OHIO ELECTRONIC RECORDS COMMITTEE

Introduction to Blockchain Concepts

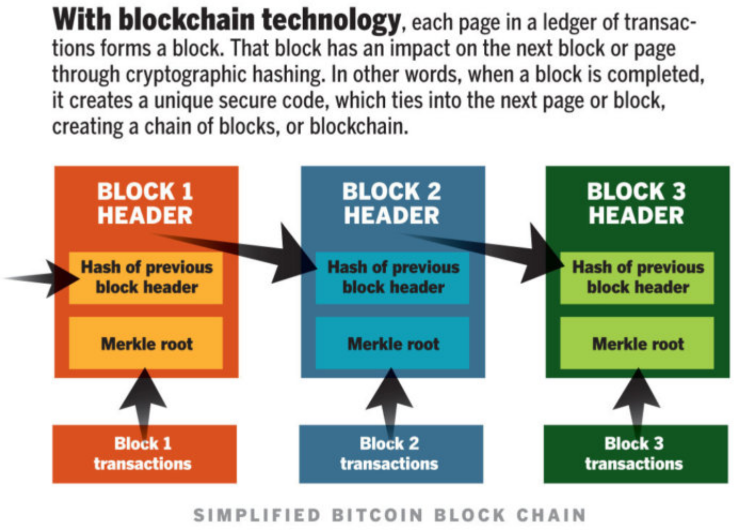
Blockchain is a growing technology with the potential to affect the records and information management profession. However, it is a technology that can be difficult to conceptualize. This high-level document is intended to define and illustrate the following common concepts associated with blockchain, or distributed ledger, technology:

- Ledger
- Distributed
- Immutability
- Encryption and Hashing
- Workflow
- Trustlessness
- Security

Ledger

Think of a paper-based ledger book found in an accountant's office, filled with blank lines. This traditional method of recording transactions requires an entry for each transaction. A full and complete record consists of *all* of the ledger entries in our imaginary book. We might record debits and credits on our ledger. A key concept is that if a mistake is found in the ledger, it is not crossed out or erased. Instead, a correcting entry is made in the ledger. The recording of the original mistake remains a part of the record, as does the correcting entry.

A first concept of blockchain is that each line in our ledger book is a transaction. Once a ledger book is filled, that is the equivalent of a "block." These blocks are then connected to form a chain. Each block contains two sections: transactions and header information. The block header contains a hash, or unique encrypted algorithm, both of all the hashes of all the ledger transactions of the block (called a Merkle root) and also of the previous block header; see Figure 1.



With blockchain technology, each page in a ledger of transactions forms a block. That block has an impact on the next block or page through cryptographic hashing. In other words, when a block is completed, it creates a unique secure code, which ties into the next page or block, creating a chain of blocks, or blockchain.

Figure 1: Pursel, Bart. (2018, January 28). Blockchain is Here to Stay. Retrieved from http://sites.psu.edu/ist110pursel/2018/01/28/blockchain-is-here-to-stay/

Distributed

If it were possible to ensure an entry could not be added to the ledger page without simultaneously adding the same line to several other books, the ledger could become "distributed," which means that copies of the ledger would exist on multiple entities (computers in blockchain). Why is this useful? It provides the means to check the validity of any entry, or block. By making copies of the ledger, it is difficult to tamper with any single entry, while maximizing the means by which to authenticate the entry.

Immutability

A key part of the architecture of blockchain is that one only *adds* blocks to the chain; blocks are not removed or changed. Once an entry has been added to a ledger, it is permanent, or immutable. An entry can be added with the instruction to "ignore that last entry" or inform users that "the new correct total number of widgets is…", but one can never go back and change or "mute" the entry. It is there permanently.

The following is an example of immutability:

> An author writes an important note on a piece of paper, and then gives a copy of it to 100 different people whom the author does not know, cannot find. The author then writes new notes, and distributes it the same way, repeatedly. Now assume the author decided to change the facts about that original note, or any one of subsequent notes, they will have to reach all 100 holders of the data and convince them to change the data. Since all subsequent notes reference the one before it in the chain, they will have to change not only the note affected by the data change, but every subsequent note, so the references all match. When the changes are complete (if they are complete), all 100 recipients check each other's sets of notes. If at least 51 of the note sets agree, the change would be effected; conversely, if 51 sets still have the original entries on them, then the 49 that the author managed to change would be abandoned, and the majority consensus would remain as the record.

In practice, the ledger entry is extremely hard to change. There is usually a short window, often just a few minutes, to change all of the entries on at least 51% of the copies before the whole ledger revalidates itself and undoes all attempts to change the record of the transaction.

New entries within a block are referred to as "transactions." The transaction only contains the data that has changed; it does not include the final state. For instance, if an account started at 100 and is increased by 10, the transaction does not state the account balance is now 110, instead, the transaction only states the account has been increased by 10. In order to determine the balance of the account, every transaction ever performed on that account is replayed. However, blockchains may cache the current balance so they do not have to replay each transaction every time they need to know that information. All new computers joining the network must first go through the entire chain's history at least once in order to be synchronized with the network.

Encryption and hashing

This is of course a simplified explanation but it should provide a basic understanding of encryption and hashing.

Encryption basically means to take information and hide it or mask it. Hashing is the process of "hiding" or "masking," which is accomplished by sending the original information through an algorithm, which is a series of calculations performed on the original information.

For a simple example, our information will be just one word: “cat”. We want to encrypt “cat” so that when others see it they do not know that “cat” is in fact the original word. A very simple hashing algorithm example is to replace each letter in the word with the letter just after it in the alphabet, with “z” being replaced by “a”. If we put “cat” through that algorithm, the result is “dbu”.

Of course, this simple algorithm is not complex enough to prevent someone from guessing the original word. But you can see how the process works; at first glance, you do not know that “cat” is our original information; all you see is “dbu”. If you were my compatriot, and therefore in the know about the algorithm, you could easily reconstruct “cat” from “dbu”.

True hashing algorithms are of course much more complicated, so that it is extremely difficult to reverse engineer the hashing algorithm and get the original information. Whether it is impossible to do so is a matter of debate and depends on the quality of the hashing algorithm and the computing power available to the hacker trying to break the code.

Workflow

The identities of the ledger entry are condensed into an irreversible fixed length value called a “hash” that can look something like this: “0xea34ad07d99a74fbe169e3eba035e633b65d55”. In order to be linked into the chain, each new block references the block previous to it by referencing this hash. The hash is an intrinsic property of the data; therefore, changing the data–even minutely–drastically changes the hash.

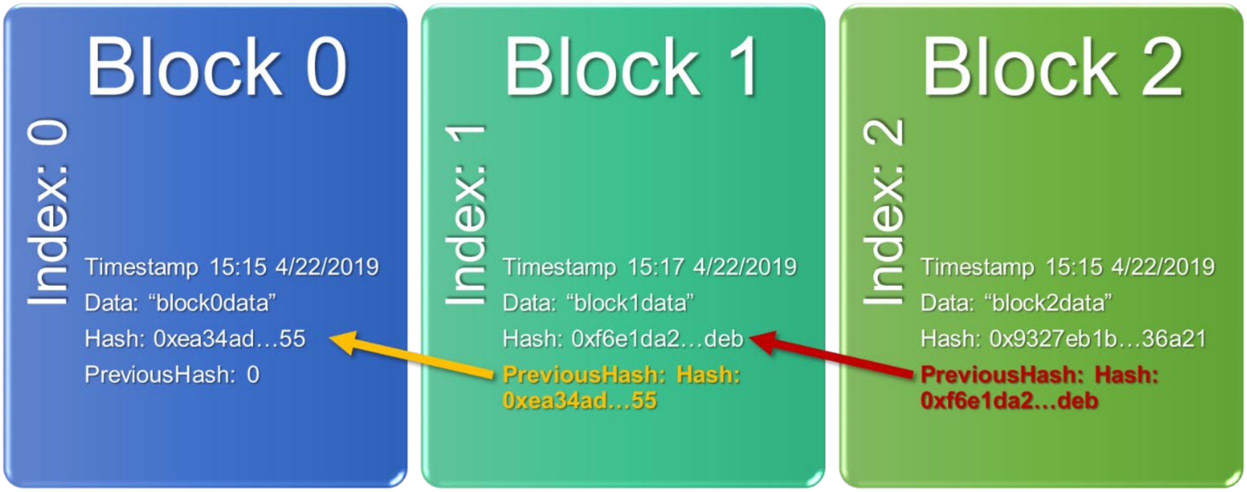


Figure 2: Hash value referencing from block to block

Consider Figure 2 above. The hash of Block 0 is included in Block 1. When calculating the hash of Block 1, the hash of Block 0 is also included in this calculation. If a change to the parent of Block 1 (i.e. Block 0) is made, the contents of Block 1 are consequently also changed with the inclusion of a new Block 0 hash value. These changes mean the hash for Block 1 must be recalculated, delivering a new value. Since Block 2 contains the original Block 1 hash as its parent, a break in the chain would occur and any computer validating the chain would quickly find the discrepancy. Therefore, the further back in the chain a change is attempted, the more impractical modifications become.

Most of this architecture will be invisible to users.

Trustlessness

Trustlessness is a key concept behind blockchain that is defined as the ability to generate transactions without having to trust anyone or anything else in the process. Parties do not need to know each other. The creation and authenticity of ledger entries are provable facts, not requiring anyone's claim. This architecture is useful in business scenarios in which the parties may not trust each other without an intermediary. While trade transactions have been identified as a use case for conducting transactions without trust, there are many scenarios where we currently use a person, their credibility, or the authority of their office, to introduce trust. Blockchain could be used in some of these scenarios to record facts in an automated, reliable way without the need for trust agents.

Audits are an emerging use case for blockchains. The basis behind an audit is the need to prove historical facts to a third party that has reason to doubt the historical claim. With a blockchain, the auditing process becomes significantly more robust and efficient to perform. While a blockchain may not prevent a person or machine from lying at the time the ledger was written, it does guarantee that someone has not changed or tampered with the ledger after the fact.

Secure

In scenarios in which data, facts, transactions, or records need to be highly available while simultaneously highly secure, a distributed ledger presents significant theoretical advantages over current data repository alternatives. While the basic architecture of blockchain is robust, complex blockchain systems may have inadvertent architectural flaws that render them vulnerable to hacking. Currently, cryptocurrency exchanges, software applications, and systems using blockchains have had many instances of being compromised or hacked. Any consideration of blockchain implementation should consider the specific security concerns relevant to their needs.