Quantum Fully Homomorphic Encryption With Verification
(extended abstract)

Full version available at http://www.florianspeelman.nl/vqfhe-qcrypt.pdf

Gorjan Alagic[1], Yfke Dulek[2], Christian Schaffner[2], and Florian Speelman[1]

[1] *QMATH, Department of Mathematical Sciences, University of Copenhagen*
[2] *CWI, QuSoft, and University of Amsterdam*

Introduction

The 2009 discovery of fully-homomorphic encryption (FHE) was a major breakthrough in classical cryptography [13]. FHE enables computation on encrypted data even by parties that do not hold the decryption key; crucially, the input, the output, and all intermediate states of the computation remain hidden from the computing party. The importance of FHE stems both from its obvious applications (e.g., secure cloud computing) and from its wide-ranging connections to other cryptographic tasks, such as secure two-party computation, non-interactive zero-knowledge proofs, and indistinguishability obfuscation [3, 12, 14]. In fact, the breadth of its usefulness has led some to dub FHE "the swiss army knife of cryptography"[3].

Recent progress on constructing quantum computers has led to theoretical research on "cloud-based" quantum computing (see, e.g., [8]). A recently-constructed quantum fully-homomorphic encryption (QFHE) scheme shows that this task can be performed with only a single round of interaction [10]. The discovery of (leveled) QFHE raises an important question: do the aforementioned classical applications of FHE have suitable quantum analogues? As it turns out, these applications require an additional property which is simple classically, but non-trivial quantumly. That property is *verification*: the ability of the user to check that the final ciphertext produced by the server is indeed the result of a particular computation, homomorphically applied to the initial user-generated ciphertext. In the classical case, this is a simple matter: the server makes a copy of each intermediate computation step, and provides the user with all these copies. In the quantum case, such a "transcript" would appear to violate no-cloning. In fact, one might reasonably suspect that the no-cloning theorem prevents non-interactive quantum verification *in principle*.

In this work, we show that verification of homomorphic quantum computations is in fact possible. We construct a new QFHE scheme which allows the server to generate a "computation log" which can certify to the user that a particular quantum computation was performed homomorphically on the ciphertext. The computation log itself is purely classical, and most (in some cases, all) of the verification can be performed on a classical computer. The addition of verification immediately leads to new applications, such as allowing users of a "quantum cloud service" to check that the server performed the desired quantum computation. As we show, verified QFHE (vQFHE) also leads to a simple new construction of quantum one-time programs (OTPs) from classical OTPs [9]. In this construction, the quantum OTP for a functionality Φ consists of a vQFHE ciphertext together with a classical OTP which performs verification of Φ. Finding other applications of vQFHE (including appropriate analogues of the above classical applications) is the subject of ongoing work.

Related Work

The first scheme for classical FHE was constructed by Gentry in 2009 [13]; a tremendous amount of work on this subject followed. We will use the scheme of Brakerski and Vaikuntanathan, which is based on the

Learning with Errors problem, and thus exhibits quantum security and decryption in $\mathbf{NC}^1$ [4]. Quantumly, partially-homomorphic (or partially-compact) schemes were constructed by Broadbent and Jeffery [6]. The first (leveled) QFHE scheme was recently constructed by Dulek, Schaffner and Speelman [10]. A parallel line of work has attempted to produce QFHE schemes with information-theoretic security [19, 16, 18, 15]. There has also been significant research on delegating quantum computation in the interactive setting (see, e.g., [1, 8, 11]). A notable approach via the so-called "trap code" was used to construct quantum OTPs from classical OTPs [9] and zero-knowledge proofs for QMA [5].

Summary of Results

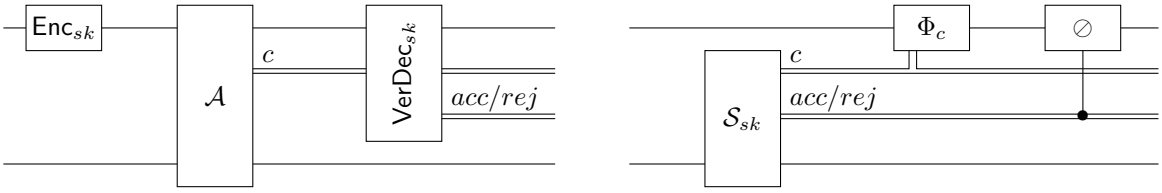
I. A new primitive: verified QFHE. A standard QFHE scheme consists of four poly-time quantum algorithms (QPTs): key-generation KeyGen, encryption Enc, evaluation Eval and decryption Dec [6, 10]. We define vQFHE similarly, except (i.) Eval provides an extra classical "computation log" output, and (ii.) decryption is now called VerDec and accepts both a ciphertext and a classical input, which is expected to contain a description of a circuit C and a computation log. Informally, correctness then demands that

$$\mathsf{VerDec}_{sk}^{C} \circ \mathsf{Eval}^{C} \circ \mathsf{Enc}_{sk} = C \tag{1}$$

for all keys sk and circuits C acting on the plaintext space. A crucial parameter is the relative difficulty of performing C and VerDec_{sk}^{C}. In a nontrivial scheme, the latter must be simpler. Our focus will be on the case where C is an arbitrary poly-size quantum circuit and VerDec_{sk}^{C} is almost entirely classical.

II. Security of verified QFHE. Security requires, apart from privacy of the input data, that a server deviating from the map Eval^{C} in (1) will cause VerDec_{sk}^{C} to reject. We give two formal versions of this security goal, which we sketch here.

1. **Semantic security (SEM-VER)**. An attacker is a QPT $\mathcal{A}$ which manipulates a ciphertext and then declares a circuit c; this defines a channel $\Phi_{\mathcal{A}} := \mathsf{VerDec} \circ \mathcal{A} \circ \mathsf{Enc}$ (see the left figure below). A simulator $\mathcal{S}$ neither receives nor outputs a ciphertext, but does declare a circuit; this defines a channel $\Phi_{\mathcal{S}}$ which first runs $\mathcal{S}$ and then runs a circuit on the plaintext based on the outputs declared by $\mathcal{S}$ (see the right figure below). We say that a vQFHE scheme is semantically secure (SEM-VER) if for all adversaries $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that $\Phi_{\mathcal{A}}$ and $\Phi_{\mathcal{S}}$ are computationally-indistinguishable channels.



2. **Indistinguishability (IND-VER)**. In this case, the QPT adversary $\mathcal{A}$ is participating in one of two protocols, based on a hidden coin flip b. For $b = 0$, this is a normal execution of vQFHE. For $b = 1$, this is a modified execution: we secretly swap out the plaintext $\rho_{\mathcal{A}}$ to a private register (replacing it with some fixed state), apply the desired circuit to $\rho_{\mathcal{A}}$, and then swap $\rho_{\mathcal{A}}$ back in; we then discard this plaintext if VerDec rejects the outputs of $\mathcal{A}$. Upon receiving the final plaintext, $\mathcal{A}$ must guess the bit b. The vQFHE scheme is indistinguishable (IND-VER) if, for all $\mathcal{A}$, the success probability does not exceed $1/2$ by more than a negligible amount.

3. **New relations between security definitions**. Our security definitions are related to existing notions of security for quantum encryption, as follows. If we restrict SEM-VER to the case where the circuit must be empty, we recover the (computational) definition of quantum authentication [11, 7]. Our definitions also generalize computational secrecy for quantum encryption: SEM-VER is a generalization of SEM [2] and IND-VER is a generalization of IND [6]. We are also able to show the following generalization of the equivalence between SEM and IND shown in [2].

Theorem 1. *A vQFHE scheme satisfies SEM-VER if and only if it satisfies IND-VER.*

III. A scheme for vQFHE for poly-size quantum circuits. Our main result is a vQFHE scheme which admits verification of arbitrary polynomial-size quantum circuits. The main technical ingredients in the construction are (i.) classical fully-homomorphic encryption (FHE) with $\mathbf{NC}^1$ decryption [4], (ii.) the trap code for computing on authenticated quantum data [17, 9, 7], and (iii.) the "garden-hose gadgets" from the first QFHE scheme [10]. Our vQFHE scheme is called TrapTP, as it is an extension of the teleportation-based QFHE scheme TP from [10]. A brief sketch is as follows:

1. **Key Generation (KeyGen).** We generate keys for the trap code and the classical FHE scheme, as well as some encrypted auxiliary states which are sent to Eval as an "evaluation key" (see Eval below).

2. **Encryption (Enc).** We encrypt each qubit of the plaintext using the trap code: we encode the qubit in m physical qubits using a quantum error-correcting code, we attach m-many $|0\rangle$ qubits and m-many $|+\rangle$ qubits, we apply a permutation $\pi \in S_{3m}$ of the qubits, and finally apply a $3m$-qubit quantum one-time pad. We also encrypt the trap code keys using the FHE scheme.

3. **Evaluation (Eval).** Paulis and CNOT can be evaluated as usual in the trap code; the keys are updated using FHE evaluation. To measure a qubit, we measure all ciphertext qubits and place the outcomes in the log. To apply P or H, we use encrypted magic states (provided in the evaluation key) plus the aforementioned gates. Finally, applying T requires both a magic state and an encrypted "garden-hose gadget." The gadget helps with applying a P-gate conditioned on the (encrypted) outcome of a measurement, which is required for the T-gate magic state circuit. In addition to all of the measurement outcomes, the log contains a transcript of all the classical FHE computations.

4. **Verified decryption (VerDec).** We check the correctness and consistency of the classical FHE transcript, the measurement outcomes, and the claimed circuit. The result of this computation is a set of keys for the trap code, which are correct provided that Eval was performed honestly. We decrypt using these keys and output either a plaintext or reject.

The scheme described above is *compact*, in the sense that the number of elementary quantum operations performed by VerDec scales only with the size of the plaintext, and *not* with the size of the circuit performed via Eval. While VerDec does perform a classical computation which scales with the size of the circuit, this is reasonable; after all, VerDec must receive the circuit as input. We remark that, when the output of the homomorphic computation is classical (as it is, e.g., in Shor's algorithm), then VerDec is completely classical.

Theorem 2 (Main result, informal)**.** *The* TrapTP *scheme outlined above satisfies IND-VER security and IND-CPA secrecy [6, 2]. Moreover, if we hard-code the appropriate inputs of* VerDec *so that it always verifies the blank circuit, then the resulting encryption scheme is authenticating [11, 7].*

IV. Application: quantum one-time programs. Recall that a one-time program (or OTP) is an idealized device which implements a circuit, but self-destructs after the first use. Classical OTPs are impossible without hardware assumptions, since circuits are easily copied. Broadbent, Gutoski and Stebila investigated the possibility of OTPs in the quantum setting [9]. They showed that, contrary to what one might expect from no-cloning, quantum states cannot be used to construct OTPs. They did show that quantum OTPs (i.e., OTPs that implement quantum circuits) can be built from classical OTPs [9].

As a first application of vQFHE, we give another simple construction of quantum OTPs. Our construction is somewhat weaker than that of [9], since vQFHE requires a computational assumption; on the other hand, it is conceptually very simple and serves to demonstrate the power of verification. In our construction, the quantum OTP for a quantum circuit C is simply a (vQFHE) encryption of C together with a classical OTP for verifying the universal circuit. To use the resulting quantum OTP, the user attaches their desired input, homomorphically evaluates the universal circuit, and then plugs their computation log into the classical OTP to retrieve the final decryption keys.

References

[1] D. Aharonov, M. Ben-Or, and E. Eban. Interactive Proofs For Quantum Computations. *ArXiv e-prints*, October 2008.

[2] Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael StJules. Computational security for quantum encryption. In *9th International Conference on Information Theoretic Security (ITICS), to appear.*, 2016.

[3] Boaz Barak and Zvika Brakerski. Windows on theory: The swiss army knife of cryptography, 2012. URL https://windowsontheory.org/2012/05/01/the-swiss-army-knife-of-cryptography/.

[4] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4571-4. doi: 10.1109/FOCS.2011.12. URL http://dx.doi.org/10.1109/FOCS.2011.12.

[5] A. Broadbent, Z. Ji, F. Song, and J. Watrous. Zero-knowledge proof systems for QMA. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40, Oct 2016. doi: 10.1109/FOCS.2016.13.

[6] Anne Broadbent and Stacey Jeffery. *Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity*, pages 609–629. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. ISBN 978-3-662-48000-7. doi: 10.1007/978-3-662-48000-7_30. URL http://dx.doi.org/10.1007/978-3-662-48000-7_30.

[7] Anne Broadbent and Evelyn Wainewright. Efficient simulation for quantum message authentication. *arXiv preprint arXiv:1607.03075*, 2016.

[8] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 517–526. IEEE, 2009.

[9] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Advances in Cryptology–CRYPTO 2013*, pages 344–360. Springer, 2013.

[10] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology – CRYPTO 2016*, pages 3–32. Springer Berlin Heidelberg, 2016. ISBN 978-3-662-53015-3. doi: 10.1007/978-3-662-53015-3_1. URL http://dx.doi.org/10.1007/978-3-662-53015-3_1.

[11] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology–CRYPTO 2012*, pages 794–811. Springer, 2012.

[12] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49, Oct 2013. doi: 10.1109/FOCS.2013.13.

[13] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536440. URL http://doi.acm.org/10.1145/1536414.1536440.

[14] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *J. Cryptol.*, 28(4): 820–843, October 2015. ISSN 0933-2790. doi: 10.1007/s00145-014-9184-y. URL http://dx.doi.org/10.1007/s00145-014-9184-y.

[15] M. Newman and Y. Shi. Limitations on Transversal Computation through Quantum Homomorphic Encryption. *ArXiv e-prints*, April 2017.

[16] Y. Ouyang, S.-H. Tan, and J. Fitzsimons. Quantum homomorphic encryption from quantum codes. *ArXiv e-prints*, August 2015.

[17] Peter W. Shor and John Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85:441–444, Jul 2000. doi: 10.1103/PhysRevLett.85.441. URL http://link.aps.org/doi/10.1103/PhysRevLett.85.441.

[18] S.-H. Tan, J. A. Kettlewell, Y. Ouyang, L. Chen, and J. F. Fitzsimons. A quantum approach to homomorphic encryption. *Scientific Reports*, 6:33467, September 2016. doi: 10.1038/srep33467.

[19] Li Yu, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Limitations on information-theoretically-secure quantum homomorphic encryption. *Phys. Rev. A*, 90:050303, Nov 2014. doi: 10.1103/PhysRevA.90.050303. URL https://link.aps.org/doi/10.1103/PhysRevA.90.050303.