Quantum Secured Permissioned Blockchain. Consensus and Logic.

Xin Sun[1,2], Mirek Sopek[3], Quanlong Wang[2] and Piotr Kulicki[4]

1 - Institute of Logic and Cognition, Sun Yat-sen University, China: xin.sun.logic@gmail.com
2 - Department of Computer Science, University of Oxford, UK: quaang@cs.ox.ac.uk
3 - MakoLab SA, Lodz, Poland: sopek@makolab.com
4 - Department of the Foundations of Computer Science, Catholic University of Lublin, Poland: kulicki@kul.pl

Abstract

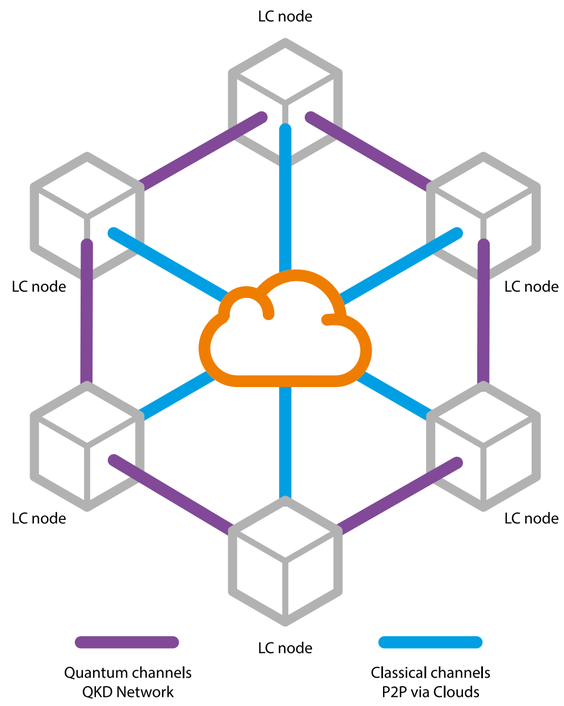
While Blockchain technology is universally considered as a significant technology for the near future, some of its pillars are under a threat of Quantum Computing. In this poster, we propose measures against this threat by developing a framework of quantum secured, permissioned Blockchain called LogiContract (LC). LC adopts the message authentication scheme based on QKD mechanisms and a vote-based consensus algorithm to achieve consensus on the blockchain.

Intro

There is a considerable amount of research related to the quantum-safe Blockchain [1–4] which could withstand attacks powered by forthcoming quantum computers. One of the most prominent proposal is the Quantum-secured Blockchain (QB) developed by Kiktenko [1]. Due to the application of unconditionally secure message authentication based on QKD, QB is immune from the attacks of quantum computers. However, the major limitation of QB is the consensus protocol it adopts. In QB, the classical Byzantine agreement protocol [5] is used to achieve consensus among nodes of the distributed ledger which has serious shortcomings as it becomes exponentially data-intensive if a large number of cheating nodes are present.

In this poster we report our research on a new consensus protocol for Quantum Blockchain which exhibits quadratic dependence of resources on the number of miners. We have also developed an idea of **the scripting language for LC for the future implementation of smart contracts.**

LogiContract Architecture



We assume that in LC each pair of nodes is connected by **a classical channel and by quantum channels** which form **Quantum Key Distribution (QKD) NETWORK.**

Each pair of nodes establishes a sequence of private keys using the QKD network. Those keys will be used for **message authentication** and **the digital signature**.

Unconditionally Secure Authentication and Signature

Unconditionally secure authentication schemes have been extensively studied in the literature [6-8]. Carter and Wegman [6,7] were the first to construct authentication codes using hash functions. They also show that one-time pad encryption can be used in combination with hash functions to construct unconditionally secure authentication schemes. This approach was further studied, among others, by Krawczyk [8].

We have extended the unconditionally secure authentication scheme based on the Toeplitz hash and made it useful for the Quantum BlockChain. For the signature we propose a new scheme which we called the Toeplitz Group Signature (TGS).

Quantum Secured Consensus

The consensus algorithm in LC is the **Quantum Secured YAC (QSYAC)**, which is a variant of the original YAC[9] developed for Hyperledger Iroha Blockchain, where the **Toeplitz Group Signature (TGS)** replaces the public-key signature. There are two types of participants in QSYAC: clients and peers. The participants are connected by a quantum key distribution network which is used to distribute private keys between participants. Moreover, every two peers are connected by a classical channel. Every client and every peer are also connected by a classical channel. We assume the channels to be synchronous.

Scripting language and Smart Contracts for LogiContract

LogiContract implements a scripting language for Smart Contracts. Scripts of transactions on LC are logical formulas which are built from arithmetic expressions e with the following syntax:

$$e ::= x \mid k \mid e + e \mid Hash(e)$$
$$\phi ::= e = e \mid e > e \mid Odd(e) \mid After(e) \mid \neg\phi \mid \phi \wedge \phi$$

Here x is a variable with range over natural numbers, $k \in N$ is a constant natural number. *Hash* is a collision-resistant hash function on natural numbers. *Odd(e)* means that e is an odd number.

Using the logic-based script language we can form a formal definition of a transaction on LC. As a Proof-of-Concept we have built an example of the lottery protocol on Quantum Blockchain.

References

[1] Kiktenko, E.O. et al. **"Quantum-secured blockchain"**. Quantum Science and Technology 2018, 3.
[2] Aggarwal, D. et al. **"Quantum Attacks on Bitcoin, and How to Protect Against Them"**. Ledger 2018, 3.
[3] Stewart, I. et al. "**Committing to quantum resistance: a slow defence for Bitcoin against a fast quantum computing attack**". Royal Society Open Science 2018, 5.
[4] Sun, X. et al. "**A Simple Voting Protocol on Quantum Blockchain**". International Journal of Theoretical Physics 2019, 58, 275–281.
[5] Pease, M.C. et al. **"Reaching Agreement in the Presence of Faults"**. J. ACM 1980, 27, 228–234.
[6] Carter, L. et al. **"Universal Classes of Hash Functions"**. J. Comput. Syst. Sci. 1979, 18, 143–154
[7] Wegman, M.N. et al. "**New Hash Functions and Their Use in Authentication and Set Equality**". J. Comput. Syst. Sci. 1981, 22, 265–279.
[8] Krawczyk, H. **"New Hash Functions For Message Authentication"**. Advances in Cryptology - EUROCRYPT '95, Proceedings; Springer, 1995, Vol. 921
[9] YAC - "**Yet Another Consensus**" - https://cn.hyperledger.org/projects/iroha