



This repository Search

Pull requests Issues Gist



munichpavel / csm

Watch 2

Unstar 2

Fork 0

<> Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

Branch: master

csm / CSM_Example.ipynb

Find file

Copy path



munichpavel Update CSM_Example.ipynb

6395c59 just now

2 contributors

473 lines (472 sloc) 22.9 KB

Raw

Blame

History



Example of cryptostatemachine

Simulate P2P thing-lending with microinsurance with Python module *cryptostatemachine*

Characters

Alice: Has digital camera she rarely uses

Bob: Wants digital camera for family bbq next weekend

AZ: Offers novel, exciting microinsurance solutions on the blockchain

Data structures

StateMachine: a directed-graph based Python class implementation of a state-machine with interactions encoded in both the graph structure and ornamentation. As state evolves, cryptographic Bitcoin-like transaction data is exported to a database (currently json file, next step: blockchain)

SimpleSM: descendant of *StateMachine*, with fixed graph structure for simplest P2P thing-lending work flows.

```
In [118]: import os
import sys
import ecdsa as ec #Pure Python elliptic curve cryptography
import binascii as ba
```

```
In [119]: # Attach local path where
module_path = os.path.abspath(os.path.join('PycharmProjects/csm'))

if module_path not in sys.path:
    sys.path.append(module_path)
```

```
In [120]: import cryptostatemachine.cryptostatemachine as csm
reload(csm)
```

```
Out[120]: <module 'cryptostatemachine.cryptostatemachine' from '/home/pavel/PycharmProjects/csm/cryptostate
machine/cryptostatemachine.pyc'>
```

Define the characters

```
In [121]: # Issuer
alice_priv = ec.SigningKey.generate()
alice_priv_hex = ba.b2a_hex(alice_priv.to_string())
alice_pub = alice_priv.get_verifying_key()
alice_pub_hex = ba.b2a_hex(alice_pub.to_string())
print 'Alice\'s public key:', alice_pub_hex, '\n'

# Interested party
bob_priv = ec.SigningKey.generate()
bob_priv_hex = ba.b2a_hex(bob_priv.to_string())
bob_pub = bob_priv.get_verifying_key()
bob_pub_hex = ba.b2a_hex(bob_pub.to_string())
print 'Bob\'s public key:', bob_pub_hex, '\n'

# Microinsurer
az_priv = ec.SigningKey.generate()
az_priv_hex = ba.b2a_hex(az_priv.to_string())
az_pub = az_priv.get_verifying_key()
```







