# 4A Coin: A web-based cryptocurrency network.

Yasin Aktimur

***Summary~***

***Because the first generation cryptocurrencies such as Bitcoin, Ethereum and Litecoin distribute the money issuing task according to the power of machines such as GPU and CPU,***

***they accumulate the payments in the mempool and they mine by changing the nonce number randomly or by continuously increasing it and trying repeatedly until they find a hash summary that matches the difficulty they specify, instead of processing the payments made in real-time instantly. This solution is ingenious if you want to use digital currency as a value storage tool.***

***But when you try to use these digital currencies to pay for coffee or purchase a product online, it stands out as a disadvantage since you have to pay commissions to those miners and have to wait too long.***

***As you know, the power of large payment solutions, such as Paypal or Western Union, comes from their servers and software.***

***We also started working on a non-centralized, blockchain-protected end-to-end (P2P) cryptocurrency that works as a web service. In such a case, for the system to survive and ensure security, instead of paying for the server cost, we set up a system where servers, that are nodes in the system, can get paid, provided they just stay online for 44 hours instead of mining. People get paid for a useful service they provide instead of wasting their time. There will be a reason for people to install this system on their servers because they are rewarded as long as they remain as servers.***

***When people send an end-to-end payment, the servers, by finding a summary of this payment (They find this summary in order for all systems to meet at a common point and validate payments, and because it is a necessity of the blockchain architecture), ending with 4A, including the time, the sender, the receiver, amount, summary of the previous transaction and the data contained in a digital signature indicating that the sender is actually sending it, without any difficulty and they add it to their databases. Because there are no such concepts as block size or mining involved in any way, the registration process is completed instantly. We use a task queue management library called Celery Project, which is used by companies such as Instagram and Mozilla to prevent complication that may occur if 100 transactions take place within the same second.***

***Security~***

***If you noticed when you entered the site, I told you that it was safer than Visa.***

***This may seem an ambitious sentence for many, but if you actually know what you're doing when you buy a product online, you are aware of the risks involved.***

***Risk & Problem~***

***When you type information in the payment form, such as credit card, CVV, and so on, malicious people on the web can read the data you send through these forms. Just because of this, payment solutions require you to add an SSL certificates to your site. These SSL certificates actually save your data using cryptography and protect it until it reaches the recipient. Well, what if the seller has bad intentions?***

***Solution~***

***4A Coin encrypts your payments using the Elliptic Curve Digital Signature Algorithm (ECDSA) and sends a signature confirming that you want to pay instead of giving your digital password to the network. This is actually the idea of Satoshi Nakamoto and it is known as P2PKH in Bitcoin. So if we actually compare 4A Coin to a credit card, then no one will ever see your credit card details. In addition, SSL certificate is no longer needed because a malicious user who can tap into your network seeing your payment signature won't change anything, besides it is already publicly shared everywhere. Thus, we have already gotten rid of certificate issues like security and SSL in the first place.***

***Commission Solution~***

***When you benefit from a local payment solution, you will have to pay a commission to the companies you work with, between 5% and 20% of your sales. The coins you use in 4A Coin are already generated by the nodes, so they do not ask for a commission.***

***Thanks to this, you do not pay commissions for transactions nor for any other reason.***

***Wallets ~***

***wallet_id ~***
***4A01eaedb37fc09fdb94c6d632adf9f63d***

private_key ~

cbc949239a333559f5dd8b0b5cf3d32923c2cab37c2bde9c8042a3dafe59a6b9

Your wallet is actually an ECDSA key pair. At first, we used RSA for this, but we had to switch to ECDAS because the keys were too long in the RSA and were taking too much space.

Users have public keys, private keys, and wallet addresses created by processing public keys. Public key is not visible to users on the system. Instead, they will see a short version of a summary generated by public key processing as their wallet.

```
def generate_wallet_from_pkey(public_key):

    binmnmn = public_key.encode('utf-8')

    first_step = 34 - len(settings.CURRENCY)

    wallet_id = hashlib.sha256(binmnmn).hexdigest()

    wallet_id = wallet_id[-first_step:]

    wallet_id = "".join((settings.CURRENCY, wallet_id))

    return wallet_id
```

This function simply creates a wallet from the simply given public key's SHA-256 summary. Public keys are required for the approval of digital signatures, so it is necessary to keep the public key in transactions.

Payments~

When a user makes a payment, the payment time (in epoch format & GMT), the sender's address, the recipient's address, the summary of the previous transaction and the amount sent are transferred to a dictionary. This dictionary can be sorted differently on different computers, and to prevent this, the contents of this dictionary must be organized in a way that is ordered from A to Z, and will give the same result in everyone.

```
data = collections.OrderedDict(sorted(data.items()))
```

With the above code, we can create a stable dictionary that can work globally.
Finally, we take a summary of this dictionary and record it in our database, and broadcast it to other servers that we have recorded it.

*Peer to Peer~*

*We use the TCP port and web socket technology to ensure that the system is P2P. For the script to work, you need to use Python3. Because, we use the Twister Matrix Library and the Autobahn Python libraries to provide real time transactions. The port we use globally is the 9000th port.*

*There are 3 different types of broadcasting in real time processes: The first one is "Hi, I'm a new node, please add me to your network" and the other one is "Hi, I'm a new process, please verify me". This is parsed on the server side and necessary actions are taken. The third broadcasting type will be explained in the proof of cloud section.*

*Mining ~*

*A total of 450 million of 4A Coins will be issued with 300.000.000 Pre-mined. 150 million coins will be mined by a method*

*called Proof of Cloud.*

*Proof of Cloud~*

*Proof of Cloud or POC refers to a method of earning based on time as a server instead of mining. Each node, by staying online for 44 hours, will send a message as "I have been online for 44 hours, so check my database to see if I have been online by reviewing my recent transactions." If it proves that you have approved transactions for the last 44 hours, you will be eligible to receive the reward.*

*Celery & Redis~*

*We use Celery, a library that automatically controls tasks so that certain processes can be repeated at certain times in the system. Celery needs Redis to work. Redis is an open source NoSQL*

*(NoSQL is the name given to database systems that store "non-relational" data schematically. NoSQL is literally being used in the sense of "not-only- SQL", which means "only SQL is not used".) software written on Linux as the pure version.*

*Keeping the system up and running~*

*We use Gunicorn 'Green Unicorn' and Nginx, the Python WSI HTTP Server, to ensure that the system can handle high load and maintain its endurance; Nginx is a Web server designed to focus on high concurrency, high performance and low memory usage. It can also be used as a reverse proxy server, load balancer, and HTTP cache.*

*We were actually using Supervisord to keep all these systems up and running, but since Supervisord did not work with Python3 we started using Circusd developed by the Mozilla Foundation and we were more satisfied with it than Supervisiord. It is much easier to install and use, requires Tornado framework to work on its own, and works correctly with the 4.5.3 version of Tornado. This version is already installed automatically in requirements.txt, but it is important that you know this detail.*