Syscoin: A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business

Jagdeep Sidhu, Msc.
Syscoin Core Developer
Blockchain Foundry Inc.
Email: jsidhu@blockchainfoundry.co

***Abstract*—While Bitcoin (Peer-to-Peer Electronic Cash) [Nak] solved the double spend problem and provided work with timestamps on a public ledger, it has not to date extended the functionality of a blockchain beyond a transparent and public payment system. Satoshi Nakamoto's original reference client had a decentralized marketplace service which was later taken out due to a lack of resources [Deva]. We continued with Nakamoto's vision by creating a set of commercial-grade services supporting a wide variety of business use cases, including a fully developed blockchain-based decentralized marketplace, secure data storage and transfer, and unique user aliases that link the owner to all services controlled by that alias.**

1. Introduction

Syscoin is a permissionless blockchain-based cryptocurrency with a set of smart contracts which have been thoroughly tested and built on the Bitcoin scripting system using OP1 to OP16 standard script op-codes, representing coloured coin transactions, controlled by a hardened layer of distributed consensus logic for each smart contract (Syscoin service) while still retaining backwards compatibility with the Bitcoin protocol. These contracts can be combined to form building blocks for blockchain-based e-commerce solutions. Syscoin's hardened smart contracts are in contrast to turing-complete smart contracts, which, by definition, are not hardened due to the open-ended nature of the underlying scripting language. Commercial integrators who are looking for a secure solution to leverage the increased efficiency that blockchain technology allows compared with traditional e-commerce applications are better off trying to use a hardened service which cannot change and is well-tested with regression testing, white box and black box testing, specifically targeting rules of the application. Integrators are also inclined to choose the most powerful network available (currently Bitcoin). Syscoin combines both of these features, making it a compelling choice of network for end users.

1.1. Turing-complete scripting

Turing-complete smart-contracts are technical marvels but they may face issues when applied to practical business processes. Business logic is normally hardened during the Software Requirements and Specification stage of the software development life-cycle. Hardened contract rules make sense in the context of a system that has a measurable number of points of failure (like Bitcoin). In contrast, Turing-complete systems have infinite paths of execution and risk of failure. The current consensus appears to be that if smart contracts are not needed to solve a problem they are best avoided until the languages and toolboxes on top of the smart contract core API are hardened to a point where they certified to be used by the general public. It is an open-box software experiment that is useful for applications which can offer cost-effective contracts that must change on demand. In the context of e-commerce applications, this level of flexibility is not required. In contrast, with Syscoin we have decided to try to generalize applications into a set of hardened services that can be used in conjunction with each other to create complex use cases.

1.2. Innovative service layer

User aliases form the backbone of all Syscoin services. We have created an alias identity system whereby recognizable names facilitate the use of services on the network. Data and public keys associated with aliases are stored within the network, allowing aliases to perform blockchain-based activities such as multisignature signing, payment discovery and maintaining identity payment balances.

Once an alias has been initialized on the network, it can then create a variety of other services, including: creating an offer to buy or sell in the decentralized marketplace, creating a transferable certificate which represents proof of ownership and can contain public and private encrypted information, creating an escrow transaction including multisignature payments and refunds, and creating and transmitting secure messages to other aliases.

Each alias is cryptographically linked to a service by enforcing a rule that the alias output of the last alias transaction is linked to the creation of any subsequent Syscoin service transactions. This ensures that the owner of the alias is the only party which can make those transactions. In addition, cryptographically secure signatures (backed by enormous proofs-of-work) are required in order to make changes to the services that aliases are linked to. By linking services to an identity system it makes it much easier to integrate services

978-1-5090-2991-4/17/$31.00 ©2017 IEEE

into real-world scenarios which use identity-based work flows. Almost anything we do today requires the signature of a known actor on the service contract. By providing a cryptographically secure mechanism to create, manage and link these identities to service contracts, it ensures a seamless integration to real-world business processes.

1.2.1. Mechanism design. Bitcoin provides two incentives for miners: block subsidies through rewards, and transaction fees. As Bitcoin rewards wind down the network could become unstable due to degrading miner incentive to do what is in the best interest for network security. Issues such as selfish mining and undercutting are discussed in greater length in a paper presented at the ACM CCS [Nar16]. What we present is a novel mechanism design that prevents mining incentive from degrading by tying in usage of services to an inflation metric for block rewards. Transaction fees remain to provide incentive to mine and relay transactions but rewards depend on the demand for using the Syscoin network. A utility metric can be established by determining the number of Syscoin service transactions per block. In Syscoin 2.1, an arbitrary number was chosen (5, not network enforceable) which represents the ”high-demand” cutoff threshold for when to burn fees (under the threshold) or when to inflate the fees in the rewards (above the threshold). This means the monetary base can expand (inflate) slightly to accommodate demand for Syscoin services and contract (deflate) when there are blocks that fall under the threshold. The fees that are being burned or inflated are not the transaction fees, which are always paid to miners separately from the block reward, but the Syscoin service fee, which is an additional and separate fee from the transaction fee and is dynamically adjustable from within the rates peg aliases. This creates a democratic system that carries the fee rate which is capable of adjusting the monetary supply based on demand. The result is a price stability mechanism similar to inflation targeting by central banks but carried out in a decentralized fashion. The mechanism design closely follows the concept of Ideal Money [Nas02] discussed at a Penn State lecture given by John Nash Jr. If we apply the notion of service transaction rates to facilitate the transfer of utility between network participants we have a metric that is the first of its kind, one that denotes true demand for the currency in circulation as a public utility that is auditable and provides money transfers with transferable utility; in other words, ”quality” money which would be classified as ideal. Nash alluded to using a ”public utility” such as the supply of electric energy or water as a high quality utility for inflation targeting but those are indirectly related to demand which is indirectly related to velocity of money. Syscoin provides a way to determine the highest quality utility metric possible by providing a way to calculate true money velocity directly by averaging the service transaction creation rate over the monetary base and adjusting the base to accommodate demand in order to achieve price stability.

1.2.2. Self-governing rate system. In order to allow users to transact in currencies other than Syscoin, we developed a mechanism to validate that offers were paid in correct amounts and to the correct person. This implementation is found in the user interface as well as in the consensus code.

The sysrates.peg alias stores the current exchange rates between supported currencies, other digital tokens and Syscoin. It is updated dynamically based on data sourced from exchanges. Other data points such as transaction fees and arbiter fees are also stored and dynamically adjustable during network run-time to avoid having to do any soft or hard forks and having them take effect in real-time versus voluntary updates of miners and client wallets. Transaction fees are used for determining the amount of fees used when sending payments to escrow with Syscoin, Bitcoin or ZCash [Devb]. Because miners may change the fee amount required to mine and relay a transaction, these variables are dynamically adjustable based on market conditions.

Sysrates.peg is simply a reference implementation created to provide the marketplace with a ready-made solution. Syscoin network users also have the ability to create their own rate pegs. By allowing users to select the exchange rate alias that their offers rely on, we create a self-governing system of exchange rates and fees which adapt to the needs of the users on the network.

1.2.3. Quality assurance through network simulation. A test suite was developed to allow the simulation of live network scenarios. Tests cover pruning, expiry and general use-cases of Syscoin services. It is an integral part of achieving a commercial quality level product in any software application. The setmocktime rpc call is used to set the time in-advance of blocks to simulate expiration of services and pruning.

1.3. Alias identities

We have applied domain-name like rules to Syscoin alias identities, allowing only unique case-insensitive names. Users are now able to send coins and encrypted messages to an alias using any case formatting desired, the recipient will always be the user who owns the lowercase version of the alias.

1.3.1. Cryptographic security through alias identities. Any Syscoin service that a user creates or updates must update an alias identity input which employs a cryptographic scheme that secures the transaction with provable ownership of those transactions. Consensus code for Aliases, Offers, Certificates, Escrows and Messages all require inputs from the Unsigned Transaction Outputs (UTXO) of an alias transaction that has been signed with the owners private key. This allows for an identity to play a key role in ensuring safety, secure from impersonation or any other attempt at attacking the integrity of the relationship between the identity and services that are associated with the identity. Because the inputs need to be valid in the UTXO database, at least one network confirmation of the inputs is required to ensure that the owner is indeed the one who is the one capable of making these transactions. In order to improve usability,

five outputs (an arbitrary number) are created upon an alias transaction so that multiple service transactions relating to an alias identity can be made within the same block on the network. The alias consensus code ensures that the public key of the alias input to the transaction matches the public key of the alias. This validates the user who is creating the transaction to modify or update an alias identity.

1.3.2. Transfer of ownership. Aliases may be transferred to another public key but cannot be shared between multiple aliases. At the consensus level, transfers are checked to ensure the new public key of the alias does not already exist in another alias within the Syscoin alias database.

1.3.3. Zero-knowledge alias authentication. Alias private keys can be generated deterministically by supplying a password hashed with a generated random 256bit number known as password salt. The salt is stored within the alias constrct. Upon interactive client logins, the derived key can be regenerated based upon the user supplied password and checked against the public address of the alias that is being authenticated against through an alias information lookup from within Syscoin Core. This enabled wallet-less on-chain controls and authenticated spending of coins/services without requiring a transfer of credentials over any network.

1.3.4. Safe search. When creating aliases, certificates and offers, users can choose if they wish to enable or disable safe search. Any user who is adding content that is not suitable for public searching can set their offer to private which will hide it from searches but maintain its validity on the network.

1.3.5. Expiration. Alias expiry happens based on time. The blockchain protocol acts as a decentralized time server which stamps blocks based on height and time. All other services connect to aliases and use the alias that owns the service to detect expiry. Offers, certificates and messages expire when the alias related to it expire and escrow will expire if and only if both buyer and seller aliases involved are expired.

1.3.6. Multisignature identities. Syscoins alias identity system is linked to a public address upon creation. This can be any type of standard address including P2SH, P2PKH or even P2SH-P2WPKH (Segregated witness address).

Syscoin builds upon Bitcoin's hardened smart-contract design by allowing aliases to link ownership to P2SH addresses which are script hashes of any script defined contract that will run through the consensus layer on the Syscoin blockchain network. An example of a script-enabled smart contract that can run on Syscoin is the standard multisignature contract where multiple parties are required to sign for the completion of a transaction.

By combining an identity system with multisignature capabilities we have an easy to understand system that allows users control over their identity while providing maximum flexibility in terms of real-world usage.

1.4. Certificates

Digital certificates on the Syscoin blockchain are useful for all kinds of applications; from storing bits of data to creating data that may be sold and automatically transferred upon purchase, all with provable ownership via the blockchain.

1.4.1. Public and private data. Certificates, like aliases, have public and private data. Private data can be accessed by foreign aliases either through creating a multisignature alias and including other aliases or by transferring ownership of the certificate to the new owner.

Using a multisignature approach allows certificate owners to maintain control of their certificates while still allowing decryption of private data by other users. In this instance an owner would change the alias of the certificate to point to a new multisignature alias, then assign two aliases owned by the owner and one alias owned by another party.

1.4.2. Transfer of ownership. Certificates can be transferred to other aliases. New owners will receive reading rights for any private encrypted data and the transfer can be configured to allow editing of certificates upon transfer.

1.5. Escrow

Syscoin's integrated escrow service allows safer payments of offers by securely holding a buyer's tokens in escrow until the terms of the sale are met and the buyer releases payment to the seller. Syscoin uses an arbiter-based system, whereby arbiters act as trusted third-parties between buyers and merchant for a sale in the decentralized marketplace. An arbiter is paid based on a dynamic fee set in the rates peg for the offer that is sold. At the end of the process of completing an escrow all three parties can be rated and given feedback related to the sale.

Arbiters are chosen by buyers when accepting an offer. Normally the buyer and seller would agree on an arbiter before an offer is accepted. In most cases no dispute is filed and no arbiter action is needed.

If a merchant does not ship goods, the arbiter refunds the buyer. If the buyer receives goods as described but does not release payment, the arbiter releases funds to the merchant. The feedback and rating system should help prevent irrational behavior by aligning incentives such that it allows actors to benefit if acting honestly.

Escrow works with native payments in Syscoin as well as external payments with ZEC/BTC by signing transactions inside of the Syscoin network and posting to the appropriate network once the escrow contract is complete.

1.5.1. Escrow support for external payments. The multisignature escrow feature works well with our DirectBTC/DirectZEC integrations which allow users to sign and send raw transactions to the Bitcoin/ZCash networks and spend those coins, all implemented via the Syscoin network. In Syscoin escrow, if a user wishes to pay via

Bitcoin or ZCash they would pay to a generated P2SH representing an escrow address. The raw transactions to send those coins to the merchant, reseller or buyer would all be done in Syscoin. Fully signed payments are sent to the Bitcoin/ZCash networks automatically upon release or refund with no manual merchant interaction required.

1.6. Offers and decentralized marketplace

We have developed a marketplace where users can securely and reliably buy and sell a variety of items. Entire stores can be created directly through the marketplace to sell a user's own products or resell others' products for commission.

1.6.1. Alias rates peg. The sysrates.peg alias is used by default in all offers as it is managed and updated by the Syscoin team and provides fiat and cryptocurrency price updates based on data sourced from exchanges. Setting the currency of an offer looks up the conversion rate at the time of sale and applies it in taking tokens from the consumer sending to the merchant. Since the offer consensus code can look up what price peg was used and at which block height, it has the ability to detect that a correct payment was made at any given time. This means any other nodes synchronizing from a previous block will be able to deterministically detect payments and discard those that do not pay enough.

1.6.2. Digital sales. Certificates may be sold in conjunction with offers to create sales of digital ownership. A certificate may hold private information such as codes or registration keys that are redeemed for some service by the buyer of the offer. Certificates can be automatically transferred to the buyer upon completion of sale.

1.6.3. Reselling with whitelists. Merchants may leverage a whitelist feature to offer resellers the chance to sell their offers for a commission. This allows drop-shipping of goods and services while offering provable sales through the decentralized marketplace. The merchant who created the offer controls the whitelist and can add a discount level on a per entry basis for each reseller. If the merchant sets their offer to private, then end users must purchase the item through one of the participating reseller offers.

1.6.4. Feedback and rating system. Escrows and offers sold through the marketplace offer a convenient way to rate and leave feedback on a per sale basis. For an escrow, one rating is accepted (a number from 1 to 5) to represent a users satisfaction level with a transaction, with 1 being the least satisfactory and 5 being completely satisfied and recommending the user to others. Ratings and feedback can be given to and from arbiters, merchants and buyers.

1.6.5. Multiple payment options. Syscoin currently offers three payment options which can be used in combination. Syscoin, ZCash and Bitcoin are currently the three offer options for payment. Syscoin is the native token and, as acceptance of the network grows, will likely be the token of choice for payments. However, to achieve network effect Bitcoin was added which has the highest liquidity of any digital token. It allows a vast community of users to use Syscoin services with little to no cross-chain configuration. ZCash is helpful for anonymous payments and was added in the same manner as Bitcoin. The Syscoin private key of the merchant who creates an offer is the same private key used for payments in Bitcoin and ZCash. The ease of use and convenience provided by this feature makes it a key part of the potential growth and network effect for Syscoin services.

1.6.6. Private payments via ZCash. Because Syscoin addresses are compatible with ZCash Transparent addresses we can offer ZCash support with complete multisignature compatibility; allowing for optional Syscoin escrow functionality. A merchant has the ability to select a combination of payment options from a list of SYS, BTC or ZEC. Once a buyer tries to buy the offer they will see the payment options available. Once they select ZEC, a ZCash transparent address will be generated which uses the same private key as the merchants Syscoin address.

1.6.7. Shipping notification system. A payment acknowledgement button on escrow and offer payments allows a multi-use notification system to the buyer that either the merchant acknowledges payment and/or they are about to ship the product. Tracking and other shipment information can then be sent via the encrypted messaging system.

1.6.8. Marketplace moderation. Marketplace moderation is done through the safe search feature which allows for three tiers of moderation which is effected by the use of a multisignature sysban alias owned by the Syscoin team. Users of the network are able to set services to safe search but if they are creating content not suitable for viewing and not using safe search then the team can moderate these such pieces of data to remove from public viewing. The sysban moderation does not disable the services from use on the network; rather, they become publicly unviewable and omitted from searches in a similar fashion to private offers.

1.7. Messages

Encrypted messages use asymmetric cryptography to send data to alias public keys. The identity system plays a key role in messaging because senders and recipients aliases are used to determine the keys for encryption. The sender and recipient keys are encrypted with the message so that no third parties can read the data transmission without having the private key of either of the parties involved. Multiparty encryption is also possible through the use of multisignature alias identities.

1.8. Blockchain pruning

Bitcoin has an option pruning feature which is quite different than what we describe here. Perhaps Segregated Witnesses (Segwit) is the closest related concept to Syscoin's pruning mechanism because it saves bandwidth as well as storage costs. Just as Segwit splits transactions into two with just a hash of the witness transaction that is carried forward by SegWit compliant clients for consensus validity, Syscoin's pruning mechanism works similarly with service transactions by splitting the Syscoin service transaction into two outputs. One is the ownership-provable output which links the service to a public key that is capable of modifying the service linked to the information in the output. It is a small scriptPubKey which carries just enough information to prove that a user owns a certain alias. Every other service is linked to an alias which extends provability to services outside of aliases.

Figure 1 shows these two outputs. Output 1 has only the alias output which links to an owners public key. The OP code denotes the type of service, a name and guid to be able to look up the alias from the Syscoin service DB and a hash of Output 2 which is the data carrying OPRETURN and representing the data in the Syscoin transaction.

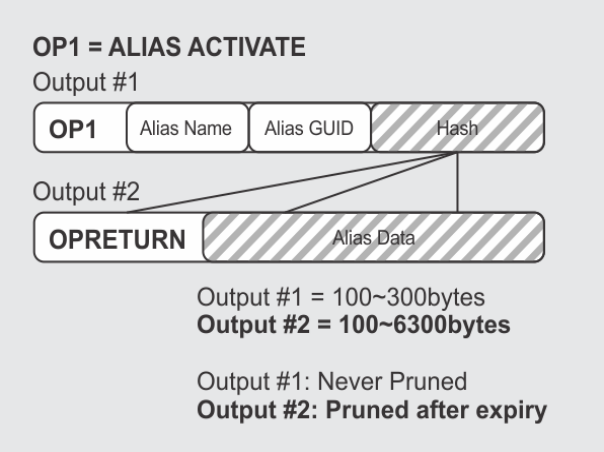

Figure 1: Syscoin OPRETURN data hashed into UTXO

Offers, certificates, messages and escrow transactions all create similar Output 1 style outputs with different OP codes, but they all must have an output for the alias which proves that the owner of the alias is the one making the transaction linked to any service. The consensus code will extract the data from Output 2 and check to see that the hash matches from Output 1 to ensure integrity of the data from data mutation attacks. Doing so allows us to avoid having to hash the contents of Output 2 inside of the blockchain transaction. The data must be available on demand inside of the database and it remains so until expiration where it is assumed it will no longer be needed because updates to services are disallowed if expired. A combination of using prunable outputs with expiration of services allows us to create a unique pruning mechanism that will save new nodes from having to download and store expired service data while syncing with the network. From preliminary tests run on node4 of our unit test suite shows that data savings are remarkable. Node4 is the node that is set to txindex=1 which disables the Syscoin pruning mechanism. As the unit tests run this node will save all of the pertinent service data inside of its database as specified by the design. In a later test, we have run it as txindex=0 meaning pruning is activated and synchronized a new client. Since most of the services were expired when the unit tests were run, the new node synchronized very little data from node4. After synchronization of the blockchain was complete we noticed that the data directory size of node4 was 335Kb while the new node was only 470 bytes while still maintaining complete protocol consensus.

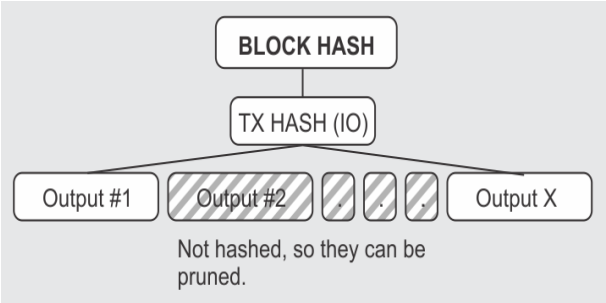

Figure 2: Syscoin outputs not hashed by blockchain

2. Future work

Depending on the demand for Syscoin services there are some useful features that can be added with minimal effort but were intentionally left out due to time constraints for the 2.1 Core release.

2.1. Lightening networks

We have tested and confirmed the usage of Lightening network on Syscoin core mainnet. This will allow for micro-transactions from within the Syscoin network and allow for the network to scale to VISA like transactions-per-second levels. It may also be useful for off-chain service interactions with offers like auctions. We will continue to research and develop this innovation further.

2.2. Aliases

Deterministic aliases with supplied passwords would be much more secure from rainbow or dictionary based super computer attacks if the password salt was not stored and accessible. If the salt was provided via an organic mechanism such as fingerprint or optical recognition this would mean the user would be able to securely and safely log in to their service portal and the requirement of unique and hard to remember passwords is mitigated.

2.3. Offers/Escrow

Proof-of-shipment is something we have innovated and are expanding upon the shipping notification system from within the escrow and offer service layers. A video can be taken by the merchant, hashes and included in a data-field from within the shipping notification transaction so that arbiters and buyers can assure that any disputes would be quickly and efficiently resolved. This helps relieve a few concerns including knowing that the merchant shipped goods as described by the offer as well as reduce buyer incentive for charge-back fraud since there is proof that the shipment took place and thus the argument that the merchant did not send goods that are as described or did not ship at all are not are invalid. Arbitration and insurance would become cost-effective means to insure true buyer protection as markets form as a result of the technology. There is currently a proof-of-concept under development for this proof-of-shipment mechanism.

2.4. Certificates

Using torrent trackers and other P2P-style hosted data source is an area of future research as it will allow for scaling certificate data above the limits of 1KB of data for private encrypted information while maintaining network security. Stored data can be encrypted to the public key of the certificate owner which is the alias identity it is assigned to. This way data can be hosted in public rather than on private servers that are maintained with rigorous security to avoid breaches of access. Systems like IPFS/IPDB are being researched to help offload data requirements of non-consensus related information to the cloud.

3. Specs

Syscoin has an 888 million maximum coin limit, 1 minute block time and is proof-of-work SHA-256 merge mineable (with the majority of network security coming from Bitcoin). Syscoin 2.0 had a block reward of 54.13 tokens per block. Syscoin 2.1, released on December 18, 2016, represented a ”halving” event in block rewards, reducing them to 16.39 tokens per block (a reduction of 330 percent).

Mining rewards, designed to be gradual and smooth, end at about 800 million coins (block 24,177,646 will happen around the year 2052) and thereafter supply is inflated via the Syscoin mechanism design of the inflation/deflation system assuming services are in high demand.

4. Conclusion

We have presented a set of hardened smart-contracts that can be used in conjunction with each other and an identity system to provide blockchain-based e-commerce solutions for small, medium and large businesses. The processes used by businesses and entrepreneurs may transfer to Syscoin without the need to re-invent the way people work today. The goal is not to force the technology and processes on the people using it but to bring people to the technology who are in need of a blockchain-based solution to their problems. The mix of unique features of Syscoin in an architectural framework that enabled high security through merged-mining and low inflation enabled trust-less payments and services to be used today in commercial ventures and partnerships as well as provide an investment proposition to holders of Syscoin token holders. A low barrier of entry for external communities can be leveraged to help create a network effect for Syscoin and its services.

Acknowledgments

We would like to thank Satoshi Nakamoto, the late Hal Finney and Gavin Andresen for bringing Bitcoin protocol and reference client to mainstream adoption state. Without the work of these people none of what we have worked on with Syscoin would have been possible.

A special thanks for the Blockchain Foundry Inc. team of Sebastien Dimichele, Chris Marsh, Brad Hammerstron, Willy Ko and Dan Wasyluk for their peer review and updates.

References

[Nas02] John F. Nash. “Ideal Money”. In: *Southern Economic Journal* 69.1 (2002), pp. 4–11. DOI: http://www.jstor.org/stable/1061553.

[Nar16] Arvind Narayanan. “On the Instability of Bitcoin Without the Block Reward”. In: *ACM CCS* (2016). DOI: https://www.cs.princeton.edu/~smattw/CKWN-CCS16.pdf.

[Deva] Bitcoin Core Developers. *Bitcoin reference client*. URL: https://github.com/bitcoin/bitcoin.

[Devb] ZCash Core Developers. *ZCash*. URL: https://github.com/zcash/zcash.

[Nak] Satoshi Nakamoto. *Bitcoin: A peer-to-Peer Electronic Cash*. URL: https://bitcoin.org/bitcoin.pdf.