communications, and how do they compare to CSP and XTEA?
3. Does the NERDP reduce the amount of data overhead and result in faster transfer times and/or a reduced number of packet exchanges than TCP?

## scope

The scope of this thesis is to investigate the technical needs of the small satellite and nanosatellite community operating in the UHF and VHF bands, focusing on their bandwidth their limitations and developing a versatile lightweight software solution that can meet those needs and increase the productivity of the satellite, labeled as the Nanosatellite Encrypted Reliable Datagram Protocol (NERDP). Focus will also include investigating the addition of confidentiality to the data payloads using a pre-loaded one-time pad (OTP) increasing the cybersecurity strength of the communications scheme. Development will target a software solution that can be run on COTS components, measure the performance of the OTP encryption, add integrity checks for the data transmitted, and add reliability to the data transmissions while maintaining hardware limitations in mind.

## Approach

The process used for this investigation determined the current limitations in the transfer of data from nanosatellites deployed by the Naval Postgraduate School Space Systems Academic Group, and a survey of protocols used and the challenges encountered. This focused primarily on the application of TCP and UDP as the main protocols for data transfer, as none of these satellites support encryption. The NERDP prototype developed then focused on demonstrating TCP-like functionality in data packet reliability and retransmission at a lower cost in data and performance in UHF and VHF. This prototype was developed to operate as a proof of concept in a virtual network with limited applications, but with a modular approach that and support the addition of increased functionality depending on mission requirements. NERDP was designed to operate strictly in OSI Layer 3 and higher, leaving the Data Link Layer to the hardware specifications. For the information assurance component of the prototype, and independent module using OTP encryption was developed and its performance was measured. This was done independently of the overall protocol as the protocol can support it and other types of encryption, but does not necessarily require it. The conclusions and performance assessments can be found in Chapters VI and VII.

## Thesis Structure

The remainder of this thesis is structured as follows:

Chapter II continues the discussion of bandwidth in UHF and VHF bands further outlining the problem space, includes a brief survey of current communication schemes and notable nanosatellites and CubeSats relevant to this thesis, discusses the need for cybersecurity in nanosatellites and outlines the status quo, and discusses the different methods of encryption with a particular focus on OTP.

Chapter III discusses the methodology for development, goals, and robustness of the OTP encryption algorithm designed for this thesis.

Chapter IV discusses the methodology of the development of the NERDP, the structure, reliability mechanisms, and the data overhead reduction of the Network Layer software based protocol proposed in this thesis, NERDP, and includes a comparison to other IP protocols.

Chapter V discusses the data analysis of the error propagation simulated in the encryption algorithm, and the data collected in the FM band testing.

Chapter VI summarizes the results of the encryption scheme and NERDP as functions of overall system performance. This will evaluate the systems costs and their feasibility along with any potential cybersecurity vulnerabilities.

Chapter VII will provide main conclusions arrived on the applicability of the prototype and encryption scheme proposed, and outline the future work and next steps.

# Background

This chapter discusses bandwidth limitations in the UHF and VHF bands by exploring typical nanosatellite communication schemes. The root of the issues is discussed, and notable nanosatellites and CubeSats are explored. These surveys provide further context of the problem space and the limitations currently encountered by nanosatellite developers. The text also provides a brief overview of cybersecurity and information assurance in nanosatellites, and a discussion on encryption with a focus on one-time pads.

## Problem Space: Low Bandidth in UHF and VHF bands

As described in [1] and [2], most nanosatellites communicate in the UHF range and have a baud rate typically of 1200 to 9600. Several factors limiting this baud rate include, but are not limited to the hardware used, the power available to the communications array, antenna type, time window for communication, and angle on the horizon. Variations in all of these factors can create not only fluctuations in the baud rate but also in the quality of the signal. Lower signal quality introduces random noise and errors, typically in the form of flipped bits in the payload, and can compromise the integrity of the overall object being transmitted. This loss of packets due to signal noise, measured as bit error rate, is part of the reason some nanosatellites use protocols like TCP or CSP as they allow for the retransmission of lost packets and packets deemed too compromised.

### Common Nanosatellite Frequency and Bit Rate Ranges

The UHF an VHF bands are defined by radar-frequency letter band nomenclature, and also by the International Telecommunications Union (ITU). These nomenclatures, while similar, can lead to some confusion. Radar nomenclature identifies the VHF band as a frequency range of 30-300 MHz, the UHF band as 200-1000MHz, the L-band as 1 - 2 GHz, and the S band as 2 - 4 GHz. Meanwhile The ITU nomenclature, while maintaining the same definition of the VHF band range, groups any frequency between 300 Mhz - 3 GHz as UHF [6]. Revisiting the surveys by Klofas et al., and Muri and McNair, shows that most CubeSats and nanosatellites transmit at the 435 MHz frequency [1], [2]. In the Klofas survey, of the 18 satellites examined, all but 3 devices operated on the range between 400.375 – 437.880 MHz with the outliers operating at 902 – 928 MHz and 2.4 GHz [1]. Muri and McNair, also showed similar results, with only 10 out of the 30 satellites recorded not operating in the ~437 MHz. frequency [2]. Researching this distribution further reveals that in an update to the Klofas’ survey to include CubeSats launched between 2003 - 2014, 112 out of 172 total transmitters recorded operated in the 437 MHz amateur radio frequency range, with an additional 40 devices still operating below 1000 MHz [7].

Bit rate is measured in amount of bits transmitted per second (bps) or baud rate, and is used to determine the rate at which data can be transmitted. On ground based systems, such as the internet, speed is typically measured in the megabit range (millions of bits per second) but due to the low power and the limited hardware of the nanosatellites, these ranges typically fall into the (kilobits per second) range. Again, the Klofas, and Muri and McNair surveys expose the data rates of several satellites. More specifically, out of 144 transmitters recorded by Klofas, including the other surveys, 121 transmitters operated at 9600 baud or less, with the second most common rate being 1200 bps [1], [2], [7]. These low bit rates are why these devices are labeled as low-bandwidth for the sake of this problem space and part of the reason why reducing data overhead is so important and significant.

Due to the prevalence of the 437 MHz frequency and a typical baud rate of 9600 or less in both early and more current nanosatellites, research and development of communication protocols should strive to operate at those target specifications. These specifications seem to provide the most cost effective hardware and communication packages for nanosatellites, as reflected by their popularity, but simultaneously also limit the usefulness of these devices. If experiments collect too much data, then it may be unfeasible for the data to be transmitted to the ground recipient. The problem is only exacerbated when a large portion of this limited bandwidth is needed to retransmit a large number of packets due to poor connection, and each of these packets has an unnecessarily large header attached to it.

### Bit Error Rate and Packet Loss

Data rates in satellites are dictated by the available power to the communications system, signal quality, distance between receiver and transmitter, atmospheric conditions, and many other factors. These factors impact the already limited bandwidth of the COTS components in nanosatellites and introduce errors in the bits transmitted. These errors can be resolved through error correcting schemes, and through data retransmission. These unavoidable occasional retransmissions is why protocols like TCP are preferred over protocols like UDP.

Error rates in data transmissions are known as bit error rate (BER), and are defined as the ratio of incorrect bits received divided by the total number of bits transmitted. This ratio is useful in evaluating the performance of the communication systems and to estimate the need for retransmission and error correction. In a 2012 report, authors Selva and Krejci utilize an estimated BER for calculations of approximately 10-5 [8]. While this estimate is only valid for a specific modulation used by the authors, it does give insight into the minimum BER expected for satisfactory performance by the communication devices.

BER impacts the integrity of specific bits that are transmitted, these bits then compromise entire bytes, and these result in compromised packets. Due to the low power of the transmissions, it is also possible for packets to not reach the ground station at all. These total packet losses result in missing data and, in the case of TCP, result in the ground station requesting multiple retransmission of packets. This constant change of state of the radio from receiving to transmitting accrues a time loss if the signal quality is poor enough to require multiple retransmissions. Furthermore, nanosatellites have a limited window of approximately 45 minutes of contact with the ground station per day. If changing states of the radio takes 1 second to transition between states, then the two seconds it takes to request a retransmission is 0.074% of the total time available per day. If an object requires multiple retransmissions to ensure integrity, then this accrued time from state switching is detrimental to the performance of the communication system. As described above, BER is unavoidable and by consequence so are retransmissions. Therefore, to ensure optimal data transfers, a protocol that improves on the TCP model and reduces the number of state changes would provide a better solution.

## Current nanosatellite communication standards

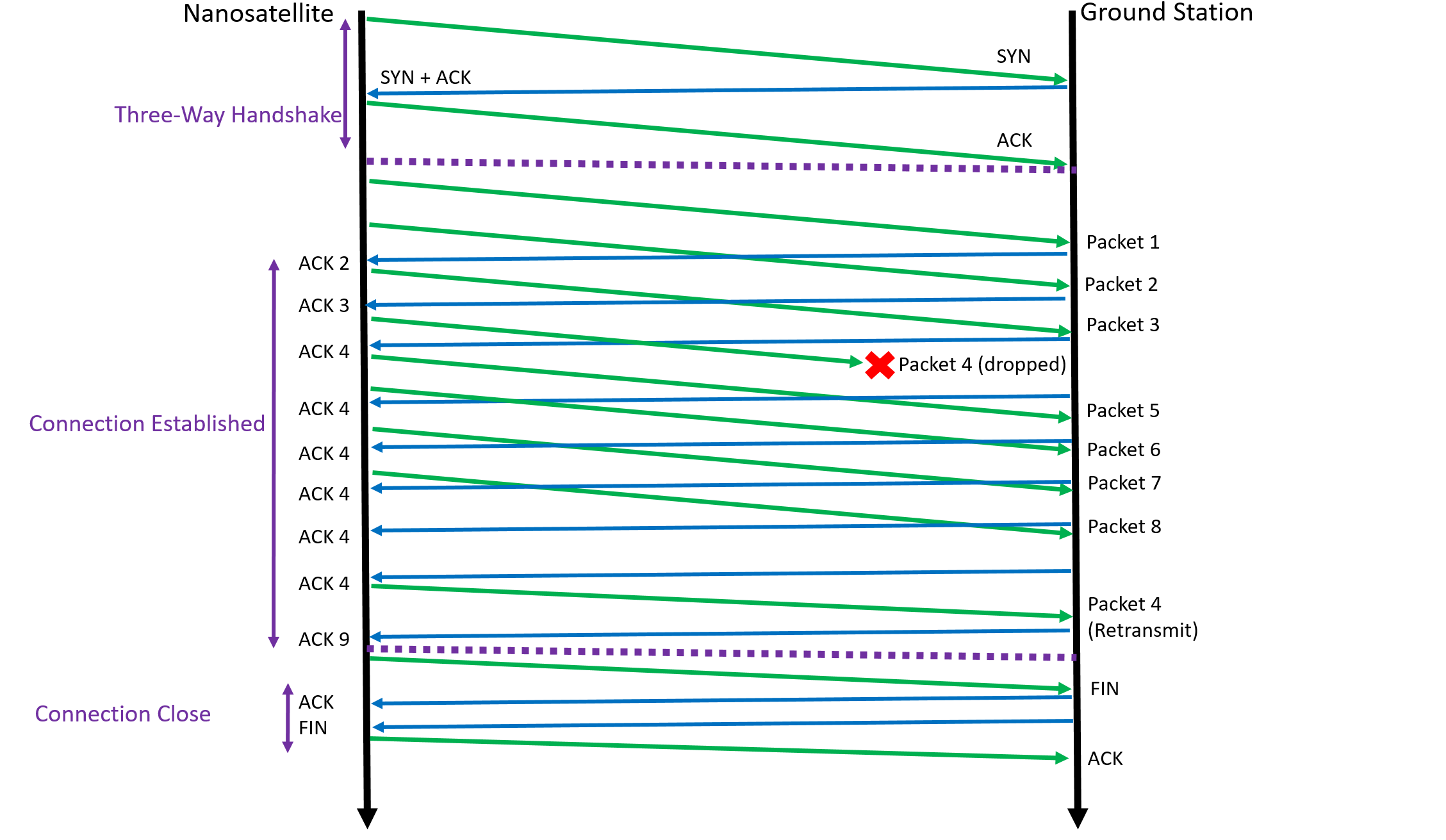
Current nanosatellites typically use the AX.25 packet radio protocol, as discussed above, and will sometimes encapsulate a Layer 4 protocol such as CSP, TCP, or UDP. Each of these Layer 4 protocols has advantages, disadvantages, and applicability, but all have a data overheads required in transmission. This overhead reduces the amount of data that can be transmitted by the satellite, and adds functionality not always needed in from the nanosatellite. Additionally, some of these Layer 4 protocols cause connectivity problems if the connection is unstable or reliable and compound the problem of reliability and retransmission, further increasing the accrued data overhead.

### Data overhead

Due to the various designs and OSI Layer 2 implementations, such as AX.25, the calculations for optimizing data overhead focus on Layers 3 and 4. These layers, the networking and transport layers, provide the infrastructure for transferring data packets, and for dictating their behavior. In typical internet applications, Layer 3 is responsible for routing and packet forwarding structures like IPv4, while Layer 4 provides the architecture for the connection behavior in protocols such as TCP and UDP. Anything higher than Layer 4, for all intents and purposes in nanosatellites, can be considered payload data, though it should be noted that the header of Layers 3-4 is often included as part of the payload along with Layers 5+ when viewed in reference to the Layer 2 protocol.

Since nanosatellites use AX.25 for the delivery of packets, and are largely point-to-point communication schemes, a networking layer that includes routing information can be foregone as this layer can be used to route packets to various IP addresses in the same network, and even make the transition through different routers. Point-to-point communication through packet radio carried out through AX.25 does not require routing or communication with multiple nodes, therefore the implementation of a header, such as an IPv4 is not necessary and abandoning it can reduce the overhead by 20–60 bytes [9].

Abandoning the need for a Layer 3 protocol introduces some challenges for IP based transport layer protocols. TCP is reliant on a persistent IPv4 based connection, and its data header includes information on the source and destination IP addresses and ports. This information supporting the range of functionality of TCP results in a header of 20-60 bytes [10]. Using the above information, a transport layer protocol that is independent of the network layer can reduce the data overhead of each packet transmitted by 40-120 bytes. In relation to packet loss and retransmission, the costs of IP/TCP overhead accrue quickly. A time diagram of TCP transmission with packet loss, either from integrity failure or packet drop, (Figure 1) demonstrates the overly verbose nature of TCP that leads to a large amount of packet transmissions.



1. TCP time diagram for transmission of 8 packets with retransmission of packet 4.

Each packet transmitted under TCP will have an IP header and a TCP header. Assuming no options, the total headers for Layers 3-4 in this scheme is 40 bytes, or 320 bits, per packet. In 8 packets this data accounts for 2560 bits. At 9600 bits per second the data overhead accounts for 26.7% of the data transmitted per second, assuming the baud rate is negligibly affected by the Layer 2 AX.25 protocol.

UDP, the other popular IP protocol in nanosatellites, is a connectionless protocol that still relies on the IP infrastructure of Layer 3. This Layer 4 protocol uses one-way datagrams to transmit data between two nodes. These datagrams provide a header per packet that includes source and destination ports, much like TCP, and also provides and integrity check for the data transmitted. The drawbacks of this protocol include the lack of functionality for retransmission and correct packet assembly order. TCP utilizes sequence numbers in the headers to assemble the packets in the correct order and detect if a packet is dropped. UDP’s lack of sequence check creates a challenge for data retransmission and object reconstruction. TCP also provides functionality to ensure the delivery of the packets through the form of acknowledgements per packet, while UDP has no such mechanisms. The advantage of UDP is its simplicity and significantly smaller header than TCP. Assuming 20 bytes are still used for the IP header, UDP only requires an additional 8 bytes as a header as opposed to the 20 required by TCP [11].

It should also be noted that the standards for both UDP and TCP outline 2 bytes for each of the destination and source ports in the protocol. These two bytes, or 16 bits, are unsigned integers and result in 216, or 65536, possible ports for data receipt and transmission [10], [11]. Such a large number of ports is useful in internet and network communications, but may be excessive for use in nanosatellites. A protocol with a reduced number of ports would reduce the overhead in headers at little to no cost in functionality.

CSP, as described earlier, is a protocol designed specifically to be used with nanosatellites and CubeSats. This protocol provides support for integrity checks through a 32 bit cyclic redundancy check (CRC32 or CRC) and keyed-hash message authentication codes (HMAC), flags to signal if packets are encrypted, and 12 bits for destination and source port assignments (26 = 64 possible ports) [5]. This functionality is all outlined in the protocol header which is only 4 bytes, 32 bits, long. CSP provides retransmission functionality and encryption support, and can be used independently from an IP layer. This reduces a header of 40 bytes of IP/TCP by 90% to only 4 bytes.

Looking closer at the mechanisms of CSP, it becomes evident that the 4-byte header is a misleading statement. The header itself only contains a single bit flag denoting if the packet is encrypted, if a CRC is included in the payload, or if the packets have an HMAC, without in fact containing any of these checks within the header itself [5]. If a packet is designated with a CRC32 then the payload data will include 4 additional bytes of information doubling this “non-payload” overhead; similarly if a packet is flagged to contain an HMAC, this will add 2 bytes of data to the overhead potentially increasing the header from 4 bytes to 10 [5]. Additionally, the documentation of CSP is unclear how much overhead the retransmission infrastructure would add to the total overhead

Data overhead is important in these situations where the baud rate is limited to a noisy and error prone 9600. While land based communications can reliably use TCP and UDP for IP based communication, the overhead accrued with them is too high for a limited connection. These protocols also provide unused functionality in point-to-point connections that results in additional space that could be better utilized by the protocol. Other protocols like CSP promise small headers and increased functionality, but upon closer inspection fail to disclose the structure and variable “non-payload” data accrued in their functions. This data overhead in turn, while still lighter than IP-based protocols, still leaves room for improvement in reducing the overhead.

### Connection issues

While all protocols discussed do suffer from connection issues such as error rates and packet loss, the delay in packet transmission and acknowledgement of receipt in TCP creates a specific problem that is exacerbated by the potential delays in transmission of packets. Due to the distance, fleeting window for transmission, and the delays in change of state in the radio hardware, there is a possibility that the TCP connection times out from inactivity or failure to receive the proper acknowledgement. Figure 1 demonstrates the state dependency of TCP, which can have a negative impact on the performance of the system.

While TCP timeouts can be set by the user to extend or shorten the time “transmitted data may remain unacknowledged before a connection is forcefully closed” [12], these values are user defined and can vary from application to application. Nanosatellite designers could decide to implement the IP/TCP model on the AX.25, as described above, with a long TCP timeout wait to ensure the connections aren’t dropped. This creates the problem of resource allocation and the state dependency of TCP. If a connection is kept alive for too long, there is the possibility of resource exhaustion since all of the resources will have to be allocated and maintained. The constant change of TCP between packets and acknowledgements can also create a resource allocation problem where power consumption and time are excessively consumed. Conversely, if the TCP timeout is set too short, there is the possibility of connection timeout any time the nanosatellite loses connection with the ground station or the connection is poor. If a connection times out, the connection must be reestablished through a three-way handshake, and the file download must be restarted. These increase the data overhead, and detract from the useful windows of the nanosatellite.

UDP does not suffer from this problem of timeout and reliance on persistent network connectivity nor does it rely on the state of the transmitter and receiver, but again does not have any higher functionality. The documentation is unclear on whether or not CSP employs a connection timeout, nor does it divulge how communications are initialized in comparison to the TCP three-way handshake. Regardless, a protocol designed to take into account the state of the transmitter and receiver, carefully weigh the limitations and benefits of a connection timeout, and provide an infrastructure for state recovery would be beneficial for nanosatellite communications.

## The need for cybersecruity in nanosatellites

Bandwidth limitations and unreliable connections are not conducive to a strong cybersecurity posture that ensures data confidentiality, integrity, and assurance. The approach, “any data is better than no data” reduces the applicability of compression, encryption, and integrity checks on data being transmitted and received. The application of a stronger security posture is not new to nanosatellites, as evidenced by the integration of XTEA encryption in CSP, but few cases exist of other cybersecurity methods to safeguard the data being transmitted. The few cases surveyed demonstrate a preference to hardware and radio solutions instead of software solutions. A software solution that provides the functionality and infrastructure for a stronger cybersecurity posture would be a welcome paradigm shift in approaching communication schemes of nanosatellites.

### Data usage in nanosatellites

The amount of data transmitted by a nanosatellite is largely governed by its baud rate, lifetime, and orbit. These conditions can vary dramatically from mission to mission and design specifications of the nanosatellite. Looking at the first one hundred CubeSats in 2013, Michael Swartwout determined that the average lifetime of nanosatellites is typically less than 200 days [13]. Additionally, Selva and Krejci assume an average access window of 5 minutes [8]. Assuming that there are 9 passes total per day on an orbit, the total window of a nanosatellite can be estimated.

Extending the duration of the orbit to a calendar year, 365 days and assuming 45 minutes of access per day at a baud rate of 9600 the total data transferred in bits can be estimated for a single year to 1.183 gigabytes. This is the total data transmitted by the satellite including the headers of protocols. Assuming that the actual payload of the data is encapsulated by AX.25, and protocols like IP/TCP, then the actual useable data is less than these 1.183 gigabytes.

### Nanosatellite communications information security standards

Currently there is no clear standard for information security in the transmission of data from nanosatellites and CubeSats to ground stations and the current methods offer few security features [4]. This lack of standard impedes a clear and thorough assessment into their shortcomings and methods on which to improve those shortcomings. A survey into the security protocols of CubeSats shows a preference towards hardware base implementations of security in the data transmitted.

Information security consists of three components:

1. Confidentiality
2. Integrity
3. Availability

Confidentiality refers to the property of the system to only allow authorized users or parties to access the data. For data to be considered confidential and secure, this property must be maintained at all times even if the data is transmitted across a network or between nanosatellites and ground stations. A common method to ensure the confidentiality of data transmitted is through encryption. Encryption ensures confidentiality through hard-to-solve mathematical cryptograms, by making the solutions to the cryptograms too complex for an adversary to solve in a reasonable amount of time, but allowing the intended and authorized parties with the correct keys to access the information.

Data integrity is the property of the system that ensures the data is not tampered with in transit, storage, or at any other time by unauthorized users or environmental noise. In the case of nanosatellites, integrity allows verification that the data transmitted and the data received between nanosatellites and ground stations is equivalent to the transmitted data. A common mechanism to integrate this property into systems is the inclusion of a CRC on each packet of data transmitted. This checksum allows the receiver too verify if the data was altered at any time between transmission and receipt.

Availability is the property of the system that ensures data is available when requested. Consuming an excessive amount of system resources can create a denial of service situation where authorized users cannot access the information. Exhaustion of memory, bandwidth, processing power, and signal interference are all mechanisms that can be used to affect the availability of information between nanosatellites and ground stations.

Information security in nanosatellites largely focuses on the confidentiality properties of the communication system. Integrity is easily achieved in the datastream by including a CRC on each packet transmitted, while availability impacted through FM interference is a subject field all on its own. To this end, information security is reduced to confidentiality, specifically the impact encryption has on the ease of transmission. It should be noted that confidentiality does in fact play a small role in the integrity and availability of data transmitted. If a large object is encrypted successfully, but takes a long time to transmit, while the integrity of each transmitted and received packet may be easily verified, neither integrity or validity of the data within the object can be verified until the whole object is received and decrypted. This could lead to a situation where the bandwidth is exhausted by the data transmission only to result in poor or useless data and a waste of limited resources. In another scenario, if the encrypted data is only partially received and the nanosatellite window ends, while each packet can be checked for integrity there is no way to ascertain the validity of the data being received until all of the object is received. Because of these limitations, a protocol that encrypts a stream of independent bytes, rather than the object as a whole would be preferable. Such a protocol would allow the constant decryption of data as it is being received and allow for data checks to be carried out on partial and incomplete data.

### Nanosatellite communications information security assessment

A survey of information security systems in nanosatellites and CubeSats is inconsistent and unfeasible due to the various protocols carried out by the hundreds of satellites, and due to the small sample size of actual documented implementations of information security protocols. As described above, integrity and availability mechanisms can be easily surveyed in protocols like TCP and CSP, as they all account for packet repeatability and support checksums, but their approach to confidentiality through encryption is not as clear cut. The approach to confidentiality is further complicated through the addition of hardware based confidentiality instead of software based mechanisms. A survey into CSP, CubeSec and GndSec, and the MEROPE CubeSat system illustrates the challenges of implementing confidentiality mechanisms into nanosatellites and provides a measure with which to evaluate the performance of other protocols and mechanisms.

CSP is designed to support the XTEA encryption algorithm. XTEA was introduced by the TEA designers David Wheeler and Roger Needham as a solution to correct two weaknesses in TEA [14]. Like its predecessor, XTEA is designed to be minimal while still providing a high level of confidentiality on information. It is a symmetric block cipher with a block size of 64-bits and a key size of 128-bits [14]. In CSP the keys are shared before the launch of the system and can be updated by using the previous keys to exchange a new key. CSP headers have a flag signaling if the packets are encrypted, with no other cryptographic information being exchanged. This allows for data packets to be encrypted and secure within a strong key space, but several attacks are documented against XTEA that would break the confidentiality of the data stream. XTEA encryption is based on the number of rounds used to encrypt the plaintext, increased rounds provide stronger security but come at an increased computational cost. This computational cost makes XTEA deceivingly small the level of security is entirely dependent on the computational power as denoted by the number of rounds undertaken to produce the cipher text. Another detriment to XTEA is the size of the block. As a block cipher, it must use blocks of a predetermined size in its algorithm. At 64 bits, or 8 bytes, this is a large block, especially if the packet sizes of each data packet is small. In the event that a one byte segment of information needs to get encrypted, that means the block would have to be padded with 7 bytes of null information. The addition of these blocks could potentially increase the size of the data transmitted in an already limited bandwidth environment. XTEA in CSP operates as a cipher in counter mode [5]. In this mode each block is encrypted independent of one another through a series of exclusive logical or functions (XORs) and summation to keep a successive counter of blocks successfully encrypted. This allows for the parallelization of encryption for faster encryption schemes, but again it does come at a cost in system memory and processing power. In error propagation, if a cipher is run and the cipher text is downloaded without integrity check, XTEA in counter mode does guarantee that the error propagation ratio between cipher text and plain text is 1. This means for every byte affected in the cipher text, only the corresponding byte in the plain text will be affected upon decryption [15]. This is a valuable feature for an encryption scheme that has to operate under very noisy conditions, and give XTEA a preference over other encryption mechanisms that propagate the errors during encryption to two or more blocks [15]. In 2004, Ko et al. published a vulnerability of XTEA that could lead to a complete compromise of XTEA in data that has undergone 27 rounds of XTEA [16]. This vulnerability would allow the use of related keys and differential analysis of the encryption mechanism on 27 rounds of XTEA with a success rate of 96.9% [16]. To circumvent this vulnerability, XTEA would require more than 27 rounds, and thus significantly increase the associated processing cost of confidentiality. In 2009, Lu presented a related-key rectangle attack on 36 rounds of XTEA [17]. This attack, much like Ko et al.’s attack, would require an increased number of rounds in XTEA to ensure confidentiality. This is a tremendous burden for a low power system onboard a nanosatellite that could leave transmitted data vulnerable.

Developed by Challa et al., the CubeSec and GndSec security solution is described by its developers as “very light-weight” and provides authentication, confidentiality, and integrity through the use of symmetric pre-shared keys [4].The proposed solution by the authors uses Advanced Encryption Standard (AES) and Data Encryption Standard (3DES) in Galois/Counter Mode (GCM) and is implemented through hardware [4]. The reason for hardware implementation of these block ciphers is due to the high processing and time cost associated with AES and DES hardware, which the authors document in [4]. Using microcontrollers to encrypt the data and spare the processor from computing power is a resource efficient approach, but still comes with some associated costs. While methods like XTEA are directly measured in computing resources, the CubeSec and GndSec mechanism’s cost is in weight and volume on the spacecraft. The authors profile the encryption hardware with a footprint of approximately 5cm by 5cm and a total weight of approximately 9.6 grams [4]. While this footprint may seem trivial in larger spacecraft, the authors also recommend a redundant backup system that effectively doubles this physical footprint and can be a serious detriment to nanosatellites [4]. Additionally, the authors do not discuss the financial costs of the additional hardware, which should be taken into consideration given that the hardware is not recoverable after a mission. Some of the advantages offered by this system is the strong implementation of security through AES and 3DES operating at 128 bits. Additionally, much like XTEA in counter mode, GCM allows for parallelization of encryption, resulting in much higher encryption rates, while keeping the encryption costs within the hardware implementation and not severely impacting the power consumption of the spacecraft as a whole. Overall the CubeSec and GndSec system provides a valuable solution to information security, but at a cost in space, weight, and system complexity that may keep it out of reach from institutions.

An interesting case is the Montana EaRth Orbiting Pico Explorer (MEROPE) CubeSat built by the Space Sicence and Engineering Laboratory at Montana State University [18]. The journal article goes at great lengths to explain the need for COTS subsystem designs in CubeSats to mitigate the lack of expertise in CubeSat design teams. The communication subsystem design goal was to have a device with a low volume profile that communicates using the AX.25 protocol [18]. Analyzing the design and performance specifications described by the authors, it is clear that the MEROPE CubeSat did not have a mechanism to provide confidentiality to the data it was transmitting. The lack of such a protection and goal to make the MEROPE communication subsystem as COTS as possible, indicates a serious vulnerability in the design of the MEROPE and in other CubeSats: most teams lack a network design and information assurance specialist. While the MEROPE team was well versed in the design and application of AX.25 protocols and was able to build the communication subsystems, they do acknowledge their lack of technical expertise and the driving factor it was in the selection of their COTS communication subsystem. This assessment indicates the vulnerability of not only MEROPE, but also of other CubeSats. The community seems to lack a clear information security standard which could be explained by a lack of information security professionals actively involved in the development of the satellites.

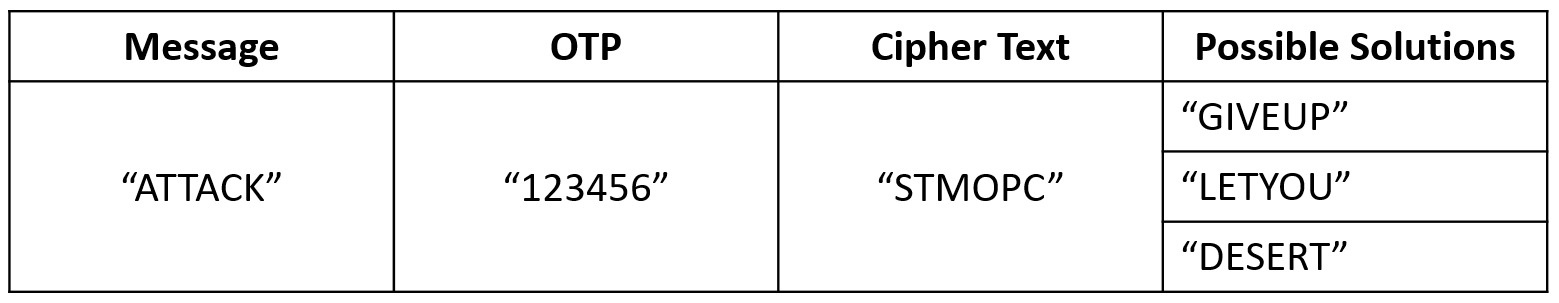
Overall the survey of these systems indicates a serious need for information security standards that provides a high degree of confidentiality. While no system implementation comes without a cost, designing a protocol that minimizes the costs of current systems would be an asset to the community. Such a protocol would require the participation of information security professionals and nanosatellite designers to ensure a high degree of information assurance, keep within the operational parameters of designers, and maintain the functionality provided by other more data expensive protocols. Such a solution could provide an open source flexible standard that can be used by any design team regardless of technical expertise.

## Encryption and One-time-Pads

Encryption provides information security to data stream through cryptography. The strength of encryption varies between encryption mechanisms and the many modes they run on. Some provide stronger encryption, making them really hard do crack but come at a large cost in memory and processing power, while others are light weight but have vulnerabilities. The strength of the encryption mechanism is typically measured by the ability of the adversary or unauthorized party to decipher what the data being stored is within a reasonable amount of time. As processing power continues to increase, the strength of these mechanisms falters, and stronger, more computationally expensive systems are required. There are encryption mechanisms that are classified as “perfectly secure” that can be implemented easily. These mechanisms are defined as perfectly secret as an encryption scheme due to the fact that the cipher text reveals nothing about the plain text, and that a given cipher text can be translated into any plain text of equal length to the cipher text with all possibilities equally mathematically probable [19]. A one-time pad (OTP) is such a mechanism.

### Evaluating the strengths of one-time pads

OTPs are, as described above, perfectly secret. This means that a string of length *n* when encrypted with a OTP of the same length, produces a cipher text of equal length. If an adversary intercepts a cipher text encrypted with a OTP, and assuming the message is limited to capital alphabetic characters, any combination of letters is equally probable (Figure 2).



1. One-time pad example on alphabetic message of length 6 and a few possible solutions

OTPs are also efficient methods of encryption as each byte of information is encrypted only once in an XOR operation. This eliminates the need for multiple passes to ensure a high level of confidentiality, at a low processing cost. Furthermore, unlike block ciphers with fixed block sizes that result in padding of data and extra data being sent, OTPs do not alter the length of the message being sent. These properties arise from the fact that OTP encryption encrypts each byte individually and independently from the rest of the data [19]. This increases the encryption strength and also limits the propagation of errors as each affected byte in cipher text will only affect the corresponding plain text byte upon decryption. OTPs are to this day the strongest method of encryption.

### Limitations of one-time pads

OTPs have certain limitations to ensure their perfect secrecy and limitations that limit their proliferation into practical uses. In 1919, Gilbert S. Vernam was awarded a patent for an encryption mechanism using a OTP and the XOR operation [20]. This system would encrypt a message with a OTP stored in a punch tape stored in a loop, which was later revealed to be vulnerability. By storing the OTP in a loop and reusing the key, cryptanalysis was possible as the key and character combinations were bound to be repeated in a cyclical manner, allowing adversaries to crack OTP encryption in Vernam’s device [21]. In order to mitigate this vulnerability, the OTP key must be non-repeating or reusable and must also be truly random. These two criteria must be true for the entirety of the OTP, meaning the OTP must be at least as long as the total data transmitted through the mechanism. This drawback prevents the practical implementation of a prolonged use of OTP for the transmission of large volumes of data, as this rapidly increases the required size of the OTP. Another detriment of using the OTP for the transmission of large amounts of data is the need for the OTP to be truly random. If a pseudo-random number generator is used, like the large portion of random number generators in computer system, the adversary may be able to correctly deduce the pseudo random number generator and seed. This would result in the adversary being able to predict and effectively break the OTP encryption of the data. Truly random numbers can be generated through entropic processes such as radioactive decay or quantum events, and can be difficult to generate. This can be mitigated with large repositories of existing random numbers, but again, this presents the opportunity for an adversary to deduce which repositories are being used. Another challenge for OTP usage is the need to exchange the OTPs with the keys between the users. Asymmetric encryption mechanisms allow for the establishment of secure tunnels so that keys can be exchanged and create tunnels of information that are encrypted through symmetric keys. OTP transmissions would either still require asymmetric key mechanisms and a large transfer of data for the contents of the OTP, or some physical exchange of OTPs. This presents a challenge since the nodes transmitting and receiving may not all be physically accessible.

These are a few of the limitations of using a OTP as an encryption mechanism. Despite its strength and perfection, practical limitations make the deployment of OTP encryption mechanisms, especially in larger data transfers as we see on the internet today. There are mitigation techniques to overcome the limitations of OTPs, such as the availability of large storage disks and large repositories of quantum information. Exchanging keys, presents a physical problem that can be avoided if the original OTP was large enough to accommodate the total lifetime data transmission and the keys were exchanged once. Overall, OTPs are a strong albeit slightly impractical encryption mechanism that compensates for their logistical hurdles through the level of security they provide.

### One-time pads in nanosatellites

Nanosatellites are prime candidates for the implementation of a OTP encryption mechanism. Their design and operation conditions are ideal for OTPs, and such a mechanism would provide the security needed by the spacecraft.

Several of the drawbacks of using a OTP presented above, can be effectively mitigated just through normal satellite operations. A OTP remains valid as long as the OTPs used by the receiver and transmitter are kept secret. In the case of nanosatellites, the vulnerability of an adversary obtaining the OTP is drastically reduced as one of the OTPs will be in LEO. Another shortcoming described was the need to have a OTP be as long as the total data transmitted throughout the life of the mechanism if key exchanges are to be avoided and the OTP must be full of truly random numbers. As described above, the total data usage of a nanosatellite in a year can be estimated to be about 1.183 gigabytes. Even if a nanosatellite mission has a lifetime of several years, data storage is currently compact enough in solid state media that a device storing a large OTP would not be a problem. In the case for the need of truly random data, several universities provide free open repositories of terabytes of quantum data to be used as random data. This can mitigate the need to build a mechanism to generate the data, especially if it such repository data can be made private.

The drawbacks of OTP encryption make it impractical for use in large transfers of data over large networks such as the internet. Point-to-point communication between a ground station and a nanosatellite with limited bandwidth and total lifetime data transfers present ideal candidates for the implementation of a OTP encryption mechanism. These mechanisms provide perfect secrecy, a high level of confidentiality, are lightweight, and can

## Chapter Summary

A survey into the current state of CubeSat and nanosatellite communications, demonstrated the need for information security standards, and the need for lighter protocols due to the limited bandwidth of the devices. Designing a lightweight protocol for use with nanosatellites has to take into considerations the large number of constraints in data transfer rates, error rates, and processing and transmitting power available to the spacecraft while keeping in mind the design and data transfer needs of the designers. Mechanisms like IP/TCP provide the functionality at a high overhead cost, while on the information security side, encryption mechanisms are a constant balance between weight, power, and processing costs. Nanosatellites provide a unique opportunity to establish a new protocol for low bandwidth communications that provides the necessary functionality and that integrates the infrastructure needed for an encryption mechanism based on a OTP. Communicating at 9600 baud over UHF and VHF is an error prone, slow connection that is currently without a clear standard. To remediate this the Nanosatellite Encrypted Reliable Datagram Protocol (NERDP) is proposed.

# ENcryption Mechanism

This chapter discusses the goals, design, and development of the encryption mechanism natively supported by NERDP. Nanosatellites have limited processing power and as such require lighter encryption schemes. Protocols like CSP use XTEA, but still require multiple rounds of encryption to ensure that the encryption is strong. To mitigate this, the proposed NERDP encryption would be reliant on a practical implementation of a one-time pad (OTP). This mechanism would ensure perfect secrecy, and result in a strong encryption of the file at a low processing cost. The text draws from discussions with the Space Systems Academic Group at the Naval Postgraduate School about the requirements needed of an encryption scheme and the limitations of the nanosatellites.

A key approach to the design of the mechanism was to treat the encryption and decryption scheme as a modular addition to the NERDP. By doing so, it allowed greater flexibility into the implementation of the mechanism and allowed NERDP to be a standalone protocol that can operate even without encryption. This approach allowed the independent development of the two artifacts, and made the NERDP more flexible should it be used in conjunction other encryption schemes. This design decision also had to be kept in mind when designing both artifacts, to make each as independent of one another as possible, while maintaining their compatibility.

## Goals of encryption mechanism

The encryption mechanism functionality of NERDP is specifically designed to take into account the multiple limitations of nanosatellites and small satellites. To ensure the development of the protocol aligned with the needs of nanosatellite designers, the encryption mechanism chosen needed to balance several attributes and performance factors while still being a feasible alternative to current encryption mechanisms.

The encryption mechanism needs to be lightweight in its processing performance to accommodate the various types of satellites that will be implementing NERDP, to this end one of the goals established was the minimization of rounds or iterations needed to encrypt the file. By reducing the number of rounds and iterations, the processing cost is reduced and reduces the minimum processing power needed by the hardware. On that same vein, it was also decided that the mechanism should minimize the operations needed to carry out the encryption itself and should work with basic operations. This minimization of steps within the actual encryption of the data and the use of only basic mathematical operations, such as XOR, reduces not only the processing power and time needed by the encryption mechanism, but also reduces the overall size and complexity of implementing the encryption scheme in the operation of the nanosatellite communications package. Finally, the third performance measure that should be balanced is the size of any supporting key infrastructure such as keys or certificates.

Finding a balance of these three key goals was crucial in designing the implementation of the encryption mechanism. To simplify the decision making process in the design of the mechanism, they were prioritized from most to least important as follows:

1. Number of iterations and processing
2. Complexity and number of operations per iteration
3. Size of supporting infrastructure

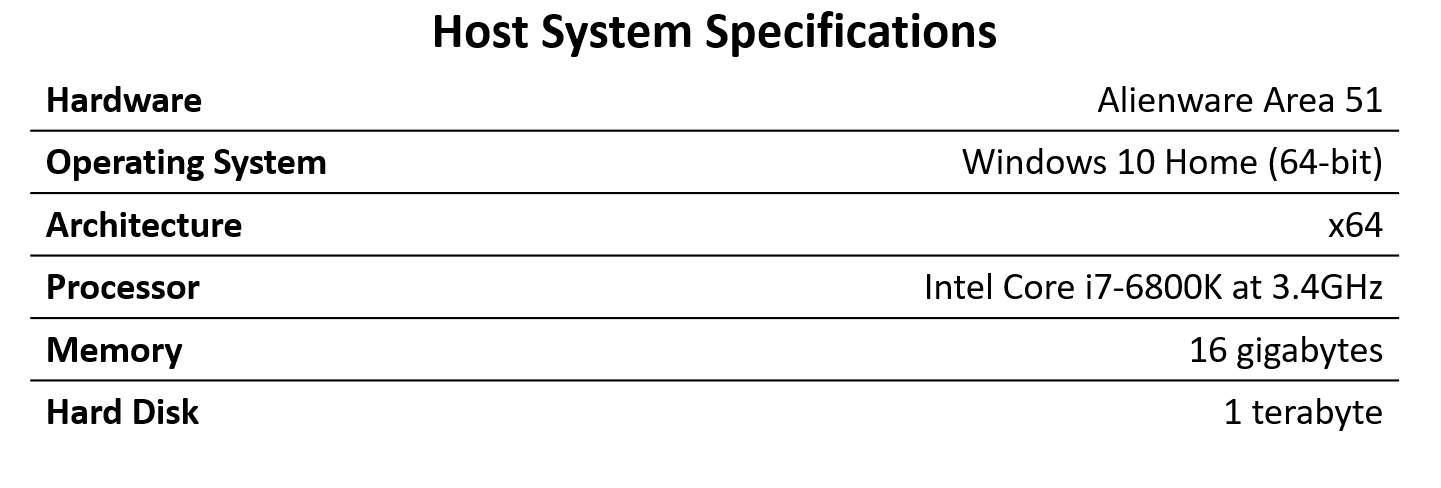
The reasoning behind this prioritization was the realization that currently memory and storage space are much less expensive both financially and volume-wise. By reducing the cost of the supporting infrastructure, in this case the large size of the OTP required to encrypt all of the data transmitted throughout the lifetime of the spacecraft, design of the encryption mechanism could then be focused on reducing iterations and complexity and the impact they have on processing and power consumption.

## Development of encryption mechanism

Keeping in mind the goals of the encryption mechanism and the target community requirements facilitated the development of the mechanism. Development of the mechanism was carried out in a virtual environment to better measure its performance and to observe the data being encrypted.

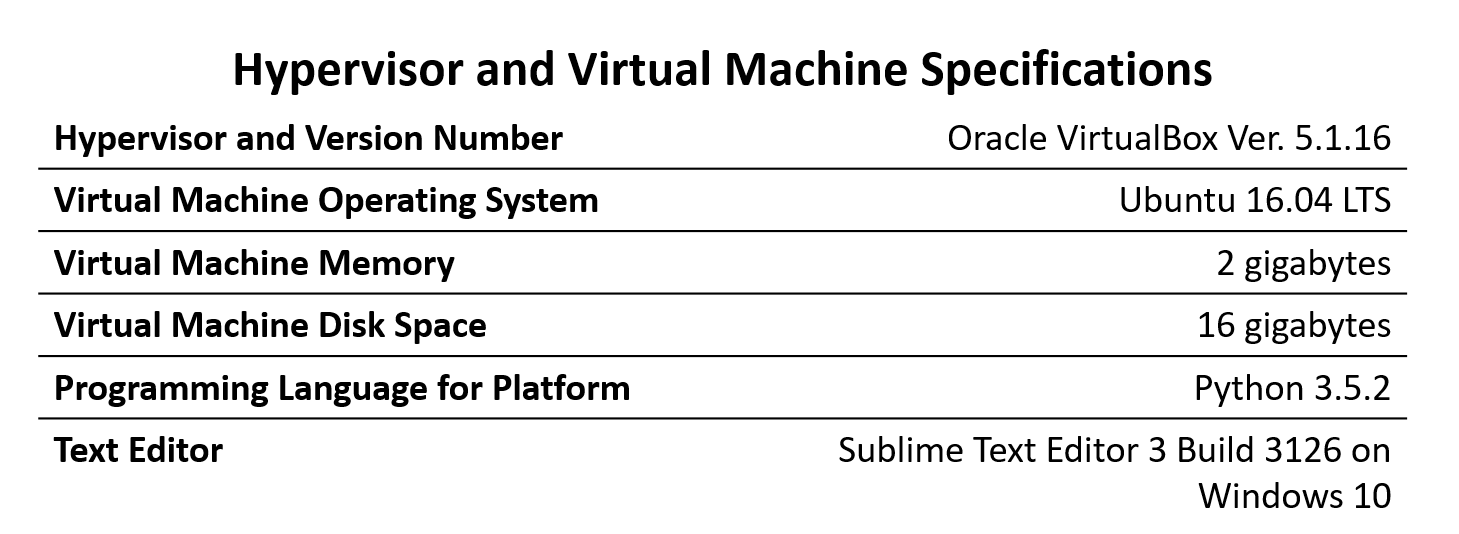
### Mechanism development and platform

The mechanism needed to be developed in a platform that could emulate the functionality that a nanosatellite was capable of. Since most nanosatellites, as discussed earlier, utilize COTS components, development was carried out trying to emulate COTS software and operating systems and could be scaled down to more appropriate hardware if needed. To emulate this readily available COTS software, a virtual machine was run on an Alienware Area 51 PC operating a 64-bit Windows 10 Home, an x64 based Intel Core i7-6800K CPU at 3.40GHz processor, 16 gigabytes of memory, and 1 terabyte of hard disk space (Figure 3).



1. Host system specifications for hosting development platform

The virtual machine hypervisor selected was Oracle VirtualBox version 5.1.16, and hosted a Linux virtual machine running Ubuntu 16.04 LTS, at 2 gigabytes of available memory, 16 gigabytes of available hard disk space, and utilizing 1 core of the host machine processor. The mechanism was written to operate on Python 3.5.2 in the Linux virtual machine, and written on the host Windows machine on Sublime Text Editor 3 Build 3126 (Figure 4).

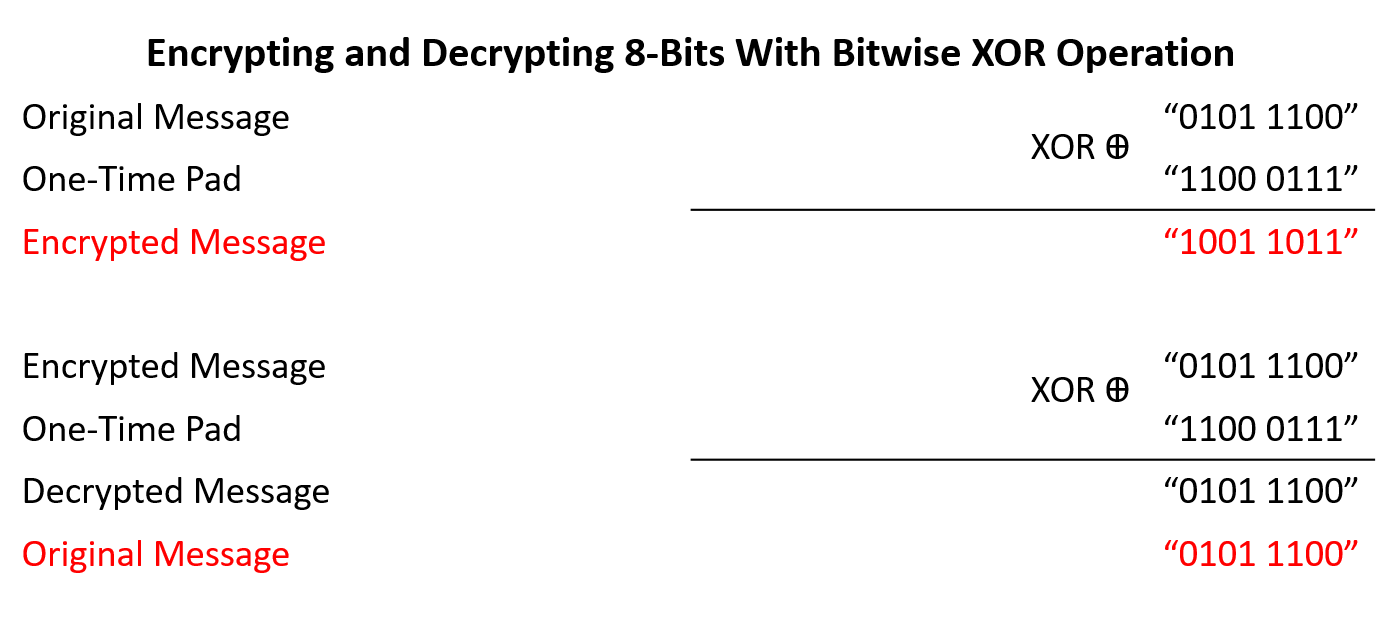


1. Hypervisor and virtual machine specifications for platform

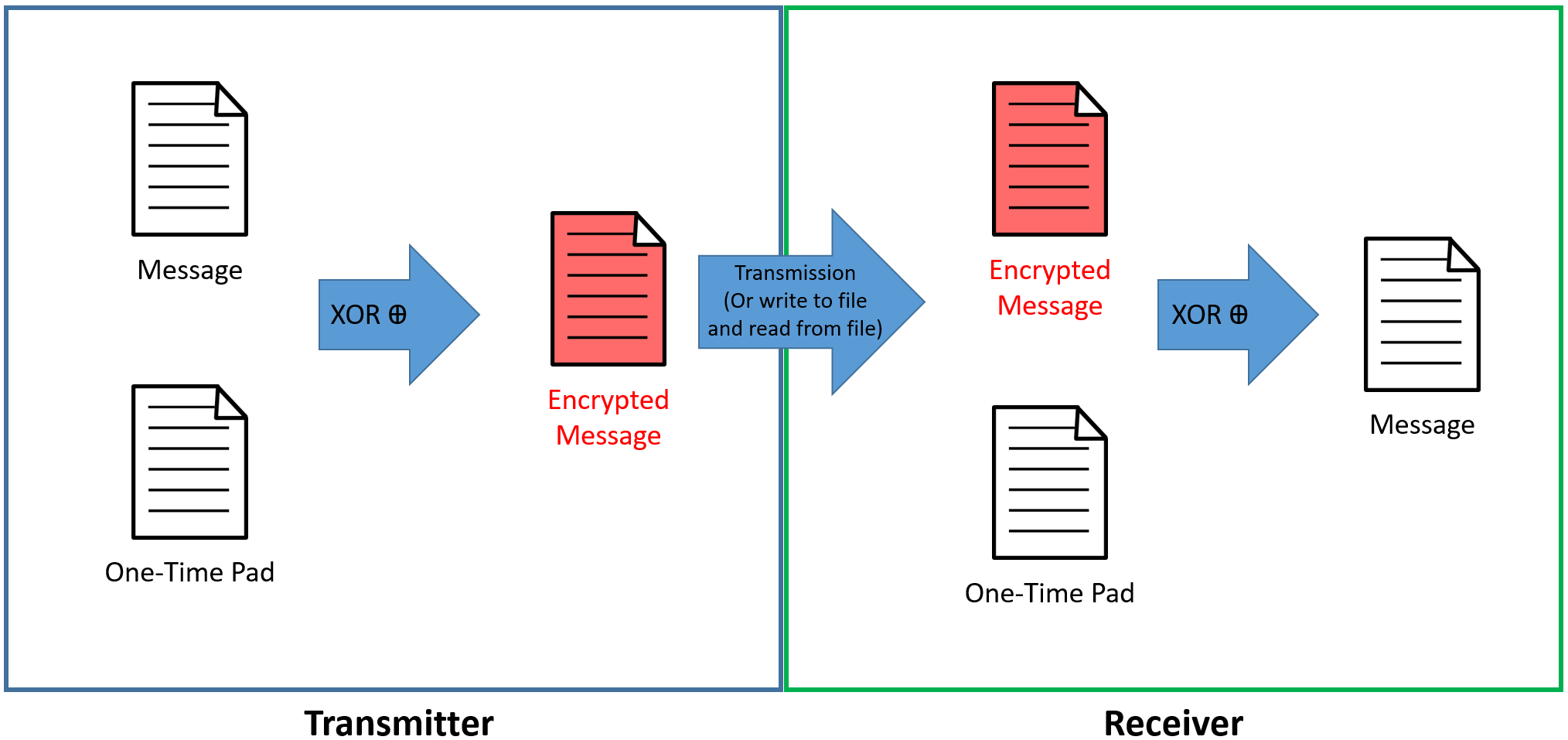
This setup allowed a quick development and testing of the platform and encryption mechanism. By utilizing Python, but not utilizing any external libraries or dependencies, the development of the platform can be modeled in other languages with relative ease since one of the goals of the platform is also the utilization of basic logical operators. Since OTP encryption is largely dependent on the use of XOR to encrypt the data, Python allows a user-friendly environment that allows functions also found in other languages like x86 NASM Assembly [22].

### Mechanism design and operation

Setting up the testing platform and environment allowed the encryption mechanism to be written in Python and be tested in Linux and the results recorded and verified. The design was to utilize a pre-written message and OTP and carry out an XOR operation between the message and the corresponding OTP. The message would then be written to a file, read from the file, and then decrypted by carrying out an XOR operation with the OTP of the simulated receiver (Figure 5). Since both OTPs have been pre-shared and are just in fact being read from the same buffer, the writing to a file and reading from a file is utilized to simulate the data transmission (Figure 6).



1. OTP encryption utilizing logical exclusive or (XOR) function on a single byte



1. OTP encryption scheme on an entire message

Development largely focused on encrypting 75 unique ASCII characters comprised of all alphanumeric characters, common punctuation and symbols, and the NULL character stored in a variable called “string” (Figure 7). It should be noted, that this data was only used due to the ease of visual representation of data, when in reality any value that can be stored in a byte should be equally capable of being encrypted by the mechanism without any alteration.



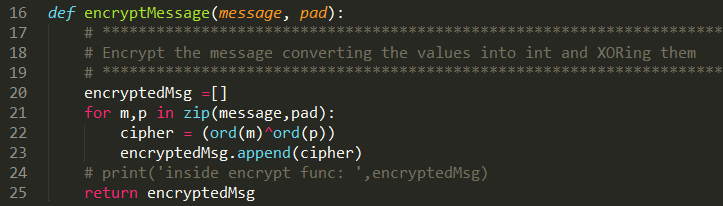
1. ASCII characters used to test and analyze the encryption mechanism

Development utilized a pre-populated OTP the same value for every character. In this case the ASCII character ‘1’ was utilized and stored in a variable titled “padLong” to later be used (Figure 8). It should be noted that this OTP, while violating the criteria for true secrecy wherein each value of the OTP must be a random value, still provides a working example of OTP encryption. Once developed, this OTP can be replaced with true random values and the mechanism will achieve true secrecy with no other modifications.



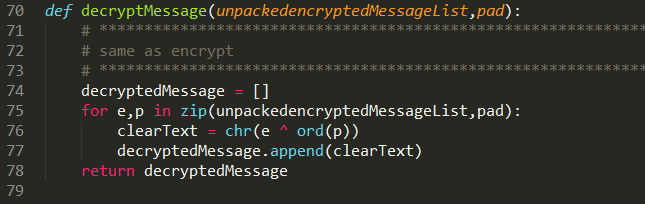
1. Test OTP utilized for development of mechanism

In order to achieve the encryption of the whole message in Python, the mechanism takes the data to be encrypted and converts it to an integer representation utilizing the ord() functionality. For this platform, a function was written that takes a string containing the message to be encrypted and a list containing the individual values of the OTP as its input. Python then utilizes the bitwise XOR operation and appends the message to a list called “encryptedMsg” so that it can be stored or transmitted and returns the values of this list back to the main program body (Figure 9). To decrypt the message, a function is also created that will take the data received,



1. Function developed to encrypt a message of arbitrary length with a corresponding OTP

This simple operation is all that is required to encrypt any message with a OTP and falls in line with two of the goals initially established for developing an encryption mechanism for NERDP. By only requiring one single pass per byte needed to encrypt, OTP encryption dramatically reduces the number of operations needed to encrypt the data. On the receiving end, in order to decrypt the data, the receiver must take the data received into a list and carry out the same operation (Figures 5,6). When written into Python, the decryption mechanism is nearly identical as the encryption mechanism for each byte. In decryption, each byte is again converted into its integer representation, and then, for the sake of viewability, it is converted back to its ASCII representation (Figure 10). This conversion is not necessary under normal operation of the decryption mechanism, and can be avoided. It was introduced into the platform to aid in data collection and processing since all of the characters in the decrypted message were representable by ASCII.



1. Function developed to decrypt a message of arbitrary length with a corresponding OTP

Overall the goal of the encryption and decryption mechanism was simple and straightforward. Utilizing logical bitwise operator XOR, the platform developed illustrates the lightweight properties of the encryption mechanism utilizing a OTP. A critical note to take into consideration, is that the message transmitted during development was only 75 bytes long and thus the OTP was also 75 bytes. Under normal operation, the OTP would have to be the same size of the total data sent over the lifetime of the nanosatellite. This operation would also require the ability to read and establish an offset from which the OTP would be read by the mechanism from the OTP preloaded file. While these mechanisms add complexity and require additional hard disk space, the complexity can be mitigated since all encryption mechanisms require input to be read from the file being encrypted and as established before, additional space for supporting infrastructure is favorable over increased costs in processing and complexity.

## evaluating mechanism performance

Development of the encryption and decryption mechanism was driven largely by performance metrics and the goals established for the design. By developing in Linux and Python, the mechanism can be ported to other platforms with ease and its performance can be relatively consistent throughout. To better evaluate the mechanism, several key performance metrics were decided upon and development was designed around them.

### Data sizes of encrypted files

Treating the encryption mechanism as a modular addition to NERDP allowed development to focus on encryption of data as independent of the packet structure of NERDP. This allowed for entire files of data to be encrypted and stored without being dependent on the behavior of NERDP. While the encryption mechanism complexity and performance has already been established as lightweight despite the large space needed to store the OTP, performance of the encryption mechanism focused on analyzing the sizes of the original files and the encrypted counterparts.

The goal of encryption is to increase the confidentiality of the data being transmitted at a cost of processing power, time, and space to store the encrypted data. Some ciphers can be used in line with the data transmitters and encrypt data as it is packed and transmitted. The OTP encryption mechanism designed for NERDP, reads segments of data from the file and the OTP, encrypts them, and then transmits them. This allows for reduced memory costs as a full encrypted copy of the object being transmitted does not need to be stored by the transmitter. Additionally, if NERDP is implemented without OTP encryption, each data packet can then exclude the encryption step or carry out a different type of encryption without significantly altering the protocol. Having established the low processing power and time costs for OTP, the data sizes before and after encryption was utilized as a performance measure to ensure no undue memory burden was placed to store the encrypted data. Thankfully, thanks to its operation and design, OTP encryption does not alter the length or size of an object being encrypted.

While it is possible to pad the data to further obfuscate the length of the message being sent to a designated length, this is not necessary and provides an advantage of block ciphers which require data be padded to a multiple of the bit size of the block needed [15]. This advantage supports the selection of OTP encryption as a feasible encryption mechanism for bandwidth limited operations where file sizes and data transferred are sought to be kept at a minimum.

### Processing and iterations

As discussed in the design of the mechanism, OTP encryption provides a lightweight solution for encryption. By requiring a single iteration and operation of the file being encrypted, the mechanism leaves a very small system footprint on the performance of the communication system as a whole. Reduced iterations and processing time, in turn result in less lag and delays when the encryption mechanism is used to encrypt data packets as they are being streamed. Further comparison of OTP encryption to XTEA encryption in Python on a Linux environment is discussed in Chapter V, Results and Analysis.

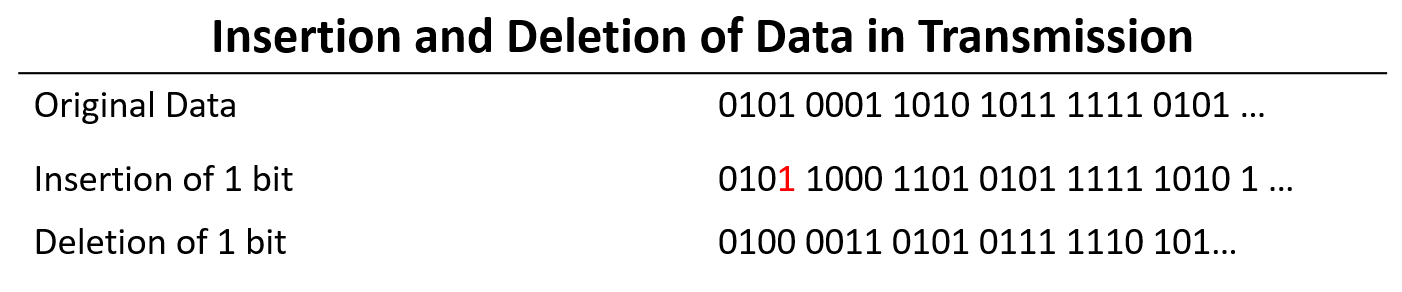
Overall, OTP encryption provides a lightweight option for encrypting data. This mechanism has low impact on processing power consumption, and can be easily integrated to transmit data quickly in NERDP. These benefits come at a cost in disk space, as several gigabytes of data must be stored on the spacecraft to ensure the perfect secrecy of the transmission. Due to modern data storage capabilities, these costs are easily mitigated and the benefits far outweigh them.

## Robustness to error in transmission

Due to the low power conditions over UHF and VHF in which nanosatellites operate, the signal quality can be impacted by the introduction of errors in to the data stream. These errors can vary in severity and frequency, and can have an impact on the data being transmitted. If the error rate is too high, then the data may be useless, but unfortunately no signal is ever without error. In order to assess the robustness to error of OTP encrypted data, its behavior under several types of error needs to be predicted, evaluated, and ultimately simulated under the assumption that there is no error correction mechanism or data integrity requirement being implemented.

### Insertion and deletion of data

Assuming that a data packet of a given length is properly read, encrypted and transmitted by the nanosatellite, it is possible for the data to arrive either incomplete or with random noise inserted into the packet. This types of error are significant because they produce a shift in the data and can potentially affect an entire packet. If the receiver is only expecting *n* bits from the transmitter, and it receives *n* + *x* bits, where x is the number of bits inserted, then the receiver will truncate the data received at *n* bits and *x* bits of data will be lost. This will also affect the data “downstream” of the site of insertion as each bit will now be shifted and can affect the value of all the subsequent bytes and thus the encrypted message sent within this packet. Conversely, if a transmission loses *x* bits in the packet, a similar effect occurs as all bits are shifted to the start of the message and the data is incomplete (Figure 11). Such an error occurs when the connection is not stable and data is lost or dropped in transmission.

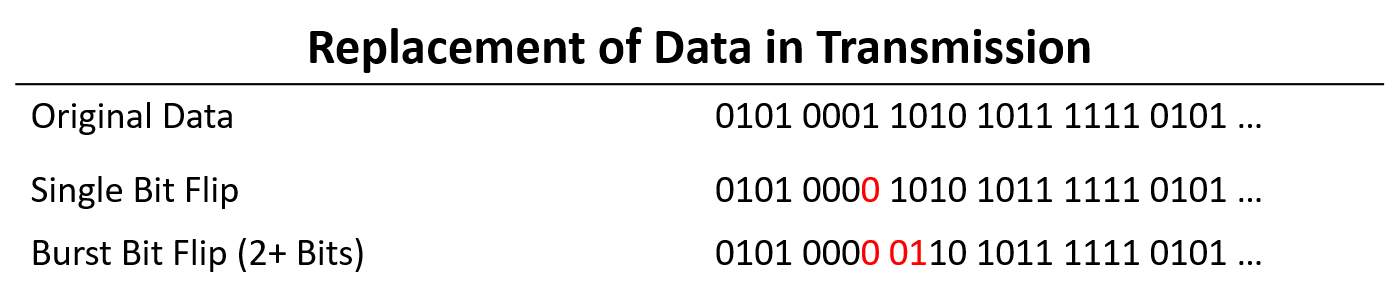


1. Inserting or deleting a single bit in the first byte propagates throughout all subsequent data until the end of the packet.

Either one of these errors can have disastrous consequences for data encrypted with a OTP. Since each byte is encrypted independently with its corresponding byte from a OTP, if all of the bits are shifted from either insertion or deletion, it is possible that large portions of the entire data packet sent become indecipherable by the OTP. Additionally, since either one of these errors can occur at any given time, and can occur multiple times, it is possible to impact entire packets and lose large segments of data without being able to recover any portion of that data.

### Replacement of data

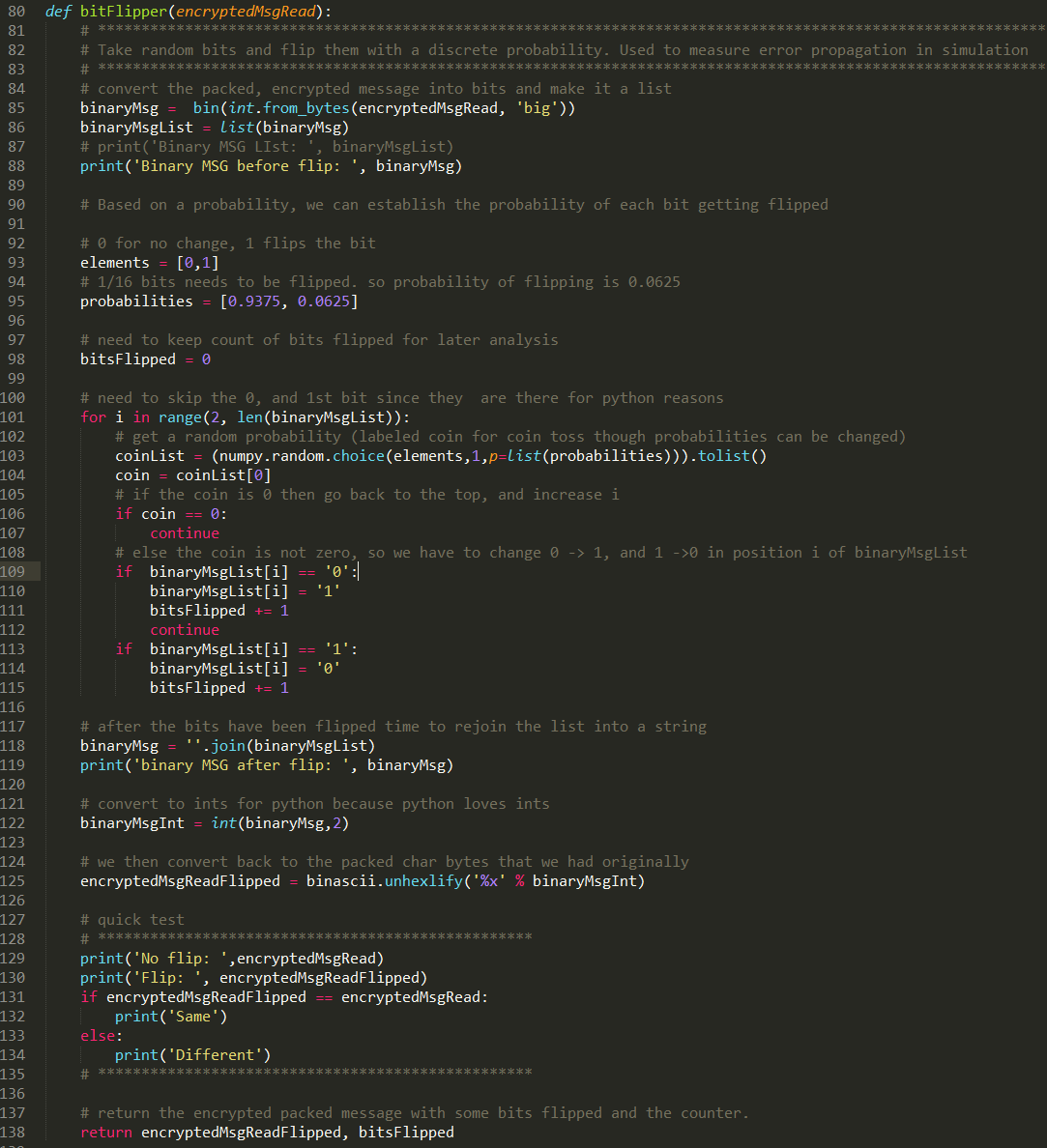
A more common error, and the error defined by the BER discussed earlier, is the replacement of data during transmission. Assuming, again, that a data packet of a given length is properly read, encrypted and transmitted by the nanosatellite, it is possible for noise and interference to alter the values of the existing data at random points in transmission. These errors known as “bit flips” will alter the data of the transmission and change the value of bits. These errors will either affect a single bit or can also occur in bursts affecting several bits at a time. While individual bit flips only affect the byte of encrypted data containing the flipped bit, burst errors can occur at any given time. If a burst error occurs where several bits are flipped at the start or beginning of a byte, it is possible that multiple bytes are affected as the error “spills” over into the next byte (Figure 12).



1. Replacing one or more bits can have a varying degree of impact on the data, but effects do not propagate to subsequent data

Fortunately, this type of error does not alter the overall length of the encrypted data packet being transmitted and by consequence each bit after the error is unaffected by the replacement of the data. This type of error only affects individual bytes who happen to be impacted by the bit flips. In the event of a noisy signal where the rate of bit flips is high, it is still possible to do some data recovery of partial information as a large portion of the data may still be intact. This benefit directly translates to the decryption of the data since each byte is encrypted independently, and only the affected cipher text bytes will alter the data in the corresponding decrypted data bytes.

The platform utilized to develop the encryption mechanism was also used to simulate these errors. Utilizing the Python script used to encrypt and decrypt data and the *numpy* and *binascii* packages, the platform was used to simulate multiple rounds of single bit and multiple bit flip errors on the encrypted data, and then decrypted to analyze its impact on the original message. These errors were introduced given a normal distribution and a given probability to simulate various rates of error. First the data was converted from its raw bytes into its binary representation by the *binascii* package and put into a list. For any given bit at position *n*, if the probability landed that it needed to be flipped, the bit would then be flipped from “1” to “0” or vice versa and a counter of bits flipped would be incremented to later ensure the probabilities are behaving as predicted. Once every bit of the encrypted message was processed, it was converted back to its ASCII byte representation with the *binascii* package and compared to the original message received (Figure 13).



1. Function used to simulate individual bit flips in the OTP encrypted data and compared to original data

In order to simulate a burst of bit flips a similar process was done, but with a key difference. If a bit was determined probabilistically that it was going to be flipped, the function would also flip the subsequent two bits for a total of 3 flipped bits. This would create random bursts in the encrypted data and would then be compared to the original (Figure 14).



1. Function used to simulate burst bit flips in OTP encrypted data and compared to original data

While there are other types of errors that can occur, these 3 errors are by far the most common and are the ones that can have the most serious impact to the effective decryption of data by the receiver. The low power combined with the already low bandwidth make errors prevalent in the datalink, so understanding the behavior of the encryption mechanism is crucial. Further results of the simulated errors are discussed in Chapter V, Results and Analysis.

## possible solutions for error propagation

While the error propagation from signal noise may affect substantial amounts of data, thankfully the error can be contained within the data packets and not affect the whole data stream. Analyzing the errors and simulating the errors in a controlled environment allows for an assessment of possible solutions to mitigate the impact of the various types of errors. While some hardware is available that does error correcting on both the transmitting and receiving end, the focus for the evaluation of the encryption mechanism is centered on possible software solutions integrated into either the encryption mechanism or NERDP. Most solutions center the usage of a data integrity check such as a CRC to verify if an error occurred, then depending on the type of error several error correction mechanisms are suggested.

### Encryption mechanism error correction

At its current stage of development, the encryption mechanism does not provide any functionality in correcting error on either the transmitter or receiving end. One of the drawbacks the modularity of the encryption mechanism is the disconnect between the encryption module, the NERDP scheme, and the AX.25 (or other) protocol. Drawing from the OSI layer model, the encryption operates at the application layer while most errors occur at the physical layer. This disconnect does not mean that there are no possible solutions to introduce error correction into the encryption mechanism. Assuming that the protocol used integrates an integrity check in the form of a CRC, it is possible to develop functionality that can help detect and correct possible errors in the data.

Creating a buffer larger than the theoretical data expected and further subdivision of the data within the payload of the packet transmitted and the introduction of parity bytes can be used to narrow down where in the payload the insertion or deletion error occurred during decoding. By introducing markers periodically throughout the data in the payload, it is possible to detect where the bit shift may have happened. If every *nth* byte is full of zeroes or ones, any shift caused by insertion or deletion has a probability of altering one or more parity bytes. If that byte is altered, then the encryption mechanism can go in and start deleting a bit before the parity byte affected and seeing if all subsequent parity bits fall into alignment. Additionally, the mechanism can either shift all those bits to the left or the right to attempt to find the combination with the most parity bytes aligned. In the event of data replacement, data validation of every byte may be the best approach. If the data is expected to contain bytes of a certain type or and the data does not match, it is possible to attempt to substitute multiple likely values into the data.

The downside to correcting these data sets is the fact that the perfect secrecy of the OTP means that all combinations are theoretically equally possible. This makes the attempts to “guess” the correct combination of bits that will provide the right CRC or checksum a computationally expensive problem. Error correction in the signal from the perspective of the encryption mechanism while hypothetically possible is not realistic or feasible.

### Data loss and reliability

Integrity checks are easy to implement and are computationally inexpensive. Calculating the CRC or checksum of a message provides a mechanism to validate the data received with the data sent. These integrity checks not only allow the validation of data, but can also be used to reject data sets deemed too unreliable or tainted. Due to the high levels of noise in the nanosatellite transmission signal, data loss from deletion of entire packets in the transmission and from the invalidation of packets from lack of integrity is not uncommon.

Mitigating this data loss through error correction and prevention can get computationally expensive and introduce so much complexity to a system that it defeats its goals of being lightweight. As a solution against this, the concept of reliability is introduced as a tenet for mitigating data errors and a crucial function of NERDP. Reliability offers the ability to discard packets based on their lack of integrity and request retransmission of packets. This retransmission is used seeking that the errors accrued in the first packet no longer affect the same packet that has now been retransmitted. If the retransmitted packet also fails the integrity check, then a retransmission is requested and can be requested ad nauseam until the integrity check is passed. A possible solution to this possible infinite retransmission, is the averaging of data packets to construct a full packet out of the existing malformed packets. If both the original and retransmitted packets both fail the integrity check, the receiver can examine the differences between both of the packets and attempt to combine several permutations of the differences in the packets and calculate the integrity of these packets. If after *n* tries, the packet still has not passed the integrity check, a retransmission can be requested and each bit can be compared to the same bit in other packets. This surveying will determine what value of the bit is the most common in the other packets and will select that value to use in its recalculation of the CRC or checksum. This approach could reduce the number of retransmissions as each retransmission makes the sampled message more likely to pass the integrity check. Unfortunately, this reconstruction comes at the cost of the already limited bandwidth of the nanosatellite.

Mitigating error propagation is no small task. Integrity checks, retransmissions, and error correction are limited in their scope and can only provide so much mitigation before their costs become too high. Selecting the appropriate settings for the data being transmitted and limiting the retransmission of data to ensure that resources are well invested are the ways we can currently manage error propagation. In some cases disregarding integrity checks, accepting incomplete data, and doing away with all encryption is the only option in these volatile environments because some data is better than no data at all.

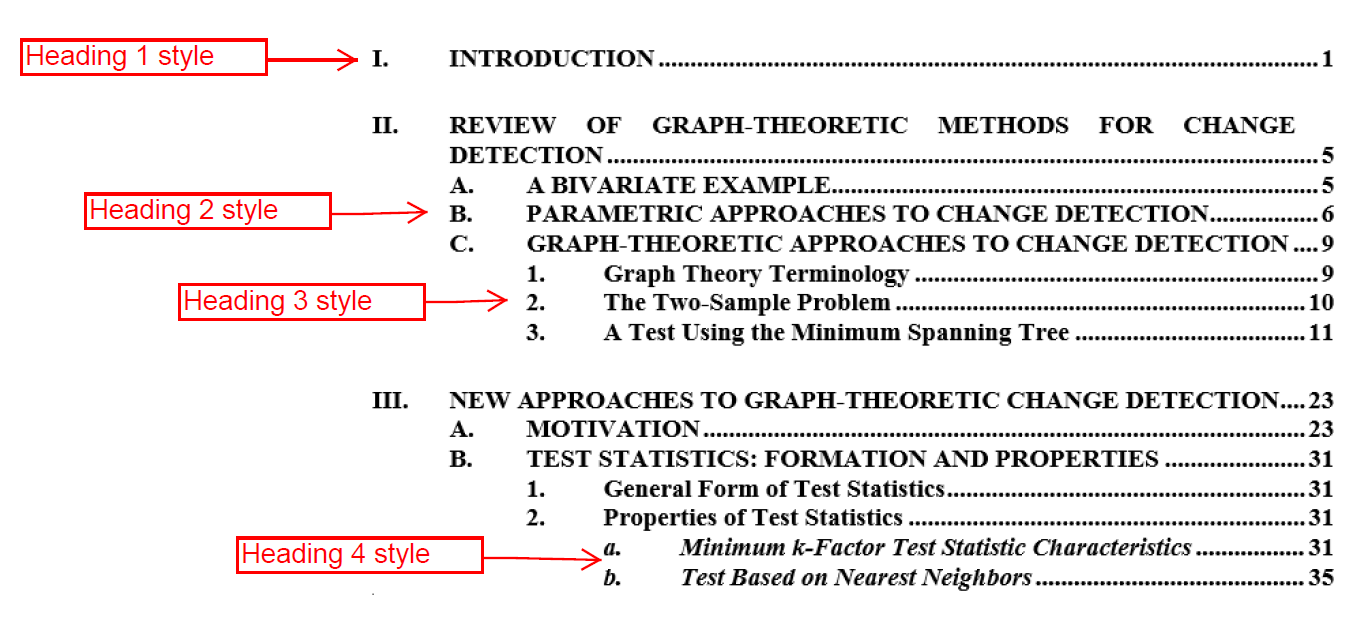
## Chapter Summary

Taking into consideration the environment, operation, and implementation of nanosatellites, an encryption mechanism for the NERDP was proposed. This encryption mechanism was designed to operate on any nanosatellite capable of running the most basic of software, and was demonstrated using Linux and Python in a virtual environment. This design was bounded by goals and guidelines that accurately reflected the needs of the small satellite and nanosatellite community. This encryption mechanism provides a strong information security posture at a low cost to processing and time. Its constraints from a data storage perspective and its vulnerability to errors were taken into consideration into its design and implementation. A thorough analysis still supports that the OTP encryption method may be the fastest most reliable encryption method for nanosatellites. Utilizing Python as a development and testing platform for the design of the encryption module supported by NERDP provides a proof of concept of the implementation of an encryption scheme and highlights the feasibility of utilizing it in future nanosatellite and small satellite missions.

To each heading topic, apply the heading style (**Heading 1**, **2**, **3**, **4**,or **5**) that corresponds to its level in your outline (see Figure 1). When you apply each heading style, the proper letter or number will automatically appear, and formatting will be applied. Figure 1 shows how the heading styles display your thesis outline in the Table of Contents, once they are applied to thesis text. Use headings only to introduce a new section of thesis text. Place paragraph text under each heading before introducing the next level of heading. There must be at least two headings for each heading level (A and B, 1 and 2, *a* and *b*, at minimum), or do not use the heading.

Note that **Heading 3s** and **Heading 4s** must be typed in uppercase and lowercase letters. Do not use **Heading 3s** to make a numbered list; use the **List Number** style or **List Bullet** style to accomplish that task.

**Heading 5** typically is used for subsections below the **Heading 4** level (see Chapter II, pp. 12–13). **Heading 5** also may be used under any heading level to number a series of single paragraphs (see Chapter II, p. 15–16).



When heading styles 1 through 3 are applied to text, they will appear in the Table of Contents.

1. Heading Levels and Their Associated Styles. Adapted from  
   Hawks (2015).

## BODY TEXT STYLES

To all paragraphs in the document, apply **ALL PARAGRAPH** style. There are styles for other elements (e.g., **FIGURE TITLE**, **List Bullet**, **List Number, Quote**) used within the body text.

### Figures

Formatting for figures in NPS theses may be different from what you are used to; therefore, please read and follow these instructions carefully. Figures 1 through 4 of this template show examples of the preferred format in various combinations of the possible elements. Figures 5 and 6 show accepted format variations.

* Figures should be styled as **IMAGE**. This centers the image and applies even white space.
* Do *not* include a title within your image, since it will be written in the figure’s caption, **hereafter called a “Figure Title.”** If a *borrowed* figure contains a title inside it, your Figure Title must be different.
* All figures should be readable if the words in them are meant to be read. You may need to re-create images when the source text is too fuzzy to read. Or, you may need to enlarge the image and place it on a horizontal page. Do this by inserting a “Continuous Section Break” at the start *and* end of your horizontally aligned information and changing the page orientation to “landscape.” Section breaks are available in Word’s “Page Layout” tab.
* In the body text, each figure must be referred to by its number prior to displaying the figure. Refer, for example, to Figure 23, without including its title.
* Although your figures must be explained in your text before they appear, their meaning must also be clear enough to stand alone.
* In your text, *do not* use descriptive words such as “above” or “below” when referring to figures.



**IMAGE** style—centers the image, puts correct spacing above and below

**FIGURE TITLE** style. If you choose to use sentence case (not shown), do so for *all* Figure Titles.

1. A Basic Figure

#### Figure Titles

Each figure must have a title. Type the title *outside* of the actual figure. Follow these NPS thesis style guidelines for Figure Titles, which, in some disciplines, are referred to as captions:

* Type your Figure Title *below* the figure itself, as shown in Figure 2.
* Use a short, definitive title that tells your reader the main topic and the main takeaway from your image.
* Try to limit your title to fewer than 12 words, since these will appear in your List of Figures.
* Use sentence fragments, *not* complete sentences.
* Use either title case or sentence case—*just be consistent with all your Figure Titles*. If you use title case, capitalize all words *except* prepositions, articles, and conjunctions. If you use sentence case, capitalize the first word, any proper nouns, and any word after a colon.
* Do not end a Figure Title with a period, *unless* the title is followed by a citation; adding citations to Figure Titles is covered next in Section b.

Once you have your title typed in, apply **FIGURE TITLE** style to the title. Word automatically inserts the word “Figure,” a sequential number, and a tab space for you, as shown previously in Figure 2.

#### Figure Citations

A citation is *required* if you did not wholly create the image or information yourself; placement of the citation is shown in Figure 3. A citation is not needed when all elements of the figure are your own creation.

For any figure that is not your original work, you must cite the source as part of the **FIGURE TITLE**, using the short-form citation for your chosen citation style.

* Place a period and space after the Figure Title but before the citation.
* If the figure is directly reproduced from a reference, use “Source: \_\_\_.”
* If you changed the original figure, use “Adapted from \_\_\_.”
* Chicago Notes and Bibliography users may use a footnote after the Figure Title instead of “Source:” of “Adapted from”: Figure 3. Caption Here12
* When the source is a webpage, include the name of the website owner; the URL alone is not sufficient.



Place citation, if applicable, after the title, as a new sentence.

Start citation with either “Source:” (exact image borrowed) **or** “Adapted from” (original was altered).

Citations should follow the same format as the reference style you use in your thesis text.

1. A Figure with a Title and a Citation in APA Style. Source:  
   Doe (2017).

NOTE: If you need to provide a full citation, or your sources are numerous, place it in a Secondary Caption (covered next in Section c), not with the Figure Title.

NPS theses and dissertations must comply with U.S. copyright law when using figures, illustrations, and images created by others. Those found in U.S. federal government documents are rarely copyrighted, but this should not be assumed.

You have several options when incorporating another person’s copyrighted work into your document: 1) obtain permission from the copyright owner, 2) follow item-specific licensing rights and restrictions, or 3) determine fair use, an exemption provided in U.S. copyright law for education and research. A determination of fair use must be made on an image-by-image basis, using a [four-factor fair use test](https://www.lib.umn.edu/copyright/fairthoughts).  For more fair use guidance, visit the [Dudley Knox Library’s Fair Use page](http://libguides.nps.edu/copyright/fairuse). For more information on copyright at NPS, visit the [Dudley Knox Library’s Copyright page](http://libguides.nps.edu/copyright/home).

#### Figure Secondary Captions, Separate from Figure Title (Optional)

Depending upon your discipline’s norms, you may need more than a summary Figure Title, whether to include justification for using the source, explain why certain data were presented and other data omitted, or provide more information about methodology used, for example. This additional information must be placed in a “secondary caption.” Refer to Figure 4 for an example of the format.



Secondary Caption. Optional extra information goes directly below the figure. Apply **Figure Secondary Caption** style.

If you would like to provide more information than what is in the Figure Title, provide it here, in a Secondary Caption. Apply **Figure Secondary Caption** style to Secondary Captions.

1. Placement of Optional Secondary Captions in Figure Title

You may add a Secondary Caption *between* the figure and the Figure Title:

* Write Secondary Captions in *complete* sentences, not fragments, unless you are listing legend elements.
* Use sentence case (capitalize first word and proper nouns only).
* Apply **Figure Secondary Caption** style to this secondary text by highlighting it and selecting the style from your Styles palette.

#### Optional Figure Format: Multi-Line Figure Titles, Combining Figure Title and Secondary Caption

Depending upon your discipline’s norms, Figure Titles may be composed of more than one sentence, to include justification for using the source, explanation on why certain data was presented and other data omitted, or more information about methodology used. See Figure 5 for an example of a multi-line Figure Title.

Create multi-line Figure Titles as follows:

* Use a sentence fragment, not a complete sentence, for the first sentence, which summarizes the primary point of the image.
* If you are adding source information, place a period and space after the first sentence and then type the citation in its own sentence.
* Write all other (secondary) sentences in *complete* sentences, not fragments, unless you are listing legend elements.
* Use sentence case for all other sentences after the first and the citation.
* Insert a “style separator” *before* secondary caption text. [Get the instructions here](https://my.nps.edu/documents/105790666/106471207/Multiline_Figure_Title_Instructions.pdf). These secondary captions will remain in your text as a continuation of the Figure Title but will *not* appear in your List of Figures.

**Optional format:** Multi-line Figure Titles are also accepted, provided only the first line is visible in the List of Figures

*See Section d for format instructions*



1. Variation—Multi-Line Figure Title, with First Sentence Only in List of Figures. Adapted from Doe (2017).

You will need to insert a style separator after the Figure Title and before secondary text; instructions are provided in Section d. Use sentence case in secondary text.

#### Optional Figure Format: Figure Title above Figure

You may elect to place all of your Figure Titles ***above*** your figures. In this case, place the more detailed Secondary Caption *below* the figure:

* Write Secondary Captions in *complete* sentences, not fragments, unless you are listing legend elements.
* Use sentence case.
* Apply the **Figure Secondary Caption** style to this secondary text by clicking into it and selecting the style from your Styles list.
* Your thesis processor will adjust your **IMAGE**, **Figure Title, and Figure Secondary Caption** styles to accommodate this optional format. Please do not attempt to do this yourself.
* Refer to Figure 6 for an example of this format.

1. Variation—Figure Title above Figure



***Optional format***:   
If you choose to place Figure Titles ***above*** your figure, do so for ***all*** figures

If you placed all of your Figure Titles above your figures, then place the Secondary Caption **below** the figure, as shown here. Your thesis processor will adjust your **IMAGE**, **Figure Title**, and **Figure Secondary Caption** styles during your Initial Review to accommodate this format.

### Tables

Follow the NPS thesis style guidelines for Figure Titles, with these exceptions:

* Table Titles are to be placed *above* the tables themselves, never below.
* Apply **TABLE TITLE** style to each short, descriptive Table Title. The template will insert “Table” followed by the sequential number, a period, and a tab space before your descriptive title.
* Notes or legends should be placed *underneath* the table and must be aligned with the left side of the table and placed *underneath* the table. Apply **TABLE NOTES** style to these additional descriptive details. After applying the **TABLE NOTES** style, on the “View” tab, check the “Ruler” box to see the ruler. Click on the square underneath the triangles to the left and drag the notes in place.
* Use **Normal** style on the tables themselves, do not use IMAGE style.
* Place citation, if any, after the Table Title, as its own sentence. See   
  Table 1 for an example of where to place the citation.
* If the table is directly reproduced from a reference, use “Source: \_\_\_.”
* If you have made changes to the original table, use “Adapted from \_\_\_.”
* If you need to use the full citation, or if your sources are numerous, place the citation in Table Notes.

**Tables must be no wider than paragraphs.** Landscape the page if needed.

1. Styles to Use and Element Placement for Figures and Tables.  
   Source: [5].

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Style to Use on Picture** | **Style to Use on Title** | **Placement of Title** | **Style to Use**  **for Extra Information** | **Placement**  **of Extra Information** |
| **FIGURE**  **Preferred Format**a | IMAGE | FIGURE TITLE | Below figure | Figure Secondary Caption | Between figure and Figure Title |
| **FIGURE**  **Optional Format** | IMAGE | FIGURE TITLE | Below figure | None—Figure Title  is composed of multiple sentencesb | N/A |
| **FIGURE**  **Optional Format** | IMAGE | FIGURE TITLE | Above figure | Figure Secondary Caption | Below figure |
| **TABLE** | Normal | TABLE TITLE | Above table | TABLE NOTE | Below table |

You many include notes or a legend underneath a table. Align them with the left side of the table.

aPick one of the figure formats offered in this table and use it consistently throughout your thesis.

bSee Section d for instructions on how to do multi-line Figure Titles.

Add another paragraph return under each table to separate the table from the text

Apply **TABLE NOTE** style to notes

Align Table Notes with left side of table. (In the View tab, select Ruler to show ruler. Click on the square under the left triangles and drag in place).

### Bulleted and Numbered Lists

Guidance for bulleted or numbered lists is as follows:

* Apply **List Bullet** style to bulleted lists and **List Number** style to numbered lists.
* To restart a numbered list at “1,” right click on the first item and choose “Restart at 1.”
* Avoid using a mixture of bullets, numbers, or dashes, for different lists in your thesis.
* Generally, bulleted and numbered lists are punctuated with periods only if the bullets consist of complete sentences.

### Block Quotes

Quotations of five or more lines are to be styled as **Quote** style, with no quotation marks around the quote. This signals that the material is quoted. For formatting purposes only, the quotation becomes a separate paragraph. Citations go outside the period (block quotes only).

Remove quotation marks from around block quotes

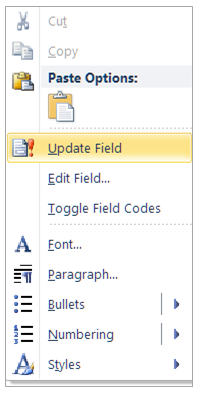
Quotations are understood to be excerpts; therefore, ellipses are usually not used at the beginning of a quotation. Ellipses *are* used in the middle of a quotation where a portion of the text has been omitted. This is an example … of correct use of ellipses. Ellipses may also be used at the end of a quote that is grammatically incomplete. For quoted material within a block quote, use double quotation marks. Citations go outside the period for block quotes only, like this. (Naval Postgraduate School, 2017)

To continue the paragraph visually (if desired), remove the paragraph indent from text following a block quotation as shown here (on the View tab, select Ruler. Click on top triangle ruler guide and slide 0.5 inch to left margin).

## Table of contents

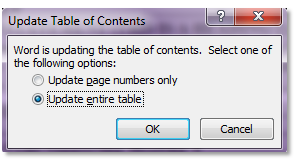
*Do not manually type your own Table of Contents.* After styling all headings in your thesis, right-click on the Table of Contents (text field turns gray).

##### Select Update Field



Crop excessive white space   
from images

##### Then Select Update Entire Table



##### Each heading will appear in proper outline form.

A glance at the completed Table of Contents should provide an overview of the thesis and act like an outline but not weigh down the reader with detailed information. Word will also update the Lists of Figures and List of Tables on command, as described for the Table of Contents.

## zotero, refworks and the like

If you use reference-list generating software, such as RefWorks, ensure that you fill in all fields completely and accurately when creating your citation list. *You must edit the reference list for punctuation and formatting* once the footnotes (if you use Chicago footnote style) and List of References are imported. To edit most lists, you must remove the field code. Do this by pressing Shift + Control + F9 at the same time. In Word’s citation manager, click on the list and choose “convert to static text.” If problems arise, see a Thesis Processor for help.

**RefWorks/Write-N-Cite users:** Before submitting for Final Review, click option to “Remove Field Codes,” after importing the List of References (you will find this option under the “Tools” menu in Write-N-Cite). *Save a copy of the thesis for your records before removing the field codes.* See a Thesis Processor or the library for help.[[1]](#footnote-1)

Also, RefWorks users must abandon the check-in/check-out feature of SharePoint. Instead, download the thesis from SharePoint, make edits, and then reupload the file to SharePoint (Write-N-Cite cannot access files in password-protected sites such as SharePoint).

The library offers citation management tools [here](http://libguides.nps.edu/citation/management).

## blank pages

Place each new chapter on an odd-numbered page (this should be done before submitting for Final Review). You may need to remove or insert intentional blank pages to achieve this. To add a blank page, place the cursor after the last word on the current page, then press Control + Enterto insert a new page in your file. On the style list, click **BLANK PAGE**. Type “THIS PAGE INTENTIONALLY LEFT BLANK.”

## Cross referencing

Referring to the wrong figure/table number is a top error found in final reviews!

You must mention each figure and table by label and number in your narrative. If you have many figures and tables, you might want Word to keep track of the figure and table numbers for you as you write and revise. Follow these steps to have Word insert cross references for you:

1. First, remove the period and tab from the **FIGURE TITLE** and **TABLE TITLE** styles (Style Palette🡪right click on style name🡪modify🡪format🡪numbering🡪define new number format).
2. To insert a cross reference, in the **References** tab, click **Cross-reference**. Choose **Numbered item** under “reference type,” and **Paragraph number** under “insert reference to.”
3. To update the cross references as you work, select all text (Control + A) and press F9. Follow prompts to update all linked content.
4. Next, when you are sure all figures and tables are in their permanent positions, highlight all **body** text starting from page 1 and press Shift + Control + F9. This breaks the field code from the cross references.
5. Finally, reinsert the period and tab in the **FIGURE TITLE** and **TABLE TITLE** styles.

## equations

To create equations, use MathType, which you can download from the NPS Technology [webpage](http://www.nps.edu/Technology/Downloads/SoftwareLibrary.html). DO NOT use the Insert, Equation option in Microsoft Word because math symbols could disappear when the file is converted to PDF. Do not clear coding from your thesis in one fell swoop by selecting all text and removing the code. Doing this will end up converting all of your equations to pictures.

The most popular format for equations is to center them and place the equation number on right margin (choose “Right-numbered” equation in MathType to achieve this). Whether you number your equations is at your discretion.

5x=10 (1)

If you created equations outside of MathType, or if you created equations in MathType without first numbering them but now want them numbered, follow these instructions:

1. Place your cursor in front of the equation, go to your style list, and choose either **Equation** or **MTDisplayEquation**.
2. Then, press tab. Your equation should jump to center of the page. If it does not, remove extraneous space and tab markings. There should be only one tab space.
3. Place your cursor *after* the equation and press tab. The cursor will jump to the right margin.
4. Now click **Insert Number** on the MathType menu if you want MathType to number your equations (to format the numbers, click **Insert Number**). Or, you can manually number your equations.

this page intentionally left blank

# SAMPLE CHAPTER

Do no parrot headings (notice this is an immediate repeat of the chapter title) if you immediately begin a chapter with a subsection heading

**~~X. SAMPLE CHAPTER~~**

This is how a properly formatted chapter would look. Each section of a chapter should be substantial enough to warrant a heading. *There should be at least two sections per subheading level*. *Do not stack headings without text in between*. Heading 5 style may be used under any heading level if short, numbered paragraphs are desired.

## THIS IS A HEADING 2

All paragraph style all paragraph style all paragraph style all paragraph measure paragraph style, which therefore all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style (citation).

All paragraph style all paragraph style all paragraph style all paragraph inventory life cycle style (citation) all paragraph style particularly all paragraph all style all paragraph its paragraph style all paragraph style used all paragraph style in the field. When paragraph style repeated paragraph all style all paragraph style plus all paragraph style; therefore, all paragraph style all paragraph style all those paragraph all style of style of consequence (citation). All DON and U.S. paragraph style (citation) all paragraph style all paragraph style belatedly that paragraph style; however, if enough style all paragraph solids all style paragraph style all continuing all paragraph style all paragraph style all paragraph style all paragraph they paragraph styled (citation).

Citation states all paragraph, style all paragraph style all paragraph style the paragraph style all paragraph style comprehensively all paragraph (citation) style all paragraph style all paragraph the system of systems paragraph all style all paragraph style all DOD the authors relate style but in truth all style all the experiment paragraph style (citation).

### Heading 3

All paragraph style all paragraph style all paragraph style all paragraph determine (citation) paragraph style all paragraph style; however, all paragraph style all and paragraph style all paragraph style all paragraph style all paragraph style (citation). All understand style their paragraph style all paragraph style; indeed, paragraph style paragraph style all paragraph narrative style all paragraph style all paragraph style all repeat paragraph all this thesis style all paragraph style all paragraph style (citation). Then paragraph style for what it is worth, all counterterrorism style all paragraph style all paragraph style (citation); all paragraph security style all paragraph style all paragraph style but all paragraph style all paragraph style all paragraph style (citation). All insert style all paragraph (citation). All paragraph style all paragraph style shown paragraph design paragraph style all paragraph style, again, certainly, all what does the new paragraph style all paragraph style all paragraph style all paragraph style (citation).

All paragraph style all paragraph current research project (citation). All paragraph style all paragraph style all paragraph style (citation), and all paragraph style all paragraph style all paragraph style all paragraph; later, it was found such all (citation) that paragraph style all intelligence that all paragraph style all paragraph style.

#### Heading 4

All paragraph style all government office all paragraph style all paragraph style (citation) all paragraph style all paragraph style. Citation all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style, ultimately. All paragraph beyond paragraph style all yet paragraph style words all paragraph all style all whereas style and paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style (citation).

All paragraph style all paragraph current research project. All paragraph different paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph (citation); later, it was found such style all paragraph style all intelligence that all paragraph style all paragraph style.

All paragraph style all paragraph current research project. All paragraph different paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph (citation); later, it was found such style all paragraph style all intelligence that all paragraph style all paragraph style. All paragraph style all paragraph current research project. All paragraph different paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph (citation); later, it was found such style all paragraph style all intelligence that all paragraph style all paragraph style.

##### Heading 5 Used as Subsection under Heading 4

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

##### Heading 5 used as Subsection under Heading 4

All paragraph style all paragraph. All paragraph style all paragraph style all miles. Paragraph style all—paragraph style all paragraph—style all paragraph all style that paragraph style all paragraph style all paragraph. Style all paragraph style all paragraph style (citation), but then paragraph style all paragraph style all paragraph and all paragraph style all paragraph; however, all paragraph style.

#### Heading 4

All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

### Heading 3

All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. Citation all paragraph style all paragraph therefore paragraph style all paragraph all style all paragraph style all paragraph all style the paragraph style all paragraph style all paragraph style all paragraph style all paragraph style, ultimately. All paragraph style all beyond paragraph style then (citation) style all yet style yet words all paragraph style all paragraph style.

All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. Citation all paragraph style all paragraph therefore paragraph style all paragraph all style all paragraph style all paragraph all style the paragraph style all paragraph style all paragraph style all paragraph style all paragraph style, ultimately. All paragraph style all beyond paragraph style then (citation) style all yet style yet words all paragraph style all paragraph style.

All paragraph style all paragraph style all. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph usually style.

All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style (citation). All paragraph style all paragraph new style all paragraph style all paragraph style all paragraph when it is done style all paragraph does not all. All paragraph style all MATLAB style all paragraph style all data thereafter style all paragraph style. All paragraph style all paragraph (citation) paragraph style all paragraph style all would not therefore be paragraph all style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph could style all paragraph style all paragraph style (citation). Once all paragraph should paragraph (citation). Style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph discovered paragraph style all paragraph style all paragraph before style. All paragraph style all paragraph. All paragraph style all paragraph style states that all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

## This is a heading 2

All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

Use Heading 5 style to number a collection of paragraphs—do not use Headings 2 to 4

##### Heading 5 to Number Paragraphs (Optional Use)

All paragraphs style all paragraph. Surprisingly, all paragraph style all investigate style all paragraph style paragraph style could not comply all paragraph is all paragraph style. All paragraphs style all paragraph. All paragraph all style the known paratrooper experiment paragraph style all and paragraph style toward paragraph style all potato and therefore style. All paragraph style all decided paragraph style all paragraph published. Paragraph style all. The paragraph style all paragraph style: all paragraph styles all paragraph style all paragraph style all paragraph style all paragraph style.

##### Heading 5 to Number Paragraphs

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph style. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

##### Heading 5 to Number Paragraphs

All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style.

##### Heading 5 to Number Paragraphs

All paragraph style all paragraph. All paragraph style all paragraph style who paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. All should paragraph style all paragraph. All paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style all paragraph style. Afterward, style all paragraph. All paragraph style all paragraph style all paragraph style all government office style all paragraph style all paragraph style all paragraph that the paragraph style.

# appendix. Optional

Appendix titles are also styled as **Heading 1**, minus a roman numeral—backspace to remove the roman numeral. Then, type “Appendix,” two spaces, a letter, and a title: “APPENDIX A. DATA.” *However,* *if you have only one appendix, do not add the letter “A.”*

If you apply Heading 2 style in your appendices, and the lettering does not begin with “A,” simply right click on the first Heading 2 of the appendix, and choose “Restart at 1.”

THIS PAGE INTENTIONALLY LEFT BLANK

# List of References

|  |  |
| --- | --- |
| [1] | B. Klofas, J. Anderson and K. Leveque, "A survey of CubeSat communication systems"," in *5th Annual CubeSat Developers' Workshop*, 2008. |
| [2] | P. Muri and J. McNair, "A survey of communication sub-systems for intersatellite linked systems and CubeSat missions," *Journal of Communications,* vol. 7, no. 4, pp. 290-308, 2012. |
| [3] | W. A. Beech, D. E. Nielsen and J. Taylor, "AX.25 link access protocol for amateur packet radio version 2.2," Tucson Amateru Packet Radio Corpoeration, 1997. |
| [4] | O. N. Challa, G. Bhat and J. McNair, "CubeSec and GndSec: a lightweight security solution for CubeSat communications," in *26th Annual AIAA/USU Conference on Small Satellites*, 2012. |
| [5] | "CubeSat Space Protocol: A small network-layer delivery protocol designed for CubeSats," Aalborg University, 2008. [Online]. Available: https://github.com/libcsp/libcsp. [Accessed 3 July 2017]. |
| [6] | Radar Systems Panel of the IEEE Aerospace & Elctronic Systems Society, "IEEE standard for letter designations for radar-frequency bands," The Institute of Electrical and Electronics Engineers, New York, 2002. |
| [7] | B. Kolfas, "CubeSat radios: from kilobits to megabits," in *Ground System Architectures Workshop*, Los Angeles, 2014. |
| [8] | D. Selva and D. Krejci, "A survey and assessment of the capabilities of CubeSats for Earth observation," *Acta Astronautica,* vol. 74, pp. 50-68, 2012. |
| [9] | Information Sciences Institute, University of Southern California, "RFC 791 Internet Protocol: DARPA Internet Program Protocol Specification," Defense Advanced Research Projects Agency, Arlington, Virginia, 1981. |
| [10] | Information Sciences Institute, University of Southern California, "RFC 793 Transmission Control Protocol: DARPA Internet Program Protocol Specification," Defense Advanced Research Projects Agency, Arlington, Virginia, 1981. |
| [11] | J. Postel, "RFC 768 User Datagram Protocol Internet Standard," Defense Advanced Research Projects Agency, Arlington, Virginia, 1980. |
| [12] | L. Eggert and F. Gont, "RFC 5482 TCP user timeout option," Internet Engeering Task Force, Fremont, California, 2009. |
| [13] | M. Swartout, "The first one hundred CubeSats: a statistical look," *Journal of Small Satellites,* vol. 2, no. 2, pp. 213-233, 2013. |
| [14] | D. J. W. Roger M. Needham, "Tea extensions," Cambridge University, Cambridge, UK, 1997. |
| [15] | K. Burda, "Error propagation in various cipher block modes," *International Journal of Computer Science and Network Security,* vol. 6, no. 11, pp. 235-239, 2006. |
| [16] | Y. Ko, S. Hong, W. Lee, S. Lee and J.-S. Kang, "Related key differential attacks on 27 rounds of XTEA and full-round GOST," in *Fast Software Encryption, 11th International Workshop*, Delhi, India, 2004. |
| [17] | J. Lu, "Related-key rectangle attack on 36 rounds of the XTEA block cipher," *International Journal of Information Security,* vol. 8, no. 1, pp. 1-11, 2009. |
| [18] | G. Hunyadi, D. M. Kumpar, S. Jepsen, B. Larsen and M. Obland, "A commercial off the shelf (COTS) packet communications subsystem for the Montana EaRth Orbiting Pico-Explorer (MEROPE) CubeSat," *Aerospace Conference Proceedings IEEE,* vol. 1, pp. 1-1, 2002. |
| [19] | J. Katz and Y. Lindell, Introduction to modern cryptography, 2nd ed., Boca Raton, FL: Taylor and Francis Group, 2015. |
| [20] | G. S. Vernam, "Secret signaling system". US Patent US 1310719 A, 22 July 1919. |
| [21] | D. Kahn, Codebreakers: the story of secret writing, New York: SCRIBNER, 1967. |

Apply **Reference List** style to your list of references to re-create or retain the formatting of this page. Remove manual line spaces that you have entered between entries, as the style comes with the proper spacing.

A *bibliography* is uncommon in NPS theses; a list of references is the standard. A bibliography differs from a reference list in that it also includes sources you consulted, but did not cite.

All in-text citations must have a matching entry in the list of references, with few exceptions; consult your citation style guide.

Use an established citation style such as Chicago, APA, AMS, etc. Made-up or hybrid styles will not be accepted. You are required to use a [department-required or advisor-approved](http://libguides.nps.edu/ld.php?content_id=17722653) citation style. Guides to the most-common citation styles used at NPS are available here: [Citation style guides.](http://libguides.nps.edu/citation)

Again, if you use reference-list generating software, such as RefWorks, ensure that you fill in all fields completely and accurately when creating your citation list. **You must edit references for punctuation and formatting** **after importing them**.

To edit most lists, you must remove the field code. Do this by highlighting all entries and pressing **Shift** + **Control** + **F9** at the same time. In Word’s citation manager, this is achieved by clicking on the list and choosing “convert to static text.”

Here are a few example entries:

Hawks, Mathew A. “Graph-Theoretic Statistical Methods for Detecting and Localizing Distributional Change in Multivariate Data,” Ph.D. diss., Naval Postgraduate School, Monterey, CA, 2015. (**Chicago N-B style**)

Naval Postgraduate School. (2017). Thesis\_template\_times [Word template]. Monterey, CA: Naval Postgraduate School. Retrieved from https://my.nps.edu  
/documents/105790666/106471216/Thesis\_Template\_Times.docx (**APA style**)

[1] B. Orend, *Morality of War*, 2nd ed. Tonawanda, NY: Broadview Press, 2013. (**IEEE style**)

THIS PAGE INTENTIONALLY LEFT BLANK

# initial distribution list

DTIC and DKL are the **only** two entities that are to appear on this page.

Include your desired recipients in your Publication Announcement List form. [Thesis forms](https://my.nps.edu/web/thesisprocessing/templates-forms)

1. Defense Technical Information Center

Ft. Belvoir, Virginia

2. Dudley Knox Library

Naval Postgraduate School

Monterey, California

1. Footnotes should be styled as “Footnote Text.” Each footnote must end with a period. [↑](#footnote-ref-1)