

AES Encryption/Decryption (Hardware)

Dayton Flores
Joseph Sharp Halpin
Advisor: Dr. Ming Zhu

Introduction:

AES (Advanced Encryption Standard) is a subset of the Rijndael block cipher, and it was established by the U.S. NIST (National Institute of Standards and Technology) in 2001. This algorithm can work with differing key lengths of 128, 192, or 256 bits. It is a symmetric-key algorithm, meaning it uses the same key for both encryption and decryption. AES replaced DES (Data Encryption Standard) as the NSA-approved federal government standard in 2002.

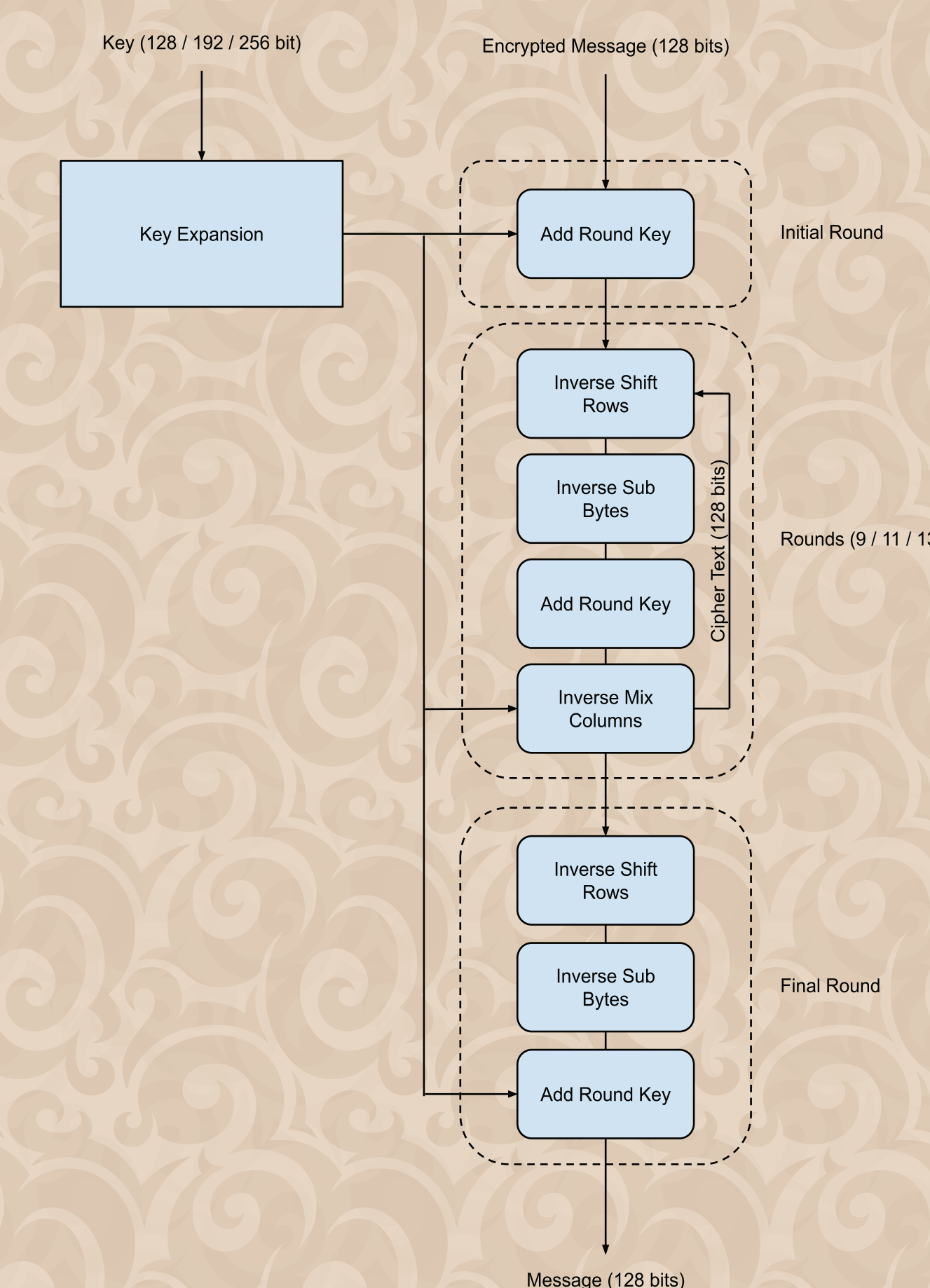
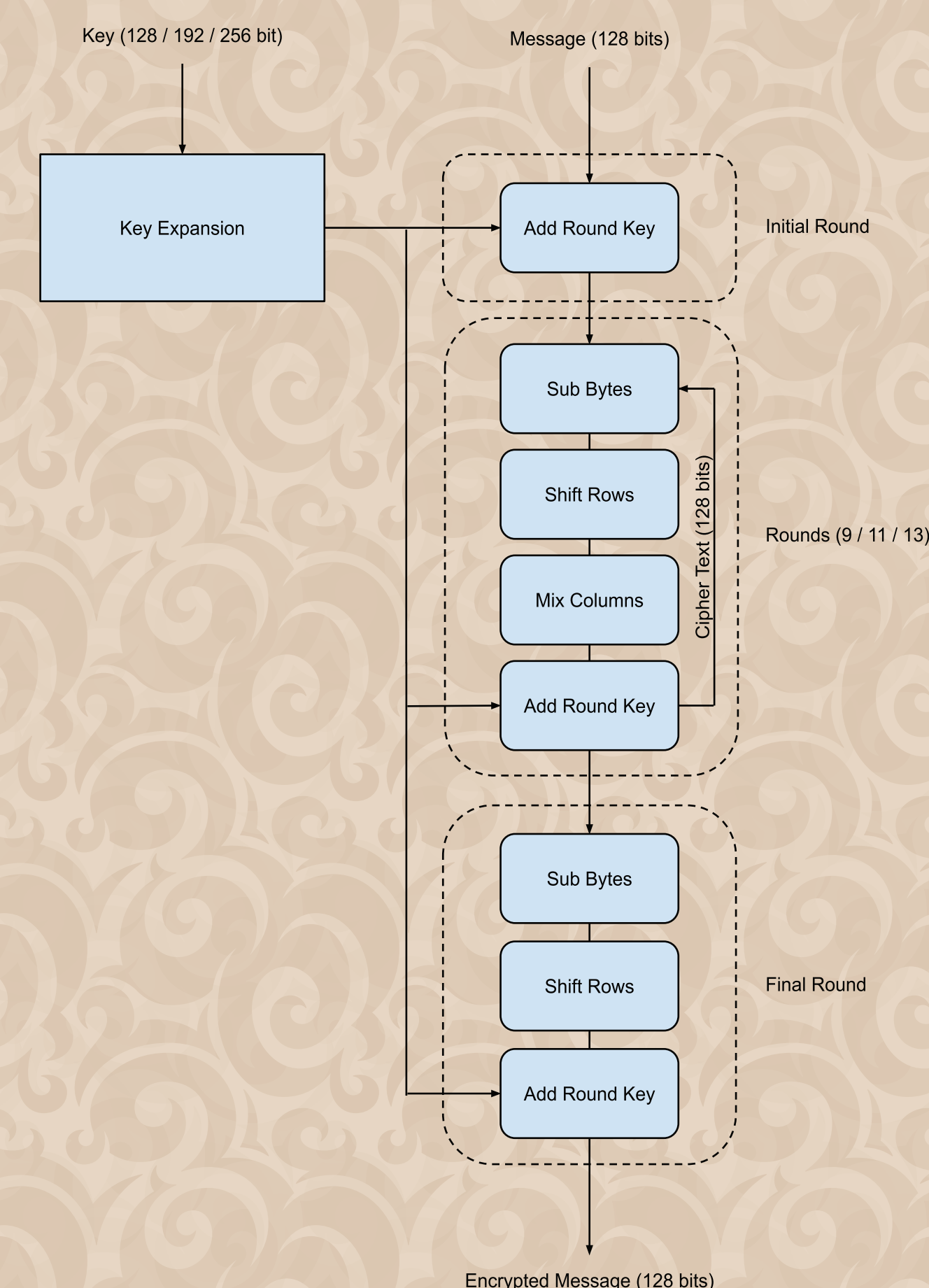
Motivation:

AES is traditionally applied in software due to its ease of implementation. Administering in hardware requires complex analysis, but is beneficial due to the following: more efficient (AES module is independent of CPU), much safer (key is built directly into hardware), and substantially faster (with a fast enough clock speed).

Monetization:

If converted into an IC, this synthesis could be mass produced using an RNG (random number generator) for each key & sold for ≈ \$1 each (based on current market).

AES Algorithm:



Encrypt/Decrypt:

The user gives the AES module three inputs: key length (128 / 192 / 256 bits), actual key (128-bits), and a block of data (16 bytes). The Key Expansion module will create an expanded key (based on the key length) of sizes: 1408-bits, 1664-bits, or 1920-bits. Add Round Key will XOR the block of data with 128-bits of the expanded key. Sub Bytes will use the lower 4 bits of each byte as a column parameter and the upper 4 bits of each byte as a row parameter. Shift Rows will rotate each column down by n bytes. Mixed Columns will use matrix multiplication on each row. Sub Bytes and Mix Columns use given matrices for calculations. Decryption uses the same steps but with different given matrices for calculations. The Initial Rounds of encryption and decryption only use Add Round Key. The Final Rounds of encryption and decryption don't use Mixed Columns.

```
Prolific USB-to-Serial Comm Port...
Reads
113462: 2019-11-20 16:40:36.0347892 +0.0000050
00 6E 73 CE 62 C8 16 AE 2B A9 37 16 5C 1F B2 8D .na1bE.0+07.\.f
68 h
113564: 2019-11-20 16:40:37.8106778 +0.0000048
00 1C 72 46 0F 15 6E 2E F9 A0 73 57 B9 F4 95 40 ...rP...n.ù sW'ô+0
F3 ô
113666: 2019-11-20 16:40:39.5840175 +0.0000048
00 8D D1 47 16 5B 51 38 0D 04 F2 B4 55 C2 53 3E . Ng.[08...ô'uAS>
<
Writes
113427: 2019-11-20 16:40:35.1211244 +0.0000309
0A 55 4E 4C 56 20 69 73 20 73 75 63 68 20 61 20 .UNLV is such a
113529: 2019-11-20 16:40:36.8977615 +0.0000282
67 72 65 61 74 20 73 63 68 6F 6F 6C 21 0A 49 20 great school!.I
113631: 2019-11-20 16:40:38.6707343 +0.0000279
6C 6F 76 65 20 63 6F 6D 70 75 74 65 72 73 0A 0A love computers..
<
```

```
Prolific USB-to-Serial Comm Port...
Reads
152878: 2019-11-20 16:45:25.7002199 +0.0000056
00 0A 55 4E 4C 56 20 69 73 20 73 75 63 68 20 61 ..UNLV is such a
20
152980: 2019-11-20 16:45:27.4723206 +0.0000049
00 67 72 65 61 74 20 73 63 68 6F 6F 6C 21 0A 49 .great school!.I
20
153082: 2019-11-20 16:45:29.2476354 +0.0000050
00 6C 6F 76 65 20 63 6F 6D 70 75 74 65 72 73 0A .love computers.
0A
<
Writes
152843: 2019-11-20 16:45:24.7846569 +0.0000283
3F C3 F8 2C 53 14 AF FB 87 EA 45 51 68 CE A6 AF ?Ås,S."âiëQh!;"
152945: 2019-11-20 16:45:26.5605277 +0.0000336
6B 02 06 C5 02 3D 73 E4 50 5A C2 7A 34 54 AC 5F k..Ä.=aPz4T-_
153047: 2019-11-20 16:45:28.3332322 +0.0000316
43 83 97 32 A6 BB 37 18 6D B7 89 BA BB D2 31 F8 Cf-2;~7.m'b'*sô1e
<
```

Future Improvements:

Given more time, we would have also implemented an external server (using Raspberry Pi) to send and receive the encrypted files via internet. We would have also interfaced our FPGA with an Arduino (or any other microcontroller) to allow for file transfer via Bluetooth, WiFi, and SPI (e.g. SD card). We also would have fabricated the AES algorithm onto an IC (integrated circuit).

Technical Details:

Our entire AES was implemented with Verilog 2001, Quartus v16.1, while using an Altera DE2-115. The AES synthesis requires 24,375 logic elements, 3,605 registers, 8 user I/O, and 8,192 memory bits. The program used for file transfer was coded in C++ 17 and used CodeBlocks IDE v17.12. We used an external library to handle serial port communication called "Serial-Port.h".

Conclusion:

Both being Computer Engineering majors, this project helped us learn about cryptography and file security. We gained valuable experience using FPGAs and serial communication.