Research & Investigation Report

Natalie Jablonsky, Matthew Leeds, Oliver Necovski

Introduction

As a team we are building a low-cost GSM-capable device designed to transmit encrypted messages over mobile networks such that they if they are to be intercepted they cannot be read by attackers.

To help decide on a method to provide encryption to the Arduino board existing implementations were examined to see how they tackle the problem of distributing encrypted messages. The pros and cons of Project Cuckoo, Text Secure, Blackberry Messenger, Medical Secure Wireless Modelling and Telegram to measure how effective existing methods are, and analyse what can be implemented in our own solution.

Existing Systems

Project Cuckoo

Background:
Cuckoo is an encryption system developed by Jochen Maria Weber (Weber, 2015). It implements arduinos with social media APIs to transfer encrypted messages over social media. Cuckoo's encryption method is to transform each letter of the message into sentences which are then passed through several social media channels (Twitter, Tumblr, Skype) (Visnjic, 2015). These sentences are posted next to randomly generated sentences to create noise and distractions. The message is then finally reassembled at the receiver's end.

Pros:

- Uses the same hardware, communicating with computers and other arduinos
- New encryption system which allows for the method of encryption to be changed with every message

Cons:

- No technical information, the designer doesn't provide any code or protocols.
- Appears to be more of a proof-of-concept device
- The encryption system hasn't been readily tested

AES on Arduino

Background:

Advanced Encryption Standard (AES) is a symmetric encryption algorithm. AES was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen (aesencryption, 2016).
As AES is a symmetric encryption algorithm, this means that the key is used for both encryption and decryption. AES supports a block length of 128 bits and uses 3 key lengths; 128, 192 and 256 bits (aesencryption, 2016) Each key length is encrypted in different rounds. Rounds refers to the number of times the data is put through the encryption algorithm.
Rounds:

- 128 bit key length: 10 Rounds
- 192 bit key length: 12 Rounds
- 256 bit key length: 14 Rounds

Overall having a higher amount of rounds increases the security of the encryption, however this also increased the memory required and the time it takes to encrypt and decrypt plaintext.

AES encryption diagram:

Encryption Libraries:
Within the Arduino software it is possible to insert various libraries to your sketches. These libraries can be used in order to connecting your arduino to various networks such as gsm or wifi and for reading and writing on different devices. Throughout the arduino community we have also managed to find various encryption libraries. Some of the types of encryption libraries available include; SHA, DES and AES.
For the implementation of AES within this project we will be using an AES library that has been previously successfully implemented with the Arduino Mega. This library works with the three key lengths of AES; 128, 192 and 256 bits.

Memory:
The Arduino Mega, comes with 256kb worth of memory. This is an incredibly low amount of memory and causes a concern for the use of an AES encryption. This is due to the storage required to successfully implement this encryption combined with the memory that is already being used in other areas of the project. However it is possible to drastically increase the amount of storage space on the arduino through the use of an SD Card Shield Adapter. This adapter will allow for read and write on the sd card as opposed to the onboard memory.

Pros

- Secure
- Reliable

Cons:

- Storage space (Arduino limitations)
- Possible SD Card Shield Requirement.

Text Secure

Background:
Text Secure was an android application (2010 - 2015) that used end to end encryption for SMS and Data transferred Messages that only other “Text Secure” users could see.
Using Data transferred messages was its main purpose although it did have an option to encrypt the message and send it over a mobile network to the receiving end which was then able to open it up in the “Text Secure” app to decrypt its message.

In 2015 it dropped support for the SMS and merged with another company to make the current version “Signal” which only uses Data transferred messages to keep messages off servers, and invisible to carriers.

Signal, Renamed Text Secure has different features available; it still uses end to end encryption without the use of servers to keep data as secure as possible but also allows for audio and video calls as well as group messages.
Signal also allows the use of a passphase which encrypts the local message database and the user's encryption keys.

Project ASM will be using some similar key points that are currently implemented with Signal, end to end encryption. This will allow us to keep the messages secure even if the encrypted message is stored on an unknown server or lurking eavesdroppers.

Pros:

- High end to end encryption
- Feature rich
- Cross platform compatibility
- Passphase local encryption

Cons:

- Needs the app
- Needs a data connection
- released source code on GitHub

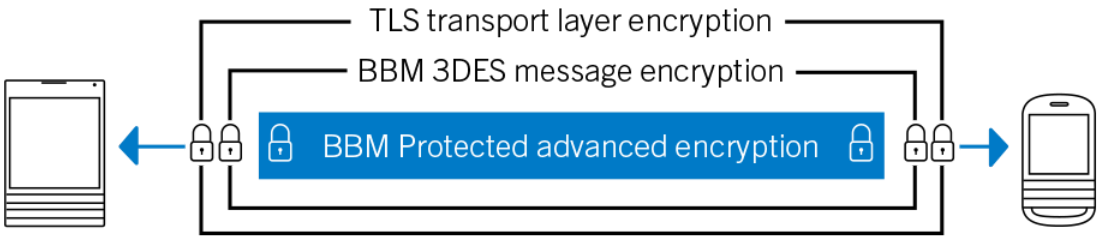
Blackberry Messenger

Background:
Blackberry Messenger is probably the most recognised secure messaging service in the world. It is a paid service which you have to register for so it is not anonymous for users but the messages within are highly secure.

Project ASM is also not fully anonymous as it requires a mobile network subscription or pre-paid service which requires ID and registration for purchase. It also tracks the usage, time stamps and number of messages that have been sent though it's service number, although this is not a big deal it requires some further thinking and may require more in depth encryption.

Pros:

- Enterprise grade secure messaging service.
- This uses 3 layers of encryption between the sender and receiver of a message.
- Each message uses a random symmetric key
- Triple DES 168 Bit scrambling
- TLS transfer layer encryption to prevent eavesdropping and manipulation of messages
- On Device AES local message encryption, Data is not stored on servers



Cons:

- Mobile data connection required
- Needs the app
- Monthly subscription fee

Medical secure wireless monitoring

Background:
The paper (Fang et al, 2012) deals with the creation of an extensible health monitoring device which uses wireless and 3G network to send data. This device sends its data to a given workstation monitored by a doctor. Fang et al.'s main motivation for the creation of this device

was to create a compact system which could integrate several sensors without the need of a PDA or mobile phone.

The device is comprised of a Samsung S3C2440 microprocessor, 256mb NAND flash memory, 2mb NOR flash memory, 64mb SDRAM, and a JTAG interface. They used a custom build of the the linux kernel (2.6.32.2) to configure the sensor drivers and hardware interactions. Overall Fang et al. were able to successfully implement a device that could communicate over a network using custom hardware and software.

Pros:

- This system itself provides us with a potential application for our project creating a device which could encrypt medical sensor data or potentially be integrated into a system like this.
- It provides an interface to the user through button controls to turn the device on/off, network configuration and sensor functions.
- It is extensible and flexible to allow for different input device configurations

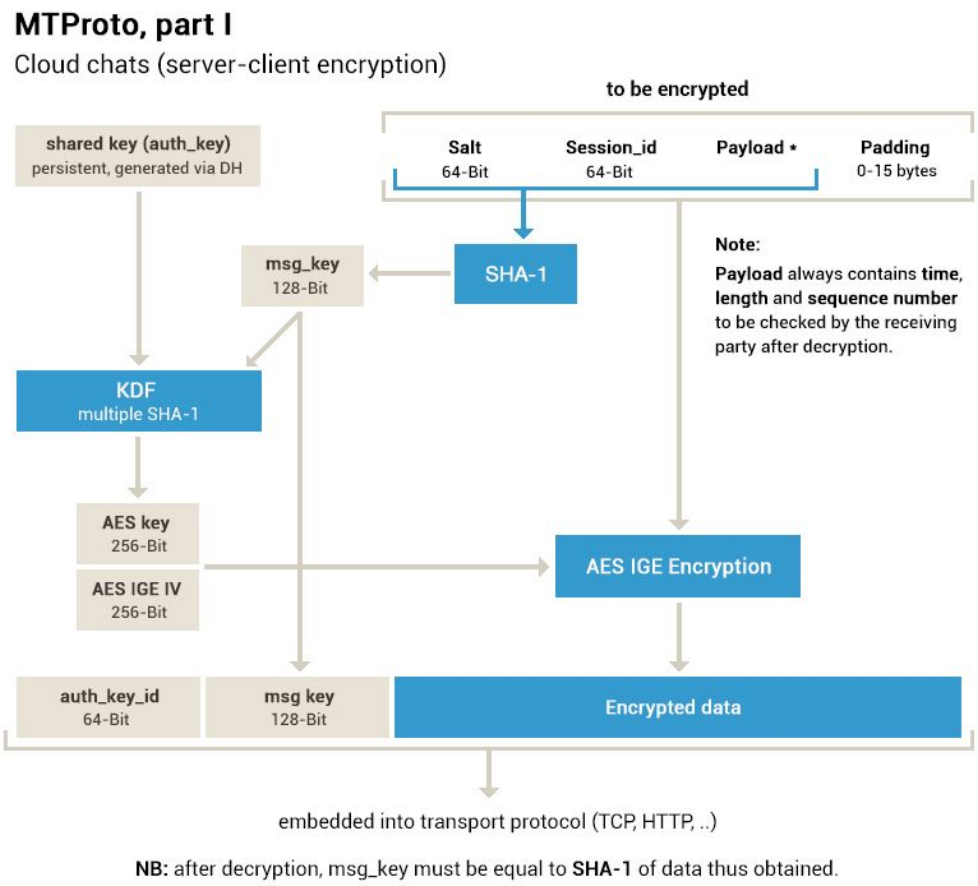
Cons:

- The paper didn't consider any sort of security measure.
- They've used different resources to us because of their scope.
- It doesn't help us in implementing our own device because they used different hardware and didn't consider any security.

Telegram

Background:
Telegram is an internet-based messaging and group chat service with a focus on security, and is available on every major desktop and mobile operating system and as a web application.

It uses a layered encryption method consisting of SHA-1, IGE and AES. Secret chats (those that occur directly between devices, bypassing Telegram's servers) utilize an additional layer of encryption on top of these (Telegram, 2016). Telegram acknowledges they use older, weaker encryption algorithms, but that they've taken steps to address the flaws in the chosen algorithms (Telegram, 2016). Several pieces of information must be decrypted and verified before the service can confirm that the message was transmitted securely, including the server salt, session ID, message length and time.



(Telegram, 2016)

Pros:

- Multi-layered encryption with the option of an end-to-end "secret chat" bypassing the Telegram server.
- Open well-documented protocols

Cons:

- Requires an app, or time spent on reimplementing the service for our own purposes
- "Secret chats" disappear if the user changes devices.

Methodology

Choosing our methodology, we are looking to find a method which can allow us to develop a solution quickly whilst giving us the option to modify the features of the device during development if need be. Since no one in our group has had experience with Arduino we need a methodology that can allow us to complete deliverables whilst learning the system.

Our goal is to develop an Arduino device which can send/receive encrypted messages from mobile phones. In order to achieve this we will need to develop software that interacts with the Arduino and a couple of Arduino modules. Depending on the outcome of developing this device, there is also a possibility of further extending the device to act as a gateway to pass encrypted messages from one phone to another.

Because of our unfamiliarity with Arduino hardware and development environment we have devised a development process to incrementally add functions to device. This way we can ensure that we have a working device early on, be able to test the device at each increment and receive feedback from Jemal.

So, the deliverables/milestones for this process are:

- Research & Investigation
- Procure hardware
- Assemble hardware
- Get the Arduino running
- Get the Arduino interact with the components
    - Display text on LCD screen
    - Connect GSM module to the network
    - Send a SMS from the GSM module
    - Receive a SMS to the GSM module
- Implement cryptography libraries
- Send encrypted SMS to a phone
- Receive an encrypted SMS from a phone
- Test

With our plan there are several methodology paradigms we can choose from:

Incremental Build

This model provides “progressive development” for software. It builds upon the waterfall method allowing incremental releases of functionality to the software. This means the requirements for the project can be broken down to subsets of task to be completed.(Schwalbe, 2014, pg 60-70)

This model does meet the criteria of adding functionality incrementally as well as breaking down tasks. However planning is an important component to the method, and since there is a level of open endedness in our requirements we may potentially struggle adopting a method using this model.

Pros:

- Flexible due to constant evolution
- Lower initial costs in time and scope
- Be able to stage priorities

Cons:

- Planning and design is crucial
- High long term cost
- Still has elements of linearity (ie. waterfall)

Spiral

Spiral is an iterative approach to the waterfall model. It incorporates the phases of the waterfall through an iterative process which allowing for changes to be added to the project. (Schwalbe, 2014, pg 60)

The spiral model will allow us to to include changes within the project requirements during each iteration of the the analysis and design phases, however the time between adding changes to the project may mean that certain components may not be added in time. We don’t have a lot of time. Furthermore, since spiral uses the phases of the waterfall method we would need to tightly manage ourselves through each phase, which may not be possible.

Pros:

- Risk Analysis takes high priority
- Used in large missions critical projects
- Better suited for long term projects

Cons:

- Costly, time delays in adding changes to the requirements
- Risk analysis isn't critical to our project
- Changes generally occur at the end of the life cycle
- Still has elements of linearity (ie. waterfall)
- Not suited for small projects

Agile

Agile is a model which allows for close collaboration between developers and clients. It incorporates iterative and incremental methodologies to be able to evolve solutions through collaboration (Schwalbe, 2014, pg 61, 69)

The core values of agile software development suit our desired approach to developing the project (Agilemanifesto.org, 2016):

- Individuals and interactions over processes and tools. This allows us to be self-organised and not restricted to a rigid structure like waterfall. Since we all have other commitments we need a method that focuses more on collaboration.
- Working software over comprehensive documentation. We have less than 12 weeks to complete the project so we want to create a functioning device, otherwise it will be a waste of time.
- Customer collaboration over contract negotiation. Since we know little about Arduino, collaborating with Jemal is important for us to learn as much as we can. Also if we get stuck we need Jemal will be able to point us in the right direction.
- Responding to change over following a plan. There is room for us to extend our device and given the ambiguity of the requirements changes will occur.

Pros:

- Allows for change
- Delivering results and updates frequently
- Better suited to short to medium term projects
- Close collaboration between developers and clients

Cons:

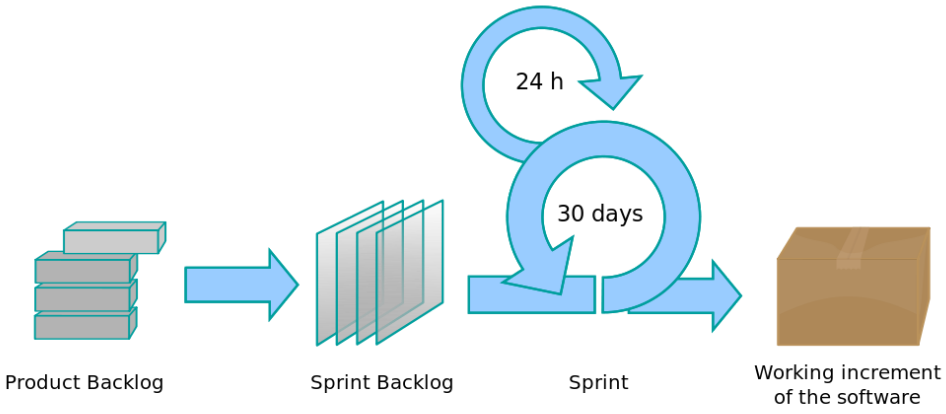
- Lack of designing and documentation
- Changes can let the project go on a tangent
- No thorough planning

The methodology we are choosing to adopt is an agile approach using the Scrum method. We have chosen this approach because the there could be changes during development, we need feedback to ensure we are on the right track, and Scrum is simple to understand and not as rigid as other methods. Furthermore this method can be adopted by all roles within the project.

Scrum

The Scrum method feeds upon these values and creates a framework that help us in completing this project. The workflow of Scrum is rather straightforward as mentioned by (Schwalbe, 2014, pg 70) for us this means:

- Jemal creates a list of tasks called the product backlog. These are the project requirements and the To Do list on Trello.
- We will begin sprint planning by choosing a portion of the tasks to implement from the product backlog. These chosen tasks form the sprint backlog which is our Doing list on Trello.
- Then we will allocate a week or two depending on the complexity of the task to complete the sprint backlog, with daily meetings (daily scrum) to assess progress.
- At the end of the sprint the new functionality to the Arduino should be working. We will then review the sprint and add any issues to the product backlog.
- And the process repeats when the we choose a new set of tasks from the product backlog.



(Lakeworks, 2009)

Technology & Resources

Programming Resources

Arduino IDE
Development software used to program the Arduino board. This is a free downloadable software. It uses coding language similar to C / C++.
http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf

Arduino Tutorial
Arduino Tutorial page which allows us to look through forums of examples of projects used guide us through sections of our project. Our project expands on multiple tutorials including SMS, Encryption, LCD Display, Memory Management and more.
https://www.arduino.cc/en/Guide/ArduinoGSMShield

Arduino References
Arduino References gives us access to libraries of code that may be useful to our project, this will expand as we go along and find different requirements we may use. Currently we have AES and SD memory management libraries listed.
https://www.arduino.cc/en/Guide/ArduinoGSMShield
https://www.arduino.cc/en/Reference/SD
https://www.arduino.cc/en/Reference/GSM

Setup Tutorial
Tronixstuff has compiled a tutorial on other methods of the GSM shield to be used in different ways, this includes making and receiving phone calls from the device, auto phone dialers, auto sms senders and other tweaks. Simpasture has a SMS setup process for mapping the pins for the Arduino mega and a walk through on getting the sms to send a message.
http://tronixstuff.com/2014/01/08/tutorial-arduino-and-sim900-gsm-modules/
http://www.simpasture.com/36023.html

Cryptography Resources

Encryption Libraries
This is a large list of encryption tools that may be useful for our encryption of a message
This includes, AES Library for 128, 192, 256 bit keys:
http://utter.chaos.org.uk:/~markt/AES-library.zip
https://github.com/Cathedrow/Cryptosuite
https://rweather.github.io/arduinolibs/crypto.html

Cryptography Books
These books provide the mathematical background to popular encryption methods. These may not be extensively used.

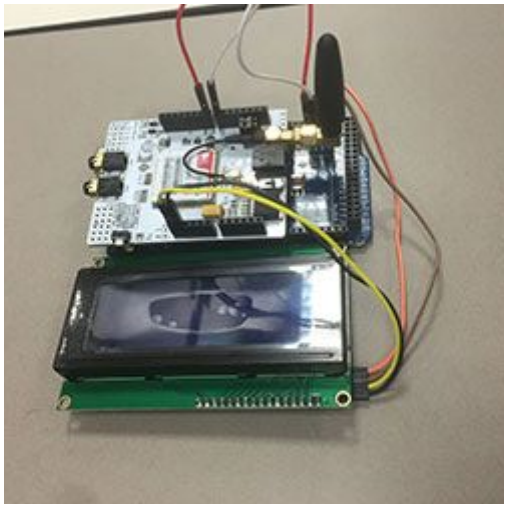
- Introduction to Cryptography - Wade Trappe & Lawrence Washington
- Public Key Cryptography - Lynn Batten

Tanzir Ismat Thesis
Thesis conducted by Tanzir Ismat provides a library that was used for his study of Arduino encryption. This may be relevant to embed the encryption to the code. (Ismat, 2015)
http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/4889/10101016.pdf?sequence=1&isAllowed=y

Priyansha Gupta Thesis
This thesis was conducted by Priyansha Gupta, where the goal was to implement security in a personal device. The encryption method used to protect the data is AES. Once again this could prove to be relevant in order to the embed the encryption to the code. (Gupta, 2013)
http://www.lasr.cs.ucla.edu/psd/AES_on_arduino.pdf

Hardware Resources

- Arduino, MEGA 2560 board
    - Arduino compatible LCD screen 20 character 4 lines with IIC 4 pin serial input
    - Arduino GSM/GPRS Shield.
    - Power Supply
    - Micro-SD card
    - USB a/b
    - Connected Windows/Linux/ MAC Computer
- Jaycar to procure hardware resources.
- Youtube for tutorial videos
- Online tutorials



Documentation & Research resources

- Deakin Library for access to papers and books on cryptography
- Trello to manage our work and scrum sprints
- Google Docs to record all information
- Internet based research

Conclusion

Throughout the duration of our investigation we have discovered many existing systems that build on a variety of platforms. These systems use distinctive features that differentiate from each other. They have been build for different purposes, eg: personal, business or professional services (including medical and emergency). Our group found our nieche area which differentiates us from other services using the Arduino boards to send messages over Telco SMS services with end to end encryption. Evidence based research on Arduino systems gave us the background knowledge on the software, hardware and existing systems such as, SMS setup and encryption tools (AES and public key RSA) which will assist in achieving our end goal. This goal will be completed through utilizing the agile method which allows the team to be self-organised and acknowledge any changes made throughout the project. As a team we are confident that we will complete all tasks productively and efficiently.

References

Scrumguides.org 2016, “Scrum Guide”, retrieved 18 March 2016, <http://www.scrumguides.org/scrum-guide.html>.

Lakeworks 2009, “Scrum process - Wikimedia commons”, retrieved 19 March 2016, <https://commons.wikimedia.org/w/index.php?curid=3526338>.

Schwalbe K 2014, Information Technology Project Management , 7th edn, Course Technology, Boston, USA.

Agilemanifesto.org 2016, “Manifesto for Agile Software Development” retrieved 18 March 2016, <http://agilemanifesto.org/>.

Telegram Messenger LLP 2016, “Telegram Messenger”, retrieved 1 Apr 2016, <https://www.telegram.org/>

Telegram Messenger LLP 2016, “MTProto, part I”, retrieved 1 Apr 2016, <https://core.telegram.org/file/811140187/1/sfBQV3Trp80/3a3c48bad836b853ed>

Telegram Messenger LLP 2016, “Telegram F.A.Q”.,retrieved 1 Apr 2016, <https://telegram.org/faq#q-so-how-do-you-encrypt-data>

Telegram Messenger LLP 2016, “FAQ for the Technically Inclined”, retrieved 1 Apr 2016, <https://core.telegram.org/techfaq>

Telegram Messenger LLP 2016, “Secret chats, end-to-end encryption”, retrieved 1 Apr 2016, <https://core.telegram.org/api/end-to-end>

Telegram Messenger LLP 2016, MTProto Mobile Protocol, retrieved 1 Apr 2016, <https://core.telegram.org/mtproto>

Blackberry Messenger, Blackberry Enterprises, retrieved 1 Apr 2016, <http://us.blackberry.com/enterprise/products/bbm-protected.html>

Text Secure, Wikipedia, retrieved 1 Apr 2016, <https://en.wikipedia.org/wiki/TextSecure>

Signal, Open Whisper Systems, retrieved 2 Apr 2016, <https://whispersystems.org>

AESEncryption 2016, “What is AES encryption?” retrieved 4th April 2016, <http://aesencryption.net/>.

Gupta P 2013, “Installing AES on Arduino”, Masters thesis, retrieved 6th April 2016, <http://www.lasr.cs.ucla.edu/psd/AES_on_arduino.pdf>.

Ismat T 2015, “‘Comparative Analysis of AES Algorithms and Implementation of AES in Arduino, Honours thesis, retrieved 9th April 2016, <http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/4889/10101016.pdf?sequence=1&isAllowed=y>

Weber J 2015, “Project Cuckoo” retrieved 24th March 2016, <http://jochenmariaweber.de/cuckoo/cuckoo.html>

Visnjic F 2015, “Cuckoo - Encrypted Communication using public social networks” retrieved 24th March 2016, <http://www.creativeapplications.net/arduino-2/cuckoo-encrypted-communication-using-public-social-networks/>

Fang X, Liao J, Zhang R, Ni P, Li B, Meng M “An extensible embedded terminal platform for wireless telemonitoring”, International Conference on Information and Automation (ICIA), Shenyang, pp. 668-673, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6246761&isnumber=6246755&url=http%3A%2F%2Fieeexplore.ieee.org%2Fielx5%2F6229821%2F6246755%2F06246761.pdf%3Ftp%3D%26arnumber%3D6246761%26isnumber%3D6246755>