evolutionQ

A resource estimation framework for quantum attacks against cryptographic functions - recent developments

GRI quantum risk assessment report Feb. 2020

Vlad Gheorghiu and Michele Mosca

evolutionQ Inc., Kitchener ON, Canada

February 15, 2020

Abstract. We analyze the security against quantum adversaries of currently deployed asymmetric (public-key) cryptographic schemes that include RSA and Elliptic-Curve Diffie-Hellman (ECDH), as well as symmetric schemes (ciphers) that include the family of AES cryptographic ciphers. We use the latest advances in cryptanalysis, circuit compilation and fault-tolerant theory (such as surface-code lattice surgery techniques [1–3]) and windowed arithmetic [4,5] when providing the updated estimates.

In addition to the more conservative (from a cybersecurity perspective) choice of a physical error rate per gate of 10^{-5}, here we also highlight the security parameters for a 10^{-3} physical error rate per gate, which is more realistic in the short term[1].

1 Introduction

Quantum computers pose serious threats to current deployed cryptography, weakening symmetric cryptography and hash functions via Grover's quantum searching algorithm [6, 7] and breaking public-key systems based on factoring large numbers (RSA [8]) or solving discrete logarithms in finite groups (Elliptic Curve Cryptography (ECC) [9, 10]) via Shor's algorithm [11].

As mentioned in detail in our previous reports [12–15], a realistic attack using a fully fault tolerant quantum computer attack against a cryptographic scheme requires several layers, depicted here again for the sake of completeness in Fig. 1. Any improvement in any of the layers below decreases the resources (space, i.e. number of qubits, or time, or both) needed to break the scheme.

[1] Assuming a quantum computer that will run on a surface-code based fault-tolerant error-correcting layer, which, up to today, seems to be the most promising candidate for quantum error correction.

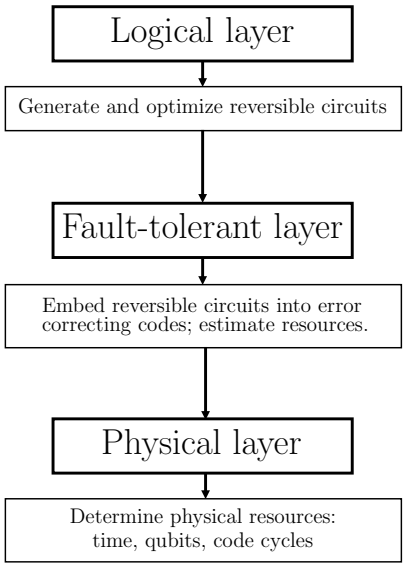

Fig. 1. Analyzing an attack against a cryptographic scheme with a fault-tolerant quantum adversary by considering several layers, going from the most abstract one (the logical layer), to the fault-tolerant layer that implements the circuit in a fault-tolerant way by taking into account that the physical implementation is imperfect, to the underlying physical layer itself.

Therefore keeping track of the latest developments and advances related to any of those layers is of paramount importance in quantum cryptanalysis.

In the remainder of this paper we investigate the security of asymmetric (or public-key) cryptographic schemes such as RSA and Elliptic-Curve Diffie-Hellman (ECDH) against quantum attacks, as well as the security of symmetric ciphers (AES) using the latest developments and advances related to the layers depicted in Fig. 1.

In our previous reports we analyzed the security of hash functions. From the time we performed our latest analysis until today there has been no major advance in the quantum cryptanalysis of hash functions, hence there is no revision of their security estimates since our latest report on hash functions [14].

2 Methodology

Most of the recent progress in quantum cryptanalysis is related to improved reversible windowed-arithmetic techniques [4, 5] at the logical layer in Fig. 1, recently introduced by Gidney and Ekerå [4]. Together with better magic state distillation schemes [16] and surface code lattice surgery methods [1–3] corresponding to the fault tolerant layer in Fig. 1, those allow for significant reduction of the *quantum resources* occupied by the quantum algorithm used to attack public key crypto-systems based on the hardness of factoring large numbers (RSA) or solving the discrete logarithm problem over elliptic curves groups (ECC).

We represent the *quantum resources* as a single-number quantity that roughly quantifies the product between the space (number of qubits) occupied by a quantum circuit and the time required to run it (which is proportional to its depth, i.e. the number of non-parallel operations). Note that here is a tradeoff between space and time, i.e. one can reduce the time required to run a quantum circuit by increasing the number of qubits (parallelization) and viceversa, while keeping the product between space and time relatively constant, hence the quantum resources roughly quantifies the efficiency of the implementation of a quantum circuit. In this report we represent the quantum resources in units of *megaqubit-days*, i.e. millions of qubits required to break the scheme in 24 hours (1 day).

For symmetric schemes (AES), most of the recent advances were made at the logical layer of the circuit in terms of improving the T gate counts [17], which, overall, slightly reduced the quantum security parameter[2] of the aforementioned schemes.

As mentioned in detail in our previous reports [12–15], and repeated here for the sake of completeness, any quantum algorithm can be mapped to a quantum circuit, and the latter "executed" on a quantum computers. The quantum circuit represents what we call the "logical layer". Such a circuit can always be decomposed in a sequence of "elementary gates", such as Clifford gates (CNOT, Hadamard etc. [18]) augmented by a non-Clifford gate such as the T gate.

[2] The quantum security parameter is defined as the logarithm base two of the number of fundamental operations (in our case surface code cycles) required to break the scheme.

Running a logical circuit on a full fault-tolerant quantum computer is highly non-trivial. Since physical quantum gates are imperfect, one first needs to map the logical circuit into a fault-tolerant implementation of it, followed by mapping the latter to sequences of surface code measurement cycles (see e.g. [19] for extensive details). By far, the most resource-consuming (in terms of number of qubits required and time) is the T gate[3].

In the following we consider the best up-to-date optimized quantum logical circuits for attacking RSA and ECC public-key schemes [4, 5], as well as AES symmetric ciphers [17] then perform a resource estimation analysis using lattice surgery techniques. For RSA public key schemes we tabulate the number of logical qubits required to break the scheme, the total number of physical qubits, the overall running time, and the corresponding quantum resources (in megaqubit-days) for a physical error rate p_g of 10^{-3} and 10^{-5}, respectively. For symmetric schemes, we tabulate their quantum security parameter (in bits), the number of logical qubits required to break the scheme, and the total number of physical qubits.

Note that in all our estimates we used a surface code cycle time of $200ns$. For this reason, if one wants to compare our running times (or the overall quantum resources) with the ones mentioned in [4], one should multiply our estimates by a factor of 5. For example, for RSA-2048 in Table 2, with a physical error rate of $p_g = 10^{-3}$, we display an expected running time of 1.46 hours, however in [4] they mention 7.3 hours (a factor of 5 longer than ours); similarly for the quantum resources.

3 Public key cryptographic schemes – RSA

For all our running time estimates in this Section, the running time is the same for both old estimates and the current estimates (e.g., in Table 1 for $p_g = 10^{-3}$ the running time is 0.27 hours, for both old estimates and current estimates). This is intentional, so one can perform a fair comparison. For the current estimates, the running time was obtained using the circuits of [4], then the number of physical qubits corresponding to this running time using the old estimates was obtained from the corresponding space/time tradeoff curve in our previous report [15]. Moreover, for the new estimates, the running time and the quantum resources are denoted as "expected time" and "expected quantum resources", respectively. This is because the techniques of [4] have an associated failure probability which we take into consideration.

[3] Clifford gates are "cheap", i.e. they require relatively small overhead for implementation in the surface code, but are not universal, hence a non-Clifford gate is required. One such gate is the T gate. There are other possible choices, however all of the non-Clifford gates require special techniques such as magic state distillation [3, 20] and significant overhead (orders of magnitude higher than Clifford gates) to be implemented in the surface code. In fact, to a first order approximation, for the purpose of resource estimation, one can simply ignore the overhead introduced by the Clifford gates and simply focus only on the T gates.

3.1 RSA-1024

RSA-1024	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	2050	1889	30	0.27	3093	9.62	0.11
10^{-5}	2050	111	2.14	0.21	3093	4.83	0.04

Table 1. RSA-1024 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is 80 bits.

3.2 RSA-2048

RSA-2048	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	4098	2632	172	1.46	6190	19.2	1.17
10^{-5}	4098	206	9.78	0.84	6190	9.66	0.34

Table 2. RSA-2048 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is 112 bits.

3.3 RSA-3072

RSA-3072	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	6146	5655	641	2.55	9288	37.9	4.03
10^{-5}	6146	292	25.5	1.89	9288	14.5	1.14

Table 3. RSA-3072 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is 128 bits.

3.4 RSA-4096

RSA-4096	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	8194	7057	1182	4.44	12387	54.6	10.10
10^{-5}	8194	383	57.0	3.37	12387	19.3	2.71

Table 4. RSA-4096 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is approximately 156 bits.

3.5 RSA-7680

RSA-7680	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	15362	84735	77049	22.4	23239	92.5	86.5
10^{-5}	15362	11219	7411	15.9	23239	28.4	18.9

Table 5. RSA-7680 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is 192 bits.

3.6 RSA-15360

RSA-15360	**Old estimates**				**Current estimates**		
p_g	n_ℓ	n_p	quantum resources	**(expected) time**	n_ℓ	n_p	expected quantum resources
10^{-3}	30722	365259	4.8×10^6	96.5	46508	204	821
10^{-5}	30722	35556	76437	47.5	46508	72.5	143

Table 6. RSA-15360 security estimates. Here n_ℓ denotes the number of logical qubits, n_p denotes the number of physical qubits (in millions), *expected time* denotes the expected time (in hours) to break the scheme, and *quantum resources* (*expected quantum resources*) are expressed in units of megaqubitdays. The corresponding classical security parameter is 256 bits.

4 Public key cryptographic schemes – Elliptic-Curve Diffie-Hellman (ECDH)

In this Section we use the highly optimized circuits of [5] to produce our resource estimates, in particular the ones optimized for low T gate count.

4.1 256-bit modulus

NIST P-256	**Old estimates**		**Current estimates**	
p_g	n_ℓ	quantum resources	n_ℓ	quantum resources
10^{-3}	2330	67.7	2619	7.43
10^{-5}	2330	4.64	2619	0.89

Table 7. NIST P-256 curve security estimates. Here n_ℓ denotes the number of logical qubits, and *quantum resources* is expressed in units of megaqubitdays. The corresponding classical security parameter is 128 bits.

4.2 384-bit modulus

NIST P-384	**Old estimates**		**Current estimates**	
p_g	n_ℓ	quantum resources	n_ℓ	quantum resources
10^{-3}	3484	227	3901	10.0
10^{-5}	3484	12.8	3901	1.00

Table 8. NIST P-384 curve security estimates. Here n_ℓ denotes the number of logical qubits, and *quantum resources* is expressed in units of megaqubitdays. The corresponding classical security parameter is 192 bits.

4.3 521-bit modulus

NIST P-521	**Old estimates**		**Current estimates**	
p_g	n_ℓ	quantum resources	n_ℓ	quantum resources
10^{-3}	4719	606	5273	15.6
10^{-5}	4719	23.0	5273	1.56

Table 9. NIST P-521 curve security estimates. Here n_ℓ denotes the number of logical qubits, and *quantum resources* is expressed in units of megaqubitdays. The corresponding classical security parameter is 256 bits.

5 Symmetric key cryptographic ciphers (the AES family)

In this Section we used the highly optimized circuits of [17] to produce our resource estimates.

5.1 AES-128

AES-128	**Old estimates**			**Current estimates**		
p_g	s_q	n_ℓ	n_p	s_q	n_ℓ	n_p
10^{-3}	106.13	60553	2.71×10^9	101.66	15265	7.17×10^8
10^{-5}	101.73	10633	6.85×10^6	97.19	2545	1.77×10^6

Table 10. AES-128 security estimates. Here s_q denotes the quantum security parameter (in bits), n_ℓ denotes the number of logical qubits, and n_p denotes the number of physical qubits.

5.2 AES-192

AES-192	**Old estimates**			**Current estimates**		
p_g	s_q	n_ℓ	n_p	s_q	n_ℓ	n_p
10^{-3}	142.52	1030449	8.00×10^9	137.39	163793	2.93×10^9
10^{-5}	137.84	137649	2.66×10^7	132.81	23393	7.81×10^6

Table 11. AES-192 security estimates. Here s_q denotes the quantum security parameter (in bits), n_ℓ denotes the number of logical qubits, and n_p denotes the number of physical qubits.

5.3 AES-256

AES-256	**Old estimates**			**Current estimates**		
p_g	s_q	n_ℓ	n_p	s_q	n_ℓ	n_p
10^{-3}	175.68	1410681	1.92×10^{10}	170.49	218465	6.56×10^9
10^{-5}	170.79	157881	5.57×10^7	166.0	34865	1.61×10^7

Table 12. AES-256 security estimates. Here s_q denotes the quantum security parameter (in bits), n_ℓ denotes the number of logical qubits, and n_p denotes the number of physical qubits.

6 Conclusions and future directions

In this report we provide new estimates for the security of public key cryptographic systems based on the hardness of factoring large numbers (RSA) or on

the hardness of solving the discrete logarithm problem over elliptic-curves groups (ECC) using the latest advances in quantum cryptanalysis, such as windowed-arithmetic techniques [4, 5] and improved magic state distillation schemes [16]. We show a significant reduction (1 to 2 orders of magnitude) in the quantum resources required to break the schemes, which again stresses the importance of preparing for migration to quantum-resistant cryptographic systems [21].

In our prior report [15], the quantum resources required to break RSA-3072 (corresponding to 128 bit classical security parameter) was 641 megaqubitdays (for a realistic physical error rate $p_g = 10^{-3}$), whereas the quantum resources required to break ECC NIST-P256 (with the same 128 bit classical security parameter) was 67.7 megaqubitdays. In our current report, for the same choices of physical error rate, the quantum resources required to break RSA-3072 and ECC NIST-P256 are 4.03 megaqubitdays and 7.43 megaqubitdays, respectively.

It is sometimes claimed that "ECC is easier to break on a quantum computer than RSA", for similar choices of security parameters. This refers to the fact that (based on currently known classical attacks) n bits of classical security for ECC requires keys of length proportional to n, and Shor's algorithm requires on the order of n qubits to break those keys. For n bits of classical security, RSA requires keys of length proportional to n^3, and thus asymptotically on the order n^3 qubits are needed by Shor's algorithm. However, for a classical security parameter of $n = 128$, the gap between breaking ECC NIST-P256 and RSA-3072 is currently not very pronounced, although becomes more pronounced for higher classical security parameters. Resilience to classical attacks is likely a more important factor in deciding whether to use ECC or RSA in "hybrid" fashion with an appropriate quantum-safe algorithm.

In addition, we analyzed the security of symmetric ciphers (the AES family) in the light of novel developments [17] at the logical circuit layer (reduction in the number of T gates). In comparison with public key cryptographic systems, the security of AES schemes is less dramatically impacted, being only reduced by approximately 4–5 bits in comparison with our previous estimates [14].

As mentioned in our previous reports, estimating the strength of current cryptographic schemes against realistic quantum attacks is a moving target that depends on a variety of parameters, including fault-tolerant quantum error correction, circuit optimization and compilation, novel cryptanalysis results and improved quantum algorithms. Monitoring all these (future) advances remains our paramount priority.

Acknowledgements

We thank Craig Gidney for useful clarifications regarding window-arithmetic techniques employed in [4].

References

1. Fowler, A.G., Gidney, C.: Low overhead quantum computation using lattice surgery (2018), arXiv:1808.06709 [quant-ph]
2. Litinski, D.: A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery (2018), arXiv:1808.02892 [quant-ph]
3. Horsman, C., Fowler, A.G., Devitt, S., Meter, R.V.: Surface code quantum computing by lattice surgery. New Journal of Physics 14(12), 123011 (2012), http://stacks.iop.org/1367-2630/14/i=12/a=123011
4. Gidney, C., Ekerå, M.: How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits (2019), arxiv:1905.09749 [quant-ph]
5. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms (2020), arXiv:2001.09580 [quant-ph]
6. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. 79, 325–328 (Jul 1997), http://link.aps.org/doi/10.1103/PhysRevLett.79.325
7. Zalka, C.: Grover's quantum searching algorithm is optimal, e-print arXiv:quant-ph/9711070
8. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (Feb 1978), http://doi.acm.org/10.1145/359340.359342
9. Miller, V.S.: Use of elliptic curves in cryptography. In Hugh C. Williams, editor, Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings, volume 218 of Lecture Notes in Computer Science, pages 417–426. Springer, 1985 (1985)
10. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48, 203–209 (1987)
11. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing 26(5), 1484–1509 (1997), http://link.aip.org/link/?SMJ/26/1484/1
12. Gheorghiu, V., Mosca, M.: A resource estimation framework for quantum attacks against cryptographic functions (part 1) (2017), Global Risk Institute quantum risk assessment report, Sep. 2016 - Feb. 2017
13. Gheorghiu, V., Mosca, M.: A resource estimation framework for quantum attacks against cryptographic functions (part 2) (2017), Global Risk Institute quantum risk assessment report, Feb. 2017 - Aug. 2017
14. Gheorghiu, V., Mosca, M.: A resource estimation framework for quantum attacks against cryptographic functions (part 3) (2018), Global Risk Institute quantum risk assessment report, Aug. 2017 - Feb. 2018
15. Gheorghiu, V., Mosca, M.: A resource estimation framework for quantum attacks against cryptographic functions (part 4) (2018), Global Risk Institute quantum risk assessment report, Feb. 2018 - Aug. 2018
16. Gidney, C., Fowler, A.G.: Efficient magic state factories with a catalyzed $|CCZ\rangle$to 2$|T\rangle$ transformation. Quantum 3, 135 (Apr 2019), https://doi.org/10.22331/q-2019-04-30-135
17. Langenberg, B., Pham, H., Steinwandt, R.: Reducing the cost of implementing aes as a quantum circuit. Cryptology ePrint Archive, Report 2019/854 (2019), https://eprint.iacr.org/2019/854
18. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, 5th edn. (2000)
19. Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: Surface codes: Towards practical large-scale quantum computation. Phys. Rev. A 86, 032324 (Sep 2012), http://link.aps.org/doi/10.1103/PhysRevA.86.032324
20. Bravyi, S., Haah, J.: Magic-state distillation with low overhead. Phys. Rev. A 86, 052329 (Nov 2012), http://link.aps.org/doi/10.1103/PhysRevA.86.052329

References

21. Mosca, M.: Cybersecurity in an era with quantum computers: will we be ready? Cryptology ePrint Archive, Report 2015/1075 (2015), https://eprint.iacr.org/2015/1075