

METHOD AND SERVER FOR DELEGATED QUANTUM COMPUTING USING A HARDWARE ENCLAVE

Technical Field

5 [01] The invention relates to delegated quantum computing. In particular, it allows blind delegation to a quantum-enabled server over a secured communication channel, so that the server cannot learn some information received from the clients.

Background

10 [02] Quantum computers are still difficult to develop and costly pieces of equipment. It is therefore highly probable that quantum computers will only be accessible over communication networks to remote clients, as are supercomputers today. They will provide computing services according to a client/server scheme, wherein the clients send requests and receives results
15 from the server.

[03] Delegated Quantum Computation (DQC) allows a client with limited quantum power to delegate a computation to a quantum-enabled server, in order to obtain the outcome of a computation. Generally speaking, the scenarios of delegated quantum computation involve three main stages:
20 preparation stage, computation stage and output correction stage.

[04] A strong requirement for such scheme is the protection of the data provided by the client to the server. A main question asks whether the client can securely delegate a quantum computation to an untrusted server. In particular, a blindness characteristic shall be ensured, i.e. the fact that the
25 computations and the data of the client remain secret, even from the server itself.

[05] Different proposals have been made regarding a Blind Delegated Quantum Computation.

[06] The concept was first introduced in Andrew M. Childs, “*Secure assisted quantum computation*” in *Quantum Information and Computation*, 5.6, ISSN15337146, arXiv: quant-ph/0111046, and, further developed, in particular, in Anne Broadbent, Joseph Fitzsimons and Elham Kashfi,
5 “*Universal Blind Quantum Computation*” in *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, October 2009, pp. 517-526, arXiv: 0807.4154

[07] These schemes assume a quantum communication link between clients and the server. They therefore require the clients to be quantum-
10 enabled and thus embed the needed, and costly, quantum equipments. They might also require a conversion from systems used by the client for communicating to systems used by the server for computation, raising further technological issues.

[08] First attempts have been proposed to have a classical, or legacy, client,
15 i.e. not embedding any quantum features, exchanging with the server through a classical non-quantum communication link. For instance, Vedran Dunjko and Elham Kashefi, “*Blind Quantum Computing with two almost identifiable states*” in arXiv: 1604.01586, addresses the question of “whether a fully classical Alice can securely delegate a quantum computation to an
20 untrusted server Bob. The main two flavours of security one is interested in are *blindness* – meaning Alice’s computation remains private and hidden from Bob – and *verifiability* – meaning Alice has a mechanism which ensures the declared output of the computation is indeed correct.”

[09] However, as of today, the work proposing blind quantum computing
25 with remote classical clients induce large overhead in the computation.

Summary

[10] In particular, according to embodiments, a method and a server are proposed based on the use of a hardware enclave, enabling classical

communication with a classical client, and the blindness characteristics, including with regard to the server itself.

[11] In a first example embodiment, a quantum-enabled server is provided. It comprises an enclave, a qubit source, and quantum-computing means, said enclave comprising means for:

- receiving information from a remote client through a secured communication channel;
- determining transformation data from said information;
- transforming at least one qubit received from said qubit source, according to transformation data ;
- providing the at least one transformed qubits to said quantum-computing means.

[12] This server may comprise other features, alone or in combination, such as:

- said transformation data are contained in said information and adapted to transform said at least one qubit;
- said information represents application code configured for being executing within said enclave and for generating said transformation data.
- said transformation data are modulation data, and said enclave comprises means for modulating at least one qubit received from said qubit source, according to modulation data
- said enclave comprises
 - a classical secured enclave comprising a trusted execution environment, a memory comprising an operating system and adapted to host an application code, and configured to

generate said transformation data as digital transformation data;

- a digital-to-analog converter for converting said digital transformation data into analog digital data;
- 5 – transforming means for transforming a degree of freedom of said at least one qubit according to analog transformation data;
- the digital-to-analog convertor is encapsulated into a cage preventing any electromagnetic signals been read from outside of said enclave;
- said information is ciphered and deciphered at said enclave at receipt;
- 10 - said enclave sends an attestation back to said client to if said Trusted Execution Environment successfully runs said classical enclave;
- said enclave is configured to:
 - determine test qubits among said at least one qubit;
 - transform said test qubits according to an unknown
 - 15 transformation;
 - instruct said quantum-computing means to measure the transformed test qubits according to a given base;
 - verify if the result of the measurements is compatible with said unknown transformation; and
 - 20 – trigger an error if a number of failed verification is above a given threshold;
- The quantum-enabled server is configured to retrieve a public key from said information, encrypt said transformation data with said public key, and transmit the encrypted transformation data to said
- 25 client.

[13] In a second example embodiment, a method is provided, for delegating quantum computing to a quantum-enabled server, comprising:

- sending information from a remote client to an enclave inside said quantum-enabled server through a secured communication channel;
- 5 - determining, at said enclave, transformation data from said information;
- transforming, at said enclave, at least one qubit received from said qubit source, according to said transformation data ;
- 10 - providing the at least one transformed qubits to said quantum-computing means.

Brief Description of the Figures

[14] Some embodiments are now described, by way of example only, and with reference to the accompanying drawings, in which :

- The Figure 1 schematically illustrates a system comprising a quantum-enabled server, a client and a communication channel, according to embodiments of the invention.

– The Figure 2 schematically illustrates a functional architecture of quantum-enabled server according to embodiments of the invention.

Description of Embodiments

- 20 [15] In the following, the terms “classical” and “legacy” refer, interchangeably, to any communication means or computing means that is not quantum-enabled.

[16] In reference to FIG. 1, a quantum-enabled server S and a client C can communicate through a communication channel T.

- 25 [17] The quantum-enabled server S comprises apparatus able to operate on quantum.

[18] Quantum computation implies the manipulation of data in form of quantum bits or “qubits”. Whereas in classical computation a bit of information is used to represent only one of two possible logical states, namely “1” or “0”, in quantum computation, a qubit can represent both logical states simultaneously as a superposition of quantum states.

[19] In practice, qubits can be coded by using degrees of freedom of different physical particles. Degrees of freedom are a physical property of physical systems, which can be described by quantum mechanics.

[20] In particular, qubits can be physically implemented by various supports including photons, coherent state of light, electrons, nucleus, optical lattices, Josephson junctions for superconducting qubits, etc. A non-exhaustive list may be found, for instance, on Wikipedia: <https://en.wikipedia.org/wiki/Qubit>

[21] Accordingly, degrees of freedom depend on the physical support and comprise phase, phase differences, frequency, polarization, time localization of photons. Also, spins of electrons, superconducting charge, electron number, etc. can also be used.

[22] Qubits can be written as a vector in a Hilbert vector space of dimension d .

[23] In a dimension $d=2$, the qubit can be defined by a basis consisting of two states, which are denoted $|0\rangle$ and $|1\rangle$. Thus, the state $|\alpha\rangle$ of the qubit can be represented as:

$$|\alpha\rangle = a|0\rangle + b|1\rangle$$

where a and b are complex number coefficients.

[24] The qubit can store information as a combination of 0 and 1, using different values of a and b . However, a measurement of the qubit will cause it to project onto $|0\rangle$ or $|1\rangle$ state and return the result 0 or 1 respectively. The probabilities of returning these values are $|a|^2$ and $|b|^2$ respectively

[25] A quantum-enabled apparatus comprises means to operate on the qubits. Such apparatuses comprise sources of qubits, receivers, and any other devices able to operate qubits, e.g. to transform their states.

5 [26] The quantum-enabled server S comprises such apparatus and in particular a quantum source S_S , transformation means S_{mod} and further quantum-computing means S_M . These quantum-enabled apparatuses shall be adapted to operate on a same quantum technology.

[27] According to embodiments of the invention, the client is a classical client, meaning it is a computing device having no quantum-enabled mean.
10 It can be a legacy computer or any other platform or device comprising classical communication means.

[28] Accordingly, the communication channel can also be a legacy communication channel. This comprises local communication network, when the client C and the server S are located in a close vicinity, like
15 Ethernet links, Wi-Fi, Bluetooth, etc. This also comprises wide-area networks, like in particular the Internet, and any mixes of communication networks of different technologies.

[29] Figure 1 shows an example wherein only one client is depicted. However, embodiments of the invention apply as well in situations where a
20 plurality of clients can communicate with a server S. In the following, one assumes that interactions between each client and the server can be performed in an independent way, so that only interactions between one client and a server will be described without any loss of generality.

[30] Figure 2 depicts with more details a functional architecture of
25 embodiments of such a quantum-enabled server S.

[31] According to embodiments, the quantum-enabled server S comprises an enclave S_E that comprises itself means for receiving information from the client C through the communication channel T.

- [32] According to embodiments, the communication channel T is a secured channel. Security can be ensured by cryptographic mechanisms, wherein transmitted information is cyphered at client's side before transmission and deciphered at enclave's side at receipt. This allows that the communication
5 between the client and the enclave are private and secured, even from the server S itself (i.e. outside of the enclave).
- [33] Various security schemes known in the art can be used, including RSA, DSA (Digital Signature Algorithm), AES (Advanced Encryption Standard)...
- [34] Still another scheme can NTRU. The NTRUEncrypt public key
10 cryptosystem, also known as the NTRU encryption algorithm, is a lattice-based alternative to RSA and ECC (Elliptic-Curve Cryptography) and is based on the shortest vector problem in a lattice, which is not known to be breakable using quantum computers.
- [35] According to embodiments, instead of generating a secret at its side,
15 the client instructs the server S to generate a random key. Embodiments of this process will be explained later. In particular, the client can transmit to the enclave S_E of the server an executable code enabling the generation of a random key inside the enclave. Then, this key can be encrypted and sent back to the channel through the authenticated communication channel T.
- [36] According to embodiments, then, only the transmission from the
20 enclave to the client may be encrypted. This may be sufficient for universal blind quantum computation. A property of this embodiment is to prevent malicious server from potentially reverse compiling the secret in the first place.
- [37] Furthermore, the server S further comprises a qubit source S_S and
25 quantum-computing means S_M . Other components may (not depicted on figure 2) can be further embedded inside the server S.

[38] In consequence, globally, the quantum-enabled server of the invention has a hybrid nature, since it embeds both means for quantum operations and means for classical operations.

5 [39] The enclave S_E enables the quantum-enabled server S to perform classical operations in collaboration with the client C , in a secured way. By “secured”, it is meant that no party apart the client C and the components inside the enclave S_E can access the content of the data.

10 [40] For doing so, both physical or logical isolation of the storage, computing and communication means can be enforced, in particular with regard to other components of the server S . Also, encryption mechanisms can be put in place, so that any malicious attempts to access data will result in getting an enciphered form, preserving the privacy of the real data.

15 [41] In other words, the enclave S_E can be functionally defined as a group of elements of the server, including components and information (code, data) that are isolated from the other elements of the server for preserving their privacy (i.e. so that none of the latter can access to their content).

20 [42] According to embodiments, the enclave S_E comprises a classical subpart, comprising a classical enclave S_{SE} , and quantum part dealing with quantum operations and comprising transformation means S_{Mod} configured to transform at least one qubit received from a qubit source according to transformation data. The enclave S_E can thus be called “quantum enclave”, or “Q enclave”, after its quantum characteristics.

25 [43] A digital-to-analog converter S_{DAC} forms sort of a junction between the two subparts and belongs to both to the digital realm, northbound, and to the analog realm, southbound.

[44] The classical enclave S_{SE} can be an enclave, also known as “secure enclave”, according to techniques known in the art, e.g. running on a Trusted Execution Environment, TEE.

[45] As the enclave S_E , it corresponds to a technique for creating and maintaining a secured, protected, or isolated partition or environment. It is a set of information and processing capabilities that are protected as a group.

5 [46] The classical enclave S_{SE} runs a dedicated microkernel and undergoes a secure boot process separate from the rest of the server S . It may thus be designed to be secured even if the server S is hacked and/or host malicious applications or Operating System (OS) components.

[47] Several implementations of such enclaves have already been proposed.

10 [48] Apple proposes for its iPhones an enclave based on dedicated hardware components including a coprocessor.

[49] Intel proposes an enclave called "Software Guard eXtension" (SGX) that is embedded inside the Intel microprocessor itself. This mechanism allows user-level as well as operating system code to define private regions
15 of memory, called enclaves, whose contents are protected and unable to be either read or saved by any process outside the enclave itself, including processes running at higher privilege levels. SGX involves encryption by the CPU of a portion of memory. The enclave is decrypted on the fly only within the CPU itself, and even then, only for code and data running from within the
20 enclave itself. The processor thus protects the code from being "spied on" or examined by other code. The code and data in the enclave utilize a threat model in which the enclave is trusted but no process outside it can be trusted (including the operating system itself and any hypervisor), and therefore all of these are treated as potentially hostile. The enclave contents are unable
25 to be read by any code outside the enclave, other than in its encrypted form.

[50] Documentations about Intel's SGX are available on the Intel website, and also on Wikipedia:

https://en.wikipedia.org/wiki/Software_Guard_Extensions

[51] Some academic literature has been published as well, like e.g. Schwarz, Michael; Weiser, Samuel; Gruss, Daniel; Maurice, Clémentine; Mangard, Stefan (2017). "Malware Guard Extension: Using SGX to Conceal Cache Attacks". arXiv:1702.08719

5 [52] According to embodiments, the server S is based on a Linux platform. The Intel's SGX enclave is available for Linux-based server, as for instance documented in "Intel® Software Guard Extensions SDK for Linux OS".

[53] According to embodiments, the classical enclave S_{SE} runs on a specific secured area in the main processor called a Trusted Execution Environment, TEE, which guarantees the integrity and the confidentiality of the code and
10 of the data of a running application inside the classical enclave S_{SE} .

[54] The Trusted Execution Environment, TEE, has been first defined by the Open Mobile Terminal Platform (OMTP) standards, in "OMTP Hardware Requirements And Defragmentation", [gsma.org](https://www.gsma.org). It is also documented on
15 Wikipedia, at https://en.wikipedia.org/wiki/Trusted_execution_environment

[55] The CPU with the micro-architecture of the TEE integrates a set of instructions that allows user-level applications, as well as OS (Operating System) level code to define private region of memory.

[56] The classical enclave S_{SE} provides thus hardware resources (based on
20 a TEE), OS resources (running on the TEE), as well as a secured memory able to store an application S_{app} and related data.

[57] The application S_{app} is configured to control transformation means S_{mod} embedded in the enclave S_E .

[58] The application may be natively stored and installed in the memory of
25 the classical enclave. In variants, several applications may be installed, for instance, at startup of the classical enclave. The information received from the client C may then contain an identifier of the particular application S_{app} to trigger.

[59] According to other embodiments, the information received from the client C represents the application code of the application S_{app} .

[60] The code may be transmitted over the communication channel as binary code. The binary code shall be adapted for being directly executed
5 by the OS and hardware of the classical enclave S_{SE} . The latter can then start the execution of the provided application S_{app} , for instance at receipt.

[61] The information transmitted by the client C may also comprise parameters influencing the behaviour of the application S_{app} . In particular, the parameters can be passed as parameters of the application considered
10 as a software function.

[62] According to another embodiments, the information received from the client C comprises directly transformation data. In this case the application S_{app} may be configured to receive these transformation data and forward them, with some potential data formatting) to the digital-to-analog converter,
15 S_{DAC} .

[63] According to embodiments, the classical enclave S_{SE} may be adapted to implement different approaches. In particular, it may contain an application for receiving and forwarding transformation data and may also receive application code(s) as previously described. This way, the client can
20 control the transformation of qubits in several ways.

[64] Accordingly, the invention allows the client C to remotely define the application S_{app} which behaviour aims in transforming the qubits that will be provided to the quantum computing means of the server. Accordingly, the client can remotely control the transformation of the qubits provided to the
25 server.

[65] The application S_{app} generates transformation data θ , when executed on the classical enclave S_{SE} . These transformation data, at this step, are digital transformation data.

[66] The digital transformation data are provided to a digital-to-analog converter S_{DAC} . The digital-to-analog converter converts the received digital transformation data θ into analog transformation data $\bar{\theta}$.

5 [67] Different types of digital-to-analog converters may be used in the context of the invention. In particular, FPGA (Field Programmable Gate Arrays) DAC can be used.

[68] According to embodiments, the digital-to-analog converter S_{DAC} is encapsulated into a cage preventing any electromagnetic signals been read from outside of the enclave S_E . In particular, this prevent any malicious
10 devices (oscilloscopes...) to measure any transformation schemes of the analog transformation data $\bar{\theta}$ so as to decode the transformation data θ . The cage may be a continuous or meshed structure made of conductive materials like metal.

[69] According to embodiments, the digital transformation data θ are also
15 provided back to the client C, preferably in a secured way.

[70] This security may be ensured by having first the client C determining a couple of associated private and public keys, and transmitting to the enclave this public key together with the application S_{app} . The application can then encrypt the digital transformation data with this public key, so that the client
20 can decrypt them at receipt with its private key.

[71] Furthermore, according to embodiments an attestation is provided to the client C if the trusted Execution Environment, TEE, successfully runs the classical enclave S_{SE} , e.g. the application S_{app} , as a proof of the integrity of the TEE and of the enclave running on the TEE.

25 [72] According to the document Intel® Software Guard Extensions Developer Guide, "[a]n attestation is the process of demonstrating that a piece of software has been established on a platform. In case of Intel SGX, it is the mechanism by which a third party establishes that a software entity

is running on an Intel SGX enabled platform protected within an enclave prior to provisioning that software with secrets and protected data.”

[73] The digital-to-analog converter S_{DAC} constitutes a junction between the classical and digital subpart S_D of the enclave S_E and the quantum and analog subpart S_A of the enclave (this junction being depicted as a dashed line on FIG. 2).

[74] According to embodiments, the transformation performed by the transformation means is a modulation. Accordingly, the (digital and analog) transformation data are (respectively digital and analog) modulation data.

[75] As different qubits can be coded by using degrees of freedom of different physical particles, different types of modulating means S_{mod} can be used. In particular photonic modulator can be used to modulate one quantum parameter of the photons.

[76] For instance, in an embodiment based on a photonic modulator, the modulator can act on phases, phase differences, frequencies, polarization, time localization, etc. of the incoming photons, provided by the quantum source.

[77] However, many other implementing technologies are possible and the invention is independent of these technologies.

[78] As depicted on FIGS. 1 and 2, a qubits source S_S generates qubits q and transmits them to the transformation means S_{mod} (for instance a modulator). The generated qubits can have fixed quantum states.

[79] As explained earlier, the qubits generated by the qubit source S_S may be supported by various physical particle, like photons, electrons, etc. The state of the qubit is implemented by a degree of freedom of the physical particle.

[80] The transformation means then modulate a degree of freedom of the qubit q according to the analog transformation (e.g. modulation) data $\bar{\theta}$. The

resulting qubits $\hat{q}$ can then be provided to the quantum computing means S_M of the server, outside of the enclave S_E .

[81] The quantum computing means S_M can then use the transformed qubits for further computation, including transmission to another quantum-
5 enable device. These aspects are let outside of the invention, which relates to the preparation of transformed qubits for the server S .

[82] In particular, it may be possible that the quantum computing means S_M provides some results to the enclave S_E . The enclave may in addition
10 comprise some quantum-enabled input interfaces, and in particular some detector and an analog-to-digital converter (ADC) to get quantum states of quantum signals received from the quantum computing means S_M . Then these digital data can be encrypted and transmitted to the client C , for instance, or to another party. These arrangements, as well as other
15 arrangements, are possible without departing from the scope of the invention.

[83] In consequence, the quantum computing means of the server receives qubits but cannot learn anything of their quantum states, because of a secured separation of the quantum means and of the enclave S_E , which is aware of the transformation data

20 [84] A Remote State Preparation (RSP) scheme can thus be ensured, by moving the quantum part of the preparation stage totally from the client to the server. Accordingly, the client may be a classical client, i.e. without any quantum means.

[85] This RSP scheme can then be used to ensure quantum computation
25 with the required blindness property and verifiability

[86] Verifiability could be achieved while composing an alternative Delegated Quantum Computation (DQC) protocol which introduces trap qubits in UBQC scheme, as described in Joseph F. Fitzsimons and Elham

Kashefi, "Unconditionally Verifiable Blind Quantum Computation" in *Physical Review A*, 96(1):012303, 2017.

[87] Full blindness can be ensured by the embodiments of the invention, by the use of a chain of security mechanisms, i.e. cryptographic mechanisms between the client and the server, a classical enclave on the server, and secured means to provide analog transformation data to the transformation means, as described here-above.

[88] In addition, according to embodiments, a further security mechanism is provided to ensure of the non-mischievous nature of the quantum source S_S itself.

[89] In particular, a mechanism is proposed to verify that the qubits prepared and received from the quantum source have a fixed quantum state.

[90] A mechanism may consist in identifying some test qubits among a flow of qubits received from the qubit source. An unknown transformation can then be applied to these test qubits by the transformation means S_{mod} , which are then transmitted to the quantum computing means S_M of the server.

[91] Then, the enclave S_E instructs the quantum computing means to measure these transformed test qubits according to a given base.

[92] The quantum computation means transmit then the result of these measurements to the enclave S_E .

[93] The enclave may then verify this result: if the result is compatible with the unknown transformation applied by the quantum transformation means S_{mod} , then the enclave knows that the quantum source provided a fixed quantum state. Otherwise, the enclave may trigger an error.

[94] As some error may happen in the generation of the qubits, without any mischievous behaviour of the latter, a threshold may be set to trigger the triggering of an error only if the number of cases of incompatibility between

the result of the measurement and the applied transformation (i.e. failed verification) is above this threshold.

[95] According to embodiments, the server may a standalone apparatus, wherein the enclave, the qubit source and the quantum-computation means are provided. According to variants, the server may be a platform composed of several elements located in a vicinity but not necessary as a standalone apparatus. For instance, the server may be made of several boxes connected by the appropriate communication means.

[96] EXAMPLE OF A RANDOM NUMBER GENERATOR

[97] In order to illustrate various embodiments of the invention, an example is provided to generate a random number to be used for further treatment by the quantum computing means S_M of the server. Thanks to the invention, a classical client (without any quantum-enabled means), like a legacy computer or even a smartphone, can trigger the generation of a random number at a quantum-enabled server.

[98] A quantum random number generation can be defined to exploit the unpredictability of quantum mechanics as resource of randomness. According to embodiments, however, one can still use classical resource of randomness to encode random information, as described here-below.

[99] The client C can initially design a RNG (Random Number Generation) application S_{app} in binary code and seals a public key p_k with it. The associated private key, s_k , is kept secret at client's side.

[100] Then the RNG application is transmitted to the enclave S_E of the server S through a secured communication channel T. Accordingly, classical encryption mechanisms can be put in place, as described here-above.

[101] The enclave S_E executes the RNG application in the Trusted Execution Environment, TEE, as a true enclave application. If so, the client C will

receive an attestation which verifies the integrity of the TEE and of the enclave.

[102] If no such an attestation can be transmitted to the client, the latter may then consider that the preparation phase cannot be accepted. Accordingly
 5 any results received from the server shall then be discarded, and depending on the case, other actions can be triggered, like raising an alert, etc.

[103] Example of RNG application may be:

Procedure RNG(n, p_k)

```

    i ← 0
10    θ[n] ← 0

     $\mathbb{Z}_{\frac{\pi}{4}} = \left\{ 0, \frac{\pi}{4}, \frac{2\pi}{4}, \dots, \frac{7\pi}{4} \right\}$ 

    While i < n do
        θ[i] ←  $R\mathbb{Z}_{\frac{\pi}{4}}$ 
        i ← i+1
15    End while

    S ←  $Enc_{p_k}((\theta[n]))$ 
  
```

[104] The binary code corresponding to this pseudocode is transmitted to the enclave, together with the two parameters the number n of random number to generate and the public key p_k .

20 [105] According to this code, as an illustrative example, a set $\mathbb{Z}_{\frac{\pi}{4}}$ is defined corresponding to rotation angles of a modulated quantum elements. Each rotation angle corresponds then to a value (or state) of the respective qubit.

[106] Then, for each i of the n qubits, a random rotation $R\frac{\pi}{4}$ is determined and assigned to the respective qubit's value, $\theta[i]$. These values correspond to the digital modulation data.

[107] Once all qubits have been assigned a random value, these values can
5 be encrypted with the received public key p_k and sent back to the client C . (In the pseudocode, the function Enc_{p_k} corresponds to this process)

[108] The client can then decrypt them by using its private key s_k and learn the random numbers $\theta[i]$, $0 < i < n$.

[109] Once the application S_{app} has executed, the generated digital
10 modulation data $\theta[i]$ are input to the digital-to-analog converted S_{DAC} and the resulting analog modulation data are then provided to the modulating means S_{mod} .

[110] The modulating means S_{mod} receives a qubit prepared by the qubit source in a fixed state, for instance $|+\rangle$.

[111] By applying the modulation to the whole set of qubits, the resulting
15 quantum state, outputted by the modulating means is:

$$|e\rangle = Z_1(\theta[1]) \otimes Z_2(\theta[2]) \dots \otimes Z_{1n}(\theta[n])|I\rangle$$

with

$$|I\rangle = |+\rangle^{\otimes n}$$

20 $Z(\theta)$ represents a rotation according to angle θ around the Z axis on the Bloch sphere from an initial state $|+\rangle$, which originally lies on the x axis.

[112] The quantum-computing means S_M of the server S can then receive
25 the qubits and can compose any further processing, including computing, transmission and measurements.

[113] Also, according to embodiments, the random number generated by this RNG application can be used as a secret key for ciphering the communication from the enclave to the client, as explained above.

CLAIMS

1. Quantum-enabled server (S) comprising an enclave (S_E), a qubit source (S_S), and quantum-computing means (S_M), said enclave comprising means for
 - receiving information from a remote client through a secured communication channel;
 - determining transformation data from said information;
 - transforming at least one qubit received from said qubit source, according to transformation data ;
 - providing the at least one transformed qubits to said quantum-computing means.
2. Quantum-enabled server according to the previous claim, wherein said transformation data are contained in said information and adapted to transform said at least one qubit.
3. Quantum-enabled server according to claim 1, wherein said information represents application code configured for being executing within said enclave and for generating said transformation data.
4. Quantum-enabled server according to any of the previous claims, wherein said transformation data are modulation data, and said enclave comprises means for modulating at least one qubit received from said qubit source, according to modulation data
5. Quantum-enabled server according to any of the previous claims, wherein said enclave comprises
 - a classical secured enclave (S_{SE}) comprising a trusted execution environment, a memory comprising an operating system and adapted to host an application code, and

configured to generate said transformation data as digital transformation data (θ)

- a digital-to-analog converter (S_{DAC}) for converting said digital transformation data into analog digital data ($\bar{\theta}$);
 - 5 – transforming means (S_{Mod}) for transforming a degree of freedom of said at least one qubit according to analog transformation data.
6. Quantum-enabled server according to the previous claim, wherein the digital-to-analog convertor is encapsulated into a cage preventing any
- 10 electromagnetic signals been read from outside of said enclave.
7. Quantum-enabled server according to any of the previous claims, wherein said information is ciphered and deciphered at said enclave at receipt.
8. Quantum-enabled server according to claim 4, wherein said enclave
- 15 sends an attestation back to said client to if said Trusted Execution Environment successfully runs said classical enclave.
9. Quantum-enabled server according to any of the previous claims, wherein said enclave is configured to
- determine test qubits among said at least one qubit;
 - 20 - transform said test qubits according to an unknown transformation;
 - instruct said quantum-computing means to measure the transformed test qubits according to a given base;
 - verify if the result of the measurements is compatible with said unknown transformation; and
 - 25 - trigger an error if a number of failed verification is above a given threshold.

10. Quantum-enabled server according to any of the previous claims, configured to retrieve a public key from said information, encrypt said transformation data with said public key, and transmit the encrypted transformation data to said client.

- 5 11. A method for delegating quantum computing to a quantum-enabled server, comprising:
- sending information from a remote client to an enclave inside said quantum-enabled server through a secured communication channel;
 - determining, at said enclave, transformation data from said
10 information;
 - transforming, at said enclave, at least one qubit received from said qubit source, according to said transformation data ;
 - providing the at least one transformed qubits to said quantum-computing means.

15