import pandas as pd

from web3 import Web3

from loguru import logger

import random

import time

from tqdm import tqdm

from moralis import evm\_api

from config import

from eth\_abi import

from eth\_utils import

holo\_abi = '[{inputs[],stateMutabilitynonpayable,typeconstructor},{stateMutabilitypayable,typefallback},{inputs[],nameadmin,outputs[{internalTypeaddress,name,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,nametarget,typeaddress},{internalTypebytes,namedata,typebytes}],nameadminCall,outputs[],stateMutabilitypayable,typefunction},{inputs[{internalTypeuint256,name,typeuint256},{internalTypeuint32,namefromChain,typeuint32},{internalTypeaddress,nameholographableContract,typeaddress},{internalTypeaddress,namehToken,typeaddress},{internalTypeaddress,namehTokenRecipient,typeaddress},{internalTypeuint256,namehTokenValue,typeuint256},{internalTypebool,namedoNotRevert,typebool},{internalTypebytes,namebridgeInPayload,typebytes}],namebridgeInRequest,outputs[],stateMutabilitypayable,typefunction},{inputs[{internalTypeuint32,nametoChain,typeuint32},{internalTypeaddress,nameholographableContract,typeaddress},{internalTypeuint256,namegasLimit,typeuint256},{internalTypeuint256,namegasPrice,typeuint256},{internalTypebytes,namebridgeOutPayload,typebytes}],namebridgeOutRequest,outputs[],stateMutabilitypayable,typefunction},{inputs[],namegetAdmin,outputs[{internalTypeaddress,nameadminAddress,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeuint32,nametoChain,typeuint32},{internalTypeaddress,nameholographableContract,typeaddress},{internalTypeuint256,namegasLimit,typeuint256},{internalTypeuint256,namegasPrice,typeuint256},{internalTypebytes,namebridgeOutPayload,typebytes}],namegetBridgeOutRequestPayload,outputs[{internalTypebytes,namesamplePayload,typebytes}],stateMutabilitynonpayable,typefunction},{inputs[],namegetFactory,outputs[{internalTypeaddress,namefactory,typeaddress}],stateMutabilityview,typefunction},{inputs[],namegetHolograph,outputs[{internalTypeaddress,nameholograph,typeaddress}],stateMutabilityview,typefunction},{inputs[],namegetJobNonce,outputs[{internalTypeuint256,namejobNonce,typeuint256}],stateMutabilityview,typefunction},{inputs[{internalTypeuint32,name,typeuint32},{internalTypeuint256,name,typeuint256},{internalTypeuint256,name,typeuint256},{internalTypebytes,name,typebytes}],namegetMessageFee,outputs[{internalTypeuint256,name,typeuint256},{internalTypeuint256,name,typeuint256},{internalTypeuint256,name,typeuint256}],stateMutabilityview,typefunction},{inputs[],namegetOperator,outputs[{internalTypeaddress,nameoperator,typeaddress}],stateMutabilityview,typefunction},{inputs[],namegetRegistry,outputs[{internalTypeaddress,nameregistry,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypebytes,nameinitPayload,typebytes}],nameinit,outputs[{internalTypebytes4,name,typebytes4}],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,namesender,typeaddress},{internalTypeuint32,nametoChain,typeuint32},{internalTypeaddress,nameholographableContract,typeaddress},{internalTypebytes,namebridgeOutPayload,typebytes}],namerevertedBridgeOutRequest,outputs[{internalTypestring,namerevertReason,typestring}],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,nameadminAddress,typeaddress}],namesetAdmin,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,namefactory,typeaddress}],namesetFactory,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,nameholograph,typeaddress}],namesetHolograph,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,nameoperator,typeaddress}],namesetOperator,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,nameregistry,typeaddress}],namesetRegistry,outputs[],stateMutabilitynonpayable,typefunction},{stateMutabilitypayable,typereceive}]'

lzEndpointABI = '[{inputs[{internalTypeuint16,name\_chainId,typeuint16}],stateMutabilitynonpayable,typeconstructor},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,nameversion,typeuint16}],nameDefaultReceiveVersionSet,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,nameversion,typeuint16}],nameDefaultSendVersionSet,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,nameversion,typeuint16}],nameNewLibraryVersionAdded,typeevent},{anonymousfalse,inputs[{indexedtrue,internalTypeaddress,namepreviousOwner,typeaddress},{indexedtrue,internalTypeaddress,namenewOwner,typeaddress}],nameOwnershipTransferred,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,namesrcChainId,typeuint16},{indexedfalse,internalTypebytes,namesrcAddress,typebytes},{indexedfalse,internalTypeuint64,namenonce,typeuint64},{indexedfalse,internalTypeaddress,namedstAddress,typeaddress}],namePayloadCleared,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,namesrcChainId,typeuint16},{indexedfalse,internalTypebytes,namesrcAddress,typebytes},{indexedfalse,internalTypeaddress,namedstAddress,typeaddress},{indexedfalse,internalTypeuint64,namenonce,typeuint64},{indexedfalse,internalTypebytes,namepayload,typebytes},{indexedfalse,internalTypebytes,namereason,typebytes}],namePayloadStored,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeuint16,namechainId,typeuint16},{indexedfalse,internalTypebytes,namesrcAddress,typebytes}],nameUaForceResumeReceive,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeaddress,nameua,typeaddress},{indexedfalse,internalTypeuint16,nameversion,typeuint16}],nameUaReceiveVersionSet,typeevent},{anonymousfalse,inputs[{indexedfalse,internalTypeaddress,nameua,typeaddress},{indexedfalse,internalTypeuint16,nameversion,typeuint16}],nameUaSendVersionSet,typeevent},{inputs[],nameBLOCK\_VERSION,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[],nameDEFAULT\_VERSION,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[],namechainId,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[],namedefaultReceiveLibraryAddress,outputs[{internalTypeaddress,name,typeaddress}],stateMutabilityview,typefunction},{inputs[],namedefaultReceiveVersion,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[],namedefaultSendLibrary,outputs[{internalTypecontract ILayerZeroMessagingLibrary,name,typeaddress}],stateMutabilityview,typefunction},{inputs[],namedefaultSendVersion,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_dstChainId,typeuint16},{internalTypeaddress,name\_userApplication,typeaddress},{internalTypebytes,name\_payload,typebytes},{internalTypebool,name\_payInZRO,typebool},{internalTypebytes,name\_adapterParams,typebytes}],nameestimateFees,outputs[{internalTypeuint256,namenativeFee,typeuint256},{internalTypeuint256,namezroFee,typeuint256}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_srcChainId,typeuint16},{internalTypebytes,name\_srcAddress,typebytes}],nameforceResumeReceive,outputs[],stateMutabilitynonpayable,typefunction},{inputs[],namegetChainId,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_version,typeuint16},{internalTypeuint16,name\_chainId,typeuint16},{internalTypeaddress,name\_userApplication,typeaddress},{internalTypeuint256,name\_configType,typeuint256}],namegetConfig,outputs[{internalTypebytes,name,typebytes}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_srcChainId,typeuint16},{internalTypebytes,name\_srcAddress,typebytes}],namegetInboundNonce,outputs[{internalTypeuint64,name,typeuint64}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_dstChainId,typeuint16},{internalTypeaddress,name\_srcAddress,typeaddress}],namegetOutboundNonce,outputs[{internalTypeuint64,name,typeuint64}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,name\_userApplication,typeaddress}],namegetReceiveLibraryAddress,outputs[{internalTypeaddress,namereceiveLibraryAddress,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,name\_userApplication,typeaddress}],namegetReceiveVersion,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,name\_userApplication,typeaddress}],namegetSendLibraryAddress,outputs[{internalTypeaddress,namesendLibraryAddress,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,name\_userApplication,typeaddress}],namegetSendVersion,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_srcChainId,typeuint16},{internalTypebytes,name\_srcAddress,typebytes}],namehasStoredPayload,outputs[{internalTypebool,name,typebool}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name,typeuint16},{internalTypebytes,name,typebytes}],nameinboundNonce,outputs[{internalTypeuint64,name,typeuint64}],stateMutabilityview,typefunction},{inputs[],nameisReceivingPayload,outputs[{internalTypebool,name,typebool}],stateMutabilityview,typefunction},{inputs[],nameisSendingPayload,outputs[{internalTypebool,name,typebool}],stateMutabilityview,typefunction},{inputs[],namelatestVersion,outputs[{internalTypeuint16,name,typeuint16}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name,typeuint16}],namelibraryLookup,outputs[{internalTypecontract ILayerZeroMessagingLibrary,name,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,name\_newLayerZeroLibraryAddress,typeaddress}],namenewVersion,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name,typeuint16},{internalTypeaddress,name,typeaddress}],nameoutboundNonce,outputs[{internalTypeuint64,name,typeuint64}],stateMutabilityview,typefunction},{inputs[],nameowner,outputs[{internalTypeaddress,name,typeaddress}],stateMutabilityview,typefunction},{inputs[{internalTypeuint16,name\_srcChainId,typeuint16},{internalTypebytes,name\_srcAddress,typebytes},{internalTypeaddress,name\_dstAddress,typeaddress},{internalTypeuint64,name\_nonce,typeuint64},{internalTypeuint256,name\_gasLimit,typeuint256},{internalTypebytes,name\_payload,typebytes}],namereceivePayload,outputs[],stateMutabilitynonpayable,typefunction},{inputs[],namerenounceOwnership,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_srcChainId,typeuint16},{internalTypebytes,name\_srcAddress,typebytes},{internalTypebytes,name\_payload,typebytes}],nameretryPayload,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_dstChainId,typeuint16},{internalTypebytes,name\_destination,typebytes},{internalTypebytes,name\_payload,typebytes},{internalTypeaddress payable,name\_refundAddress,typeaddress},{internalTypeaddress,name\_zroPaymentAddress,typeaddress},{internalTypebytes,name\_adapterParams,typebytes}],namesend,outputs[],stateMutabilitypayable,typefunction},{inputs[{internalTypeuint16,name\_version,typeuint16},{internalTypeuint16,name\_chainId,typeuint16},{internalTypeuint256,name\_configType,typeuint256},{internalTypebytes,name\_config,typebytes}],namesetConfig,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_newDefaultReceiveVersion,typeuint16}],namesetDefaultReceiveVersion,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_newDefaultSendVersion,typeuint16}],namesetDefaultSendVersion,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_newVersion,typeuint16}],namesetReceiveVersion,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name\_newVersion,typeuint16}],namesetSendVersion,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeuint16,name,typeuint16},{internalTypebytes,name,typebytes}],namestoredPayload,outputs[{internalTypeuint64,namepayloadLength,typeuint64},{internalTypeaddress,namedstAddress,typeaddress},{internalTypebytes32,namepayloadHash,typebytes32}],stateMutabilityview,typefunction},{inputs[{internalTypeaddress,namenewOwner,typeaddress}],nametransferOwnership,outputs[],stateMutabilitynonpayable,typefunction},{inputs[{internalTypeaddress,name,typeaddress}],nameuaConfigLookup,outputs[{internalTypeuint16,namesendVersion,typeuint16},{internalTypeuint16,namereceiveVersion,typeuint16},{internalTypeaddress,namereceiveLibraryAddress,typeaddress},{internalTypecontract ILayerZeroMessagingLibrary,namesendLibrary,typeaddress}],stateMutabilityview,typefunction}]'

holograph\_ids = {'polygon'4,

'avax' 3,

'bsc' 2}

Lz\_ids= {'bsc'102,

'avax'106,

'polygon'109}

gas\_holo = {'bsc'4500000000,

'avax'37500000000,

'polygon'400000000000}

wallets = []

results = []

class Bridger

def \_\_init\_\_(self, privatekey,chain,to,delay,api,mode=1)

self.privatekey = privatekey

self.chain = chain

self.to = to

self.w3 = ''

self.scan = ''

self.account = ''

self.address = ''

self.mode = mode

self.delay = delay

self.moralisapi = api

self.HolographBridgeAddress = Web3.to\_checksum\_address('0xD85b5E176A30EdD1915D6728FaeBD25669b60d8b')

self.LzEndAddress = Web3.to\_checksum\_address('0x3c2269811836af69497E5F486A85D7316753cf62')

self.BLDGAddress = Web3.to\_checksum\_address('0x8C531f965C05Fab8135d951e2aD0ad75CF3405A2')

def check\_status\_tx(self, tx\_hash)

logger.info(f'{self.address}{self.chain} - waiting for confirmation transaction {self.scan}{self.w3.to\_hex(tx\_hash)}...')

while True

try

status = self.w3.eth.get\_transaction\_receipt(tx\_hash)

status = status['status']

if status in [0, 1]

return status

time.sleep(1)

except Exception as error

time.sleep(1)

def sleep\_indicator(self,secs)

for i in tqdm(range(secs), desc='wait', bar\_format={desc} {n\_fmt}s {total\_fmt}s {bar}, colour='green')

time.sleep(1)

def check\_nft(self)

if self.mode == 0 and self.chain != 'opti'

cc = {'avax''avalanche',

'polygon''polygon',

'bsc''bsc'}

api\_key = self.moralisapi

params = {

chain cc[self.chain],

format decimal,

token\_addresses [

self.BLDGAddress

],

media\_items False,

address self.address}

try

result = evm\_api.nft.get\_wallet\_nfts(api\_key=api\_key, params=params)

id\_ = int(result['result'][0]['token\_id'])

if id\_

logger.success(f'{self.address} - BLDG {id\_} nft founded on {self.chain}...')

return id\_

except Exception as e

return False

elif self.mode == 0 and self.chain == 'opti'

contract\_abi = [

{

constant True,

inputs [{name \_owner, type address}],

name balanceOf,

outputs [{name balance, type uint256}],

payable False,

stateMutability view,

type function

},

{

constant True,

inputs [{name \_owner, type address}],

name tokensOfOwner,

outputs [{name tokenIds, type uint256[]}],

payable False,

stateMutability view,

type function

}

]

contract = self.w3.eth.contract(address=self.BLDGAddress, abi=contract\_abi)

bal = contract.functions.balanceOf(self.address).call()

if bal

id\_ = contract.functions.tokensOfOwner(self.address).call()[0]

logger.success(f'{self.address} - BLDG {id\_} nft founded on {self.chain}...')

return id\_

else

return False

elif self.mode == 1

for chain in ['avalanche', 'polygon', 'bsc']

api\_key = self.moralisapi

params = {

chain chain,

format decimal,

token\_addresses [

self.BLDGAddress

],

media\_items False,

address self.address}

try

result = evm\_api.nft.get\_wallet\_nfts(api\_key=api\_key, params=params)

id\_ = int(result['result'][0]['token\_id'])

if id\_

logger.success(f'{self.address} - BLDG {id\_} nft founded on {chain}...')

if chain == 'avalanche' chain = 'avax'

return chain, id\_

except Exception as e

if 'list index out of range' in str(e)

continue

logger.error(f'{self.address} - BLDG nft not in wallet...')

return False

def start(self)

if self.mode == 0

self.w3 = Web3(Web3.HTTPProvider(info[self.chain][1]))

self.scan = info[self.chain][0]

self.account = self.w3.eth.account.from\_key(self.privatekey)

self.address = self.account.address

data = self.check\_nft()

if data

nft\_id = data

else

return self.address,'BLDG nft not in wallet'

elif self.mode == 1

self.w3 = Web3(Web3.HTTPProvider(info['bsc'][1]))

self.address = self.w3.eth.account.from\_key(self.privatekey).address

data = self.check\_nft()

if data

chain,nft\_id = data

self.chain = chain

self.w3 = Web3(Web3.HTTPProvider(info[self.chain][1]))

self.scan = info[self.chain][0]

self.account = self.w3.eth.account.from\_key(self.privatekey)

self.address = self.account.address

if chain == self.to

chains = ['avax', 'polygon', 'bsc']

chains.remove(self.to)

self.to = random.choice(chains)

else

return self.address,'BLDG nft not in wallet'

payload = to\_hex(encode(['address', 'address', 'uint256'], [self.address, self.address, nft\_id]))

gas\_price = gas\_holo[self.to]

gas\_lim = random.randint(450000, 500000)

holograph = self.w3.eth.contract(address=self.HolographBridgeAddress, abi=holo\_abi)

lzEndpoint = self.w3.eth.contract(address=self.LzEndAddress, abi=lzEndpointABI)

lzFee = lzEndpoint.functions.estimateFees(Lz\_ids[self.to],self.HolographBridgeAddress,'0x',False,'0x').call()[0]

lzFee = int(lzFee 2.5)

to = holograph\_ids[self.to]

while True

logger.info(f'{self.address}{self.chain} - trying to bridge... ')

try

tx = holograph.functions.bridgeOutRequest(to, self.BLDGAddress, gas\_lim, gas\_price,payload).build\_transaction({

'from' self.address,

'value' lzFee,

'gas' holograph.functions.bridgeOutRequest(to, self.BLDGAddress, gas\_lim, gas\_price,payload).estimate\_gas({'from' self.address, 'value' lzFee,'nonce' self.w3.eth.get\_transaction\_count(self.address), }),

'gasPrice' self.w3.eth.gas\_price,

'nonce' self.w3.eth.get\_transaction\_count(self.address),

})

sign = self.account.sign\_transaction(tx)

hash = self.w3.eth.send\_raw\_transaction(sign.rawTransaction)

status = self.check\_status\_tx(hash)

self.sleep\_indicator(10)

if status == 1

logger.success(f'{self.address}{self.chain} - successfully bridged BLDG {nft\_id} to {self.to} {self.scan}{self.w3.to\_hex(hash)}...')

self.sleep\_indicator(random.randint(self.delay[0],self.delay[1]))

return self.address, 'success'

except Exception as e

error = str(e)

if 'nonce too low' in error or 'already known' in error

logger.success(f'{self.address}{self.chain} - successfully bridged BLDG {nft\_id} to {self.to}...')

self.sleep\_indicator(random.randint(self.delay[0],self.delay[1]))

return self.address, 'success'

logger.error(f'{self.address}{self.chain} - error {e}')

return self.address, 'error'

def main()

logger.info(f'{ 32}creator - httpst.meiliocka{ 32}')

with open(keys.txt, r) as f

keys = [row.strip() for row in f]

random.shuffle(keys)

for key in keys

to\_chain = random.choice(to) if type(to) == list else to

holo = Bridger(key,chain,to\_chain,delay,api,mode)

res = holo.start()

wallets.append(res[0]), results.append(res[1])

res = {'address' wallets, 'result' results}

df = pd.DataFrame(res)

df.to\_csv('results.csv', index=False)

logger.success('Bridging done...')

print(f'n{ 32}creator - httpst.meiliocka{ 32}n')

print(f'n{ 32}donate - EVM 0xFD6594D11b13C6b1756E328cc13aC26742dBa868{ 32}n')

print(f'n{ 32}donate - trc20 TMmL915TX2CAPkh9SgF31U4Trr32NStRBp{ 32}n')

logger.success('...')

if \_\_name\_\_ == '\_\_main\_\_'

main()