Applying Grover's algorithm to AES: quantum resource estimates

Markus Grassl[1], Brandon Langenberg[2], Martin Roetteler[3] and Rainer Steinwandt[2]

[1] Universität Erlangen-Nürnberg & Max Planck Institute for the Science of Light

[2] Florida Atlantic University

[3] Microsoft Research

February 24, 2016

Overview

- Quantum circuits for implementing an exhaustive key search for the Advanced Encryption Standard (AES)
- Analyze the quantum resources required
- Consider the overall circuit size, number of qubits, and circuit depth
- Focus on the Clifford+T gate set
- Establish precise bounds for qubits and gates needed to implement Grover's algorithm for all three versions (128, 192, and 256 bit) that are standardized in FIPS-PUB 197

1 AES: Rounds

2 AES: Key Expansion

3 Resource Estimates

4 Grover

5 Uniqueness

6 Conclusion

AES: Rounds

128 qubits hold the current internal state

$$\begin{array}{|c|c|c|c|}\hline S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ \hline S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ \hline S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ \hline S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \\ \hline \end{array} \Rightarrow \begin{array}{c} |S_{0,0}\rangle \\ |S_{1,0}\rangle \\ \vdots \\ |S_{3,3}\rangle \end{array}$$

Each round of AES applies the following four operations:

- `SubBytes`
- `ShiftRows`
- `MixColumns`
- `AddRoundKey`

AES: SubBytes

- Treat each byte as $\alpha \in \mathbb{F}_2[x]/(1 + x + x^3 + x^4 + x^8)$
- Finds α^{-1} (leaving 0 invariant)
- Applies an affine transformation

AES: `SubBytes`

- Treat each byte as $\alpha \in \mathbb{F}_2[x]/(1 + x + x^3 + x^4 + x^8)$
- Finds α^{-1} (leaving 0 invariant)
- Applies an affine transformation

- Classical AES can employ a lookup table

AES: SubBytes

- Treat each byte as $\alpha \in \mathbb{F}_2[x]/(1 + x + x^3 + x^4 + x^8)$
- Finds α^{-1} (leaving 0 invariant)
- Applies an affine transformation

- Classical AES can employ a lookup table

- Decided explicity calculating result was more resource friendly

- Treat each byte as $\alpha \in \mathbb{F}_2[x]/(1 + x + x^3 + x^4 + x^8)$
- Finds α^{-1} (leaving 0 invariant)
- Applies an affine transformation
- Classical AES can employ a lookup table
- Decided explicity calculating result was more resource friendly
- Itoh-Tsujii inverter:

$$\alpha^{-1} = \alpha^{254} = ((\alpha \cdot \alpha^2) \cdot (\alpha \cdot \alpha^2)^4 \cdot (\alpha \cdot \alpha^2)^{16} \cdot \alpha^{64})^2$$

AES: SubBytes: $\alpha^{-1} = ((\alpha \cdot \alpha^2) \cdot (\alpha \cdot \alpha^2)^4 \cdot (\alpha \cdot \alpha^2)^{16} \cdot \alpha^{64})^2$

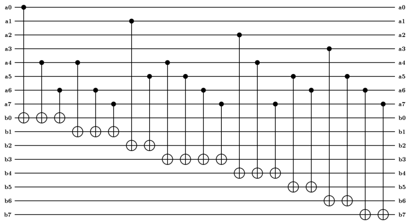
Qubits	0	1	2⋆	3	4⋆	5⋆	6
00-07	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$
08-15	$\|0\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$
16-23	$\|0\rangle$	$\|0\rangle$	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|0\rangle$	$\|0\rangle$
24-31	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|\alpha^{12}\rangle$	$\|\alpha^{12}\rangle$	$\|\alpha^{12}\rangle$	$\|\alpha^{48}\rangle$
32-39	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$

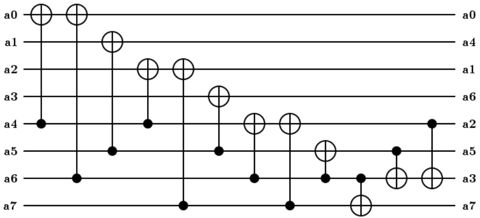
Qubits	7⋆	8	9	10⋆	11	12	13⋆
00-07	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha^{64}\rangle$	$\|\alpha^{64}\rangle$	$\|\alpha^{64}\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$
08-15	$\|\alpha^2\rangle$	$\|0\rangle$	$\|0\rangle$	$\|\alpha^{127}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$
16-23	$\|\alpha^{63}\rangle$	$\|\alpha^{63}\rangle$	$\|\alpha^{63}\rangle$	$\|\alpha^{63}\rangle$	$\|\alpha^{63}\rangle$	$\|\alpha^{63}\rangle$	$\|0\rangle$
24-31	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$	$\|\alpha^{48}\rangle$
32-39	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$

Qubits	14	15	16⋆	17	18	19⋆	20
00-07	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$	$\|\alpha\rangle$
08-15	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$	$\|\alpha^{254}\rangle$
16-23	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|\alpha^3\rangle$	$\|0\rangle$	$\|0\rangle$
24-31	$\|\alpha^{48}\rangle$	$\|\alpha^{12}\rangle$	$\|\alpha^{12}\rangle$	$\|0\rangle$	$\|\alpha^2\rangle$	$\|\alpha^2\rangle$	$\|0\rangle$
32-39	$\|\alpha^{15}\rangle$	$\|\alpha^{15}\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$

Example: Squaring in $\mathbb{F}_2[x]/(1+x+x^3+x^4+x^8)$

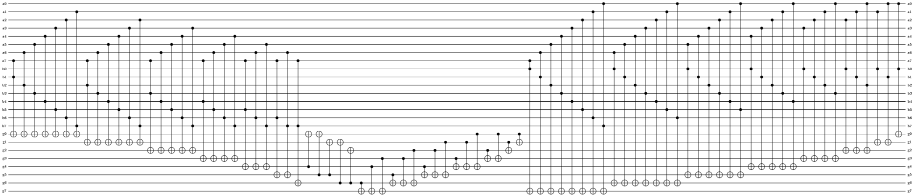
$$\begin{bmatrix} 1&0&0&0&1&0&1&0\\ 0&0&0&0&1&0&1&1\\ 0&1&0&0&0&1&0&0\\ 0&0&0&0&1&1&1&1\\ 0&0&1&0&1&0&0&1\\ 0&0&0&0&0&1&1&0\\ 0&0&0&1&0&1&0&0\\ 0&0&0&0&0&0&1&1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&1&1&1&0\\ 0&0&0&0&0&0&1&1 \end{bmatrix} \cdot \begin{bmatrix} 1&0&0&0&1&0&1&0\\ 0&1&0&0&0&1&0&0\\ 0&0&1&0&1&0&0&1\\ 0&0&0&1&0&1&0&0\\ 0&0&0&0&1&0&1&1\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$





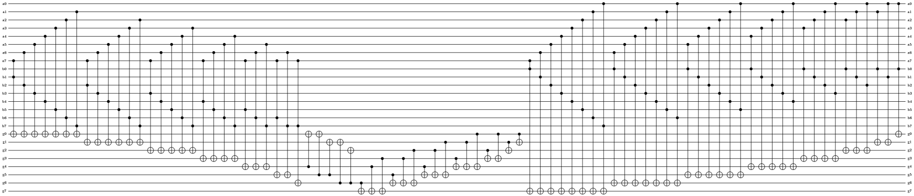
AES: SubBytes

Multiplication: Maslov et al.'s design



AES: SubBytes

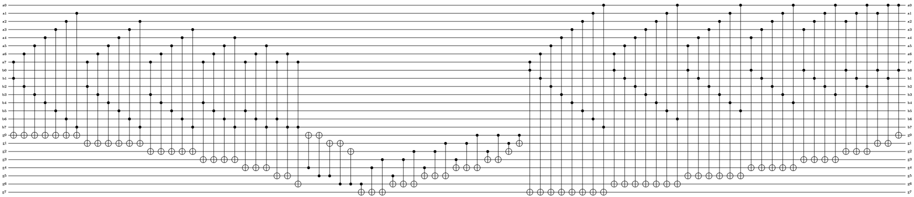
Multiplication: Maslov et al.'s design



64 Toffoli and 21 CNOT gates $\rightarrow$ 448 T plus 533 Clifford gates

AES: SubBytes

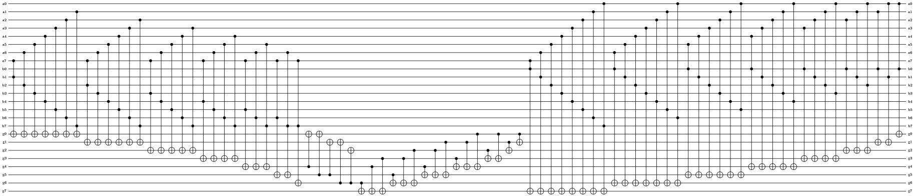
Multiplication: Maslov et al.'s design



64 Toffoli and 21 CNOT gates → 448 T plus 533 Clifford gates
8 total mulitplications per inversion → 3584 T plus 4264 Clifford gates

AES: SubBytes

Multiplication: Maslov et al.'s design



64 Toffoli and 21 CNOT gates → 448 T plus 533 Clifford gates
8 total mulitplications per inversion → 3584 T plus 4264 Clifford gates

EXPENSIVE

Final Step: Affine transformation computed, LUP decomposition used.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Final Step: Affine transformation computed, LUP decomposition used.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,0\,0\,0\,1 \\ 1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,1\,1\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

SubBytes:

AES: SubBytes

Final Step: Affine transformation computed, LUP decomposition used.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

`SubBytes`:

One 8-bit S-Box $\rightarrow$ 3584 T-gates and 4569 Clifford gates.

AES: SubBytes

Final Step: Affine transformation computed, LUP decomposition used.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

`SubBytes`:

One 8-bit S-Box $\rightarrow$ 3584 T-gates and 4569 Clifford gates.

16 S-Boxes per round!

AES: `ShiftRows`

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$		$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$\Leftarrow$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$\Leftarrow \Leftarrow$	$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$\Leftarrow \Leftarrow \Leftarrow$	$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

- Permutation of current AES state
- Permutation of qubits
- Instead adjust position of subsequent gates
- Addressed during next `SubBytes`

AES: MixedColumns

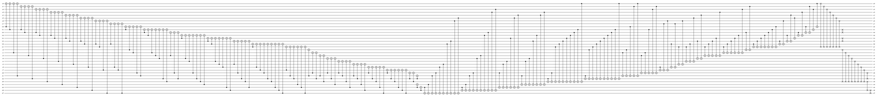
Operates on entire column (32 (qu)bits) at a time. LUP decomposition on 32×32 matrix to compute in place $\rightarrow$ 277 CNOT gates, depth of 39.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

$$\{02\} \rightarrow \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0 \end{pmatrix}; \quad \{03\} \rightarrow \begin{pmatrix} 1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \quad .$$

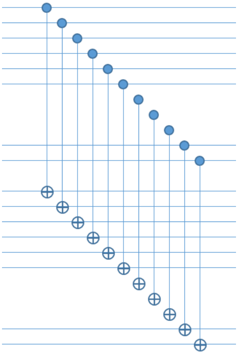
AES: `MixedColumns`

`MixColumns`: 277 CNOT gates with total depth of 39



AES: `AddRoundKey`

Bit-wise XOR of the round key with the current state



128 CNOT gates executed in parallel

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

- `RotWord()` - Similar to `ShiftRows`, requires no current action

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

- `RotWord()` - Similar to `ShiftRows`, requires no current action
- `SubWord()` - Applies `SubBytes` once to each of the 4 bytes

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

- `RotWord()` - Similar to `ShiftRows`, requires no current action
- `SubWord()` - Applies `SubBytes` once to each of the 4 bytes
- `Rcon[i/Nk]` - Flip of bits - One or two uncontrolled NOT gates

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

- `RotWord()` - Similar to `ShiftRows`, requires no current action
- `SubWord()` - Applies `SubBytes` once to each of the 4 bytes
- `Rcon[i/Nk]` - Flip of bits - One or two uncontrolled NOT gates
- `XOR` - XORing words - 32 controlled NOT gates executed in parallel

AES: Key Expansion

Key Expansion employs the following to 32 bits (1 word) at a time:

- `RotWord()` - Similar to `ShiftRows`, requires no current action
- `SubWord()` - Applies `SubBytes` once to each of the 4 bytes
- `Rcon[i/Nk]` - Flip of bits - One or two uncontrolled NOT gates
- `XOR` - XORing words - 32 controlled NOT gates executed in parallel

`SubWord()` (expensive) only applied to every 4 (AES-128, 256) or 6 (AES 192) words in key expansion. These words are treated differently.

AES: Key Expansion: `SubWord()`

`SubWord()` like `SubBytes` was costly $\rightarrow$ storing seemed cost effective

- AES-128 - 4 words into 44. 10 use `Subword()`, constructed & stored

AES: Key Expansion: `SubWord()`

`SubWord()` like `SubBytes` was costly → storing seemed cost effective

- AES-128 - 4 words into 44. 10 use `Subword()`, constructed & stored
- AES-192 - 6 words into 52. 8 use `Subword()`, constructed & stored

AES: Key Expansion: `SubWord()`

`SubWord()` like `SubBytes` was costly $\rightarrow$ storing seemed cost effective

- AES-128 - 4 words into 44. 10 use `Subword()`, constructed & stored
- AES-192 - 6 words into 52. 8 use `Subword()`, constructed & stored
- AES-256 - 8 words into 60. 13 use `Subword()` constructed & stored

AES: Key Expansion: `SubWord()`

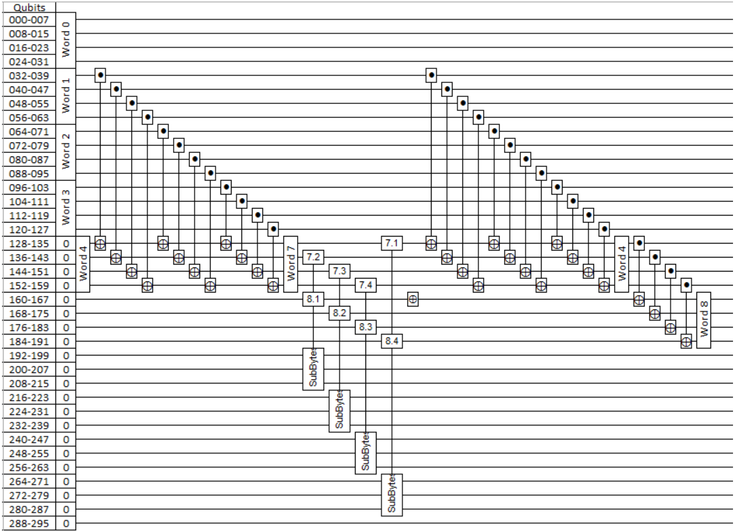
`SubWord()` like `SubBytes` was costly $\rightarrow$ storing seemed cost effective

- AES-128 - 4 words into 44. 10 use `Subword()`, constructed & stored
- AES-192 - 6 words into 52. 8 use `Subword()`, constructed & stored
- AES-256 - 8 words into 60. 13 use `Subword()` constructed & stored

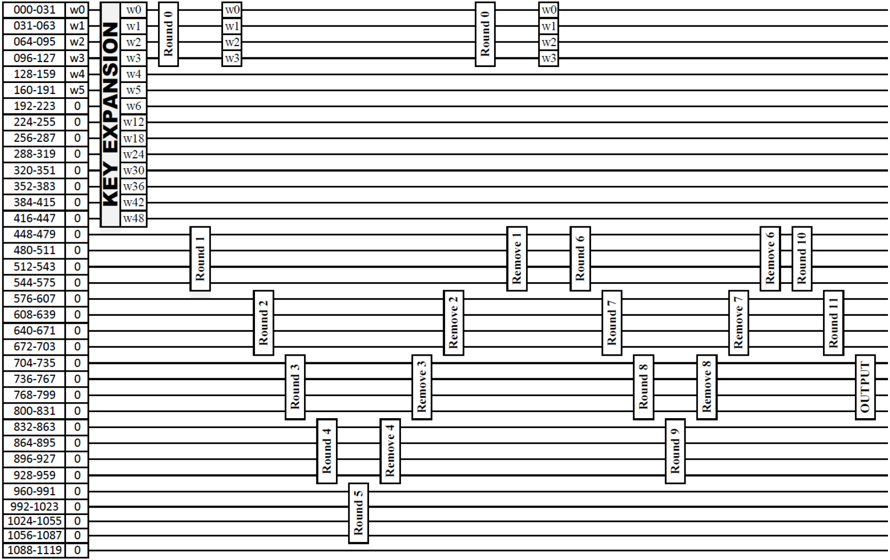
	#gates			depth		#qubits	
	NOT	CNOT	Toffoli	T	overall	storage	ancillae
128	176	21,448	20,480	5,760	12,636	320	96
192	136	17,568	16,384	4,608	10,107	256	96
256	215	27,492	26,624	7,488	16,408	416	96

AES: Key Expansion:

Remaining words constructed only using `XOR` $\rightarrow$ generate as needed.
Example: AES-128 - $word7 = word4 \oplus word3 \oplus word2 \oplus word1$



To save and reuse qubits, cleaned up along the way (Ex. AES-192)

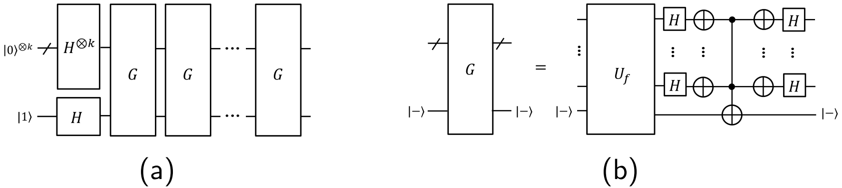


Resource Estimates

		#gates		depth		#qubits
		T	Clifford	T	overall	
AES-128	Key Gen	143,360	185,464	5,760	12,626	320
	10 Rounds	917,504	1,194,956	44,928	98,173	536
	Total	1,060,864	1,380,420	50,688	110,799	856+128

		#gates		depth		#qubits
		T	Clifford	T	overall	
AES-192	Key Gen	114,688	148,776	4,608	10,107	256
	12 Rounds	1,089,536	1,418,520	39,744	86,849	664
	Total	1,204,224	1,567,296	44,352	96,956	920+192

		#gates		depth		#qubits
		T	Clifford	T	overall	
AES-256	Key Gen	186,368	240,699	7,488	16,408	416
	14 Rounds	1,318,912	1,715,400	52,416	114,521	664
	Total	1,505,280	1,956,099	59,904	130,929	1080+256



(a) Grover circuit applied $\lfloor \frac{\pi}{4}\sqrt{2^k} \rfloor$ times for $k = 128, 192, 256$
(b) Shown is the circuit decomposition of G

$$G = \Big((H^{\otimes k}(2|0\rangle\langle 0| - 1)H^{\otimes k}) \otimes \mathbf{1}_2 \Big) \; U_f$$

Uniqueness

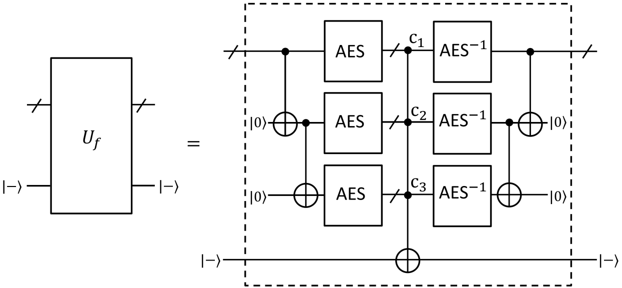


Figure: Reversible implementation of U_f. For $k = 128$, $r = 3$ invocations of AES suffice to make the target key unique. For $k = 192$ number of parallel AES boxes increases to $r = 4$ and for $k = 256$ to $r = 5$. Overall structure of the circuit is common to all key sizes.

Conclusion

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	$2,953$
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	$4,449$
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Table: Resource estimates for Grover to attack AES-k, $k \in \{128, 192, 256\}$.

Conclusion:
Only `SubBytes` involves T-gates and called a minimum of 296 times (AES-128) and up to 420 (AES-256). Results in quantum circuits of quite moderate complexity. Seems prudent to move away from 128-bit keys.