A Small Java Application for Learning Blockchain

Xing Liu
Dept. of Computer Science and Information Technology
Kwantlen Polytechnic University
Surrey, Canada
xing.liu@kpu.ca

***Abstract* — This paper introduces a small Java application named ChainTutor for learning basic Blockchain concepts. Although the term Blockchain is widely known and Blockchain technologies are finding applications in various areas such as banking, healthcare and Internet of Things, some concepts of Blockchain are not easy for beginners to understand. Fully text-based tutorials are often difficult to follow. General picture of Blockchain operations gets lost in lengthy textual descriptions. With the Java application introduced in this paper, users can experiment with key Blockchain concepts through a graphical user interface. They can generate keys, hashes, transactions, blocks and wallets. They can see the low level details of a blockchain such as encryption keys and hashes. They can see how mining works and how blocks are added to a blockchain. Parameters of a blockchain can also be varied in order to observe their impact on performance or even to make a blockchain invalid. The Java application is intended to be used in classroom environment by instructors when they teach introductory Blockchain courses.**

Keywords — Blockchain, learning tool, Java application

I. INTRODUCTION

In recent years, there has been a surge of interests in Blockchain technology. Various industries are looking into adopting the technology in order to benefit from what Blockchain has to offer: improved security and privacy, distributed storage and data immunity to attacks [1]. Universities have started teaching courses on Blockchain in order to prepare students for Blockchain-based software and application development or for providing technical support to systems with blockchains in operation [2].

The learning curve of Blockchain technology is relatively steep for beginners. Although large Blockchain platforms such as Ethereum and Hyperledger Fabric are readily available, low level details of Blockchain technology are hidden. This is especially true for programmers and developers who are interested in knowing the very low-level information of Blockchain and understanding how Blockchain works.

The author has research interest in applying Blockchain technology to embedded systems and IoT, particularly to low-level software for resource-limited devices which can be used in applications such as smart homes, smart buildings and smart cities [3]. Understanding the low level principles of Blockchain is essential because it might be logical to utilize only selected functions of Blockchain in resources-limited devices, or to select appropriate blockchains for different use cases and applications.

The author taught a course which included a topic on Blockchain-based IoT. It was difficult to explain the concepts of Blockchain to students without a suitable learning tool. The author tried to find a simple tool which has a graphical user interface (GUI) for the students to learn and test the basics of Blockchain. However, nothing was found after a number of attempts. It motivated the author to develop the small Java GUI application introduced in this paper.

The work of this paper benefitted from the information provided in open resources found on the Internet, such as [4]. The author's main contribution is the creation of a compact GUI-based learning tool that can be used in a classroom environment by instructors and students. However, the project is still a work-in-progress because the development is not fully completed yet.

The paper is organized as follows. Section II gives a general introduction to Blockchain. Section III describes in detail on how the Java application was designed and developed. Section IV presents the test results and Section V provides the conclusion.

II. GENERAL INFORMATION ABOUT BLOCKCHAIN

A. What Is A Blockchain

Blockchain is the technological foundation of Bitcoin. Many definitions of Blockchain are available in the literature. The definition given by NIST (National Institute of Standards and Technology) [5] is a preferred one. The NIST definition states that Blockchain is "a distributed digital ledger of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify. New blocks are replicated across all copies of the ledger within the network, and any conflicts are resolved automatically using established rules".

From the definition above, it can be seen that a blockchain is essentially a database with a special structure of chained blocks. Data are encrypted to preserve privacy and are stored in the blocks. The chained structure and the validation and consensus mechanisms ensure data integrity. Replicated copies and distributed storage prevents data loss and enhance data security as well.

978-1-5386-7266-2/18/$31.00 ©2018 IEEE

As the world is generating more and more data every day because of the Internet, the World Wide Web, and recently the Internet of Things, the protection of data becomes increasingly important. On the other hand, Blockchain is considered to be a "revolutionarily" viable solution for protecting data due to its ability to decentralize data storage, enable data immunity against attacks, and provide enhanced security, integrity, and privacy to data. It is predicted that Blockchain will potentially affect many aspects of our lives.

B. Application Domains of Blockchain

A quick Internet search reveals that Blockchain is being applied to virtually any area we can think of, such as banking, finance, election voting, education, insurance, supply chain management, farming, healthcare, and agriculture, just to name a few. Another important area of Blockchain application is IoT [6]. IoT is trying to connect sensors and physical objects around the world and data security is of prime importance. Concerns about IoT security have been in existence since the concept was conceived. Blockchain is certainly a promising technology which may provide a feasible solution.

C. How Blockchains Work

A blockchain is very similar to a data structure in computer science called linked list. Instead of having "chained nodes" as what a linked list does, a blockchain has "chained blocks" [7][8]. In fact, "node" has a different meaning in a Blockchain system. However, the blocks in a blockchain are not chained based on block addresses. They are chained based on cryptographic hashes derived from the data in the blocks. That is, each block contains the hash of its previous block to form a chain. Due to this structure, if the data in a published block is changed due to a malicious attack, then its hash will be changed. This will cause the hashes of all subsequent blocks to change. This mechanism can be used to detect unwanted modifications of published blocks. This is also why there is the statement "published blocks are not modifiable in a blockchain".

An operational Blockchain system consists of numerous distributed nodes. Many or all nodes keep a copy of the blockchain. Every node can initiate transactions of data as well.

A transaction is a record of a data transfer often identified by its hash. Transactions are signed and can be verified. A block can contain multiple transactions. Transactions need to be valid otherwise they will not be included in a block. The validity refers to a transaction being properly signed and the signer has funds to make the transaction. The nodes that check for transaction validity are called *mining nodes*. Blocks containing invalid transactions will not be added to the blockchain.

To add a new block to a blockchain, a mining node prepares a candidate block using unspent transactions and other required information to build a block, such as the hash of the previous block, as well as a *difficulty* parameter called *nonce*. The candidate block is then propagated to other participating mining nodes which will validate and solve a challenging puzzle if a consensus model named *Power of Work* is adopted.

A mining node gets the right to publish the next new block and receive a reward if it is the first node to find a nonce value which solves the puzzle. Other mining nodes will validate the block and add the new block to their copies of the blockchain.

However, in a working blockchain, many mining nodes are trying to solve the puzzle and they may find a suitable nonce value at almost the same time. This is called a *conflict*. There are strategies in Blockchain that can help resolve such conflicts.

In Blockchain, there is another element called *wallet*. A wallet is a piece of software that stores participants' addresses (private and public keys) and fund balances. Wallets are used to initiate transactions too.

III. The Java Application ChainTutor

The technical concepts and operations of Blockchain are not simple. The learning curve may be steep for many people, and is certainly true for young students who want to learn programming and develop software involving Blockchain. Without a visual tool, the concepts of hashing and mining are difficult to grasp by beginners. Course instructors might have felt that it was not easy to explain the concepts to students as well. A software program that is tangible for students to "play with" would surely help with their learning.

With beginning learners in mind, a small Java application named ChainTutor has been developed which can be used in classroom teaching.

A. The Design of the Java Application

The structure of the Java application is represented by Fig.1. The application consists of the core Blockchain modules, plus a "Learn" Module, a "Set Up" module, a "Build Wallets" module, a "Build Transactions and Blocks" module, and a "Build Nodes and Network" module. These modules work together under the same graphical user interface and allow students to create, verify, and validate different components of a Blockchain.

B. Blockchain Core Modules

These are the core modules needed to build a Blockchain. They include a hashing module, a mining module (currently Power of Work only), and a validator module. These modules communicate with other modules to help them accomplish their tasks.

C. The Learn Module

This module is designed to help learners refresh their minds on key Blockchain concepts. It is similar to a vocabulary list. However, instructions and steps on how to build a blockchain are also included in this module.

D. The Set Up Module

This module is used to set up global parameters that apply to the entire blockchain. Currently this module only allows users to set up the value of *difficulty* for mining blocks. Other

parameters will be added as new modules and functionalities are added to the application.

E. The Build Wallets Module

Wallets are required in this application to send and receive transactions. In real Blockchain applications, wallets will be held by nodes that are members of the Blockchain system. This module lets users create wallets and their corresponding private and public keys. The wallets created can be used to create transactions later. The wallets also store fund balances.

F. The Build Transactions and Blocks Module

This module does the most important work. It allows users to create transactions between wallets, add transactions to blocks, create blocks, and add blocks to the blockchain. The module automatically chains the blocks up using their hashes, calls the mining function in the core module to mine a new block based on the *difficulty* value, and validates the entire blockchain after a new block is added.

G. The Build Nodes and Network Module

The proposed Java application can be installed on different nodes to form a distributed system. Each installation will allow the hosting computer (a PC, a Raspberry Pi, a microcontroller, or even a microwave in a smart home) to become a node on the blockchain. This module will facilitate the communications between nodes. This way the application can be used to simulate a small blockchain in a local environment such as a smart home, or in a larger environment involving the Internet.

H. Other Design Considerations

The Java application is designed with students in mind. Each module designed allows students to "experiment stuff", such as observe the amount of time needed to mine a block with different *difficulty* values, or intentionally change a hash value to "break the chain". The students will be able to see "what a hash value really looks like", and "what a private key looks like" and so on. They will be able to see the contents of all nodes in a blockchain, together with the details of the hashes after mining is completed.

IV. TESTS

The Java application developed is still some way from being complete. However, a basic functional version has been achieved. The following paragraphs provide test results for different scenarios when the application is running. All figures are placed at the end of the paper in order to have better readability.

A. The Main Window

Fig. 2 shows the screen when ChainTutor is started.

There are several main menus provided for users to experiment with different aspects of a blockchain. There are several submenus under each main menu as well. The blank area is the "stage" where new windows and details of Blockchain components will be displayed.

B. The Learn Menu

Clicking this menu will display a list of key words or terms frequently used in Blockchain. The user can select any item in the list to get a detailed description of that item. Fig.3 shows what the screen looks like after the phrase "Block Header" is clicked.

The list contains key words and phrases, as well as instructions of building blockchains in a step-by-step format.

C. The Set Up Menu

In a blockchain, there is a *difficulty* parameter that can be assigned a value which is needed in block mining. It is essentially an integer representing the number of prefixed zeros the hash of a mined block should have in order to qualify as a candidate to be added to the blockchain. Users should perform "Set Up" before moving to other steps of building the blockchain. Fig.4 shows the menu for setting up the *difficulty* parameter. In Fig.4 the integer number "3" is entered by the user.

The *difficulty* value is the only set-up step needed at the moment. Other parameters will be added when the application is enhanced to include network connections and server capabilities.

D. The Build Wallets Menu

The Build Wallets menu has three submenus: Add Wallets, View Wallets, and Clear Wallets, as shown in Fig.5.

Selecting the Add Wallets submenu allows users to create wallets, as indicated in Fig.6.

Wallets need to be created before transactions and blocks are created. A root wallet is created in the initialization code of the application to build a "central bank" which distributes funds to other wallets. Wallets need to have some intial funds before they can participate in transactions.

Fig. 7 shows the contents of three wallets after they are just created. In this case, Wallet 0 is the central bank with 100 coins. Wallets 1 and 2 are empty initially.

E. The Blockchain Menu

The Blockchain menu also has three submenus: Create Genesis Block, Add Block, and Display Block, as shown in Fig.8.

The user needs to create a genesis block first by clicking "Create Genesis Block". Then the user needs to click "Add Block" to add new blocks. On the "Add Block" window, the user should click "Create Block" to create a candidate block which has a timestamp, the hash of the previous block, a nonce value of zero, and a pre-mining hash for itself. Then the user can add as many transactions to the candidate block as possible by clicking the "Add Transactions to Block" button which also checks for transaction validity. The candidate block is added to the blockchain using the "Add Block to Chain" button. This button also starts the mining process, finds a suitable nonce value, and generates the post-mining hash which has the given number of zeros at the beginning of the hash. A screenshot for the Add Block submenu is depicted in Fig. 9.

F. *The Network Menu*

The Network menu is still under development. This menu allows the users to create nodes and set up networking and server parameters in order to form a distributed system and further simulate the operation of a real Blockchain system. The current implementation looks like Fig. 10. More submenus will need to be added in the future to support networking.

Overall, with ChainTutor, the users can performs different tests. For example, they can set a different *difficulty* value in Fig.4 to see how it affects the time used in block mining. They can change the values of the keys in Fig.6, or the hash values in Fig.9 to make the blockchain invalid so that it will not pass the validation. They can also use the "View Wallet" and "Display Chain" submenus to examine the contents of the blockchain and see how the blocks are joined together by their hashes.

V. CONCLUSION

This paper introduces a small Java application named ChainTutor that can be used in classroom teaching or self-learning of Blockchain concepts. The application allows users to experiment with the construction and operations of a blockchain using different parameters in order to gain better understanding of the basic concepts. The application gives users access to the internal details of a blockchain as well.

With the networking module added in the future, the app can be installed in nodes located in different computers to form a distributed system which models a real blockchain and provides the possibility of building an IoT system based on the blockchain to facilitate research on IoT systems based on blockchains.

ACKNOWLEDGMENT

The author would like to thank the Office of Provost of Kwantlen Polytechnic University for its support.

REFERENCES

[1] S. T. Aras and V. Kulkarni, "Blockchain and Its Applications – A Detailed Survey", International Journal of Computer Applications, vol.180, no.3, pp.29-35, December 2017.

[2] P. Andrew, "The Rise of Blockchain Courses at Top American Universities", https://coincentral.com/blockchains-at-university/. Accessed on August 4th, 2018.

[3] X. Liu, "Trends in Building Hardware and Software for Smart Things in Internet of Things", CYBER 2017 - The Second International Conference on Cyber-Technologies and Cyber-Systems, pp.65-69, November 12-16, 2017.

[4] Kass, Creating Your First Blockchain with Java, https://medium.com/programmers-blockchain. Accessed on August 4th, 2018.

[5] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview", Draft NISTIR 8202, NIST, January 2018.

[6] M. Conoscenti, A. Vetro, J. C. D. Martin, "Blockchain for the Internet of Things: a Systematic Literature Review", 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 29 Nov.-2 Dec. 2016.

[7] K. Christidis, and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things", IEEE Access, vol.4, pp.2292-2302, June 2016.

[8] W. Cai et al, Decentralized Applications: The Blockchain-Empowered Software System, IEEE Access, vol.8, pp.2169-3536, 2018

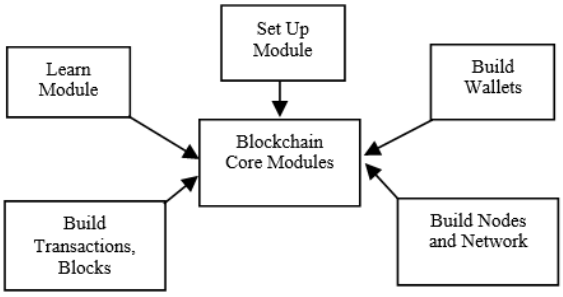

Fig.1 Modular structure of the Java application ChainTutor

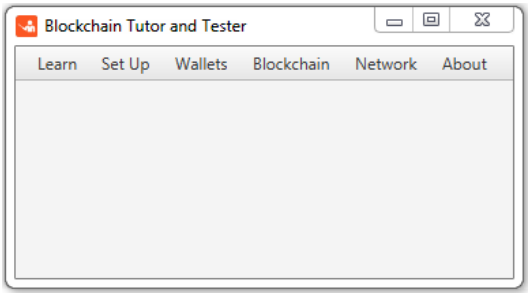

Fig.2 The main window of the Java application ChainTutor

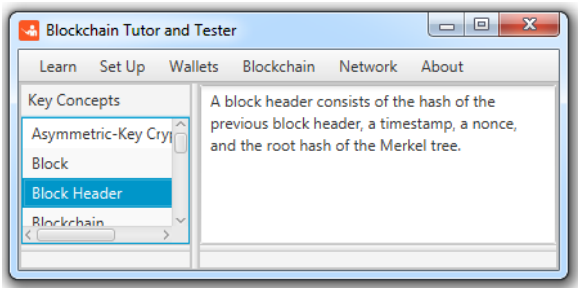

Fig.3 A screenshot after the Learn menu is clicked

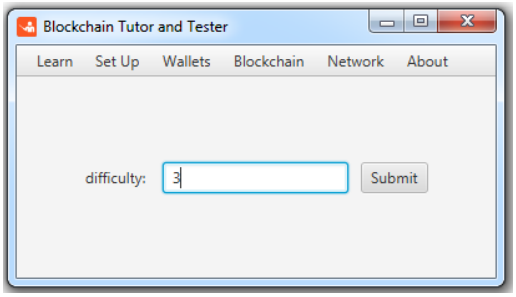

Fig.4 The Set Up window that receives a *difficulty* value of 3

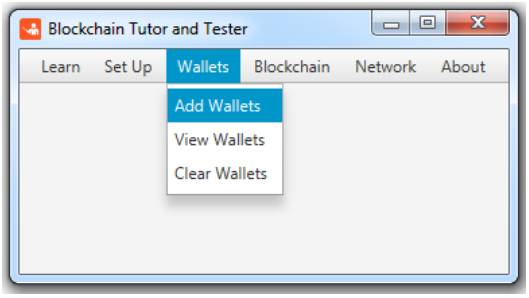

Fig. 5 The Wallets menu and its submenus

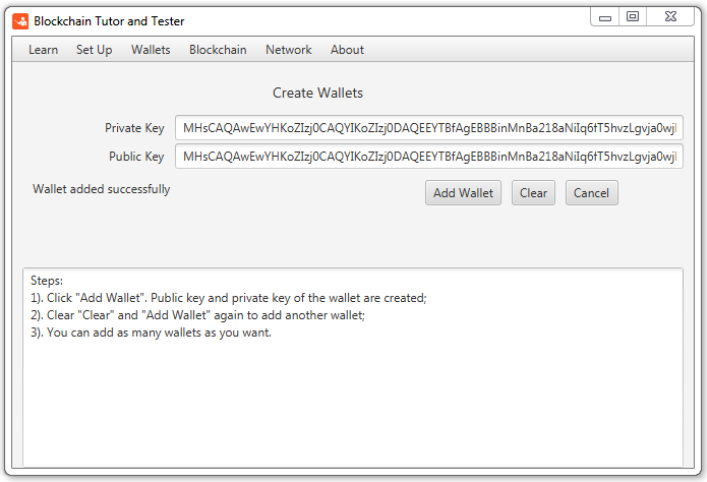

Fig.6 The Add Wallets window

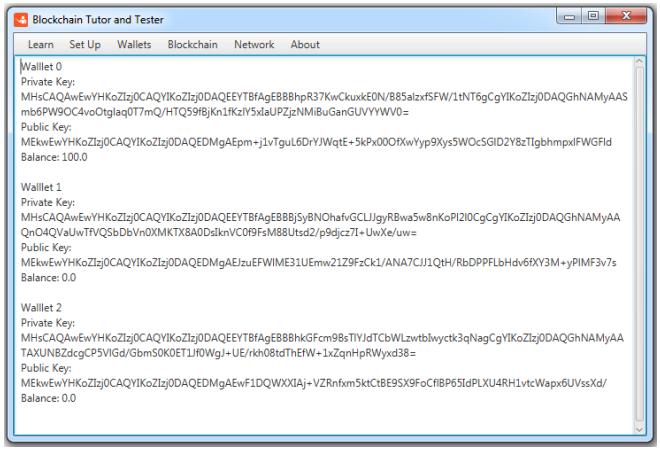

Fig.7 Content of three wallets

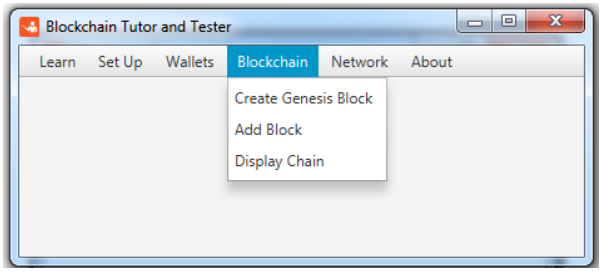

Fig.8 The Blockchain menu

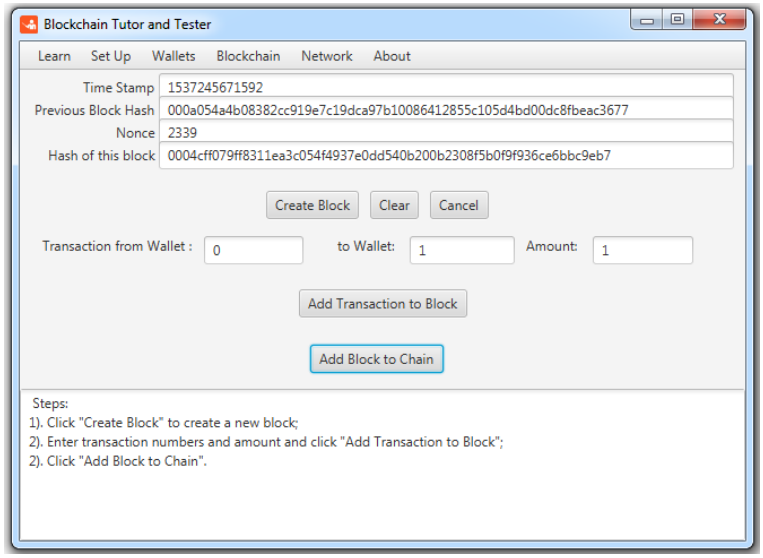

Fig.9 Adding a block to the blockchain

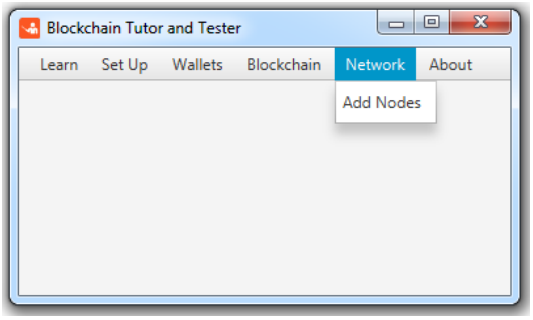

Fig.10 The Network menu pending further development