



# Journal of Discrete Mathematical Sciences and Cryptography

ISSN: (Print) (Online) Journal homepage: [www.tandfonline.com/journals/tdmc20](http://www.tandfonline.com/journals/tdmc20)

## Computational quantum key distribution (CQKD) on decentralized ledger and blockchain

Gerardo Iovane


**To cite this article:** Gerardo Iovane (2021) Computational quantum key distribution (CQKD) on decentralized ledger and blockchain, Journal of Discrete Mathematical Sciences and Cryptography, 24:4, 1021-1042, DOI: [10.1080/09720529.2020.1820691](https://doi.org/10.1080/09720529.2020.1820691)

**To link to this article:** <https://doi.org/10.1080/09720529.2020.1820691>



Published online: 27 Apr 2021.



Submit your article to this journal 



Article views: 60



View related articles 



View Crossmark data 

## Computational quantum key distribution (CQKD) on decentralized ledger and blockchain

Gerardo Iovane  
*Department of Computer Science*  
*University of Salerno*  
*84084 Fisciano SA*  
*Italy*

---

### Abstract

In this work, we propose a new protocol for data transmission, using both the potential of quantum encryption expressed by the BB84 protocol and the possibilities offered by distributed ledgers and blockchain. As we shall see, this approach allows the transmission of keys in maximum security whether it is owned in quantum communication channel, or by using traditional channel and emulating on it a quantum functionalization thanks to the use of network nodes as new virtual quantum components, called i) Quantum Spin Generator (QSG), ii) Base Generator (BG), iii) Quantum Photon Polarizer (QPP), iv) Quantum Photon Meter (QPM), v) Quantum Photon Collider (QPC). The end result will be the security of one time pad encryption and quantum encryption, an intrinsic crypto agility linked to the dynamic allocation and functionalization of nodes, a growing security proportional to the growth of the number of network nodes used to encrypt and transmit the information.

---

*Subject Classification:* 00A69.

*Keywords:* Cryptography, QKD, Block chain.

### 1. A short scenario

Quantum Resistant Cryptography (QRC) is a very active and current field at date. In this context, algorithms are developed to be secure against attacks by quantum computers. The most popular public-key algorithms are not quantum resistant at date. In fact, the security of these algorithms relies on one of three hard mathematical problems: the integer factorization problem, the discrete logarithm problem or the elliptic-curve discrete

---

E-mail: [giovane@unisa.it](mailto:giovane@unisa.it)

logarithm problem. While all of these problems can be easily solved on a sufficiently powerful quantum computer running Shor's algorithm. This is particularly true with respect to current public-key algorithms. In contrast, most current symmetric cryptographic algorithms are considered to be relatively secure against attacks by quantum computers [1], [2].

At date QRC research is focused on the following approaches mainly:

- Lattice-based cryptography [3]-[9],
- Multivariate cryptography [10]-[13],
- Hash-based cryptography [14]-[16],
- Code-based cryptography [17],
- Supersingular elliptic curve isogeny cryptography [18]-[20],
- Symmetric key quantum resistance [21]-[22].

Such a significant effort and such valuable results make clear how important the challenge in the field of encryption is in light of Quantum Computing's recent achievements and the impact it will have on all areas, first of all the security of information and infrastructure.

Thanks to the project Horizon 2020 ICT-645622 PQCRYPTO Post-Quantum Cryptography for Long-Term Security, the effort — made by D.Augot et al. — generated a very effective work, i.e. *Initial recommendations of long-term secure post-quantum systems* starting from 2015.

In addition recently, some very interesting results stimulated the attention on design new protocols, on simulate BB84, into the context of quantum key distribution [23]-[26].

The scenario above is deeply inspired by the state of art in [27]; in fact by following it, we can summarize the main results at date in the Table 1 and in addition significant results are produced in Open Quantum Safe (OQS) project [28], [29].

In this broad scenario, the goal of this work is to enhance the ideas behind the work of Bennett and Brassard BB84 taking advantage of the still unexpressed and not totally discovered potential of Distributed Ledger and Blockchain. As we will see the result — rather than a new encryption algorithm — will be a protocol which makes a Computational Quantum Key Distribution, which can operate both in the presence of a physical quantum channel, both with computational quantum components and using both possibilities in relation to the technologies used on computational nodes.

**Table 1**  
**A classification of results at data**

Algorithm	Type	Public Key	Private Key	Signature
NTRU Encrypt	Lattice	6130 B	6743 B	
Streamlined NTRU Prime	Lattice	1232 B		
Rainbow	Multivariate	124 KB	95 KB	
SPHINCS	Hash Signature	1 KB	1 KB	41 KB
SPHINCS+	Hash Signature	32 B	64 B	8 KB
BLISS-II	Lattice	7 KB	2 KB	5 KB
GLP-Variant GLYPH Signature	Ring-LWE	2 KB	0.4 KB	1.8 KB
New Hope	Ring-LWE	2 KB	2 KB	
Goppa-based McEliece	Code-based	1 MB	11.5 KB	
Random Linear Code based encryption	RLCE	115 KB	3 KB	
Quasi-cyclic MDPC-based McEliece	Code-based	1232 B	2464 B	
SIDH	Isogeny	751 B	48 B	
SIDH (compressed keys)	Isogeny	564 B	48 B	
3072-bit Discrete Log	<b>not PQC</b>	384 B	32 B	96 B
256-bit Elliptic Curve	<b>not PQC</b>	32 B	32 B	65 B

While here we consider the scenario shortly, in Sect.2 we will introduce the Quantum Key Distribution (QKD). In Sect.3 we will go towards a new quantum encryption protocol, oriented to Blockchain and Distributed Ledger. Sect.4 will be devoted to an optimization of the results in Sect.3, by getting as result a computational quantum encryption protocol, oriented to Blockchain and Distributed Ledger. Finally Sect.5 reports the Conclusion and Perspectives.

## 2. QKD – Quantum Key Distribution: a short resume

Below we bring to the attention of the reader the basic idea of C.H.Bennett and G.Brassard of 1984 on which is based the protocol of quantum key distribution, known as BB84 [30], which resumed the work conceived by S.Wiesner in 1970 and published in 1983 [31].

The idea behind BB84 aims to put in place everything that is necessary for the real use of Gilbert S. Vernamm's cipher, One Time Pad (OTP).

So, let us briefly look at the fundamentals of the OTP.

In the OTP, the sender Alice and receiver Bob have the same key, which must meet the following three properties:

1. The key must be the same length as the message to be encrypted and transmitted;
2. The key must be a completely random sequence;
3. The key should never be reused.

For example, by using the binary alphabet 0, 1 and  $m = m_1, m_2, \dots, m_r$  a binary message to send and  $k = k_1, k_2, \dots, k_r$  the key, then encryption can be the simple addition element by element, that is,

$$c_i = m_i + k_i \pmod{2}$$

with  $i = 1, \dots, r$  and therefore, the encrypted message will be  $c = c_1, c_2, \dots, c_r$ .

As it is well known, the unassailability of the Vernamm cipher lies in the computational complexity that grows like the factorial of  $r$  and therefore grows at the growth of  $r$ , but above all because in cryptanalysis all the clear messages obtained thanks to a permutation would be equally likely.

The problem with Vernamm's cipher is in key transmission. In fact, the transmission must be absolutely safe. QKD — in general — and the BB84 in particular have aimed to give a brilliant solution to this goal.

Let us see below, in a consussive and schematic way, how this was achieved.

### Step 1 : Communication by transmitting photons through a quantum channel.

*Alice)* The sender Alice has four possible polarizations ( $-\pi/4, 0, \pi/4, \pi/2$ ) and two binary numeric values (0, 1), as agreed with Bob and below:

Polarizations:	$\uparrow$	$\rightarrow$	$\nearrow$	$\searrow$
Corresponding binary digits:	1	0	1	0

Alice chooses a  $r$  number much larger than the length of the key she intends to send Bob.

Alice chooses an  $r$ -pla of polarized photons (that is, a binary string) that she sends to Bob and that she preserves.

Alice's digits of the candidate key:     1    1    1    0    1    0    1    0  
Polarization of Alice:                     $\uparrow$     $\uparrow$     $\nearrow$     $\searrow$     $\uparrow$     $\searrow$     $\uparrow$     $\rightarrow$

If Bob read in plain text, that is, without the use of photons, and used this as a key there would be no security, as a third Eve could enter illegally and maliciously snort and decode the message, or retransmit erroneous messages.

*Bob*) Bob also has four polarization filters that he can apply to the message received by Alice on quantum channel and he proceeds as follows:

Filter used by Bob:                     $\uparrow$     $\nearrow$     $\searrow$     $\rightarrow$     $\uparrow$     $\searrow$     $\nearrow$     $\nearrow$   
Polarizations received by Bob:      $\uparrow$     $\searrow$     $\emptyset$     $\uparrow$     $\uparrow$     $\searrow$     $\nearrow$     $\searrow$   
Message received by Bob:           0    0    1    1    1    0    1    0

Then, Bob misses the filters at pointers 2, 4, 7, 8, but only in case 2 and 4 the transmitted message is incorrect.

The presence of errors is crucial, because on the one hand they can be corrected (as we will see here), but on the other hand similar mistakes will also be made by Eve.

**Step 2: Communicate over an unsecured channel for error elimination and key extraction.**

Alice knows the string of polarizations, which have been sent to Bob, and Bob knows the base he used as filters.

Bob tells Alice the base he uses, and Alice tells Bob which ones are right.

So both Alice and Bob match the figures corresponding to the positions where Bob's filter was wrong and where it was correct.

Continuing the example above, then, Bob communicates to Alice

   +         $\times$          $\times$         +        +         $\times$          $\times$          $\times$

where we remember  $+= (\rightarrow, \uparrow) = (0, \pi/2)$  and  $\times = (\searrow, \nearrow) = (-\pi/4, \pi/4)$  and Alice responds to Bob that the right filters are:  $1^\circ, 3^\circ, 5^\circ, 6^\circ$ .

As a result, Alice and Bob delete the cyphers in positions  $2^\circ$ ,  $4^\circ$ ,  $7^\circ$ ,  $8^\circ$ , and they get the final key 1110, which remains secret, since in the exchange of information about Bob's errors, even on unsafe channel, they communicate only positions, but not the values they contain, that is, key values.

So even if Eve listens to that conversation, but not the one on the quantum channel, Eve can only know where Bob used a correct filter; so this doesn't allow Eve to get key information.

### **Step 3: Communicate over an unsecured channel for error elimination and key extraction.**

The possibility of identifying Eve lies in the fact that Alice and Bob used both straight and diagonal polarization at the same time. If not, Eve could intercept Alice's transmission and then imitate Alice, retransmitting it to Bob (opaque intrusion). Instead, in the current scheme, if Eve intrudes between Alice and Bob, Eve must behave exactly as Bob behaves, that is, to choose random bases of polarization, committing mistakes to Bob as she did, and sending Bob the filtered message herself. That message will be different from Alice's and so Alice and Bob comparing their keys are likely to find out that there has been an intrusion of Eve.

Conversely, the case where Eve does not relay Alice's message back to Bob is trivially discovered when Alice verifies on an unsecured channel that Bob has not received her message.

And so Alice and Bob can make new and repeated attempts until they are aware of Eve's non-attendance.

## **3. Towards a new quantum encryption protocol, oriented to Blockchain and Distributed Ledger**

As we shall see in this section, the scope of BB84 goes beyond QKD; in fact, here we report a methodology specifically designed for blockchain and distributed ledgers, which although based on concepts proper to Quantum Mechanics, it can be used whether the channel nodes are physically quantum, — as we shall see — that they are “computably quantum”.

Therefore, the use of such a methodology on the one hand may encourage the development of quantum nodes, but on the other hand as a new methodology it can be used immediately even on non-quantum nodes and channels. In addition, we should not really wonder if we see computational solutions for the creation of keys that simultaneously

use quantum nodes with quantum channels in fiber optics, traditional communication channels, Wi-Fi channels and Li-Fi (i.e. LED light-fi).

However, as this is only possible when the number of nodes which will process the information for the creation of the key between Alice and Bob is significant. So this work combines at the same time and is made conceivable thanks on the one hand to *QKD* and on the other hand to the technologies of Distributed Ledger and Blockchain, without which it would not be practicable for the limits exposed in section 1, when we have indicated that the BB84 algorithm would have no security if there was no quantum channel, that is, if there were no carrier photons.

Before we begin the protocol description, we need to define the physical state of a node and its possible functions.

Let us consider a Blockchain with a number of  $D$  nodes. Each  $d_i \in D$  node can be in one of the following states:

- o-off
- b-busy
- a-active, i.e. idle and ready.

Once the sender Alice verifies the availability of nodes to receive a computational load related to key generation and transmission, Alice proceeds to functionalize the nodes. Each node can behave in one of the following functions:

1. QSG — Quantum Spin Generator,
2. BG — Base Generator,
3. QPP — Quantum Photon Polarizer,
4. PFE — Photon Fusion Engine,
5. QPM — Quantum Photon Meter,
6. QPC — Quantum Photon Collider.

Let us look in detail at the IO of the different functions of the nodes, their behavior and their status.

Typically, a node will receive a statement record which contains the following information:

```
[key_gen_id, process_id, node_type, sender_address, reciver_address]
```



where `key_gen_id` will serve to disambiguate competing key construction processes, `process_id` will disambiguate different processes within the same key building process, `node_type` is the type of job you wish assign to the node, with respect to the fixed key generation process (this means that in another key generation process, the same node can act in a different way, `sender_address` will be the sender address and `receiver_address` will be the address (i.e. addresses) of the receiver (i.e. receivers) of the specific key generation process (i.e. `key_gen_id`) and subprocess (i.e. `process_id`).

*QSG — Quantum Spin Generator.* A functionalization of a node of type  $QSG_i$  requires that node will randomly chose between spin up or spin down or more easily to generate a random binary digit (i.e. 0 or 1). Therefore, a node, which is functionalized in this way, will put itself in the Busy state for the other nodes, it will generate the binary digit, and it will transmit it to the sender Alice and  $QPP_i$  receiver addresses; then it will refresh itself, and it will re-enter the Active state.

*BG — Base Generator.* A functionalization of a node of type  $BG_{A,i}$  requires that node to behave as a generator of a randomly chosen base between  $+= (0, \pi/2)$  and  $x = (-\pi/4, \pi/4)$ . Therefore, a node, that is functionalized in this way, will put itself in the Busy state, it will generate the base, then it will transmit the base to the  $QPP_i$  receiver address; consequently it will refresh itself, and it will put itself in the Active state.

*PFE — Photon Fusion Engine.* A functionalization of a node of type *PFE* requires that node will perform a fusion among the different polarizations received from  $QPP_i$ . This job is very delicate and as we will see it requires big attention, since this node could be fragile. Considering that the node cannot know the order of arrival of polarizations from  $QPP_i$  and also taking into account that — for security reasons — does not know how many  $QPP_i$  will send it polarizations, it is necessary to functionalize with two more elements from Alice. So Alice has to send the length  $lk$ , which is how many  $QPP_i$  will send the polarization and the law to sort them. Therefore, for considering this type of nodes, the initialization record transmitted by Alice will be changed in

```
[key_gen_id, process_id, [node_type], sender_address, receiver_ad-
                        dress]
```

where if the node is of *PFE* type then

```
[node_type]=[PFE, lk, sorting rule]
```

otherwise it will be

```
[node_type]=[node_type, 0, 0]
```

Therefore, a node, which is functionalized in this way, will put itself in the Busy state for the other nodes, it will perform the polarization fusion, it will transmit to *QPM* receiver addresses; then it will refresh itself, and it put itself in Active state.

*QPM – Quantum Photon Meter.* A functionalization of a node of type *QPM* requires that node will perform a measurement of polarizations of Alice packet using the bases  $BG_{B,i}$  randomly chosen by Bob. This job is delicate too as in the previous case and it requires big attention since this node could be fragile. Considering that the node cannot know the order of arrival of  $BG_{B,i}$  — for security reasons too — *QPM* will ask to Bob to send to *QPM* the following information record

```
[key_gen_id, [process_id], [node_type], sender_address, reciver_ad-
dress]
```

where `key_gen_id` is the same received by Alice, while `[process_id]` is the vector containing the unique assignment codes given by Bob to the  $BG_{B,i}$ . In this way *QPM* is able to process the information and will sort the information coming from  $BG_{B,i}$  according to Bob's record .

Therefore, for considering this type of nodes, the initialization record transmitted by Alice or Bob will be changed in

```
[key_gen_id, [process_id], [node_type], sender_address, reciver_ad-
dress]
```

where if the node is of *QPM* type then

```
[node_type]=[PFE, lk, sorting rule]
```

otherwise it will be

```
[node_type]=[node_type, 0, 0]
```

while for data coming from Alice

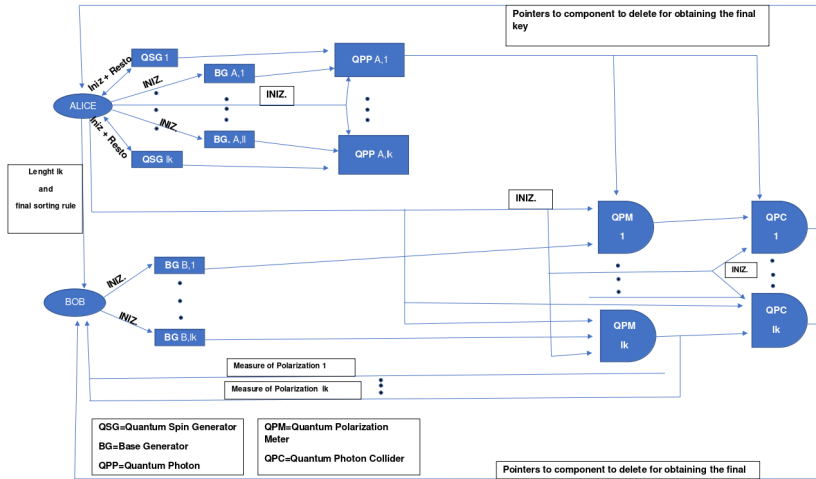
```
[process_id]=[process_id, 0'0],
```

the `[process_id]` coming from Bob is

```
[process_id]=[process_id,  $BG_{B,1\_address}$ ,  $BG_{B,lk\_address}$ ].
```

Therefore, a node which is functionalized in this way, will put itself in the Busy state for the other nodes, it will perform the measurements of Alice's polarization in Bob's base , it will transmits to Bob by fragmenting the results on  $lk$   $QPP_{B,i}$ , and by sending to Bob the sorting rule, and it will transmits to *QPC* receiver addresses the polarization strings of Bob; then it will refresh itself, and it put itself in the Active state.

*QPC — Quantum Photons Collider.* A functionalization of a node of type *QPC* requires that the node will compare two strings of polarization



**Figure 1**  
An unoptimized CQKD protocol.

as received by QPM and it will select the positions where they are equal. Therefore, a node, which is functionalized in this way, will put itself in the Busy state for the other nodes, it will performs the matching and it will send to Alice and to Bob the correct indexes, by indicating to delete the wrong ones; it will refresh itself, and it will put itself in the Active state.

Here in the following we represent the protocol to perform a CQKD.

Let us see in detail how the protocol works.

1. Alice measures the length of the code to be encrypted  $lc$  and sets a key length of  $lk = \lfloor 2.5lc \rfloor$ ;
2. Alice pings the nodes of the Blockchain network and annotate the nodes which respond and are active, noting the number of active nodes  $\#(n)$ ;
3. Alice verifies if  $\#(n) \geq \alpha$  with  $\alpha = 4lk + 3$ ;
  - 3.1 if  $(\#(n) \geq \alpha) = TRUE$  then select the first  $\alpha$  nodes randomly,
  - 3.2 else Iteration Jobs  $IJ = \lceil \alpha / \#(n) \rceil$  and allocate  $IJ$  jobs per node;
4. Scrolling the list of nodes from top to bottom, Alice chooses  $lk$  nodes which will operate as QSG; if  $\#(n) < lk$  she will assign more than one job of this type to the nodes;

5. Scrolling the list of nodes from bottom to top, Alice chooses  $lk$  nodes which will operate as  $BG$ ; if  $\#(n) < lk$  she will assign more than one job of this type to the nodes automatically;
6. Alice selects for each pair  $(QSG)_i - (BG)_{A,i}$  a  $(QPP)_{A,i}$ ;
7. Alice opens a unique transaction ticket,  $\alpha \in \mathbb{Z}_q^*$ ;
8. Alice transmits the following command to each node

```
[id_process, id_subprocess, node_type, num==0, send to:
      Address 1, Address 2]
```

If  $node\_type$  is type  $QSG$ , the node will return Alice as address 1 and a  $QPP$  as address 2 the value of the spin;

If  $node\_type$  is type  $BG$ , the node will return the base component only to address 2 i.e.  $QPP$ ;

If  $node\_type$  is type  $QPP$ , the node will return the polarization component to address 2 i.e.  $PFE$ ;

9. Each node carries out its job and transmits the result to the competent nodes in the following format

```
[id_process, id_subprocess, result]
```

10. Alice receives  $lk$   $QSG$  values, starting from table of the received records as

```
[id_process, id_subprocess, node_type, num, send to:
      Address 1, Address 2],
```

- = and she generates a new random order by changing the order of the records;

11. Alice transmits the new order to  $PFE$ , i.e. the way in which the node will have to forge the array of polarizations, giving the following command

```
[id_process, id_subprocess, node_type == PFE, [num],
      send to: address 2 == QPM]
```

12. Alice notifies Bob the length  $lk$  of the key and the  $QPM$  address in the following format

```
[id_process, lk, send to: address 2 == QPM]
```

13. Bob proceeds as Alice in step 2;
14. Bob proceeds as Alice in step 3;
15. Bob checks if  $\#(n) \geq lk$  and proceeds similarly to Alice in point 5;

16. Bob assigns  $lk$  jobs to the nodes, which will operate as  $(BG)_{B,j}$  with  $j = 1, \dots, lk$ , giving them the following command

```
[id_process, id_subprocess, node_type==BG, value, send to:
      address1==Bob address2==QPM];
```

17. For each  $id\_process$ ,  $QPM$  will receive from the processing establish by Alice, i.e. from  $PFE$  a vector of polarization states and from Bob, i.e.  $(BG)_{B,j}$ ,  $lk$  base components;
18. According to the rules of the Quantum Mechanics — that is by working as polarization projector —  $QPM$  will match one by one the records, that is, it will represent the polarizations of Alice along the base of Bob and will return (without keeping copies): i) to  $(QPP)_{B,j}$  the respective polarizations to be transmitted to Bob, ii) to Bob and Alice the ordering law (which obviously and generally it will not coincide with the order of arrival by the  $(QPP)_{B,j}$ , nor with the original order received by  $PFE$ ), and iii) the  $QPC$  will receive from  $QPM$  the photon state for collisioning (i.e. the positions where Alice and Bob's keys do not match and which must be deleted);
19. Each  $(QPP)_{B,j}$  will provide Bob with the secret key fragment in the form

```
[id_process, id_subprocess==subprocess of (BG)_{B,j},
      [polarization], send to: address 1==Bob],
```

= with value = 0, 1;

20. Bob will build the key according to points 19 and 20;
21. Alice will perform a key straightening according to point 19;
22.  $QPC$  notifies Bob and Alice of the pointers to the correspondents of the key which must be deleted;
23. Bob and Alice carry out the cancellation and get the final key with which to encrypt the message if the final length  $l > le$  otherwise the process will start again;
24. Alice sends Bob an information containing

```
[id_process, sorting rule]
```

= where the sorting rule is the law to rearrange the key — after receiving it from  $QPC$  — and before encrypting the message.

From what is described, we can conclude the following considerations.

The nodes are dynamically allocated, their functionalization changes dynamically over time depending on the process that the node is performing and of which it is an operational member; an ill-intentioned operator Eve

cannot synchronize with all nodes without having complete control of the network behind the Blockchain; therefore, she could reach the node and pierce it, but unfortunately for her only at processing concluded for that fixed key generation process. These elements make the proposed solution inherently crypto agile and dynamically mutant. In addition, using a one time pad the procedure makes the solution inherently quantum resistant.

The dynamic allocation of nodes makes difficult for Eve to attack because even if Eve can intercept information about *PFE*, *QPM*, *QPC* — from a continuous and competing process perspective — she would have the polarizations from which she potentially goes back to the components of the key, but she wouldn't know who to report that information to, i.e. Alice and Bob. To know this information too, she should reconstruct the addresses of *QSG* and *BG* and track down Alice and Bob. On the other hand, starting from listening to the channel of Alice and Bob, she should also go to pierce *PFE*, *QPM* and *QPC*. While difficult, as this is possible, the infrastructure solution can be made even more robust. This is because *PFE*, *QPM* and *QPC* lend themselves as targets sensitive to attacks. In this regard, communications to and from *PFE*, *QPM* and *QPC* could be secured via *RSA* or *ECDH* for example. In addition, the *QSG* and *BG* nodes could transmit information to *PFE*, *QPM* and *QPC* — through a consensus mechanism — in order to create a deterrent for an attacker such as Eve. Although this solution can be effective, it is not satisfactory if you take into account the real potential which Blockchain or Distributed Ledgers can offer. Therefore, in the next section we will present an optimization of the previous protocol, which offers a robust and durable solution.

To better prepare the solution which will be presented in the next section it is useful to make a critical analysis of the solution presented here with respect to the target.

Starting from BB84, the goal was to create a transmission and encryption mechanism, which could use the same principles as Quantum Mechanics; but instead to use only a quantum channel to transport photonic vectors, it could work on a traditional network, wi-fi, li-fi, etc, enhancing above all the concept of decentralization, thanks to the use of a new ideation, which we could call “computational photon”.

In fact, the decentralization took place, thanks to the *QSGs*, *BGs* and *QPPs*, but with respect to the *PFE*, *QPM* and *QPC* nodes they again function as centralists compared to Alice and Bob. So we have arrived at a distributed solution, but not decentralized one. Put another way, if the attacker Eve is focused on Alice and Bob will not succeed in her intent, since the salient information will be on other nodes, namely on *PFE*, *QPM*

and *QPC*. Conversely, if the attacker Eve, by focusing on *PFE*, *QPM* and *QPC*, could trace the key, but never find out that it is related to the message that Alice and Bob will exchange.

From the terms used in these last sentences, namely distribution and decentralization, it is easy to understand that the proposed solution achieves a protocol that is only partially decentralized (if you consider the *QSG*, *BG* and *QPP* nodes) and partially centralized (if you consider the *QSG*, *BG* and *QPP* nodes). The result is a distributed solution. From a conceptual perspective, we conclude, therefore, that distribution is not equivalent to decentralization, in terms of security. In fact, the computation which is distributed on *PFE*, *QPM* and *QPC* is more easily attacked, as it is functionally distributed on these three nodes, than decentralized computation on *QSG*, *BG*, *QPP* type nodes. In other words, the distribution is conceptually and theoretically attackable, although the attacker would probably have very little chance of success.

The revised and optimized solution of this protocol, which will be presented in the next section, leads to a full and total decentralization of the computation, making the solution extremely robust.

#### 4. A computational quantum encryption protocol, oriented to Blockchain and Distributed Ledger

Starting from the limitations and useful considerations, which we observed in the previous section, here we easily introduce an optimized solution. It realizes a full decentralization of key production, that Alice and Bob can use as key in a one time pad perspective.

Let us see in detail how the protocol works.

1. Alice measures the length of the code to be encrypted  $lc$  and sets a key length of  $lk = \lceil 2.5lc \rceil$ ;
2. Alice pings the nodes of the Blockchain network and annotate the nodes which respond and are active, noting the number of active nodes  $\#(n)$ ;
3. Alice verifies if  $\#(n) \geq \alpha$  with  $\alpha = 5lk$ ;
  - 3.1 if  $(\#(n) \geq \alpha) = TRUE$  then select the first  $\alpha$  nodes randomly,
  - 3.2 else Iteration Jobs  $IJ = \lceil \alpha / \#(n) \rceil$  and allocate  $IJ$  jobs per node;
4. Scrolling the list of nodes from top to bottom, Alice chooses the nodes which will operate as *QSG*; if  $\#(n) < lk$  she will assign more than one job of this type to the nodes;

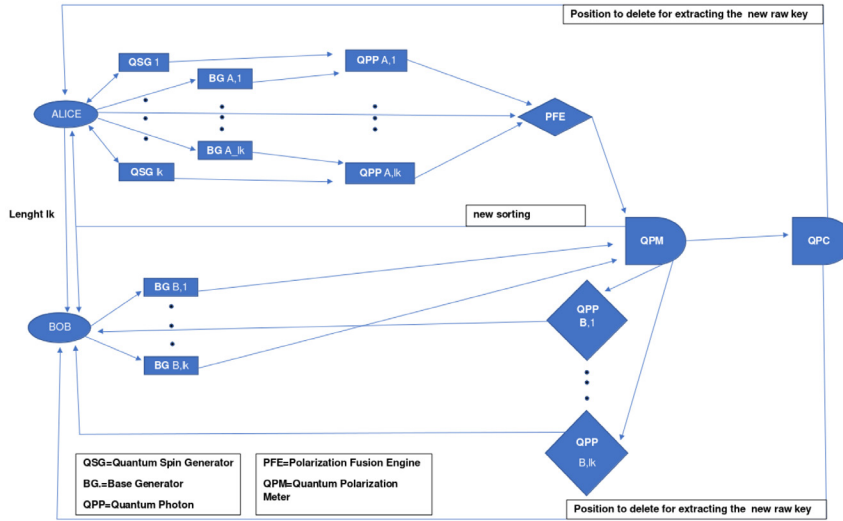


Figure 2  
An optimized CQKD protocol.

5. Scrolling the list of nodes from bottom to top, Alice chooses the nodes which will operate as  $BG$ ; if  $\#(n) < lk$  Alice will assign more than one job of this type to the nodes automatically;
6. Alice selects for each pair  $(QSG)_i - (BG)_{A,i}$  a  $(QPP)_{A,i}$ ;
7. Alice opens a unique transaction ticket,  $id\_process$ ;
8. Alice transmits the following command to each node

```
[id_process, id_subprocess, node_type, num=0,
  send to: Address 1, Address 2]
```

= If **node\_type** is type  $QSG$ , the node will return Alice as address 1 and a  $QPP$  as address 2 the value of the spin;

If **node\_type** is type  $BG$ , the node will return the randomly selected base component only to address 2 i.e.  $QPP$ ;

If **node\_type** is type  $QPP$ , the node will return the polarization value into the base  $BG$  to address 2 i.e. to  $QPM$ ;

9. Each node carries out its job and transmits the result to the competent nodes in the following format

```
[id_process, id_subprocess, result]
```



10. Alice receives  $lk$  QSG values, starting from table of the received records as

```
[id_process, id_subprocess, node_type, num, send to:
      Address 1, Address 2],
```

- = and she generates a new random order by changing the order of the records;

11. Alice notifies Bob the length  $lk$  of the key, the final sorting rule to rearrange the results coming from QPC, and the QPM address in the following format

```
[id_process, lk, sorting rule, send to: address 2==QPM]
```

12. Bob proceeds as Alice in step 2;

13. Bob proceeds as Alice in step 3;

14. Bob checks if  $\#(n) \geq lk$  and proceeds similarly to Alice in step 5;

15. Bob assigns  $lk$  jobs to the nodes, which will operate as  $(BG)_{B,j}$  with  $j = 1, \dots, lk$ , giving them the following command

```
[id_process, id_subprocess, node_type==BG, value, send to
      address2==QPM];
```

16. For each  $id\_process$ , a QPM will receive from the processing establish by Alice, i.e. from a QPP a polarization state and from Bob, i.e.  $(BG)_{B,j}$ , the verses;

17. According to the rules of the Quantum Mechanics — that is by working as polarization projector — a QPM will match a record, that is it will represent the polarizations of a photon by Alice along the base of Bob and will return (without keeping copies): i) to Bob the respective polarizations and ii) to QPC each polarization of Alice and that one measured by Bob;

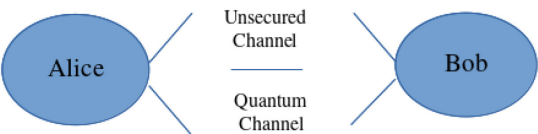
18. QPC will perform the collision and notifies Bob and Alice of the pointers to the correspondents of the key that must be deleted;

19. Bob and Alice will perform the sorting of records according to the order given by Alice to Bob and they carry out the cancellation by getting the final key with which to encrypt the message if the final length  $l > le$  otherwise Alice will command to start again the process until  $l > le$ .

In principle a similar decentralized solution can also work without encryption since an attacker Eve to reconstruct the key need to have the control on the total set of nodes involved into the key construction; this means  $6lk$ , or a smaller number than  $6lk$  if the active nodes are a limited

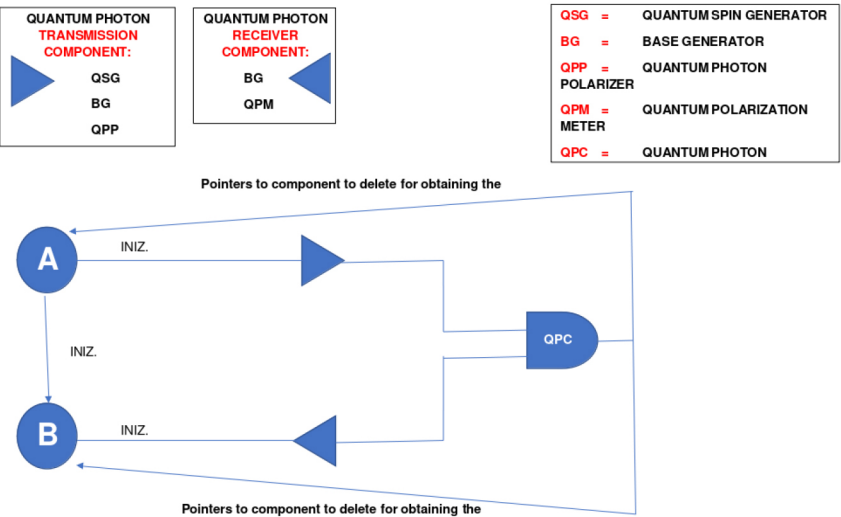
number smaller than  $6lk$ , but in this case Eve must be persistent, i.e. she must have the control of the nodes and be able to synchronize the sniffed results. We can understand that an Eve action is theoretically possible, but it is not effective.

We started from a scheme of BB84 to realize a one time pad encryption, which can be sketched as



The strength was in the approach to send on the unsecured channel just information of good pointers to extract a reduced key, while on the quantum channel the polarizations, perturbed by an attacker Eve.

If we introduce two computational quantum component, named Quantum Photon Transmission Component (QPTC) and Quantum Photon Receiver Component (QPRC), then the protocol shown in Figure 2 can be arranged in the following conceptual way easily, by taking into account that QPTC and QPRC are decentralized.



5. Conclusion and Perspectives

In this work we presented a new way to use the power of nodes and a distributed ledger and blockchain to decentralize the key

construction for exchanges a message peer to peer. This is made by using the Quantum Mechanics, but in a novel vision environment, named Computational Quantum Key Distribution (CQKD), where the nodes act as agents to generate dynamically and randomly: i) Spin (QSG), ii) Base of Representation (BG), iii) Photon Polarization (QPP), iv) Quantum Measure (QPM), v) Quantum Collision (QPC).

The beauty of the present approach is that it can work with any kind of channel, secured and unsecured, common communication network, wi-fi, light-fi, quantum via fiber optics, and any combination of them with respect to the infrastructure used by users at date.

Thanks to the results presented at the end of Sect.4, we can observe the key generation process can be considered as a mining process; it can be done autonomously and continuously by the nodes organized in mining pool to produce key in any time and distribute the keys immediately and just in time when an Alice and Bob need to exchange peer to peer a message. This is strongly effective and show the power of blockchain and distributed ledger clearly.

By taking into account the full decentralization of the key production, in principle, the communication channels could be also unsecured ones. But nothing prevents you from using Perfect Forward Secrecy or PFS as we tested if the number of nodes is very small, according to [19], [32]. As it is known since the end of 2011, Google has provided forward secrecy with Transport Layer Security (TLS) by default to users of its Gmail e-mail service, together with Google Docs and encrypted search among its services. In late 2013, Twitter also introduced this service to its users and Microsoft said it will implement it. About 50% of sites that enable TLS support some cipher suites which provide forward secrecy. In November 2014, WhatsApp said it had introduced end-to-end chat encryption in the latest release, limited to the Android version, excluding multiple chats and chats containing multimedia messages. This guided our choice of first test with a further security of the transmission channels, but nothing prevents us from using other techniques such as ECDH for example. As mentioned above, this additional security measure makes sense only if you operate in contexts where an attacker could take control of the entire network or of all its nodes, vice versa encryption of key fragments becomes unnecessary, as the CQKD is intrinsically secure, thanks to the two basic supporting ideas, i.e. a dynamic and evolving procedure for generating key fragments by network nodes via Quantum Mechanics Principles (i.e. intrinsic crypto agility), the total decentralization of key production (full decentralization). Another element to consider is that the numbers of nodes of a blockchain

is larger and larger at date, so that really the encryption of key fragments could result a surplus in common communication, thanks to the big decentralization.

## References

- [1] P.W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal on Computing*, 26, 5, 1484–1509, 1997.
- [2] D.J. Bernstein, Introduction to post-quantum cryptography, Post-Quantum Cryptography, pp.1-14, Eds.Springer, ISBN 978-3-540-88702-7, 2009.
- [3] C.Peikert, Lattice Cryptography for the Internet, Proc.of CRYPTO 2014 is the 34rd International Cryptology Conference, pp1-25, 2014.
- [4] T.Güneysu, V.Lyubashevsky, T.Pöppelmann, Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems, Proc. of Cryptographic hardware and embedded systems - CHES 2012, 14th international workshop, Leuven, Belgium, Proceedings, pp.530-547, 2012.
- [5] J. Zhang, Z. Zhang, J. Ding, M. Snook, Ö. Dagdelen, Authenticated Key Exchange from Ideal Lattices, Oswald E., Fischlin M. (eds) *Advances in Cryptology - EUROCRYPT 2015*, Lecture Notes in Computer Science, vol. 9057, pp.719-751, Springer, Berlin, Heidelberg, 2015.
- [6] L. Ducas, A. Durmus, T. Lepoint, V. Lyubashevsky, Lattice Signatures and Bimodal Gaussians, R. Canetti and J.A. Garay (Eds.), CRYPTO 2013, pp. 40–56, 2013.
- [7] V. Lyubashevsky, C. Peikert, O. Regev, On Ideal Lattices and Learning with Errors Over Rings, H. Gilbert (Ed.), *Proceedings of Advances in Cryptology – EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.1-23, 2010.
- [8] D.Stehlé, R.Steinfeld, Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over Ideal Lattices, Paterson K.G. (eds) *Advances in Cryptology – EUROCRYPT 2011*, Lecture Notes in Computer Science, vol 6632. Springer, Berlin, Heidelberg, pp.27-47, 2011.

- [9] C. Easttom, An Analysis of Leading Lattice-Based Asymmetric Cryptographic Primitives, Proc. of 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), 2019.
- [10] T. Matsumoto, H. Imai, Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption, EUROCRYPT 1988: Advances in Cryptology' EUROCRYPT '88, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp.419-453, doi:10.1007/3-540-45961-8\_39, 1988.
- [11] J. Patarin, Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88, Advances in Cryptology' CRYPTO' 95, Berlin, Heidelberg: Springer, pp.248-261, doi:10.1007/3-540-44750-4\_20, 1995.
- [12] J. Patarin, Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new Families of Asymmetric Algorithms, U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, Lecture Notes in Computer Science Vol.1070, pp. 33-48, pp.33-48, 1996.
- [13] J. Ding, D. Schmidt, Rainbow, a New Multivariable Polynomial Signature Scheme, Ioannidis, John (ed.), Proc. of Third International Conference, ACNS 2005, New York, NY, USA, Lecture Notes in Computer Science, Vol. 3531, pp. 164-175, 2005.
- [14] J. Buchmann, E. Dahmen, A. Hülsing, XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions, Lecture Notes in Computer Science Vol. 7071 (Post-Quantum Cryptography. PQCrypto 2011), pp.117-129, doi:10.1007/978-3-642-25405-5\_8, 2011.
- [15] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, Z. Wilcox-O'Hearn, Oswald, Elisabeth; Fischlin, Marc (eds.), SPHINCS: practical stateless hash-based signatures, Lecture Notes in Computer Science, Vol. 9056. Springer Berlin Heidelberg. pp. 368-397, doi:10.1007/978-3-662-46800-5\_15, 2005.
- [16] M. Naor, M. Yung, Universal One-Way Hash Functions and their Cryptographic Applications, Proceedings of STOC 1989, pp. 33-43, DOI: 10.1145/73007.73011, 1989.
- [17] R. Overbeck, N. Sendrier, Code-based cryptography, in D.J. Bernstein, J. Buchmann, E. Dahmen (Eds), Post-Quantum Cryptography. pp. 95-145, doi:10.1007/978-3-540-88702-7\_4, 2009.

- [18] L.De Feo, P.Jao, Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies, In: Yang BY. (eds) Post-Quantum Cryptography, PQCrypto 2011, Lecture Notes in Computer Science, vol 7071. Springer, Berlin, Heidelberg, pp.19-34, 2011.
- [19] W.Diffie, P.C.van Oorschot, M.J.Wiener, Authentication and Authenticated Key Exchanges, in Designs, Codes and Cryptography, Vol.2 pp.107–125, DOI:10.1007/BF00124891, 1992.
- [20] X.Sun, H.Tian,Y.Wang, Toward Quantum-Resistant Strong Designated Verifier Signature from Isogenies, in IEEE Proc. Intelligent Networking and Collaborative Systems (INCoS), 2012, pp. 292–296, doi:10.1109/iNCoS.2012.70, 2012.
- [21] R.A.Perlner, D.A.Cooper, Quantum Resistant Public Key Cryptography: A Survey, IDtrust '09: Proceedings of the 8th Symposium on Identity and Trust on the Internet, pp.85-93, doi: 10.1145/1527017.1527028, 2009.
- [22] M.Campagna, T.Hardjono, L.Pintsov, B.Romansky and T.Yu, Kerberos Revisited Quantum-Safe Authentication, in ETSI Quantum-Safe-Crypto Workshop September 15 2013, pp.1-18, 2013.
- [23] Manish Kalra & Ramesh C. Poonia (2017) Design a new protocol for quantum key distribution, *Journal of Information and Optimization Sciences*, 38:6, 1047-1054, DOI: 10.1080/02522667.2017.1374723
- [24] Manish Kalra & Ramesh C. Poonia (2018) Simulation of BB84 and proposed protocol for quantum key distribution, *Journal of Statistics and Management Systems*, 21:4, 661-666, DOI: 10.1080/09720510.2018.1475075
- [25] Ankit Kumar, Pankaj Dadheech, Vijander Singh, Ramesh C. Poonia & Linesh Raja (2019) An improved quantum key distribution protocol for verification, *Journal of Discrete Mathematical Sciences and Cryptography*, 22:4, 491-498, DOI: 10.1080/09720529.2019.1637153
- [26] Ankit Kumar, Pankaj Dadheech, Vijander Singh, Linesh Raja & Ramesh C. Poonia (2019) An enhanced quantum key distribution protocol for security authentication, *Journal of Discrete Mathematical Sciences and Cryptography*, 22:4, 499-507, DOI: 10.1080/09720529.2019.1637154
- [27] VV.AA. [https://wiki2.org/en/Post-quantum\\_cryptography](https://wiki2.org/en/Post-quantum_cryptography), retrieved 18-01-2020.
- [28] VV.AA, OpenQuantumSafe, [openquantumsafe.org](https://openquantumsafe.org), retrieved 18-01-2020.

- [29] D.Stebila and M.Mosca, Post-Quantum Key Exchange for the Internet and the Open Quantum Safe Project, Invited Lecture at Selected Areas in Cryptography (SAC) 2016 by D. Stebila. References and related work updated, <https://eprint.iacr.org/2016/1017>, July 28, 2017.
- [30] C.H.Bennett, G.Brassard, Quantum Cryptography: public key distribution and coin tossing, Int.Conf. on Comp., Syst. Signal Processing, Bagalore, India, pp.175-179, 1984.
- [31] S.Wiesner, Conjugate Coding, SIGACT News, 15:1, pp.78-88, 1983.
- [32] P.Higgins, Pushing for Perfect Forward Secrecy, an Important Web Privacy Protection, Electronic Frontier Foundation, Retrieved 18-01-2020.

*Received February, 2020*

*Revised June, 2020*