
VERGE
The most privacy focused cryptocurrency
BLACKPAPER

1. Introduction

Bitcoin was developed and released in 2009 in response to an inherent flaw in the way transactions were processed on the Internet. In his whitepaper, Nakamoto explains that "Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model" [1]. Since its original inception in 2009, Bitcoin has been rapidly adopted into today's modern marketplaces. A primary issue with Bitcoin's rapid adoption is the increase of demand on the original blockchain to handle varying degrees of large transactions. With increased demand comes increased transactional waiting periods, and this has resulted in higher transactional fees in attempts to try and speed-up transaction confirmation times.

The core innovation behind Bitcoin is its decentralized structure. Unlike traditional fiat currencies, Bitcoin has no central control, no central repository of information, no central management, and no central point of failure. However, one of the challenges facing Bitcoin is that most of the actual e-services and e-businesses built around the Bitcoin ecosystem are centralized. Due to the centralized nature of the current system, e-commerce is ran by individuals in specific locations that utilize vulnerable computer systems, that are susceptible to legal entanglements. Verge is one of the truly decentralized currencies available today due to its standing commitment to building off of the core fundamentals of Bitcoin, while bringing an entirely new layer of anonymity to realization.

2. Tor Integration

Tor, derived from an acronym for the original software project name "The Onion Router" is an IP obfuscation service which enables anonymous communication across a layered circuit based network. Tor directs internet traffic through a free worldwide volunteer overlay network consisting of more than seven thousand relays to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis. The layers of encrypted address information used to anonymize data packets sent through Tor are reminiscent of an onion, hence the name. That way, a data packet's path through the Tor network cannot be fully traced. Tor's use is intended to protect the personal privacy of users, as well as their freedom and ability to conduct confidential communication by keeping their Internet activities from being monitored.

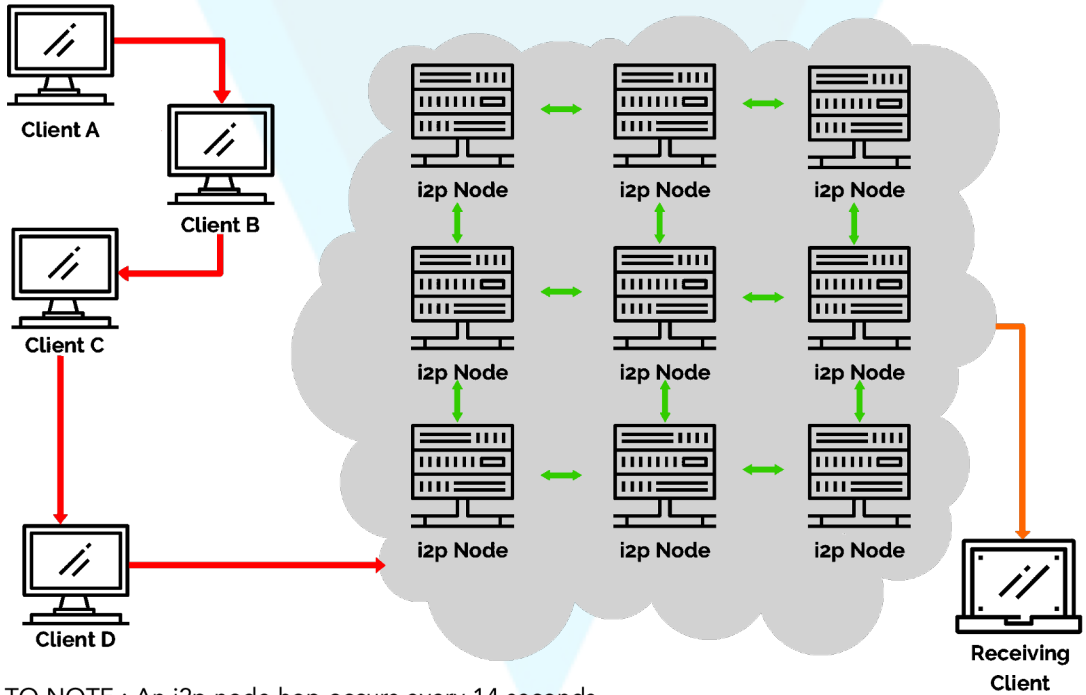
Onion routing is implemented by encryption in the application layer of a communication protocol stack, nested like the layers of an onion. Tor encrypts the data, including the next node destination IP, multiple times and sends it through a virtual circuit comprising successive, randomly selected Tor relays. Each relay decrypts only enough of the data packet wrapper to know which relay the data came from, and which relay to send it to next. The relay then rewraps the package in a new wrapper and sends it on. The Final relay decrypts the innermost layer of encryption and sends the original data to its destination without revealing, or even knowing, the source IP address.

Because the routing of communication is partly concealed at every hop in the Tor circuit, this method eliminates any single point at which the communicating peers can be determined through network surveillance that relies upon knowing its source and destination.

3. I2P Integration

I2p was originally built to provide hidden services which allow people to host servers at unknown locations. I2p provides many of the same benefits that Tor does. Both allow anonymous access to online content, make use of a P2P-style routing structure, and both operate using layered encryption. However, I2p was designed to be a “network within the internet,”(see figure 2.1) with traffic staying contained in its borders. I2P performs packet based routing as opposed to Tor’s circuit based routing. This provides the benefit of permitting I2p to dynamically route around congestion and service interruptions in a manner similar to the internet’s IP routing. This provides a higher level of reliability and redundancy to the network itself.

Figure 2.1
How an i2p Transaction Occurs



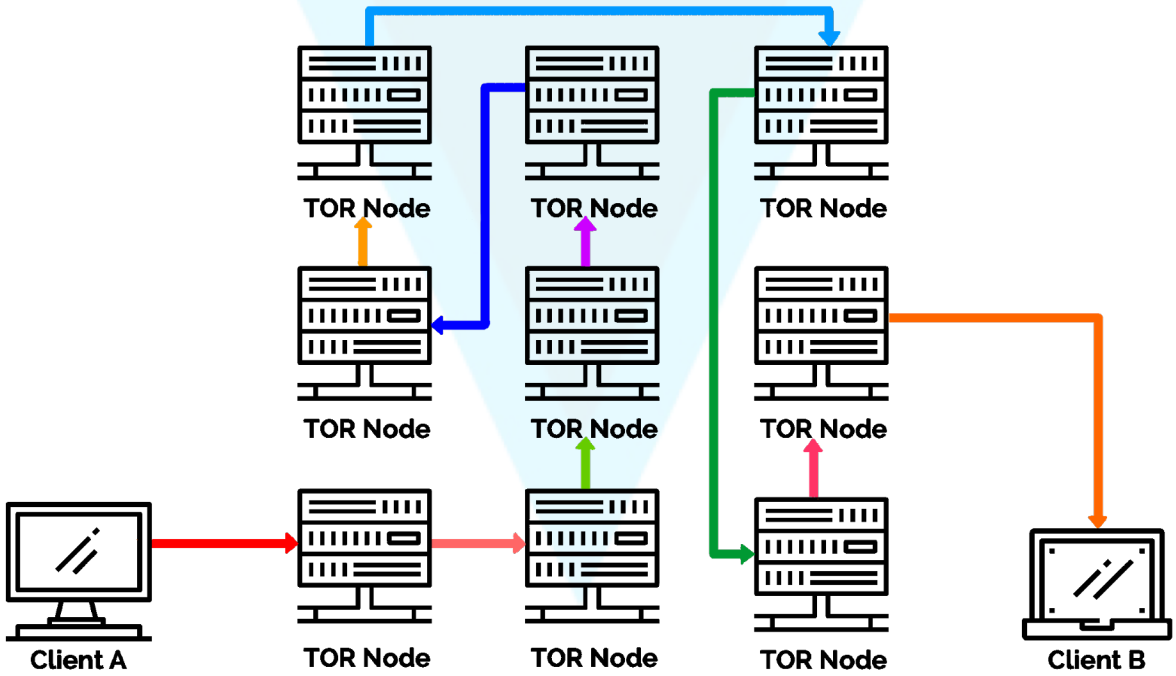
TO NOTE : An i2p node hop occurs every 14 seconds.

The first time a client wants to contact another client, they make a query against the fully distributed "network database" - a custom structured distributed hash table (DHT) based off the Kademlia algorithm [2]. This is done to find the other client's inbound tunnels efficiently, but subsequent data between them usually includes that information so no further network database lookups are required.

I2p is a highly obfuscated tunneling service using ipv6 that anonymizes all Verge data being sent over the network. Each client application has their i2P "router" build several inbound and outbound "tunnels" - a sequence of peers that pass data in one direction (to and from the client, respectively) [2]. In turn, when a client wants to send Verge data to another client, the application passes the message through one of their outbound tunnels targeting one of the other client's inbound tunnels, eventually reaching the destination.

Rather than relying on a centralized set of directory servers, like Tor, I2p uses two distributed hash tables to coordinate the state of the network. Distributed hash tables or DHTs are a distributed and often decentralized mechanism for associating hash values with content. The primary advantage to DHT's are their scalability. A successful decentralized P2P network requires good scalability of its services to ensure the size of content or transaction sharing can continue to grow as required. Additionally I2P does not rely on a trusted directory service to get route information. Instead, network routes are formed and constantly updated dynamically, with each router constantly evaluating other routers. Lastly, I2p establishes two independent simplex tunnels for traffic to traverse the network to and from each host as opposed to Tor's formation of a single duplex circuit (see figure 1.1).

Figure 1.1
How a TOR Transaction Occurs



TO NOTE: A TOR node hop occurs every 10 minutes.

4. Electrum

Electrum's strength is speed and simplicity, with low resource usage. It uses secure remote servers that handle the most complicated parts of the Verge network and also allows users to recover their wallets with a secret seed phrase. Additionally, Electrum offers a simple and easy to use cold storage solution. This allows users to store all or part of their coins in an offline manner. Moreover, Electrum is one of the only wallets to provide native Tor and i2P support. By integrating Electrum with Tor and i2P, one can achieve anonymity while using the desktop/mobile wallet. Both IP address and transaction information is secured and does not leak to the connecting servers; increasing user privacy.

Electrum enables multi-signature support, which requires more than one key to authorize a Electrum transaction. Standard transactions on the Verge network could be called “Single-signature transactions” [4], because transfers require only one signature - from the owner of the private key associated with the Verge address. An Electrum transaction, with multi-signature support, requires the signatures of multiple people before the coins can be transferred. Verge then requires multiple different party addresses to be provided in order to do anything with them.

Here is an example:

> “One Electrum wallet is on your primary computer, the other on your smart phone - the coins cannot be spent without a signature from both devices. Thus, an attacker must gain access to both devices in order to steal your coins”

Key Features of an Electrum Wallet

Deterministic Key Generation
If you lose your wallet, you can recover it from its seed. You are protected from your own mistakes.

Instant On
The client does not download the blockchain, it requests blockchain information from a server. No delays, always up-to-date.

Locally signed Transactions
Your private keys are not shared with the server. You do not have to trust the server with your coins.

Freedom and Privacy
The Electrum server does not store user accounts. You can also export your private keys, meaning YOU own your address.

5. Multi-Algorithm Support

Verge is a multi-algorithm cryptocurrency that is designed to enable people with different types of mining devices to have equal access to earning coins. It is one of the only cryptocurrencies to support 5 hash functions combined on one blockchain. This results in increased security and a wider range of people and devices that can mine Verge hence equal distribution of Verge is ensured for everyone.

The total supply of Verge is 16.5 Billion coins. What makes Verge stand out from other cryptocurrencies are the 5 Proof-of-Work algorithms that run on its blockchain, namely Scrypt, X17, Lyra2rev2, myr-groestl and blake2s. All 5 algorithms have a 30-second block target block time. The difficulty is influenced only by the algorithm's hash rate. This allows improved security and protection against 51% attacks.

6. Android Tor + I2P

Verge sits at the forefront of innovation in the mobile cryptocurrency space. We have pioneered and developed two very unique and first of their kind android wallets. One of which operates exclusively on The Onion Router Network (Tor) and the other operating exclusively on The Invisible Internet Project (i2P). The Verge Tor and I2p wallets are built around the premise of anonymity. The wallets have no built-in ability to connect to or broadcast user information over Clearnet. Transactions are completed via Simple Payment Verification (SPV), a technique described in Satoshi Nakamoto's paper that allows for the wallet to verify transactions through proof of inclusion; a method for verifying if a particular transaction is included in a block without downloading the entire block (similar to how an Electrum wallet functions).

SPV allows for nearly instant payment confirmations because it acts as a thin client that only needs to download the block headers, which are drastically smaller than full blocks. The Verge Tor and i2P wallets also have built in security features such as a 4 digit pin code and biometric locking options for an added layer of physical security.

Additionally, the Verge Tor and i2P wallets are able to handle P2P QR code scan transactions with instant verification. Clients are able to also import QR codes from paper wallets to pull balances from cold storage if required.

7. P2P Platform-Integrated Portals

Peer-to-Peer (P2P) transaction support for Telegram, Discord and Twitter is supported by Verge, Slack and Steam integrations are currently in development. Telegram is a free cloud-based instant messaging service that supports Android, iOS, Windows Phone, Windows NT, macOS and Linux. Telegram uses a symmetric encryption scheme called MTProto. The protocol was developed by Nikolai Durov and other developers at Telegram and is based on 256-bit symmetric AES encryption, RSA 2048 encryption and Diffie–Hellman key exchange. Discord is a proprietary freeware VoIP application that has widespread adoption in the crypto community. Like Telegram, Discord has support on Windows, macOS, Android, iOS and has a browser accessible web client. Implementing Verge P2P capabilities on these platforms allows users to send and receive funds on the fly, no matter where they are (regardless if they have an actual wallet installed or not).

P2P is an online technology that allows users to transfer coins via the internet or mobile device. To do this, consumers use an online application, or in this case a bot – to designate the amount of coins to be transferred. The recipient is designated by just their username and once the transfer has been initiated by the sender, the recipient then receives a notification to use the online bot. that he has received a payment at a newly established deposit address. The user is then allowed to tweet or message the bot with a simple command such as “!withdraw” and is then prompted with a set of instructions on how to receive their newly acquired Verge. This service does not require any additional information past the amount you want to send and who to send to. No privacy information such as IP addressing, location, name is retained during this process. Your personal identity outside of initiating the transaction remains completely anonymous.

Verge is one of the only cryptocurrencies to already offer P2P solutions for Telegram, Discord, Twitter and Internet Relay Chat (IRC) with Reddit, Slack and Steam support coming at a future date. These P2P offerings allow users to transfer Verge to anyone on the same social platform as them.

8. Wraith Protocol

What is Wraith Protocol?

Wraith Protocol makes it possible to choose between a public or private ledger. Through this new system, users who value transparency and accountability, e.g. merchants, have the option to have transactions viewable on the blockchain. On the other hand, it also provides an option to those who prefer transactions to vanish entirely. Wraith Protocol allows for complete anonymity to be maintained while providing a safe and secure method of sending and receiving Verge coins without transactions being traceable on a publicly accessible ledger. The update includes stealth Addressing and the latest Tor+SSL integration that will take our core QT users off of clearnet, and migrate them to exclusively operate on the latest Tor network.

Also included are the capabilities to designate which ledger a user wishes to transact across, public or private. With elegant simplicity, the Wraith Protocol update will enable users to toggle a switch within the Core QT wallet that allows them to transact via stealth addressing with an additional layer of IP obfuscation through the Tor Network.

Deep Dive

Let's start by taking a look at some of the basics and key concepts associated with the Key agreements, The Diffie-Hellman algorithm and Elliptic-curve cryptography.

What is a Key Agreement?

A Key agreement scheme is a procedure by which two or more parties agree upon a value from which they can subsequently derive one or more keys for use in symmetric encryption. Neither party completely determines the key value on their own. Instead, they both contribute to the final key value and most important, anyone who observes the exchanges between the two parties cannot tell what the final result will be. It is important to note that in their basic form, key-agreement schemes are anonymous, they do not tell either party the identity of the other party.

What is the Diffie-Hellman algorithm?

The original Diffie-Hellman key agreement scheme is based on multiplication of integers modulo a large prime number, specifically numbers greater than one and less than p, where p is a large prime. ECDH is an analogous scheme based on addition of points on an elliptic curve.
In both schemes, the basic operations are combined to create a primitive function known as a keyed one–way function. A Keyed one-way function is a function that takes two inputs, one of which is private (e.g., the key), and produces one output. Given the two inputs, it must be straightforward to calculate the output but, it must be computationally infeasible to calculate the key, using only the other input and the output. In this way each party can use their private key without revealing it to anyone else, either the other party or an eavesdropper (Man-in-the-middle).

What is an Elliptic-curve Diffie-Hellman (ECDH)?

ECDH is a variant of the Diffie-Hellman algorithm for elliptic curves. It is a Key-agreement protocol which means that ECDH defines how keys should be generated and exchanged between parties. How to actually encrypt data using these keys is up to us. EDCH is implemented to solve the following problem:

Two parties (Anthony and Billy) want to exchange information securely such that a third party (Man-in-the-middle) may intercept them, but may not decode them.

Here's how it works:

1. First, Anthony and Billy generate their own private and public keys. We have the private key ***dA*** and the public key ***HA=dAG*** for Anthony, and the keys ***dB*** and ***HB=dBG*** for Billy. Note that Anthony and Billy are using the same base point of ***G*** on the same elliptic curve on the same finite field.

2. Anthony and Billy exchange their public keys ***HA*** and ***HB*** over an insecure channel. The man in the middle would intercept ***HA*** and ***HB***, but won't be able to find out neither ***dA*** nor ***dB*** without solving the discrete logarithm problem.

3. Anthony calculates ***S=dAHB*** (using his own private key and Billy's public key), and Billy calculates ***S=dBHA*** (using his own private key and Anthony's public key). Note that ***S*** is the same for both Anthony and Billy, in fact:

$$S=dAHB=dA(dBG)=dB(dAG)=dBHA$$

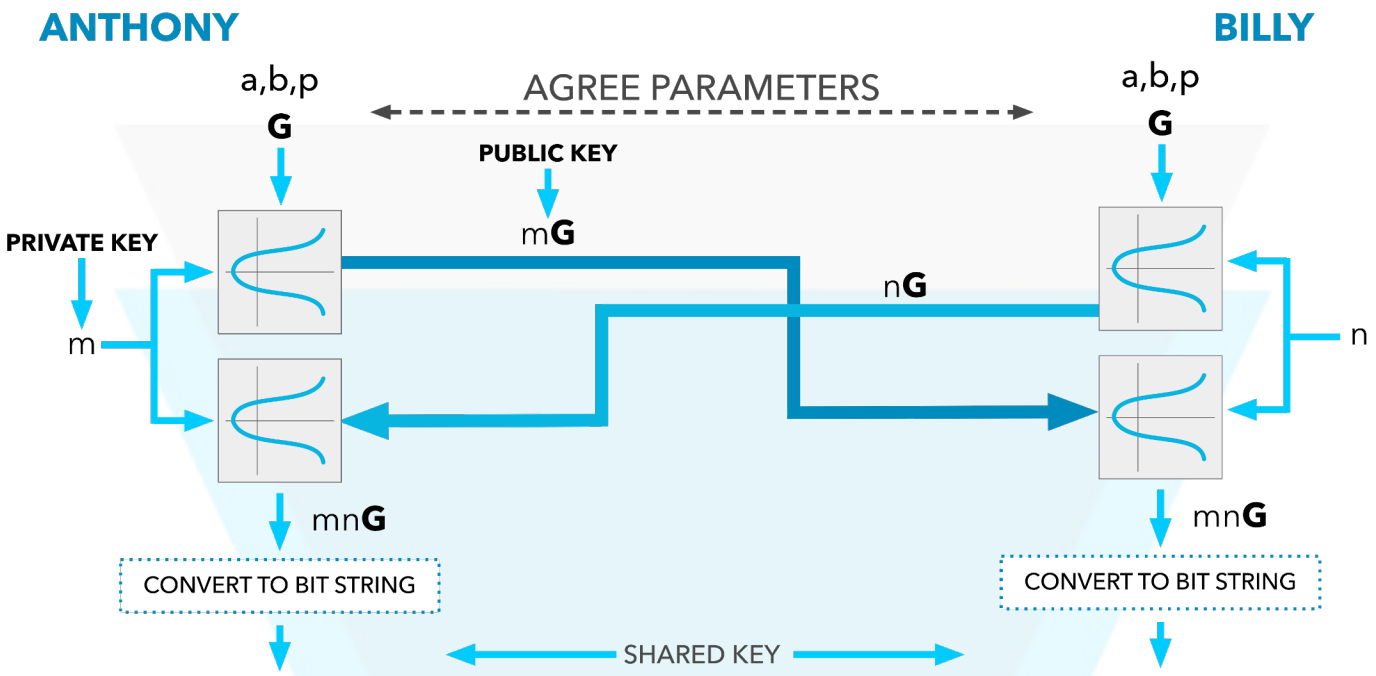
The Man in the Middle, however, only knows ***HA*** and ***HB*** (together with the other domain parameters) and would not be able to find out the shared secret ***S***.

In our particular use case before the protocol or in this case transaction can begin, both Anthony and Billy must agree to transactional parameters, a, b, p and G. (see figure 3). Each party then generates a random integer to use as its private key. For Anthony this is m, and for Billy this is n. Each then multiplies the base-point, G, by their private key to form a new point that represents their public key. In terms of an elliptic curve remember that each point comprises an x coordinate and a y coordinate.

Anthony and Billy then exchange their public keys and multiply the other's public key by their own private key. This produces a new point which is the same for each party. It remains only to convert this point to a bit string suitable for use as a key.

Note: An eavesdropper may be able to observe the agreed parameters and may see the exchange of public keys, but, he will not be able to determine what either private key is, nor the key that the two parties have agreed upon.

Figure 3: ECDH



What is Stealth Addressing?

Stealth Addressing allows senders to create an unlimited number of one-time destination addresses on behalf of the recipient without any interaction between the parties. These addresses can only be recovered and spent by the recipient and cannot be publicly linked to either the sender or receiver addresses from which they were derived. This is achieved through a system of cryptography known as Elliptic Curve, or more specifically in this case – Elliptic Curve Diffie-Hellman (ECDH for short). ECDH works by allowing any two individuals who know each others' public keys to be able to calculate a shared secret that nobody else can either duplicate or link to either party's public keys. Due to the unique cryptographic properties of the ECDH algorithm, the shared key cannot be reverse engineered to arrive at either the sender or receiver addresses.

A Verge Currency Stealth Address is a 95 character string that consists of a public view key and a public send key. When Anthony sent Verge to Billy, Anthony will use Billy's public view key and public spend key as well as some random data to generate a unique one-time public key (Stealth Address) for Billy's new output. Everyone can see the one-time public key on the blockchain but only Anthony and Billy know that Anthony sent Verge to Billy. The output is created in such a way that Billy is able to locate the output destined for him by scanning the blockchain with his wallet's private view key. Once detected and retrieved by Billy's wallet he would be able to calculate a one-time private key that corresponds with the one-time public key and spend the relevant output with his wallets private spend key. This whole process occurs without ever having Billy's wallet address publicly linked to any transaction.

Key Take-aways

1. it is publicly unlinkable to the original public address;
2. it is publicly unlinkable to any other one-time address;
3. only the recipient can link all their payments together
4. only the recipient can derive the secret key associated with the one-time address

Stealth addresses enhance user privacy in every transaction by allowing the user to generate a one-time public key which automatically generates and records who can spend an output in a later transaction. Stealth addresses prevent outputs from being associated with wallet addresses by effectively allowing users to transact outside of the publicly viewable blockchain. An outside observer has no way to tell if funds have been moved from one user to another nor do they have the ability to link wallet addresses together simply by looking up a transaction on the blockchain. When Anthony sends Verge to Billy, the output Billy receives will not be associated with his public wallet address. Stealth addressing has built in methods of ensuring funds have been sent by allowing the sender, in this case, Anthony, to be able to verify that payment was sent by checking the transactional confirmation within his wallet. Billy can rest assured that no one else can see when or if any Verge was sent to him.

Tor + SSL Integration

Previously, our CoreQT wallet had our users transacting across clearnet. With Wraith Protocol we are migrating all of our QT users away from clearnet and onto Tor. That being said our QT wallet will no longer have the ability to establish a connection with any networks outside of the Tor network which will help to ensure that our users remain anonymous. Tor is a decentralized system that allows users to connect through a network of relays which serve to obfuscate user IP addressing information by bouncing your connection from node to node at random, effectively eliminating any information trails. Our Tor integration also includes SSL encryption which establishes a secure and encrypted link between wallets to ensure that all data passed between the wallets remains private and integral. SSL encryption also ensures that data will make it from wallet to wallet without being intercepted or altered. Additional information on how Tor works can be found in sections 2.0.

9. Wraith Protocol Use Cases

Meet Jessica. As a nursing student who is finishing her degree, money is often tight and access to liquidity is of paramount importance. Recently, she made a purchase online using her credit card. Unfortunately, due to no fault of her own, her credit card number was skimmed and used to buy a luxury handbag in Perth. While her card company agreed to reimburse her, it would be several days before her new card would arrive. After this experience, she knows that financial security has to be her own responsibility. She knows that she can use Verge and the Wraith Protocol to make payments to her favorite e-commerce stores through Coinpayments.net and be guaranteed that her payment will not be interdicted or tampered with in any way. She can transact her business without any fear of theft and in the knowledge that she is in control of her financial destiny.

Now meet Randal. As an entrepreneur, he is very aware of the importance of protecting the identities and finances of his clients safe. This is especially true as he provides anonymous genetic screening for diseases such as Parkinson's Disease and Dementia. A breach of client data could ruin the lives of his clients, not only his business. After realizing that typical financial solutions provided no actual guarantee that leaks and breaches would not affect his business or his client, he began to use Verge to transact business. Thanks to the Stealth Addressing available through the Verge QT wallet, he is able to accept payment and provide truly anonymous testing services and give people information they may need to save their lives without risking identifiable data breaches.

10. Atomic Swaps

Atomic swaps, aka atomic cross-chain trading, allows for interoperability between Verge and all other cryptocurrencies in circulation with Atomic swap capabilities enabled. An Atomic swap works in the same way users would send funds to one another by allowing users to cross-trade different cryptocurrencies without relying on centralized parties. Verge will be implementing BIP65 Check Lock Time Verify (CLTV) otherwise known as Hash Time-Locked Contract. (HTLC). HTLC is a class of payments that use hash-locks and time-locks that require the receiver of a payment to acknowledge receiving the payment prior to a deadline by generating cryptographic proof of payment or forfeit the ability to claim the payment, returning it to the payer. For example, both parties submit their individual transactions to the appropriate blockchain. User A sends Verge on the Verge blockchain, and user B sends ETH on the Ethereum blockchain. The recipient can only claim this transaction by revealing a secret hash (proof of payment). This results in both transactions being linked to one another, despite them taking place across two different blockchains. If the recipient does not reveal their secret hash - the payment is then forfeit and returned to the payer.

Our users will be able to leverage Atomic Swaps while transacting across the Tor network via Wraith Protocol, thereby maintaining IP obfuscation and personal identity integrity while sending and receiving Verge through cross-chain transactions. Furthermore, this implementation not only allows for cross-chain transactions but it also paves the way for future implementations such as the Lightning Network, which will allow for automatic execution of cross-chain transactions and trading.

11. Encrypted Chat: Visp

Visp is a P2P (peer-to-Peer) Instant messaging system utilizing state-of-the-art encryption technology to keep your communications private. All messages are encrypted by the proven AES-256-CBC algorithm, and distributed between nodes in such a way as to prevent the recipients of messages from being inferred by assailants utilizing sophisticated traffic analysis. Whisper utilizes The Elliptic Curve Digital Signature Algorithm, which is a variant of the digital signature algorithm used in elliptic curve cryptography. ECDSA is used to give you the confidence of knowing messages you receive come from the original recipient and remain untouched in propagation. Messages are distributed via the preexisting Verge P2P network, and a copy of each encrypted message is stored on each node for a period of 48 hours.

As with stealth address transactions, the Elliptic Curve Diffie-Hellman key exchange method allows a secret key for encryption to be shared between the sender and the recipient using the data embedded in the message along with the private keys of Verge Stealth addresses held by the sender and recipient, thus allowing for the distribution of messages of whom nobody knows the recipient of. In order to send an encrypted message, much like sending Verge, you must possess the public key of the intended recipient. The public keys embedded in the Verge transaction blockchain when any amount is spent. If you are sending to an address that has not spent a transaction in the blockchain, the public key to that address must be provided manually.

Verge uses curve secp256k1 for all elliptic curve functions. This is the same curve used by Bitcoin along with the vast majority of altcoins. With such widespread use, underpinning systems of immense value it is extremely unlikely that curve secp256k1 is not secure. Messages are signed by the keys they were sent with. This allows you to be confident of the origin of the messages you receive and also allows the public key of the sender to be extracted from the message, providing you all the information needed to send a reply.

12. Bloom Filters: BIP37

BIP37 is a filter used primarily by SPV clients to request only matching transactions and merkle blocks from full nodes, which in turns speeds up transaction times. This BIP adds new support to the peer-to-peer protocol that allows peers to reduce the amount of transaction data they are sent. Peers have the option of setting filters on each connection they make after the version handshake has completed. A filter is defined as a Bloom filter on data derived from transactions. A Bloom filter is a probabilistic data structure which allows for testing set membership - they can have false positives but not false negatives.

13. Future Development: RSK Smart Contracts

Rootstock, or commonly referred to as RSK, is a two-way pegged sidechain that grafts smart contract functionality onto the Verge network. It also introduces an off-chain protocol for near-instant payments. RSK is an independent blockchain that does not have its own token, it instead relies on existing tokens (such as Verge). RSK is able to do this by pegging (or matching) its smart token to Verge, so that the value of an RSK token is exactly that of a Verge token. Users have the capabilities to freely move their tokens back and forth between the two chains.

A smart contract works by placing a user's Verge into a type of reserve where it is locked up and then used to back the RSK token, known as smartVerge. Think of it as putting your Verge into a checking account and then using the RSK network to spend that money. It is important to note that simple contracts have been in place for Bitcoin which allow users to create contracts, like mutlisig, that requires two or more users to sign off on a payment before it can be released. With the implementation of RSK on Verge, simple smart contracts are taken to a whole new level, with turing-complete smart contract capabilities that will go head-to-head with Ethereum's current offerings.

Another added benefit of RSK is its ability to scale. RSK currently achieves 400 payment transactions per second, which is a huge progressive leap when compared to our current standing transaction rate; around 100 per second. The RSK development team has stated that the eventual goal is to push the bar even higher with future goals to support 2,000 transactions per second using a second layer technology called Lumino. As stated in the LCTP whitepaper, the Lumino Network is an off-chain payment system that relies on a protocol known as the Lumino Transaction Compression Protocol. The LTCP can be compared to the Lightning Network, a scaling solution originally designed for bitcoin that Is currently being tested on Litecoin.

14. References

[1] Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

[2] I2P: A scalable framework for anonymous communication - I2P. (n.d.). Retrieved from https://geti2p.net/en/docs/how/tech-intro

[3] Multisignature - Bitcoin Wiki. (n.d.). Retrieved August 8, 2017, from https://en.bitcoin.it/wiki/Multisignature

[4] Voegtlin, T. (n.d.). Welcome to the Electrum Documentation! — Electrum 2.5 documentation. Retrieved August 8, 2017, from http://docs.electrum.org/en/latest/

Additional References:

Gribble, S., Brewer, E., Hellerstein, J., & Culler, D. (2000, October 23). Scalable, Distributed Data Structures for Internet Service Construction. Retrieved from https://www.usenix.org/legacy/events/osdi2000/full_papers/gribble/gribble_html/index.html

Anonymity and the Block Chain • IHB News™. (2014, November 18). Retrieved August 08, 2017, from https://ihb.io/2014-11-17/news/anonymity-block-chain-13570

Holden, E. (2017, July 18). An Introduction to Tor vs I2P. Retrieved August 08, 2017, from https://www.ivpn.net/privacy-guides/an-introduction-to-tor-vs-i2p

Distributed hash table. (n.d.). Retrieved August 08, 2017, from http://infoanarchy.org/Distributed_hash_table

Scharr, J. (2013, October 23). What Is Tor - How Does Tor Work - How to Use Tor. Retrieved August 08, 2017, from https://www.tomsguide.com/us/what-is-tor-faq,news-17754.html

Durumeric, Z., Wustrow, E., & Halderman, J. (2013). ZMap: Fast Internet-Wide Scanning and its Security Applications. Retrieved from https://zmap.io/paper.pdf

Voegtlin, T. (2015). Simple Payment Verification — Electrum 2.5 documentation. Retrieved from http://docs.electrum.org/en/latest/spv.html

SPV, Simplified Payment Verification - Bitcoin Glossary. (2017). Retrieved from https://bitcoin.org/en/glossary/simplified-payment-verification

https://atlas.torproject.org/#search/flag:authority

http://torstatus.blutmagie.de/

15. Contributors

As an open source project we find it very important to thank our contributors who have given us a helping hand in order for us to get to where we are today.

To that we say

Thank you

The Author

CryptoRekt

Verge Marketing Team

Core Marketing Team

CryptoRekt - The Hammer
Sasha Kolupaev - VP Of Operations
Maeotsu - Graphics and Marketing Lead: **Japan**
Greg Franko - Web Design and Marketing Specialist
Kieran - Marketing Specialist
Frank Dashwood - The Man | Marketing Strategist
@SpookyKid - Marketing Strategist
Rondoparisiano - Marketing Strategist
Crypth - Marketing Strategist
Feyzi Ozsahin - Graphics Design and Marketing
@CYANO - Graphics Design and Marketing
Cees Van Dam - Social Media Expert
Patrick - International Project Manager

Contributors

@LuckLight - Community Manager
Harry - The Esskay, Software Developer
CryptoGrok - Marketing Advisor
Alexander Hourani - Marketing Lead: **Australia**
VergeKorea - Marketing Lead: **South Korea**
Lalo Trage - Marketing Lead: **Brazil**
Hristomir - Marketing Lead: **Bulgaria**
CapoDiCrypto - Marketing Lead: **Netherlands**
Frank v H - Marketing Contributor: **Netherlands**
Akshay P. - Marketing Contributor: **India**
Toko - Data Analysis Specialist: **Japan**
Kei Japan (lero003) - Digital Marketing: **Japan**
ripplechan - GUNDAM: **Japan**
Simon Cheng - Marketing Lead: **China**
@TongTong9 - Marketing Contributor: **China**
@Dejvid - Marketing Lead: **Poland**
MXCSM - Marketing Contributor
Mr. Wolf - **Official Verge Hype Man**
Joaquin - Marketing strategist
SmartTrader - Marketing Advisor
Frank v H - Marketing Contributor: Netherlands
Michael Stollaire - Marketing Ambassador
@Dejvid - Marketing Lead: **Poland**
Jason (g0ldm0ney10) - Marketing Strategist
Emanuel Goldstein - Community Manager

Contact Info

Twitter Telegram Slack Facebook IRC Reddit Steam Verge Discord BitCoinTalk Radio Station