Netcoin: A Traceable P2P Electronic Cash System

Hitesh Tewari and Eamonn O Nuallain
School of Computer Science and Statistics
Trinity College Dublin
Ireland
Email: Hitesh.Tewari@scss.tcd.ie, Eamonn.ONuallain@scss.tcd.ie

***Abstract*—This paper introduces a new P2P Electronic Cash system called Netcoin. The purpose of Netcoin is to facilitate inexpensive peer-to-peer monetary transactions on the Web. Its salient features are that it is a traceable system with an efficient mechanism for verifying transactions. Netcoins are reusable and can be easily passed from one user to another. The issuing of virtual currency and verification of transactions are performed by trusted mints, which act as the gateway between the fiat and virtual currency worlds. There is no need to maintain a public ledger, which would inhibit use on a global scale because of rapidly increasing memory and bandwidth requirements. The system is neither inflationary nor deflationary in nature and does not purport a new economic model. As a fiat-backed currency it should not suffer the volatility issues associated with Bitcoin. In this paper the two most prominent electronic payment systems of the last forty years, namely Ecash and Bitcoin, are examined. Netcoin is then introduced in detail and is designed to address the shortcomings of these payment systems.**

Keywords—Bitcoin, Ecash, Block Chain, Proof-of-Work

I. Introduction

Electronic cash gained prominence late in the twentieth century when David Chaum presented his groundbreaking research on untraceable electronic cash known as Ecash [1]. The advent of the World Wide Web in the '90s brought about the development of new electronic payment systems to support Internet commerce [2]. A number of micropayment schemes were also proposed to aid low-value transactions. However, despite a plethora of proposed schemes, none bar credit card payments over the Internet were successful. In particular, PayPal emerged as the clear forerunner to facilitate such payments.

Very little came in the form of new payment systems during the first decade of this century until the emergence of the Bitcoin protocol by "Satoshi Nakamoto" [3]. Bitcoin captured the imagination of the research community and the general public unlike any other digital currency before it. In essence, Bitcoin allowed its adopters to create a decentralized system, thereby eliminating the need for a central authority to "police the currency". In 2013 the US Senate Committee on Homeland Security posted a letter noting that "online currencies appear to be an important emerging area", and that the federal government needed to understand better the benefits and threats of such virtual currencies [4].

Although Bitcoin is a major breakthrough in the development of a decentralized electronic payment system, which for the first time bypasses the traditional banking systems, it has its own unique problems that have prevented its mass adoption. One particular aspect of the currency that is of concern is that there is an upper limit of the number of bitcoins that can be minted and this has led to the hoarding of the currency by some users [5] which in itself is inflationary. Also the Bitcoin system does not make it easy for users to buy-in or cash-out of the system. We will elaborate on the Bitcoin protocol in detail in section II-B. Another electronic transfer system called Stellar has been proposed recently [6]. The inventors describe Stellar as "public infrastructure for money". Stellar is positioning itself as a gateway between digital and fiat based currencies and hopes to greatly simplify the process of moving between the two.

In this paper we propose Netcoin, an electronic cash scheme that is inspired by the above mentioned systems and one that allows users to easily conduct peer-to-peer and traditional Internet Commerce transactions. In particular we leverage the idea of getting other participants in the network to help verify the authenticity of transactions and to reward them for their effort. One of the main advantages of the Netcoin scheme is that it greatly reduces the processing fees (typically 2-3% of the transaction costs) that are charged by financial institutions today. Netcoin also allows users to easily buy-in or cash-out of the system i.e., easily move between the virtual and fiat currency worlds. Just like fiat currencies today, Netcoins can be reused again and again as they are exchanged between users for goods and services.

II. Related Work

In the real world, financial instruments such as coins and notes are used to transact for goods and services. In order for the general public to have confidence in these instruments, security measures are required to ensure that these notes cannot be easily forged. In the digital world strings of bits are used to convey information and these can be easily stored and replicated. There is therefore a clear need to ensure that digital currency cannot be forged or double spent. Public key cryptographic techniques such as digital signatures are used to solve this problem. However, unlike fiat currencies, a recipient cannot accept a digital coin based solely on a digital signature. Additional third-party verification is required to ensure that a coin is not being double spent by its owner. Addressing this *double spending* problem is at the heart of the design of the two most prominent electronic cash schemes of the past thirty years, namely Ecash and Bitcoin. Below we describe the Ecash and Bitcoin schemes in detail that are the genesis of the Netcoin scheme we present in this paper.

A. *Ecash*

The first digital cash protocol which was truly anonymous was proposed by David Chaum [7]. The scheme was essentially an on-line software solution. A buyer would spend coins with a merchant. By examining the coins neither the issuer nor the merchant would be able to deduce the identity of the customer. The protocol was designed such that the issuer was not able detect the serial numbers of coins that it issued to users of the system, even if it colluded with the other participants in the system.

A client could withdraw digital coins from a real-world bank against an existing account. The scheme used what is referred to as the *blind signature* protocol. This protocol allowed the user to mint a number of coins and forward these unsigned coins to a bank. As long as these coins met certain criteria, the bank signed them with its private key without knowledge of the serial numbers associated with the coins. This feature allowed for truly anonymous cash. On receiving the coins back from the bank, the user removed the blinding factor and used the coins to pay for goods at any merchant participating in the system. On receipt of the coins, a merchant had to immediately forward them to the bank for verification. The bank maintained a database of the serial numbers of all coins that had been spent in the system and was thus able to detect double spending. Chaum's blind signature analogy is as follows:

- A user takes a piece of blank paper and a piece of carbon paper - inserts both into an envelope and seals the envelope.
- She goes to her bank and hands the clerk a one dollar note and asks her to stamp the sealed envelope with the banks "$1 stamp".
- She opens the stamped envelope and takes out the piece of paper which now has an impression of the bank's $1 stamp.
- The bank stamps many of these notes every day and has no idea of which note is associated with which customer.
- The customer can then go ahead and spend the stamped note with any merchant and the bank will not be able to tell which bank customer spent that note.

In the real system the "blinding factor" is a random number used to obfuscate the serial number of the electronic coin from the bank. Mathematically the RSA Blind Signature scheme can be illustrated with the following steps. Let *m* be the coin's serial number, *r* the blinding factor, *e* and *n* the bank's public key exponents. The sender raises *r* to the bank's public key exponent *e* and computes the product of the serial number and blinding factor:

$$m' \equiv mr^e (mod\ n) \tag{1}$$

The bank signs the blinded serial number with its private key:

$$s' \equiv (m')^d (mod\ n) \tag{2}$$

Returns the coin to the user who removes blinding factor:

$$s \equiv s' \cdot r^{-1} (mod\ n) \tag{3}$$

The user now has a coin signed with the bank's private key:

$$s \equiv (m')^d r^{-1} \equiv m^d r^{ed} r^{-1} \equiv m^d r r^{-1} \equiv m^d (mod\ n) \tag{4}$$

The classic example opponents regularly cite against adoption of this system was that this type of currency could be used for criminal and/or subversive activities. An example is where a kidnapper demands a ransom and asks for the "blinded serial number" which had been signed by the bank's private key to be published in a national newspaper. Only the kidnappers would be able to ascertain the serial number as they have the blinding factor. There would hence be no need for a drop-off point for cash and a much-reduced chance of the kidnappers ever getting caught!

The main drawback of the scheme was the "server centric" nature of the protocol. The bank could easily become a target of an attack or could be forced to shutdown by the government agencies if they felt that the Ecash system was a threat to their sovereign currency. Another major drawback of the scheme was that Ecash was not reusable. Once an Ecash token was used the bank would need to keep the spent serial number in an ever-growing database of spent coins, and check against the same before accepting any new coins.

B. Bitcoin

Bitcoin is a decentralized, pseudo-anonymous electronic cash scheme [3]. The Bitcoin protocol is decentralized in the sense that it eliminates the need for any central authority such as a central bank or server. In the Bitcoin system the participants in the network are collectively the bank. There is no requirement for mutual trust between users or the need to employ two-phase commit protocols to verify transactions in the network [8]. This in itself is a major shift from current banking practice and is a concept that has long-term repercussions for financial institutions and sovereign currencies alike. With a little effort i.e. by using a Bitcoin exchange, it is possible to try and hide the identity of the users of the system [9]. It is however important to note that all Bitcoin transactions are stored in a public ledger known as the *block chain* and this can be parsed by others in an attempt to obtain the identity of the users of the system.

There are two main players in the Bitcoin network, namely users with digital wallets and bitcoin miners. Anyone who wants to start using the Bitcoin system needs to download a software based digital wallet which is used to send, receive and store bitcoins. Prior to using Bitcoin, a user (e.g., a merchant) will need to generate a Bitcoin address to which bitcoins can be sent. The wallet software generates an asymmetric key pair (public & private key). A cryptographic hash of the public key is used as the merchant's Bitcoin address. Placing it prominently on the merchants web site can publicize this address. End-users can send their Bitcoin address to other users via email, or make it a part of their email signature. A hash of the public key, as opposed to the key itself, is used because a hash is much smaller (e.g., 256 bits) in length than a public key (e.g., a RSA public key is typically 2048 bits).

A user who wishes to transfer some bitcoins to another user inputs the transfer amount and instructs their wallet software to complete the transaction. The wallet initiates a transaction and identifies the recipient using their Bitcoin address. This includes the sender's address, the amount etc. The wallet software digitally signs the transaction with the private key of the sender. The transaction is then broadcast to all participants on the Bitcoin network. The transaction can then be verified by anyone using the public key of the sender to ensure that the transaction is legitimate and that the user spending the bitcoins is actually the owner of these coins.

The other main entity in the Bitcoin network are the *miners* whose main task is to try and be the first to verify the next "transaction block". A transaction block consists of all transactions that have been broadcast on the Bitcoin network in the preceding ten minutes. If successful the miner is rewarded for her effort with a small number of bitcoins. This figure started out at 50 bitcoins and was reduced to 25 bitcoins in 2012. The reward will halve approximately every 210,000 blocks or every four years until the year 2140, after which "transaction fees" will be replaced as the reward mechanism. The salient features of the Bitcoin protocol are as follows:

1) Block Chain: The block chain is a timestamped public ledger of all transactions that have ever been conducted on the Bitcoin network. The first block in the chain is known as the *genesis block*, followed by blocks that have been verified by miners. Each new block contains one or more new transactions that have been received by the miner. These are repeatedly hashed in pairs to form a Merkel Tree [14]. The root of the Merkel Tree along with the hash of the previous block is stored in the block header thereby chaining all the blocks together. This ensures that a transaction cannot be modified without modifying the block that records it and all following blocks. This property of the block chain makes double spending of bitcoins very difficult.

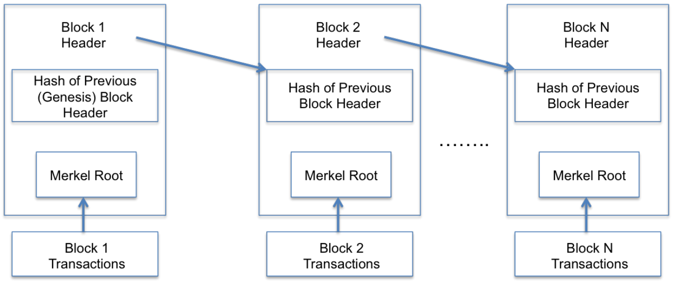


Fig. 1. Bitcoin Block Chain

2) Bitcoin Transaction: Figure 2 shows the various fields of a Bitcoin transaction where a user Alice is transferring ten satoshi (smallest fraction of a Bitcoin) to Bob. A transaction starts with a transaction identifier, which is a hash of the rest of the transaction data. The version number of the set of rules under which the transaction is to be evaluated follows this. This allows for the Bitcoin protocol rules to be updated and for different versions to coexist. The third and fourth fields specify the number of inputs and outputs for the transaction. In Figure 2 Transaction# 0 has only one input and one output. This is followed by the size of the transaction block in bytes. Finally we have the actual details for each of the inputs and outputs for this transaction. The second last entry is the only input to the transaction in question and refers to the output of a previous transaction where 100 satoshis were given to Alice. It also contains the signature of the sender as well as their public key. The last entry is the only output for this transaction and consists of the value of the amount being transferred and the address of the recipient. This last entry states Alice spent tall 100 satoshis with Bob.

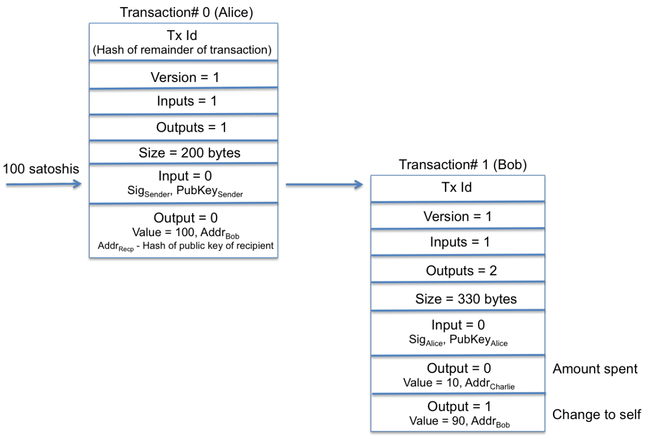


Fig. 2. Bitcoin Transaction

Each output of a transaction can only be used once. This seems like an inconvenience, as a user would require the exact amount for each transaction. However multiple outputs help us solve this problem. The second transaction has one input (100 satoshis from Alice) and two outputs. Bob spends 10 satoshis with Charlie and transfers the remainder to himself. Multiple outputs in a Bitcoin transaction allow the user to spend a fraction of the input to pay a third party and to pay the remainder to oneself as "change". So we see that in Bitcoin there are no real coins but just transaction data held within a public ledger. This in turn means that there is no requirement for serial numbers in the Bitcoin protocol.

3) Double Spending: As pointed out at the beginning of this paper, sending a digitally signed coin is not enough to convince a recipient that it is a valid coin. Suppose Alice sends a signed Bitcoin transaction to a merchant Bob. In order for Bob to fulfill the order he must be able to verify that the coin has not been spent elsewhere in the system. Bob can check his version of the block chain to see if the coin does indeed belong to Alice. However if Alice wishes to cheat she could simultaneously send the same coin to Charlie who will incorrectly assume that the coin has not been spent elsewhere. In order to prevent such a situation from occurring, Bob broadcasts the transaction to all other participants in the network and asks for their assistance in verifying the transaction.

4) Proof-of-Work: The problem with the above "collective verification" approach is that Alice could still cheat Bob by introducing a large number of nodes into the network that are controlled by her. These nodes could then all reply back to Bob saying that the coin given to him by Alice was legitimate and trick Bob into accepting the transaction. To prevent this from happening the Bitcoin protocol employs the Hashcash "Proof-of-Work" concept [10].

The Hashcash system was proposed by Adam Black in 1997 as a means to prevent spammers from sending large amounts of unsolicited emails to users on the Internet. In the Hashcash system the sender of an email is required to create a "hashcash stamp" using a hash function. A cryptographic hash function (e.g., SHA-256) takes an arbitrary length input and produces a fixed length output. Hash functions are designed to be collision resistant i.e., it is computationally hard to find two inputs that produce the same output:

$$H(x) = H(y) \ where \ x \neq y$$

For SHA-256 this would require on average 2^{128} or 4×10^{38} attempts to find a collision [12], which is a near impossible task given current computational capabilities. Hashcash simplifies this requirement considerably by only looking for a *partial collisions*. A *k-bit* partial collision would be where only the first k most significant bits match. In practice the hash output is preceded by k zero bits. For example, say we are required to find a partial collision for the string s (which depicts a list of Bitcoin transactions). Let s =*"hello, world!"* such that the hash output begins with four zeros ('0000') [8][11]. We vary the length of the string s by concatenating an integer value x to the end called a nonce and incrementing it each time until we find the desired result. Concatenating the nonce *x=0* does not produce a match:

$$H(hello, world!0) => \mathit{1312af178c253f84028d480} \ ...$$

We keep on incrementing the value of x until we reach the number *x=4250* which gives us the desired result, with four zeros at the start of the hash output:

$$H(hello, world!4250) => \mathit{0000c3af42fc31103f1f} \ ...$$

This task can be made more or less difficult by varying the number of zeros required to obtain the partial collision. As we saw from the above example a relatively simple proof-of-work problem that requires four zeroes at the start of the hash output is quick to solve. A more difficult proof-of-work problem might require a much longer run e.g., ten consecutive zeros which would take on average a much longer time to solve.

Suppose Trudy, who is a miner in the Bitcoin network, wants to earn some bitcoins. Trudy verifies each transaction that is broadcast on the Bitcoin network by consulting her version of the block chain and then adds them into a block of pending transactions that have not yet been endorsed by the network peers. In order to convince her peers in the network that these pending transactions are valid, she attempts to solve the proof-of-work problem, which requires a substantial amount of computational effort on her part. Since all previous blocks are chained together, this ensures that untrustworthy peers have to work harder than honest peers if they want to modify previous blocks and include them in the block chain. The Bitcoin proof-of-work problem requires the hash of a block's header to be lower than or equal to a number known as the *target*. The target is a 256-bit number that all Bitcoin clients share. The SHA-256 hash of a transaction block's header must be lower than or equal to the current target for the block to be accepted by the network. The lower the target, the more difficult it is to generate a block [13]. Trudy broadcasts the results of her computation to the network and the other participants can easily verify it.

Forks can occur in the block chain when two or more miners happen to validate a block of transactions and broadcast their results near simultaneously to the network. Other participants in the network will then use the block they receive first thereby creating two versions of the block chain. Note that each miner's transaction block will differ from that of any other miner in the network, due to the fact that they insert a unique transaction (a.k.a. *coinbase transaction*) at the beginning of the block. The coinbase transaction pays the miner for trying to solve the proof-of-work problem and any other transaction fees from other transactions in the block. It cannot be spent until a further hundred blocks have been added to the block chain to ensure that stale blocks are flushed out of the chain. In Bitcoin, a transaction is not considered confirmed by the network participants until it is part of a block in the longest fork and at least five blocks follow it in the longest fork. Shorter chains will eventually be discarded by the other miners in favor of the longest chain (see Figure 3) thereby maintaining a consistent view of the block chain across the Bitcoin network.

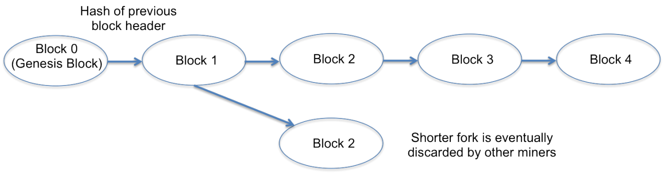


Fig. 3. Fork in the Block Chain

III. Netcoin System Architecture

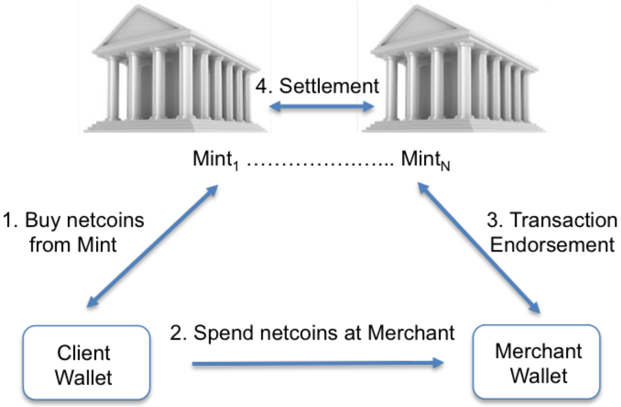


Fig. 4. System Architecture

In this paper, we propose a new electronic cash scheme that is inspired to some extent by the Ecash and Bitcoin protocols. Like its peers, the main aim of the Netcoin system is to facilitate low-cost peer-to-peer transactions by eliminating the traditional banks and financial clearinghouses from the system. The main entities in the Netcoin network are the user and merchant wallets, and the mints. As with other systems, the user and merchant wallets are software applications running on a myriad of desktop and mobile devices. The mints on the other hand are trusted entities in the system whose function is to mint netcoins and endorse transactions in the network. A mint is paid a small amount of netcoins for its work in endorsing each netcoin in a particular transaction. For example a mint could be paid \$0.001 for each netcoin that it endorses. The mints also convert netcoins into fiat currency on request. The mints could, for example be private operators, and perhaps licensed by government. These might be large online retailers such as Amazon who want to reduce transaction costs, thereby making their products cheaper. Other examples may be search engines that want to reduce transaction costs for clients who advertise for them etc.

Like the Bitcoin system all transactions are broadcast to the Netcoin network and the various mints compete with one another to endorse these transactions. However unlike the Bitcoin system, the Netcoin network does not maintain a single public ledger of all transactions in the network. Instead we have an individual transaction history for each coin in the system and refer to this as the "Netcoin Block Chain". Our reasoning behind this architecture is twofold. Firstly, we do not believe that the proof-of-work method is an efficient mechanism for verifying transactions, as it wastes a lot of energy and computing power. Secondly we believe that maintaining a single network-wide block chain is not sustainable for any system that aims to support electronic payments on a global scale. Unlike Ecash, netcoins can be reused repeatedly and this means that there isn't an ever growing database of "spent coins".

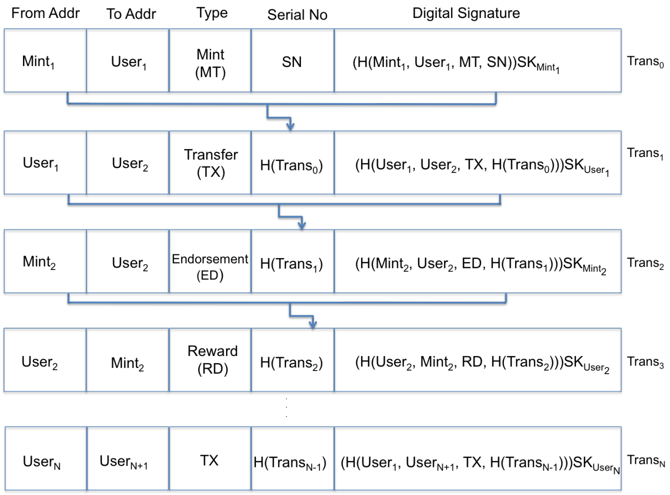
IV. Payment Protocol

The high-level steps in the Netcoin payment protocol are outlined below. We will go into the details of the individual steps in the subsequent sections:

- A user buys netcoins from a local mint in exchange for fiat currency. The mint in question generates the required coins and ties them to the identity of the purchaser using cryptographic techniques. The mint broadcasts the details of the newly minted coins to all other mints in the network, which also store the coin details in their local databases.
- At a later date when the user tries to spend some netcoins with a merchant, the merchant broadcasts the transaction to all mints in the network asking them to verify the legitimacy of the coins.
- Each mint checks its own database to ensure that the coins actually belong to the user that initiated the transaction. Each mint then adds a signed endorsement to the coin's block chain and sends the result back directly to the merchant.
- If the merchant receives a positive verdict from all the mints in the system endorsing the transaction, it randomly picks one of the mints and rewards them for their effort. This is reflected as another entry in the coin's block chain and is broadcast to all trusted entities in the network. The mints update their database to reflect the latest version of the coin. This broadcast mechanism allows a ledger of all coins in the system and to whom they belong at any point in time.

V. Netcoin Block Chain

As stated previously the mints in the system are trusted entities. The role of the mints is to enroll new users into the Netcoin ecosystem and manage the flow of fiat currency in and out of the system. Each mint has a public key pair, the public key of which is widely known in the system. Each user (end users & merchants) is required to generate their own cryptographic key pair and register the public key component and their user ID via a chosen mint in the system.



From Addr	To Addr	Type	Serial No	Digital Signature	
$Mint_1$	$User_1$	Mint (MT)	SN	$(H(Mint_1, User_1, MT, SN))SK_{Mint_1}$	$Trans_0$
$User_1$	$User_2$	Transfer (TX)	$H(Trans_0)$	$(H(User_1, User_2, TX, H(Trans_0)))SK_{User_1}$	$Trans_1$
$Mint_2$	$User_2$	Endorsement (ED)	$H(Trans_1)$	$(H(Mint_2, User_2, ED, H(Trans_1)))SK_{Mint_2}$	$Trans_2$
$User_2$	$Mint_2$	Reward (RD)	$H(Trans_2)$	$(H(User_2, Mint_2, RD, H(Trans_2)))SK_{User_2}$	$Trans_3$
$User_N$	$User_{N+1}$	TX	$H(Trans_{N-1})$	$(H(User_1, User_{N+1}, TX, H(Trans_{N-1})))SK_{User_N}$	$Trans_N$

Fig. 5. Netcoin Block Chain

Figure 5 depicts the Netcoin block chain structure. In contrast to the Bitcoin system, the Netcoin block chain is based around an individual netcoin in the system. Our reasoning for this approach is that having a per coin block chain removes the need for an ever growing ledger of transactions in the system. If the block chain for a particular coin grows beyond a certain length, the coin can easily be removed from the system and a new one issued in its place. Like the Bitcoin block chain we also link all transactions in the Netcoin block chain by including a hash of the previous transaction thereby preventing unauthorised changes to the coin's transaction list.

A. Minting Netcoins

When a user contacts a mint to purchase some netcoins it will need to transfer funds to the mint via an electronic funds transfer (EFT) process, or present a credit/debit card so that the corresponding amount can be deducted from the users account. The first option is the cheapest (but slower than a credit/debit card transaction) as it avoids the charges set by the banks and card processors. The mint will then generate a number of netcoins. Each coin will have a starting block chain entry (see Figure 5 $Trans_0$) which ties the netcoin to the purchaser. It has four fields namely, the issuer's ID, the purchaser's ID, the type of transaction (which in this case is a minting operation MT) and the serial number of the coin. A digital signature follows this on all the above fields with the private key of the mint.

B. Spending Netcoins

Subsequently, when the user spends netcoins at a merchant premise, the merchant creates a second block chain entry ($Trans_1$), which like before contains the user identifier fields and the type of transaction field. The serial number field is replaced with a hash of the previous transaction ($Trans_0$). This links the previous transaction to the current transaction, thereby creating a chain of transactions (similar to that in the Bitcoin protocol). The final field of the transaction is a digital signature on all the above fields signed with the private key of the user (or purchaser in this example). The merchant then broadcasts the new version of the coins to all mints in the system, asking them to verify the authenticity of these coins.

C. Endorsement

Each mint in turn checks their own local database to ensure that the coins do indeed belong to the user who is trying to spend them at the merchant's site. For each coin it hashes the relevant fields from the previous transaction (e.g., fields 1-5 of $Trans_0$) of the version of the coin that it has stored on its database and concatenates it with the first three fields of the newly added transaction (e.g. fields 1-3 of $Trans_1$) and then hashes the result. The mint then tries to match the computed hash with that of the one contained within the digital signature associated with the new transaction ($Trans_1$). It obtains the public key of the initiator of the transaction by using the contents of the first field (*From Addr*) as an index into the public key database. If there is a match then it assumes the new transaction is valid. The mint then adds a further entry into each coin's block chain using an endorsement field (ED). As can be seen in the third transaction ($Trans_2$), the mint links in the previous transaction ($Trans_1$) and signs all the fields with its private key. It then forwards the coins to the merchant, in the hope that it will be selected as the endorser and get paid for its effort.

D. Reward

On receiving a positive endorsement of the coins from all the mints in the network, the merchant picks one of them at random and rewards the mint with a small sum. It adds a fourth entry into the block chain using a reward field (RD), which is broadcast to all mints in the network. This is a public acknowledgement of the fact that a user has spent the coins at the merchant and that the mint that helped in endorsing these coins is the beneficiary of a reward from the merchant. An alternative mechanism could be where all participating mints in the system get a small reward for their effort in endorsing a transaction.

E. Length of Block Chain

From time-to-time a mint may re-issue netcoins to users if the length of a block chain crosses a certain threshold. This measure will prevent a coin's block chain from getting too large - something that will have a direct positive impact on bandwidth and storage costs in the network.

F. Change

Netcoin operates on the principle of *exact change* i.e., a user must provide the exact amount netcoins to the recipient of a transaction. In the event that the user's wallet does not have the required change, the wallet will automatically contact the user's preferred mint and exchange their existing coin(s) for new ones, in order to be able to complete the transaction.

VI. Security Considerations

This section tries to address a number of security concerns associated with the Netcoin system. First and foremost is the trusted nature of the mints in the system. Like the Bitcoin system we too make use of a broadcast mechanism to enlist the help of other entities to verify transactions. However, unlike the Bitcoin system we do not use a computationally intensive process such as the proof-of-work algorithm to lock-in transactions. The Bitcoin proof-of-work algorithm typically takes an average of ten minutes to complete. This means that during this time it is not possible for a merchant to know if the transaction that it is processing is a valid one or not. Such a delay may be acceptable for Internet transactions whereby a merchant will only ship goods or information once the transaction has been approved. However such delays are not necessarily acceptable for point-of-sale transactions whereby a user buys goods or services at merchant premises.

Hence in the Netcoin system we have relaxed the proof-of-work requirement and replaced it with a set of trusted entities, all of which maintain a global view of all transactions in the network and which netcoins belong to which users in the system. The mints can be privately operated under a strict licensing regime. The mints are profit-making bodies and it is in their interest to verify transactions that are broadcast in the network with the hope that they will get rewarded for their effort. It is not in the interest of the mints to collude with users or other mints to defraud the system, since all transactions are broadcast to the whole network and any discrepancies will be quickly identified. The cost of verifying transactions in the Netcoin system is a fraction of what it costs in current financial networks. The reason for this is that once a transaction has been endorsed by all the trusted entities in the system and accepted by the merchant, then the coins are permanently transferred from one user to another and the transaction cannot be reversed, thus removing chargeback costs. We can keep administration overheads to a minimum.

Another concern is that users try to double-spend netcoins at multiple outlets. Again, since the system is transaction based and all Netcoin transactions are broadcast to the entire network, it will not be possible to double spend coins in the system. Any user that steals netcoins from another user must also be able to steal their public key pair in order to create any new transactions or redeem the coins in exchange for fiat currency from a mint. This makes the theft of netcoins difficult.

VII. Conclusion

This paper introduces Netcoin, a new electronic cash system that provides for an efficient means of electronic payment. It addresses the shortcomings of its prominent predecessors, namely Ecash and Bitcoin. These shortcomings are (though not necessarily attributable to both) lack of traceability, deflationary attributes, volatility, an inefficient verification mechanism, and the need to maintain a public ledger that requires a rapidly increasing amount of memory and bandwidth requirements.

With Netcoin these shortcomings have been addressed by the introduction of mints which act as gateways between the fiat and virtual currency worlds. It is envisaged (though by no means necessary) that these mints would operate privately, perhaps with government license. The motivation on the part of mints would be to make electronic commerce more attractive by keeping transaction costs down. This is obviously in the interests of large-scale online merchants such as Amazon and eBay who themselves may be willing to take up this task. Another motivation could simply be to implement government policy of providing for efficient electronic commerce. Major search engines such as Google and Bing in whose interest it is to simulate electronic commerce for their advertising clients may also take up this task.

As a next step, we hope to simulate a full lifecycle of the Netcoin protocol using for example a network simulator such as NS. Our aim will be to show that even with large numbers of users and simultaneous transactions on the system, the network load will be within manageable parameters.

Finally, it is important to note that Netcoins are based on fiat currency and not on blind faith as is the case with Bitcoin. For this reason Netcoin should not suffer from the same volatility issues as Bitcoin. It is also important to point out that Netcoin in itself has no attributes that shape the Internet economy other than facilitating inexpensive peer-to-peer transactions.

References

[1] D. Chaum, "Security Without Identification: Transaction Systems to Make Big Brother Obsolete", *Communications of the ACM*, vol 28, no 10, pages 1030-1044, October 1985.

[2] D. O'Mahony, M. Peirce and H. Tewari, "Electronic Payment Systems for E-Commerce", *Artech House Publishers*, 2nd Ed., 2001.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", http://www.bitcoin.org, 2008.

[4] T. Carper and T. Coburn, "Letter to the United States Committee on Homeland Security - Virtual Currencies", http//www.hsgac.senate.gov/download/letter-to-secretary-napolitano-on-virtual-currencies, 2013.

[5] D. Ron and A. Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph", *Eprint Archive*, https://eprint.iacr.org/2012/584.pdf, 2012.

[6] Stellar - Public Infrastructure for Money, http://www.stellar.org, 2014.

[7] D. Chaum,"Untraceable Electronic Cash," *Proceedings of Crypto 88*, Springer-Verlag, vol 403, pp. 319-327, 1990.

[8] M. Nielsen, "How the Bitcoin Protocol Actually Works", http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/, December 2013.

[9] Bitcoin Fog - Bitcoin Anonymization, http://www.bitcoinfog.com/.

[10] A. Black, Hashcash Proof-of-Work System, http://www.hashcash.org/, 1997.

[11] Bitcoin Proof-of-Work, https://en.bitcoin.it/wiki/Proof_of_work.

[12] The Birthday Paradox, http://en.wikipedia.org/wiki/Birthday_attack.

[13] Bitcoin Target, https://en.bitcoin.it/wiki/Target.

[14] R.C. Merkel,"Protocols for Public Key Cryptosystems", *Proceedings of the Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.