Quantum Technologies in Support for 5G services: Ordered Proof-of-Transit

Alejandro Aguado[1], Diego R. Lopez[2], Victor Lopez[2], Fernando de la Iglesia[2], Antonio Pastor[2], Momtchil Peev[3], Waldimar Amaya[4], Ferran Martin[4], Carlos Abellan[4] and Vicente Martin[1]

[1]Center for Computational Simulation, Universidad Politécnica de Madrid 28660 Madrid, Spain
[2]Telefonica Investigacion y Desarrollo, Ronda de la Comunicacion s/n 28050 Madrid. Spain
[3]Huawei Technologies Duesseldorf GmbH, Riesstrasse 25, 80992 Munchen. Germany
[4]Quside Technologies S.L., Castelldefels 08860 Barcelona, Spain
email: a.aguadom@fi.upm.es

Keywords: Quantum Cryptography, Network Functions Virtualization, 5G

Abstract

This work presents a method for supporting traffic attestation (ordered proof-of-transit) in service chaining and other traffic flows that uses cryptographic algorithms in combination with quantum technologies (QKD, QRNG). This capability is implemented over the production Madrid Quantum Network facilities.

1 Introduction

The continuous development of new bandwidth-consuming applications makes it necessary that the network achieves a new degree of flexibility to accommodate such services in the existing infrastructure. In addition, the introduction of new class of service profiles and service level agreements (SLAs) due to the requirements coming from 5G networks makes this situation even more dramatic, forcing the network to evolve and embrace new architectural approaches. Apart from hardware developments targeting to achieve higher throughputs and lower latencies, the fundamental changes have come from advances in software architectures applied to the networking environment. Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) have raised as the two main avenues for enabling automation and making the infrastructure flexible enough to cope with these new demands. Particularly, the introduction of concepts such as service function chaining (SFC) allows to define a service workflow and to place network functions where they are needed on demand to maintain the defined SLAs. However, such flexibility comes at a price, specifically with respect to security and reliability of the service and its data.

One of the most common concerns in virtualized environments is related with data and traffic attestation. When it comes to SFCs, it is important for both, the end user and the operator, to be sure that the traffic is forwarded across all the network elements (NEs) in the user-defined chain, so then no intermediate node (e.g. a firewall) is bypassed, situation that could cause potential security breaches. Within the IETF, the SFC working group created a draft for a proof-of-transit solution. Recently, the authors of this paper contributed to enhance the solution by integrating a security measure for fixing potential vulnerabilities but, more importantly, to verify that the defined flow is traversed in the correct order (ordered proof-of-transit - OPoT) [1]. Still, this method relies on the security of existing key exchange techniques and on the level of randomness generated at/by the source node.

Quantum technologies provide the means to solve both issues. Firstly, Quantum Key Distribution (QKD) is considered to be an information theoretically secure (ITS) primitive for key distribution, as the information leaked by the systems can be bounded as tight as desired (also known as **ε**-security). As its security based is based on the laws of quantum physics, it is not dependent on advances in (quantum) computing, while it can guarantee forward and backward secrecy. Secondly, Quantum Random Number Generators are a hardware solution for providing truly random numbers from quantum physical processes, rather than using pseudo-random algorithmic-computational approaches.

This work provides a quick overview of the existing proposal for PoT, describes how quantum technologies allow OPoT, to later show the first experimental demonstration of the OPoT service integrating QKD based methods and true randomness brprovided by QKD and QRNG respectively.

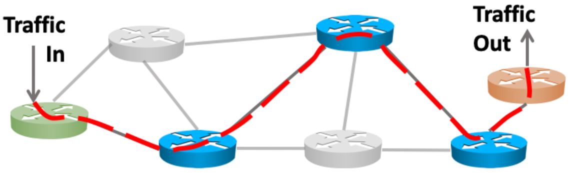


Fig 1 Nodes composing a PoT scheme. The scheme is formed by source (green), validator (orange), and intermediate (blue) nodes.

2. Proof-of-Transit

The IETF internet draft [1], to which authors of this work are contributing, defines a method for verifying that a traffic flow traverses all the desired nodes in a path. The technique is based on a modified Shamir's Secret Sharing scheme, in which all the parties must participate to reconstruct a secret. Every node in the path uses its share to this end. This secret (verification value) is finally validated by the termination node, which applies policies based on the outcome traffic flow received.

The mathematical problem behind the scheme is to construct a polynomial of degree *n* (in our case, over a finite field of order *p*) out of polynomial values (points). To do so we need these values at minimally *n+1* points. Each pair (*x*-axis value, *y*-axis value) is shared, together with some additional information, to each of the nodes being part of the scheme, so then all of them must contribute to build the final the secret (the constant term of the polynomial). However, this procedure can only be used for a single packet, since multiple usage allows an adversary to reconstruct the secret and render the procedure insecure. Therefore, the original idea proposed by Adi Shamir requires an extension that allows a random change of the polynomial per packet. Ref. [1] introduces a second polynomial such that its constant coefficient is randomly generated for each packet by the source node in the path. The reconstruction (interpolation) at each node creates then a cumulative value that is calculated as follows:

$$(1) \qquad CML_i^j = ((f_1(x_i) + f_2(x_i, RND^j)) * LPC_i) + CML_{i-1}^j (mod\, p)$$

where j is the packet transmitted, CML the cumulative value generated at node i, f_1 and f_2 represent the first and second polynomial, respectively, $RND^{\ j}$ is the random number generated by the source node and LPC_i is the Lagrange polynomial constant, provided to each node and defined as:

$$LPC_i = \prod_{\substack{l=1 \\ l \neq i}}^{n} \frac{x_l}{x_l - x_i} (mod\, p)$$

where x_q the point assigned to the node q. The destination node performs the very same reconstruction and finally verifies if the result is the same as the constant coefficient of the first polynomial plus the random number generated for that packet.

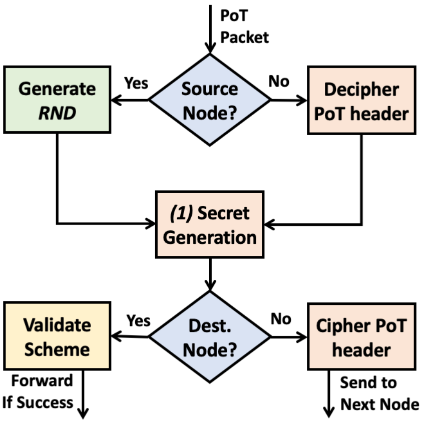


Fig 2 Flowchart representing the PoT logic implemented by the participant nodes

The secret reconstruction requires the packets to carry additional information (overhead) in the packet header including the *RND* and the *CML* values, the latter being used and updated by each node. A possible attacker can gather this information and try to perform a differential analysis of the values to replicate the schema. To tackle this issue, we propose to cypher or mask this data in a per hop basis to avoid any eavesdropper to learn from the PoT scheme. This provides an additional capability which is to also verify the order of the traffic flow, as any non-correct transition will generate an invalid decipher of the metadata, leading into an incorrect interpolation. The source of this masking or encryption comes from QKD for inter data centre instance-to-instance communications, whilst in a data centre domain QRNG acts as the source for the random masks. In addition, if the source of RND values is pseudo-random, backward and forward secrecy might not be guaranteed, whilst the schema requires a high throughput of random numbers at the source node, which can be handled by the QRNG service. A quantum random number generation service is set up to deliver sustained rates of 1 Gb/s. The quantum entropy system is based on sampling phase diffusion dynamics in gain-switched semiconductor lasers [2].

3 Enhanced Solution Approach

The enhanced solution requires a minimal set of modifications on each participant node: the capacity to cipher and decipher the packets metadata and the access to a source of random symmetric keys and numbers. The workflow diagram of the logic to be implemented by a participant node can be seen in Fig. 2. Firstly, every node shall match the incoming packets against its flow table. Any node, except the source, must gather the appropriate key to unmask the PoT metadata. Instead, the source generates the random number for the validation (green box). After this, the node uses its secret shares to reconstruct its part of the scheme, as depicted in the equation (1). The result is used to update the packet header, which is then ciphered before sending the packet to the next node. If the node is the verifier (destination), the last step is not necessary. Instead, the node validates if the scheme has succeeded and, if not, it will apply a policy for the given packet (e.g. drop it).

To provide an illustrative example, let us define a simple 2 hop (3 nodes) scenario. A controller selects a prime number $p=23$ and generates the two polynomials $P_1(x)= 11x^2 + 4x + 12$ and $P_2(x)= 22x^2 + 21x + C$, with C unknown and randomly generated by the source node. The controller also selects a point for every node in the path, calculating the *LPC* values, and shares the following data: (LPC_i, $P_1(x_i)$, $P_2(x_i)$, p). In this example, node$_1$:(3, 20, 22, 23), node$_2$:(1, 11, 16, 23), and node$_1$:(20, 12, 12, 23). The reconstruction works as follows:

- The source node generates a random number *RND=5*. This is used to generate the first *CML* based on its shares and the equation *(1)*. *CML=(22+20+5)*3 (mod 23)=3*. The node encrypts *RND (XOR(0x05,0x2C)=0x29)* and *CML (XOR(0x03,0x19)=0x1A)* and sends them to the next node.
- The intermediate node decrypts *CML* and *RND* with the correct keys (*0x2C, 0x19*). It uses *(1)* to update *CML=(11+16+5)*1 (mod 23)=12*. After this, it ciphers *CML* and *RND* again with next hop keys (*0x31,0x7b*) and sends both values to the next node (the validation or destination node).
- The validator deciphers *CML* and *RND* (*0x31,0x7b*), builds its own part of the share *CML=(12+12+5)*20 (mod 23)=17* and verifies the reconstruction using the secret of the first polynomial (the constant coefficient, *Secret=12)*, validates if the schema has succeeded *Secret + RND == CML*, or *12+5 == 17 (mod 23)* and removes the PoT headers from the packet, before sending it out.

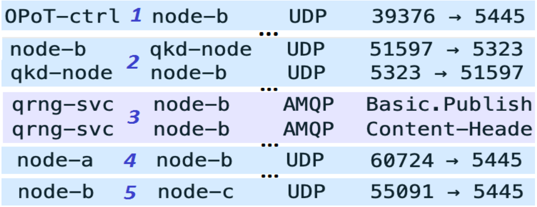
An incorrect reconstruction will lead into packet drops or will trigger an appropriate alarm to identify the reason for which the traffic is not forwarded in the appropriate order.

3 Implementation and Results

The testbed for this demonstration was implemented on top of the Madrid Quantum Network [3]. Three points-of-presence (PoPs) were distributed across the Madrid metropolitan area and connected using commercial equipment and production facilities. Two of the links implemented a quantum channel each, having the intermediate node in trusted relay mode for creating a virtual link between the two external ones. The QKD channels were co-propagating with other 17 data channels on the same fibres. In each PoP, we have installed hardware and software appliances to support a virtualization platform based on Linux containers. To showcase an OPoT-enabled service chain, a set of four instances is deployed distributed across the three PoPs, as shown in Fig. 3. Two platforms are in charge of managing the instances: a MANO orchestration securely instantiating the nodes (as in [4]) and a SDN controller with the capacity to compute and deploy the PoT scheme.

The PoT headers have been implemented following the structure defined by the in-situ OAM IETF internet draft, IOAM PoT type 0 [5], with 20 bytes of overhead. The source node gets a random number from a QRNG service. We implemented a proxy between the QRNG and the instances which is based on RabbitMQ. This proxy gathers a stream of random numbers from the QRNG service via REST API, while exposing two RMQ queues for control synchronization and RND delivery. This service is synchronized in a way that the very same random numbers are delivered to nodes b and c for ciphering their PoT metadata, via subscription. Other hops in the PoT chain cipher the metadata using QKD (a-b, c-d), which is better suited for securing inter-DC communications.

The metadata was updated as shown in the example at the bottom of figure of Fig. 4. The two polynomials generated by the controller are $P_1(x) = 1972880938x^3 + 679528847x^2 + 1178385690x + 1712903192$ and $P_2(x, RND) = 2102759155x^3 + 927893117x^2 + 329055405x + RND$ with a prime number $p = 2468775313$ and $RND = 397274676$ for the packet depicted in Fig. 4, in a 32-bit size scheme.



```
OPoT-ctrl 1 node-b        UDP    39376 → 5445
                     ...
 node-b     qkd-node      UDP    51597 → 5323
 qkd-node 2 node-b        UDP    5323 → 51597
                     ...
 qrng-svc   node-b        AMQP   Basic.Publish
 qrng-svc 3 node-b        AMQP   Content-Heade
                     ...
  node-a  4  node-b       UDP    60724 → 5445
                     ...
  node-b  5  node-c       UDP    55091 → 5445
```

Fig 3 Traffic capture at node B, including communication with controller, QRNG and QKD services and PoT forwarding

Fig. 3 shows the exchange of UDP and AMQP messages at node *b*: configuration from the controller (1), the access to QKD/QRNG services (2-3), and the final data traffic (4-5).

Both services, QKD and QRNG, could easily cope with the demand for keys and random numbers. To give an example, for a given 10Gbps traffic service, assuming that the maximum traffic is consumed and that we have an MTU of 1500, we will approximately have a total of 840.000 packets per second. This implies a maximum consumption of 27Mbps of random numbers from the QRNG. If multiple OPoT services are aggregated to a maximum of 100G, the consumption will reach 270Mbps, which is well within the limits of the QRNG service. The secret key demand for masking the PoT metadata is a lax requirement, negligible if the refresh is relaxed down to a key per minute (between 2 and 5 bits per second).

4 Conclusion

SFC is one of the bases to support future 5G services. This paper presents, for the first time, a method for verifying ordered proof-of-transit of a given service chain via Quantum technologies. The solution integrates a secret sharing algorithm, that is combined with QKD and QRNG services to enhance the security of the scheme and to provide an additional capability: order verification. This technique will serve as a security measure to certify new VNFs, providing traffic attestation in complex virtualized environments.

5 Acknowledgements

Authors would like to thank the support of the FET Flagship on Quantum Technologies, European Union's Horizon 2020 programme under grant agreement No 820466: Continuous Variable Quantum Communications (CiViQ), the Spanish Ministry of Economy and Competitiveness, MINECO /FEDER for the grant CVQuCo, TEC2015-70406-R, and Qrypt Inc. for providing the QRNG service.

6 References

[1] F. Brockners, et al. "Proof-of-Transit" IETF internet draft, draft-ietf-sfc-proof-of-transit

[2] C. Abellán, et. al "Ultra-fast quantum randomness generation by accelerated phase diffusion in a pulsed laser diode," Opt. Express 22, 1645-1654 (2014)

[3] V. Martin et al. "The Madrid SDN-QKD Network" presented at Qcrypt 2018.

[4] A. Aguado, et. al. "Hybrid Conventional and Quantum Security for Software Defined and Virtualized Networks", in JOCN, Vol. 9, Issue 10, pp. 819-825 (2017).

[5] F. Brockners, et al. "Data Fields for In-situ OAM" IETF internet draft, draft-ietf-ippm-ioam-data.

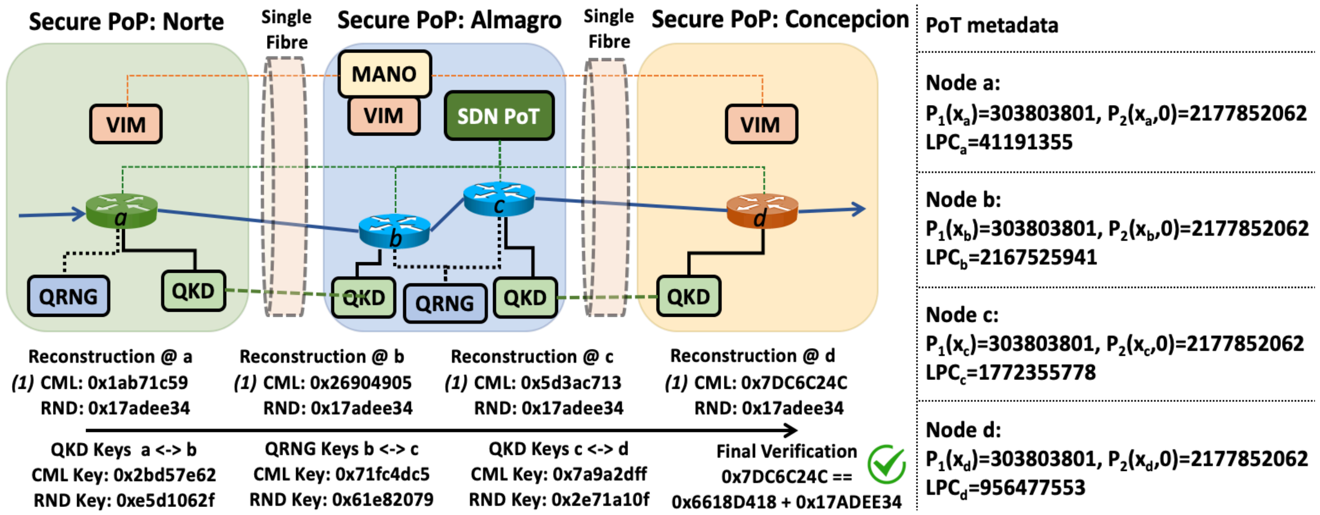


Fig 4 PoT scheme and implementation testbed deployed on top of the Madrid Quantum Network