CredaCash™ Blockchain Assembly

Creda Software, Inc.

CredaCash™ uses a blockchain as a public ledger to ensure only valid payments can be spent, and that they can be spent only once. In the CredaCash system, transactions are assembled into a blockchain by a small number of pre-selected “witnesses”. The system was developed to meet the following goals:

- There should be a definitive point in time at which a block becomes permanent and “indelible” and cannot under any circumstances be replaced with another block. Users can then rely on the permanence of transactions inside these blocks when accepting payments.

- All nodes on the system should have the same view of the permanent blockchain; in other words, it should be impossible for two nodes to accept non-identical blocks in what each sees as the indelible part of the blockchain.

- Transaction processing should be capable of operating at a high rate of speed, ideally as fast as a dedicated payment processing network.

- The system has to operate correctly even in the presence of an unreliable network, which might include delayed and out-of-order delivery of blocks.

- It should operate reliably even if some limited number of witnesses go offline, malfunction and generate incorrect blocks, or are taken over and operated maliciously in an effort to subvert the blockchain.

- Every node on the network can determine when a block and the blockchain are valid, and when a block is invalid or missing from the blockchain.

- It is resistant to forgery and tampering.

- It is resistant to denial-of-service attacks.
- It is reasonably efficient, i.e., it can meet the speed and security goals without excessive use of computational power.

In order to meet those goals, the following system was created:

1. The witnesses are preselected from a group of reliable, geographically diverse hosts.

2. Witnesses communicate with each other and with the rest of the network using the Tor network and Tor hidden services in the same way relays communicate. Witnesses receive transactions forwarded by the relays, and, after assembling them into blocks, the blocks are sent back out across the network to all connected relays.

3. Each block contains a 64-bit level, which increments sequentially for each block added, and the 512 bit Blake2b hash of the prior block in the blockchain. This data uniquely identify the sequential chain of blocks in the blockchain, while the large hash prevents a witness from replacing a block it created with a different block after another witness has built a block "on top" of it.

4. Each witness signs the blocks it creates using Ed25519-SHA3. The initial public signing key for each witness is pre-programmed in the network node software and is used to verify the initial block signatures. When a witness assembles a block, it also generates a new signing key pair it will use to sign the next block in the chain, and includes the public key with the current block. In this way, the signing keys are constantly changing. In addition, as soon as a block becomes indelible, the private key used to sign the block is erased from the witness's memory and is gone forever. This makes it difficult for anyone to manipulate the historical record of the blockchain if they were to succeed in obtaining a private signing key.

5. The system allows the possibility that, at any point in the blockchain, one or more witnesses might malfunction or be exploited and operated by a malicious party with the

goal of executing a double-spend attack or causing the block assembly to malfunction. In the protocol, these malfunctioning or maliciously operated witnesses are referred to as "mal witnesses".

6. There are two important parameters that control the operation of the blockchain:

nwitnesses := the number of witnesses that are allowed to create blocks at a particular point in the blockchain.

nmaxmal := the maximum number of "mal witness" at a particular point in the blockchain that can malfunction or be maliciously operated without affecting the operation of the blockchain.

7. For the system to operate correctly:

$$\text{nmaxmal} < \text{int}((\text{nwitnesses} + 1) / 2)$$

In other words, correct operation cannot be guaranteed if a majority of the witnesses malfunction or are operated maliciously since the mal witnesses could create a blockchain or multiple blockchains that violate the system requirements.

8. From the above two parameters, two additional important parameters are computed:

nconfsigs := the number of witnesses that need to confirm a block (including the witness that created the block) in order for the blockchain to continue advancing.

nindelblocks := the number of blocks that need to confirm or build upon a block (including the block itself) in order for the block to become indelible.

9. In the CredaCash system, the values of these two parameters are:

$$\text{nconfsigs} = \text{int}((\text{nwitnesses} - \text{nmaxmal}) / 2) + 1 + \text{nmaxmal}$$

$$\text{nindelblocks} = \text{nwitnesses} + \text{nmaxmal}$$

10. The first of these parameters, nconfsigs, can be intuitively understood as a majority of the maximum possible number of correctly operating witnesses plus the maximum number of mal witnesses. It might be tempting to say that the blockchain with more

than one possible witness should be able to advance with only one correctly operating witness. The problem with that approach is that, due to network transmission errors or delays, two good witnesses might operate without receiving any blocks from the other. If both could proceed, they would produce two completely different blockchains in violation of the requirement that there can be only one authoritative blockchain. In order to ensure every node sees the same valid block chain, the blockchain can only proceed if it contains blocks from a majority of the maximum number of correctly operating witnesses. It then becomes impossible for two different blockchains to exist since only one can contain blocks from a majority of the correctly operating witnesses.

The maximum number of correctly operating witnesses is nwitnesses – nmaxmal, and a majority of the maximum number of correctly witnesses is int((nwitnesses – nmaxmal) / 2) + 1. To this number we must add the maximum number of mal witnesses. A mal witness may not be following the rules, and may attempt to build on two different blockchains. Since it is not known specifically which witnesses are mal—we are just making an allowance that at any time up to nmaxmal witnesses could malfunction or be exploited—we must account for that by ignoring the contributions of nmaxmal witnesses. The total number of witnesses required to advance the blockchain is therefore int((nwitnesses – nmaxmal) / 2) + 1 + nmaxmal.

11. The value of the nindelblocks parameter can be intuitively understood as the maximum number of correctly operating witnesses, nwitnesses – nmaxmal, plus two times nmaxmal, which is (nwitnesses – nmaxmal) + 2 * nmaxmal = nwitnesses + nmaxmal. This number arises because a block may be created by a mal witness and then followed in turn by blocks from all the other mal witnesses, by all the correctly operating witnesses, and then again by all the mal witnesses. At that point, the original block has nindelblocks confirmations (including the original block itself), and if the rules for blockchain assembly set forth below are followed, no chain that competes with the original block can advance as far, and therefore the block with nindelblocks

“confirmations” has become indelible since it is in the only chain that can continue to advance.

12. It is possible for the values of these parameters to vary over time, i.e., witnesses can be added or removed and the allowance for mal witnesses can be increased or decreased. These parameters can be varied by inserting a control message in a block, and become effective for all blocks built on top of the block containing the control message. If any other witness does not agree with the change, it can refuse to build on the chain that contains the control message, and instead build on one of its predecessor blocks. If fewer than nconfsigs witnesses are willing to agree to the change, the chain containing the control message will not advance since blockchain advancement requires the agreement of at least nconfsigs witnesses.

13. For the purpose of describing the rules below, each witness is assigned an integer witness number called witness_id from 0 to nwitnesses-1 inclusive. Each block is also assigned a witness_id, which equals the witness_id of the witness that creates the block.

14. The simplest possible implementation of blockchain assembly using a set of witnesses would be to have the witnesses operate round-robin, each creating a block in turn and adding it to the blockchain, so that for every block, the witness_id would equal the prior block’s witness_id + 1 modulo nwitnesses. However, that system would come to a halt if one of the witnesses for any reason did not build a valid block. In order to continue advancement of the blockchain when some number of witnesses are not operational, the system must allow for skips in the witness_id sequence.

15. Let the skip between two consecutive blocks be defined as:

skip = (next - ((prev + 1) % nwitnesses)) % nwitnesses

where prev := the witness_id of the earlier block in the chain

and next := the witness_id of the later block in the chain

From this definition, if two blocks have consecutive witness_id's (for example, 0 and 1) then the skip is zero. If the witness_id's differ by 2, then the skip is 1, etc.

16. Unique Signatures Rule: The CredaCash system does not allow arbitrary skips in the witness_id sequence. In order for a block to be valid, the sum of the nconfsigs-1 skips that immediately precede the block (including the skip between it and its predecessor) must be less than or equal to nwitnesses:

sum(skip[i]) over the nconfsigs-1 skips preceding the block <= nwitnesses

If a block violates that rule, it is invalid and discarded. Any number of witnesses can attempt to violate this rule without affecting the integrity of the blockchain since the blocks that violate this rule will be rejected by the other nodes in the system.

17. The Unique Signatures Rule ensures no witness can create more than one block within any span of nconfsigs blocks. More importantly, it means that for every block, the next nconfsigs-1 blocks will come from different witnesses, so that after nconfsigs-1 additional blocks, the original block will have been confirmed by nconfsigs different witnesses (including the witness that created the block). This rule prevents one or a small number of witnesses acting alone to advance the blockchain. In order for the blockchain to advance, at least nconfsigs different witnesses need to be operational and agree on the blocks to be added. This property is required to ensure the blockchain assembly operates correctly even in the presence of network transmission delays.

18. The Unique Signatures Rule also imposes an ordering property that causes the block witness_id's to be ascending modulo nwitnesses, i.e., for all blocks in a span of nconfsigs blocks, the skip must be <= nwitnesses - nconfsigs. While enforcement of this property by itself is not required for correct operation, the ordering property ensures no block will be added to a chain if the skip from that block would result in a chain that would eventually come to an end due to the nconfsigs different witnesses requirement; in other words, the ordering property imposed by that rule prevents the system from pursuing dead end chains.

19. The witnesses must also nominally adhere to the following rules. The word "nominally" is used because up to nmaxmal witnesses can violate the following rules without affecting the validity of the blockchain:

20. Chain to Indelible Rule: A witness may only build a block on top of a chain that ends in or leads back to the most recent indelible block. In other words, if two competing chains exist, and the blockchain advances to the point that the earliest block in one of the two chains becomes indelible, the other chain must be disregarded and not further built upon. This rule ensures the witnesses acting as a group will not attempt to replace an indelible block.

21. Note however that for this and for all rules, a witness is only required to act based on the blocks it has received; it is not required to act based on blocks that may have been created by other witnesses but it has not yet received due to network transmission delays. For example, using the prior rule, one witness may believe a block in one of the two competing chains has become indelible based on the arrival of a new block, while a second witness that has not yet received the new block may continue to build on either chain because that witness does not yet consider the prior block to be indelible. This situation does not violate the rules—a witness is only required to act on the blocks that it has seen, not on blocks it has not yet received.

22. Better Path Rule: A witness will only create a block that has a "better" path than any previous block it has created on a chain that leads back to the most recent indelible block. A path is better when it has a lower "skip score" which is defined as the string of skips concatenated together from left to right starting from the most recent indelible block and ending at the block of interest. Scores are compared from left to right and the lower score is the one with the lower skip at the left-most position at which the strings differ. If the strings do not differ at any position, then the longer string has the lower score.

23. The Better Path Rule allows a witness to begin building on a lower score path than it has built on previously, but it prohibits a witness to begin or continue to build on a higher score path after it has built on a lower score path. This prevents the witnesses as a group from building indefinitely on more than one competing path, since once a majority of witnesses have built on the lower score path, they can no longer build on the higher score path.

24. Note that as the blockchain advances, if a block that a witness created no longer chains back to the most recently indelible block, that block is no longer used to compute the witness's best previous skip score. As a result, if a majority of the witnesses choose a higher score path, the witnesses who built on a lower score path will begin to follow the majority after the blockchain has advanced and the lower score path no longer chains back to the most recently indelible block.

25. Increasing Level Rule: A witness will only build a block at a higher level than the block it last created. In other words, if a witness created a block at level 204 on one chain, it will not subsequently create a block at level 204 or lower on that or any other chain. This rule works in conjunction with the Better Path Rule to ensure the witnesses choose a single path for the blockchain.

26. Note that up to nmaxmal witnesses can violate the above rules without affecting the correct operation of the blockchain. If more than nmaxmal witnesses violate the rules, then more than one block may appear to become indelible. The other nodes in the system will detect the conflicting indelible blocks and immediately halt any further acceptance of blocks and advancement of the blockchain until the issue is resolved. All nodes on the network therefore work together to keep the witnesses "honest" and ensure they operate correctly.

27. Note also it might be contemplated that one or more witnesses might conspire to attempt a double-spend attack against two parties X and Y. This might be attempted by sending party X a block containing a payment to X, while sending party Y a block

paying the same output to Y. This attack is essentially made impossible by the use of the Tor network. Because all of the other nodes connect to the witnesses using Tor, the witnesses have no way to identify X and Y in order to target them with different blocks. Furthermore, due to the highly interconnected relay mesh, any conflicting blocks would be quickly detected and the network would halt. For these reasons, the witnesses cannot successfully execute a double-spend attack.

The rules described above are sufficient for correct operation of the blockchain. There are however a few aspects left to describe.

28. The Better Path Rule allows a witness to create a block at any location as long as the skip score of the new block is lower than all blocks the witness has previously created that chain back to the last indelible block. Under this rule then, if a witness can create blocks at more than one location in the block chain, it can choose any of those locations regardless of their relative skip scores. While not required for proper operation, the blockchain advances faster and more efficiently (i.e., with fewer sidechains) if all witnesses, when they have a choice, always create the block with the lowest possible skip score.

29. The rules do not require any particular ordering of the witness work--the witnesses could all build blocks simultaneously wherever permitted by the above rules. That would however lead to all nodes in the system validating multiple blocks to find the best possible path forward. It is more efficient for the witnesses to go round-robin to the extent possible. In such a protocol, a nominal block rate could be chosen, for example, one block every two seconds. The witnesses would then go in turn, with each witness creating a block two seconds after the prior witness. If a witness fails to generate a block, the next witness in order would wait for its turn based on the block rate and then create a block. If there were no work to do, i.e., there were no transactions that had yet to be added to blocks and no blocks containing transactions that had yet become indelible, then all witnesses could pause and wait to receive a transaction. When one is

received, they would all restart their clocks and resume the witness sequence where it left off.

30. In the event a witness crashes or needs to be restarted, the remaining witnesses can send a block containing a control message to first drop that witness and then to add another witness. The add message would include the new witness's public block signing key.

31. If however more than nconfsigs witnesses go offline, the system needs another way to resume operation. To address this, each witness is also associated with a key pair that can be used to sign a reset block. The public key is preprogrammed in the network node software while the private key is kept on an air-gapped host. If required, a reset block containing a new block signing public key is created on the air-gapped host and then securely transferred to the network. This is repeated for as many witnesses as necessary to resume operation.

32. The protocol above can run in the steady state with only nconfsigs witnesses generating valid blocks. There can however arise a situation where, due to the above rules, the blockchain is unable to advance even though nconfsigs witnesses are operating correctly. This can occur for example when a witness generates a block and then goes offline leaving only nconfsigs witnesses operating. One example is the following: With nwitnesses = 4 and nconfsigs = 3, witness 0 creates a block; witness 1 builds on 0 but that block is not seen by the other witnesses; 2 also builds on 0; 3 builds on 2 then goes offline; 0 builds on 3; and 2 builds on 0. The blockchain at that point cannot advance further since witness 1 will not build on any path but its own since that would violate the Best Path Rule, while witnesses 0 and 2 will not build on 1 because that would violate the Increasing Level Rule.

There are three potential methods to clear such a jam:

Method 1 is to manually restart one or more witnesses using a block signed with the witness's reset key.

Method 2 is fully automated and requires no control messages but works only when nmaxmal is zero. In this method, the witnesses all evaluate every block to see if it is "viable". A viable block is one that can gain nconfsigs sigs from the most recent indelible block. The blocks are evaluated from the lowest score block first. The Increasing Level Rule is always applied, but the Better Path Rule is only applied against lower score blocks that have been found to be viable and against higher score blocks that have not yet been evaluated. When a block is found to not be viable, the witness marks the block as excluded from the set of candidate blocks upon which it might build a new block, and that block is no longer included in the Better Path Only Rule. After evaluating all blocks for viability, the witness attempts to build on one or more of the viable blocks. Note that this method can be run continuously to prevent some jams, or it can be run only when a jam is detected.

Method 3 is fully automated but requires the witnesses to exchange control messages.

Step 1: The witnesses exchange messages to determine which witnesses are still working, and to agree on the most recently indelible block and the set of blocks that chain back to the most recently indelible block.

Step 2: The blocks are evaluated to determine which blocks can become indelible by blocks generated solely by the non-working witnesses with help from every possible permutation of nmaxmal malicious witnesses. The result of this computation should be either no blocks or a single linear chain of blocks from the most recently indelible block. The highest level block in this chain is temporarily considered to be an indelible block and the level of this block is considered to be the highest level at which all witnesses have created a block.

Step 3: The remaining non-indelible blocks are evaluated to find the lowest scoring block that is viable. A block is viable if a chain of at least nconfsigs can be built on top of it using only the still-working witnesses. If no viable block is found, then the blockchain remains jammed until one of the non-working witnesses comes back online. Otherwise, by prior agreement, the working witnesses will only build blocks on top of the lowest scoring viable block, and they begin doing so. The new blocks reference the state exchanged above and the resulting computation of the lowest scoring viable block to ensure the witnesses are in sync, and the witnesses will not confirm each other's blocks if there is a discrepancy.

The system described in this document has been demonstrated to work correctly under all conceivable situations using extensive simulation testing, and meets all of the goals set forth in the introduction.

https://CredaCash.com

Rev 1
2016-01-26