QUANTUM INTERNET ALLIANCE

D4.3 Hardware Parameter Report

Document History

Revision Nr	Description	Author	Review	Date
0.1	First draft	Chin-Te LIAO		27/09/2021
1.0	Expansion and correction	Marc Kaplan		28/09/2021
2.0	Update backward simulation	Chin-Te LIAO		09/11/2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820445.

Index

1. Abstract

Multiple protocols have been simulated via NetSquid. The code was developed with maintainability and reusability in mind, and the simulation codes can easily be applied to different sets of hardware parameters. The protocols developed and uploaded in the software library are: E91 quantum key distribution, quantum tokens, quantum state teleportation, verifiable blind quantum computing, quantum anonymous transmission with W-states.

The developed protocols were used for benchmarking and analysing their performance in realistic conditions. The technical paper on protocol benchmarking is currently under submission. Realistic hardware parameters were applied according to state-of-the-art literature [6]. We also provided statistical plots to show how figures of merit are affected by certain protocol parameters. All of our benchmarks are repeatable, with the open-source code made available fully available.

2. Keyword list

quantum internet, quantum protocol, NetSquid,

3. Acronyms & Abbreviations

NS	NetSquid
VBQC	Verifiable Blind Quantum Computing
QKD	Quantum Key Distribution
QDS	Quantum Digital Signature

4. Introduction

As applications for quantum networks, quantum protocols provide different functionalities for various needs.

One of the main difficulties in research related to quantum communication consists in analysing how hardware limitations impact the performances and security of the protocols. Devices with optimal efficiency merely do not exist yet. However, using quantum network simulators like NetSquid, we are able to assess the security of these protocols with no access to the hardware, using only simulation techniques.

Apart from running simulations for certain sets of hardware parameters, with the help from our research partners, we can do the opposite and answer questions such as: what parameters are required in order to reach some target given a specific figure of merit of a protocol.

5. Implementations

This section presents our protocol simulations. We have implemented on Netsquid the following tasks for quantum networks:

- Quantum money
- Quantum key distribution
- Quantum digital signatures
- W-state Anonymous Transmission
- Verifiable Blind Quantum Computing

The list of protocols was established with our partners from SAP. They were selected on the basis of their potential value for profitable projects. Moreover, these protocols were also used in D4.1 *Quantum Applications and Use Case Report*. In this deliverable, the same protocols considered here were matched with potential end-user applications in various domains such as payment, banking, machine learning, etc…

5.1. Protocols

Quantum money

Quantum money is a set of applications in which the feature of unforgeability is applied to some quantum object used as “money”. These money objects can later be verified by parties that hold certain side information. We chose to simulate quantum tokens as an example, since it is relatively straightforward in terms of functionality.

Details of this protocol and analysis can be found in reference [1].

We considered the following figures of merit:

- The time waited for a party holding a token before verification.
- The possibility that a fake token be verified.

The simulation parameters were chosen based on nitrogen-vacancy (NV) center implementation. Our simulation quantifies the effect of the client wait time on the number of pairs required to achieve a given security level. The following graph, and following ones are extracted from Reference [3].

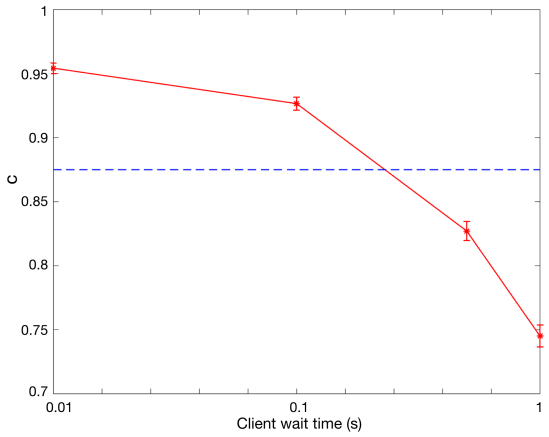


Figure 1. c versus client wait time, T. The blue dashed line shows the security threshold 0:875.

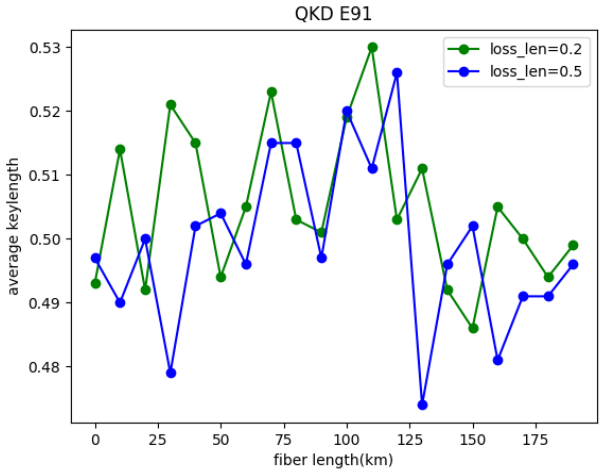
Quantum Key Distribution

Quantum Key Distribution (QKD in short) is one of the most fundamental and critical parts of quantum-related application. QKD allows to establish shared keys among distant parties that can be used, for example, in a symmetric encryption algorithm. We have implemented the E91 protocol which, compared to BB84, is slightly more practical in terms of logic implementation.

The details of this protocol and analysis can be found in Reference [2].

Figure of merit considered:

- Keyrate (How many key bits can be generated per unit time).
- Average keybit length per qubits.



The figure above shows how the figure of merits varies with different hardware settings.

Quantum Digital Signature

In-depth studies of Quantum Digital Signature are mentioned in reference [3]. The full protocol description is in Appendix C of the paper. We sketch it here for completeness.

This protocol has three parties, Alice, Bob, and Charlie. Alice sends the signed message to Bob. Bob authenticates the message and forwards it to charlie. The protocol then consists of two parts: a distribution stage and messaging stage. In the distribution stage, a key generation protocol (KGP) is performed by Alice-Bob and Alice-Charlie separately for each possible message m = 0 or m = 1. In the messaging stage, Alice sends a signed message to Bob. Bob checks the mismatches between the signature and the data exchanged at the distribution stage. He accepts the message and forwards it to Charlie if the number of mismatches is below some threshold. Similarly, Charlie will accept the message if the number of mismatches is below some threshold.

Figure of merit considered:

- Abort probability when all parties are honest
- Probability of forging Bob's signature
- Probability of repudiation

We have investigated the effect of channel losses and measurement errors on these figures of merits. This has allowed us to identify the secure regime for these parameters. The parameter range is described in Reference [3].

W-state Anonymous Transmission

Anonymous transmission addresses the issue of concealing the identity of two communicating nodes in a quantum network with N nodes. W-state anonymous transmission is a protocol which is considered to be the most efficient among anonymous transmission protocols. This protocol also contains quantum state teleportation as a sub-protocol. The figures of merit here are the quality of teleported states and the probability of protocol failure. Since the algorithm significantly reduces the chance of mistakes during the anonymous phase, we simply use the similarity between the original qubit and the teleported qubit as its figure of merit.

Details of this protocol and analysis can be found in reference [4].

We have used NetSquid to simulate this protocol for four users. We evaluated the effect of the noise introduced by quantum memories on the quality of the teleported state, the performance of the protocol in the presence of noise at the X and Z gates in the final step of teleportation and the effect of sources of loss in the system on the probability of protocol failure.

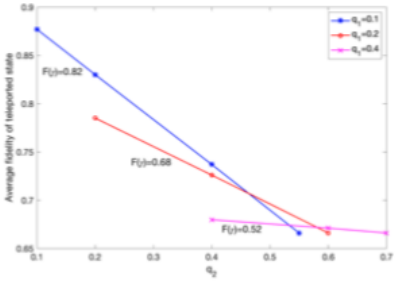


Fig. 5. Average fidelity of the teleported state for different values of q_1 and q_2.

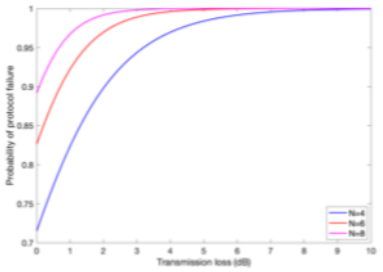


Fig. 7. Probability of protocol failure versus transmission loss for different number of users.

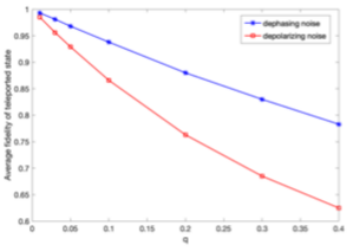


Fig. 6. Average fidelity of the teleported state versus q.

Verifiable Blind Quantum Computing

Verifiable Blind Quantum Computing is the most complicated one among the above protocols, because it consists of two parts: verification and computing. The latter does the real computing while the former checks the integrity of the message.

The figure of merit we considered for this protocol is the abort rate when no intentional hacking has occurred. The protocol aborts only when the integrity check fails. Of course, we want the abort rate to be as low as possible. Another figure of merit for this protocol is probability of correctness, which is defined as the probability of correct output assuming the protocol is not aborted.

The details of this protocol and analysis can be found in Reference [3]. The protocol description is in Appendix B.

We studied the effect of noise in quantum gates and measurement errors. Using realistic parameters, we have determined the number of test runs required to get a correctness probability of 0.929±0.0092 with a confidence level of 95%.

5.2. Simulations

Simulations of the above protocols are run by NetSquid in python. They were all made open-access. The repository can be found at https://github.com/LiaoChinTe/netsquid-simulation (Reference [5]).

5.3. Backward simulations

We introduced an alternative way to benchmark protocols. Our current workflow follows a number of steps. First we fix parameters. Secondly, we run the simulation. Then finally extract the value of the targeted figure of merit.

Thanks to some additional software optimization machinery, and with the help of Francisco Ferreira da Silva at TUDelft, we are able to turn the workflow backward. For a certain value of the figure of merit, we are able to directly compute the parameters required to achieve it.

However, to integrate the optimization machinery with protocol simulations requires some further work. The methodology differs greatly depending which protocol it is applied to. Therefore, we have only applied it on quantum money so far. Our results show that this approach is promising to explicitly compute the parameters required to achieve some target value according to some figure of merit. In this case, the security threshold of 0.875 accuracy on final measurement.

Note that the computational power required for simulation goes higher exponentially as each parameter we added to optimizing. Therefore, in our study, we applied only two parameters(T1 and T2) to optimize for each value of “party A storage time”:

Table 5. Optimum solutions for values of T_1 and T_2.

Storage time (s)	T_1 (h)	T_2 (s)
1	10.037	3.25
2	10.05	6.21
5	10.099	16.007

We believe that the minimal viable demonstration could be performed under the assumption that other hardware parameters are reached. And since the hardware parameters are based on the NV-platform, we have more confidence in the results.

The detailed methodology, analysis and parameters we applied are reported in reference [3].

6. Conclusion

The long version of our works can be found references [3] and [5]. This Includes the description of the protocol, the realistic choices of hardware parameters we applied, the results obtained, and the complete open-source code that we have developed for this project.

Using the software library we have built, simulations and performance analysis are easier to run. The atomic functions are well defined and called by our various protocol implementations. Furthermore, it is easy to extend with more protocols, and to apply different sets of hardware parameters.

Using this methodology allows us to have a much clearer idea on how quantum applications will be implemented in the future. We expect more studies to follow from the work described in this deliverable.

7. References

[1] https://wiki.veriqloud.fr/index.php?title=Quantum_Token

[2] https://wiki.veriqloud.fr/index.php?title=BB84_Quantum_Key_Distribution

[3] “Benchmarking of Quantum Protocols” by Chinte Liao, Sima Bahrani, Elham Kashefi, submitted, availabile at https://cloud.veriqloud.fr/index.php/s/iiw1SxU4D22FyQ7

[4] "Anonymous transmission in a noisy quantum network using the W state" by V. Lipinska, G. Murta, and S. Wehner

[5] https://github.com/LiaoChinTe/netsquid-simulation

[6] “Netsquid, a discrete-event simulation platform for quantum networks” by T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. Oliveira, M. Papendrecht, J. Rabbie, F. Rozpedek, M. Skrzypczyk et al