Date: January 20, 2016
To: Drs. Stine, Latino, Krasinski, and Piao
From: Christian Coffield, Matthew Dekoning, Nathan Lea, Kevin Seitz
Subject: Proposal to Develop a Secure Photograph Capture System

The purpose of this proposal is to introduce our team's take on a Secure Photograph Capture System. In it, we will lay out the project definition, our interpretation of how to accomplish it, and outline our approach to achieving that. This includes a description of the various sections of development, a realistic schedule of set deadlines that we intend to meet, information on the team members' skills, responsibilities, organization, and acknowledgement of potential roadblocks with alternative plans in mind to counteract them. It also includes a basic rundown of necessary, big purchases that will be made for the project's development. We hope that with this proposal, you will feel confident in our team's ability to complete the project in the coming months.

The Project

The Secure Photograph Capture System is defined as follows:

"A system that snaps a picture and stores the picture in memory securely. A good system will enable a security protocol, hopefully with military-grade encryption, so that any system cannot possibly be compromised by a user or thief. The picture should also be able to be recalled at a later time."

To accomplish this, we have devised a three-piece solution, consisting of a camera-equipped and internet-connected microcontroller to capture and encrypt an image, a remote server to store it, and software to receive and decrypt the image on a separate terminal.. The communication between the device and server does not need encryption, because the files being transferred are already secure.

The Goal

The full aim of this project is to produce a system that, once powered on, is entirely self-sufficient in capturing, encrypting, and storing images. These images should be stored in a secure location away from the device taking the photos. In this way, the photographs are stored independent of the camera device once taken; if a thief noticed the device took a photo and stole it, they would not be able to access the pictures (because all but the most recent are stored off-site, and the most recent is encrypted) and the owner of the device would still have access, meaning they would hopefully be able to have a photographic record of the thief. This means a system comprised of three sub-systems.

The physical device itself must be small and lightweight, with a stable internet connection, a consistent source of power, and a medium-duty casing or shell. As for electronic components, it

must contain a camera of a sufficient resolution to capture clear photographs, and a microcontroller that is fast enough to a) capture and store an image as a standard file format, b) encrypt the image using military-standard AES-256 encryption, and c) upload the encrypted image to a remote, offsite server for safekeeping. This entire process should be done in under one minute.

The server must be capable of storing thousands of reasonable quality images. This can be done in a variety of ways, though one of the more popular and easiest implementations is with a microcontroller such as a Raspberry Pi and an external hard drive or set of hard drives, using RAID.

To access the images on the server, and decrypt them, an external application is necessary. This application should be robust enough to run on, at the least, Windows PCs; if at all possible the app should support Windows, Mac, and Linux. There is also potential for mobile support, such as Android and iOS. This app should be fairly simple in nature, using the same key from the physical device to decrypt the photos that were encrypted and stored on the server. Optionally, this application may show the decrypted photos and prompt the user with an option to save them to their hard drive. The app should automatically connect to the server (be generally plug and play, much like the physical device). It should also be simple enough for a non-technical user to retrieve images.

Project Specifications

Attached, please find the detailed list of requirements and specifications for this project.

Development Plan

Our development plan consists of three separate phases that will be occurring simultaneously. These three phases correspond to the above mentioned subsystems in the project, which includes the camera-equipped device itself, the remote server, and the application used to access and decrypt the photos.

Phase I: Photography and Encryption

The process of taking and storing an image, encrypting it, and uploading it remotely is fairly complex. Doing so requires more processing power than a smaller IC-style microcontroller can deliver promptly. With this system, photographs need to be delivered to the server as soon as possible, so that additional photos can be taken and they are secure in a new location before a potential intruder can destroy, disable, or simply steal the device. As such, we are considering the use of a microcontroller with support for network connection and USB built in, such as an ARM processor/breakout board combo, Arduino, or Raspberry Pi.

The encryption method we have chosen is AES-256. This is currently NSA’s method of choice for protecting up to top secret information. It is a symmetric block cipher that is commonly used

to encrypt long strings of bytes systematically, based on a key. This key is necessary to both encrypt and decrypt data; it's used in an algorithm that involves logical XOR, or exclusive-or operations, to change the data into something unrecognizable. XOR is a reversible operation, so once the data is encoded with a key that only the system knows, it can be decoded by applying the same key. Because the project requires that military-grade security protect the photographs, this has been determined to be the fastest, most secure method of encryption.

The photos themselves must be taken autonomously, which can be done in a variety of ways--motion or light sensing, pressure plates, etc. The option that makes the most sense is a passive infrared motion sensor with a conical view. This avoids the external wiring of a pressure plate and keeps everything self-contained in the device, and allows for it to function regardless of lighting conditions.

Phase II: Off-Site Storage

In order to store the encrypted photos safely away from the camera a server will be set up off site to store them. The server will be responsible for receiving data from the camera, storing it by date, and correctly interacting with the desktop application to serve images to correctly authenticated users. The server consists of a Raspberry Pi connected to a mass storage device(s) via USB. This mass storage device could be a terabyte hard drive or an array of flash drivers. RAID 5 will be implemented if the mass storage consists of three or more devices.

The server will accept encrypted pictures from the Camera Pi and save them to the mass storage device(s) named as the 'date time' received. The server will continue to save pictures as the camera takes them. When contacted by the desktop client program the server will first check to make sure the user is properly authenticated, then serve the encrypted data over Wi-Fi to the client. The authentication step is still being devised.

Phase III: Retrieval and Decryption

AES-256 is an encryption algorithm that is easily reversed, so long as the decrypting agent knows the key used in the initial encryption. We will develop a software application that matches the key provided to the encrypting microcontroller, and uses this key to decrypt files from the off-site storage server. Java is a popular programming language that can compile and run on all major OS', A useful alternative would be C/C++. Eventually, it should run in a self-contained .JAR file (in Java) or .EXE (in some variation of C).

The application would have a standard GUI that displays the image server's contents for selecting. Once a file is selected, it will be automatically decrypted. The resulting decrypted file should either be saved to the user's hard drive or displayed on-screen, with an option to save it. AES open-source libraries exist for most programming languages, and they can be used to decrypt the photographs once they are retrieved. Most of these libraries only support up to AES-128, but the process can be expanded upon to encompass the more secure algorithm if need be.

As stated above, the application will allow for the creation of new keys. This method will be done

by having the user enter a passphrase and that will be used as the seed for the new key. This method of creating the keys will be duplicated on the photography system so that the keys will never be passed over the internet. The keys will not be stored on the host computer, the user will have to enter the passphrase each time they start the application.

The Schedule

Following is a chart with milestones, deadlines, and who is responsible for completion. For a more in depth look, please see the attached Gantt chart:

Deadline	**Milestone**
Jan 26	Design Proposal due
Feb 18	Working prototypes for all components
Feb 22	Prototype presentation
April 1	All systems integrated successfully
April 11	Trial Project Inspection
April 18	Oral Report
April 25	Design Day presentation
April 25	Final Report due

The Team

Team Member	Qualification	Responsibility
Christian Coffield	● Microcontroller programming experience ● General experience with programming and discrete math/encrypting ● Writing experience	● Manage encryption of files on microcontroller ● Send files to remote server ● Produce content for wiki ● Format and edit all documentation for submission
Matthew Dekoning	● Raspberry Pi experience ● Digital hardware design ● Low level software experience	● Record notes at Meetings ● Set up image server ● Connect device to internet
Nathan Lea	● Software ● iOS developer ● High level user interface	● Select microcontroller and advise on its usage ● Write decryption client application/client access to system ● Manage wiki/website (TBD)
Kevin Seitz	● Android and Java programming ● Digital and Analog hardware design	● Camera selection and management ● Photo capture programming ● Device case design
All	● Technical experience in hardware and software ● Experience with technical documentation ● Professional teamwork experience	● Document own work ● Contribute to team discussion and decisions ● Cooperate with team members to produce a high quality system

Team Management and Organization

This project has no formal team leader. All team members are expected to contribute as equally as they can to the project, and play to their strengths to produce a high quality system. Conflicts will be resolved by vote, and in the event where a ¾ agreement cannot be made, the team will seek advisement from Dr. James Stine, their advisor.

Meetings with Dr. Stine will occur every Thursday at 3:30 for the duration of the semester. All teammates are expected to attend each meeting. Matthew Dekoning is charged with taking notes over the highlights of the meeting and recording them into a document on Google Docs, called the Meeting Minutes. This document will provide a frame of reference for all teammates to look back on and ascertain what they should be working on.

Attached is a block diagram, showing the current plan of work distribution for all members of the team. Each block is representative of a section of the project that teammate is in charge of researching, managing, and prototyping. Team members are actively encouraged to assist with others' work if their own is finished, to produce a higher quality device and support a smoother integration. Working alongside others to achieve tasks decreases the likelihood of incompatibility between components later on in the project.

All project documentation will be saved in Google Drive for easy sharing. Any code written will be uploaded both to the Drive and to a GitHub repository to provide version control. All team members are expected to document their own work as they progress, with reasonable quality and content, and contribute towards all proposals, presentations, and reports. Christian Coffield is charged with revising and formatting all documents that are to be submitted to advisors or committee members.

Managing Roadblocks

After two working days of being stuck on the same problem, team members are required to consult each other. After three working days without significant progress on a specific part of the project, Dr. Stine will be contacted to seek further guidance.

The first predicted roadblock is a camera interface problem. To overcome this the team will become proficient using openCV libraries, and investigate different image handling libraries. There are many options of varying overhead costs to process images, so we will use one that fits our needs, even if it is not one we are familiar with.

We also have a contingency in place for capturing photos consistently. We would like to pursue motion-triggered capture to ensure a photograph would be taken as soon as possible if its location is invaded. However, in the event that the otion triggering is unreliable, we have the capacity and storage to manage a large volume of photos at the resolution we plan to use. We have enough storage space and the communication is lightweight enough that uploading one image every minute would not apply undue pressure on bandwidth or server space.

Another possible problem is the Raspberry Pi being unable to capture and encrypt the image in a timely manner. In this event, the camera capture controller will be shifted to the BeagleBone Black, a more powerful microcontroller.

Necessary Resources

The following is a brief cost prediction of the resources needed to complete the project. It is not a guarantee of what the total expenditure will be. Some components may fail, and others may not work as intended and need substitutions or alternatives.

Resource	**Cost**
Google Drive	Free
Raspberry Pi 2	$35
Raspberry Pi Zero	$5
Webcam/Camera	$15
PIR Motion Sensor	$3
USB Hard drive/flash memory storage	$40

Conclusion

Thank you for taking an interest in this project. We hope to receive feedback on our proposal to be as effective as possible in designing a quality device. Once approved, we would like to proceed in designing the three subsystems mentioned in the above proposal. If there are any questions, please contact any of the team members through their okstate email, in the OSU directory. We look forward to hearing from you, and hope to produce something that Oklahoma State University can be proud of. Go Pokes!