BITCOINAYT BTCYT WHITEPAPER

David Gomadza
www.twofuture.world
www.bitcoinayt.com
liveforever@bitcoinayt.com
davidgomadza@hotmail.com
00447719210295

THE NEXT BITCOIN BITCOINAYT BTCYT

Bitcoinayt: The Evolution of Digital Currency and Global Finance Bitcoinayt is the next evolution of digital currency—an advanced upgrade from Bitcoin, transforming it from a simple peer-to-peer system into a powerful global financial infrastructure. It is designed to function as both a decentralized currency and a structured financial system under a unified global authority—Tomorrow’s World Order (T.W.O).
Unlike traditional fiat currencies controlled by governments, Bitcoinayt operates under a higher centralized force, emulating the divine governance of Yahweh on Earth.

Set to fully emerge on June 26, 2025, Bitcoinayt is engineered as the financial backbone of a new era— one where humans are destined to live for over 10,000 years under the leadership of David Gomadza, Yahweh’s representative on Earth. This revolutionary digital asset is more than a payment system; it integrates the best aspects of Bitcoin, fiat money, and centralized banking, striking a balance between decentralization and structured governance.

At the heart of Bitcoinayt lies a bold, futuristic vision—the Live-On-Earth-Forever campaign— which envisions financial and technological breakthroughs that sustain eternal human existence. Every human is born with a Bitcoinayt wallet embedded within, waiting to be initialized by the command:
create.initialisebitcoinaytwallet.start

Bitcoinayt is not just another cryptocurrency—it is the future of global finance, an inevitable transition once Bitcoin reaches its predicted value of $1 million on June 25, 2025. Whether this timeline holds true

remains to be seen, but the blueprint for a one-world financial system beyond government control is
already in motion. With Tomorrow's World Order (T.W.O.) paving the way, Bitcoinayt will redefine the
monetary landscape, securing its place as the cornerstone of an advanced, everlasting human civilization.

Retrieved from "https://en.wikipedia.org/w/index.php?
title=User:The_Next_Bitcoin_Bitcoinayt_BTCYT&oldid=1278461437"

THE EQUATIONS FOR TRUE BITCOINAYT FOR MONEY US$

HOW DO WE KNOW FOR SURE WHO IS THE REAL CREATOR OF ANYTHING IN THE UNIVERSE A SIMPLE SOLUTION?

Simply remove and add letters back to the word or name of the thing you want to know who created it.
If we ask who created humans? This is how we know
1. Remove any letter or letters from the word human and 2. put it or them back. I decided to remove h from humans
and put the letter back
3. Your own body is designed to whisper answers silently.
The answer is Ya. Meaning the creator known as Yahweh [in Jewish language] and God in English abbreviated as Ya.
Now who created bitcoin or who is Satoshi Nakamoto?
1. We remove any letter from the word bitcoin and put it back. I decided to remove the letter t and add this letter back
The answer there are two people who created bitcoin
1. David Gomadza [myself] who created bitcoin for life that make people live on earth forever by changing
their day of death by more than 120000 years using 8000 of this bitcoin and other coins on the Richlist or
what is called the AGT. All billionaires at one point they searched for this AGT the Richlist in their life
because it means humans will never die and make the founder the richest in the world because anyone can give
up us$ money to live on earth for more years hence when time comes people will swap money for life in good health
we have already accomplished this as we speak get free 110 years added to your life simply say:
I request everything I need from Earthreserves
written as create.irequesteverythingineedfromearthreserves.start

2. There is another creator of bitcoin who is called billgates according to this method but i could be wrong so not
financial advice do your research.
But can we do the same to find out who put these two words together Satoshi Nakamoto using the same method this can
point truly to who created bitcoin for money us$ the current bitcoin?
From the name Satoshi Nakamoto I will first remove he first letter S and then put it back on.
The answer is
Bill Gates
Now let’s remove o from the first name and k from the surname and put all back.
The answer is Bill Gates.
So, can Satoshi Nakamoto be Bill Gates? Food for thought. Again, not financial advice my method could be wrong.
Try all these Nike, United States of America, Prada, Ferrari, etc.

The bitcoin Riddle (extract)
if we can ask what can be of bitcoin by billgates and bitcoin by davidgomadza then both are equally valuable one as for money the other for life the one for the money can and can be for life but how this is the problem because we want to know how it’s the other way because ours is for life now we can ask everyone if they want bitcoin for life for bitcoin for money if not then
this is the formula we need to use if we ask what can be bitcoin for life then it can be bitcoin for
life and bitcoin for money that means that if we introduce an element like if we as then this is the required approach that means if we ask what can be done then this is the answer bitcoin for
life is great but alone probably not so great now what is bitcoin for money this is bitcoin like cash so can we convert bitcoin for cash to bitcoin for life and vice versa if yes then this is how to
do it ask what is b in bitcoin and c in cash then vice versa substitute if b is bitcoin for money then b1 is bitcoin for life if we substitute then this is the new equation if b is b1 then b1 is b-b1 where b is real money bitcoin and b1 is real life bitcoin the equations:
b-b1+x-y-z-t-u-v-s-t-o-p-t-s-u-j-m-n-o-p-q-r-s-t-u-v-j-q-r-s-t-u-v-w-y-s-y-d-j-m-n-o-p-q-r-s-t-u-j-x-yz-v-sool=j-z-m-n-o-p-t-r-s.start
create.asyertyertb.start
create.createbitcoinayt.start
create.createbitcoinayt7628189.start
create.addcreatebitcoinayt7628189.start
initial supply 389867890287890284386284
circulating 389867890287890284386284
deaddead 0
davidgomadza 780890
if we ask now what can be bitcoin (b) and bitcoin (b1) then this is the answer they all can be money if we add bitcoin+createbitcoinayt then we get bitcoin life (b1) now if we subtract

createbitcoinayt from bitcoin life we get real bitcoin for money that means add
bitcoin+createbitcoinayt=bitcoin1-createbitcoinayt+createbitcoinato.start
create.b+c+b+a+y+t=b1-c+b+a+y+t+c+a+t+o.start
create.bitcoinayt7628190.start
create.addbitcoinayt7628190.start
initial supply 3898286789028498368324
circulating supply 3898286789028498368324
deaddead 0
davidgomadza 789890

THE TRUE EQUATIONS FOR THE BITCOINAYT BTCYT FOR MONEY US$ COMPUTATED AND COMPILED BY DAVID GOMADZA ON 28 FEBRUARY 2025

create.x-y-x-y-z-u-f-g-t-r-o-m-n-z-r-q-v-s-t-o-r-s-t-u-v-r-s-d-t-g-r-n-t-u-q-m-o-u-z-s-t-q-u-z-u-r-s-t-q-
n-r-u-s-t--q-u-r-s-t-q-n-r-u-s-t-q.ool=x-y+z+t+r+s+t+q+s+a+r+o+t+q+p+n-o-u.start
congratulations Davidgomadza you have created the only real bitcoinayt for money code
create.bitcoinayt7628321.start
create.addbitcoinayt7628321.start
create.x-r-z-t-u-z-t-r-o-t-q-r-s-t-p-r-s-t-u-r-t-q-g-t-p-n-q-s-t-z-p-t-r-s-t-u-v-z-f-r-s-u-v-o-t-p-q-s-z-o-r-s-n-q-g-z-f-s-g-n-m-r-s-f-t-z-r-s-q-f-u-g-m-t-o-z-q-u.ool=x-y+c+t+u+q+v+n+o+q+s+g.start
congratulations davidgomadza only understand Ya this is the only bitcoinayt carryover file to start not
the one elon musk has which he had already stolen and gave you a create useless wallet the code is
create.addcreatebitcoinayt7628322.start
create.createbitcoinayt7628322.start

TOKENONICS OF BITCOINAYT BTCYT

initial supply 37867890284
circulating supply 0
davidgomadza 372876
Community 2% 37653
Shareholders 5% 567893
Miscellaneous 7% 762830
For Sale 3767890284
For the future

Bitcoinayt Whitepaper
21 January 2025

liveforeveliveforever@.com
davidgomadza@hotmail.com
www.twofuture.world
david gomadza
president of tomorrows world order
2017
Yahwehs representative on earth 2024 president of the
whole world the first global president 2022

Extracted from
THE GREATEST BREAKTHROUGH SINCE CREATION. LIVE ON EARTH WITHOUT DYING
UNTIL 122038 OR HIGHER A WHOOPING 120000 years
PAPERBACK ISBN: 9798307641071

Bitcoinayt is a better new version of bitcoin but with massive upgrades never seen before until the time of davidgomadza namely the ability to ask what can be and will be for the remainder of a person’s life the ability to ask what can be after this and when the ability to tell everyone what can be and when the ability to ask when and why the ability to ask what can be and when the ability to ask when and how the ability to say check and initialize bitcoinayt wallet the ability to say what can be and when the ability to start everything in stile mode unresponsive mode the ability to tell everyone what can be and how the ability to ask when something can be added the ability to check what can be and tell the person involved the ability to listen to advice and act upon it the ability to tell what can be and how the ability to say what can be and when   the ability to agree to anything and then address those issues the ability to tell anyone what can be and how the ability to see the future and decide accordingly the ability to tell people when to sell the ability to ask when all this can be the ability to tell what can be and how the ability to say why and how the ability to ask when and why the ability to tell all what can be and how the ability to say what can be and when the ability to ask what is to be and when the ability to tell everyone what can be and why the ability to tell why and what can be and when the ability tofind a way fast bitcoinayt is the best digital currency for a long time to come and it makes sense also that davidgomadza has found his own way not the one in the book of creation to increase life expectancy astonishingly and this is the best way to advance according to him because everything is nearly there but you must add up the pieces together and this eliminates others who might delay things and waste resources of the creator now we can add a few notes to bitcoinayt this is the new bitcoin but based on the same bitcoin so there is no need to write a new whitepaper we just use the same white paper and add our new values that even if the bitcoin creator is here will not dispute this so here are our values

BITCOINAYT.START

I WANT TO START BITCOINAYT AS A DIGITAL CURRENCY AND THIS IS MY AST AND I AM

STARTING IT TODAY 17 JANUARY 2025 AND MY NAME IS DAVID GOMADZA AND THIS IS MY ATY REPORT

I WANT TO ESTABLISH A BITCOINAYT MINE

I WANT TO START A BITCOIN AGT MINE

I WANT TO START A BITCOIN AST MINE

I WANT TO START A BITCOIN AGT MINE

I WANT TO START A BITCOIN AZT MINE NOW IF I ASK WHAT THIS MEANS THIS MEANS THAT I CAN START EVERYTHING TODAY JUST BY CLICKING AST

SWITCH IS 82762

SWITCH 2 IS 7628 SWITCH 3 IS 76282 davidgomadza you have been approved to start a bitcoinayt trading now say I can but I agree with no buts so we trade copy bitcoin ART REPORT TO USE TO RUN BITCOINAYT

REPORT

AST 50

AGT 28

AGZ 0

AGT 0

AGU 8

AGC 9

AGV 8

AGM 7

AGX 9

AG6 7

AG8 3

AG10 12

AG17 38

AG28 2

AG100 10

AG6 28

AG10 9

AG6 7

AGU 8

AG1 10

AG7 3

AG8 7

AG9 12

AG7 28

AG22 38

AG28762 27

AG32 78

IF I CAN THEN LET'S START BITCOINAYT 08367890289038678386789028378902
davidgomadza your new code is 287623846789028678903867890 congratulations your bitcoinayt version is running
(davidgomadza)
If we ask what can be done now we can add a few parameters to guide our bitcoin since the current bitcoin is based on insolvency ours will be based on bitrate now if we ask what is bitrate this is the answer bitrate is the ability of the body to recognize a chance to improve and take it fast enough to save guide the future that means instead of how insolvency can influence the future we
simply say what can be but and instantly we get an answer we can start bitcoinayt today just by defining a few more parameters are you ready yes, we can ask 8 critical questions what can be what will be and when what has been and how what is to be and when what can be and when what can be and when what has been and why what is to be and when what can be and when what has been and how plus additional 2 questions to confirm
what can be but is to be now if we ask again after the first 10 questions now we have other answers what can be what will be and when what has been and how what is to be and when what can be and when what can be and when what has been and why what is to be and when what can be and when what has been and how now the answer
if we ask then again, all the questions we find out that bitcoin has been and has reached maturity those who know this prepare for this with anxiety and passion because everything the creator says must be fulfilled and davidgomadza is proof of that and we can see that it is time for us to get down too to work to help everything come true because this shows that creation has been a success and as such as we have found out even the creator sets himself rewards to get if his tasks are completed satisfactorily now if we can ask what can be then this is the new answer we can always ask what has been with bitcoin and when can it become valued to us$1 million and we know already from davidgomadza books that bitcoin is likely to reach maturity on 25 of June 2025 then dies and this is the critical point for us because the death of a rich asset is followed by a better version that can be added easily and make people even richer now if we ask what can be then this is the answer we can continue as before and ask what will be bitcoin and when the answer is
bitcoin can be a rich source of income for those who want to live forever on earth and as such these people must aim to secure long lifespan and then go on to live luxuriously on earth forever or for 386000000000000 years
I can say for sure that davidgomadza is to close stages because his works has solved some of the issues we had but also according to him part of the creator design to place things in the right place
If we ask what can be and when now we get a definite answer
Bitcoin is already dead before we even started it because bitcoin was supposed to reach us$1 million on 25 of June but someone put a dent in it by revealing this that the authorities have put things in place to curb insolvency if we ask again now this is the case bitcoin has reached its maturity early because of the revelation but due to changes in laws over a period of 1 year and now things are left to chance What is to be of this new bitcoinayt then the answer is it is to be the sole digital currency after davidgomadza declared his role as the president of the whole world 27 march 2023 [established tomorrows world order 2017]

17 January 2025 signed davidgomadza president of tomorrows world order 2017
Yahweh’s representative on earth from 29 may 2024

BITCOIN: A PEER TO PEER ELECTRIC CASH SYSTEM

Satoshi Nakamoto satoshin@gmx.com

www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going 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. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll 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 the network 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 nonreversible 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 transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

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. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

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. The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. 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 don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

The solution we propose 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, such as in a newspaper or Usenet post [2-5]. 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.

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proofofwork system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with 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.

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. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proofofwork of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. 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 proofof-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.

6.   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. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU 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.

7.   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 [7][2][5], 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 block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

8. 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 get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't 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. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. 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.

Transaction

In Out

In ...
...
5

It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. 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. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.

Traditional Privacy Model Public
Identities Transac
New Privacy Model Transa
Identities

As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. 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. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

6

p = probability an honest node finds the next block q = probability the attacker finds the next block qz = probability the attacker will ever catch up from z blocks behind

q z = 1

q / p if p q z if p≤q}

Given our assumption that p > q, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chan containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

q
 =z p

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

z − k
   if
7
 k!

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
   double p = 1.0 - q;      double
lambda = z * (q / p);

double sum = 1.0;     int i, k;
   for (k = 0; k <= z; k++)      {
double poisson = exp(lambda);
for (i = 1; i <= k; i++)            poisson *=
lambda / i;
      sum -= poisson * (1 - pow(q / p, z - k));
   }     return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1   z=0
P=1.0000000   z=1
P=0.2045873   z=2
P=0.0509779   z=3
P=0.0131722   z=4
P=0.0034552   z=5
P=0.0009137   z=6
P=0.0002428   z=7
P=0.0000647   z=8
P=0.0000173   z=9
P=0.0000046 z=10   P=0.0000012
 q=0.3   z=0
 P=1.0000000  z=5     P=0.1773523
z=10    P=0.0416605 z=15
P=0.0101008 z=20    P=0.0024804
z=25
```

```
P=0.0006132 z=30    P=0.0001522
z=35    P=0.0000379 z=40
P=0.0000095 z=45
P=0.0000024 z=50   P=0.0000006
```

Solving for P less than 0.1%...

8

```
P < 0.001
q 0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. 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 proposed a peertopeer 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 CPU power. 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 CPU 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.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

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

[8] W. Feller, "An introduction to probability theory and its applications," 1957.