Dimecash: A Language Coin

John Rigler
john@rigler.org

Abstract. A fork of litecoin which puts special restriction on how proof-of-burn operates by creating a new token and awarding it to an address for burning one entire coin. The burning of currency is not something specially coded into the blockchain software, but rather a happy coincidence of visual cognition and mathematics. Since blockchain transactions are inherently irreversible, an erroneous transfer to an Obviously Unspendable Crypto-Currency Address (OUCA) would normally be permanent. By creating the new token, this problem is eliminated. Advanced Satoshi Codes (ASC) are used to override this behavior when pure burning for financial regulation is required. The currency can safely begin to function as linguistic ledger pages. The mechanism of burning bitcoin is built upon the mathematical truism that an obviously readable address could never have been hashed from a valid private key.

This system greatly increases the use of unspendable addresses and normally danger exits that a large sum of money could be accidentally burned in a system that draws such attention to these addresses (especially when their readable message is anything other than a reference to the actual burn process). Once an unspendable address has been rendered onto the ledger, it will have an associated first transaction. This transaction owner might be the intended recipient of the transaction. An affinity between this address and the unspendable address could then be created and manipulated via various systems. Because Advanced Satoshi Codes are quite easy to send, a series of transactions could be presented with a revealed off-chain secret to legally argue intent. The process of creating the secrets is programmatically repeatable and is linted by the shell when it is typed in. It need not ever be written to disk, because the entire process can be done in the shell environment and printed off. For that matter, a secret could be coded on a typewriter or hand-written.

Introduction.

In this paper, I suggest a radically new, and elegantly simple way to use a cryptocurrency as a ledger of meaning for those who are privy to secret files or codes. The files are very small text files that can be read, but are also contain text that is constrained in certain ways so that they could be printed off, photographed, or hand-written and still could very likely be reproduced exactly the same way.

Two weak hashes are proposed. Over time the weak hashes of these files would collide with each other, but since they are shared offline or in a limited way, collision is very improbably and could visually be resolved.

The short codes are then either used as part or all of a cryptocurrency transaction. Later, a savvy reader, armed with even a paper copy of the codes, could make sense of a conversation carried out across the blockchain. I believe that a jury could also do this sense making and thus the secrets may be admissible in court. The codes could also be used as a sort of moveable history of successful actions.

Advances Satoshi Codes.

The original proposal for Satoshi Codes only consisted of two digits:

https://en.bitcoinwiki.org/wiki/Satoshi_codes

Because alt-coins can be very inexpensive and still function quickly and safely, an eight-digit code would still be a trivial amount of money.

100.00000000 ← This transaction has no Satoshi Code
432.00000013 ← This Satoshi Code says "Good luck/Please check our mail"

The code 13 modifies the transaction of 432 tokens. It is still paid, but is far less than even most mining fees, and is simply financially inconsequential.

The following code, however, implies location:

0.00070601 ← This is a ZIP code in Lake Charles, LA

By itself, the fact that this is a ZIP code could be a coincidence, but if it is seen as a pattern of other ZIP codes, then such an explanation is more likely.

Obviously Unspendable Addresses.

By using a python script, I am able calculate the following unspendable address:

DSENDxYoURxZiPCoDExASxFUNDSxydsw7y

Now the argument the codes are simply ZIP codes is very strong. Someone would have to obscure the central mechanism of this system to hide the fact that a request is being made:

SEND YoUR ZiP CoDE AS FUNDS

Upon these relatively simply hacks, all sorts of meaning can be attributed to the ledger.

Weak Hashes.

Cryptocurrency is bases on very strong hashes, which are really very long numbers often expressed in a system of letters and numbers (some are removed because they are too similar to the numbers zero and one). In unix, the shasum command can create these hashes, but a very old command was used for something similar in the early days of unix to describe files. A strong has can't easily be read or memorized, but weak ones can.

Bash Functions.

By typing messages into a bash function format, they can be loaded into the shell and reduced to a sum hash with the declare function:

```
declare -f This.Is.A.Test
This.Is.A.Test ()
{
   : this is only a test
}
```

The body of the function will be formatted perfectly once it is played back in this way, so even if it was typed in with hidden spaces, it will always resolve to the same thing:

```
declare -f This.Is.A.Test | sum
51986 1
```

The secret of the function is 51986. The function can be written to a file:

This.Is.A.Test-51986

And the file can be shared. The received could check that that the function and the sum match and discard it if they don't. Both sender and receiver might then view an 5-digit (sometimes 3 or 4) AST as one of these secrets. When revealed, its meaning might be obvious. It will be very difficult to create a fake bash function that resolves to 51986 and conveys any sort of meaning at all, much less meaning that makes sense in a larger context.

Glossary:

Obviously Unspendable Crypto-currency Address (OUCA) –

When viewed in a ledger, an obviously unspendable address will show a pattern that statistically could not be randomly arrived at. A valid cryptocurrency address contains a few leading characters, a relatively long hash, and then six characters that complete a formula. Unspendable addresses are not created this way, but simply made mathematically complete by calculating the last segment. Obviously unspendable addresses are visually identifiable because they can contain messages, such as:

DDogepartyxxxxxxxxxxxxxxxxxxxw1dfzr

In contrast, an address like this would not be obviously unspendable:

DNA4N57QCoJBvvXAxidG6zQYb3cVTyFLrV

By checking the ledger, we see that the second address can be confirmed to be spendable as it has been used to transfer Dogecoin to the first address. However, what makes it spendable is that the last characters (TyFLrV) are calculated from the first part. Were someone to randomly enter the correct characters and numbers, then an address would be create that would be unspendable in reality, but would not appear as such. This is dangerous. More dangerous though is that a trickster could create an address requesting funds as a very special sort of financial attack. The funds would be destroyed, but a person who did not understand cryptocurrency could easily be fooled into believing that they could be used for what appears to be a stated cause. One such example is:

DFiNiSHTHEWALLXXXXXXXXXXXXXXbFELgC

It is not hard to imagine that an enthusiastic supporter of a US Southern Border Wall could easily be duped into sending funds to the above readable address. Thousands of years of powerful processing would be needed to stumble upon an address that makes any sense, much less a partisan political message.

Advanced Satoshi Codes (ASC)

Satoshi Codes are added to the end of transactions to express meaning. Only two-digits were called for originally, but I propose that by using from five to eight digits, a whole slew of much more robust systems are possible.

Secret Sharing (SS)