From: Moody, Dustin (Fed)
To: Perlner, Ray A. (Fed); Liu, Yi-Kai (Fed); Jordan, Stephen P (Fed); Miller, Carl A. (Fed)
Subject: Re: Thermodynamic analysis of brute force Key Search, Claw finding problems.
Date: Friday, June 30, 2017 4:10:49 PM

Ray,

I don't know about the quantum aspect as much as the rest of you, but I would be interested in working on this in regards to the consequences for SIDH. The quantum security for SIDH pretty much depends directly on the claw finding algorithm. The complexity is O(log p)^(1/6). If the quantum algorithm for claw finding becomes O(log p)^(1/4) as Ray thinks might be the case, then key sizes for SIDH can be shrunk 33%, as then both the best classical and quantum attacks would be O(log p)^(1/4). Instead of needing to choose your parameter p = 6 * (security level) as is currently done, it would be p = 4 * (security level).

Dustin

From: Perlner, Ray (Fed)
Sent: Friday, June 30, 2017 11:21:54 AM
To: Liu, Yi-Kai (Fed); Jordan, Stephen P (Fed); Miller, Carl A. (Fed); Moody, Dustin (Fed)
Subject: Thermodynamic analysis of brute force Key Search, Claw finding problems.

It's a well-known fact (https://cr.yp.to/hash/collisioncost-20090517.pdf) that if you're just counting gates, the complexity of the quantum collision finding algorithm from Brassard Hoyer and Tapp is no better than the best classical algorithm (Van Oorschot Weiner.) I believe the same argument indicates that Tani's claw finding algorithm doesn't really buy you anything in terms of circuit size/depth.

However, given that quantum memories can be just as energy efficient as classical memories at least in theory (see e.g. https://arxiv.org/abs/0708.1879) I was wondering whether maybe these algorithms could buy you something thermodynamically. It appears they can't:

I compared a classical reversible implementation of Van Oorschot-Weiner to a reversible version of quantum claw finding/ collision search: Recall (see e.g. https://www.math.ucsd.edu/~sbuss/CourseWeb/Math268_2013W/Bennett_Reversibiity.pdf) that the energy per operation in a reversible computation goes like the circuit depth, divided by the physical time required to compute. To find collisions/claws in a range of size N, using a memory of size M, in time t, the quantum algorithm requires sqrt(N/M) gates in series (excluding memory lookups, which we're assuming have lower power requirements than ordinary gates). The energy cost per gate is therefore sqrt(N/M)/t, resulting in a total energy cost of E = sqrt(N/M)/t * sqrt(N/M) = N/(Mt). The classical algorithm requires sqrt(N) gates in total, parallelized M ways. The circuit depth is then Sqrt(N)/M resulting in an energy cost of Sqrt(N)/ (Mt) per gate, and a total energy cost of E = Sqrt(N)/ (Mt) * Sqrt(N) = N/(Mt). Needless to say this is exactly the same.

What's even more surprising, though, is that Grover's algorithm has the same thermodynamic requirements as an idealized classical algorithm for key search. The classical algorithm is mostly

unpowered. In order to randomly sample keys, we can use thermal noise to do a random walk on the internal state of a reversible circuit that selects a key and reaches a dead end if the key is incorrect. If the key is correct, on the other hand, the circuit goes through a thermodynamically IRreversible transition to a halt state. The only parts of this algorithm that need to dissipate power are the final transition to the halt state (which requires a negligible amount of power,) and the initial construction of the circuit, which requires energy proportional to the memory times the temperature of the thermal noise used to do the random walk. In order to search a key space of size K in time t using M memory units, the energy cost per memory unit is K/Mt, and the total energy cost is E = K/Mt*M = K/t. Compare to a reversible/parallel version of Grover's algorithm. The circuit depth is sqrt(K/M) resulting in a per gate energy cost of sqrt(K/M)/t and a total energy cost of E = sqrt(K/M)/t* sqrt(K/M)* M = K/t.

I think the above is likely a somewhat deep result, and it give a good reason to be skeptical of the utility of quantum claw finding at the very least. There is a sensible argument that Grover's algorithm might turn out to be useful after all, since if you try to duplicate the performance of Grover's algorithm at fixed power, but for arbitrary key size, using the classical algorithm, you either require an amount of memory that grows linearly with the time, or you need to run the system at an energy scale that grows linearly with time. (Niether of these is required for Grover.) Arbitrarily high memory requirements or temperatures are notably inconvenient. That said, my calculations indicate that you could get away with using terrestrial scale resources at room temperature to duplicate Grover's algorithm for about a year.

Also worth noting, I neglected small factors (like the circuit size and depth of individual block cipher or hash function queries, and just set them to 1. I did work it out, not suppressing these factors, and both pairs of algorithms still came out the same, though.

Anyway, I think this is a very interesting result. I would be interested to know if any of you want to collaborate on it.

Cheers,
Ray