COINsolidation

oneCKey (one Consolidated private Key) and CUA (Consolidated Universal Address) algorithms.

Guillermo Vidal
vidal@coinsolidation.org
www.coinsolidation.org / www.coinsolidation.io

Abstract: An algorithm to consolidate the private key of addresses of different blockchains, currently used in the financial area (cryptocurrencies), applications and various sectors. We propose two algorithms the first one we call **oneCKey** (**one** Consolidated prívate **Key**) is applied to consolidate private keys (One private key for N-addresses of different blockchain technologies). This is used to create more efficient private key management systems. This system is in sets of addresses in pairs, called series to be identified, the first Series A1 or "Genesis" series we apply oneCKey in Bitcoin and Ethereum blockchains with a ratio of 1:2 (one private key to generate two addresses) and later we will have Series AX and BX where there will be ratios of 1:N (one private key for N-addresses). We use another algorithm called **CUA** (Consolidated Universal Address) for the creation of universal addresses that we will call consolidated addresses. The CUA algorithm is broken down into three types of address consolidation which are; **CUA** "Consolidated Universal Address" an address formed from Cryptocurrency with Cryptocurrency, **DAC** "Direct Address Consolidated" an address formed from Token with Token and the **HAC** "Hybrid Address Consolidated" an address formed from Token with Cryptocurrency.

INTRODUCTION.

Currently 2021 there is a trend towards the use of blockchain systems in different sectors such as financial (cryptocurrencies) and emerging public and private sectors. The different blockchain technologies manage the generation of their own common addresses for the deposit operations of their corresponding assets and each address manages its assets independently by means of its unique private key referenced to a single public deposit address, currently there is only a 1:1 (one deposit address referenced to one and only one private key for asset transfers) so that the management of multiple addresses is complicated by always having a number of N-asset deposit addresses referenced to N-private keys for asset transfers, an N:N relationship. Where N is the number of addresses created by a user in different cryptocurrencies, this N:N relationship always holds.

On the other hand, let's also consider that the management is complicated for private keys in the same way with the N: N ratio, i.e. a user who has N escrow addresses will have directly proportional N private keys to manage. Applying the oneCKey algorithm will create a 1:N ratio, a ratio of one private key for several escrow addresses of different blockchain technologies (we consolidate the private keys to only one that will manage N escrow addresses). Finally, there is also the need that each user wishing to make a deposit or an asset transfer needs to know the destination address, the proposal is to apply the second algorithm called CUA - "Consolidated Universal Address", an address that consolidates two or more blockchain technologies resulting in a single address.

All of the above must be working with the cryptography systems currently used by each cryptocurrency. When consolidating the primary key, the validation and current cryptography of each native blockchain of the cryptocurrency to be used is used, so that when each series is developed with a 1:N+1 ratio, the same cryptography and hashing protocols will be applied, i.e. consolidation in security.

After surveying the vast world of the most important and outstanding cryptocurrencies on the market, we have found that there are two fundamental trends for the generation of addresses and private keys. The two used ECC (*Elliptic curve cryptography)* curves that cover 95% of the current "most prominent" cryptocurrencies, are:

I. The secp256k1 cryptographic curve used by Bitcoin and Ethereum. (**Series A**).

II. The Ed25519 cryptographic curve has a tendency to be used as one of the fastest ECC curves and is not covered by any *patents*. (**B series**).

Example:

Name	Type	Signing alg	Curve	Hash	Address encoding	Address hash
Bitcoin	UTXO	ECDSA	secp256k1	SHA-256	base58, bech32	SHA-256, RIPEMD-160
Ethereum	account	ECDSA	secp256k1	Keccak-256 *	none (just hex) *	last 20B of Keccak-256 *
XRP	account	ECDSA *	secp256k1 *	first half of SHA-512	base58 with different alphabet *	SHA-256, RIPEMD-160
Litecoin	UTXO	ECDSA	secp256k1	SHA-256 *	base58, bech32	SHA-256, RIPEMD-160
EOS	account	ECDSA	secp256k1	SHA-256	none *	none *
Bitcoin Cash	Same as Bitcoin *					
Stellar	account	EdDSA	ed25519	SHA-256 and SHA-512 in EdDSA *	base32	none
Binance Coin	Ethereum ERC-20 token *					
Tether	Bitcoin Omni layer / Ethereum ERC-20 token					
TRON	UTXO	ECDSA	secp256k1	SHA-256	base58	last 20 bytes of Keccak-256 *
Cardano	UTXO	EdDSA	ed25519	none and SHA-512 in EdDSA *	base58	none
Monero	UTXO *	*it's complicated**	ed25519	Keccak-256 *	base58	Keccak-256 *
IOTA	UTXO	Winternitz one time signature scheme	-	Curl, Kerl *	none	Kerl
Dash	UTXO	ECDSA	secp256k1	SHA-256 *	base58	SHA-256, RIPEMD-160
Maker	Ethereum ERC-20 token					
NEO	account	ECDSA	secp256r1	SHA-256	base58	SHA-256, RIPEMD-160
Ontology	account	ECDSA	nist256p1	3x SHA-256	base58	SHA-256, RIPEMD-160
Ethereum Classic	Same as Ethereum					
NEM	account	EdDSA	ed25519	none and Keccak-256 in EdDSA *	base32	Keccak-256, RIPEMD-160
Zcash	UTXO	ECDSA, zk-SNARKs *	secp256k1, Jubjub *	SHA-256	base58, bech32	SHA-256, RIPEMD-160
Tezos	account	EdDSA, ECDSA *	ed25519, secp256k1, secp256r1	BLAKE2 and SHA-512 in EdDSA *	base58	BLAKE2

OINSOLIDATION

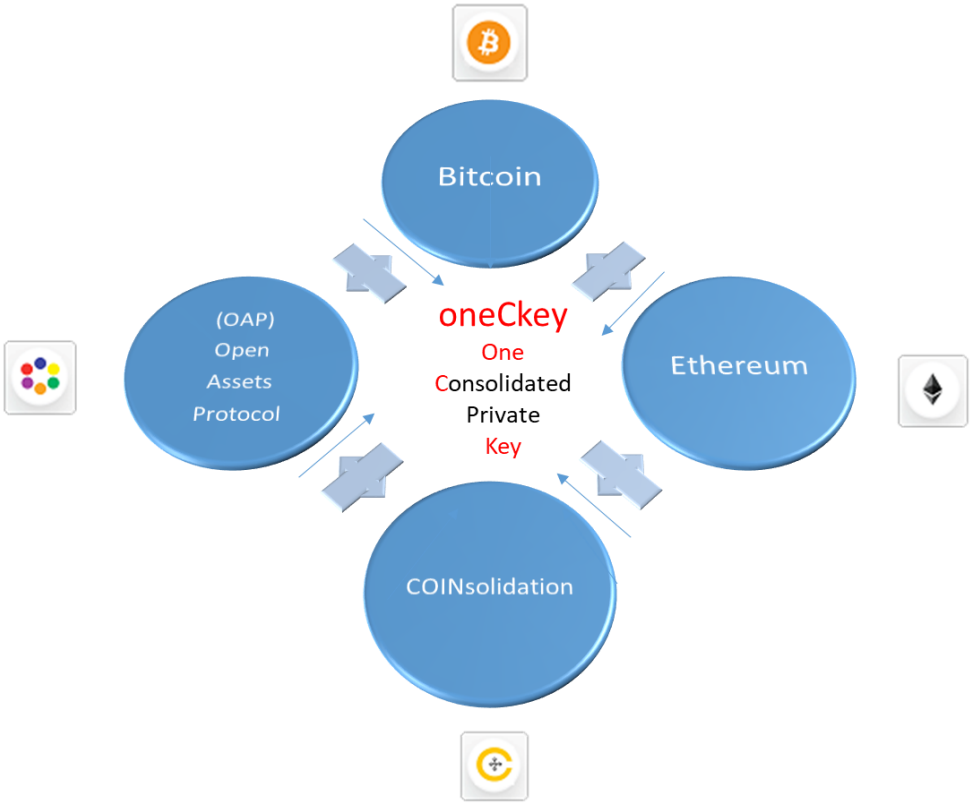
Cryptography for Series A or "Genesis" series.

The A series or "Genesis" series is the first consolidation of private keys. This series consists of the consolidation of the 2 main cryptocurrencies and 2 tokens (Bitcoin, Ethereum, OAP and COINsolidation).

The cryptography applied in both cases is ECDSA (*Elliptic Curve Digital Secure Algorithm*), which is currently used in the generation of public and private keys by both Bitcoin and Ethereum.

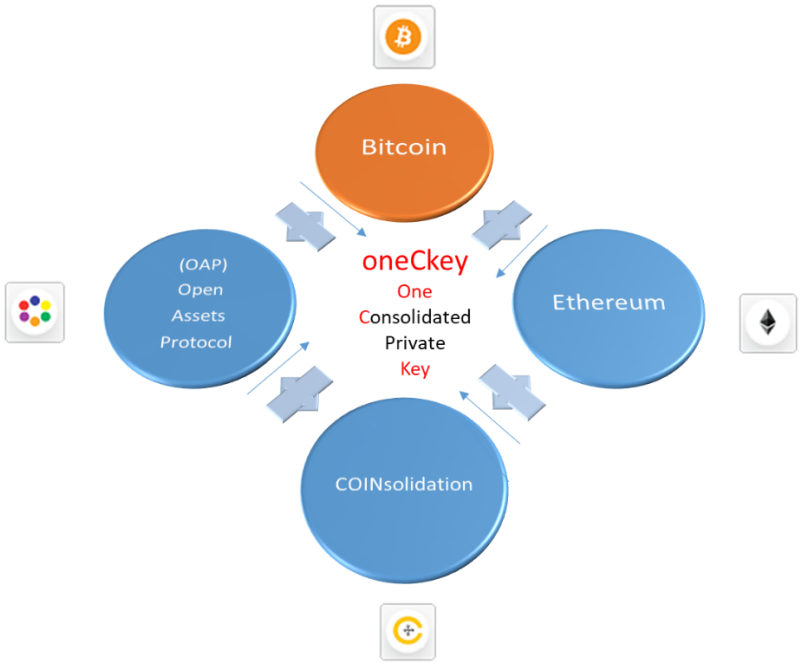
The generation of the consolidated private key integrated by both the Bitcoin and Ethereum blockchain can be applied in the following cases (3 options).

1.- Generation of a consolidated private key from QRNG (*Quantum Random Number Generator*)



We generate a random number from a QRNG source, propose the generation of the optical-quantum mechanics using a RAW format photo to produce sufficient entropy. This is done using the smartphone lens, we apply hash (SHA254) and make the conversion to hexadecimal to create the consolidated private key, we apply the oneCKey-secp256k1 algorithm to generate the respective deposit addresses in each Bitcoin and Ethereum blockchain. We apply the Bitcoin address in OAP (Open Assets Protocol) format to generate an ExoToken. Using oneCKey will give us a single private key for two different deposit addresses of different blockchains such as Bitcoin and Ethereum, as well as the generation of an OAP (Open Assets Protocol) token to be used by the user according to their interests, opening a possibility to expand their business prospects.

2.- Generation of a consolidated private key from an existing Bitcoin address. The Bitcoin private key is applied to the oneCKey-secp256k1 algorithm to generate an Ethereum address and we apply the Bitcoin address in OAP (Open Assets Protocol) format to generate ExoToken.



Example:

Existing Bitcoin private key (in this one we apply oneCKey to create other addresses):

28a0f97c6921e43872eb0640af41a54b9bde57c71cf4efe0db9d829f8b2cf645

Bitcoin address:

18XmTwfTeurjKQ8i1rEQT1DAx8BDjdR96A

Ethereum address:

0x9c789b22758c85f456dca3ac02e1fb00a059a4e

OAP (Open Assest Protocol) address based on the previous Bitcoin address.

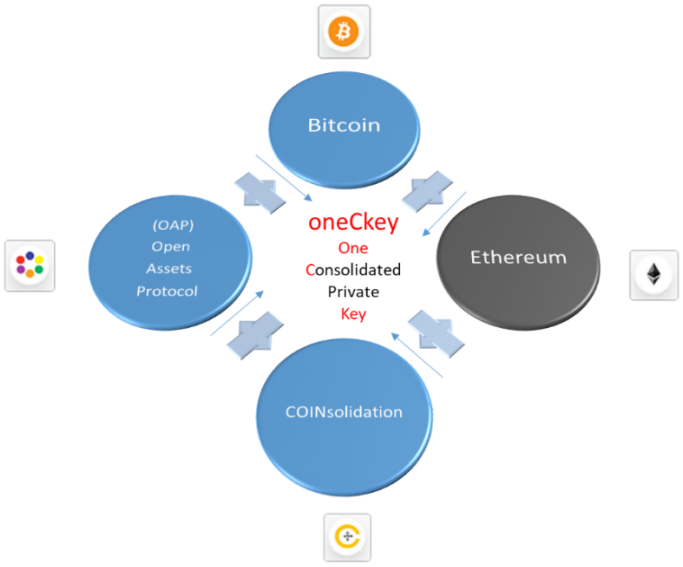
akJVVei7Uo8PkRBeJ54ULb6s7kHjMPhs8UjG

Address Token COINsolidation:

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

Generation of a consolidated private key from an existing Ethereum address. The Ethereum private key is applied to the oneCKey-secp256k1 algorithm to generate a Bitcoin address and we apply the OAP (Open Assets Protocol) format to the Bitcoin address to generate an ExoToken.

.



Cryptography for A1 Series.

A total of 5 cryptocurrencies and 2 tokens are involved.

In this case we will have 5 options to create the oneCKey, we can use every existing private key of the different supported cryptocurrencies.

OINSOLIDATION

Cryptography for A2 Series.

Example: If we take the DASH private key by applying oneCKey we will generate 5 addresses based on the DASH private key and the two tokens will also depend on it. A ratio of 1:7 (1 consolidated private key for 7 addresses).

DASH private key (we apply oneCKey to create other addresses).
20ba723de1fdeee66e927e30fdb3ada74ae23bfdb370da378f301ce4dbf27312

DASH Directorate:
XtAGtBbnzykDSwoWUSjgQUF4gG1h3uyYKW

Bitcoin address:
1JUS3vwu3GXdJ1CvcZRTYwZGqvRzwT7FEk

Ethereum address:
0x9d4a5854955c8e498e61eaaae7d3846917381f5b

Address OAP:
akUSKJ6mEWkRKAFNHfBXeCoTrBXcAyavXnS

Address COINsolidation Token:
0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

Litecoin address
LchPK9Fj7vmgYou5nhQkpxd348oHAJ3aSu

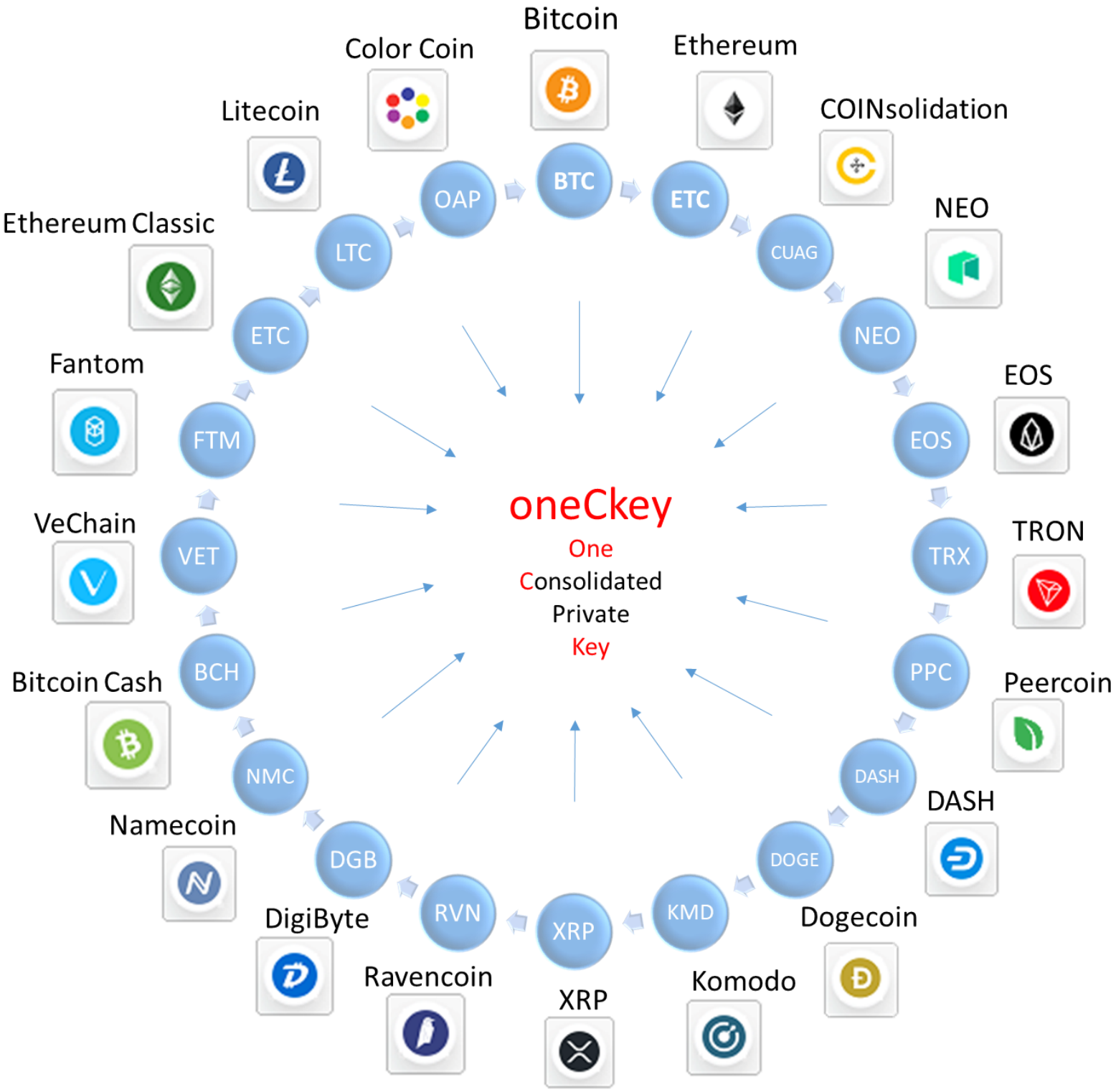
Address Dogecoin:
DNcXbBtYLgRuq1PXM9R26hisj4AJGJJj4pT

The above can be applied in the same way for any existing address of the above mentioned cryptocurrencies supported by oneCKEY in the B series. This results in a private key for use in 5 cryptocurrencies and 2 tokens.

OINSOLIDATION

Cryptography for A3 Series.

The A3 series applies to 18 cryptocurrencies and 2 tokens. This would be a 1:18 ratio (one consolidated private key for common use in 20 directions).



The above can be applied in the same way for any existing address of the aforementioned cryptocurrencies supported by oneCKEY in the A3 series.

Example: If we take the Bitcoin private key by applying oneCKey we will generate 18 addresses based on the Bitcoin private key and the two tokens will also depend on it. A ratio of 1:20 (one consolidated private key for 20 addresses).

Bitcoin private key (we apply oneCKey to create other addresses).
d2783ceae51a0c74fb733640f7128092dba3093605232b2b76fb102dca092d69

1. Bitcoin address:
   1N67xWPN9F1sEMXparSpzT24erp5F2p6LK
   https://www.blockchain.com/explorer

2. Ethereum address:
   0x91efb31bcd0bd12a088f9625344bbe92c1543bc3
   https://etherscan.io/

3. Address COINsolidation:
   0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41
   https://etherscan.io/

4. EOS Wallet Address:
   5KQykH8ssQkwsmUWffvrbgsMRf5wh8iHofZGvkHou5h1HYCqJYp

5. Address TRON:
   TPGr8Vp78vNcfhSvnpp7EnucaBKEBXns7h (*The account is not activated.)*
   https://trx.tokenview.com/

6. Address Peercoin:
   PVgJ7UnDCAW4DCCavvmMfLzLGbyxJ4HRpP
   https://blockbook.peercoin.net/

7. DASH Directorate:
   Xwmxnm3G6xETPJ8QSjm3qyhrVCPmECp3dt
   https://explorer.dash.org/insight/

8. Address Dogecoin:
   DSEDVmL1Sev9mMiRKSSPYDBfXzYNYadkdV
   https://blockchair.com/dogecoin

9. Address Komodo:
   RWNK32Gek4pSJMu242Rx5yMGR8GfprvY69
   https://kmdexplorer.io/

10. XRP address:
    r4afxWP49Er1NMXF2iSFzTpheiFnEpFaLK (*The account is not activated.)*
    https://bithomp.com/explorer/

11. Address Ravencoin:
    RWNK32Gek4pSJMu242Rx5yMGR8GfprvY69

https://ravencoin.network/

12. Address DigiByte:
    DSEDVmL1Sev9mMiRKSSPYDBfXzYNYadkdV
    https://digibyteblockexplorer.com/

13. Address Namecoin:
    NHfVA9tM4d7RktnKrfmQCyAndP6D8AQ2rA5
    https://www.cryptoground.com/namecoin-block-explorer

14. Address Bitcoin Cash:
    1N67xWPN9F1sEMXparSpzT24erp5F2p6LK (New format to see explorer site)
    https://explorer.bitcoin.com/bch
    New format address:
    bitcoincash:qrn49p6cy49tzgatt9gz9xz3gf8qxwl9tu800hul9m

15. Address VeChain:
    0x91efb31bcd0bd12a088f9625344bbe92c1543bc3
    https://explore.vechain.org

16. Address Fantom:
    0x91efb31bcd0bd12a088f9625344bbe92c1543bc3
    https://explorer.fantom.network/

17. Ethereum Classic address:
    0x91efb31bcd0bd12a088f9625344bbe92c1543bc3
    https://etcblockexplorer.com/

18. Litecoin address:
    LgK5DihCDuFvVVADykzS8GU5ps5BMGWfCXX
    https://blockchair.com/litecoin

19. Address Color coin:
    akY41CgChciuZ6bhBdUZ1eJvdzTzF9Sizv3
    https://blockchainexplorer.lykke.com/

20. WIF (Wallet Import Format) - NEO use secp256r1, Do not use secp256k1).
    To get address import WIF site: https://neotracker.io/wallet/open-wallet uses WIF as Private Key.

    WIF (Wallet Import Format) NEO:
    L4GqU2DmXVryD5JRwgphv6yF9q6pQ93FAyBTJE5BzshaFnwXPB4H
    Address NEO:
    AKkcm37QeMxZpFfRrrLGeqFRgkFzMPH9Gn
    https://neotracker.io/

21. Address: ExoCrypto:
    Exo41CgChciuZ6bhBdUZ1eJvdzTzF9Sizv3
    https://exoCrypto.com

COINSOLIDATION

Summary of cryptocurrencies supported and classified by the two ECC curves. (secp256k1 & Ed25519)

A3 Series (secp256k1) - Ready	B1 series (Ed25519) -Q42021
1. Bitcoin* Bitcoin* Bitcoin* Bitcoin* Bitcoin* Bitcoin* Bitcoin* Bitcoin* Bitcoin	1. Polkadot
2. OAP*** OAP*** OAP*** OAP*** OAP*** OAP*** OAP	2. Cardano
3. Ethereum**	3. Monero
4. COINsolidation***	4. COINsolidation***
5. XRP	5. Stellar
6. NEO	6. Algorand
7. EOS	7. IOTA
8. TRON	8. Elrong
9. Dash	9. Decred
10. Bitcoin Cash	10. Nano
11. Ethereum Classic**	11. Horizen
12. Litecoin	12. Siacoin
13. Dogecoin	13. Stacks
14. VeChain**	14. Lisk
15. DigiByte	15. Qtum
16. Fantom**	16. Waves
17. Ravencoin	17. …. Others.
18. Komodo	
19. Peercoin	
20. Namecoin	

(*) Bitcoin cryptocurrencies with the same address as the algorithm for generating them is the same, only the blockchain changes, which is different for each one's transactions.
(**) Ethereum cryptocurrencies with the same address as the algorithm for generating them is the same, only the blockchain changes, which is different for the transactions of each one.
(***) Tokens generated and dependent on the consolidated primary key.

Roadmap Series A4 includes Series A3 + Zcash, Binance BNB, Filecoin, Tether, Tezos, Cosmos, Zilliqa and Avalanche. Q4-2021

Roadmap starts Series B1 with Curve Ed25519. (Polkadot, Cardano, Monero, Stellar,Algorand, IOTA, Elrong, Algorand, and Waves) release Q4-2021.

NOTE: The crypto-assets of each series may vary depending on the behaviour of the cryptocurrency market and the list may be modified, depending on the liquidity of each asset.

Security implemented in oneCKey, PQC - (Post-Quantum Cryptography) algorithms embedded in the repository where the consolidated key generated locally in each Smartphone or an existing key entered by the user is stored.

For the secure encryption of the oneCKey we have implemented a combination of PQC security algorithms consisting of: AES-CGM + chacha20poly1305.

Chacha20poly1305: https://tools.ietf.org/html/rfc7539
AES-CGM: https://tools.ietf.org/html/rfc5288

U = PQC algorithm application operator.

Encryption of oneCKey = U (AES-CGM (chacha20poly1305 (OneCKey)))

We apply Mini PQC software for android with operator; º23Insert DB SQLite in (A).

https://github.com/openqbit-diy/MiniPQC

Calculation of AES-CGM to support quantum computing attacks.

Cryptosystem	Category	Key Size	Security Parameter	Quantum Algorithm Expected to Defeat Cryptosystem	# Logical Qubits Required	# Physical Qubits Required	Time Required to Break System	Quantum-Resilient Replacement Strategies
AES-GCM	Symmetric encryption	128 192 256	128 192 256	Grover's algorithm	2,953 4,449 6,681	4.61×10^6 1.68×10^7 3.36×10^7	2.61×10^{12} yrs 1.97×10^{22} yrs 2.29×10^{32} yrs	
RSA	Asymmetric encryption	1024 2048 4096	80 112 128	Shor's algorithm	2,290 4,338 8,434	2.56×10^6 6.2×10^6 1.47×10^7	3.58 hours 28.63 hours 229 hours	Move to NIST-selected PQC algorithm when available
ECC Discrete -log problem	Asymmetric encryption	256 386 512	128 192 256	Shor's algorithm	2,330 3,484 4,719	3.21×10^6 5.01×10^6 7.81×10^6	10.5 hours 37.67 hours 95 hours	Move to NIST-selected PQC algorithm when available
SHA256	Bitcoin mining	N/A	72	Grover's algorithm	2,403	2.23×10^6	1.8×10^4 years	
PBKDF2 with 10,000 iteration	Password hashing	N/A	66	Grover's algorithm	2,403	2.23×10^6	2.3×10^7 years	Move away from passwordbased authentication

COINSOLIDATION

CUA algorithm creation of consolidated addresses.

Nowadays, mergers are commonplace, whether for economic, technological or market reasons.

We present the first crypto-asset or crypto-token fusion model based on an algorithm to create a consolidated address that is used and generated in the COINsolidation environment.

We create three types of consolidated addresses.

- **CUA** (Consolidated Universal Address) is used to consolidate and create a new ExoToken (asset) to be used by the user. The combination will be Cryptocurrency with Cryptocurrency. In the case of CUA you have the first series created called CUAG (Coinsolidated Universal Address Genesis).
- **HAC** (Hibric Address Consolidated) is used when we need to consolidate a cryptocurrency and/or token address and a normal asset transfer address.
- **DAC** (Dual Address Consolidated) is used to manage and consolidate two normal token addresses from the same blockchain or from two different technologies, they are simple asset transfer addresses referred to as Token-Token.

Let's start by looking at the advantages of the CUA.

A CUA address is made up of the address of the COINsolidation token (static address) and an additional token known as a "Coloured Coin" (variable address). In this case we can see that CUA addresses will always be made up of addresses of some combination of assets (cryptocurrencies or tokens).

In our case when we consolidate the COINsolidation token and an OAP token we will know it as the "CUA genesis" or CUAG (Consolidated Universal Address Genesis).

The COINsolidation token is created on the Ethereum blockchain and uses the ERC20 (Ethereum Request for Comments 20) standard.

The Colored coin token is based on the Bitcoin blockchain and uses the Open Assest Protocol (OAP) standard.

Let's start by reviewing the potential and benefits of consolidating addresses.

- For users who create a CUA, a token (OAP) can be created that can be personalised by the user creating the CUA, the user will have the possibility of having his own token or crypto-asset so that he can use it in the creation, support or expansion of his business(es), in a simple and easy way he will have an asset in the world of crypto-tokens.

- For companies that create a CUA, they will be able to have an OAP token that they can use to create value in their supply chain or use the asset in liquidity transactions based on the economic support of their company's assets and liabilities.

- For existing cryptocurrencies and tokens, by creating a CUA, you can use your address that identifies your asset and by consolidating it with the token (OAP) you can grow your demand by offering your current and future investors their own token for your users.

Our Consolidated Universal Addresses (CUA) are created using the following algorithm:

OINSOLIDATION

Step 1.- Select the Bitcoin and Ethereum addresses.

Bitcoin Address - (BTC). - address A1

1Hjx3CanChCytqVz7vek1SSvN1momghJ42

Ethereum address - (ETH) - address A2

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

Step 2.- The SHA512(address String-Text) of each address is obtained by removing the first element of each address and taking from SHA512 the two characters located in places 120 to 121 of each hash operation symbolised by "U". Verifier numbers.

U(SHA512(**Hjx3CanChCytqVz7vek1SSvN1momghJ42**)) = **bf**

U(SHA512(**x9d08c0ac0ac0f2fdf078c883db6fa617b15776e4b41**)) = **28**

Step 3.- The first element (character) of each address is taken starting with the address with the smallest number of elements and the string "10" is obtained.

A10[0] = 1

A20[0] = 0

Step 3.- The SHA512 is obtained without the elements of step 3 and only the four characters from 120 to 123 are taken.

U(SHA512(**Hjx3CanChCytqVz7vek1SSvN1momghJ42x9d08c0ac0f2fdf078c883db6fa617b15776e4b41**)) = **140c**

Step 4.- The characters of each address are concatenated one by one, starting with the address that has the least number of characters, if they have the same number of characters, the concatenation can start from any address.

Address 1 = A10[0], A11[1], A12[2], A13[3], A14[4]..... A1N[n], A1N+1[n+1].

Address 2 = A20[0], A21[1], A22[2], A23[3], A24[4]..... A2N[n], A2N+1[n+1].

Address concatenation:

A10[0] + A20[0] + A10[1] + A20[1] + A11[2] + A22[2] + ... A1N+1[n+1] + A2N+2[n+1].

Last characters that cannot be concatenated are put at the beginning of the string and step 3 is appended to the end of the concatenated string.

776e4b41 Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb4125 **10**

Step 4.- The number of characters in the string that could NOT be concatenated in step 4 is added at the beginning.

OINSOLIDATION

8776e4b41Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb412510

Step 5.- At the beginning of the string two XX integer verifiers are added to help us to verify if the difference of the strings (subtraction) the bigger one minus the smaller one, must always give a positive integer, this pair of integers will help us to avoid errors in the concatenation. In case the difference is less than or equal to "9" the verifier number will be "00" in case it is greater than "9" the difference will be marked in these two digits.

In our case of generation between Bitcoin and Ethereum, the two verification digits will always be "00".

008776e4b41Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb412510

Step 6.- The two pairs of verifiers from step 2 in each direction are concatenated at the beginning of the string resulting from step 3 in the same order A1 + A2.

bf28008776e4b41Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb412510

Step 7.- The 4 digits of the SHA512 from step 3 are integrated at the end of the string.

bf28008776e4b41Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb412510140c

Step 8.- The **CUA** (Consolidated Universal Address "Genesis") ID is integrated at the beginning of the address created in step 5.

CUAfd55008776e4b41Hxj9xd30C8acn0CahcC0yft2qfVdzf70v7e8kc18S8S3vdNb16mfoam6g1h7Jb412510140c

In the case of Bitcoin and Ethereum address consolidation, it will give an address consisting of **90** hexadecimal characters.