WojakCoin

A Peer-to-Peer Meme Currency System

The WojakCoin Developers
WojakCoin Project — *github.com/WojakCoinProj/wojakcore*

Abstract

A purely peer-to-peer meme currency would allow online payments to be sent directly from one party to another without passing through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. WojakCoin proposes a solution to the double-spending problem using the same peer-to-peer architecture pioneered by Bitcoin, hardened with parameters chosen for a fast-moving, community-driven currency. The network timestamps transactions by hashing them into an ongoing chain of double-SHA-256 proof-of-work, forming a record that cannot be changed without redoing the work. WojakCoin targets a two-minute block interval and adopts a responsive difficulty controller that re-evaluates the network hash rate at every block, so that confirmation times remain stable even as participation fluctuates. The longest chain serves both as proof of the sequence of events witnessed and as proof that it came from the largest pool of honest CPU/ASIC power. As long as a majority of hashing power is controlled by nodes that are not cooperating to attack the network, they will generate the longest chain and outpace attackers. The network itself requires minimal structure: messages are broadcast on a best-effort basis, and nodes can leave and rejoin at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual payments.

WojakCoin is a continuation of this idea rather than a reinvention of it. It descends directly from the Bitcoin Core codebase and preserves Bitcoin’s consensus model, transaction format, and scripting system. What WojakCoin changes is the economic and timing envelope around that model: a shorter block interval for faster confirmations, a larger and faster-issued coin supply, a shorter coinbase maturity window, and a modern difficulty controller that reacts to hash-rate swings within a single block instead of waiting for a long retarget window. The result is a network with the security properties of proof-of-work and the responsiveness expected of a community currency. What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership. WojakCoin uses the same elliptic-curve signature scheme (ECDSA over secp256k1) and the same Script verification engine inherited from Bitcoin, so ownership and spending conditions are expressed and validated identically.

The problem of course is that the payee cannot verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority that checks every transaction for double spending. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank. We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we do not care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. To accomplish this without a trusted party, transactions must be publicly announced, and we need a system for participants to agree on a single history of the order in which they were received.

3. Timestamp Server

The solution begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it. WojakCoin embeds this proof of history in the coinbase of its genesis block, which carries the headline *“382017 Price Phillip Retires”* as evidence that no block could have been produced before that date.

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we use a proof-of-work system similar to Adam Back’s Hashcash. The proof-of-work involves scanning for a value that when hashed, such as with double SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. WojakCoin uses double SHA-256 (SHA-256d), the same hashing function as Bitcoin, which means existing SHA-256 mining hardware can secure the network from day one.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block’s hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it. The proof-of-work also solves the problem of determining representation in majority decision making. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of hashing power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.

5. Difficulty Adjustment

A short block interval makes a currency more usable but also more vulnerable to large, transient swings in hash rate, which can stall a chain for long periods if difficulty is allowed to drift. WojakCoin therefore operates two difficulty regimes. From the genesis block through block 999 the network uses the classical Bitcoin retargeting rule, recomputing difficulty over a fixed window so that early blocks settle predictably while the chain bootstraps.

From block 1,000 onward the network activates a second-generation controller that retargets at every block. It computes a weighted moving average of the difficulty targets of the most recent 24 blocks and compares the time those blocks actually took against the time they should have taken at the two-minute target. To prevent oscillation and timestamp manipulation, the measured timespan is clamped: difficulty can rise by at most a factor of two per block but is permitted to fall up to fourfold. This asymmetric damping lets the network recover quickly when hash rate leaves, without letting a sudden influx of hash rate spike difficulty out of reach. The effect is a smoother, self-correcting confirmation cadence that resists the “hash-and-run” mining behaviour that often afflicts smaller proof-of-work chains.

6. Network

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

7. Incentive

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. In WojakCoin the block subsidy begins at 100 coins and is halved every 210,000 blocks. With a two-minute target interval, each halving epoch lasts approximately 9.7 months, so issuance front-loads supply into the early life of the network and then tapers geometrically. Summing the series across all epochs yields an effective maximum supply of approximately 42,000,000 WJK; the consensus money range is bounded above by 44,210,526 coins. Each coin is divisible to eight decimal places, identical to Bitcoin’s smallest unit.

The steady addition of a constant amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU and ASIC time and electricity that is expended. Once a predetermined number of coins have entered circulation, the incentive transitions entirely to transaction fees, which keeps the network secure while remaining completely inflation-free thereafter. The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more hashing power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or

using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

8. Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle tree, with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored. A node that wishes to reclaim space can prune the spent outputs from its database while retaining the block headers and the unspent transaction set required to validate new blocks.

9. Simplified Payment Verification

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can obtain by querying network nodes until he is convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it is timestamped in. He cannot check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. Because WojakCoin inherits Bitcoin's header and Merkle structure unchanged, existing lightweight and SPV wallet designs apply directly.

10. Combining and Splitting Value

Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender. This UTXO model is preserved exactly in WojakCoin.

11. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. WojakCoin addresses for standard payments begin with the letter **W**, while pay-to-script-hash addresses begin with **3**, providing a clear visual distinction between simple and contract-style destinations.

12. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1. The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Given an attacker controlling a fraction q of the hash rate against honest fraction p = 1 – q, and a recipient who waits for z confirmations, the probability that the attacker ever catches up drops exponentially in z. The following tables, computed for WojakCoin's parameters, show that probability.

z (confirmations)	P(attacker succeeds)
0	1.
1	0.2045873
2	0.0509779
3	0.0131722
4	0.0034552
5	0.0009137
6	0.0002428
7	0.0000647
8	0.0000173
9	0.0000046
10	0.0000012
15	0.
20	0.
25	< 0.0000001
30	< 0.0000001
40	< 0.0000001
50	< 0.0000001

Table 1. Attacker with q = 10% of network hash power.

z (confirmations)	P(attacker succeeds)
0	1.
1	0.6277491
2	0.4457171
3	0.3245841
4	0.2391269
5	0.1773523
6	0.1321112
7	0.0987125
8	0.0739244
9	0.0554572
10	0.0416605
15	0.0101008
20	0.0024804
25	0.0006132
30	0.0001522
40	0.0000095
50	0.0000006

Table 2. Attacker with q = 30% of network hash power.

Solving for the number of confirmations z required to drive the attacker's success probability below 0.1%, we obtain: q=10% needs 5 confirmations; q=15% needs 8; q=20% needs 11; q=25% needs 15; q=30% needs 24; q=35% needs 41; q=40% needs 89; and q=45% needs 340. Because each WojakCoin block targets two minutes rather than ten, a given number of confirmations is reached roughly five times faster than on Bitcoin, so equivalent settlement assurance is available in a fraction of the wall-clock time.

13. Network Parameters

The following parameters are taken directly from the WojakCoin Core consensus rules and define the main network.

Parameter	Value
Ticker / unit symbol	WJK
Base codebase	Bitcoin Core 0.12.1 (client version 1.12.1.0)
Consensus / hashing algorithm	Double SHA-256 (SHA-256d)
Target block interval	120 seconds (2 minutes)
Difficulty regime (blocks 0 - 999)	Classical fixed-window retarget
Difficulty regime (block 1000+)	Per-block, 24-block moving average,
	+2x max rise / -4x max fall damping
Initial block subsidy	100 WJK
Halving interval	210,000 blocks (~9.7 months)
Effective max supply	~42,000,000 WJK
Consensus money-range ceiling	44,210,526 WJK
Smallest unit (decimals)	8 (1 WJK = 100,000,000 units)
Coinbase maturity	20 blocks
Maximum block size	1,000,000 bytes
P2PKH address prefix	W (version byte 73)
P2SH address prefix	3 (version byte 5)
WIF private-key prefix	201
BIP32 extended keys	xpub / xprv (0x0488B21E / 0x0488ADE4)
Default P2P port (mainnet)	20759
Default P2P port (testnet)	30759
Network magic (mainnet)	0x6f 0x8d 0xa5 0x79
Genesis timestamp	2017-08-03 01:45:14 UTC
Genesis coinbase headline	"382017 Price Phillip Retires"
Genesis block hash	000000004536a4f8fa9d88f0001ca9f9
	825f8d9fd3ba6383a2f030c0427bf085
Activated soft forks	P2SH, BIP34, BIP68/112/113 (CSV);
	BIP65 & BIP66 disabled
License	MIT

Table 3. WojakCoin main-network consensus parameters.

14. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we adopted the peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of hashing power. WojakCoin retains this proven foundation while tuning the timing and issuance schedule for a fast, community-scale currency,

and adds a responsive per-block difficulty controller so that those fast confirmations stay reliable under real-world conditions. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best-effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their hashing power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

WojakCoin is free and open-source software released under the MIT license and is experimental in nature. This document describes the protocol as implemented in the reference client; it is a technical specification, not investment advice or a solicitation. Participants should evaluate the software and any associated risks for themselves.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[2] Bitcoin Core developers, “Bitcoin Core 0.12.1,” source repository, 2016.

[3] A. Back, “Hashcash — a denial of service counter-measure,” 2002.

[4] W. Dai, “b-money,” 1998.

[5] R. C. Merkle, “Protocols for public key cryptosystems,” In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[6] P. Todd et al., “BIP68/112/113: Relative lock-time using consensus-enforced sequence numbers,” Bitcoin Improvement Proposals, 2015.

[7] WojakCoin Project, “wojakcore — reference implementation,” github.com/WojakCoinProj/wojakcore.