

Quantum Internet Hackathon Pad

in case you want to keep up to date after the hackathon, please subscribe to the mailing list:
<https://www.irtf.org/mailman/listinfo/qirg>

Bacis info:

This pad: <https://pad.riseup.net/p/quantum-internet-hackathon-2018-keep>

Announcement:

<https://labs.ripe.net/Members/becha/join-the-quantum-internet-hackathon-2018>

IRC Channel:

irc.oftc.net #ipv6hackathon

Mailing list:

<https://lists.ripe.net/mailman/listinfo/quantum-internet-hackathon>

Vesna's phone number:

+31 6 2125 8191

GitHub Repos

RIPE NCC / RIPE Atlas / hackathons! Main:

<https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib>

The the directory for quantum internet hackathon:

<https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/upload/master/quantum-internet-hackathon/>

& a directory for presentations

<https://github.com/RIPE-Atlas-Community/ripe-atlas-community-contrib/upload/master/quantum-internet-hackathon/presentations>

More links:

<http://quantum-internet.team/simulagron-hackathon/>

Twitter:

#QuantumHackathon #QIA

Quantum Internet Alliance @eu_qia @QuTech_news @RIPE_NCC @JuniperNetworks @Ms_Multicolor

Photos:

Hiro: <https://drive.google.com/drive/folders/1xYyc1fvAVsH8VpWVAvgGCd6UhutKZyM?usp=sharing>

Tim: <https://www.dropbox.com/sh/79u3kfb1xhbua11/AAA1FiCOJWVQFLO5kQr4Xgi-a?dl=0>

Time Planning

13-14.10. Saturday & Sunday = working days for the hackathon: VolksHotel: Wibautstraat 150

12.10. = 7PM Friday evening: visit to the hackerspaces: LAG and/or TechInc

https://wiki.techinc.nl/Hackathon_visit

15-19.10. RIPE77, Okura Hotel: <http://ripe77.ripe.net>

Useful Links:

*

https://github.com/SoftwareOuTech/SimulaQron/raw/Hackathon2018/hackathon_instructions.pdf

Instructions document for getting started with SimulaQron and proposed challenges.

* <https://github.com/SoftwareOuTech/SimulaQron> The SimulaQron code.

* www.simulaqron.org The SimulaQron website.

* <https://arxiv.org/abs/1712.08032> The SimulaQron paper.

* <https://softwarequtech.github.io/SimulaQron/html/GettingStarted.html> SimulaQron documentation.

* <https://github.com/mdskrzypczyk/QChat> The code for QChat, an application pruning on SimulaQron.

PROJECTS

* Simple Docker image of SimulaQron

A lot of people had problems installing and starting SimulaQron. Here's a very basic SimulaQron Docker image that runs the default startAll.sh giving you five nodes:

<https://hub.docker.com/r/mchackorg/simulaqron/>

The ports exposed are 8081--8085.

* implications of quantum internet on routing protocols (Kireeti Kompella)

As I understand (your diagram shows both a QNIC and a NIC), the quantum internet may be paired with the classical internet, the former for key exchange, the latter for data exchange. Furthermore, quantum networking may be more restricted by distance than classical networking (more quantum repeaters). Are those two assumptions correct?

So the question of how to guarantee secure connections in such a quantum-classical pairing.

* Expanding a quantum network simulator (Takaaki Matsuo)

The simulator is built on top of a classical network simulation (OMNeT++), and we are trying to design and figure out how we can share Bell pairs across 2 nodes over networks. This includes stochastic link-level Bell pair distribution (protocol that does this autonomously/periodically), quantum routing (classical routing techniques can be adopted directly, but we will need to define the cost of each link, and how to estimate it, error correction/purification (so we will consider noisy channels, memory, gate operations and etc.) and so on.

the abstract we wrote for a conference may give you an idea of what we are trying to achieve.

[Challenges in Simulating the Quantum Internet] <http://www.ngc.is.ritsumei.ac.jp/~ger/static/AQIS18/OnlineBooklet/176.pdf>

[End-to-End Connection Setup in a Quantum Internet] <http://www.ngc.is.ritsumei.ac.jp/~ger/static/AQIS18/OnlineBooklet/81.pdf>

* Quantum BitCoin (Wojciech Kozlowski)

https://join.slack.com/t/quantum-bitcoin/shared_invite/enQtNDU1NDc0MDkxNTg3LWI2OWM2ZTZjNWZlZWQ0YzEwODNmZjIyZWZhYWY0MjBiYzZmZDA1YjRiMTU2YjNkYTg0MTYyYjE5ZDEyMzg0WE

At the core of BitCoin is a distributed algorithm for reaching consensus without trust that relies on a proof-of-work mechanism. Could we come up with a better consensus algorithm that is quantum? I think there are quantum leader election algorithms already, but a cryptocurrency has more stringent conditions: https://en.wikipedia.org/wiki/Byzantine_fault_tolerance.

What is there to be improved upon in the current mechanism:

1. The existing BitCoin mechanism is incredibly wasteful in terms of energy as it requires using up lots of CPU power without actually producing a useful outcome (other than consensus). Would a quantum algorithm be possibly more efficient in terms of processing power required?
2. Bitcoin is pretty secure, but it is vulnerable to what is called a 51% attack. That is, if a single party controls more than 50% of the CPU resource in the BitCoin network they would be able to completely take it over, block payments, change transaction history etc. Can a quantum algorithm improve upon this?

What would be within the scope of the project?

- 1) Find a suitable quantum algorithm
- 2) If can't find one, try to modify an existing one or settle for the best option available
- 3) Implement in SimulaQron
- 4) Integrate with the actual bitcoin source code (<https://github.com/bitcoin/bitcoin>).
- 5) Run a quantum cryptocurrency network as a demo.

In addition to a technical interest, I think this project has potential for some good PR for quantum networks. Cryptocurrency and blockchain easily makes it into the news and everybody reads about it

W state generation: <https://arxiv.org/pdf/1606.09290>

* An attempt at Golang bindings to SimulaQron

SimulaQron has a network protocol/sets of instructions known as CQC. Current bindings to CQC are in Python and C. Seeing that QUTech want to use CQC on real quantum computers in the future it would be nice to have more language bindings, for instance Go bindings.

During reading of CQC specs found a number of bugs in the specification as compared to the Python `pack()` format strings. PR #88, now merged.

CQC interface should probably use network byte order. In Python just change from "=" to "!" in `pack()`. C code needs more changes, but simple calls to `ntohs()` and friends.

Made a stupid simple Docker image for SimulaQron. <https://hub.docker.com/r/mchackorg/simulaqron/>
Get it easily with `docker pull mchackorg/simulaqron`

/MC

* Quantum Digital Signature (Shota Nagayama)

We implement quantum digital signature algorithm on SimulaQron.
https://en.wikipedia.org/wiki/Quantum_digital_signature

Note from the organization (8 Oct): we have defined a few challenges that you might choose to work on during the weekend. We briefly outline them below:

Challenge A: testing the network. Real-life networks are faulty; can a node find out how faulty they are on the fly, so that it can adjust its actions to it? You could think about adding a network ping (or a quantum version of a ping) and visualizing network diagnostics.

Challenge B: take SimulaQron's network tools to a new level. Currently, networks in SimulaQron are static: no nodes can be added or removed. Again, real life is so different: new devices are added and removed to networks all the time, and what to think about server breakdowns?

Challenge C: taking QChat to new heights. The chat application QChat is built on top of SimulaQron for generating a shared secret classical key for secure communication. The current version does not work optimally in a noisy network (think error correction) and might also be improved in terms of speed.

Challenge D: whatever use of SimulaQron you come up with yourself. If you have any creative ideas for using or extending SimulaQron, you are of course free to work on them!

Equipment:

please bring with you:

- your own laptop (and charging cable)
- electricity converters from your country to EU standard

Optional:

- ethernet / LAN cable (and a converter from/to USB if you are Mac user ;-)
- VGA / HDMI video converter if you want to show slides or demos
- extra "extensions cord" with multiple sockets - specially if you have UK or Japanese plugs & want to charge your phone, or other equipment

We will have available:

- tables & chairs
- video projector
- audio equipment
- 4 flipcharts
- WiFi
- LAN cable
- electricity cables
- one big room - and break-out space
- switches - provided by Juniper
- extra laptops - provided by RIPE NCC, installed by QuTech

this is a "closed" event, so only people that have applied and are selected to participate are invited.

Please DO NOT bring extra people - colleagues, supporters , guests - without letting me know in advance, because we have the places & food & other provisions for the exact number of participants.

Detailed time planning:

Saturday, 13th October

- 09:00 Welcome
- 09:30 Presentations by RIPE NCC: Logistics, thanks to sponsors, timeline (Vesna)
- 09:40 Round of introductions (All)
 - * Round of personal introductions (1 minute per person)
- 10:00 Presentations from QuTech & Juniper (?)
- 10:30 Brainstorm & teams-forming
 - * Projects, Data Sources etc
 - * Ideas suggestion (2 minutes per person/team/idea)
 - = ALSO TEA&COFFEE BREAK!
 - + RE-ARRANGE THE TABLES in needed !!
- 11:00 - 12:00 WORK on projects (1 hours)
- 12:00 - 13:00 LUNCH break
- 13:00 - 19:00 WORK (6 hours)
- 18:30 = all = presenting progress & asking for input / help
- 19:00 dinner in VolksHotel
- 22:00 - midnight: OPTIONAL - continue working! (3 hours, OPTIONAL)

Schedule Day2:

- 9:00 arrive & start working
- 09:00 - 11:00 WORK (2 hours)
- 11:00 (15 minutes) all: Presenting progress
- 11:15 - 13:15 WORK (2 hours)
- 13:15 LUNCH break
- 14:00 - 17:00 WORK (3 hours)
- 17:30 wrap-up: presenting final results & planning follow-up steps
 - * 5 minutes per team!!

18:30 - JURY gathers in a smoke-filled room & comes up with feedback & a "winner" of stroopwafels!

19:00 - closing ceremony

19:55 - LEAVE FOR DINNER

20:00 dinner in Drovers Dog

Sunday dinner venue: Drovers Dog, next-door to Volkshotel <http://drovers-dog.com/dog3/menu/>

List of participants, alphabeticly sorted, 21-09-2018

Name	What You Can Do
Amjd Yousef Majid web developer.	C, Python, Java, MAC protocol, Embedded Software dev, full stack
Anders Rehult	
Andrey Hoursanov	best used for brainstorming, moderation, critical thinking;
under pressure might recall how to code in .py or .java :)	
Anne Marin	
Anton Karazeev	
Asbjørn Sloth Tønnesen	
Axel Dahlberg	
Bram Dobbelaar	
Christopher McIntyre	Python, Linux, Docker, Ansible, Erlang/Elixir
Cristian Sirbu	Python, Ansible, Linux, Docker, Legacy :) Networking,
PM/Moderator/Aggregator	
Daniel Karrenberg	
Daniel Quinn	Python, Javascript, APIs
Georg Harder quantum, python	
Harsh	
Hiroataka Nakajima	Python, Automation, Classical Networking (Routing, Application
Layer), blockchain (little)	
Kireeti Kompella	C, Python (same as Andrey :)); routing protocols, classical
networking	
Leon Wubben	
Marc Gaensler	Physics, Quantum Algorithms, little Python, convert emerging
tech into new products and applications	
Marc Ibrahim	Python, C/C++, classical networking
Markus Braun	
Martin Stam	
Matt Skrypczyk	
Matthias M. Hudobnik	
Melchior Aelmans	
Michael "MC" Cardell Widerkrantz	C, Go, Python, FreeBSD, Linux, network protocol dev.
Oleksandr Mykhalevych	C/C++, low-level dev, Networks, Cybersecurity, Linux
Pedro da Silva Vaz	
Petros Gigis	
Przemyslaw Pawelczak	
Rakshit Midha	
Sergio Freitas	
SeyedAlireza Vaziri	
Shinnosuke Ozawa	
Shota Nagayama	C/Python, Quantum Networking, IPsec, Internet Key Exchange
Shraddha SINGH	
Stephanie Wehner	
Syed Affan Aslam	
Takaaki Matsuo	
Takahiko Satoh	
Tim Coopmans	
Tudor-Stefan Tabacel-Manea	Python, Ansible, Bash, Linux
Vesna Manojlovic	
Wojciech Kozlowski	Quantum physics PhD, C/C++/Rust/Python programming, network
protocol dev	
Yao MA	Low-level system design, Linux, C, C++

venue for Saturday & Sunday is

Volkshotel

Wibautstraat 150

1091 GR Amsterdam

+31 (0)20 2612 100<https://www.volkshotel.nl/en/>

To get there:

From Schiphol you can take the train to Amsterdam Central Station or Amsterdam Amstel Station.

From Amsterdam Central Station you can take Metro line 51,53,54 and get out at 'Wibautstraat'. Take exit "Gijsbrecht van Aemstelstraat", make a u-turn and you will see Volkshotel on your left. Metro runs: weekdays 06:15-00:30 and weekends 07:30-00:30.

Trams that stop not-too-far-away are 3,7 & 12

Getting around

=====

Using public transport:

- train & metro/bus/tram tickets are electronic, but not interchangeable
- You can pre-order your tickets online, for a single train ride or unlimited travel across Amsterdam for 1, 2 or 3 days.

- there are machines to buy tickers from, and you can buy them on some trams & buses - but NOT on the train!

<http://ns.nl><http://en.gvb.nl><https://www.discoverholland.com/tickets>

Or you can rent a bike:

- there is "swapfiets" on the station: <https://swapfiets.nl/en/>
- or MacBike: <https://www.macbike.nl/>

Accommodation tips:

=====

Accommodation in Amsterdam is easy to book, but hard to find : everything is always either full or very expensive.

For a listing of some hotels, recommended for the RIPE meeting, see:<https://ripe77.ripe.net/venue/accommodation/>

For the less expensive options, some people have recommended:

1. Across the street:<https://www.thestudenthotel.com/amsterdam-city/>
2. In the boonies, but along the metro line:<https://www.aohostels.com/en/amsterdam/amsterdam-zuidoost/>
3. Across the water, in the "north" part of the city:<http://www.clinkhostels.com/ClinkNoord>

Team aMBiQuouC (Blind Quantum Computation)

=====

Project repository: <https://github.com/cgmcintyr/SimulaQron>,
<https://github.com/cgmcintyr/SimulaQron/projects/1>

Team QHamser (QChat challenges)

=====

Project repo+branch: <https://github.com/mdskrzypczyk/QChat/tree/hackathon>

Accomplished Work:

1. Re-organized code to separate client, server, and core functionality
2. Added XML-RPC server for clients
3. Added XML-RPC console client to interact with XML-RPC server
4. Added XML-RPC web client to interact with XML-RPC server
5. Substituted signature mechanism with PKCSv1.15 for improved speed
6. QChat can use remote Simulacron nodes for CQC backend

16:17 < asbjorn> Current development tree:

<https://github.com/SoftwareQuTech/SimulaQron/tree/Develop>

16:17 < asbjorn> Not yet merged: <https://github.com/SoftwareQuTech/SimulaQron/pulls>

16:18 < asbjorn> Well becha hasn't re-joined yet

DISCLAIMERS:

WARNING: This pad will be DELETED if 60 days go by with no edits. There is NO WAY to recover the pad after this happens, so be careful!

If you want a pad to be kept for up to one year (365 days) without edits, append the word "-keep" to the end of the pad name.

If you want a pad to be deleted after one day, append the word "-tmp" to the end of the pad name.

For example:

- * <https://pad.riseup.net/p/1234-tmp> => deleted after 1 day of inactivity
- * <https://pad.riseup.net/p/1234> => deleted after 60 days of inactivity
- * <https://pad.riseup.net/p/1234-keep> => deleted after 365 days of inactivity

Riseup is a collective providing secure online communication tools for people and groups working for liberatory social change. Riseup depends on donations from users like to keep going. Please visit <https://riseup.net/donate> to contribute.

Abusive behavior is not allowed on this service. Please visit <https://support.riseup.net> to report any problems.