ALIAS

White-paper

Eirik Korsell, Philip Mueller, Yves Schumann

October 1, 2020
Alias version 4.3.x (interim edition)
https://alias.cash

Contents

1 Abstract

Cryptocurrencies (*digital assets designed to work as mediums of exchange*) permit users to securely send money without trusting any other human intermediary or any other centralised third-party system or institution, such as a bank, to verify those transactions[1]. Instead the strong cryptography and mathematics inherent in the software algorithms secure the peer-to-peer network and safeguard against forgeries and ensure transaction finality. The network of participating nodes together creates the blockchain where all transactions and balances are recorded and ensures network immutability. This is a truly transforming technology and has the potential to benefit people across the globe. However, the blockchains in classic cryptocurrencies (*such as Bitcoin*) are transparent public ledgers that are accessible to anyone and the full transaction history is preserved and the blockchain is readily available for analysis[2]. This presents a very serious issue for users online and financial privacy. Even though the pseudonymous users of classic blockchains are not directly associated with their real-world identities, every transaction among these pseudonyms is potentially traceable and every transaction is recorded for all posterity and for anyone to access and view.

In this document, we present **Alias**; a cryptocurrency that uses a range of advanced cryptographic techniques, such as dual-key stealth addresses and ring-signatures to achieve unlinkable, un-traceable and private transactions on its blockchain. **Alias** also comprises a novel privacy-preserving consensus mechanism, '**Anonymous-Proof-of-Stake**' that let its users retain full privacy as they support the network by running the software and staking their coins. **Alias** also protects the user's online identity by integrating Tor (*The Onion Router*)[3] in the software and is therefore a comprehensive privacy focused cryptocurrency. **Alias** also retains the ability to conduct '*open*' public transactions (*much like Bitcoin*) and this may serve certain use cases and blockchain audit-ability. **Alias** provides the best of both worlds, total privacy and public transactions, without any compromise. In this paper we present the current technology and functionality of **Alias** to the common reader with some experience and knowledge of blockchain and cryptocurrencies. We also discuss how we achieve confidential and privacy maintaining consensus. Further references to explain and expand on certain topics are included within the text as footnotes.

[1] https://en.wikipedia.org/wiki/Cryptocurrency

[2] https://www.respublica.org.uk/disraeli-room-post/2015/03/24/bitcoin-is-not-anonymous/

[3] https://www.torproject.org/

2 Acknowledgements

Alias would not have been possible without the foundations laid by what came before and in particular the *Bitcoin*[4], *Blackcoin*[5] and *ShadowCash*[6] developers and the work by the authors of the *CryptoNote*[7] and *Zerocoin*[8] protocol that provided inspiration for some of the technologies developed for use in Alias. Although the Alias lineage can be traced back through previous projects, open source software provides inspiration for innovation and progress. Alias has taken a particular direction to improve, enhance, innovate and further develop the unique privacy technology inherent in its code base. The Alias developers have also written copious amounts of original code and developed innovative technology not seen in any other comparable cryptocurrency.

[4] https://bitcoin.org/bitcoin.pdf

[5] https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf

[6] https://www.cryptoground.com/shadowcash-white-paper

[7] https://cryptonote.org/whitepaper.pdf

[8] http://zerocoin.org/

3 Disclaimer

The Alias software and the source code is being developed and maintained by an independent collective of blockchain enthusiasts. The Alias collective does not hold any personal information on any users of the Alias software and will never ask for any identifying information from any user in order to download and use the software. The Alias collective does not gather any personal data. **The Alias collective will NEVER ask users for their private keys, wallet / data files or any other information that could enable us to access a user's fund or wallet or identify the user.** The Alias collective does not hold any kind of key / information / data or have any other means or technology that would enable it to analyse transactions on the Alias blockchain with a view to identify any user. The creation of the Alias network did not involve any trusted setup and no personal information was ever collected. No person associated with the Alias collective process any kind of personal data and we do not believe any relevant data protection laws apply. We would be happy to assist with any formal request from any relevant authority but would have to inform them that we do not hold any data and that we are unable to conduct any kind of network analysis.

The Alias collective understands and appreciates that the Alias software holds and protects your money and we take this responsibility seriously. The Alias collective is doing everything it can to make sure that the network and the software is as secure as it can be. However, the software comes with no guarantees and must be considered experimental. In general, confidentiality and privacy should never be considered as an absolute and users should do their own due diligence and risk assessment and take whatever measures they deem appropriate to protect themselves online. When conducting a risk assessment with regards to your own privacy online; consider that we do not have conclusive knowledge of any possible adversary's capabilities or resolve to achieve their aims. The Alias collective does not in any shape or form condone any illegal activity and Alias has not been created to facilitate any kind of illegal activity. If we would become aware of any illegal activity we would inform the relevant authority without fail.

The Alias collective is not responsible in any way whatsoever for any lost password, hacking, acts of God, war, natural disasters or any other unforeseen event that could compromise the network and cause any loss of funds. The Alias collective cannot be held responsible for any users incompetence or recklessness that may result in any loss or liability. The current parameters of the software are set according to how we believe the network should operate, but network parameters and characteristics could change if deemed necessary. The Alias collective will always work to increase the privacy of the Alias software, increase the value of the company and hence the value of your investment. All decisions are made with this in mind. This is not investment advice and we do not offer any investment or financial advice. Alias could be worth zero and you could loose all your money.

If you own Alias and you run a Alias wallet, it is strongly recommended that you always keep up to date with developments and always run the latest version of the software. This is best done by either subscribing to our Discord server[9] or use Blockfolio[10].

[9] https://discord.gg/ckkrb8m

[10] https://blockfolio.com/

4 Notes

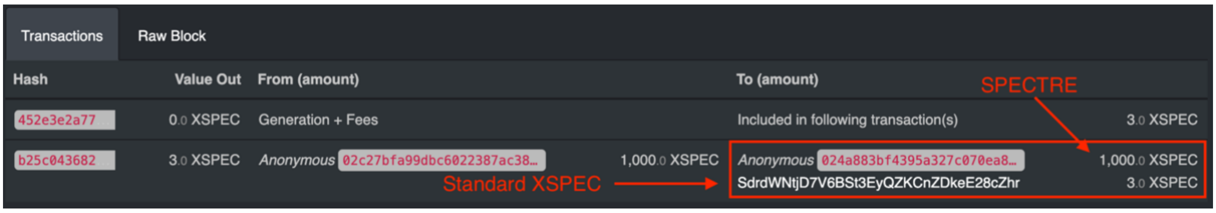
ALIAS can be sent in a public transaction or a private transaction. For the purpose of this document a public transaction will send ALIAS (public) (*public-ALIAS*) and a private transaction will send ALIAS (private) (*private-ALIAS*). (p)ALIAS transactions employs well known technology albeit used in creative ways and some parts of the White-paper is to a large degree referencing already published material from various sources. The newly added section on '**Anonymous-Proof-of-Stake**' however is unique to this White-paper and an original ALIAS (private) technology. There is no scope in this document to discuss cryptography or mathematics and the author is neither a cryptographer nor a mathematician. The descriptions of cryptographic functions are taken from the relevant source documents that are referenced throughout this document and if you are so inclined you can read up on the details. This is not meant as an academic paper or as a reference document but rather as a brief description of the ALIAS (private) network and the underlying technology used. This document does not intend to contribute to the debate about privacy online and this discussion is beyond the scope of this document. We simply believe that we have an absolute right to privacy in our financial affairs online as we do in the real world and so our ideology is also simple; to provide real decentralised resilient privacy and confidentiality on the blockchain and offer provable private transactions for users.

5 Commonly Used Terms

Below we have listed the most commonly used terms you are likely to come across in this document along with a short simple explanation.

Blockchain	A shared, immutable ledger for recording the history of transactions in blocks.
Block	A defined data structure that contains a record of transaction data and other values.
ALIAS (public)	The the name of the public coins on the blockchain.
ALIAS (private)	The name used for the private coins on the Alias blockchain in this document.
UTXO	Unspent Transaction Output that can be spent as an input in a new transaction.
ATXO	UTXO that can be spent as in input in a private transaction using a ring signature.
Keyimage	A unique value associated with a specific ATXO calculated using a private key.
Spent (UTXO)	A UTXO is spent when it has been 'consumed' as an input in a new transaction.
Spent (ATXO)	An ATXO is spent when the related 'keyimage' has been included in a valid ring signature.
Hash Function	A mathematical one-way function that generates fixed size data from an arbitrary input.
Hash Value	A numeric value of a fixed length that uniquely identifies the data input in a hash function.
Block Hash	The hash of a block's header.
Kernel Hash	A hash value used in Proof-of-Stake.
MIXIN	A chaff or dummy ATXO not being spent in a current transaction, used in a ring signature.
Tor	The Onion Router. A layered network that attempts to hide your IP address.
VIN	The 'collection' of input data for a transaction, including the UTXOs/ATXOs to be consumed.
VOUT	The 'collection' of output data for a transaction, including the new UTXOs/ATXOs generated.
PoS	Proof-of-Stake. A consensus mechanism introduced with Peercoin.
PoSv3	Proof-of-Stake v3. Consensus mechanism developed by the Blackcoin developers.
APoS	Anonymous-Proof-of-Stake. Privacy consensus mechanism introduced by the Alias developers.

We will use some screenshots from the Alias block explorer[11] to show examples of transactions and to explain some of the features. The block explorer is not custom made for the Alias Blockchain and will always show '*ALIAS*' as the designation behind a value. When this is preceded by the word '*Anonymous*' it designates a ALIAS (private) value. References are designated by superscript and relate to the relevant footnotes and links at the bottom of each page. Please follow the links to explore certain topics in more detail.



Hash	Value Out	From (amount)		To (amount)	
452e3e2a77	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	3.0 XSPEC
b25c043682	3.0 XSPEC	*Anonymous* 02c27bfa99dbc6022387ac38…	1,000.0 XSPEC	*Anonymous* 024a883bf4395a327c070ea8… SdrdWNtjD7V6BSt3EyQZKCnZDkeE28cZhr	1,000.0 XSPEC 3.0 XSPEC

[11] https://chainz.cryptoid.info/ALIAS/

6 Alias Characteristics

The Alias software is encompassing and integrating the following:

Bitcoin Core	Core technology of the Alias blockchain.
Proof-of-Stake.v3 (PoSv3)	Secure open consensus mechanism for ALIAS (public).
Anonymous-Proof-of-Stake (APoS)	Secure private consensus mechanism for ALIAS (private).
Private transactions	Using dual-key stealth technology and ring signatures for ALIAS (private).
Tor Hidden Services v3	to hide your real IP address (*all Alias nodes run as hidden services*).
OBFS4	To hide the fact that you are using Tor to avoid censorship.
HD Wallets	Hierarchical Deterministic wallets (BIP-0032 specifications).

Given the specific specifications and design, Alias takes on the characteristics of both a Bitcoin like blockchain and a Monero like blockchain. This combined with a pure PoS strategy makes Alias unique.

6.1 Alias Specifications

Genesis block	Block #1 mined on 20/11/2016 (later transition to PoS only)
Ticker	ALIAS
Initial supply	20,000,000 ALIAS
Additional supply	3,000,000, ALIAS on 27/09/2019
Network outputs (public)	ALIAS (public) – public coins
Network outputs (private)	ALIAS (private) – private coins
Consensus (ALIAS (public))	Proof-of-Stake v.3 (PoSv3)
Consensus (ALIAS (private))	Anonymous-Proof-of-Stake (APoS)
Difficulty retarget	Every block
Target block time	96 seconds
Block reward (PoSv3)	2 ALIAS (public)
Block reward (APoS)	3 ALIAS (private)
Transaction fees	Standard fees are the same for ALIAS (public) and ALIAS (private)
Coin maturity (confirmations)	450 for stake reward / 10 for ALIAS (private) / 6 for ALIAS (public)
Max supply	No max supply (see illustrations on page 10)
Inflation	Decreasing over time tending to zero
Code repository	https://github.com/aliascash/alias-wallet
Supported platforms / OS	MS Windows, OSX, Linux, Raspberry Pi
Website	https://alias.cash/
Block explorer	https://chainz.cryptoid.info/ALIAS/

It is important to understand that the total '*outstanding*' amount is the sum of ALIAS (public) + ALIAS (private). On the next page we have projected both a minimum and a maximum inflation rate and total '*outstanding*' amount of ALIAS (public) + ALIAS (private) over 20 years. The real value of the total '*outstanding*' amount will depend on the ratio of ALIAS (public) / ALIAS (private) created by the two different consensus mechanisms over time.

6.2 Inflation rate over 20 years

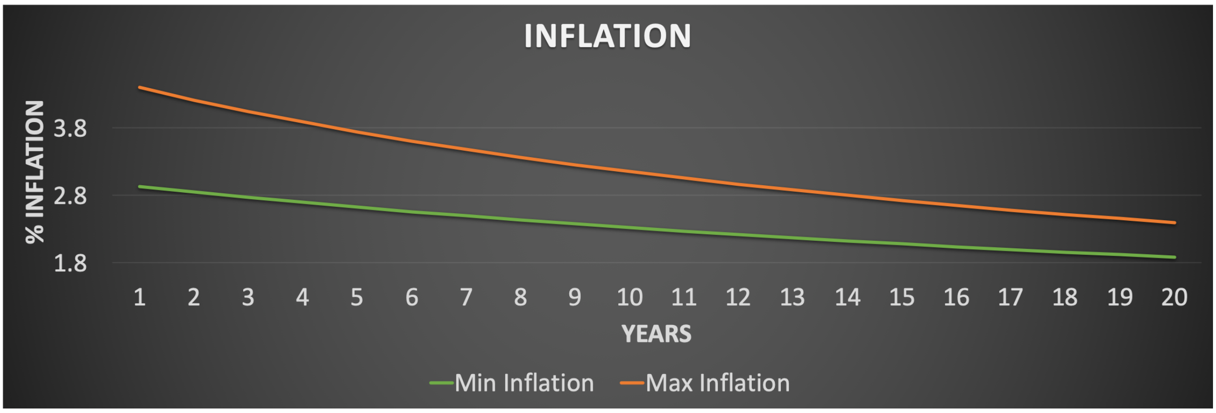


Figure 1: Min inflation rate after 20 years = 1.88%, Max inflation rate after 20 years = 2.40%

The minimum values represent a scenario where 100% of the blocks are created by PoSv3. The maximum values represent a scenario where 100% of the blocks are created by APoS.

There are different strategies employed by different cryptocurrencies around Inflation. Alias differs from Bitcoin and some other cryptocurrencies in that we have a constant coin generation scheme but a relative inflation that tends to zero over time. There will always be an incentive to stake Alias and the system will never rely on fees alone to provide this incentive. It is beyond the scope of this paper to have an in-depth discussion about inflation and money supply and the nature of money and currencies. Alias can be seen as a 'medium of exchange' and the aim is that Alias will be used to buy/sell goods and other fiat currency in its intended use in future online cash transfers. Economists discuss and debate this point but some level of inflation and value creation appears to be beneficial for long term adoption and will prevent Alias from the potential dangers of entering a 'deflationary spiral' relying on fees alone to sustain the Alias ecosystem. There doesn't seem to be a consensus among scholars of what might be '*the best*' strategy.

6.3 Total supply ('outstanding' amount) over 20 years

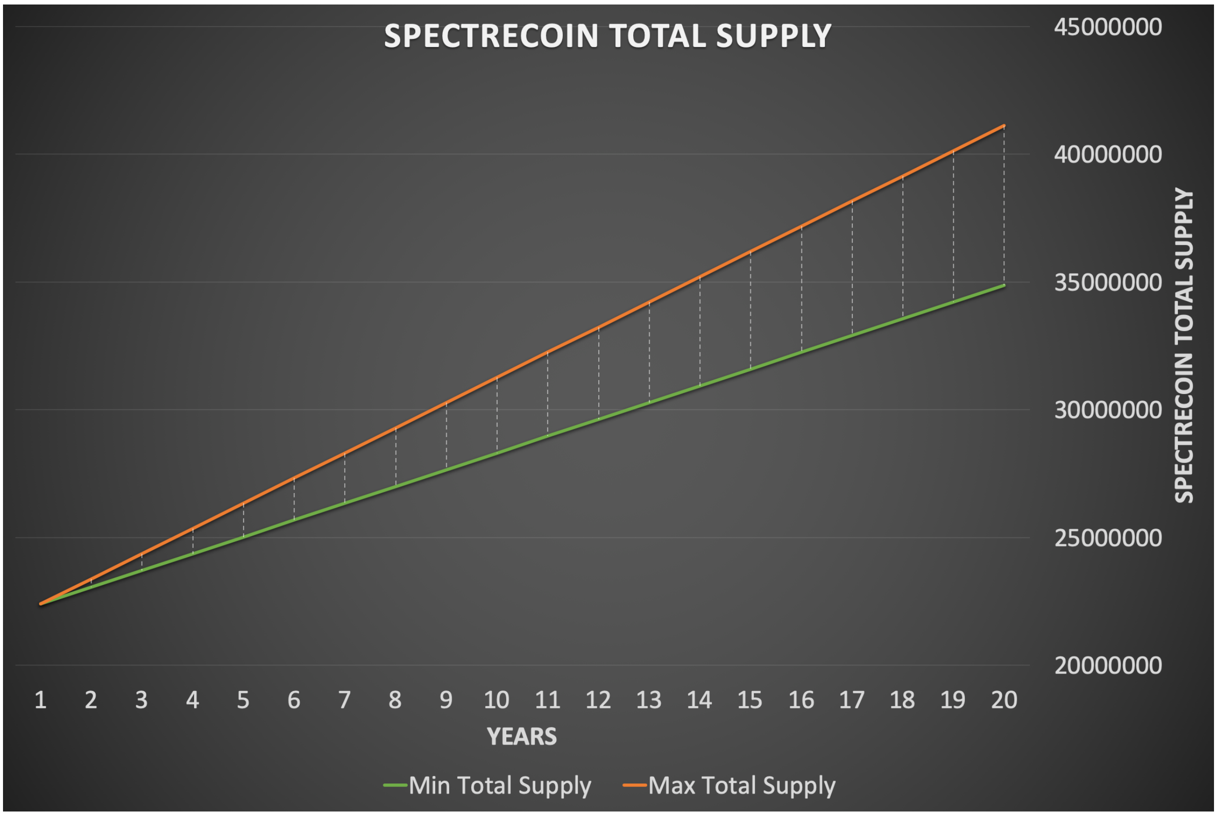


Figure 2: Min supply after 20 years = 34,883,000 ALIAS (public)+ALIAS (private) - Max supply after 20 years = 41,124,500 ALIAS (public)+ALIAS (private)

The block explorer[12] allows you to explore the Alias blockchain and you can see the total supply at any given time.

6.4 Maturity

The maturity calculations have changed such that the minimum maturity for staking and for spending stakes has been increased from 288 to 450 blocks. That means that the new maturity time is approximately $96seconds * 450 = 12hours$ and hence the 8-hour maturity rule for staking has been removed. Also, note that the updated maturity rule is also considered for all ATXOs used in the ring signature of the staked VIN.

[12] https://chainz.cryptoid.info/ALIAS/

6.5 Proof-of-Stake vs. Proof-of-Work

Alias uses both PoSv3 and APoS algorithms to keep the network consensus and to secure and confirm transactions. Both PoSv3 and APoS appear to be more resilient against various attacks that could be instigated against a Proof-of-Work (PoW) system like Bitcoin, Litecoin and DASH for example. It is also known that PoW systems are susceptible to so called 51% attacks where a sufficiently funded and motivated attacker can "*take control*" over the network and generate double spend transactions[13]. It is more difficult to attack a PoS system in this way as it would be unfeasible to acquire the majority of Alias in circulation and doing so would undermine the value and remove the incentive for the attack in the first place. There are obviously other attack vectors, such as the recently discovered so called "*Fake Stake*" attack against PoSv3[14]. This has since been fixed by the Alias developers and Alias is no longer susceptible to such an attack.

It is also well known that large PoW driven networks expend huge amounts of energy and appear to lead to some level of centralisation of mining power due to the huge expense involved in mining new blocks. In a recent research paper entitled "*The Bitcoin Mining Network - Trends, Composition, Average Creation Cost, Electricity Consumption & Sources*" by Christopher Bendiksen & Samuel Gibbons of CoinShares Research[15], it was found that the Bitcoin network expends more energy than the whole country of New Zealand.

The report calculated that the global Bitcoin mining industry draws 4.7GW of power every second. Hashing computations for the Proof-of-Work algorithm consumed 4.3GW, up 0.4GW from the last CoinShares report in November 2018. Based on these figures, researchers calculated **an annual consumption of 41TWh of electricity**. That's roughly 2.2TWh more than New Zealand – a country of 4.7M people – consumed in 2017, according to the country's Electricity Authority[16].

In comparison, Alias will run on a standard Raspberry Pi and in addition to all the privacy features, Alias is also truly eco-friendly, sustainable and '*green technology*'. The estimated loose upper bound, **annual consumption for a Raspberry Pi running Alias is 16.6kWh**.

Bitcoin network:
$41TWh = 41 * 1012Wh = 41,000,000,000,000Wh$

Alias node:
$16.6KWh = 16.6 * 103Wh = 16,600Wh * 1000(nodes) = 16,600,000Wh$

That means that the whole *Bitcoin network consumes almost* **2.5 million times more energy** *than what an imaginary Alias network would consume*, assuming 1000 Alias Raspberry Pi nodes.

[13] https://www.crypto51.app/

[14] https://medium.com/@dsl_uiuc/fake-stake-attacks-on-chain-based-proof-of-stake-cryptocurrenciesb8b05723f806

[15] https://coinshares.co.uk/#mailmunch-pop-792759

[16] https://www.ea.govt.nz/

In summary, the PoSv3 / APoS protocols are both potentially more secure, immensely more energy efficient and provide for better decentralisation. It is beyond the scope of this paper to discuss this further and there are various discussions around the internet if you are interested in the PoW vs. PoS debate.

In the next sections we will explore the different aspects of Alias in more detail. We start on the next page with a detailed introduction to the Alias privacy features before we go on to discuss Proof-of-Stake v3 (PoSv3) in detail. We move on to explain the privacy features and the new Anonymous-Proof-of-Stake (APoS) protocol.

7 Alias Privacy Features

Before we go on to explain some of the features and technologies of Alias in more detail, we will give you a short overview of the privacy technology used. The Alias blockchain is a '*dual-coin*' system or a system where two distinct types of transaction outputs can exist in the same block. Both non-private or standard UTXOs (*hereafter referred to as public coins or* **ALIAS (public)**) and private coins or ATXOs (*hereafter referred to as private coins or* **ALIAS (private)**) exists side by side in the blockchain. Transactions can be carried out with both public and private coins and they are interchangeable and independent. We introduce the terms **ALIAS (public)** for the public coins spent in standard UTXO based transactions and **ALIAS (private)** for the private coins spent in confidential ATXO based transactions using ring signatures. **ALIAS (public)** is used in the PoSv3 consensus mechanism and **ALIAS (private)** is used in the APoS consensus mechanism.

7.1 Dual Coin System

The private coin '*subsystem*' was inspired by the principles of the Zerocoin protocol which can be summarised as '*Anonymity by destruction / creation of basecoins*', i.e. destroy / consume one base unit, create a private token and create a proof that the user owns it and the system later agrees to re-create one base coin from that proof when requested. The Zerocoin protocol utilises a so called *zero-knowledge proof* (ZKP) to create the private coins and to prove ownership. Zerocoin is computationally intense and requires a trusted setup and we have recently seen that the Zerocoin protocol can be subject to certain attacks due to what can be described as flaws in the theory and implementation[17]. Some well-known Zerocoin based cryptocurrencies such as Zcoin, PIVX and NIX were forced to shut down their privacy system and work to implement fixes.

The Alias network instead employs dual-key stealth address cryptography to facilitate the creation of privacy maintaining **ALIAS (private)** coins on the blockchain whilst consuming **ALIAS (public)**. This is done without the trusted setup required for Zerocoin and without using the computationally intense Zerocoin cryptographic methods. Where the Zerocoin protocol use ZKP to anonymise and unlink the transactions, **the Alias network use ring signatures**[18][19].

The '*dual-coin*' system can be seen as a feature allowing for complete transparent transactions and network audit functions if needed but without any privacy indebted overhead such as resource intensive calculations. Privacy maintaining **ALIAS (private)** can ONLY be created by consuming **ALIAS (public)** at this time and the total supply on the network will always be transparent. There is no '*bleed through*' between the two types of transaction outputs and no compromise in privacy.

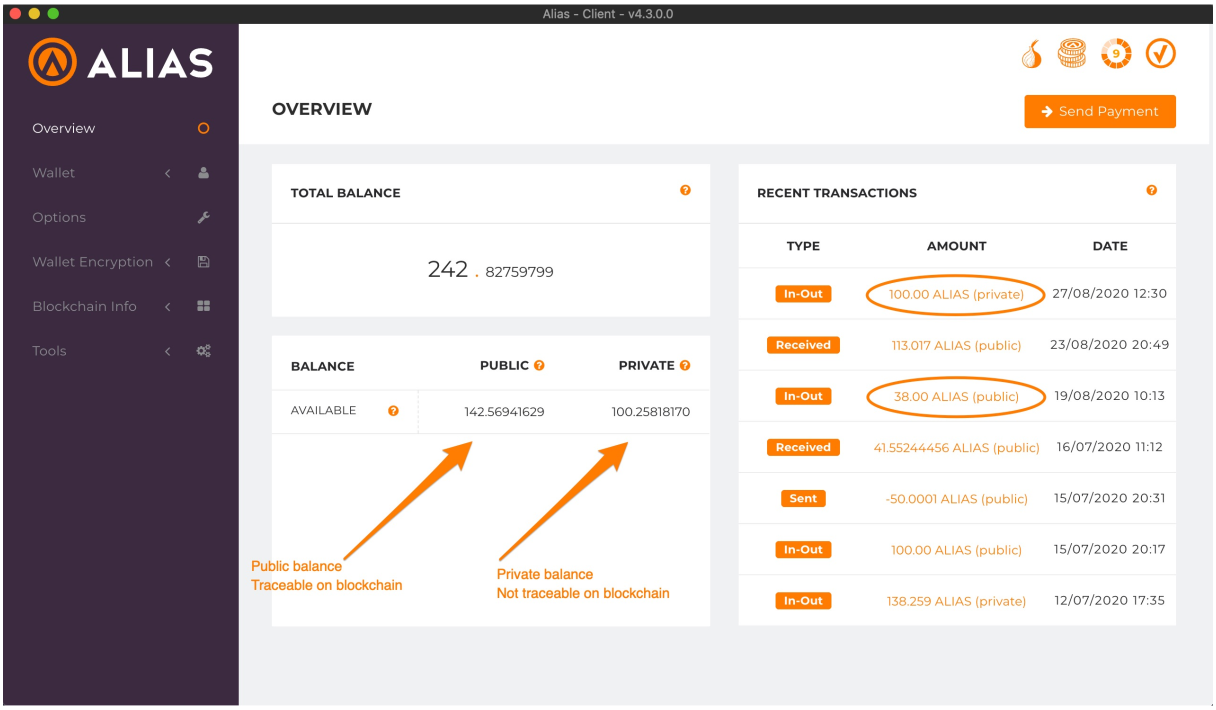
[17] https://www.chaac.tf.fau.eu/2018/04/12/zerocoinzcoinpivxzoinsmartcashhexxcoin-attack/

[18] https://people.csail.mit.edu/rivest/pubs/RST01.pdf/

[19] https://arxiv.org/pdf/1612.01188.pdf

7.2 ALIAS (public)/ALIAS (private) Conversion

Each user can convert (*non-private*) **ALIAS (public)** coins into (*private*) coins, **ALIAS (private)**. Users can then send **ALIAS (private)** to other users and split or merge the **ALIAS (private)** they own in any way that preserves the total value. Once **ALIAS (private)** has been created and matured the user will also be able to stake in private through the APoS protocol. Users can also convert **ALIAS (private)** back into **ALIAS (public)**, though in principle this is not necessary: all transactions can be made in terms of **ALIAS (private)**.



TOTAL BALANCE

242.82759799

BALANCE	PUBLIC	PRIVATE
AVAILABLE	142.56941629	100.25818170

Public balance
Traceable on blockchain

Private balance
Not traceable on blockchain

RECENT TRANSACTIONS

TYPE	AMOUNT	DATE
In-Out	100.00 ALIAS (private)	27/08/2020 12:30
Received	113.017 ALIAS (public)	23/08/2020 20:49
In-Out	38.00 ALIAS (public)	19/08/2020 10:13
Received	41.55244456 ALIAS (public)	16/07/2020 11:12
Sent	-50.0001 ALIAS (public)	15/07/2020 20:31
In-Out	100.00 ALIAS (public)	15/07/2020 20:17
In-Out	138.259 ALIAS (private)	12/07/2020 17:35

Figure 3: Your wallet will show your **ALIAS (public)** balance and your **ALIAS (private)** balance.

The core privacy technology used in Alias is:

Recipient privacy	Stealth addresses are used to protect the recipient's privacy. (ALIAS (private) only)
Sender privacy	Ring Signatures are used to protect the sender's privacy. (ALIAS (private) only)
IP address privacy	All Alias nodes run as Tor hidden services to protect users IP address.
Un-traceability	for each incoming transaction all possible senders are equiprobable.
Un-linkability	for two outgoing transactions it is impossible to prove the same receiver.(ALIAS (private) only)

Now let's have a look at the different features in some more detail in the following sections. We aim to briefly explain the background and nature of stealth addresses and ring-signatures and how this is used in Alias.

8 Stealth Addresses

Before we go on to explain what stealth technology is and how it is used in Alias we need to get some background and some knowledge about how standard UTXO transactions might be de-anonymised and why stealth technology came to be used to counter this issue. It will also become apparent why stealth address technology in itself is not sufficient to provide reasonable privacy.

The Alias blockchain is based on the design from Bitcoin Core (*except the consensus mechanisms*) and in both systems there are $1.46 * 10^{48}$ possible receiving addresses. This is an extremely large number and it would give every person on Earth $2.05 * 10^{38}$ different receiving addresses to use. The fact that it is possible to re-use an address more than once can be considered a fluke and is not by design.

First, let's have a quick look at how a UTXO blockchain might be deanonymized. The three major factors that can reduce privacy for the user and are exploitable through transaction graph analysis are *address re-use, change addresses and the merging of outputs*.

Address re-use is treating ALIAS (public) addresses like a bank account where a single address is used for multiple transactions. ALIAS (public) addresses are not designed to be used in this way. There are in fact no restrictions on the number of ALIAS (public) addresses one person can use and for each transaction a new ALIAS (public) address should be created.

When addresses are re-used, all other transactions performed by that address can be seen by examining the blockchain. If you are aware of a transaction made by a person of interest and that transaction comes from the same address by which this person receives all their payments, then their balances can easily be determined. You will also be able to look back at the history of that address, following the chains of transactions, to ascertain what other information can be extracted.

Address re-use also weakens the security of the coins stored in those addresses. Transaction signing requires 256 bytes of random data (r-value) so that the private key cannot be reverse engineered. If the *r-value* is not truly random then the private key can be determined, which can be used to sign other transactions for that particular address. This attack can be negated by not re-using addresses, as once a transaction is signed from an address, it remains empty.

Furthermore, each input in a standard transaction must be a full UTXO from a previous transaction as UTXO's cannot be partially spent. This means that if you spend / send less than a full UTXO you will generate an output that is your change address. Therefore, an attacker examining the blockchain may generally assume that one output in any transaction belongs to the creator of the transaction.

Also, if a transaction is generated where two or more UTXOs are pooled together to create the total input required an assumption can be made that the addresses merged together belongs to the same person.

Let's then introduce **Stealth Address techniques** which allow public keys appearing in the blockchain to be fully disconnected from "*stealth*" public keys which can be publicised by a payee[20]. The public "*stealth*" keys publicised serve as a "*master public key*" from which "*ephemeral public keys*" are derived. The "*stealth*" public key is never recorded in the blockchain. This enables the payee to receive infinite un-linkable payments by publicising only one stealth address. The problem of address re-use is therefore solved[21]. **A Stealth address therefore is a privacy technique that protects the privacy of the recipient.** The first stealth address technique was invented by a user known as '*bytecoin*' in 2011 in the Bitcointalk forum. Later improvements to stealth tech were proposed by van Saberhagen in 2013/14 and by Peter Todd in 2014.

The original stealth technology had various problems and on 02/08/2014 one of the ShadowCash developers known as '*rynomster*' announced a first fully working implementation of stealth address technology known as "*dual-key stealth addresses*" that solved some of the issues in previous proposals. A "*dual-key stealth address*" has two public keys and solves certain problems associated with previous schemes. For a full and in-depth explanation of stealth addresses see the paper cited below.

(*Courtois N. and Mercer R. (2017). Stealth Address and Key Management Techniques in Blockchain Systems. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy ISBN 978-989- 758-209-7, pages 559-566. DOI: 10.5220/0006270005590566*)

The use of dual-key stealth addresses provide privacy for the receiver of funds and introduces various forms of un-linkability:

- It is hard to link different public keys/addresses of the same user,
- It is hard to link different transactions of the same user,
- It is hard to link the sender to the recipient.

8.1 Limitations of Stealth Addresses

As mentioned above, stealth addresses generate a new standard address for every payment but if you receive for example 10 transactions using your stealth address you will have 10 UTXOs available to form further transactions. If you then use some or all of the UTXOs to form a transaction an observer will be able to link the UTXOs together and assume that they all belong to one user (*merging of outputs*). Furthermore, an attacker could create a number of dust transactions with a stealth address and then monitor the blockchain to see if the user ever joins those UTXOs together or with others, in order to make an input to a higher value transaction in the future. Blockchain analysis can easily link this. **This is the reason why any cryptocurrency that relies on stealth addresses ONLY is not private.**

[20] https://www.scitepress.org/Papers/2017/62700/62700.pdf

[21] http://www.nicolascourtois.com/bitcoin/paycoin_privacy_monero_6_ICISSP17.pdf

8.2 Anonymous Alias Conversion

Dual-key Stealth technology is used in the process to create anonymous **ALIAS (private)** by consuming the equivalent value of **ALIAS (public)**. The creation of **ALIAS (private)** involves the creation of an ATXO with a bundled one-time key-pair that will allow that ATXO to be '*spent*' by providing a valid ring signature using your remaining public key corresponding to the ATXO you previously created and the corresponding '*keyimage*'[22]. The fact that an ATXO has been '*spent*' is only known to the sender and an observer cannot determine if the ATXO has been '*spent*'.

8.3 Keyimage

The '*keyImage*' is the result of a cryptographic one-way function derived from a user's one-time keypair. The '*keyimage*' is unique to the ATXO contributing the value to the new ATXO being created in an anonymous transaction. The '*keyimage*' is then recorded in the blockchain to prevent double spends, but without revealing which ATXO is the value-contributing member in the ring signature. Although the '*keyimage*' is recorded in the blockchain it cannot be reverse engineered due to the one-wayness of the cryptographic function that generated it. The calculation of the '*keyimage*' includes the users private key associated with the ATXO being '*spent*'. Hence, if the user tries to spend the same ATXO again the same '*keyimage*' will be generated and the system will reject the transaction as that '*keyimage*' has been seen before.

The following ALIAS (private) denominations are possible:
$1000, 500, 400, 300, 100, 50, 30, 10, 5, 4, 3, 1,$
$0.5, 0.4, 0.3, 0.1, 0.0(000000)5, 0.0(000000)4, 0.0(000000)3, 0.0(000000)1$

We have seen how the use of stealth address tech can be used to solve the problem of address re-use and to create un-linkable transactions. Now, we still have the problem of UTXOs being linked together in future transactions. To resolve this issue Alias employs the use of ring signatures in transactions formed by ATXO outputs using **ALIAS (private)**.

[22] https://monero.stackexchange.com/questions/2883/what-is-a-key-image

9 Ring Signatures

In a standard UTXO transaction the sender signs the transactions using his/her private key and the signatory can be explicitly determined and identified. In cryptography, **a ring signature** is a type of digital signature that can be performed by any member of a defined group of users that each have the required keys. A distinctive **ring signature** is produced through a process that combines the keys of all possible signers and other values and which are then subject to a hash function.

In cryptography a ring signature is a form of a *non-interactive zero knowledge proof*[23]. In layman's terms, what this means is simply that you can prove the correctness of a statement/transaction to a verifier without leaking any additional information by just using a shared common reference string (*public key*). This system must include cryptographic completeness, soundness and zero-knowledge.

Completeness means that if the statement is correct, then the verifier will always accept. Soundness is a property of such a system that requires that no prover can make the verifier accept a false or incorrect statement. If the statement is incorrect or false, then the verifier will always reject. The last part is zero knowledge. It is not possible to gain any extra information from the proof itself for any malicious verifier except for the correctness of the statement.

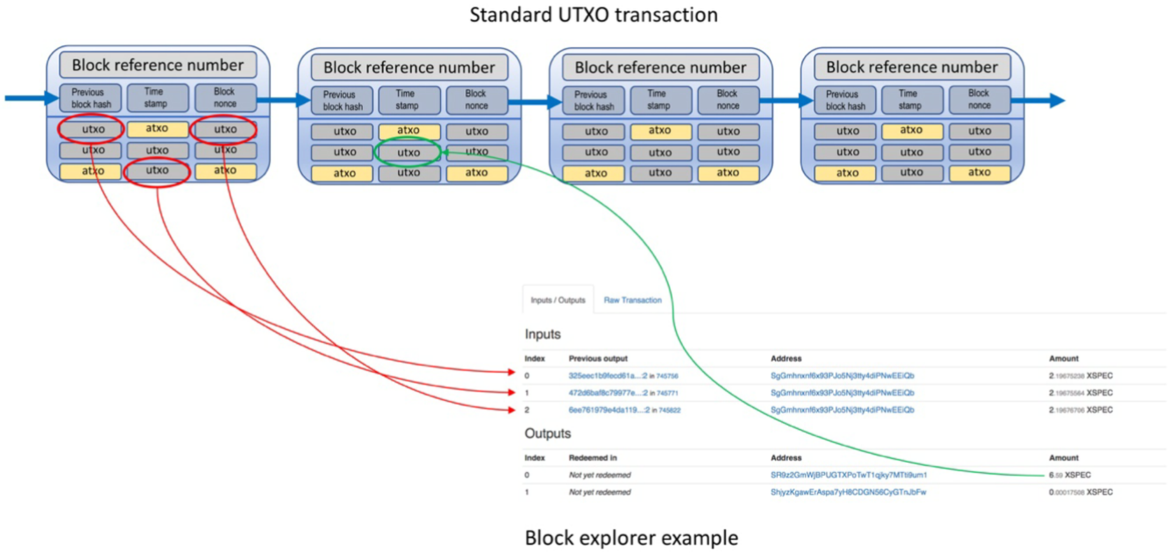
This offers a group member a level of anonymity not attainable through generic digital signature schemes. This is a property known as '*plausible deniability*', or anonymity with respect to an anonymity set. With a ring size of 10 for example there are 10 possible signatories, i.e. 10 public keys and an observer cannot determine which one corresponds to the **ALIAS (private)** spent in the transaction. This is only known to the sender. This protects the privacy of the sender. With every transaction using a ring-signature the network '*transactional entropy*' increases and it becomes increasingly hard to link input/output on the blockchain.

Look at it like this; scattered along the Alias blockchain are ATXOs of various denominations from 1000 to 0.00000001 ALIAS (private). These ATXOs may be spent or unspent but this cannot be determined by an observer. The proof of an ATXO being spent is formed on an ad-hoc basis through the creation of a '*keyimage*' and there is nothing contained within the data of the ATXO itself or the transaction data written to the blockchain that signifies if it has ever been '*spent*'. In a standard UTXO transaction on the other hand an observer can explicitly determine that an UTXO has indeed been spent to create a new UTXO.

See the illustrations on the next page to visualise the difference between an UTXO and ATXO.

[23] https://en.wikipedia.org/wiki/Non-interactive_zero-knowledge_proof

9.1 Standard UTXO transaction



Inputs

Index	Previous output	Address	Amount
0	325eec1b9fecd61a...:2 in 745756	SgGmhnxnf6x93PJo5Nj3tty4diPNwEEiQb	2.19675238 XSPEC
1	472d6baf8c79977e...:2 in 745771	SgGmhnxnf6x93PJo5Nj3tty4diPNwEEiQb	2.19675564 XSPEC
2	6ee761979e4da119...:2 in 745822	SgGmhnxnf6x93PJo5Nj3tty4diPNwEEiQb	2.19676706 XSPEC

Outputs

Index	Redeemed in	Address	Amount
0	*Not yet redeemed*	SR9z2GmWjBPUGTXPoTwT1qky7MTti9um1	6.59 XSPEC
1	*Not yet redeemed*	ShjyzKgawErAspa7yH8CDGN56CyGTnJbFw	0.00017508 XSPEC

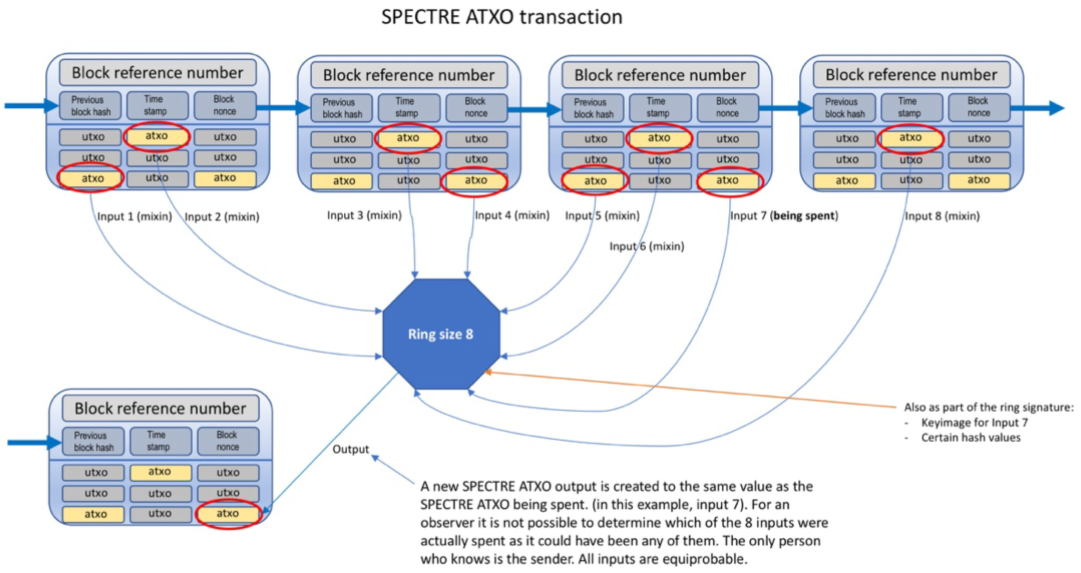
Figure 4: Standard UTXO transaction

On the blockchain there is a **direct** correlation between the inputs and the outputs, and all the transactions can be traced back to the genesis transactions. This is still the case even if a mixing strategy is used, such as in DASH. There are increasingly sophisticated methods to analyse blockchain data and this is a growth industry. You should consider any standard UTXO transactions to be non-anonymous and public, whether with Alias or Bitcoin.

9.2 Anonymous ATXO transaction

In the Alias software we talk about ring sizes and this refers to the group or set of possible signers. So, in the example below, we have a ring size of 8 which simply means that amongst the 8 public keys that form part of the digital signature, 7 are so called '*mixins*' or chaff or decoys and only 1 is the public key corresponding to the ALIAS (private) being spent. When conducting an anonymous transaction, we use ring signatures to hide spent output in a set of the same denomination.

Anonymous transactions in Alias can be said to have levels of '*transactional entropy*' as there is an interface between the '*public*' and the '*anonymous*' coins. Entropy level 0 can be said to be at the interface between ALIAS (public) >> ALIAS (private) and between ALIAS (private) >> ALIAS (public). Once ALIAS (private) has been created from ALIAS (private), i.e. an ATXO used as an input to create a new ATXO we can say that this is entropy level 1 as the freshly created ATXO has no public UTXO origin. Once these ATXOs are used to create further ATXOs this would be entropy level 2 and so on. The entropy increases with every level, IF AND ONLY IF a minimum ring size is used and the ATXOs have not been part of any ring size 1 transaction.



A new SPECTRE ATXO output is created to the same value as the SPECTRE ATXO being spent. (in this example, input 7). For an observer it is not possible to determine which of the 8 inputs were actually spent as it could have been any of them. The only person who knows is the sender. All inputs are equiprobable.

Figure 5: ALIAS (private) ATXO transaction

9.3 Minimum Ring Size

A minimum ring size of 10 is being enforced as part of the consensus since Alias v3.x.

Any ATXO used in a ring size 1 transaction will be marked as '*compromised*' and will not be selected as a '*mixin*' in any future ring signature transaction.

We have seen how dual-key stealth address techniques are used to create ATXOs on the Alias blockchain with bundled one-time key pair that can only be 'spent' by providing a valid ring signature. In this way Alias protects the privacy of both the sender and the receiver.

9.4 Ring-Signature formula

Alias uses a ring-signature formula developed by Dr. Adam Back. He created a formula based on the traceable ring signature described in the CryptoNote white-paper[24] using maths based on the paper "*1-out-of-n Signatures from a Variety of Keys*"[25] by Abe, Ohkubo and Suzuki. He was able to create a linkable ring signature which tends to be ½ of the size of the CryptoNote ring signature as the signature is 3+n values vs. 2+2n values.

[24] https://cryptonote.org/whitepaper.pdf

[25] https://www.researchgate.net/publication/221326919_1-out-of-n_Signatures_from_a_Variety_of_Keys

Dr. A. Back showed how to add traceability in a way that made it compatible with CryptoNote[26]. The original description contains 4 parts (**KEYGEN**, **SIG**, **VERIFY**, **LINK**) and is summarized here for comparison:

Definitions:

q: a prime number; $q = 2^{255} - 19$

d: an element of $\mathbb{F}_q$;. $-121665/121666$

E: an elliptic curve equation; $-x^2 + y^2 = 1 + dx^2y^2$

G: a base point; $G = (x, -4/5)$

l: a prime order of the base point; $l = 2^{252} + 27742317777372353535851937790883648493$

$\mathcal{H}_s$: a cryptographic hash function $\{0,1\}^* \to \mathbb{F}_q$

$\mathcal{H}_p$: a deterministic hash function $E(\mathbb{F}_q) \to E(\mathbb{F}_q)$

KEYGEN: The signer selects a secret key $x \in [1, l-1]$ and calculates a public key P, as well as the key-image I with the following equations:

$$\begin{aligned} P &= xG \\ I &= x\mathcal{H}_p(P) \end{aligned} \tag{1}$$

SIG: The signer selects a set S' of n other users public keys P_i, and adds his own public key to the set.

$$\begin{aligned} S &= S' \cup \{P_s\} \\ &= \{P_i\}, i \in [0..n] \end{aligned} \tag{2}$$

Note that the public key $P_{i=s}$ is the signer's own public key, while the others are random selected public keys that will be used in the ring signature. The index s is the signer secret index. The signer then pics $n+1$ random private keys q_i and n private keys w_i.

$$\begin{aligned} q_i &\in [1...l], i \in [0...n] \\ w_i &\in [1...l], i \in [0...n], i \neq s \end{aligned} \tag{3}$$

Now he computes:

$$L_i = \begin{cases} q_iG, & \text{if } i = s \\ q_iG + w_iP_i, & \text{otherwise} \end{cases}$$

and similarly

$$R_i = \begin{cases} q_i\mathcal{H}_p(P_i), & \text{if } i = s \\ q_i\mathcal{H}_p(P_i) + w_iI, & \text{otherwise} \end{cases}$$

[26] https://bitcointalk.org/index.php?topic=972541.msg10619684

Now he calculates the challenge c:

$$c = \mathcal{H}_s(m, L_1, ..., L_n, R_1, ..., R_n) \tag{4}$$

And finally the response

$$c_i = \begin{cases} w_i, & \text{if } i \neq s \\ c - \sum_{i=0}^{n} c_i \quad mod \quad l, & \text{otherwise} \end{cases}$$

and similarly

$$r_i = \begin{cases} q_i, & \text{if } i \neq s \\ q_i - c_s x \quad mod \quad l, & \text{otherwise} \end{cases}$$

The resulting signature σ is then

$$\sigma = (I, c_1, ..., c_n, r_1, ..., r_n)$$

One can immediately see that the signature σ from the CryptoNote white-paper has $2(n+1)$ values for c_i and r_i.

VERIFY: The first step is to apply the inverse transformation.

$$\begin{aligned} L'_i &= r_i G + c_i P_i \\ R'_i &= r_i * \mathcal{H}_p(P_i) + c_i I \end{aligned} \tag{5}$$

A verifier now checks that the following condition holds:

$$\sum_{i=0}^{n} c_i \stackrel{?}{=} \mathcal{H}_s(m, L'_0, ..., L'_n, R'_0, R'_n) \tag{6}$$

LINK: This step is only applied when the previous step succeeds. The verifier uses a set $\mathcal{I}$ of previous signatures and rejects the signature $\sigma = (I, ...)$ if $I \in \mathcal{I}$.

Dr. Adam Back's variation, and the version of the ring-signature algorithm used by Alias, is as follows:

KEYGEN: Like above, pick a random private key $x \in [1..l-1]$ and calculate:

$$\begin{aligned} P &= xG \\ I &= x\mathcal{H}_p(P) \end{aligned} \tag{7}$$

SIG: Pick n random public keys P_i from other users, generate n random private keys $s_i \in [1...l-1]$ as well as a private key $\alpha \in [1..l-1]$ for the signer with index $j, 0 <= j <= n$. This gives you the following structure:

$$\begin{aligned} \textbf{public keys} : [P_0, ...P_j, ..., P_n] \\ \textbf{private keys} : [s_0, ...s_j, ..., s_n] \end{aligned} \tag{8}$$

NOTE that s_j is to be calculated (see below) and P_j is the public key of the signer. The next step is to calculate the parameters c_i recursively yielding the vector $[c_0, ..., c_j, ..., c_n]$.

$$\begin{aligned} c_{j+1} &= \mathcal{H}_s(P_1, ..., P_n, \alpha G, \alpha \mathcal{H}_p(P_j)) \\ c_{j+2} &= \mathcal{H}_s(P_1, ..., P_n, s_{j+1}G + c_{j+1}P_{j+1}, s_{j+1}\mathcal{H}_p(P_{j+1}) + c_{j+1}I_j) \\ &... \\ c_j &= \mathcal{H}_s(P_1, ..., P_n, s_{j-1}G + c_{j-1}P_{j-1}, s_{j-1}\mathcal{H}_p(P_{j-1}) + c_{j-1}I_j) \end{aligned} \tag{9}$$

The formula shows that we start to calculate at the position $j + 1$, where j represents the signer's index. To keep the index i of c_i within the allowed range, we have to take the mod of the number of signers. Effectively we calculate the sequence $c_{j+1}, c_{j+2}, ..., c_n, c_0, c_1, ...c_j$

Next is to find the s_j value: Adam argues that since $\alpha G = s_j G + c_j P_j$, then $\alpha = s_j + c_j x_j$ or $s_j = \alpha - c_j x_j \quad mod \quad n$.

The signature σ based on this approach is than given by $\sigma = (m, I_j, c_1, s_1, ..., s_n)$

VERIFY:
We compute the following equation $\forall_{i=1..n}$

$$\begin{aligned} e_i &= s_i G + c_i P_i \\ E_i &= s_i \mathcal{H}_p(P_i) + c_i I_j \\ c_{i+1} &= \mathcal{H}_s(P_1, ..., P_n, e_i, E_i) \end{aligned} \tag{10}$$

Then we check if $c_{n+1} = c_1$.

LINK: Like above, we reject duplicate I_j values.

As we can see from the new signature $\sigma = (m, I_j, c_1, s_1, ..., s_n)$, this version of the ring signature uses only 1/2 of the size of the Cryptonote ring signature.

9.5 The 'All_Spent' Dilemma

A '*special*' situation could occur where all the ATXOs available for a certain denomination are spent except for your own ATXO to be used in a transaction. We are referring to this situation as the **'ALL_SPENT' dilemma** and although this appears to be a very low probability situation in V3 it could have dire consequences for privacy and compromise a number of ATXOs. Let's first explain this dilemma in some detail:

A valid ring signature (*assume ring size 10*) needs: **(1)** An unspent ATXO of a certain denomination to be used/spent in the transaction, and **(2)** Nine (9) '*mixins*' of the same denomination (*spent or unspent*). These '*mixins*' (ATXOs) of the same value as the one being spent provides '*plausible deniability*' with regards to the sender. The sender could own any one of the ten ATXOs used in the ring-signature and an observer can not determine which one of the 10 ATXOs forming the ring-signature is being spent. This is at the core of ring-signature privacy and needs to be preserved.

ATXOs of each denomination are "*scattered*" along the Alias blockchain and exist in various blocks where they were once created and they can all be used as '*mixins*' in a ring-signature whether they have ever been spent or not. Each ATXO is a '*unique unit of data*' identified by its associated '*public key*'. However, only the owner of an ATXO can determine if an ATXO value has been spent or not as this requires the corresponding '*private key*'.

The ATXO picking algorithm for a ring-signature selects 9 '*mixins*' at random from the available pool of ATXOs. In each block there could be a number of ATXOs of different denominations (*spent or unspent*) that could be selected as the 9 '*mixins*'.

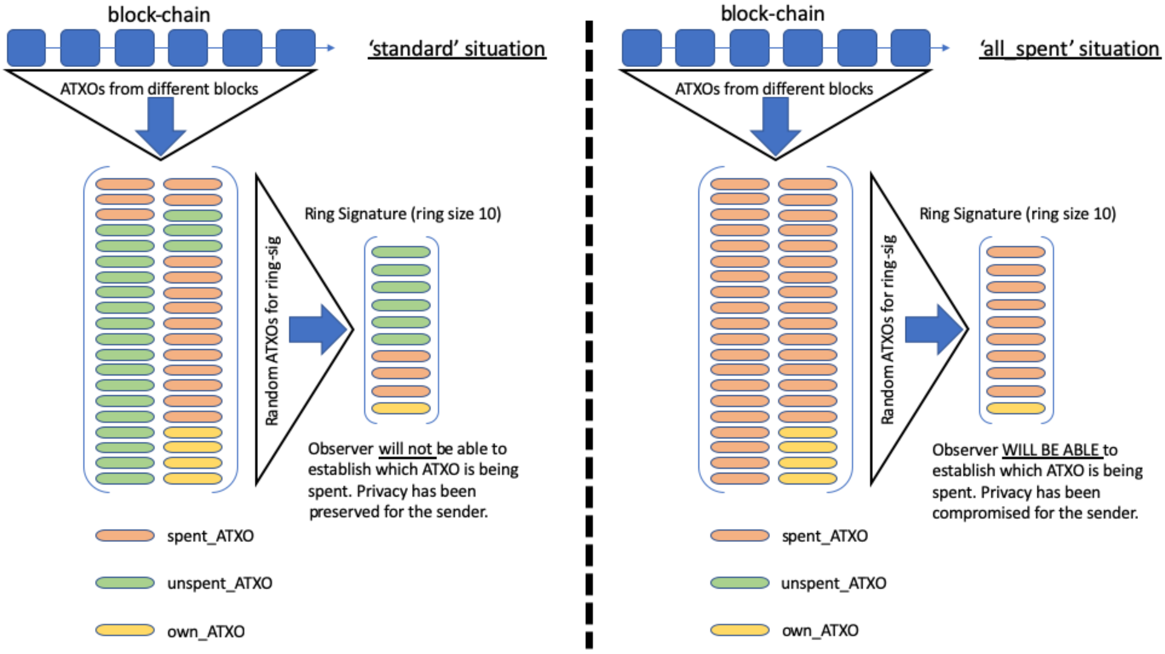
Any observer will be able to establish the following: **(1)** In each valid ring-signature on the blockchain there will be one and only one ATXO of a certain denomination being spent. **(2)** In each valid ring-signature there will be nine '*mixins*' used of the same denomination. **(3)** The observer will be able to count the number of existing ATXOs for each denomination by scanning the blockchain. **(4)** The observer will be able to count the number of ATXOs for each denomination that has been spent by counting the number of valid ring-signatures where this denomination has been used.

As a result, if the 'ALL_SPENT' situation occurs, i.e.:
$(number\_of\_\_existing\_ATXOs = number\_of\_spent\_ATXOs)$

An observer will be able to categorically determine that each of the ATXOs used as a '*mixin*' in the ring-signature has previously been spent. The sender's privacy has been compromised as the '*real*' input ATXO can be identified, and we no longer have the '*plausible deniability*' afforded by the ring-signature. It is important to point out the following regarding this issue:

- This only occurs when ALL the ATXOs of a denomination have been spent and a new transaction is created using the depleted denomination.
- The privacy of the transaction spending the last unspent ATXO, although creating an 'ALL_SPENT' situation is not compromised.
- If the 'ALL_SPENT' occurs, it would only compromise transaction created after the 'ALL_SPENT'. All the previous transactions would still retain their privacy.

9.5.1 ’ALL_SPENT’ illustration


block-chain
‘standard’ situation
ATXOs from different blocks
Random ATXOs for ring-sig
Ring Signature (ring size 10)
spent_ATXO
unspent_ATXO
own_ATXO
block-chain
‘all_spent’ situation
ATXOs from different blocks
Random ATXOs for ring-sig
Ring Signature (ring size 10)
spent_ATXO
unspent_ATXO
own_ATXO

Observer will not be able to establish which ATXO is being spent. Privacy has been preserved for the sender.

Observer WILL BE ABLE to establish which ATXO is being spent. Privacy has been compromised for the sender.

Figure 6: ALL_SPENT problem

9.5.2 Solutions

A range of measures are being implemented to negate the so called *‘ALL_SPENT’ dilemma*. We realised that this is only likely to occur in a situation where there is a very limited supply of ATXOs of a certain denomination.

Therefore, the main approach to solve this problem is to make sure that there is always a sufficient supply of ATXOs of varying denominations. We have therefore introduced an advanced ATXO balancer algorithm. This algorithm will measure the number of ATXO existing for each denomination and either seek to consolidate ATXO values or split ATXO values as part of the staking transaction in APoS. This will ensure that there is always a sufficient supply of ATXO to act as ‘*mixins*‘.

In addition a new algorithm has been implemented to detect an ‘*ALL_SPENT*‘ situation and if this situation occurs the network will ‘*remember*‘ the block height (*block number*) and the picking algorithm will no longer pick ‘*mixins*‘ from below that block height. This ensures that users get the full benefit of the privacy the ring-signature offers.

9.6 The ‘ATXO_Set’ Dilemma

We can say that all the ATXOs created in the same transaction are part of an ‘*ATXO_Set*’, i.e. they will forever be linked to each other as they were created at the same time and it can be assumed that they all belong to the same user. There will always be multiple outputs per transaction because of the denomination system, because amounts have to be split into discrete values.

The following dilemma may then occur; the ATXO picking algorithm (as it is now) could select ATXOs from an '*ATXO_SET*' as '*mixins*' in a ring-signature and this could potentially compromise privacy by making the signature more susceptible to deductive analysis. In other words, an observer could be able to determine which of the '*mixins*' are fake.

9.6.1 Solutions

New algorithms have been created to negate these issues and strengthen the privacy of the network. This way depending on the random pick of the initial '*mixins*' and the denominations to fill, there will be a random amount of '*mixins*' from the same transaction in the final transaction.

This will also ensure that each output is only used once as a '*mixin*' in one of the ring-signatures (VINs). Something which was previously only assured by chance. Same '*mixin*' in different VINs can only be fake.

- When the first 9 '*mixins*' for an output are chosen, it is completely random, but is ensured that all '*mixins*' come from different transactions.
- When '*mixins*' for the second ring-signature are chosen, there is a 33% chance that the algorithm tries to choose outputs from the transaction chosen as '*mixins*' for the first ring-signature. If no outputs can picked that way, a new random output is chosen and the corresponding transaction is added as new source of '*mixins*'.
- Repeat.

We believe that these issues also exist or existed in Monero and other CryptoNote based systems and may have been described first in a paper entitled *"A Traceability Analysis of Monero's Blockchain"* [27] and in particular in chapter: 5.2 Heuristic II: Leveraging Output Merging in this paper.

[27] https://eprint.iacr.org/2017/338.pdf

10 The Proof-of-Stake (PoSv3) Protocol

The following section, in its entirety, has been taken directly from, and is based on a blog post by Qtum[28] developer '*Earlz*' from 27/07/2017 called *"The missing explanation of Proof of Stake Version 3"*[29] and is the best-known source of information on the PoSv3 protocol. The Alias PoSv3 implementation adheres completely to this protocol description for both blocks and transactions and so the explanation from '*Earlz*' can be used for Alias.

The core concept of Proof-of-Stake (PoS) is very similar to that of Proof-of-Work (PoW), a lottery. However, the big difference is that, in PoS, it is not possible to "get more tickets" to the lottery by simply changing some data in the block and generate a new ticket. Instead of the "block hash" being the lottery ticket to judge against a target, PoS invents the notion of a "kernel hash". The 'kernel hash' is composed of several pieces of data that are not readily changeable in the current block. And so, because the miners do not have an easy way to modify the 'kernel hash', they cannot simply iterate through a large amount of hash values like in PoW. PoS blocks also add many additional consensus rules in order to realise it's objectives. First, unlike in PoW, the coinbase transaction (the first transaction in the block) must be empty and the reward must be 0 tokens. Instead, to reward stakers, there is a special "staking transaction" which must always be the 2nd transaction in the block.

A 'staking transaction' is defined as any transaction that:

- Has at least one valid VIN
- It's first VOUT must be an empty script
- It's second VOUT must not be empty

Furthermore, 'staking transactions' must abide by these rules to be valid in a block:

- The second VOUT must be either a pubkey (not pubkeyhash!) script, or an OP_RETURN script that is unspendable (data-only) but stores data for a public key
- The timestamp in the transaction must be equal to the block timestamp
- The total output value of a stake transaction must be less than or equal to the total inputs plus the PoS block reward plus the block's total transaction fees. output <= (input + block_reward + tx_fees)
- The first spent VIN's output must be confirmed by at least 450 blocks (in other words, the coins being spent must be at least 450 blocks old)
- Though more VINs can used and spent in a 'staking transaction', the first VIN is the only one used for consensus parameters

[28] https://qtum.org/en

[29] http://earlz.net/view/2017/07/27/1904/the-missing-explanation-of-proof-of-stake-version

These rules ensure that the stake transaction is easy to identify and ensures that it gives enough info to the blockchain to validate the block. The empty VOUT method is not the only way staking transactions could have been identified, but this was the original design from Sunny King[30] and has worked well enough. Now we understand what a 'staking transaction' is, and what rules such transaction must abide by.

The next part is to understand the rules for PoSv3 blocks:

- Must have exactly 1 staking transaction.
- The staking transaction must be the second transaction in the block.
- The coinbase transaction must have 0 output value and a single empty VOUT
- The block timestamp must have its bottom 4 bits set to 0 (referred to as a "mask" in the source code). This effectively means the blocktime can only be represented in 16 second intervals, decreasing its granularity
- The version of the block must be 7
- A block's "kernel hash" must meet the weighted difficulty for PoS
- The block hash must be signed by the public key in the staking transaction's second VOUT. The signature data is placed in the block (but is not included in the formal block hash)
- The signature stored in the block must be "LowS", which means consisting only of a single piece of data and must be as compressed as possible (no extra leading 0s in the data, or other opcodes)
- Most other rules for standard PoW blocks apply (valid merkle hash, valid transactions, timestamp is within time drift allowance, etc)

There are a lot of details here that we'll cover in a bit. The most important part that really makes PoS effective lies in the "kernel hash". The kernel hash is used similar to PoW (if hash meets difficulty, then block is valid).

However, the kernel hash is not directly modifiable in the context of the current block. We will first cover exactly what goes into these structures and mechanisms, and later explain why this design is exactly this way, and what unexpected consequences can come from minor changes to it.

[30] https://whitepaperdatabase.com/peercoin-ppc-whitepaper/

10.1 The Proof of Stake Kernel Hash

The kernel hash specifically consists of the following exact pieces of data (in order):

- Previous block's "stake modifier" (more detail on this later)
- Timestamp from "prevout" transaction (the transaction output that is spent by the first VIN of the staking transaction)
- The hash of the prevout transaction
- The output number of the prevout (ie, which output of the transaction is spent by the staking transaction)
- Current block time, with the bottom 4 bits set to 0 to reduce granularity. This is the only thing that changes during staking process

The stake modifier of a block is a hash of exactly:

- The hash of the prevout transaction in PoS blocks, OR the block hash in PoW blocks.
- The previous block's stake modifier (the genesis block's stake modifier is 0)

The only way to change the current kernel hash (in order to mine a block), is thus to either change your "*prevout*", or to change the current block time.

A single wallet typically contains many UTXOs. The balance of the wallet is basically the total amount of all the UTXOs that can be spent by the wallet. This is of course the same in a PoS wallet. This is important though, because any output can be used for staking. One of these outputs are what can become the "*prevout*" in a staking transaction to form a valid PoS block.

Finally, there is one more aspect that is changed in the mining process of a PoS block. The difficulty is weighted against the number of coins in the staking transaction. The PoS difficulty ends up being twice as easy to achieve when staking 2 coins, compared to staking just 1 coin.

If this were not the case, then it would encourage creating many tiny UTXOs for staking, which would bloat the size of the blockchain and ultimately cause the entire network to require more resources to maintain, as well as potentially compromise the blockchain's overall security.

So, if we were to show some pseudo-code for finding a valid kernel hash now, it would look something like this:

```
while(true){
    foreach(utxo in wallet){
        blockTime = currentTime - currentTime % 16
        posDifficulty = difficulty * utxo.value
        hash = hash(previousStakeModifier << utxo.time << utxo.hash <<
utxo.n << blockTime)
        if(hash < posDifficulty){
            done
        }
    }
    wait 16s -- wait 16 seconds, until the block time can be changed
}
```

This code isn't so easy to understand as our PoW example, so I'll attempt to explain it in plain English. Do the following over and over for infinity:

```
Calculate the blockTime =
'current time' - 'itself' (mod 16)
(modulus is like dividing by 16, but taking the remainder)

Calculate the posDifficulty =
the 'network difficulty' * 'number of coins held in a UTXO'

Cycle through for each UTXO in the wallet.

With each UTXO calculate a SHA256 hash using:
'previous block's stake modifier'
'some data from the the UTXO' + 'the blockTime'

Compare this hash to the posDifficulty.

If hash is less than posDifficulty, then 'kernel hash' is valid

You can then create a new block.

After going through all UTXOs, if no valid 'kernel hash'

Then wait 16 seconds and do it all over again.
```

Now that we have found a valid kernel hash using one of the UTXOs we can spend, we can create a staking transaction. This staking transaction will have 1 VIN, which spends the UTXO we found that has a valid kernel hash. It will have (*at least*) 2 VOUTs. The first VOUT will be empty, identifying to the blockchain that it is a staking transaction. The second VOUT will either contain an OP_RETURN data transaction that contains a single public key, or it will contain a pay-to-pubkey script. The latter is usually used for simplicity but using a data transaction for this allows for some advanced use cases (such as a separate block signing machine) without needlessly cluttering the UTXO set.

Finally, any transactions from the mempool are added to the block. The only thing left to do now is to create a signature, proving that we have approved the otherwise valid PoS block. The signature must use the public key that is encoded (either as pay-pubkey script, or as a data OP_RETURN script) in the second VOUT of the staking transaction. The actual data signed in the block hash. After the signature is applied, the block can be broadcast to the network. Nodes in the network will then validate the block and if it finds it valid and there is no better blockchain then it will accept it into its own blockchain and broadcast the block to all the nodes it has connection to.

It is highly recommended that you read the blog post by '*Earlz*'[31] and also the associated white papers for PoS[32] and PoSv2[33] if you want to fully understand how Proof-of-Stake works. PoSv3 as described here also forms the basis of Anonymous-Proof-of-Stake that we will explore a bit further on.

[31] http://earlz.net/view/2017/07/27/1904/the-missing-explanation-of-proof-of-stake-version

[32] https://whitepaperdatabase.com/peercoin-ppc-whitepaper/

[33] https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf

11 The Anonymous-Proof-of-Stake (APoS) Protocol

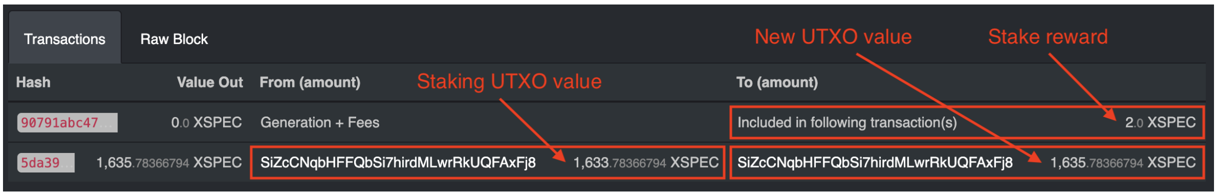
This section introduces a new staking algorithm that was introduced to the ALIAS (private) mainnet through the release of v3.0.9 and a hard-fork on 17th May 2019. This is the first known implementation of a private staking protocol employing ring-signatures in the staking transactions. This protocol was designed and coded by ALIAS (private) lead developer Philip Mueller (*@Tek*). The protocol is based on the PoSv3 protocol that we described in the previous section.

11.1 Introduction

We consider that a privacy focused pure Proof-of-Stake (*PoS*) network such as ALIAS (private) needs to be able to maintain consensus through a mechanism that maintains privacy, prevents easy blockchain analysis and is censorship resistant. Hence, such a network is not complete without a way to stake in private. There should be a way to maintain the privacy of all the network participants throughout the staking process. The participants should also be able to acquire their stake reward whilst maintaining their privacy.

11.2 The Problem

In a standard staking transaction (*PoSv3*) a value known as the '*kernel hash*' is calculated from several inputs including values taken from the last valid block, called a '*StakeModifier*' and the value of the user's UTXO. A valid '*kernel hash*' needs to be below a certain threshold that is determined by a separate calculation. The user (*the wallet*) who is able to generate a valid '*kernel hash*' will be granted the right to add the next block to the blockchain. The newly added block includes the generated stake reward (ALIAS (private)) and any transactions currently in the memory pool + any fees. The UTXO used to calculate the valid '*kernel hash*' will be spent and the generated stake reward + the value of the spent UTXO will be included in a newly generated UTXO associated with the same public address. In this way every stake that has been generated as a result of UTXOs associated with a certain public address will forever be linked to that address and it's plain for all to see. Below is an example of a standard staking transaction[34]:



Hash	Value Out	From (amount)		To (amount)	
90791abc47...	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	2.0 XSPEC
5da39...	1,635.78366794 XSPEC	SiZcCNqbHFFQbSi7hirdMLwrRkUQFAxFj8	1,633.78366794 XSPEC	SiZcCNqbHFFQbSi7hirdMLwrRkUQFAxFj8	1,635.78366794 XSPEC

Figure 7: Example of a staking transaction

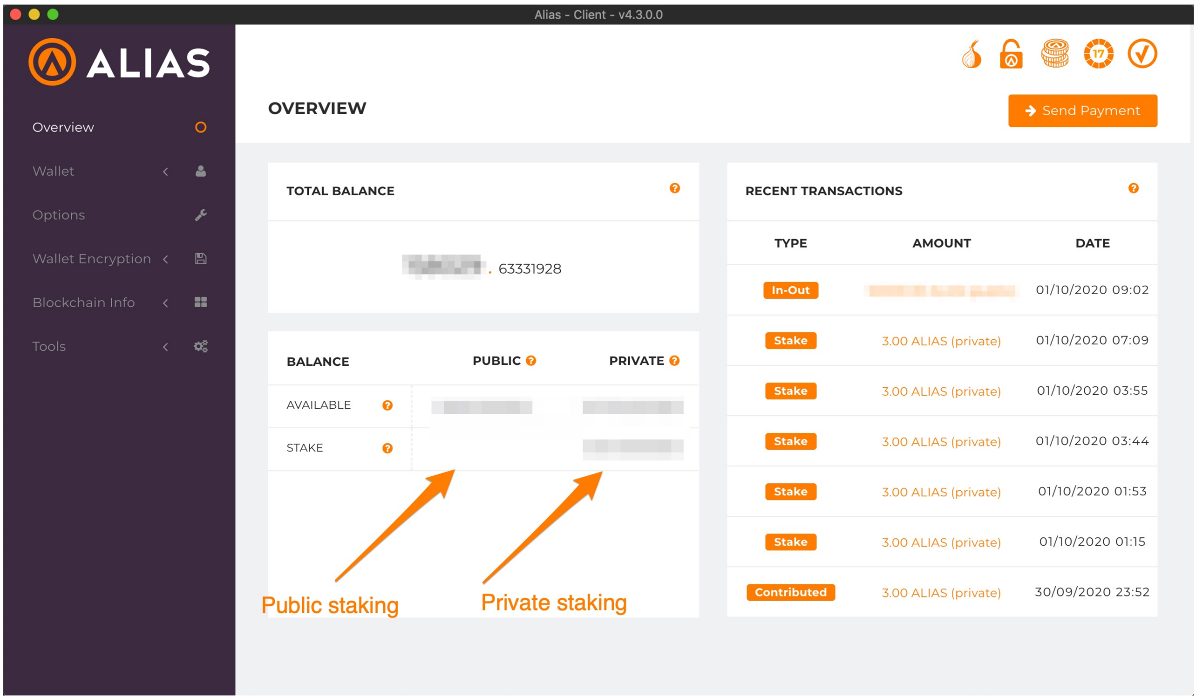
As with any typical PoS cryptocurrency the staking transactions suffer from all the privacy issues of a standard UTXO transaction, and these transactions are potentially traceable and linkable on the blockchain. It would therefore require some effort from the users to try to maintain anonymity in a standard PoS system and that will in turn weaken the overall resilience of the network against analysis. The network should not have to depend on the participants to maintain anonymity. All the staking transactions completed by the same user can potentially

[34] https://chainz.cryptoid.info/ALIAS (private)/block.dws?1190944.htm

be linked and users' balances can be estimated, hence the '*rich list*' feature of many block explorers. As explained previously, most blockchain forensic analysis focuses on address re-use and change addresses and this is exactly what you get with a standard PoS network.

11.3 The Solution

We have therefore developed what we call '**Anonymous-Proof-of-Stake**' (APoS) to solve this problem. This is a brand new and novel staking protocol utilising only ALIAS (private) and ring-signatures in the staking transactions and the rewards are also paid in ALIAS (private). This offers a through-and-through confidential way to maintain consensus and provide strong privacy for participants whilst securing the network. It is also appropriate to emphasise that this confidential and private consensus mechanism is totally decentralised and does not depend on any central servers or authority and is 100% peer-to-peer and there is no trusted setup. This provides very strong network resilience with no single point of failure.



TOTAL BALANCE

[redacted]. 63331928

BALANCE	PUBLIC	PRIVATE
AVAILABLE	[redacted]	[redacted]
STAKE		[redacted]

RECENT TRANSACTIONS

TYPE	AMOUNT	DATE
In-Out	[redacted]	01/10/2020 09:02
Stake	3.00 ALIAS (private)	01/10/2020 07:09
Stake	3.00 ALIAS (private)	01/10/2020 03:55
Stake	3.00 ALIAS (private)	01/10/2020 03:44
Stake	3.00 ALIAS (private)	01/10/2020 01:53
Stake	3.00 ALIAS (private)	01/10/2020 01:15
Contributed	3.00 ALIAS (private)	30/09/2020 23:52

Figure 8: New updated UI for the ALIAS (private) wallet in v4.x

11.4 Benefits of Anonymous Staking

The benefits are straight forward and easy to appreciate; if you transfer your holdings to *ALIAS (private)*, our anonymous coin, nobody will be able to know your balance, nobody will know how much you stake and if you keep transactions *ALIAS (private)* <> *ALIAS (private)* you preserve your privacy at all times. It is useful however to remember that once ALIAS (private) is converted to ALIAS (private) all your *ALIAS (private)* <> *ALIAS (private)* transactions are again potentially traceable. Note that the potential traceability only becomes an issue after the conversion from ALIAS (private) >> ALIAS (private) and does not affect the previous ALIAS (private) transaction.

With the addition of anonymous staking to ALIAS (private) and the updated UI it is now easy to distinguish between the different transactions and you will clearly see if you are staking ALIAS (private), ALIAS (private) or both. The previously described ‘*Development Contribution Blocks*’ (marked as ‘contributed’) will still occur every 1 in 6 blocks regardless of whether the block has been staked via the standard staking transaction or the anonymous ATXO staking transaction.

In addition to the pre-programmed 1/6 ‘*Development Contribution Blocks*’, you can still choose to donate extra via the ‘*Options*’ menu in the wallet.

11.5 APoS Implementation Detail

In ALIAS (private)’s APoS the basic structure of the staking transactions and the rules remain mainly intact. But, instead of using the UTXO transaction hash as an input in the '*kernel hash*' calculation and the difficulty calculations, the so called '*keyimage*' associated with an ATXO is used instead as an input to calculate the '*kernel hash*' and difficulty. An ATXO is an '*Anonymous Transaction Output*' and associated with a discreet ALIAS (private) value. ALIAS (private) takes on discreet values, like 100, 50, 40, 10, 1, 0.5 and so on. Each ATXO has a unique cryptographically determined '*keyimage*' associated with it.

This '*keyimage*' can be used to prevent double spend and once an ATXO is spent the '*keyimage*' is stored on the blockchain. The ATXO is totally dissociated from any user and as each and every ATXO is a unique '*data-entity*', ATXOs cannot be linked to each other. Only the user holds the '*secret*' key needed to prove ownership and no information is written to the blockchain that can in any way identify the owner of an ATXO.

In order to spend an ATXO or to use an ATXO in the staking transaction a user need to provide a valid ring-signature with a ring size of 10. The staking transaction therefore uses an ATXO as an input with 9 additional mixins for the ring-signature and new ATXOs are generated to the value of the stake reward in discreet units.

11.5.1 ATXO staking logic

Below is a summary of the logic for ATXO based staking w/ ring signatures.

An ATXO staking transaction is valid, if:

- The transaction is PoSv3 compliant
- VIN[0] has a valid ring-signature of MIN_RING_SIZE 10 (must has ring size 10)
- The ‘*keyImage*’ of the ATXO to be consumed is unspent (*not the mixins*)
- All ring-signature ATXOs meets minDepth maturity requirement
- The kernel hash calculated is below target

11.5.2 ATXO coinstake kernel protocol

The coinstake kernel (input 0) must meet hash target according to the formula:

$$hash(nStakeModifier + keyImage + nTime) < bnTarget * nWeight$$

This ensures that the chance of getting a coinstake is proportional to the amount of coins one owns.

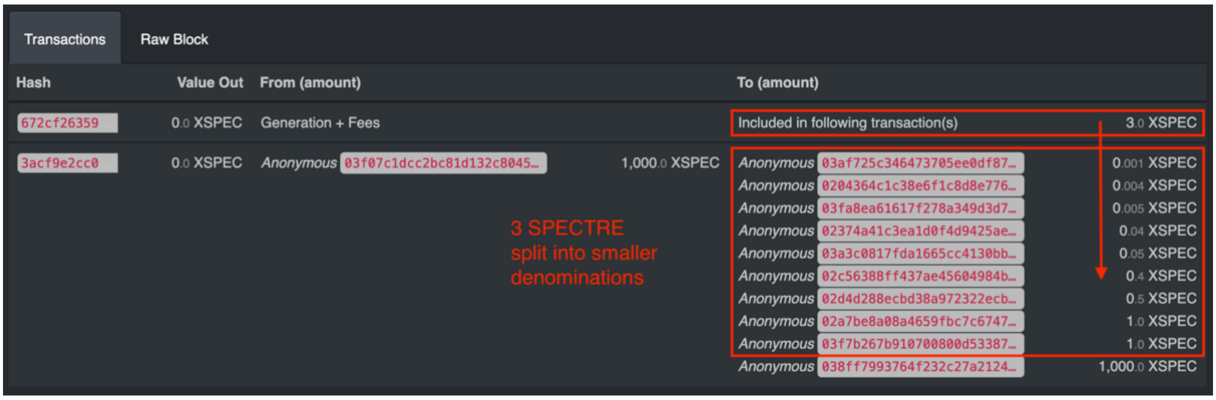
The reason this hash is chosen is the following:

- nStakeModifier: scrambles computation to make it very difficult to precompute future proof-of-stake.
- nStakeModifier is either the UTXO hash (PoSv3) or keyImage (APoS) used for the last staking transaction plus the previous block's stake modifier.
- keyImage: the keyImage of the ATXO used for staking is unique regardles the mixins, makes sure an ATXO can only be used once for generating a kernel hash.
- nTime: current timestamp.

11.6 ATXO Splitting

If an ATXO is staked, the algorithm searches for the denomination with the least amount of unspent in the range of $>= BASE\_FEE * 10$ and $<$ staked ATXO value. If a denomination in this range is found with less than 100 unspent ATXOs, the staked ATXO will be split to create one ATXO of the "*low running*" denomination.

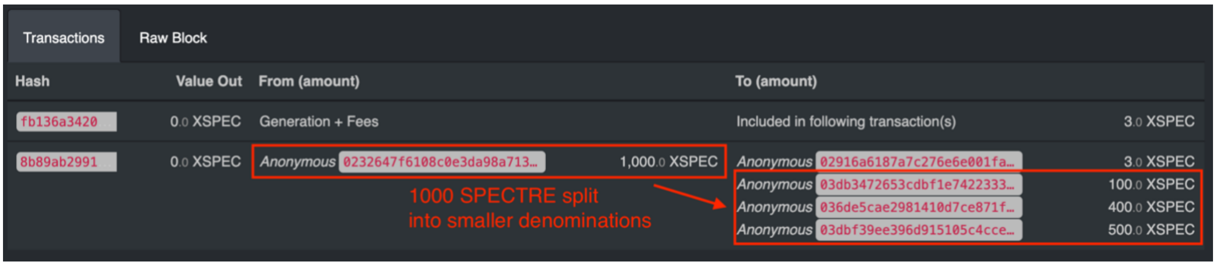
Please see the examples below taken from the ALIAS (private) Block explorer:



Hash	Value Out	From (amount)		To (amount)	
672cf26359	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	3.0 XSPEC
3acf9e2cc0	0.0 XSPEC	*Anonymous* 03f07c1dcc2bc81d132c8045…	1,000.0 XSPEC	*Anonymous* 03af725c346473705ee0df87…	0.001 XSPEC
				Anonymous 0204364c1c38e6f1c8d8e776…	0.004 XSPEC
				Anonymous 03fa8ea61617f278a349d3d7…	0.005 XSPEC
				Anonymous 02374a41c3ea1d0f4d9425ae…	0.04 XSPEC
				Anonymous 03a3c0817fda1665cc4130bb…	0.05 XSPEC
				Anonymous 02c56388ff437ae45604984b…	0.4 XSPEC
				Anonymous 02d4d288ecbd38a972322ecb…	0.5 XSPEC
				Anonymous 02a7be8a08a4659fbc7c6747…	1.0 XSPEC
				Anonymous 03f7b267b910700800d53387…	1.0 XSPEC
				Anonymous 038ff7993764f232c27a2124…	1,000.0 XSPEC

3 SPECTRE split into smaller denominations

Figure 9: 1000 ALIAS (private) is staked, 3 ALIAS (private) is staking reward for the block, 0.001 ALIAS (private) is the lowest running and hence 3 ALIAS (private) gets split into $0.001 + 0.004 + 0.005 + 0.04 + 0.05 + 0.4 + 0.5 + 1 + 1 = 3$ ALIAS (private)



Hash	Value Out	From (amount)		To (amount)	
fb136a3420	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	3.0 XSPEC
8b89ab2991	0.0 XSPEC	Anonymous 0232647f6108c0e3da98a713…	1,000.0 XSPEC	Anonymous 02916a6187a7c276e6e001fa…	3.0 XSPEC
				Anonymous 03db3472653cdbf1e7422333…	100.0 XSPEC
				Anonymous 036de5cae2981410d7ce871f…	400.0 XSPEC
				Anonymous 03dbf39ee396d915105c4cce…	500.0 XSPEC

1000 SPECTRE split
into smaller denominations

Figure 10: A further example of where 1000 ALIAS (private) is staked, 3 ALIAS (private) is the staking reward for the block, 100 ALIAS (private) is the lowest running and hence 1000 ALIAS (private) gets split into $100 + 400 + 500 = 1000$ ALIAS (private)

11.7 ATXO Consolidation

Through the '*Stealth Staking*' process and in any ALIAS (private) <> ALIAS (private) transactions there will be constant creation of small value ATXOs from staking rewards and transaction change. This will tend to result in larger transaction sizes in subsequent transactions as the number of smaller value ATXOs are combined for the desired output amount.

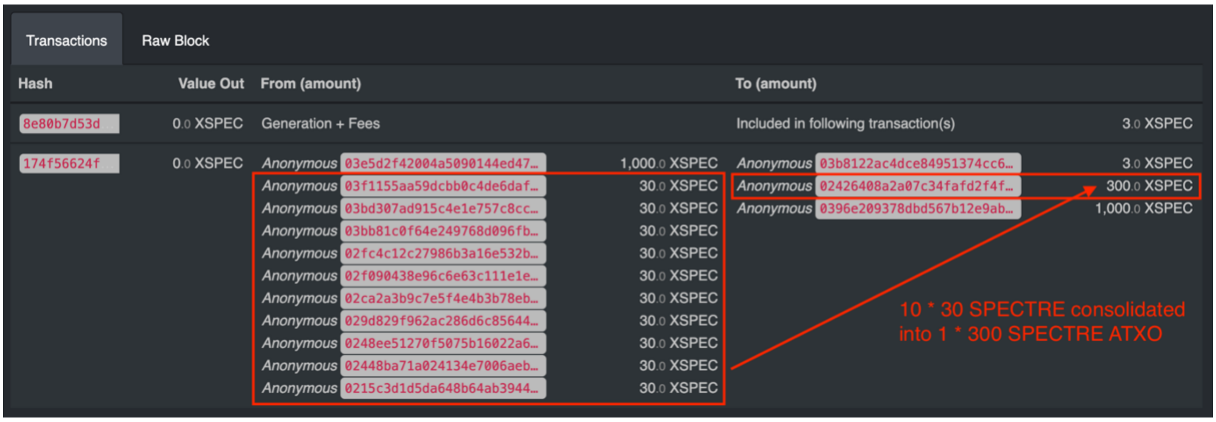
To solve this problem an extra ATXO '*consolidation algorithm*' has been implemented as part of the ATXO staking transaction where same value ATXOs are consolidated into larger value ATXOs. After a valid '*Kernel Hash*' is found, the ATXO used for staking is added as the first VIN to the transaction. The '*consolidation algorithm*' then tries to consolidate up to 50 ATXOs. The ATXOs are iterated from the lowest to the highest amount with a maximum ATXO value to be consolidated of 100 ALIAS (private) (MAX_STAKING_OUTPUT / 10). Every time 10 ATXOs are found with same value they are added as an additional input. A maximum of 50 Inputs are added. In the below example we have chosen some values as an illustration.

Multiple		**ATXO value**		**Consolidated ATXO value**
10	$*$	0.00000001	>>	0.0000001
10	$*$	3	>>	30
30	$*$	1	>>	30

Table 1: Consolidation examples

The new consolidation algorithm ensures that at least 200 unspent '*mixins*' of any denomination remain. This means that if less than 200 unspent ATXOs of a certain denomination exist, there is no consolidation of that particular denomination. This serves a dual purpose; it reduces the number of small value ATXOs and so reduces the subsequent transaction sizes as discussed above, but it also works to increase what we can call the '*transactional entropy*' of the network to reduce the possibility of blockchain analysis by linking ATXOs in transactions. It is apt at this stage to remember that each of the ATXOs whether consolidated or not can all act as '*mixins*' in all future transactions, and an observer can NEVER establish if the ATXO value has been spent. Consequently, what we call the '*Transactional Entropy*' increases by the block with minimal blockchain bloat.

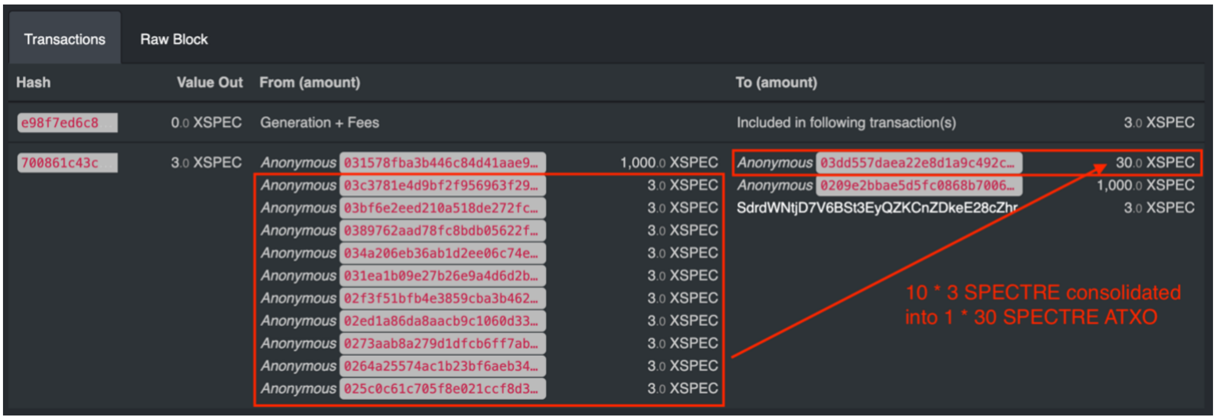
Please see the examples below taken from the ALIAS (private) Block explorer:



Hash	Value Out	From (amount)		To (amount)	
8e80b7d53d	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	3.0 XSPEC
174f56624f	0.0 XSPEC	Anonymous 03e5d2f42004a5090144ed47…	1,000.0 XSPEC	Anonymous 03b8122ac4dce84951374cc6…	3.0 XSPEC
		Anonymous 03f1155aa59dcbb0c4de6daf…	30.0 XSPEC	Anonymous 02426408a2a07c34fafd2f4f…	300.0 XSPEC
		Anonymous 03bd307ad915c4e1e757c8cc…	30.0 XSPEC	Anonymous 0396e209378dbd567b12e9ab…	1,000.0 XSPEC
		Anonymous 03bb81c0f64e249768d096fb…	30.0 XSPEC		
		Anonymous 02fc4c12c27986b3a16e532b…	30.0 XSPEC		
		Anonymous 02f090438e96c6e63c111e1e…	30.0 XSPEC		
		Anonymous 02ca2a3b9c7e5f4e4b3b78eb…	30.0 XSPEC		
		Anonymous 029d829f962ac286d6c85644…	30.0 XSPEC		
		Anonymous 0248ee51270f5075b16022a6…	30.0 XSPEC		
		Anonymous 02448ba71a024134e7006aeb…	30.0 XSPEC		
		Anonymous 0215c3d1d5da648b64ab3944…	30.0 XSPEC		

10 * 30 SPECTRE consolidated
into 1 * 300 SPECTRE ATXO

Figure 11: In this example you can see $10 * 30$ ALIAS (private) being consolidated into $1 * 300$ ALIAS (private)



Hash	Value Out	From (amount)		To (amount)	
e98f7ed6c8	0.0 XSPEC	Generation + Fees		Included in following transaction(s)	3.0 XSPEC
700861c43c	3.0 XSPEC	Anonymous 031578fba3b446c84d41aae9…	1,000.0 XSPEC	Anonymous 03dd557daea22e8d1a9c492c…	30.0 XSPEC
		Anonymous 03c3781e4d9bf2f956963f29…	3.0 XSPEC	Anonymous 0209e2bbae5d5fc0868b7006…	1,000.0 XSPEC
		Anonymous 03bf6e2eed210a518de272fc…	3.0 XSPEC	SdrdWNtjD7V6BSt3EyQZKCnZDkeE28cZhr	3.0 XSPEC
		Anonymous 0389762aad78fc8bdb05622f…	3.0 XSPEC		
		Anonymous 034a206eb36ab1d2ee06c74e…	3.0 XSPEC		
		Anonymous 031ea1b09e27b26e9a4d6d2b…	3.0 XSPEC		
		Anonymous 02f3f51bfb4e3859cba3b462…	3.0 XSPEC		
		Anonymous 02ed1a86da8aacb9c1060d33…	3.0 XSPEC		
		Anonymous 0273aab8a279d1dfcb6ff7ab…	3.0 XSPEC		
		Anonymous 0264a25574ac1b23bf6aeb34…	3.0 XSPEC		
		Anonymous 025c0c61c705f8e021ccf8d3…	3.0 XSPEC		

10 * 3 SPECTRE consolidated
into 1 * 30 SPECTRE ATXO

Figure 12: In this example you can see $10 * 3$ ALIAS (private) being consolidated into $1 * 30$ ALIAS (private)

12 Tor (The Onion Router)

Tor aka. '*The Onion Router*' is described on the project website[35] like this; "*Tor is free software and an open network that helps you defend against traffic analysis, a form of network surveillance that threatens personal freedom and privacy, confidential business activities and relationships, and state security.*"

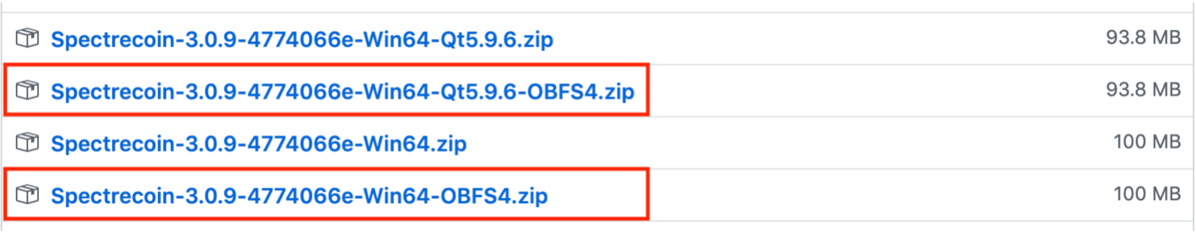
Tor, in essence, hides your real IP address and instead provides you with a so called *.onion* address that should be extremely difficult to de-anonymise for any attacker. A normal IP might look like this: 123.45.67.8 and an *.onion* address might look like this: *fpzcf23ifxpiucjm.onion*. When you broadcast your normal IP address it will be possible for an adversary to identify you through your internet service provider that will hold all the data on all connections. If you broadcast a .onion address only this is not possible and any traffic can only be traced back to another hidden node on the Tor network. Alias has Tor '*built-in*' and running as a process and when you start the software you will connect to the Tor network with an *.onion* address and you will not broadcast your real IP address through the Alias software for as long as you use the software. The Alias network will reject all non-Tor connections.

We are aware of ongoing discussions about Tor vs. I2P[36] vs. Monero's Kovri project that is currently under development. You can read up on this, but it is clear that both may have both advantages and disadvantages. Tor is in wider use and will have more development effort behind. We are also aware of technology such as Dandelion and we will keep an eye on developments in this area of network security and we are considering this.

If you want yet another layer of network security, you can also use a VPN service with Tor so explore this if you feel that you need this level of security.

12.1 OBFS4

OBFS4 is what is known as a Tor '*pluggable Transport*' and is a means to hide the fact that you are connecting using Tor. A range of countries including China and Iran for example are using technology to block Tor traffic and we have implemented this to try and circumvent any such censorship. It's also useful to bear in mind that some countries might view a Tor user with suspicion although it's not blocked. When you go to the download page for Alias look for the OBFS4 versions of the software.



Spectrecoin-3.0.9-4774066e-Win64-Qt5.9.6.zip	93.8 MB
Spectrecoin-3.0.9-4774066e-Win64-Qt5.9.6-OBFS4.zip	93.8 MB
Spectrecoin-3.0.9-4774066e-Win64.zip	100 MB
Spectrecoin-3.0.9-4774066e-Win64-OBFS4.zip	100 MB

Figure 13: Alias wallet version with includet OBFS4

[35] https://www.torproject.org/

[36] https://geti2p.net/en/

It's difficult to say with 100% certainty which countries block Tor but you can find more information on this[37] topic on the Tor project website, that will show you the top 10 countries where Tor bridges are used. This[38] website may also be of interest and for further information on Tor bridges and transports you can find a lot of information here[39] on the Tor website and elsewhere. We will be considering various options in this area in future development.

[37] https://metrics.torproject.org/userstats-bridge-table.html

[38] https://grobox.de/tor/

[39] https://2019.www.torproject.org/docs/bridges