Funktioneers

Michael Robinson, Eden Clements

Darrel Smith, Corbin Jackson

Senior Design 11/19/14

Project Design Analysis

As a group we sat down a came up with several designs for our project. After comparing them we found some differences and similarities in our ideas. We continued to think and compare ideas both old and new. We then needed a method of comparing our ideas in a systematic manner. There were several key elements of our design that needed comparison as to what would be more effective for our project. Following these comparisons will be the working model of our design

The retrofitting of the electronic lock.

We choose an embedded lock design because there are so many retrofit electronic lock designs already on the market (Lockitron is a company that specializes in this area). The main issue seen with Lockitron and retrofitted locking mechanisms is that having all electronic components outside of the lock creates a vulnerability in security.

RetroFit Lock Design Matrix

	Ease of Installation	Security	Durability
Retrofitted Lock	8	7	6
Embedded Electronic Lock	5	9	8

When it came down to how we were going to power the lock we decided that a bettery would work. The question came as to what the type of battery that was going to power the lock.

- 9 volt battery

The 9 volt battery that was previously used in planned designs for the project was dropped due to new design ideas and constraints. The 9 volt battery is what is called a primary battery which changes chemical energy into electrical energy. As you have at some

point in your life encountered when all of the chemical energy has be depleted the battery is no longer of any use and must be replaced. Due to the nature of the project this would provide a limited functional period for the lock and also place security concerns upon the consumer in the terms of security anxiety.

-Cell phone battery (Lithium ion)

During the conceptual redesign phase the team decided to implement the use of a 2600mAh battery. This battery was chosen to due to main reasons its ability to be recharged through a micro-usb port, also it proven reliability and longevity in modern cell phones. When doing research it has been seen that with a full charge and left in standby mode this battery can maintain a charge for upwards of 30 days. The battery we plan to use is a lithium ion battery which is classified a secondary battery due to its rechargeable nature. This battery was also chosen due to it durability in ideal conditions the battery will most likely outlast the components in which they are placed.

Type of Battery	Ease of Implementation	Length of Charge	Environment Impact	Cost Effectiveness
Cell Phone	7	10	10	9
9 Volt	9	5	6	5

We choose a cell phone battery because its more cost efficient, durable, and rechargeable.

The type of microcontroller (board) that we were going to use.

-Ardunio vs. Raspberry Pi

The use of which board was to be used in the completion of this design was the topic of much debate with it being one of the only constraints posed by the cornell cup competition board. This is what began the teams search for the proper tool to accomplish our particular job. We came to the comparison of the ardunio uno vs the raspberry pi model a+. When comparing these two chips it must be understood what each is. The ardunio is just a microcontroller which is only on part of a computer as whole, while the raspberry pi is a fully self contained computer with all the trimmings. When comparing the spec sheet of each in contrast with the tools needed for the design to function the raspberry pi was an easy winner due to its easy accessibility to those with limited coding knowledge, its inclusion of a micro usb charging port, and also having a significantly higher number of inputs and outputs over the ardunio.

	Number of Inputs	Access to Preexisting Code	Ease of Conversion to GSM	Amount of Memory

Raspberry Pi	8	10	7	10
Ardunio	5	10	7	5

The specific type of lock design we were going to choose.

-Single cylinder deadbolt

A single cylinder deadlock will accept a key on one side of the lock, but is operated by a twist knob on the other side. Most common of the locks. A deadbolt cannot be moved to the open position except by rotating the lock cylinder with the key. Most single cylinder locks have a key hole on one side and a twisting knob on the other side. A more secure but less safe option to a single cylinder lock is a double cylinder, which would have a key hole on both sides.

-Jimmy proof deadbolt

Jimmy proof locks can also be single or double cylinder but have a different shape and arent located inside the door. The lock mechanism is enclosed in the metal casing that is the lock. With this lock there are vertical bolts that slide into place and out of place as the lock is turned. This makes the lock a lot more secure. It prevents the lock from being opened by someone not wanted. The only part on the outside is the key hole but on the inside is an oval shaped metal casing that connects the door with the door frame.

-Tumbler locks

The pin tumbler is commonly used in cylinder locks. In this type of lock, an outer casing has a cylindrical hole in which the *plug* is housed. To open the lock, the plug must rotate. When the plug and outer casing are assembled, the pins are pushed down into the plug by the springs. The point where the plug and cylinder meet is called the *shear point.* With a key properly cut and inserted into the groove on the end of the plug, the pins will rise causing them to align exactly at the shear point. This allows the plug to rotate, thus opening the lock. When the key is not in the lock, the pins straddle the shear point, preventing the plug from rotating.

We didn't want to use the tumbler lock because it naturally is easier to break into than a single cylinder or Jimmy proof lock. We chose the single cylinder because we believe that the mechanical parts will be easier to manipulate than the Jimmy proof. Yet, the Jimmy proof is the most secure of them all which is most ideal, but harder to manipulate.

	Ease of installation	Accessibility	Security	Total
Single Cylinder	8	10	8	26
Jimmy Proof	7	10	9	26

Tumbler	6	10	6	22

How we were going to implement the actual encryption.

- Vocoders versus

- AES encryption

We didn't need to just choose one method of encryption because both will obtain a high level of security. Although there are many options in the world for security not all of them must be the sole providers of security, in fact to be more secure someitmes it is smarter to have two different systems to increase overall security. We chose that although distinctly different there was no reason to chose one method over another. When it came down to which specific cypher to use for the AES encryption we made an encryption matrix comparing the different types.

Encryption Matrix

Type of Encryption	Cost	Time for Encryption to be broken	Level of Security
AES-256	$0	10	10
AES-128	$0	8	9

Rating Scale

0-4 Poor

5-7 Fair

8-9 Good

10 Very Good

Our Current Design:

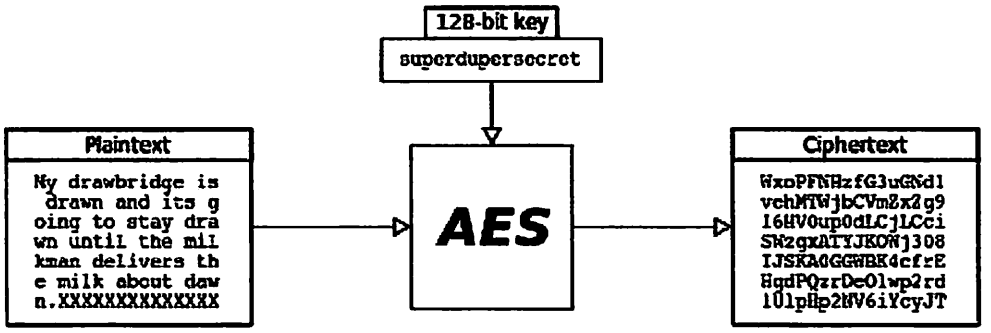
In Figure A you can see the steel bar (1) which is a part of the locking mechanism (19). Unit is installed into the door with a normal key entry as an option(2). The consumers smartphone(10) would act as a key Via voice recognition. The signal (5) would be transmitted over the GSM network (6) and would be received by the module on the board(13). The lock would have its own dedicated number given to it by the simcard (14). The signal is relayed to the Raspberry Pi controller(12). The controller activates the motor(16) which turns a cog(17) this cog turns a larger cog(18) which would pull the bolt of the lock (1) back into the door thus opening the lock. When the bolt moved into a locked position a sensor(20) will be triggered and

a signal(4) will be sent to the phone over the GSM(6) network to the phone(10) letting the user know the door is unlocked(9). Installed in both the lock and the phone app will be an encryption(201) that secures the messages sent over the GSM network(6). There will also be customizable options for extra encryption settings(8) in the app such as a vocoder(11) that will transmit the signal at different higher frequencies.

AES Encryption Research

AES is an cryptographic cipher and it's based on the Rijndael cipher.. It runs on a 4x4 matrix of bytes and the key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, into the final output. The number of cycles of repetition are : 10 cycles of repetition for 128-bit keys (201), 12 cycles of repetition for 192-bit keys,14 cycles of repetition for 256-bit keys. Each of these cycles consist of several steps, and they each contain four similar but different stages. Then, a set of reverse rounds are performed to transform the encrypted message back into the original plaintext using the same encryption key. The diagram below provides a simplified illustration of AES-128 byte encryption. The code must generate a key that's going to be used for the encryption. (202) Then the encryption is implemented. (203) There must also be a function that calls for the cipher being used. (204)

(These figures of code are example Java code from a group members internship in Thailand dealing with AES)



(201)

(202)

```
public static byte[] getRawKey(byte[] seed, String callingalgo) throws Exception {
                    KeyGenerator kgen = KeyGenerator.getInstance(callingalgo);
                    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG"); //callingago is a variable which
i used to set the string for the list of algorithms
                    sr.setSeed(seed);
                    kgen.init(128,sr);
                    SecretKey skey = kgen.generateKey(); //generates the password that is going to be used for the
AES algorithmic encryption
                    byte[] raw = skey.getEncoded();
                    return raw;

        }
```

(203)

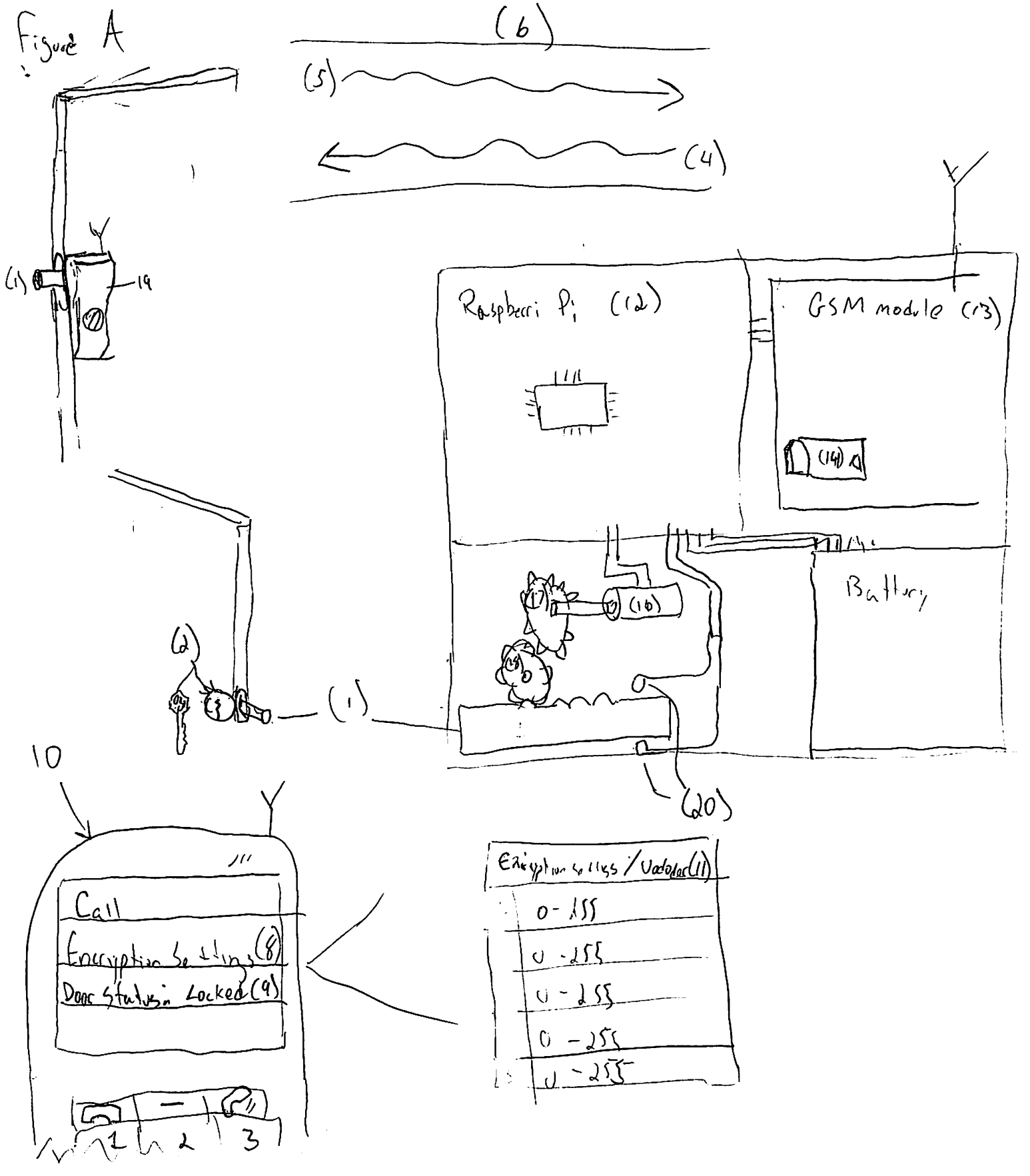
```
private static byte[] encrypt(byte[] raw, byte[] clear, String callingalgo) throws Exception {
                    SecretKeySpec skeySpec = new SecretKeySpec (raw, callingalgo);
                    Cipher cipher = Cipher.getInstance(callingalgo);
                    cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
                    byte[] encrypted = cipher.doFinal(clear);
                    return encrypted; //method that actually encrypted the data
                    private void dropperboxfunction(){ //function for the drop down box
          dropperbox = (Spinner) findViewByld (R.id.dropdownbox1); //links variable to actual widget
```

(204)

```
String[] algolist = {
                "AES","DESEDE", "BLowfish", "ARC4"      //CREATES AN ARRAY THAT MAKES THE LIST
FOR DROP BOX
   };

                    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,
                                android.R.layout.simple_spinner_item, algolist);
                    dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
                    dropperbox.setAdapter(dataAdapter);
   }
```

Figure A


(6)
(5)
(4)
(1)
19
Raspberri Pi (12)
GSM module (13)
(14)
Battery
(17)
(16)
(15)
(2)
(3)
(1)
(20)
10
Call
Encryption Settings (8)
Door Status: Locked (9)
1
2
3
Encryption settings / Voltage (11)
0 - 255
0 - 255
0 - 255
0 - 255
0 - 255