# **The protocol of the Dagger (XDAG) crypto currency**

### **Version 0.3. January 19, 2018**

## **Terms**

*Address of block* – base64 encoding of truncated hash of the block.

*Amount* – quantity of XDAG measures in cheato.

*Block* – the basic structure of the Dagger. Each block has fixed size 512 bytes. Structure of block is defined in section 2.

*Chain* – a sequence of blocks each block of which is referenced by the previous block.

*Cheato* – one XDAG consists of power(2,32) cheato*.*

*Dagger era* – time 0x5A500000 in Unix format, i.e. January 6, 2018, 22:45:20 GMT.

*Difficulty of block* – integer number equal to (power(2,128)-1) / (hash\_little / power(2,160)), where hash\_little = little-endian(hash(block)), it is the integer representation of block hash in little-endian format, hash\_little has 256 bits.

*Difficulty of chain* – sum of difficulties of blocks from the chain.

### **Hash – hash(block) = sha256(sha256(block)).**

*i-Reference* – block B is i-referenced by block A if the i-th field of block A is link to a block C and block C is referenced to the block B.

*Link* – block A is linked to block B if A has field of type 2 or 3, which contains the truncated hash of the block B, see the structure of block in the section 2.

*Main block* – any block from main chain.

*Main chain* – separate chain with maximum difficulty.

*Reference* – block B is referenced by block A if there is sequence of blocks starting from A and ending on B in which each block is linked to the next block.

*Separate chain* – a chain in which each two block belong to different time frames.

*Shortest chain* – a chain from block A to B in which each block is a link to the next block and each block is a i-reference to the block B for smaller possible i.

*Time frame* – time segment from t seconds to t + 64 seconds where t is a time in Unix format and low 6 bits of t equal zero. First time frame begins on the Dagger era.

*Time of block* – time coded in the header of block, measured in 1024th fractions of second since Unix era i.e. January 1, 1970.

*Transaction* – synonym of block.

*Truncated hash* – 24 low bytes of hash in little-endian representation.

## **2. Block**

A block consists of 16 fields. Size of a field equals 32 bytes. Each field has type from 0 to 15. For regular blocks field 0 must have type 1. For transport pseudo-blocks field 0 must have type 0. See 5. Transport layer for description of transport preudo-blocks.

The next table describes fields of different types.

|  |  |
| --- | --- |
| Type of field | Description |
| 0 | Nonce. Field may contain any bytes. |
| 1 | Header of the block. Has the structure:     * transport layer header, 8 bytes; this field must be set to zero when counting hash; * 64-bit little-endian number, contains types of fields 0 .. 15. Each type has 4 bits. Type of field 0 is encoded in low 4 bits. * 64-bit little-endian number, time of the block. * 64-bit little-endian number, amount of the fee of the transaction. |
| 2 | Input of transaction. Link to another block B. Has the structure:     * truncated hash of block B, 24 bytes; * 64-bit little-endian number, the amount that is taken from the balance of block B then the transaction is applied. |
| 3 | Output of transaction. Link to another block B. Has the structure:     * truncated hash of block B, 24 bytes; * 64-bit little-endian number, the amount that is added to the balance of block B then the transaction is applied. |
| 4 | Half of input signature. Input signature consists of two subsequent fields of type 4. The first field contains the **r** number, and the second – the **s** number from ECDSA signature. Signature is ECDSA signature derived from the following digest and signed by the private key corresponding to the same public key. Digest is:    hash(modified\_block # key\_prefix\_byte # public\_key)    where:     * # is concatenation * modified\_block is the block in which these two signature fields and all following input and output signature fields are filled by zeroes; * key\_prefix\_byte – byte 0x02 if the number **y** of public key is even, 0x03 if the number **y** of public key is odd. * public\_key – the number **x** of public key, 32 bytes. |
| 5 | Half of output signature. The structure is the same as for input signature. See 3. Algorithm for difference between input and output signatures. |
| 6 | Even public key. This field contains the number **x** of ECDSA public key if the number **y** is even. |
| 7 | Odd public key. This field contains the number **x** of ECDSA public key if the number **y** is odd. |
| 8 - 15 | Reserved for future usage. |

***3. Algorithm***

Each block is a transaction. Transaction may have several inputs, outputs, public keys, input and output signatures, and fee. Block A is valid if the following conditions are met:

* time of block A is not less than the Dagger era;
* time of each input or output of block A is less than the time of block A;
* each input or output of block A is a valid block;
* sum of all input amounts of block B is less than power(2,64);
* sum of all output amounts of block B plus its fee is less than power(2,64);
* if there is at least on input than sum of all inputs must be not less than sum of all outputs plus fee; otherwise sum of all outputs must be zero;
* for each input B of the block A there are public key K and input or output signature S in the block A and output signature T in the block B such that signature S is obtained from block A using key K and signature T is obtained from block B using the same key K (informal description: only owner of block B can withdraw money from it).
* number of output signature fields must be even instead of number of input signature fields may be odd; in this case the last input signature field may be used as nonce which can be altered without rebuilding any signatures.

In every moment of time there is one main chain of blocks (see 1. Terms). Blocks from main chain are called main blocks. Every two main blocks belongs to different time frames. So, there are one or zero main blocks in each time frame. In every main block new XDAG are mined. Mined amount for each of first power(2,21) main blocks of the main chain is 1024 XDAG. For the second power(2,21) main blocks of the main chain the mined amount is 512 XDAG. And so on, for the next power(2,21) main blocks from main chain mined amount is decreased by 2 times.

Blocks are ordered by the following order rules:

1. If block A is referenced by a main block M, but block B is not referenced by M, then A precedes B.
3. If blocks A and B are equal by the rule 1 and M is the minimum main block such that A and B are referenced by M, and C is nearest to M common block in shortest chains from M to A and from M to B, and A is i-referenced by C, and B is j-referenced by C where i < j, then A precedes B
5. If blocks A and B are equal by the rules 1-2 and block A is referenced by B, then A precedes B.

Transactions are applied in the above order. If transaction can’t be applied due to some reason described below it is not applied.

Each block also has its own amount of XDAG. At the start point amount of each block equals zero. Amount of block A changes in the following cases:

* new XDAG are mined in the block A. In this case amount of block A is increased by the mined amount.
* a transaction B is applied and block A is input of transaction B. In this case amount of block A is decreased by the amount written in the corresponding input field of block B.
* a transaction B is applied and block A is output of transaction B. In this case amount of block A is increased by the amount written in the corresponding output field of block B.
* a transaction B is applied and block A is the minimum block in the above order which is linked to the block B. In this case amount of block A is increased by the transaction B fee.
* Transaction A itself is applied and sum of all its output amounts plus fee is less then sum of all its input amounts. In this case amount of block A is increased by the difference of input sum and output sum plus fee.

Transaction A is applied if and only if all of the following conditions are met:

* for each input B of the transaction A the current amount in the block B is not less than the input amount from the block B written in the transaction A.
* sum of input amounts of transaction A plus current amount of block A is not less than sum of output amounts of block A plus fee of transaction A.

Since each block has its own amount, is it an account. User owns a block A if it has private key for any output signature in block A. User can access his account using its address. Address of block is base64-coded truncated hash of the block. Address is 32-bytes string of characters from the set A-Z, a-z, 0-9, /, +. User can withdraw any amount from its own account and transfer it to any valid block in the system.

## **4. Cryptography and Security**

For signing purpose, standard ECDSA signature in the openssl implementation is used. The elliptic curve is Secp256k1. Private key has a length of 32 bytes. Public key has a length of 32 bytes (the x coordinate of point in elliptic curve) plus one bit (oddness of the y coordinate of the point). Signature has a length of 64 bytes and consists of two fields. They are numbers r and s from the ECDSA signing algorithm.

Private keys are stored in the file wallet.dat placed in the work catalogue of the xdag program. Each private key has a length of 32 bytes. Public keys are not stored, they are calculated from private keys. When the user starts the xdag program for the first time, the program asks the user to enter random sequence of characters. This sequence is the seed for random generator for this user. The sequence is coded and saved in the permanent file dnet\_key.dat placed in the work catalogue of the xdag program. The file dnet\_key.dat is loaded every time the xdag program is started and its content is used for random generation. All private keys of user are generated using the above seed.

## **5. Transport layer**

Dagger hosts can exchange blocks using any transport protocol. The first 8 bytes of the block may be used for protocol-specific information. Mention, that this filed must be set to zero before counting any hash of block. The default xdag program uses the dnet network as transport layer. Transferred blocks are encrypted. Each transferred block is coded using the author’s semi-symmetric algorithm and the temporary key and decoded by opposite side using the same key. Temporary key is generated by the sender, then coded using private key of the sender which is stored in the file dnet\_key.dat, and then decoded by opposite side using the public key of the sender. The RSA cryptography is used here with the length of key 8192 bits.

Hosts can exchange not only blocks, but pseudo-blocks too. Pseudo-block is request to another host to send some information of reply for a such request. For pseuso-block, type of the field 0 must be 0, and type of the field 1 contains the type of message.

Types of pseudo-block messages:

|  |  |
| --- | --- |
| Type | Description |
| 0 | Request to send all blocks belongs to the given time interval. Start time of interval is placed in the time field of block header. End time is placed in the amount field of the header. |
| 1 | Reply for request of type 0. This reply should be sent after all requested blocks. |
| 2 | Request to send control sums of all block belongs to the given time interval. Start time of interval is placed in the time field of block header. End time is placed in the amount field of the header. |
| 3 | Reply for request of type 2. Reply is the single block in which last 256 bytes is the array of 16 sums structures. The time interval is divided into 16 equal parts and each structure corresponds to its part. Structure has two fields     * the first 64-bit little-endian number is the sum of all 64-bit numbers of all blocks of this sub-interval; * the second 64-bit little-endian number is the sum of lengths of all blocks from this sub-interval. |

Each request has its own ID placed in the field 1. Each reply must have the same ID in the field 1. Each pseudo-block message has the statistics structure placed in the fields 2, 3, and the beginning of the field 4.

The statistics structure:

* difficulty of main chain in the sender host, 16 bytes;
* maximum known difficulty on the main chain in the network, 16 bytes;
* number of valid blocks in the sender host, 8 bytes;
* maximum known number of valid blocks in the network, 8 bytes;
* number of main blocks in the sender host, 8 bytes;
* maximum known number of main blocks in the network, 8 bytes;
* number of hosts known to the sender host, 4 bytes;
* maximum known number of hosts in the network, 4 bytes.

Any other space in pseuso-block is filled by public host addresses known to the sender. Each address has length 6 bytes and consists of IP (4 bytes, big-endian order) and port (little-endian order).