

# Practical 1 : A simple client classs that generates the private and public keys using the built-in python RSA algorithm and test it

```
In [1]: import random
import string
import binascii
import collections
```

```
In [2]: import Crypto #pip install crypto
```

```
In [3]: pip install pycryptodome
```

Requirement already satisfied: pycryptodome in c:\users\stewi\anaconda3\lib\site-packages (3.14.1)  
Note: you may need to restart the kernel to use updated packages.

```
In [4]: import Crypto.Random
```

```
In [5]: from Crypto.Hash import SHA
```

```
In [6]: from Crypto.PublicKey import RSA
```

```
In [7]: from Crypto.Signature import PKCS1_v1_5
```

```
In [8]: import binascii
class Client:
    def __init__(self):
        random = Crypto.Random.new().read
        self._private_key = RSA.generate(1024, random)
        self._public_key = self._private_key.publickey()
        self._signer = PKCS1_v1_5.new(self._private_key)

    @property
    def identity(self):
        return binascii.hexlify(self._public_key.exportKey(format='DER')).decode("as
```

```
In [9]: Blockchain=Client()
```

```
In [10]: print(Blockchain.identity)
```

```
30819f300d06092a864886f70d010101050003818d0030818902818100958a16043d75234fbfb0815457
a37cee5a60f9007116f503b25be198b548d392c0bd707094ab8a894755a9a7141ea4bb27730e9843ee18
adce5a01d47a64595efbfd0639644d7e20a247874967767839c8896f41b60c22b37343cdf0eae6d9ea3f
37d5f7d5bf15ebd7b0440c5f6abf6f719c0bddcc07e635c0d786e9eca24190203010001
```