Back-Step Blockchain Security

Georg Krause, Blockchain Systems Pte Ltd, Singapore

December 30, 2018

Abstract

The Bitcoin systems has many distributed copies of their blockchain. The reason, why the number increased was primarily the change to earn mining rewards and not contributing to security. To make a blockchain secure, it must be huge in nodes? This may be true for some digital currency implementations, but definitely not for the majority of the blockchain applications. Huge size has quite a few disadvantages, like network traffic. "Proof-of-work" based chains have the huge disadvantage of energy hunger. The question was, can these negative properties be eliminated by a different architecture of the undelying chain? I found a promising solution that does not require proof-of-work and huge networks. The Back-Step blockchain replaces proof-of-work with one-way values running from the future to the past. This idea allows secure chains where manipulations can be easily detected and disputes can be fast resolved. The article shows how this is done and explains why the involved Hardware provides a much higher security level at dramatically reduced cost.

1 Current Blockchain Security

In a traditional Blockchain are many nodes required. Take the Bitcoin implementation, where one miner finds a hash and the new block is replicated to all other nodes, or at least to those on-line. Then all active nodes have the same history of blocks. At Bitcoin all miners (many thousand) compete for finding the next hash, which matches the "proof-of-work" criteria. After one miner succeeds, the data are replicated to all peers having a bitcoin blockchain copy.

Proof of work is the mechanism that prevents from modifying an earlier block and recalculating all following blocks until the current one. Such a modified chain would need to be replicated to the majority of the blockchain computers to be accepted as the correct chain.

Proof of work is an expensive solution, for the users and for the environment. An average of 1000 transactions are contained in one block. Hashing the header of such a block takes 10 minutes, if thousands of miners are working together. Current calculation shows that a single transaction takes 57kW/h at 25 cent,

which is 14 Dollar for electricity plus a transaction fee. Transaction fees are paid by the users directly, the electricity is paid indirect. The electricity fee reduces the value of the Bitcoins, so every Bitcoin holder pays it, independent if he actively creates transactions or not. This is the price we pay for the security.

2 Back-Step Blockchain Security

In this article we use the term BSB for Back-Step Blockchain. We assume that the reader has studied the basic mechanisms and advantages of the BSB already. BSB is different to traditional chains. As we eliminate the proof-of-work we have to proof that we can achieve the same or a higher amount of security. For redundancy reason a few nodes may be required to provide operational safety. That makes sense and should be implemented for every mission critical application. These passive copy nodes are not able to create further blocks, because they have no access to the Back-Step generator. Replicated chains can be used to verify the consistency and authenticity of the chain. They may also be used to provide access in case of network disruption. Replication by itself does not create security.

Lets consider different attack scenarios. First lets assume assume an attacker attempts to change a past payload. He can access a replicated chain, copy it, modify the record he wants and recalculate the rest of the chain easily, because no proof of work will slow him down. He can calculate up to the current record. But here it ends, the faked chain cannot calculate the next block, because it has no access to the BSB generator. As soon a chain is detected with different hash values one must be the faked. The one who **cannot** calculate future blocks is the faked one. A network application like a Crypto currency can request from both chains to generate a next block. Only the one who can generate the block is the true chain. The other(s) can be eliminated. As seen, this does not require a huge amount of nodes and no majority consensus mechanism is required.

Actually the security is dependent on the **ownership** of the Back-Step generator. As this component is a highly secured HW device with hierarchical separation to the target system, it cannot be attacked by software. Each HW device supports one block chain. The delegation of the security functions to the hardware creates high security, which is most desired. When we consider the finite lifetime of every physical device, the takeover process to a follow up chain becomes important. There are two situations. First the planed end of life of a hardware product. Second a surprise failure of a BSB or the system it is attached to. These two scenarios are explained in more detail below.

To stay undetected the attacker would need to copy the manipulated chain to all distributed backup machines. In addition the attacker needs to exchange the chain on the machine containing the blockchain Hardware. Such a replacement will be detected by the BSB hardware. As soon as the manipulated chain

tries to process the next block, the Hardware detects that the value for the previous hash provided does not match the internally stored last recent hash and the operation stops. The manipulated chain cannot continue. For that, the hardware of the Back-Step generator keeps a copy of the most recent calculated block hash. The strict separation of the Back-Step hardware from the software of the blockchain provides a high level of security and pays off here again. A software cannot change the hardware. Any theoretical attack requires to change the Hardware of the Back-Step Generator while it is running. This is not practicable. This is even more unrealistic when the chain is spread among multiple systems, like in a public coin application.

3 Security by Seal

In addition each block contains a **seal** value, which is calculated by the Back-Step generator internally. This value cannot be verified with the chain data alone. An additional value called back-step seal value is required. This value can be requested from the BSB. The back-step seal value is delivered for past blocks, but not for the current block. This value allows to detect chain manipulation by then owner of the Back-Step generator.

The construction of the seal combines the previous hash and a backwards running one-way value. The exact calculation is as follows:

Let be
s the seal stored in the block
n the number of the current block
h the hash value of block
b the back-step seal value of the current block
i an intermediate value

$$i = f(xor, h^{n-1}, b^n)$$
$$s = f(owf, i)$$

The figure 1 shows first the seal value generation in an unmodified system. The value (1) represents the previous hash. After two operations, XOR and hash, the seal value is available that is in the output of the block. During generation of the block the back-step seal value is only available inside the generator. This means without knowing it the seal cannot be verified. This is an intentional design decision. Now lets see which values are changed when an attack manipulates a previous payload or the meta-data. This is illustrated in figure 2.

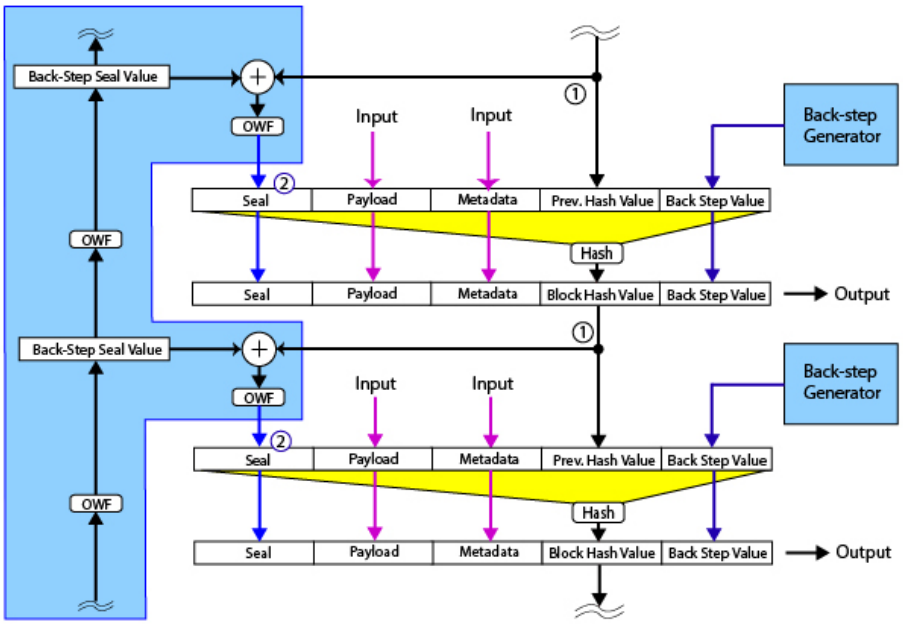

Figure 1: BSB without manipulation

How does the protection work?

A faked chain has the problem that it cannot calculate the correct seal. Because the seal is dependent on the previous hash, the seal is broken when the chain is manipulated or replaced. Outside of the hardware the back step seal value cannot be calculated for any block. When a back-step seal value of a previous block is released from the hardware the verification can be performed for all previous blocks in the chain. Any manipulation will be immediately detected. It is a decision of the application designer if he requests a previous back-step seal value whenever a new block is created, or only when a reason for verification exists. The value of the most recent generated block is never released.

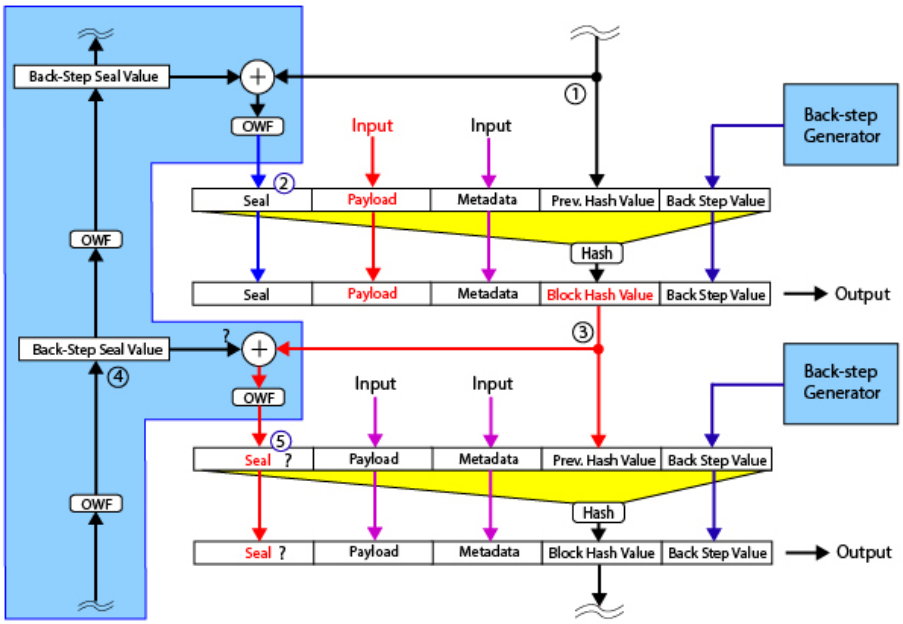

Figure 2: Effect of payload manipulation

In figure 2 we assume that the red input shows a value after a manipulation. For that block the previous hash (1) reflects the unchanged situation up to the last block before the manipulation. The seal value always contains information about the previous block, never about the current one. After recalculating the hash of the upper block the hash value (3) has changed. The attack ends here, because the attacker has no way to calculate a new seal (5). He misses the back-step seal value (4).

A separate article describes how multiple chain, each with its own generator, can run in parallel for high availability solutions like in banking, on-line payment systems or air traffic control systems.

End of document.