Implementing Grover Oracles for Quantum Key Search on AES and LowMC

Samuel Jaques[1], Michael Naehrig[2], Martin Roetteler[3], **Fernando Virdia**[4]

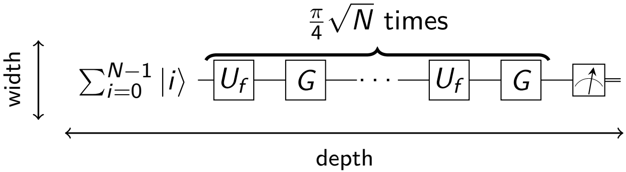
[1]Department of Materials, University of Oxford, UK
[2]Microsoft Research, Redmond, WA, USA
[3]Microsoft Quantum, Redmond, WA, USA
[4]Information Security Group, Royal Holloway, University of London, UK

Eurocrypt 2020
The interwebs

- In 2016, NIST put out a call for post-quantum cryptography proposals [Nat16].
- The call defines security *categories* that candidate schemes should belong to.
- Categories 1, 3, and 5's definitions are based on the hardness of key recovery against AES-128, -192, -256, respectively.

- How hard is it to break AES with a quantum computer?
- The only known strategy is “Groverising” exhaustive key search.

Grover’s search sketch



- Early termination of Grover’s search results in low success probabilities.

What are the cost metrics for a quantum circuit? Some options:

D-cost: depth of the circuit

- The depth is considered proportional to the time it requires to evaluate the circuit.

G-cost: number of gates and measurements

- Idle qubits don't have a cost.

DW-cost: depth-times-width of the circuit

- Captures the need for error correction on the idle qubits.
- One can then try to compare to the classical cost required to error-correct to the cost of equivalent classical attacks [JS19, AGPS19].

In all three cases, different gates can be assigned different weights.

- For Grover's search, Zalka [Zal99] showed that using S machines saves only $\sqrt{S}$ depth, optimally.
- This non-trivial tradeoff means using more machines to cut attack duration may result in larger costs.
- To capture this, NIST suggest having an explicit $\mathtt{MAXDEPTH} \in \{2^{40}, 2^{64}, 2^{96}\}$ parameter bounding quantum circuit depth.
  - `MAXDEPTH` is related to the total depth of the circuit, and not to the qubit's coherence times.

- They then infer the cost of using Grover's against AES.
  - Say non-parallel Grover requires depth $D = x \cdot \mathtt{MAXDEPTH}$, for some $x \geq 1$ and G gates.
  - To cut depth by x, x^2 machines are needed. Each uses $\approx G/x$ gates.
  - Total gate count: $(G/x) \cdot x^2 = G \cdot D/\mathtt{MAXDEPTH}$.

Attack gate counts

AES-128	$2^{170}/\mathtt{MAXDEPTH}$ quantum gates
AES-192	$2^{233}/\mathtt{MAXDEPTH}$ quantum gates
AES-256	$2^{298}/\mathtt{MAXDEPTH}$ quantum gates

Table: Attack costs using D and G from Grassl et al. [GLRS16].

- Our initial idea: NIST cares about limiting depth, but uses [GLRS16] which optimizes for width. What if we minimize depth?
- Hindsight: parallelisation is bad, so crucially beneficial to minimise depth!
- Let's design parallel-friendly circuits, and implement them in Q#:
  - testable,
  - friendly to read/modify,
  - automated circuit size estimates,
  - easy to translate already existing AES components!

Assumptions

- We only work with logical qubits.
- We do not assume any particular framework (e.g. the surface code).
  - Hence no costs for idle qubits or need for gates to operate locally.
  - But also no speedups like free CNOT fan-outs.
- Swapping qubits is free, by “rewiring” (keeping track of the swaps).

This is not necessarily realistic, but is what the previous literature on AES (and hence NIST in [Nat16]) uses.

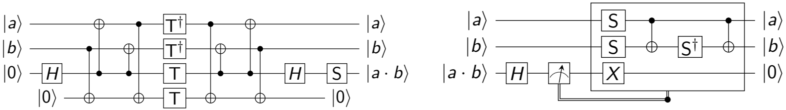
Let's look at our design choices for a smaller Grover oracle for AES.

S-box: well investigated in the hardware literature.

- Lots of linear programs to port to Q# and test.
- Tried various variants of [BP11].
- Scooped! In concurrent indepedent work, Langenberg et al. [LPS19] propose a similar S-box change.
  - They provide an implementation of their S-box.

Logic gates:

- [GLRS16] use a 7 T-gates, T-depth 4 implementation of Toffoli gates.
- We replace Toffoli's with AND gates, using a custom design by Mathias Soeken, based on Selinger [Sel13] and Jones [Jon13].



(a) AND gate. (b) AND† gate.

- It reduces T-depth to 1 and T-gates to 4, and has a "T-free" adjoint operator. It does introduce measurements.

KeyExpansion:

- [GLRS16] caches costly-to-compute bytes. Tricky to keep track of.

In-place round key expansion

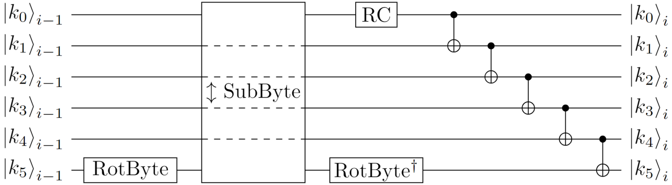


Figure: AES 192 in-place i^{th} round key expansion.

- This saves us qubits with respect to full round-key precomputation, while not increasing depth due to the computations running in parallel to the round.

Other improvements:

- We cost both [GLRS16]'s MixColumn design, and a recent, shallower (but wider) design by Maximov [Max19].
- Fix to the key uniqueness computation.
  - To uniquely identify a secret key, more than one message-ciphertext pairs are needed.
  - [GLRS16] overestimates how many are needed for a $p \approx 1$ attack.
  - As Langenberg et al. [LPS19] also noticed, we suggest using 1, 2, 2 pairs for high probability attacks ($\approx 1/e$, ≈ 1, $\approx 1/e$) in the unbounded-depth setting.

scheme	pairs	width	#Clifford	#M	#T	T-depth	full depth	G-cost	DW-cost	p_{succ}
Grassl et al. [GLRS16]										
AES-128	3	2 953	86	—	86	80	81	87	92	1
AES-192	4	4 449	119	—	118	112	113	120	125	1
AES-256	5	6 681	151	—	151	144	145	152	158	1
Langenberg et al. [LPS19]										
AES-128	1	865	82	—	81	77	79	83	89	$1/e$
AES-192	2	1 793	115	—	114	109	111	116	122	1
AES-256	2	2 465	148	—	147	141	143	148	154	$1/e$
this work										
AES-128	1	1665	82	77	79	70	75	82	85	$1/e$
AES-128	2	3329	83	78	80	70	75	83	86	1
AES-192	2	3969	115	110	112	102	107	115	119	1
AES-256	2	4609	147	142	144	134	139	147	151	$1/e$
AES-256	3	6913	148	143	145	134	139	148	152	1

$\log_2$

AES-128 in $\texttt{MAXDEPTH} = 2^{96}$ is the only attack fitting. For the others, we consider the two strategies from Kim et al.e [KHJ18]:

Outer parallelisation

Run S independently, and stop early. Success probability $\xrightarrow{S\to\infty}$ 0.915.

Inner parallelisation

- The total search space has size N. Partition it into S disjoint subsets. Only one subset contains the correct key.
- Run S machines, each on a different subset of size N/S, and measure their output.
- To reduce depth by $\sqrt{S}$, we run for $\frac{\pi}{4}\sqrt{\frac{N}{S}}$ iterations. These are the right number of iterations to find the key with $p \approx 1$ in its subset of size N/S.
- The correct key will be measured with $p \approx 1$ in its subset. Classically check all S outputs to win.

Side effect:

- For AES-128, we need 2 plaintext-ciphertext pairs to uniquely identify the secret key $K \in \mathbb{K} = \{0, 1\}^{128}$.
- Using 1 pair (m, c), the probability that only one key in $\mathbb{K}$ maps $m \mapsto c$ is $1/e$.
- Let's partition $\mathbb{K}$ into S subsets. Say $K \in \mathbb{K}_K$. The probability that another "spurious" key mapping $m \mapsto c$ exists in $\mathbb{K}_K \subset \mathbb{K}$ shrinks as S grows.
- In practice, sometimes 1 plaintext-ciphertext pair in the quantum phase is enough. $\implies$ Less qubits are needed.

		log$_2$					
scheme	pairs	`MD`	D	S	W	G-cost	DW-cost
AES-128	1	40	40	69	80	117	120
AES-192				133	144	181	184
AES-256				197	209	245	249
AES-128	1	64	64	21	32	93	96
AES-192				85	96	157	160
AES-256				149	161	221	225
AES-128*	2	96	75	0	11	83	86
AES-192			96	21	33	126	129
AES-256			96	85	98	190	194

Some observations:

- Say a candidate scheme for category 5 does a similar analysis, and the best quantum attack with $\texttt{MAXDEPTH} = 2^{40}$ has G-cost 2^{230}.
  - Does it not meet the criteria? Nobody is going to build 2^{197} quantum computers anyway, so Grover is not really an attack against AES-256 there.
- Logical qubits won't be free. Should we introduce `MAXWIDTH`? What would it mean?
  - Maybe that we try to fit Grover within `MAXWIDTH`, compute the success probability for the resulting attack, and then do the same for candidates ("Cat 5, $\texttt{MD}\ 2^{40}$, $\texttt{MW}\ x$ means no quantum attack with success prob $\geq 2^{-\cdots}$")?

Finally, we can recompute NIST's table, taking into account inner parallelisation advantages.

NIST Security Strength Category	source	G-cost for `MAXDEPTH` 2^{40}	2^{64}	2^{96}	approximation
1 AES-128	[Nat16]	2^{130}	2^{106}	2^{74}	$2^{170}/\mathtt{MAXDEPTH}$
	this work	$\mathbf{2^{117}}$	$\mathbf{2^{93}}$	$*\mathbf{2^{83}}$	$\approx \mathbf{2^{157}}/\mathtt{MAXDEPTH}$
3 AES-192	[Nat16]	2^{193}	2^{169}	2^{137}	$2^{233}/\mathtt{MAXDEPTH}$
	this work	$\mathbf{2^{181}}$	$\mathbf{2^{157}}$	$\mathbf{2^{126}}$	$\approx \mathbf{2^{221}}/\mathtt{MAXDEPTH}$
5 AES-256	[Nat16]	2^{258}	2^{234}	2^{202}	$2^{298}/\mathtt{MAXDEPTH}$
	this work	$\mathbf{2^{245}}$	$\mathbf{2^{221}}$	$\mathbf{2^{190}}$	$\approx \mathbf{2^{285}}/\mathtt{MAXDEPTH}$

Another application: LowMC.

- LowMC [ARS+15] is a block cipher family designed for FHE and MPC.
- It is used as part of the Picnic [ZCD+17] submission.
- We used the same tools and techniques used for AES to investigate its security.

key size	AES G-cost	LowMC G-cost
128	$2^{157}/\mathtt{MAXDEPTH}$	$2^{163}/\mathtt{MAXDEPTH}$
192	$2^{221}/\mathtt{MAXDEPTH}$	$2^{231}/\mathtt{MAXDEPTH}$
256	$2^{285}/\mathtt{MAXDEPTH}$	$2^{297}/\mathtt{MAXDEPTH}$

Further research directions:

- Improve the AES oracle with better S-boxes
  - Sacrificing simulatability, it would be possible to use a compiler based on [GKMR14, ZC19] to automatically synthetise smaller circuits.
  - An orthogonal automatic technique could be to use the classical circuit minimizer by [MSR+19, MSC+19] to attempt to further reduce the linear program components.
- Improve the LowMC design by adopting the approach from [DKP+19].
- Redo the analysis in the surface code setting (it would require new implementations probably, maybe a specific surface-code compiler).

- Take some of the quantum algorithms proposed for the candidates (most use Grover), and do a similar analysis of their quantum component. Do they always/never/sometimes hit `MAXDEPTH`?
- What happens if we introduce `MAXWIDTH`? Or some other bound?
- How do the new oracles impact multi-target attacks? E.g. Banegas and Bernstein [BB17].

Thank you

See you at the panel discussion!

- Paper @ `https://ia.cr/2019/1146`
- Code @ `https://github.com/microsoft/grover-blocks`

Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck.
Quantum speedups for lattice sieves are tenuous at best.
Cryptology ePrint Archive, Report 2019/1161, 2019.
`https://eprint.iacr.org/2019/1161`.

Martin R Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner.
Ciphers for MPC and FHE.
In *EUROCRYPT 2015*. Springer, 2015.

Gustavo Banegas and Daniel J Bernstein.
Low-communication parallel quantum multi-target preimage search.
In *SAC 2017*. Springer, 2017.

Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp.
Tight bounds on quantum searching.
Fortschritte der Physik, 46(4-5):493–505, 1998.

Joan Boyar and Rene Peralta.
A depth-16 circuit for the AES s-box.
Cryptology ePrint Archive, Report 2011/332, 2011.
`http://eprint.iacr.org/2011/332`.

Itai Dinur, Daniel Kales, Angela Promitzer, Sebastian Ramacher, and Christian Rechberger.
Linear equivalence of block ciphers with partial non-linear layers: Application to lowmc.
In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 343–372, Cham, 2019. Springer International Publishing.

David Gosset, Vadym Kliuchnikov, Michele Mosca, and Vincent Russo.
An algorithm for the t-count.
Quantum Information & Computation, 14(15-16):1261–1276, 2014.

Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt.
Applying grover's algorithm to AES: quantum resource estimates.
In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, volume 9606 of *Lecture Notes in Computer Science*, pages 29–43. Springer, 2016.

Cody Jones.
Low-overhead constructions for the fault-tolerant Toffoli gate.
Physical Review A, 87(2):022328, 2013.

Samuel Jaques and John M Schanck.
Quantum cryptanalysis in the ram model: Claw-finding attacks on sike.
In *Annual International Cryptology Conference*, pages 32–61. Springer, 2019.

Panjin Kim, Daewan Han, and Kyung Chul Jeong.
Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to aes and sha-2.
Quantum Information Processing, 17(12):339, Oct 2018.

Brandon Langenberg, Hai Pham, and Rainer Steinwandt.
Reducing the cost of implementing aes as a quantum circuit.
Cryptology ePrint Archive, Report 2019/854, 2019.

https://eprint.iacr.org/2019/854.

Alexander Maximov.
Aes mixcolumn with 92 xor gates.
Cryptology ePrint Archive, Report 2019/833, 2019.
https://eprint.iacr.org/2019/833.

Giulia Meuli, Mathias Soeken, Earl Campbell, Martin Roetteler, and Giovanni De Micheli.
The role of multiplicative complexity in compiling low t-count oracle circuits.
CoRR, abs/1908.01609, 2019.

Giulia Meuli, Mathias Soeken, Martin Roetteler, Nikolaj Bjørner, and Giovanni De Micheli.
Reversible pebbling game for quantum memory management.
In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*, pages 288–291, 2019.

National Institute of Standards and Technology.
Submission requirements and evaluation criteria for the Post-Quantum Cryptography standardization process.
http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf, December 2016.

Peter Selinger.
Quantum circuits of t-depth one.
Phys. Rev. A, 87:042302, Apr 2013.

Christof Zalka.
Grover's quantum searching algorithm is optimal.
Phys. Rev. A, 60, 10 1999.

Fang Zhang and Jianxin Chen.
Optimizing t gates in clifford+t circuit as $\pi/4$ rotations around paulis.
arXiv preprint arXiv:1903.12456, 2019.

Greg Zaverucha, Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, and Daniel Slamanig.
Picnic.
Technical report, NIST, 2017.