Encrypted Communication

Shane Wechsler, Krish Patel, Jacel Evangelista, Jackson McCoy, Enoch Chigbu, Paul Nusser, Brandon Skeens

Motivation

The goal for this project is to create two arduino boards in communication with one another and utilizing encryption and decryption to send and protect messages. This struck us as a very unique and interesting project as it is similar in scope to military applications.

Encrypted radios go back all the way to the second world war, and the field of computing was of course directly inspired by code breaking efforts. These pitted computing machines against the enigma machines and the failure of axis powers to protect their communications when compared to allied nations gave us a powerful boost to the war effort, showing the potential power of this technology.

In tandem with the unique structure of this project, the historical background makes it a fascinating exploration, so we were very interested in seeing if we could create our own, small-scale digital encryption and decryption to simulate encrypted communications.

Also, this project is super cool, unique, and interesting.

Related Works

- AES Encryption between two Arduino Boards
  - Symmetric encryption to communicate
  - Advantages: Easy implementation (AESLib/ArduinoCrypt)
  - Disadvantages: Potential security compromise if key is accessed by third-party
- Wireless Sensor Networks (WSN) with Encryption
  - Communication with multiple (2+) devices using symmetric encryption and multiple sensor nodes
  - Advantages: Scalable, practical
  - Disadvantages: Latency
- RSA (Rivest–Shamir–Adleman) Encryption for communication between two devices
  - **Asymmetric encryption**, different keys for encryption and decryption
  - Advantages: Strong Security, more suitable for complex networks
  - Disadvantages: Slower, inefficient with large dataAES Encryption between two Arduino Boards
    - Symmetric encryption to communicate
    - Advantages: Easy implementation (AESLib/ArduinoCrypt)
    - Disadvantages: Potential security compromise if key is accessed by third-part

Our Method

"expa"	"nd 3"	"2-by"	"te k"
Key	Key	Key	Key
Key	Key	Key	Key
Counter	Counter	Nonce	Nonce

ChaCha Encryption Benefits

- Naming: ChaCha algorithm is derived from the Salsa algorithm, which is why it has this strange name. It’s also related to the Rumba algorithm.
- ChaCha improves both performance and diffusion (distributing plaintext randomly over the cipher) when compared to Salsa
- ChaCha is **not** patented which makes it ideal
- ChaCha Algorithm
  - Pseudorandom add-rotate-xor (ARX)
  - Maps a 256 bit key, 64 bit nonce*, 64 bit counter to a 512 bit block of the key string
  - If two messages are the exact same aside from 1 single bit, on average the two ChaCha outputs (encrypted messages) will have a difference of 12.5 bits.
  - ChaCha is significantly faster than the more common AES on mobile devices, in particular
- Nonces are randomly generated in order to prevent the message from being decrypted even if the key is guessed

Source: https://en.wikipedia.org/wiki/Salsa20#ChaCha_variant (also we looked at the algorithm and can confirm it actually does this stuff)

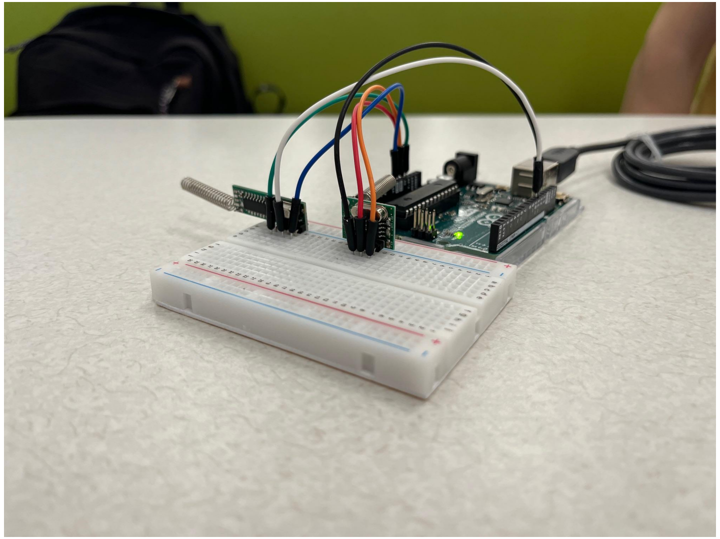
Intellectual Merit

- Cha Cha Cryptography
  - Modern stream cypher that is faster than AES which is more common
  - Single Key Approach which is efficient and better for low-power storage devices such as the arduino uno
  - Randomly Generated Nonce increases security which is better for our encryption/decryption
- Accomplishments:
  - Designed and implemented a successful encrypted communication system using arduino boards and Cha Cha Encryption.
  - Designing a system to generate a key to exchange between both radio transmitters and communicators
  - Unique use of hardware through 2 arduinos and the extension of range with antennas
  - Up to 150 Feet of free back and forth transmission

Broader Impacts

- As with many forms of encryption, safety and security are main considerations
- Considerations for law enforcement
    - Police scanners/radio
    - Protect sensitive information during sensitive situations
- Applications with emergency response
    - HIPAA for hospitals
    - Emergency response teams use of secure radio systems
- Personal privacy
    - Protect personal data from unwanted eyes

Technical Content

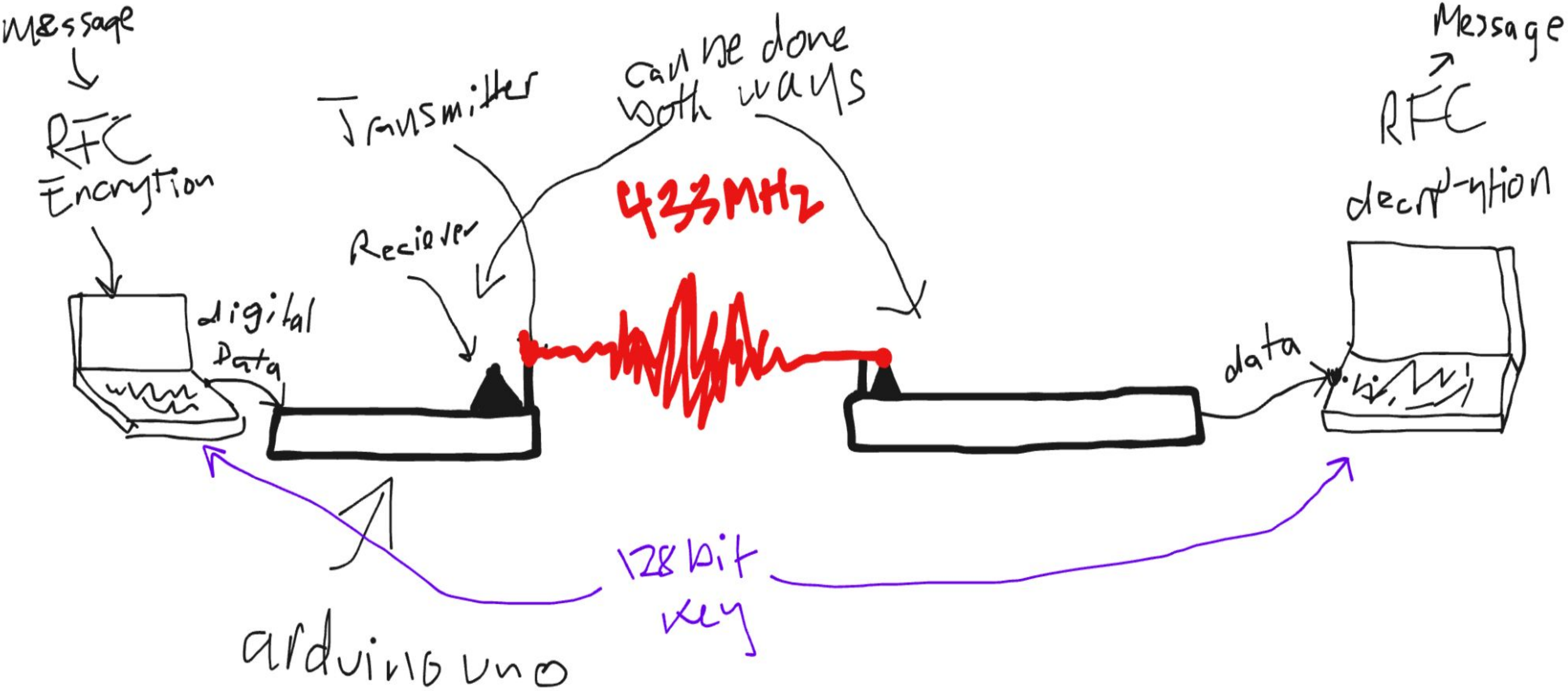


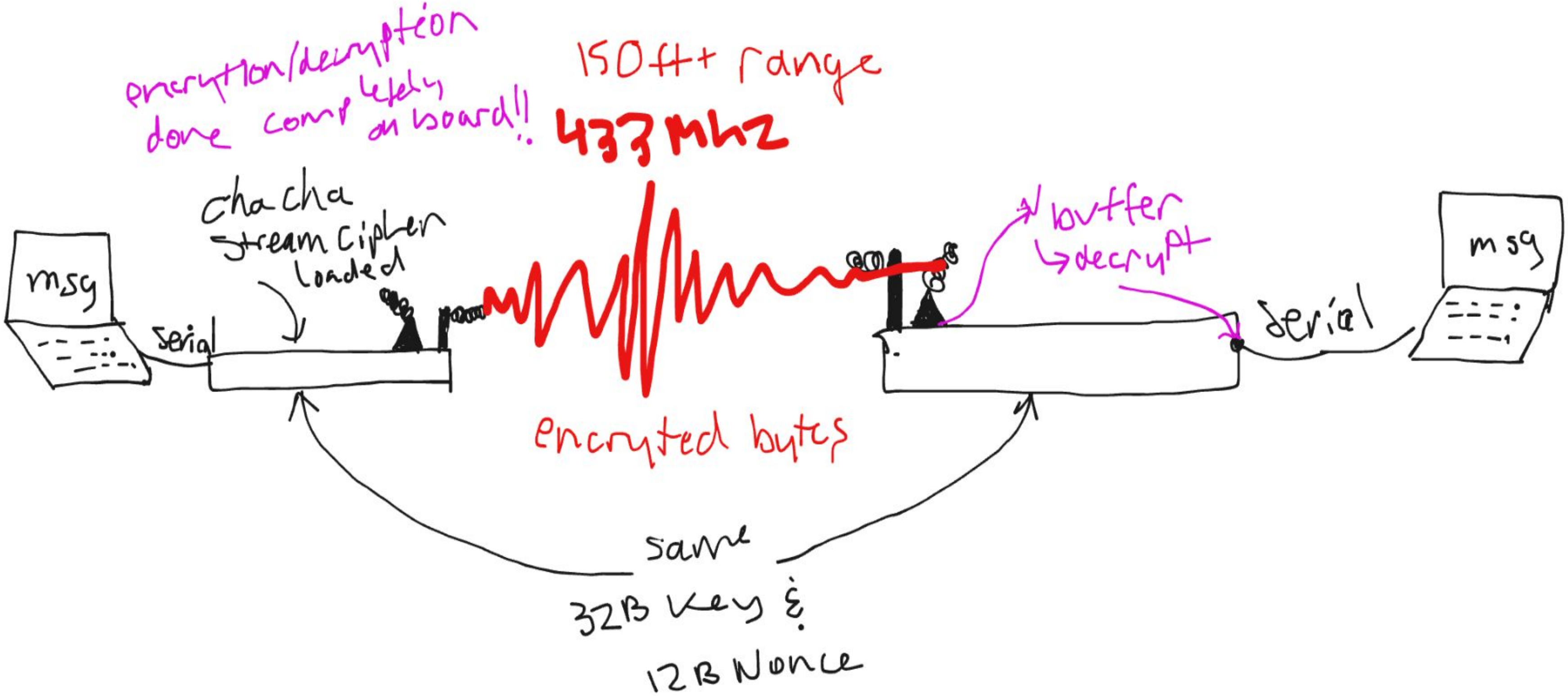
Hardware:

- 2 arduino boards
- 2 433Mhz radio transmitter/receiver sets(updated/advanced version) with soldered antennas.

Software:

- Arduino radio data stream drivers
- ChaCha cryptography library(arduino capable and lightweight)


Message
RFC
Encrytion
Transmitter
can be done
both ways
Message
RFC
decryption
433MHz
Reciever
digital
Data
data
128 bit
key
arduino uno


encrytion/decryption done completely on board!!
150ft+ range
433 Mhz
chacha stream cipher loaded
msg
serial
encryted bytes
buffer
↳decrypt
serial
msg
same 32B Key & 12B Nonce

Testing

Base Transmission: 80ms

Sending Message: ~75ms

IE:

1 Char - 79 milliseconds

2 Char - 85 milliseconds

3 Char - 90 milliseconds

Results and Discussion

The result of this project was a total success. As mentioned above, the resulting product was significantly more lightweight than the original plan. Although the transmitter was not able to penetrate the thicker walls we tested with, a line of sight approach worked at distances of at least 150 feet, showing that the technology is not just a simple gimmick to send messages while right next to one another. The current design takes messages of up to 128 bytes, but it is conceivable that we could modify the design to split longer messages up and send them as multiple messages.

Communication Example

```
Message: crazy stream cipher

Message: lmao

Message: kinda wack

Me: received your messages

Encrypted Text:

#IE_It���ZL��*�S��.\�{7�V��ϒ��5

Me: top secret information has been conveyed

Encrypted Text:

%CVSg�����F��4�F��3v�ŸFC\T.��������w��Q������2��`)|Ń�
{��w|w�����N�M�����~�;��/��?���z���Q���W��
�W;Z���~��o�r�8~���sS��y������V8ф�3������D�����7>A����|
����z�u�w���[�L���}��]
���q��  9��q2p��������|����  s���/Z��|O�3Y���#Y.Jy�}
���O����S□□�Z�e�������U���□����iy��  �^���nI�9�n����7���y��
s□��o����V����m����^�c�]{ᴗm��3n�†_�����r<f<���G.s1#~S���}w�

Message: very very strange
```

Note- Message: indicates something sent to me, while Me: and Encrypted Text: are from the messages I sent out

While in this example, the longer message had a longer encrypted message, this is not always the case, which is important for security

Team Roles

Shane: Project lead, radio, hardware, presentation, and testing

Jackson: Radio, hardware, and presentation

Enoch, Krish, and Paul: Encryption and decryption code (ChaCha algorithm)

Jacel: Video and Testing

Brandon: Summary