DNATIX

DNAtix DevOps Whitepaper

Transferring Data through an Ethereum Blockchain using Transactions

DNATIX

DNAtix Development Team – Internal Research

The DNAtix development team has conducted an internal test in order to perform a proof of concept and technical research on transferring DNA sequences on a blockchain.

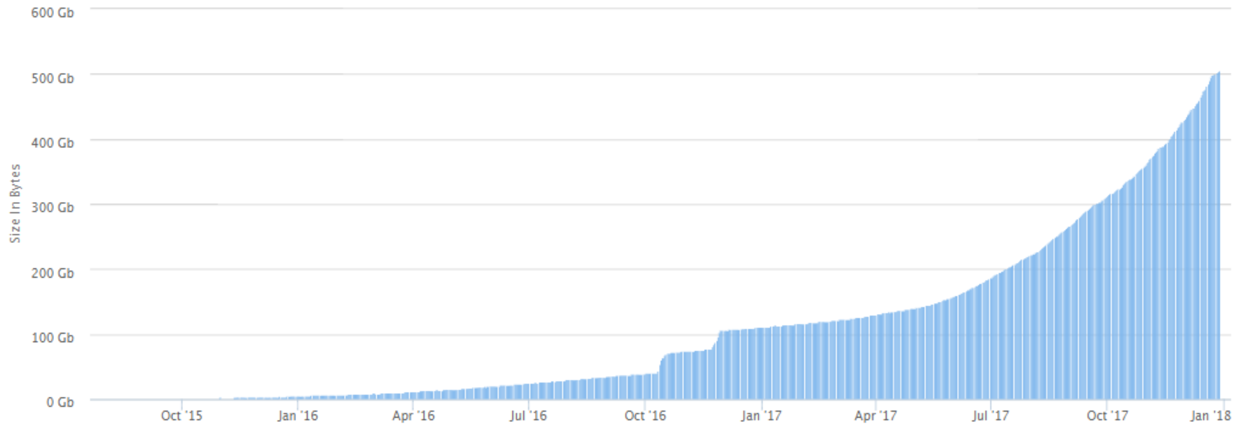
The trial test was conducted during December 2017 using the Ethereum Blockchain infrastructure to store a full DNA sequence of the smallest virus known.

As part of the research and development that is being done by the innovative development team at DNAtix, new technologies and solutions are being developed to be part of the DNAtix Blockchain eco-system.

Since the beginning of human culture, people traded valuables for exchange of money or other valuables. Since the invention of credit cards, people found a way to store large amounts of money. As time passed, the technologies to store money and spend money improved significantly but so the tools and techniques to hack those technologies.

News about hacking another "wallet" technology terrifies people across the world. In some cases, people got fired, companies went bankrupt, and money was taken from hard working people by a group of hackers. Because of these incidents, people knew exactly who to blame – the organization which wasn't cautious enough to fix the vulnerabilities the hackers exploited, and this logic is understandable – One company which holds all the passwords of all users as well as personal information regarding those users, on one or more servers, held on another server that coordinates their operations. Every server is a potential attack vector to obtain access and control over the servers.

To fix the centralized issue, the cryptocurrency was invented. We will focus Ethereum on this article. To fix the issue, Ethereum has a blockchain, which is basically a ledger. The blockchain logs all transactions of Ether and can provide the current balance of an Ethereum wallet based on its transactions history. The surprise of this blockchain is the location on which it's stored – everywhere. Every Ethereum user must store a fraction of the blockchain (as of today – more than 470GB of data).



Forcing the users be part of the network will ensure every user donates to the integrity of a leech-less network. With no specific address or target to attack, it is almost impossible to alter the blockchain. With every transaction issued and logged, the user can optionally add data to it. Imagine receiving money with a letter addended together. So, With an Ethereum transaction, money and data can change hands. Then we could theoretically send data through the Ethereum network and it will be decentralized, broken to pieces and stored on many computers so it's more secured. So far it sounds great but there are some limitations on moving data through that network:

- Maximum data length is limited and varies on every new block created.
- A fee is being paid for every Byte of data transferred through the blockchain

Before going any further let's talk about Gas. Ethereum is the network, also known as the blockchain. Ether (ETH) is the fuel for that network. When you send tokens, interact with a contract, send ETH, or do anything else on the blockchain, you must pay for that computation.

That payment is calculated in Gas and gas is paid in ETH. You are paying for the computation, regardless of whether your transaction succeeds or fails. Even if it fails, the miners must validate and execute your transaction (compute) and therefore you must pay for that computation just like you would pay for a successful transaction.

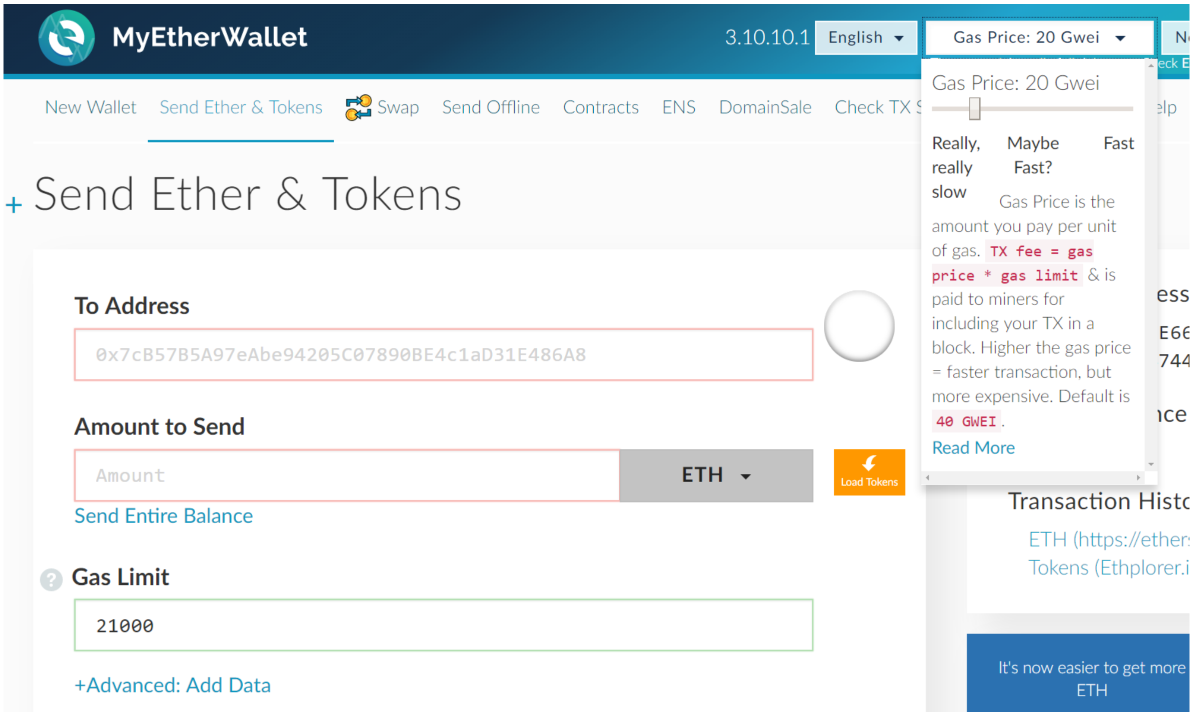
Gas is typically referring to Gas Limit or Gas Price. Because the value of ETH is too high, and gas measure unit is much smaller, then instead of referring gas as a very small fraction of ETH (which is the common range of price when referring to it) so gas is measured in Wei, which is one quintillionth of ETH (1 Wei = 10^{-18} ETH).

The total cost of a transaction (the "TX fee") is the Gas Limit * Gas Price.

You can think of the gas limit like the number of liters/gallons/units of gas for a car. You can think of the gas price as the cost of that liter/gallon/unit of gas. With a car, it's $2.50 (price) per gallon (unit). With Ethereum, it's 20 GWEI (price) per gas (unit). To fill up your car's "tank", it takes 10 gallons at $2.50 = $25. With Ethereum it takes 21000 units of gas at 20 GWEI = 0.00042 ETH. Therefore, the total TX fee will be 0.00042 Ether.

The gas limit is called the limit because it's the maximum number of units of gas you are willing to spend on a transaction. This avoids situations where there is an error somewhere in the contract, and you spend 1 ETH....10 ETH....1000 ETH..... going in circles but arriving nowhere. However, the units of gas necessary for a transaction are already defined by how much code is executed on the blockchain. If you do not want to spend as much on gas, lowering the gas limit won't help much.

You must include enough gas to cover the computational resources you use, or your transaction will fail due to an Out of Gas Error. If you want to spend less on a transaction, you can do so by lowering the amount you pay per unit of gas. The price you pay for each unit increases or decreases how quickly your transaction will be mined.



Send Ether & Tokens

To Address

0x7cB57B5A97eAbe94205C07890BE4c1aD31E486A8

Amount to Send

Amount ETH

Gas Limit

21000

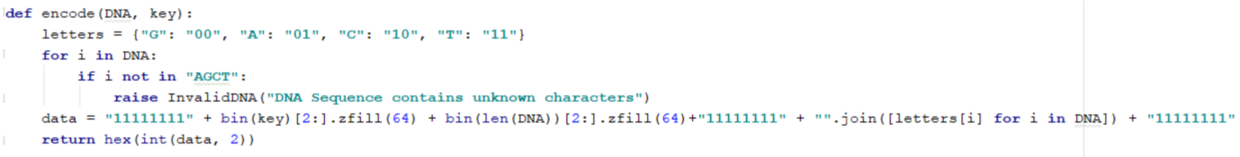
Gas Price: 20 Gwei

Gas Price is the amount you pay per unit of gas. `TX fee = gas price * gas limit` & is paid to miners for including your TX in a block. Higher the gas price = faster transaction, but more expensive. Default is `40 GWEI`.

The data (hex representation of bytes) changes the total TX fee as well.

$$(21000 + 68 * length\_of\_data\_in\_bytes) * gas\_price \ = TX\_COST$$

For our experiment, we had to find a way to send a 5390 Nucleotides long DNA sequence using a transaction while minimizing its TX fee. First, we found a way to compress and encrypt the DNA to quarter of its length and implemented that way to a Python script.



```
def encode(DNA, key):
    letters = {"G": "00", "A": "01", "C": "10", "T": "11"}
    for i in DNA:
        if i not in "AGCT":
            raise InvalidDNA("DNA Sequence contains unknown characters")
    data = "11111111" + bin(key)[2:].zfill(64) + bin(len(DNA))[2:].zfill(64)+"11111111" + "".join([letters[i] for i in DNA]) + "11111111"
    return hex(int(data, 2))
```

Using this script, we expressed the nucleotides as hex represented binary ready to be addended to the transaction.

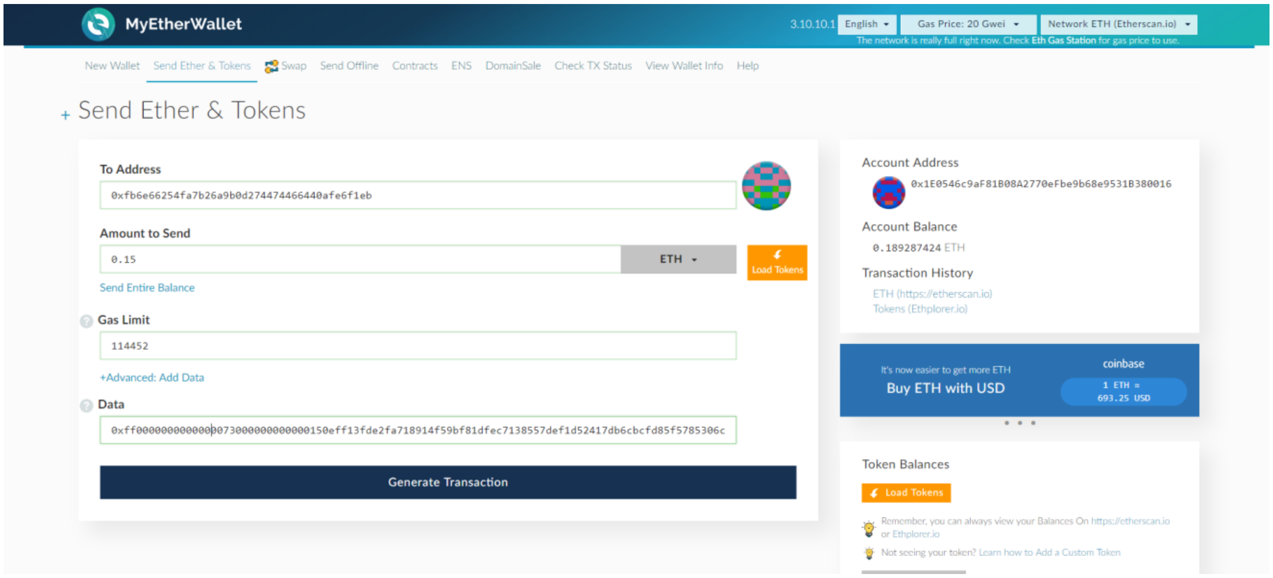
GAGTTTTATCGCTTCCATGACGCAGAAGTTAACACTTTCGGATATTTCTGATGAGTCGAAAA…

↓

0x150eff13fde2fa718914f59bf81dfec7138557def1d52417db6cbcfd8506cb0815711…

Since the blockchain is too large for a consumer to download (and very time consuming), an alternative that allow us to preform actions on the blockchain while not taking part of its gigantic blockchain, is to connect to a Remote Procedure Call (RPC) endpoint. We used etherscan.io to make the transaction. Using the easy-to-use UI and UX of MyEtherWallet.com, we imported a

Keystore of an account with about 0.2 ETH. We created another Ethereum wallet to receive the money and data.



+ Send Ether & Tokens

To Address
0xfb6e66254fa7b26a9b0d274474466440afe6f1eb

Amount to Send
0.15 ETH

Gas Limit
114452

Data
0xff000000000000073000000000000150eff13fde2fa718914f59bf81dfec7138557def1d52417db6cbcfd85f5785306c

Account Address
0x1E0546c9aF81B08A2770eFbe9b68e9531B380016

Account Balance
0.189287424 ETH

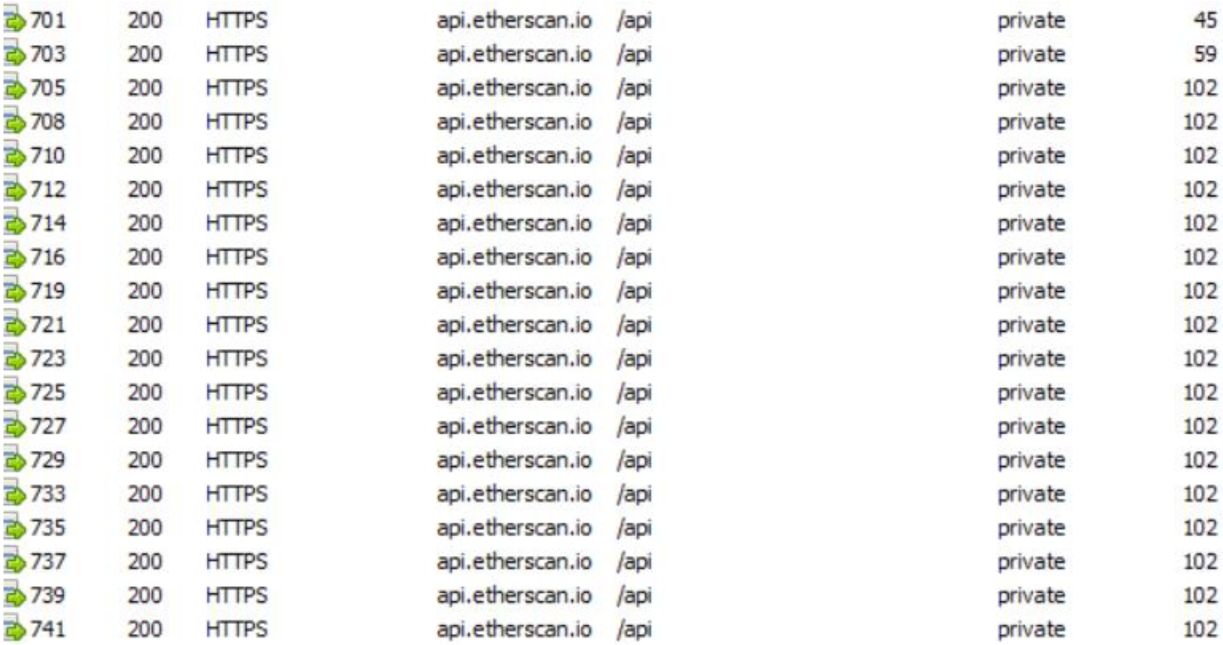
Transaction History
ETH (https://etherscan.io)
Tokens (Ethplorer.io)

Token Balances

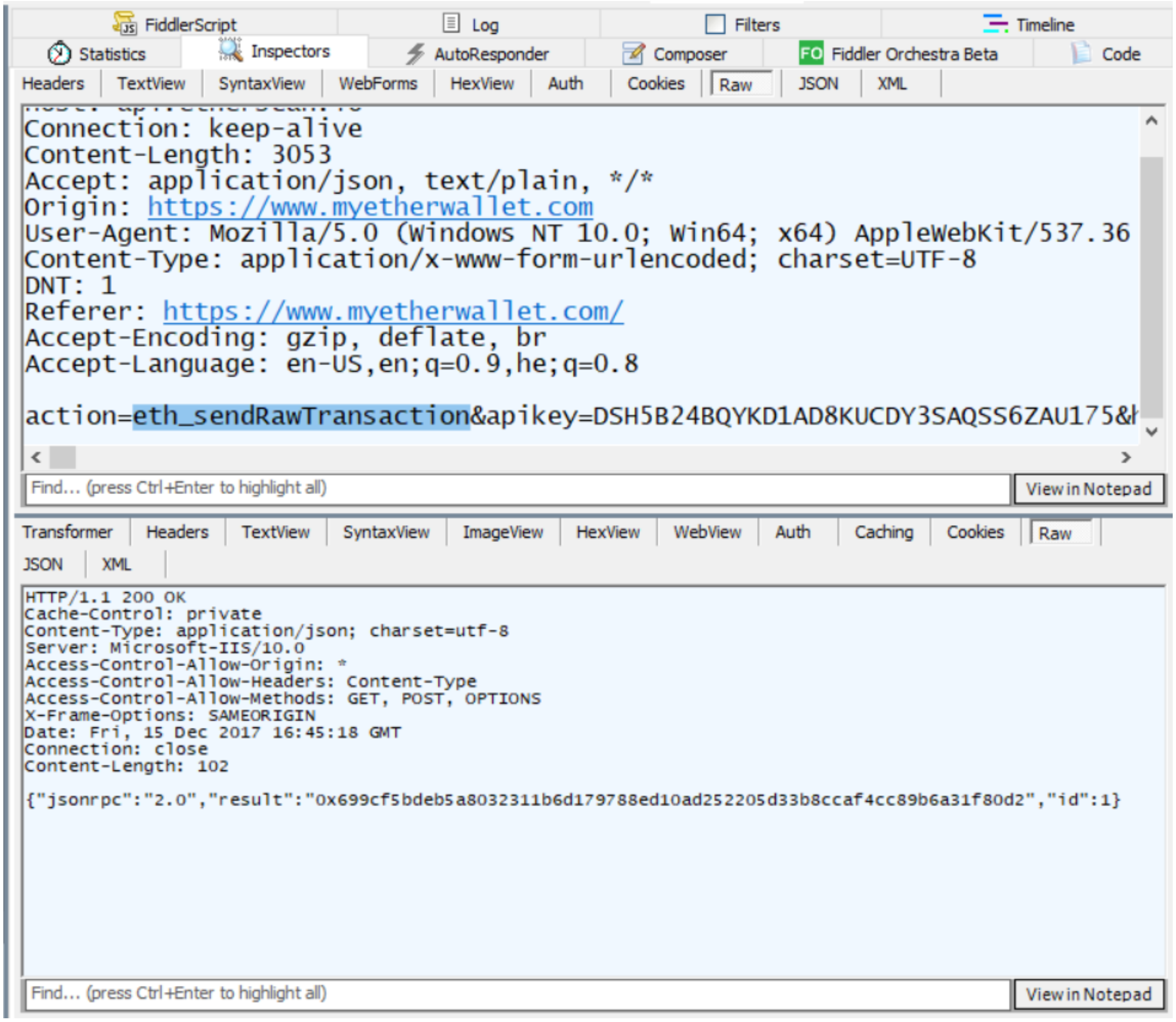
Remember, you can always view your Balances On https://etherscan.io or Ethplorer.io

Not seeing your token? Learn how to Add a Custom Token

The Website uses Etherscan's API as well - First to check balances, connect to a wallet, calculate the amount, estimate the gas required to successfully pull off the transaction, and queue the request to the blockchain.



701	200	HTTPS	api.etherscan.io	/api	private	45
703	200	HTTPS	api.etherscan.io	/api	private	59
705	200	HTTPS	api.etherscan.io	/api	private	102
708	200	HTTPS	api.etherscan.io	/api	private	102
710	200	HTTPS	api.etherscan.io	/api	private	102
712	200	HTTPS	api.etherscan.io	/api	private	102
714	200	HTTPS	api.etherscan.io	/api	private	102
716	200	HTTPS	api.etherscan.io	/api	private	102
719	200	HTTPS	api.etherscan.io	/api	private	102
721	200	HTTPS	api.etherscan.io	/api	private	102
723	200	HTTPS	api.etherscan.io	/api	private	102
725	200	HTTPS	api.etherscan.io	/api	private	102
727	200	HTTPS	api.etherscan.io	/api	private	102
729	200	HTTPS	api.etherscan.io	/api	private	102
733	200	HTTPS	api.etherscan.io	/api	private	102
735	200	HTTPS	api.etherscan.io	/api	private	102
737	200	HTTPS	api.etherscan.io	/api	private	102
739	200	HTTPS	api.etherscan.io	/api	private	102
741	200	HTTPS	api.etherscan.io	/api	private	102



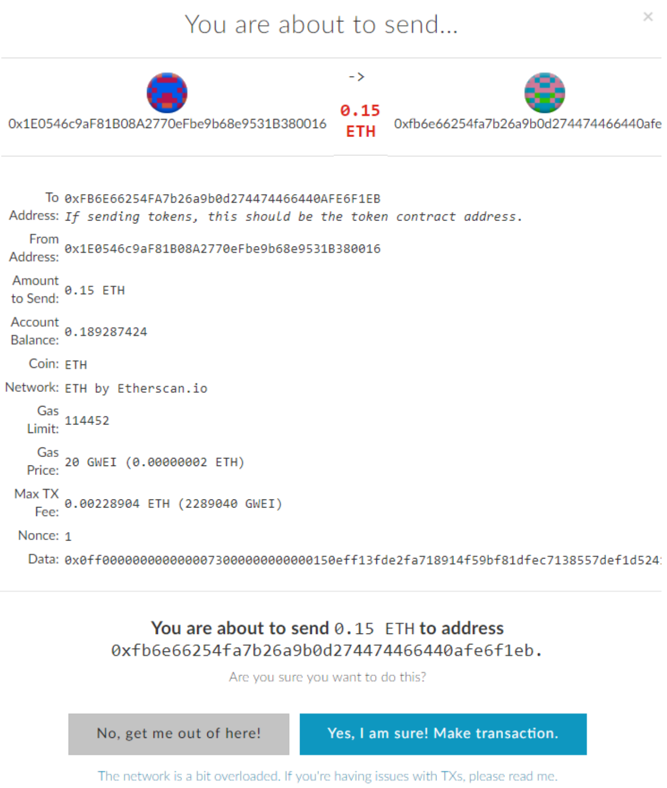
```
Connection: keep-alive
Content-Length: 3053
Accept: application/json, text/plain, */*
Origin: https://www.myetherwallet.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
DNT: 1
Referer: https://www.myetherwallet.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,he;q=0.8

action=eth_sendRawTransaction&apikey=DSH5B24BQYKD1AD8KUCDY3SAQSS6ZAU175&
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/10.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
Access-Control-Allow-Methods: GET, POST, OPTIONS
X-Frame-Options: SAMEORIGIN
Date: Fri, 15 Dec 2017 16:45:18 GMT
Connection: close
Content-Length: 102

{"jsonrpc":"2.0","result":"0x699cf5bdeb5a8032311b6d179788ed10ad252205d33b8ccaf4cc89b6a31f80d2","id":1}
```

We estimated the gas limit using the function above and added more gas to have a leeway and prevent Out of Gas Error. We sent most of the account's balance with the data of the virus.



You are about to send...

0x1E0546c9aF81B08A2770eFbe9b68e9531B380016 -> 0.15 ETH 0xfb6e66254fa7b26a9b0d274474466440afe

To Address: 0xFB6E66254FA7b26a9b0d274474466440AFE6F1EB
If sending tokens, this should be the token contract address.

From Address: 0x1E0546c9aF81B08A2770eFbe9b68e9531B380016

Amount to Send: 0.15 ETH

Account Balance: 0.189287424

Coin: ETH

Network: ETH by Etherscan.io

Gas Limit: 114452

Gas Price: 20 GWEI (0.00000002 ETH)

Max TX Fee: 0.00228904 ETH (2289040 GWEI)

Nonce: 1

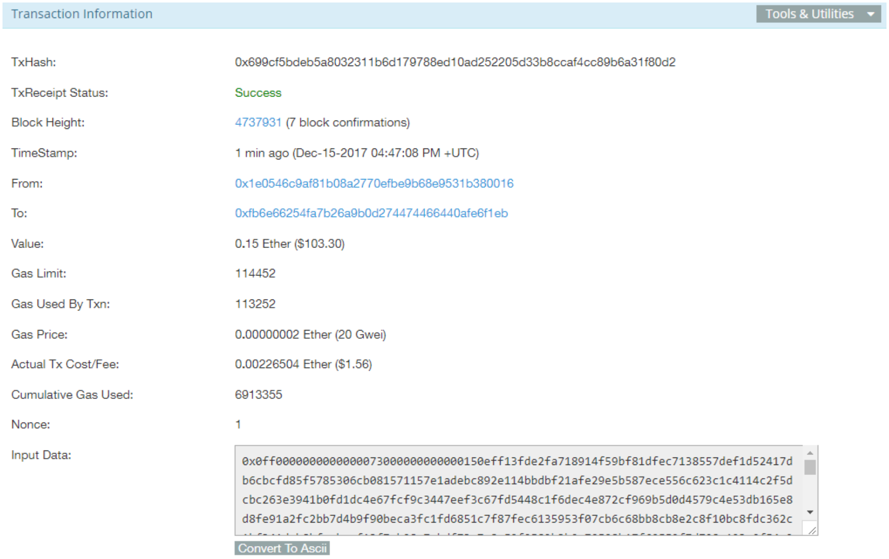
Data: 0x0ff000000000000000730000000000000150eff13fde2fa718914f59bf81dfec7138557def1d524

You are about to send 0.15 ETH **to address** 0xfb6e66254fa7b26a9b0d274474466440afe6f1eb.

Are you sure you want to do this?

The network is a bit overloaded. If you're having issues with TXs, please read me.

When the transaction was preformed, we waited and received a TX Hash on the blockchain. Once TX Hash is generated, EVERYONE has access to the data we sent.



Transaction Information

TxHash: 0x699cf5bdeb5a8032311b6d179788ed10ad252205d33b8ccaf4cc89b6a31f80d2

TxReceipt Status: Success

Block Height: 4737931 (7 block confirmations)

TimeStamp: 1 min ago (Dec-15-2017 04:47:08 PM +UTC)

From: 0x1e0546c9af81b08a2770efbe9b68e9531b380016

To: 0xfb6e66254fa7b26a9b0d274474466440afe6f1eb

Value: 0.15 Ether ($103.30)

Gas Limit: 114452

Gas Used By Txn: 113252

Gas Price: 0.00000002 Ether (20 Gwei)

Actual Tx Cost/Fee: 0.00226504 Ether ($1.56)

Cumulative Gas Used: 6913355

Nonce: 1

Input Data:

```
0x0ff000000000000000730000000000000150eff13fde2fa718914f59bf81dfec7138557def1d52417d
b6cbcfd85f5785306cb081571157e1adebc892e114bbdbf21afe29e5b587ece556c623c1c4114c2f5d
cbc263e3941b0fd1dc4e67fcf9c3447eef3c67fd5448c1f6dec4e872cf969b5d0d4579c4e53db165e8
d8fe91a2fc2bb7d4b9f90beca3fc1fd6851c7f87fec6135953f07cb6c68bb8cb8e2c8f10bc8fdc362c
```

An important note regarding posting data on the blockchain: The data on the blockchain is public and can be seen by everyone. It’s crucial to encrypt your data before sending it through the Blockchain.