Final Project Review:
HDDlock

Renzo Silva, Fadi Maalouli, Brigit Lyons, Anthony Panetta

Department of Electrical and Computer Engineering
University of Massachusetts Amherst

***Abstract* – Current implementations of hard drive security are both inconvenient and inefficient. Users must remember many different passwords in order to retrieve their information. Given the abundance of Trojan horses and key loggers, information is exposed any time a password is entered into any operating system. HDDlock aims to make the process of securing hard drive data seamless and more secure by making the encryption process fast, operating system independent, and by moving user authentication to a remote location. The goal of this project is to create a Bluetooth-enabled hard drive protection device. The main components of this system will be an Android application, custom Bluetooth-enabled locking module, and PC application for transferring files to and from the security device. This document discusses the design of the system and presents progress towards developing a prototype. Final Project Review goals are presented and the manner in which they were met is described in detail.**

I. INTRODUCTION

HDDlock is a semi-automatic encryption device which allows users to safely control access to data contained within the hard drive of a computer. Through the use of a fast and user friendly application on their mobile phone, users will be able to encrypt files on their hard drive. The HDDlock device will remove the need to run the encryption on a possibly compromised system, therefore removing the possibility of any virus, Trojan horse, or key logger to intercept the password used to encrypt stored files.

The HDDlock system can be divided into three main components: the physical Bluetooth-enabled security module, which is plugged into the USB port of any computer, the Android mobile phone application, and the operating system software. The Android application will communicate with our security module via a Bluetooth link and will send the password for encryption and/or decryption to the security module after the user has been authenticated. A great deal of security is achieved by storing passwords on the Android device, since it makes it necessary to be in possession of both the Bluetooth module and the phone in order to access any information stored on the hard drive. It is important to note that the operating system is never aware of the password or encryption scheme used; the operating system is only used to allow users to manipulate their files on a familiar environment.

In the following document, design choices to meet the requirements are discussed. The design of and current Bluetooth module, the Android application, and computer software is addressed. A section on project management provides insight into the division of labor and efforts to maintain transparent development. Final Design Review goals and the way in which they were met are outlined in detail. In the

final sections of the document, current progress is discussed.

II. DESIGN

A. *Overview and Block Diagram*

The HDDlock system consists of three major components: the Android application, a Bluetooth–enabled security device, and a PC Software/GUI. In Figure 1, a block diagram shows the basic design of the system. In the following sections each of the three main components will be discussed in detail.

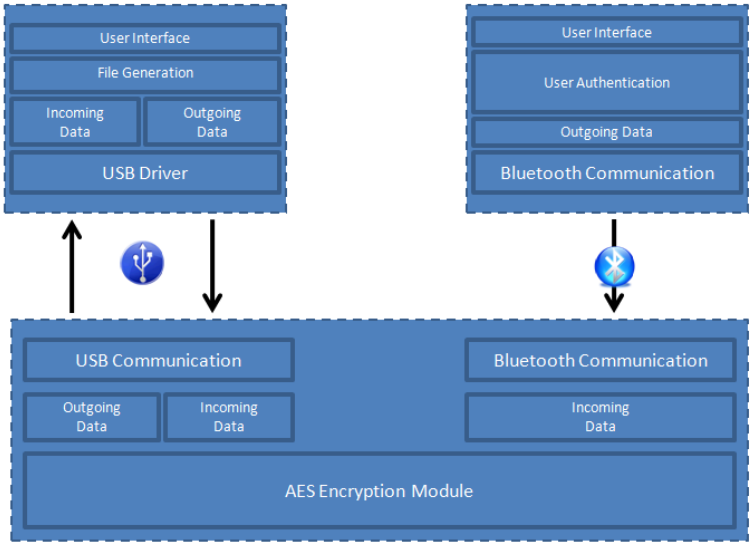

Figure 1: System Block Diagram

B. *Security Device*

The Security Device consists of two main components: an FPGA and a Bluetooth module. The device allows users to encrypt and decrypt data files stored on the hard drive of a computer.

The encryption and decryption of data files is handled by the FPGA and Bluetooth module. While in production, these devices should be small, low power, and share a common communication method. Extensive research into available products revealed that the use of Universal Asynchronous Receiver Transmitter (UART) would be a reliable method to communicate between the FPGA and Bluetooth module. With this implementation, an encryption key is sent from an authenticated user on an Android phone to the Bluetooth module. The key is then transmitted to the FPGA for further processing.

The Bluetooth module must be configurable to allow design flexibility. The module should have the ability to remain in a low power, idle state when not in use. The Bluetooth module should be only discoverable to surrounding Bluetooth-enabled devices within a short range, so that users cannot access the device unless they are within close proximity to the computer. Additionally, valid keys will only be transmitted to the security device by authenticated users. These two aspects of proximity and hard token authentication strengthen the security of the HDDlock system.

The FPGA takes the transmitted key and uses it to encrypt or decrypt user files as desired. The FPGA should have enough memory to hold an AES-128 code algorithm for encryption/decryption, and be able process the file that user wants to encrypt/decrypt. Extensive research has determined that using a DE2 development board is an ideal approach to realize a prototype for this system. The DE2 board has many desirable features like the Cyclone FPGA device, RS-232 transceiver, USB 2.0 (type A and type B) and 8-Mbyte SDRAM memory.

C. *PC Software/GUI*

The PC software/GUI provides the user interface for our embedded application, from which it allows users to encrypt or decrypt files. The software was created as a

Windows Forms Application using Visual C++. The GUI is located on a PC which is connected to the NIOS Processor (embedded software) of the DE2 board via USB and transfers data to and from the hardware encryption device. The embedded software is built by changing and adding capabilities to a reference design given by the Altera DE2 board [5]. The user interface is shown below in Figure 2.

The GUI offers the following functionality:

- Select the input file to be encrypted or decrypted
- Output file is created by overwriting the input file with the results of the encryption or decryption
- Provide real time information on encryption/decryption and transfer of data
- Lock Button erases the encryption key stored on the device. Prevents device from being detected by the GUI, unless you have a successful Phone login
- Button that directs you to our website
- Begin Loop Back (In this case, data is sent over USB to the DE2 board and returned without being encrypted).

D. *Android Application*

The Android Application consists of two main sets of components. These are a set of activities and services for managing a Bluetooth connection with the security device and a set of activities for authenticating users in the system and managing user accounts. The program flow of the Android application is shown below in Figure 3.

The Bluetooth connection activities and services are built off of the open-source project Bluetooth Chat [1] supplied by Google. The user account activities consist of a series of GUIs that allow users to setup new accounts, manage password recovery options in case of hardware loss, and login to their account.

To provide the greatest amount of user flexibility, user authentication is not reliant on a data connection; all authentication is done on the phone itself. This allows users to access their data in locations were internet and/or cellular coverage is not available. To allow users to maintain access to their data in the event of phone loss, the password for decrypting data stored on the hard drive is generated based off of the user's login information.

Figure 2: PC Software User Interface

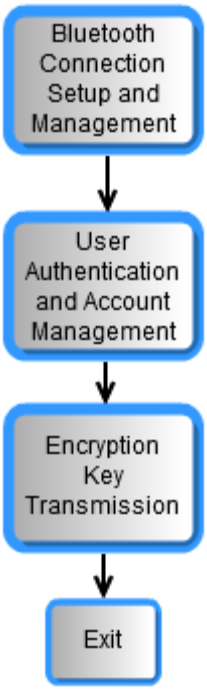

Figure 3: Android Application Program Flow Diagram

III. FINAL PROGECT REVIEW PROTOTYPE IMPLEMENTATION

A. Objectives

During Cumulative Design Review, several goals were set for the Final Project Review. These goals include implementing a user interface to send and receive files between the computer and the HDDlock device, finalize communication between NIOS, FIFOs, and encryption/decryption modules, and implement user-friendly options for the Android application.

B. Bluetooth communication scheme

Communication between the Phone and DE2 board is accomplished using Bluetooth wireless technology. All smart phones contain Bluetooth hardware allowing communication with a wide range of Bluetooth enabled devices. With proper programming and configuration, an Android application was created to facilitate communication with the Bluetooth module RN240 that is connected to DE2 board via a serial port RS-232. RN240 was chosen because it supports bi-directional RS232 signaling at a rate of up to 464Kbps. In order to receive data from smart phone on the DE2 board, embedded software had to be written onto the NIOS processor in order to grab the data one byte at a time from the Bluetooth module RN240 onto the board, so encryption/decryption can take place.

C. AES Encryption in Hardware

Based on previous analysis of AES encryption performed in both hardware and software, hardware encryption was selected

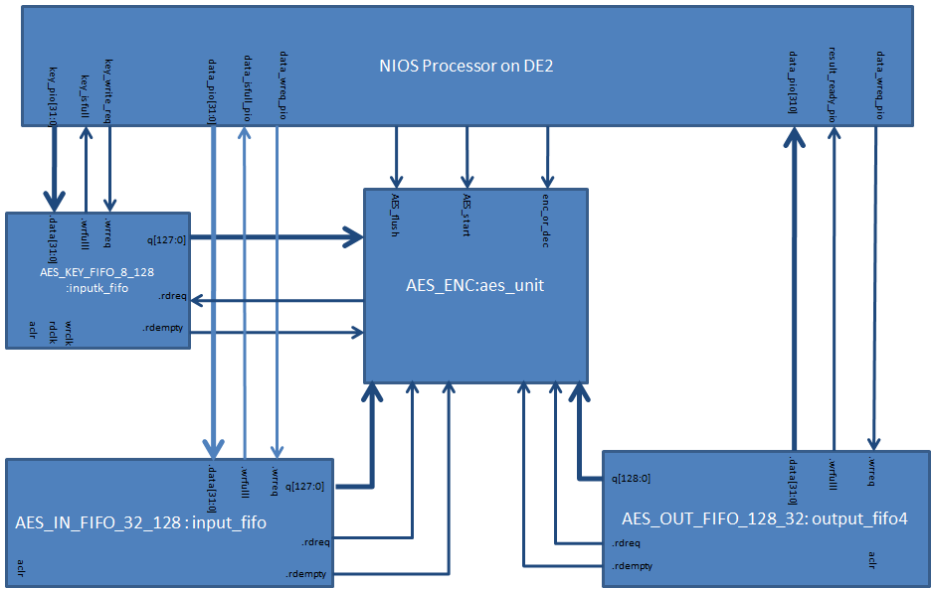

Figure 4: Hardware Architecture

for its large performance advantages. In order to implement AES encryption in hardware on our Altera DE2 development board, the following hardware architecture was created:

1. AES Control

At the center of the AES hardware implementation on the FPGA is the AES control logic. This hardware is responsible for collecting the AES key and input file from the user, performing the desired AES operation on the input file, and outputting the result back to the user. This is performed through three main module components.

i. Control State Machine

The overarching logic that the AES Control module runs is a state machine which controls data flow. This includes a selector for whether the AES operation will be encryption or decryption and collection of information to be processed from the surrounding logic.

ii. Encryption Module

The AES Control Module instantiates an encryption module. If the control flow dictates that an encryption is to be performed, the AES Control passes the input key and file to this module, waits for the encrypted result, and presents it to the output buffer.

iii. Decryption Module

The AES Control Module instantiates a decryption module. If the control flow dictates that a decryption is to be performed, the AES Control passes the input key and file to this module, waits for the decrypted result, and presents it to the output buffer.

2. Input Key and File Buffers

In order to process the AES data, two buffers had to be introduced on the input side of the AES Control logic. This includes one for the key and one for the input file. This acts as a means for holding data until it is ready to be processed by the AES logic. Once the key has been received by the RN-41 Bluetooth module, it is processed by out embedded control software and sent to the key buffer. Likewise, the file to be encrypted or decrypted is sent into SDRAM and then collected by the input file buffer for AES processing. These buffers take in data in 32-bit blocks and output 128-bit blocks. The reason for this is that the central data bus on the NIOS processor only allows 32-bit communication to the embedded software platform and all AES operations require 128-bit blocks of data.

3. Output File Buffer

Similar to the input buffer, an output buffer is required to present the result of the AES operation to the embedded control software. This receives results from the AES control module in 128-bit blocks and outputs data to the embedded software in 32-bit blocks.

4. Performance Analysis

The following shows the result of performing one 128-bit round of encryption on the full AES hardware system:

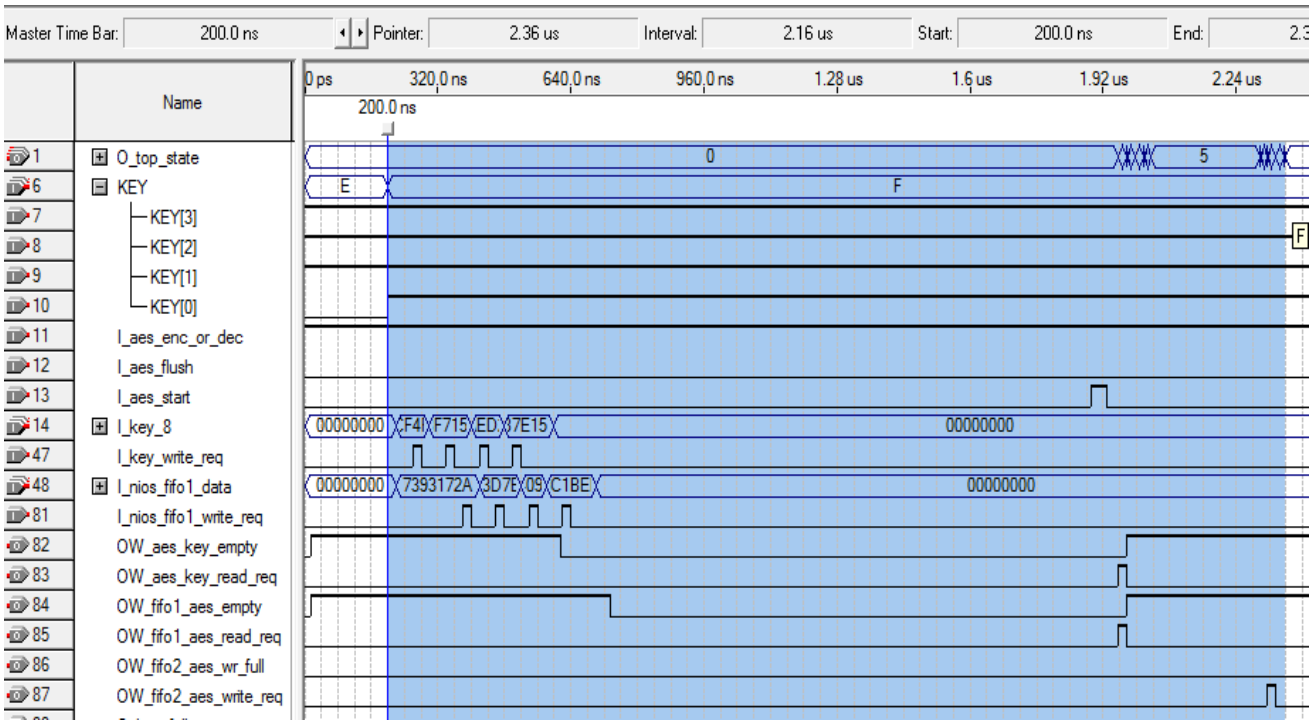

Figure 5: Simulation Result for Hardware Encryption

Here we can see by the interval highlighted in blue that one full iteration of the hardware encryption algorithm requires 2.16µs running at the DE2 clock frequency of 50MHz. This equates to about 60Mb/s. This interval includes the time required to load the input buffers, perform the encryption, and load the result into the output buffer. Encryption and decryption results were verified for accuracy using the AES test vectors found at [4]

D. USB Communications Scheme

PC software sends 4 'setup' packets (i.e. 32 bytes) to the embedded software. The first two packets are the Setup Packet ID, and the AES settings, which tell whether to encrypt or decrypt. The third and fourth packets are the size of the file to be encrypted/decrypted. As side note, two other packets (i.e 16 Bytes) come from the smart phone, which is the encryption key.

This setup data contains the parameters for the operation. The encrypted/decrypted file is transferred from the PC to the DE2 board in packets of 8 bytes each, and once the file is modified, it is sent back to the PC in modified 8 byte packets. The AES operation acts on 16 byte blocks, therefore two 8-byte packets are sent to the embedded software to be encrypted/decrypted, and also received from the DE2 board, before new data is transferred. This process continues until the entire file is transferred across the communication channel. The user also has the option of passing unencrypted data across, and often this can confirm that the file passes to and from the board correctly. If there emerges a case when the number of bytes in the file is not divisible by 16 bytes, the end of the data must be padded until the final packet equals 16 bytes in length. For example, if the file is 40 bytes long, 8 padding bytes will be added to the final block.

If the file size is equivalent to a whole number of 16 bytes, the data is padded with a block of 16 padding bytes. This ensures that there will always be padding bytes at the end of the file, so that padding bytes can be distinguished from data at the end of the file. This padding process has one side effect which is: only files previously encrypted by the PC Software can be decrypted correctly.

E. Android Application

1. Bluetooth Communication

The basic Bluetooth functionality was built around the existing BlueTerm [2] application. BlueTerm [2] combines the Bluetooth Chat [1] sample application provided by Google and the Term [3] application that is available on the Android Open Source Project. Building the application around this framework was an excellent way to debug Bluetooth communications between the phone and Bluetooth device because it allowed transmissions to be seen in a terminal format.

Since further progress has been made with the project, the terminal GUI has been hidden so that normal users cannot see data transmissions between the phone and Bluetooth device.

2. User Accounts

All account information for users is stored in a text file on the internal storage of the Android phone. All data that is stored on the phone is encrypted using AES-128. The encryption key is a 128 bit value that is generating by using a SHA-256 hash function of the username and password.

When a user creates a new account, they are asked to supply a username and password. Users are also given the option of answering some security questions that will allow them to recover their account and setup a new password in the event that they forget their current password. There are several Android activities that were created to enable login, account creation, account recovery, and account management.

In the event that a user loses their phone, they will still be able to access their data by creating a new account on a new phone. This account must have the same username and password that were used when their account was originally created on the initial phone. This functionality also means that two or more phones may be used to access the same data on the hard drive, as long as both accounts use the same username and password.

A diagram showing the GUI interactions of the various Android activities and services is shown below in Figure 7.

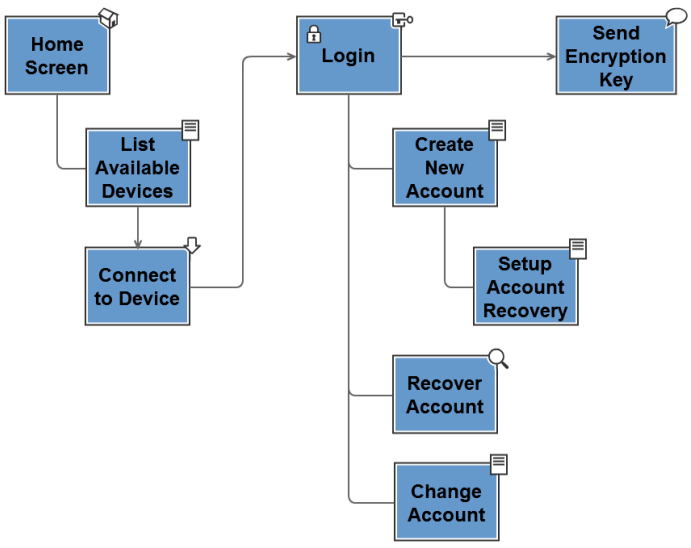

Figure 6: Android Application Layout

IV. COST ANALYSIS

Actual Product Cost

Product	Cost
RN-240 Bluetooth Module	$59.99
RN-240 Power Cable	$14.99
JTAG Transfer Cable x2	$10.00
USB Bluetooth Transceiver x2 (for testing)	$39.98
DE2 Development board	Borrowed (Cost $280 if purchased)
Total	$124.96 ($404.96 if DE2 purchased)

Projected Manufacturing Costs

Ics/Components	1 Unit Price	10 Unit Price	100 Unit Price	1000 Unit Price
Altera CycloneII FPGA	$90.70	$83.44	$77.10	$63.49
8 Mbyte SDRAM	$1.15	$1.04	$0.92	$0.81
512Kbyte SRAM	3.66	3.16	2.9952	2.2464
50Mhz Oscillator	$2.34	$2.06	$1.83	$1.59
USB-Host/slave Controller	$4.35	$3.87	$3.44	$2.96
RN-41	$24.95	$24.50	$21.88	$20.00
HOUSING				
3D Print	$99.99	$99.99	$65.00	$7.50
Circuit BOARD				
PCB fabrication	$33.00	$29.80	$11.45	$8.77
Total Price per Unit	$260.14	$247.86	$184.60	$107.36
Total Production Price	$260.14	$2,478.60	$18,460.19	$107,360.60

*Fabrication costs for the FPGA device are several thousand dollars and outside of the scope of this project's budget

V. PROJECT MANAGEMENT

Each team member was responsible for specific administrative and technical components, but all worked together when needed. Individual responsibilities were designated to meet milestones and to ensure completion of deadlines.

- Project responsibility
  - Brigit Lyons (CSE) – Web and software development, specifically Android application
  - Fadi Maalouli (EE) – Interfaces between hardware components, specifically PC to DE2 and Phone to DE2
  - Tony Panetta (CSE) – Hardware & Software development, specifically encryption/decryption
  - Renzo Silva (EE) – – Hardware & Software development, encryption interfacing
- Administrative
  - Brigit Lyons - Webmaster
  - Fadi Maalouli - Scheduling
  - Tony Panetta - Documentation
  - Renzo Silva – Purchase Orders

VI. APPENDIX

A. Application of Mathematics, Science and Engineering

In order to accomplish the goals set out in this project, we must utilize many Mathematics, Science, and Engineering skills learned in recent years. To successfully develop the system, we must use the principles of Systems Engineering learned in ECE597SE. In particular, it is necessary to construct a list of requirements that constrain the scope, cost, and performance of our system. Also, we have used the principles of Systems Engineering when considering how our different components will interface with one another. For example, the Android phone must be able to interface with our hardware module via Bluetooth and provide the correct command interface to our RN-41 module in order to pass information.

Throughout the process of developing our project, we have collected data that we needed to analyze. In order to best represent and this information for analysis, we have used techniques learned in ECE314, Probability and Random Processes. Finally, our project required both hardware and software programming on an FPGA. This made use of skills learned in ECE 354, Computer Systems Laboratory II.

B. Design and Performance of Experiments, Data Analysis and Interpretation

This project uses AES encryption in order to protect the information on the user's hard drive. In order for our project to be practical and effective, the encryption needs to be performed at a reasonable speed. We have performed tests on AES encryption algorithms written in both hardware and software development languages. The software implementation is a version of the algorithm written in the C programming language. To test the performance of this algorithm, the program was run in increasing numbers of iterations ranging from 1 to 10,000. The total runtimes were collected and plotted in terms of Gb/s. The hardware implementation was tested using an application of the algorithm written in Verilog Hardware Description Language. Here, the algorithm was run several times and the performance was obtained through analysis of the waveform output.

Comparison of the data obtained from these two experiments was not a trivial task. Due to the fact that the software test results were in terms of Gb/s and the hardware test results were in terms of clock cycles, manipulations had to be performed to compare the test results. It was decided to convert the results of the software test into clock cycles per run. This posed another difficulty as the hardware results were in terms of actual processor cycles, while the software results took into account both processor and idle cycles. In order to accommodate these differences, a conservative correction factor of 1 processor cycle per 100 running cycles was applied to the software results. This now provided common ground for comparison and allowed us to realize that a hardware implementation would be at least ten times faster than a software implementation.

C. Design of System, Component or Process to Meet Desired Needs Within Realistic Constraints

The system that we are developing is driven by two main factors: Cost and Performance relating to the speed of encryption in our HDDlock hardware module. The requirements of our system are that the cost shall not exceed $500 and the speed of encryption shall reach 3Gb/s.

Through research and testing, we have determined that these two requirements cannot both be met concurrently. It will not be possible to create a hardware encryption module capable of speeds of 3Gb/s while also maintaining our budget. We have decided to ease the speed requirement because it will allow us to stay within our budget while still being able to create a proof of concept implementation of our overall system which will show that with sufficient funds it is also possible to reach the desired speeds. We ended up with an implementation of hardware encryption that runs at roughly 60Mb/s.

D. Multi-disciplinary Team Functions

Renzo Silva, EE. Renzo worked on developing integration hardware for AES module and helped develop embedded control software for the DE2 board. He also worked on improving transmission latency between the phone and the Bluetooth module. Renzo was responsible for handling purchase orders.

Fadi Maalouli, EE. Fadi interfaced the components of the project, including integrating the PC to DE2 board, phone to DE2 board. He also worked on developing the PC Software/GUI and embedded software on the DE2 board. Fadi was also responsible for of all the scheduling for design reviews.

Anthony Panetta, CSE. Anthony optimized and tested the encryption algorithm in both Verilog and C. Worked on development of encryption control hardware. Anthony also helped with the development of embedded software.

Brigit Lyons, CSE. Brigit developed the Android application, including Bluetooth communication and user authentication and account management functionalities. Brigit was also responsible for developing and maintaining the team website.

E. Identification, Formulation, and Solution of Engineering Problems.

The main obstacle we have encountered is our inability to encrypt information at the rate of 3 Gb/s needed for SATA communication while mitigating costs. After extensive research we decided to get rid of the idea of connecting our security device directly to the hard drive and instead work with the operating system andFPGA DE2 development board. We also struggled with integration of the three main components but were able to successfully create the necessary interfaces to implement the system.

F. *Understanding of professional and Ethical Responsibility*

One professional responsibility that we had to address was the reliability of our system. Since the safety of the information relies on our system, we have to make sure that every part of our system is working exactly as planned. Our project is intended to deal with sensitive information and it is necessary to ensure all the information gets encrypted and decrypted properly. In order to ensure that we have developed a system that functions properly, we will have had to extensively test each component for correct functionality.

G. Team Communication

The team members held weekly meetings where each team member presented their progress in research and design via PowerPoint or whiteboard presentation. During these meetings, design ideas were criticized, design problems were resolved, and deliverables for the next meeting were identified. The team additionally met roughly once a week with the project advisor, Professor Zink in person

or via electronic communication. These meetings were to discuss issues, current progress, and future goals. Communication between meetings was done via email, and Google Calendar was used to facilitate meeting organization.

H. Understanding of the impact of engineering solutions in a global, economic, environmental and societal context.

The main goal of this project is to secure data on a hard drive so that it is protected against any malicious or unapproved use. This could lead to many positive societal impacts because it allows people to have peace of mind when they are away from their computers. This project can also lead to economic gain by allowing corporations to better protect proprietary information.

On the other hand, this level of security could also have negative social impacts if it is used by people with malicious intents to hide information from authorities.

I. Application of material acquired outside of coursework.

In order to complete aspects of the HDDlock project, many sources outside of coursework were used. Below are some examples of such sources:

1. Product user manuals, including user manual for RN240 Bluetooth module [1].This user manual enabled us to learn how the Bluetooth module works and how to configure it.
2. Open source code [3], for the android application. This application enables us to communicate with the Bluetooth module via an Android phone.
3. When the team ran out of resources for help on a particular issue, online discussion forums were used to resolve the issue at hand. Team members both consulted and participated in online discussions where they were able to ask people who have worked on similar issue. For example, the team faced significant problems while trying to configure the Bluetooth module. After consulting online discussion forums, it was learned that there were many reported issues with using the Windows 7 terminal emulator for configuration, and that the Windows XP terminal emulator was known to be much more successful. After consulting these forums, the team was able to solve the issues they faced in configuring the Bluetooth module.
4. The Altera forums and DE2 user manual [5] were very useful for hardware design and integration issues.

J. Knowledge of contemporary issues

With advances in technology, people want to store more information on hard drives because it is easier and more convenient than alternative physical media. Storing personal or classified information on digital media, however, opens the door for exploitation and malicious use. Current solutions to this problem are limited to software-based encryption that uses a static password. While valid, this approach can be easily broken into and is not sufficient enough to protect the most sensitive personal, corporate, or governmental data. Therefore, motivation behind this project is to provide a more secure method of protecting vital information by using a more robust encryption scheme that provides

more security to users without sacrificing usability. Examples of potential users include individuals in the consumer market, corporations and smaller businesses, as well and military and government personnel. By employing a robust Bluetooth-enabled encryption key scheme, this project offers a more secure model without requiring the user to do anything more than enter their password as they would with static-password based software encryption. By requiring users to log in with a phone application that validates the hardware identification of the phone, the project presents another layer of authentication without introducing any hardware that the user is not already carrying around with them on a day-to-day basis.

K. Use of Modern Engineering Techniques and Tools

This project involves the use of many modern engineering techniques and tools. Four examples include: Bluetooth, the Android operating system, FPGA development with the DE2 board, and the Eclipse IDE. One of the goals of this project was to introduce remote Android application authentication into the requirements for accessing secured data. Most users are likely to already be carrying a smart phone with them, so using that device for hardware authentication increases usability by eliminating the need for users to carry around a specific device, such as a hard token. This approach requires a secure, short distance communication platform to relay messages between the phone and encryption module; Bluetooth was chosen because it is widely available in contemporary smart phones, highly secure, and not dependent on outside connections such as Wi-Fi or 3G.

To allow users to authenticate with their smart phones, an application is required to facilitate account information/authentication and Bluetooth communication with the encryption module. Because of the scope of this project, this application is only being created for one smart phone platform. Android was chosen because it is open source and is rapidly growing in popularity. Android development is conveniently done in the Java programming language, which all four team members have learned in ECE 122 and ECE 242. This fact has an added benefit of allowing all four members the ability to write and debug application code without requiring all of them to proficiently learn a new programming language while executing all other aspects of the project.

The Android SDK plug-in for the Eclipse IDE is being used to efficiently develop the Android application. All four members are already familiar with Eclipse, since it was introduced in ECE 242. It is a robust IDE that allows for easy development, compiling, debugging, and management of code.

REFERENCES

[1] "BluetoothChat – Bluetooth Chat". Andoid Developers.Web. http://developer.android.com/resources/samples/BluetoothChat/index.html

[2] "BlueTerm - Android App on AppBrain." AppBrain Android Market - Find the Best Android Apps and Games. Web. http://www.appbrain.com/app/blueterm/es.pymasde.blueterm.

[3] "Term – Android Open Source Project." Web.http://android.git.kernel.org/?p=platform/development.git;a=tree;f=apps/Term

[4] "AES Test Vectors." Web. http://www.inconteam.com/software-development/41-encryption/55-aes-test-vectors

[5] "DE2 Development and Education Board User Manual." Web. ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf