Continuous Variable Quantum Computing on Encrypted Data

Kevin Marshall,[1] Christian S. Jacobsen,[2] Clemens Schafermeier,[2]
Tobias Gehring,[2] Christian Weedbrook,[3] and Ulrik L. Andersen[2]

[1] *Department of Physics, University of Toronto, Toronto, M5S 1A7, Canada*
[2] *Department of Physics, Technical University of Denmark, Fysikvej, 2800 Kgs. Lyngby, Denmark*
[3] *QKD Corp., 60 St. George St., Toronto, M5S 1A7, Canada*

In today's era of cloud and distributed computing, protecting a client's privacy is a task of highest priority. Performing computations in the cloud on encrypted data rather than on plain text is a promising tool to achieve this goal. Here, we report about a continuous variable protocol for performing computation on encrypted data on a quantum computer. We theoretically investigate the protocol and present a proof-of-principle experiment implementing displacements and squeezing gates. We demonstrate losses of up to 10 km both ways between the client and the server and show that security can still be achieved. Our approach offers a number of practical benefits, which can ultimately allow for the potential widespread adoption of this quantum technology in future cloud-based computing networks.

Clouds provide computing and data resources on-demand. A client can easily perform computationally demanding tasks in the cloud with scalable resources. Cloud services are third-party operated, for instance by companies like Amazon, and share their available resources among their customers. The cloud contains, in principle, various types of sensitive data for which security and privacy is important. For example, an individual's personal data (such as medical records and credit card information), the trade secrets and intellectual property of multinational corporations, and sensitive government information (e.g., the CIA bought cloud space from Amazon). Securing a client's privacy in the cloud is therefore one of the most important security challenges today.

A current solution to this problem is fully homomorphic encryption [1], i.e. arbitrary computation on ciphertext. Since the program completely runs on encrypted data which never needs to be decrypted, the program can be run in an untrusted environment like a cloud without revealing sensitive information. However, the best known implementations of fully homomorphic encryption are impractical for today's computers [1–4].

Quantum computers will be a vital part of future cloud services with a promising speed up of algorithms. It has been shown that perfectly secure, deterministic fully homomorphic quantum computation is only possible with exponential overhead in the size of the encrypted state with the size of the allowed operations [5]. One can, however, relax the requirements of quantum homomorphic encryption by allowing further rounds of interaction between the client and server in the cloud [6]. Seminal work by Broadbent et al. [7] showed that in the quantum world it is even possible to hide not only the input and the final result, but also the quantum program itself. Unfortunately, hiding all three, input, program and result, places stringent requirements on the experimental realization of such protocols including a lot of classical and quantum communication. Giving up the idea of hiding the program but agreeing on it beforehand, however, solves this issue [8]. Such a protocol can also be termed quantum computing on encrypted data.

In this paper we report about quantum computing on encrypted data using continuous variables. In our protocol the input state is encrypted using displacement operations to hide it from the server. We show that the result of the computation can be decrypted by displacement operations as well, except for the non-Gaussian cubic phase gate U_3 [9, 10], where an additional squeezing operation is needed. We furthermore demonstrate quantum computing on encrypted data in a proof-of-principle experiment implementing the two displacement gates, X and Z, as well as the U_2 squeezing gate and using a lossy channel simulating loss of up to 10 km in equivalent fiber length.

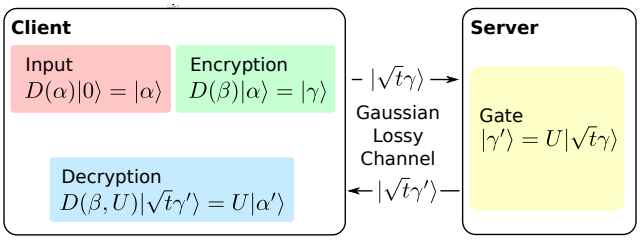

Figure 1. Protocol for quantum computing on encrypted data. Input: A displaced vacuum state is prepared. Encryption: A random displacement is applied to the initial state as an encryption procedure. Channel: The state is transmitted over a Gaussian lossy channel to the server (transmission t). Gate: The server applies the desired unitary. Channel: The state is sent back over the Gaussian lossy channel to the client. Decryption: The client applies a decryption operation to retrieve the final output state.

Our protocol for quantum computing on encrypted continuous variables comprises four stages, see Fig. 1. First, an input stage, where the input quantum state, a displacement in our case, is generated. In the second stage the input state is encrypted by further displacing it with a displacement randomly chosen from a Gaus-

sian distribution. The encrypted quantum state is then send to the server through a lossy quantum channel. The server applies one or a series of gate operations before sending the result back to the client. In the fourths and last stage the client decrypts the result of the computation.

To discuss suitable decryption operations we first define the Heisenberg-Weyl operators [11] $X(Q) = \exp(-iQ\hat{p})$ and $Z(P) = \exp(iP\hat{q})$, as well as the displacement operator $D(\alpha) = \exp(\alpha\hat{a}^\dagger - \alpha^*\hat{a})$ where $\hat{q}, \hat{p}$ are the canonical amplitude and phase operators, respectively, which obey Heisenberg's uncertainty relation $[\hat{q}, \hat{p}] = i$. The annihilation and creation operators are denoted by $\hat{a}, \hat{a}^\dagger$ respectively, and are defined by $\hat{a} = (\hat{q}+i\hat{p})/\sqrt{2}$ and its adjoint.

To achieve universal quantum computation on encrypted data we need to show that the results of the gates in the set $\mathcal{G} = \{X(Q), Z(P), U_2(T), U_3(T), F, C_Z\}$ where $U_k(T) = \exp(iT\hat{q}^k)$, $F = \exp\left[\frac{i\pi}{4}\left(\hat{q}^2 + \hat{p}^2\right)\right]$, and $C_Z = \exp(i\hat{q}_1 \otimes \hat{q}_2)$ can be decrypted with an appropriately chosen operation. Hereby $X(Q)$ and $Z(P)$ are displacement gates, $U_2(T)$ is the squeezing gate, $U_3(T)$ is the cubic phase gate, F is the Fourier gate and C_Z is the controlled not gate. While U_3 is a non-Gaussian gate which is difficult to implement, all the other gates are Gaussian. Table I lists the decryption operations for the individual gates. Except for U_3, all of these operations have decryption operators that correspond to displacements.

Gate	Decryption
$Z(T)$	$X(-Q)Z(-P)$
$X(T)$	$X(-Q)Z(-P)$
$U_2(T)$	$X(-Q)Z(-2QT-P)$
$U_3(T)$	$X(-Q)Z(3Q^2T-P)U_2(-3QT)$
F	$X(P)Z(-Q)$
C_Z	$X_1(-Q_1)Z_1(-Q_2-P_1)\otimes X_2(-Q_2)Z_2(-Q_1-P_2)$

Table I. The decryption operations corresponding to each gate, up to a phase, for the encryption operation $D(Q,P)$ for single mode gates and $D_1(Q_1,P_1)D_2(Q_2,P_2)$ for two-mode gates.

In our proof-of-principle experiment we tested quantum computing on encrypted data by performing the two displacements gates, X and Z, and the squeezing gate U_2. The experimental setup of the proof-of-principle experiment is shown Fig. 2. The input state was generated by applying an amplitude and phase modulation via electro-optical modulators to a 1064 nm laser beam. At the same modulators we performed the encryption operation by driving the two modulators with noise from two independent Gaussian noise sources. The encrypted state is then send to the server via a lossy quantum channel with the loss applied by a half-wave plate and a polarizing beam splitter. For the X and Z gates the server performed another displacement, while the squeezing gate was implemented by injecting the quantum state received by the server into a squeezed-light source. The squeezed-light source was a parametric-down conversion source with a potassium-titanyl-phosphate crystal in an optical cavity. Afterwards the resulting quantum state was send back to the client via a lossy channel with the same optical loss. The decryption of the resulting quantum state took place by displacing it using another set of electro-optical modulators. The noise from the Gaussian white noise generators used for encryption was thereby applied to the modulators according to Tab. I.

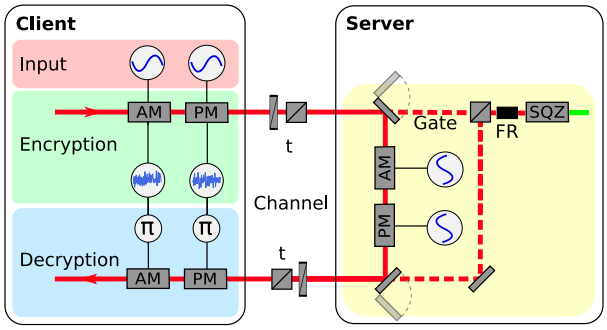

Figure 2. Schematic representation of the experimental implementation. All phase space displacements were performed using electro-optical modulations of the amplitude (AM) and phase (PM) field quadratures at 10.5 MHz. Two Gaussian noise generators were used to encrypt and decrypt the input and output states, respectively. The server could either implement the Z and X gates by performing a displacement by using two more electro-optical modulators or a squeezing operation. The squeezing operation was implemented by feeding the encrypted input state into a linear squeezed-light source (SQZ) by means of a polarizing beam splitter and a Faraday rotator (FR). The lossy channels with transmission t were implemented by variable beam splitters based on half-wave plates and polarizing beam splitters.

Figure 3 shows the effectiveness of the encryption. We measured the mutual information of the client's input state and the encrypted state received by the server. For this measurement we used Gaussian white noise to generate the input state which resembled an ensemble of coherent states. The mutual information was calculated from the data acquired by recording the Gaussian white noise used for the coherent input state generation and a homodyne measurement of the encrypted state. The solid line shows a theory model according to

$$I(\text{server}_{\text{enc}}:\text{client}_{\text{in}}) = \frac{1}{2}\ln\left(1+\frac{V_{\text{in}}}{V_{\text{enc}}}\right), \qquad (1)$$

where V_{in} and V_{enc} are the variances of the input state ensemble and the encryption noise, respectively. The plot shows the effect of a finite encryption variance, finally given by energy constraints. To reveal as little as possible

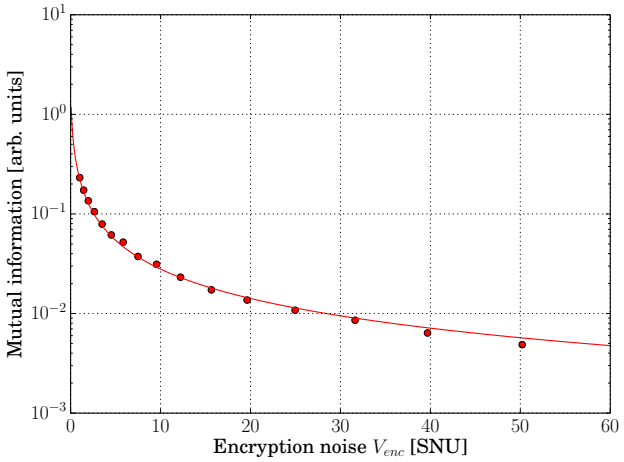

Figure 3. Effectiveness of encryption. The mutual information $I(\text{server}_{\text{enc}} : \text{client}_{\text{in}})$ for a coherent state chosen according to a Gaussian alphabet with variance $V_{\text{in}} = 0.6$ shot noise units, which is then encrypted with a varied encryption variance V_{enc} before being sent to the server. For a fixed distribution of input states the plot shows how an increased encryption noise decreases the server's knowledge of the inputs. Error bars are smaller than the point size. The transmittivity of the channel was set to $t = 1$ for this measurement.

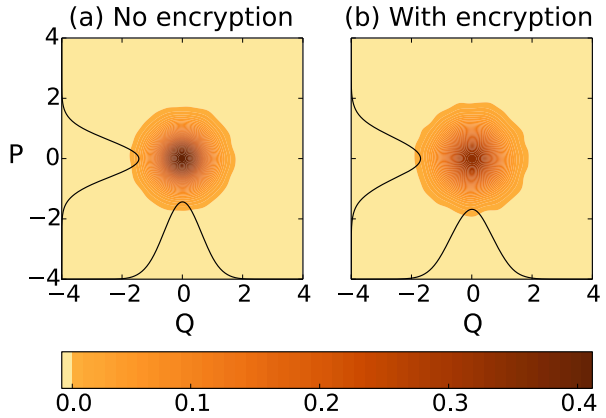

Figure 4. Reconstructed Wigner functions of the ensemble of result states of a computation using displacement gates with (a) and without (b) encryption. The input state ensemble had a variance of $V_{\text{in}} = 0.3$ shot-noise units, the gate operation ensemble was $V_{\text{gate}} = 0.6$ shot-noise units and the encryption noise had a variance of $V_{\text{enc}} = 31$ shot-noise units.

about the input state, the encryption noise variance has to be chosen as large as possible.

As an example computation we show the tomographically reconstructed Wigner functions of the ensemble of result states of a displacement gate operations. The Wigner functions shown in Fig. 4 were reconstructed using a maximum likelyhood method based on homodyne measurements with a scanned local oscillator phase. Both, the input states as well as the gate operations were produced by white noise generators to create ensembles of possible states and gate operations. Figure 4a shows the resulting ensemble without encryption, while for Fig. 4b the encryption was switched on. The resulting ensemble after decryption has some additional noise due to experimental imperfections in the decryption operation. The fidelity between the ensemble without encryption and with encryption was calculated to 99.2 % showing that the computation on encrypted data works almost as good as on plain text.

In conclusion we have developed a protocol for quantum computing on encrypted continuous variables and demonstrated a subset of the gates experimentally. Our protocol requires a baseline of two uses of a quantum channel, one to transfer the input from the client to the quantum server, and the other to transfer back the result. Only for the cubic phase gate U_3 one additional round of classical communication in each direction and one additional use of the quantum channel is needed while Gaussian gates can be implemented with no communication cost.

[1] M. Naehrig, K. Lauter, and V. Vaikuntanathan, in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11 (ACM, New York, NY, USA, 2011) pp. 113–124.

[2] Z. Brakerski and V. Vaikuntanathan, in *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11 (IEEE Computer Society, Washington, DC, USA, 2011) pp. 97–106.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (ACM, 2012) pp. 309–325.

[4] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," Cryptology ePrint Archive, Report 2012/099 (2015), version: 20150103:190644.

[5] L. Yu, C. A. Pérez-Delgado, and J. F. Fitzsimons, Phys. Rev. A **90**, 050303 (2014).

[6] A. M. Childs, Quantum Info. Comput. **5**, 456 (2005).

[7] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on* (2009) pp. 517–526.

[8] K. Fisher, A. Broadbent, L. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. Resch, Nat. Commun. **5**, 3074 (2014).

[9] P. Marek, R. Filip, and A. Furusawa, Phys. Rev. A **84**, 053802 (2011).

[10] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook, Phys. Rev. A **91**, 032321 (2015).

[11] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Rev. Mod. Phys. **84**, 621 (2012).