# Models and Related Technology

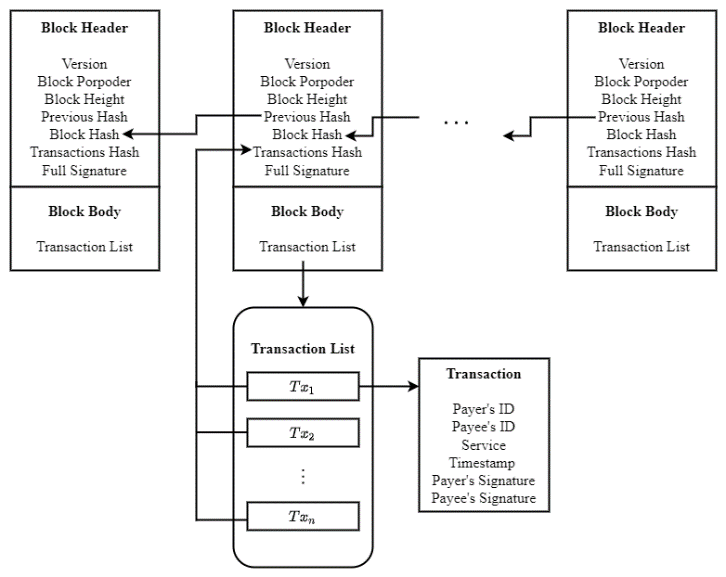
We describe the preliminaries of SWIB, including network model, blockchain setting, communication model, and attack models. (需要在润色一下)Network model is the fundamental infrastructure running the blockchain consensus protocol. Blockchain is the data architecture of distributed ledger used in SWIB. We leverage threshold BLS signature scheme to improve the performance of blockchain consensus protocol. Besides, communication model is the precondition of performance analysis in consensus protocol. In addition, we analyze the attack-resistance of SWIB under attack models.

## 3.1 Network Model

Blockchain consensus protocols are designed according to network models. Consensus nodes achieve consensus on blocks through networks transmitting messages. (通篇都是考虑无线网络场景)We only consider a wireless broadcast network consisting of nodes, which are located within communication range of each other, and communicating with each other by transmitting messages. All nodes have same functions. Each node equipped with transceiver works in a half-duplex manner. This means that nodes can transmit or receive messages, but not both simultaneously. In practice, such a network can be formed by a group of unmanned aerial vehicles or intelligent vehicles.

Nodes adopting digital signature technology can achieve node identity confirmation and the verification and integrity of communication messages. In digital signature, each node has its key pair, which used to message encryption and decryption, and generate signature. We assume that each node can get its private-public key pair and a main public key by independently running a secure distributed key generation protocol. Each node can obtain the public keys and identities of other nodes by exchanging messages. Thus, each node knows the identities and public keys of all other nodes.

## 3.2 Blockchain

Blockchain is emerged as a core technology of Bitcoin, which is known as a decentralized digital cryptocurrency and appears in 2008. Blockchain offers many benefits, such as decentralization, security, transparency, data integrity, and so on. Therefore, blockchain technology is being explored in many innovative applications, such as crypto currencies, smart contracts, Internet of Things, etc.(放在Introduction中)Blockchain is a public electronic ledger, which similar to a relational database. This ledger is openly and simultaneously shared among various nodes in system. Blockchain uses a consensus protocol to ensure the dependability and integrity of blockchain system. Therefore, lockchain can only be updated when consensus is reached between nodes in the system. Each node maintains a local blockchain, which is composed of blocks chronologically linked to create a chain. Each block contains a block header and a block body. The block header records blockchain version, block proposer, block height, previous hash, block hash, block full signature, and transactions hash root, etc. The block body stores transaction recodes, which consist of payer's information, payee's information and other necessary contents. Figure 1 shows the data structure of a blockchain.

## **Fig. 1 Blockchain**3.3 Threshold BLS Signature Scheme Technology

Threshold Boneh-Lynn-Shacham (BLS) signature scheme [29] can be used to improve the performance of consensus protocols. Consensus protocols can adopt the threshold BLS signature scheme as a voting mechanism to confirm the validity of proposals. A partial signature generated by a node for a proposal is the vote of the node for the proposal. Since several partial signatures can be aggregated into a special signature, nodes can verify multiple votes in an operation. In addition, the termination of a consensus process not relies on the leader of consensus system. Any consensus node can terminate a consensus process when it has constructed the special signature. Therefore, threshold BLS signature scheme can help consensus nodes quickly and steadily achieve consensus.

Threshold signature scheme [34] aims to allow multiple participants securely reconstruct a secret (i.e., a main private key) and perform computation (i.e., signature generation or decryption) even an adversary has corrupted some participants. In a threshold signature scheme, all participants use a share of a main private key as their private keys. To sign a message, multiple participants can execute an interactive signature generation algorithm, which uses their shares of the main private key and the message as inputs, and outputs a signature of the message. The signature can be verified by anyone using the unique main public key. The security for threshold signature scheme requires that no adversary that corrupts some participants can learn any information about the main private key or can forge a valid signature on a message. There are two important properties of threshold signature scheme: robustness and proactiveness. The former one requires that a valid signature can be generated even some malicious participants deviating from the scheme. The later one is also name periodic refreshment of shares of a main private key. The goal proactiveness is to protect a system from an adversary that obtains the information of a main private key.

Boneh-Lynn-Shacham (BLS) signature scheme [30] is closely related to bilinear maps and Gap Diffie-Hellman (GDH) groups. The signature scheme is based on bilinear maps on elliptic curves. In GDH groups, the Computational Diffie-Hellman problem is hard but the Decision Diffie-Hellman problem is easy. The BLS signature scheme based on GDH groups consists of key generation algorithm, signature generation algorithm and signature verification algorithm. In key generation, each participant can random pick a private key, which can be used to compute the public key. Given a private key and a message, a signer can compute a signature through signature generation algorithm. Given a public key, a message and a signature, verifiers can verify whether the signature is a valid by signature verification algorithm.

Threshold BLS signature scheme [29] is the combination of threshold signature scheme and BLS signature scheme. The signature share generation of threshold BLS signature scheme concurs with the BLS, and the aggregation of group signature is congruent with threshold signature scheme. Threshold BLS signature scheme includes key generation algorithm, signature generation algorithm and verification algorithm. The key generation algorithm adopts a discrete log-based distributed key generation method [31] to distribute private-public key pair and a main public key to participants. The key generation algorithm outputs a private-public key pair for each participant and a main public key. The private key of a participant is a share of a main private key. In threshold BLS signature scheme, participants jointly sign a message through the signature generation algorithm. This algorithm contains a partial signature generation method and a full signature recovery method. The former one is similar to the signature generation algorithm of BLS signature scheme. Given a message, a signer can compute a partial signature by running signature share generation method. The full signature of the message can be recovered when participants aggregate a sufficient number of partial signature and execute the full signature recovery method. The full signature should be the Lagrange interpolation polynomial of these partial signatures. All participants can use the main public key to verify the validation of the full signature by the verification algorithm of threshold BLS signature scheme. We assume that the number of honest nodes satisfies the requirement of threshold BLS signature scheme, which ensures the security of SWIB.

## 3.4 Communication Model

Consensus latency is an important metric, which used to evaluate the performance of blockchain consensus protocol. Consensus latency contains computing time and communication time. However, the computation load of nodes in SWIB is small, which means the computing time is negligibly small. Therefore, we mainly focus on the impact of wireless communication in the performance of SWIB.

In SWIB, consensus can be reached among nodes through wireless communication. During consensus process, nodes disseminate messages over wireless channel. We consider a wireless communication model with p-persistent carrier-sense multiple access (CSMA). Nodes continuously sense channel and transmit message with probability when detecting channel idle. We assume that wireless channels follow the Rayleigh fading model [32]. The channel gain between nodes follows the complex normal distribution. According to the Rayleigh fading model, the received signal-to-noise (SNR) ratio over the channel between nodes can be computed. SNR ratio is an important metric for communication quality under wireless networks.

During consensus process, messages losses can lead to consensus failure. In addition, message losses are mainly caused by channel collision and channel fading. Therefore, a successful transmission should satisfy two conditions: 1) if and only if there only one node transmitting in a time slot; and 2) the SNR ratio is equal to or bigger than the target one. In channel contention process, nodes compete for the channel with a constant transmit probability. Only if one node transmits in a mini time interval can the node transmit message successfully. Even a node competes successfully, it may fail to transmit a message due to channel fading. The SNR ratio over channel between nodes varies with the time-variant communication environment. When SNR ratio is less than a given target threshold, the communication between nodes is interrupted.

Both channel collision and channel fading can result in message losses. In order guarantee communication under unreliable networks, retransmission mechanism is necessary. When communication interruption occurs, retransmissions are carried out until the lost messages are successfully delivered.

## 3.5 Attack Models

Adversary can launch attacks to rig the consensus process to benefit itself or halt the process.(将主语改成Attacks)

The security of blockchain system is important for designing consensus protocols. We assume that an arbitrary adversary can control no more than of the total voting power and corrupt nodes. The malicious behaviors of adversary are as follows

* Adversary can launch Sybil attack, in which adversary can generate a large number of identities for corrupted nodes. These nodes can deviate from consensus protocol. Adversary may obtain inappropriate power in consensus process. Adversary can overwhelm other nodes through majority voting power, then control the generation of blocks and prevent other new nodes from entering system.
* Adversary can launch jamming attack to interfere with the message transmission in wireless networks. Jamming attack is a denial of service attack that adversary prevents other nodes from using the channel to communicate by occupying the channel they are communicating on. The consensus process can be interrupted due to adversary launches jamming attack. Since block finalization （需要明确）requires votes from more than a half of nodes in system, it will not be able to complete the consensus process. Therefore, the performance and security of blockchain system will be halted by jamming attack.

We design SWIB that can resistant both Sybil attack and jamming attack. Even an adversary launches attacks, blockchain system can still work and adversary cannot halt the consensus process.



# The Stable-aware Wireless Blockchain Consensus Protocol

In this section, we propose a stable-aware wireless blockchain consensus protocol, abbreviated as SWIB. We first give an overview of the protocol, and then present a detailed of SWIB. To ensure the security of the protocol, we propose an incentive and publishment mechanism and a synchronization mechanism.

## 4.1 Overview

In this subsection, we present the overview of SWIB, and describe four main stages of consensus process.

We design SWIB to be a protocol that adopts a random block proposer election algorithm that can prevent forks. SWIB uses a new voting mechanism that combines signature aggregation with broadcasting communication protocol. In SWIB, consensus nodes can join a blockchain system by submitting Sybil-resistant-proof. Each node registers in the blockchain system by depositing certain amount of money, which will be stored in a virtual account. Only the depositor executing unpledged operation can take out the money. Before participating consensus, each node first obtains its private-public key pair and a main public key through a distributed key generation algorithm. Besides, nodes should also request the identities and public keys of other consensus nodes as well as necessary blockchain information.

SWIB works in a round-by-round fashion. Figure 2 shows how SWIB works in a round. At the beginning of a consensus round, a block proposer will be randomly elected according to a secure random value. In general, it is not secure to allow nodes to predict who will be the block proposer in the next round. Thus, a randomness source is necessary to ensure that nodes cannot obtain the information of next block proposer in advance. SWIB adopts a distributed randomness generation scheme, which ensures that nodes can generate the same random value per round independently. (简略别人的东西，重点强调自己的东西)According to the random value, a block proposer will be randomly elected at the beginning of each round through a block proposer election algorithm based on nodes’ stability. Then, the elected block proposer will generate a block and broadcast it to other nodes. Nodes will vote on the validity of the block by generating partial signature through a partial signature generation method. Once aggregating a sufficient number of partial signature shares, consensus nodes can recover the full signature through a signature recovery method to finalize consensus round.



The operation of SWIB is presented in Algorithm 1, which shows the process of executing a complete consensus round. Before starting a new round, consensus nodes perform transaction broadcast. The broadcast operation in blockchain network is presented in Algorithm 2.（再加一段话引入广播算法，重要性。或者放在后面或者直接删除） Further details of SWIB will be given in the following subsections.



## 4.2 The SWIB Protocol

SWIB protocol contains four design components: block proposer election, block proposal, block verification, and block finalization. In the following, we shall present the detailed design of each of these components.

### 4.2.1 Block Proposer Election

（原理一段话，用稳定度选举首领，考虑质押和参与情况，用轮盘赌）

（描述一下伪代码）

（分析细节）

We present a random block proposer election algorithm (具体算法) that can prevent inherently forks. This algorithm is based on a distributed randomness generation scheme and threshold BLS signature scheme. SWIB uses three important algorithms of the threshold BLS signature scheme: a signature generation algorithm to generate partial signature; a signature recovery algorithm to reconstruct the full signature from a sufficient number of partial signature shares, and a signature verify algorithm to check the validity of signature. We define the definition of nodes stability, which is used as the elected weight of node. This strategy can prevent Sybil attack. Lines 10-11 of Algorithm 1 constitute the block proposer election process.

The block proposer election algorithm adopts a random seed to guarantee the security of election process. The distributed randomness generation scheme can enable all consensus nodes to jointly generate a round-related randomness in an unbiased and unpredictable manner. The inputs of the scheme in each round include the round number , the block hash of the previous round , and the full signature of the previous round . Using the full signature as randomness source can ensure the uniqueness and immediacy of the inputs. As shown in Fig. 3, the current round random value is performed as the normalized hash value of the above input combination:

For simplicity, the very first random value is set to be the normalized hash value of the genesis block of blockchain.



**Fig. 3. Randomness generation in each round.**

The output of the distributed randomness generation scheme is unpredictable and unique. The recovery process of full signature for each round is unpreventable, provided that majority of the consensus nodes are correct as we assume. Consensus nodes can recover the full signature or received by other consensus nodes with high probability in finite time. Although the block hash and round number are known in advance, the full signature can only be revealed at the end of round. The full signature cannot be tampered due to the security of threshold signature scheme, even a node first recovers full signature before others. Thus, no one can predict the output of the distributed randomness generation scheme in advance. Another benefit of the distributed randomness generation scheme is that nodes can enter the following round non-interactively. Upon receiving or generating a valid full signature, consensus nodes will append the corresponding block into their local blockchain. Then, each node will start a new consensus process by computing the following round random value.



**Fig. 4. The block proposer election for round .**

After execution of the distributed key generation scheme, each node obtains a private-public key pair. Nodes know the public keys of other nodes by exchanging messages. The public key list can be seen as the identities of nodes. To ensure a same view on the node list, nodes will sort the list according to the hash values of public keys. As illustrated in Fig. 4, all nodes have a same view of the public keys list.

The block proposer for each round is elected according to elected probabilities. In SWIB, the elected probability of a node is determined by its stability, which is a new concept. Let be the remaining active time of node in the system, then the sum of all consensus nodes' remaining active time will be . The active time ratio of node (denoted by ) is calculated as . The consensus ratio of node (denoted by ) is calculated as , where is the number of blocks generated by in the latest blocks on the blockchain. When the blockchain length is smaller than , the consensus ratio of every node is set as zero. The stability of node is given as follows

where and () are the weights of active time ratio and consensus ratio, respectively. According to nodes' stability values, we have the elected probability of each node :

The block proposer election algorithm ensures that the election process is randomized. To determine the block proposer for the current round, the election algorithm divides interval into consecutive intervals

If , then the node whose public key ranks the will be the block proposer of round . Each consensus node can independently determine the block proposer of current round. Meanwhile, each node can also verify the legality of an elected block proposer according to the maintained node list. Since the secure threshold of the threshold BLS signature scheme can be satisfied, a block proposer can be elected in each round when the full signature for its preceding round is recovered.

### 4.2.2 Block Proposal

The elected block proposer will pack transactions to generate a new block. The header of the block can be represented by a tuple , where is the round number, is the hash value of previous block, is the current block hash, and is the identity of the block proposer, and is the root of transactions. Block body usually stores transaction metadata, which is a transaction list. The block proposer will broadcast the block to other nodes. In addition, the full signature can only be appended to this block when a node gathers enough partial signature shares and recover the signature.

### 4.2.3 Block Verification

After receiving a block, consensus nodes will verify the legality of block proposer and the content of the block. The pseudo code of block verification is presented in Algorithm 1 (see lines 34-37 therein). In block verification phase, a node needs to check the validation of the proposed block through the following components:

* Block proposer: The result of block proposer election algorithm should be the same with the index of block proposer in its node list.
* Previous hash : The previous hash of current block has to equal the hash of latest confirmed block.
* Transactions: All transactions included in the proposed block should not conflict with previous confirmed transactions.

If all the above conditions are satisfied, then a node can generate partial signature of the block as a vote. Nodes will broadcast their votes to other nodes via wireless network.

### 4.2.4 Block Finalization

(看不出那些东西是我的，重点强调自己创新的部分)SWIB uses a new voting mechanism, which combine a secure threshold BLS signature scheme with broadcast protocol. SWIB uses the full signature as the proof of block finalization. As shown in Fig. 5, when a node gathers a sufficient number of partial signature shares of the block, the node can recover the full signature through the signature recovery algorithm of the threshold BLS signature scheme. The reconstruction of valid full signature proves that a given threshold of nodes sign on the block, which means that a sufficient number of nodes vote for the block validity. Therefore, it is feasible to use the full signature as the signal of block finalization.

The stability of consensus process is improved since full signature recovery can be done independently by any correct node. Any node that aggerates enough partial signature shares can recover the full signature. Even if malicious nodes refuse to recover the full signature or to broadcast valid full signature to other nodes, block finalization can still be reached when there are enough honest nodes. This design enables block finalization to be steadily achieved in a wireless network with faulty nodes or unreliable communication channels. Moreover, since each correct node can only vote at most once in a round, only one block can be confirmed in a complete round. Therefore, SWIB satisfies strong consistency, which means that it can prevent blockchain forking.



**Fig. 5. Block verification and finalization at a node .  
1. 强调自己的东西，多用WE做主语**

**2. 4.2.1标注**

3. 伪代码调整一下，以四个阶段来写，不要放在While里面

## 4. 3 Incentive and Punishment Mechanism

In this subsection, we present an incentive and publishment mechanism to improve the security of SWIB. An incentive mechanism is necessary to

We design an incentive mechanism to encourage consensus nodes to participate the consensus process. Block proposer might be reluctant in block generation due to considerable power consumption for block packing and broadcasting. Besides, both block verification and signature generation consume certain computational power at nodes. Rational nodes may be willing to wait for receiving full signature from others, rather than consuming their computational power to verify a block, generate partial signature, and recover full signature. Therefore, an incentive mechanism is needed to motivate nodes to participate the consensus process actively. The rewards in blockchain system are the submitted transaction fees in valid blocks. In our incentive mechanism, part of the transaction fees is rewarded to block proposer, and the rest will be averagely distributed among those nodes whose partial signature shares are used in recovering the full signature.

Part of the transaction fees is distributed to block proposers, which encourage them to generate valid blocks containing as much transactions as possible.

The rest fees will be shared among those nodes whose partial signature are used to recover the full signature that appended into the valid block in block finalization. Because the uniqueness of threshold BLS signature scheme, any different partial signature shares can recover the same full signature. Since the full signature for a block may be recovered by more than two distinct sets of partial signature shares due to message losses. In this case, the rest fees will be shared by the set of nodes who have the smallest average timestamps of partial signature shares used to recover the full signature. In other words, we reward those consensus nodes signed the block hash as quickly as possible. This incentive mechanism not only encourages nodes to verify block and generate partial signature, but also motivates nodes to broadcast partial signature and full signature. The earlier the signature broadcasting, the higher the chance being rewarded. [当存在部分消息丢失的情况下，有可能一部分节点认可某个recovered full signature，而另一部分节点认可另外一个，这样会产生分裂，这怎么解决？怎么分钱？ ]

In addition, we adopt a punishment mechanism to reduce the probability that rational nodes become malicious. The malicious behaviors of nodes contain the following: 1) when being elected as the block proposer for a round, a node generates an invalid block for that round or does not generate any block before timeout; 2) a node sends invalid signature or garbage messages in block verification and finalization phase. Irrational nodes may harm others without benefiting themselves. A punishment mechanism is necessary to suppress such malicious behaviors. In SWIB, the active time determines how long a node will work continuously in the blockchain system. Hence, reducing the active time of a node is a good measure to punish a node behaving maliciously.

Reducing the active time of malicious nodes can increase the security and performance of blockchain system. This is because reducing the active time of a node will decrease its stability value, which further reduce its probability being elected. In this way, there will be reduced chance to finalize empty blocks due to malicious node behaviors. The more valid blocks finalized, the higher the system performance will be. Moreover, irrational nodes will be quickly expelled from the system if they initiate malicious behaviors frequently. As a result, the security of blockchain system will be improved. Therefore, the punishment mechanism can reduce the continuous impact of malicious behaviors and encourage nodes to behave honest. This is because rational nodes prefer to follow the consensus protocol to win reward rather than become malicious. In this way, the system throughput will not be significantly decreased since honest block proposers still strive to generate valid blocks.

## 4. 4 Synchronization Mechanism

In our protocol, we consider a more efficient and secure self-initiated mechanism. When a new node joins the blockchain system, it is necessary to get information of other consensus nodes and necessary blockchain history before participating the consensus process. In SWIB, a joining node will request necessary blocks of blockchain history from nodes with high stability. Nodes with higher stability are more likely to maintain the latest blockchain, and have lower probability of sending fault blocks.

When the total number of blocks of blockchain history is , a new joining node can request different blocks from different neighbors with high stability and small distance. Let be the communication interruption probability between and , then can successfully receive a block from with probability . [前面给出的那个式子，不是block interruption probability，最多是个message interruption probability] The probability that a node *v* can receive historical blocks successfully from node *u* is . This design choice can reduce the overhead of blockchain history retrieval. The load balancing strategy used in this process can also effectively prevent a single node failure due to large energy consumption for blockchain history transmission. Meanwhile, synchronization mechanism will reduce the risk that a node transmits error blockchain history to mislead new nodes. Once a node transmits error blockchain history information, the receiving node can immediately detect it through blocks received from other nodes. In this case, the new node can request blocks from other trustworthy nodes to ensure the receipt of correct blockchain history information.

Synchronization process can also happen when a consensus node receives a partial or full signature without receiving the corresponding block. In this case, the node will request the corresponding block from its neighbor with high stability and good link quality if the node itself is a honest node. [逻辑有点问题，因为不知道对方是否已经收到该block，这种request方式有些盲目；一个解决办法是，从多个收到partial/full signature的节点里，选择与自己链路质量最好的] If the previous hash of a received block does not equal the hash of the latest block on the node's local blockchain, the node will also request blockchain history from its neighbors. In this case, each received block will be appended to the node's local chain if it is absent. When the maintained chain is the latest, the node participates the consensus process via generating the round-related random number according to the full signature of latest block. This procedure ensures that blockchain system will not be stopped provided that the number of honest nodes meets the security threshold. Thus, SWIB can utilize the synchronization mechanism to guarantee the security of SWIB.