IoT Software Upgrade Scheme Based on Block Chain

Jian DU, Ping ZHEN, Yin-zi TU and Jie GAN

Beijing Smartchip Microelectronics Technology Company Limited
Beijing Engineering Research Center of High-reliability IC with Power Industrial Grade
Haidian District, Beijing, 100192, China

Keywords: Software upgrade, Block chain, Decentralization, IoT.

Abstract. Software upgrade is necessary, which can not only get the new software features, but also fix software vulnerabilities to improve security. With the rapid development of Internet of things (IoT), all kinds of smart devices increase sharply, so traditional C/S (client/server)-based software upgrade model is difficult to satisfy the application scenario. In order to guarantee massive terminal software upgrade in secure and efficient way, this paper proposes a novel software upgrade scheme for IoT terminals based on the block chain technology. In the scheme, there exists a variety of terminal devices in the IoT scenario, and each terminal device has a variety of softwares, and each software upgrade record is stored in the block chain by consensus mechanism. Terminal device can query software update record from block chain and download the latest software version from adjacent terminal device in IoT, which can complete software upgrade spontaneously, with the advantage of decentralization. At the same time, the scheme can ensure the timeliness and traceability of software upgrade. This scheme takes advantage of the untamperability of block chain data to guarantee its security and reliability, which provides a new solution for terminal device software upgrade in IoT scenario.

Introduction

In 2017, Wanna Cry ransomware suddenly broke out, and at least three hundred thousand users in 150 countries were been affected, which caused about $8 billion loss, which caused serious harm in finance, energy, medical treatment and many other industries. The main reason of the disaster is that the user failed to upgrade the operating system software in time and then the hacker made use of vulnerability to launch the attack. Software upgrade is necessary, which can not only get new feature to promote the user experience, but also prevent software vulnerabilities. With the development of the Internet of things (IoT) and 5G technology, the Internet of everything era is coming. It predicts that 25 billion terminal devices will be connected in IoT by 2020, and the number will continue to grow. Large-scale IoT not only brings convenience to the persons in life, but also puts forward the higher security requirements at the same time, in which how to realize software upgrade among the mass terminal devices is a great challenge.

Software upgrade not only needs to implement the examination and verification of software version, but also should guarantee the integrity of the downloaded software data. Existing software upgrade scheme mainly adopts C/S(Client/Server) model, as shown in figure 1 (a). The client side downloads the latest software version from the server side, which is likely to cause network congestion and security risk in belated software upgrade. The centralized manner of software upgrade also has the disadvantage of single point of failure, namely when the software upgrade server breaks down, the whole system will fall into malfunction. What’s more, C/S upgrade manner cannot prevent the temper of software patch data, which cause the low reliability and security.

Block chain technology originated in 2008, proposed by Satoshi Nakamoto in his foundational paper “Bitcoin: a peer-to-peer electronic cash system” [1]. Block chain is the core technology to build Bitcoin network and encrypted transaction information, with the pivotal advantage of decentralization. It can implement the point-to-point decentralized credit transactions in distributed system without mutual trust in the nodes, which can provide the solution to remedy the deficiencies in
centralized organizations, such as high cost, low efficiency and insecurity, etc. Researchers has proposed many good decentralized applications for IoT scenarios by block chain [2], such as decentralized data privacy protection scheme [3], the cloud data integrity audit scheme [4], e-commerce transaction model [5], identity management model [6], etc.

This paper proposes a decentralized software upgrade scheme, as shown in figure 1 (b), which can implement software update in terminal device from peer node. In the scenario, every software upgrade record is recorded in block chain and other terminal nodes can query the software upgrade record from block chain and select adjacent nodes to get the latest software version and complete automatic software update, which can guarantee the reliability of software source and the traceability of historical versions. The integrity of software source is guaranteed by the untamperability of block chain data, which can effectively prevent the spread of malicious software. This scheme can complete software upgrades spontaneously in decentralized form, and has the advantages of security and high efficiency to meet the demand of mass software upgrade in terminal devices, so it has extensive application prospect in IoT.

This paper is organized as follows: section 2 introduces the basic concept of block chain technology; Section 3 introduces the proposed software upgrade scheme based on block chain; The security and performance analysis are discussed in section 4 and section 5, respectively. The full text is summarized in the last section.

**Block Chain**

The block chain is a distributed ledger. It is based on the principle of cryptography instead of credit to make any direct payment between both parties, which doesn’t require the participation of the third intermediary. Block chain has solved two important problems in digital currency: double pay and Byzantine failures. Block chain has the characteristics of decentralization, time-series data, collective maintenance, programmability, security and so on. The structure of the block chain is shown in Fig 1.

**Figure 1.** (a) Traditional C/S software upgrade model; (b) Decentralized software upgrade model.

**Figure 2.** The structure of block chain.

Fig 2 is the simplified structure of block chain. Each block includes one or more transactions, which are used to compute the Merkle root by Hash operation in the way of Merkle tree. Merkle root is stored in the block header. Each block header also stores the Hash value of the previous block, thus
forming a chain structure. This means that if the transaction information on a block is modified, it must modify all data of the rear blocks. Transaction is recorded on the block by consensus mechanism, which is realized in the way of PoW (Proof of Work).

The transaction in the block chain is composed of one or multiple inputs and outputs, which also form the chain structure. The Bitcoin is transferred from one transaction to another, namely the input of the current transaction is the output of previous one. Each transaction can have multiple outputs, but each output can only be treated as one input, to prevent double payment. The transaction structure of block chain is as shown in Fig 3, in which the input of transaction 0 is 100 k (satoshi is the unit of Bitcoin, 1 BTC = 10^8 satoshis). It has two outputs, where output 0 is the input of transaction 1 with the transferred amount of 40k, and output 1 is the input of transaction 2 with the transferred amount of 50k. The remaining 10k is the transaction fee, used as the income of miners.

Figure 3. The transaction structure of block chain.

Block chain technology is the focus of current research, with the great application potential in areas such as securities trading, e-commerce, IoT, etc. This project intends to adopt block chain to implement decentralized software upgrade scheme in terminal devices. Software upgrade is implemented by the mutual authentication of peer-to-peer nodes, which can be used to replace the traditional centralized software upgrade model.

**Software Upgrade Scheme Based on Block Chain**

There exists many kinds of devices and each device has installed many kinds of softwares. Each software upgrade process is recorded in the block chain. The basic model of this scheme is shown in Fig 4. Each block contains multiple software upgrade records, organized by Merkle tree. They are stored on the block chain through consensus mechanism, to ensure the reliability of the data record.

Figure 4. Software upgrade model based on block chain.
Software update record details are shown in table 1. The first row includes identification of terminal device and software, to show the software upgrade record source. Each upgrade record only has one input and one output. The input is index, which indicates the source of new version software. Then the software source and transmission process can be traced by index link. The output includes software version number, size, Hash checksum, timestamp information. In order to prevent the spread of malicious software, software version and Hash checksum information must be consistent with records linked by index, otherwise it cannot pass the validation of consensus mechanism, so the first index for empty software update record is crucial and must be authorized to release by the administrator of the terminal device. Finally, it is the digital signature to prevent counterfeit software upgrade record.

<table>
<thead>
<tr>
<th>Device and Software Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Output</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The implementation process of software upgrade scheme is shown in Fig 5, in which different graph represents different device. A, B and C are the same terminal device. The detailed process of software upgrade is as follows.

1. If the device A has completed the software upgrade, the record is stored on the block chain at position \((m, i)\), where \(m\) denotes the block number and \(i\) denotes the \(i\)-th record on the block \(m\).
(2) The terminal device B query the latest software version periodically, when it finds that A has the latest software version at position \((m, i)\) on the block chain, then B realizes it needs the software upgrade.

(3) Then terminal device B sends the software upgrade request to A.

(4) Having received the software upgrade request from B, A sends the latest software data to B, as shown in Fig 5. The software data is transmitted through node C.

(5) After receiving the software data from A, B will check the data integrity by comparing the record checksum on block chain at position \((m, i)\). If the verification succeeds, B will complete the software upgrade and broadcast the upgrade record to the block chain, as shown in Fig 5, at the position \((n, j)\). Then its index points to the position \((m, i)\).

Other devices in the network can implement the upgrade process in the similar way, until all of the terminal device complete the software update. In this way, the traceability of software upgrade version can be achieved.

**Security Analysis**

**Software Version Management**

Due to the chain structure of software update record on the block chain, the software version of the whole chain structure must be consistent, otherwise it cannot pass the verification. In order to prevent spread of malicious software, we must ensure the correctness of the first software upgrade record. So the first software upgrade record must be authorized by the software issuer and its index is empty. Ordinary device node is not allowed to release software update record with empty index, namely the software upgrade record on block chain can be traced back to an empty index eventually. In this way, the software upgrade version management can be realized. Ordinary device node cannot release the latest version of the software for the first time, so it can prevent the spread of malicious software.

**Consensus Mechanism**

In order to prevent the malicious tampering of software upgrade packet in the spread process, the size of software upgrade packet and Hash checksum value are stored on every software update record by the verification of consensus mechanism. The terminal devices are acted as the miners in this consensus mechanism. Similar with PoW process of the Bitcoin, the implementation of the consensus mechanism is shown as follows:

1. The miner collects all the unconfirmed software upgrade records in the current network, and puts all the legal records into the current block.

2. The miner calculates the Merkle root value of all software upgrade record and keeps it in the block header. Then fill in the other data field, in which nonce is set to zero;

3. The miner increases the nonce by plus one, and calculates Hash value of the block head. If it is less than or equal to the target Hash value, the nonce is successfully found to generate the current block; Otherwise the miner will continue the exhaustive search by increasing the nonce until certain node has obtained the appropriate nonce.

4. If the nonce search is not successful in a valid period, the miner will upgrade the timestamp and restart the collection of unconfirmed software update record. Then the miner recalculates Merkle root value to continue the search.

It takes about ten minutes for Bitcoin to complete the transaction confirmation. For distributed software upgrade scheme purposed in this paper, we can adjust the search difficulty of the nonce according to actual application demand and confirm the record on the block chain in a flexible way.

The security of block chain depends on the hash function. It is estimated that \(2^{48}\) years will be spent to produce one SHA256 hash collision according to the computing power of Chinese supercomputer Tianhe-2 [7], which fully shows the security of the chain block. With the development of the future quantum computer, new threats are proposed to the existing cryptography algorithm. So
quantum-resistant cryptography algorithm should be used, such as lattice-based cryptography, which is an alternative to traditional algorithm. This does not affect the practicability of the scheme. The challenge of PoW consensus mechanism is the 51% computing power attack. Namely if the attacker control 51% computing resources of the whole block chain network, then it has the ability to forge the record data on the chain. Researchers have put forward a solution [7] by relying on PoW consensus algorithm with high computing power and large memory and further research is needed to design more secure and effective consensus mechanism.

**Record Signature**

In order to ensure the effectiveness of each software update record, digital signature is needed for each record, then Identity-based cryptosystem (IBC) can be used [8]. The identity information of the device can be used as public key to sign the transaction record. In this case, every device needs a unique identification. The complex digital certificate management problem of traditional PKI system can be avoided by IBC signature. In 2013, international IBC standard IEEE Std1363.3 was released [9]. China has also launched the standardization work of IBC. The State Cryptography Office published SM9 standard in March 2016, which can be also used to sign software upgrade record.

**Performance Analysis**

Compared with the existing C/S software upgrade manner, the software upgrade scheme based on block chain is more effective, which can automatically complete software upgrade in a distributed way. The scheme can complete software upgrade target quickly and avoid the single point failure problem effectively. At the same time, it can prevent tampering problem of the software upgrade packet with high reliability. According to the network topology, the device can select a close node as the source of software upgrade, to prevent network congestion.

In the block chain network, each node needs to save a copy of all the software upgrade record data. With the huge number of users access to the network, it inevitably lead to the expansion of huge amounts of data. To solve the problem, it only needs the key nodes to save all the data backup and other nodes save needed data. Or on the premise of guarantee security, those old software upgrade version data can be abandoned to reduce the block chain data capacity.

Smart contract is defined in block chain [10] and it encapsulates the predefined number of state and transformation rules, triggering the contract execution scenario, such as a specific time or specific events and the response of certain situations. The real-time state of smart contracts can be monitored. Through the verification of external data sources, trigger condition can be met to activate and execute the contract. Through the trigger condition of smart contract, the scheme can determine the device software upgrade time and source flexibly, to customize suitable software upgrade manner for practical application requirements.

**Conclusions**

In order to solve the software upgrade problem of mass devices in IoT, this paper proposes a software upgrade scheme based on block chain technology, which can complete software upgrade in decentralized way spontaneously. The scheme can replace the traditional C/S architecture model of software upgrade. In this scenario, each software upgrade record is kept in block chain through consensus mechanism. Other devices can query block chain and select adjacent node to obtain the latest software version. Through the transmission in a distributed way, it can timely complete the device software upgrade in the network. The untamperability of block chain data can guarantee the actual reliability of the software update records and Hash checksum is used to ensure the integrity of the software upgrade data, preventing the spread of malicious software. This paper only puts forward the basic framework of the software upgrade scheme based on block chain technology and the specific details about the consensus mechanism, smart contract and block chain query technology to meet the demand of IoT scenario application still need further research.
Acknowledgement

The research is supported by technology project of State grid Corporation: Key Techniques Research of Resistance to Physical Attacks in Chip Circuit Level and Samples Development” (No. 546816170 002).

References


