**تمرین کامپیوتری شماره 2:**
**ایجاد تراکنش در شبکه بیتکوین و**
**راه‌اندازی نود لوکال اتریوم**
مهلت تحویل: ۵ خرداد 1403

**ارزهای رمزنگاری شده**

---


مدرس: **دکتر شریعت پناهی**

طراحان: سپهر آذردار، پوریا تاج‌محرابی، محمد صادقی، علیرضا اربابی


# فهرست


**ارزهای رمزنگاری شده ..... 1**
**فهرست ..... 1**
**بخش اول: تعامل با شبکه بیت کوین ..... 2**
قسمت اول: Address Generation ..... 3
قسمت دوم: ایجاد تراکنش ..... 3
**بخش دوم: راه اندازی نود لوکال اتریوم ..... 6**
گام اول: نصب geth ..... 6
گام دوم: کانفیگ کردن مشخصات بلوک اولیه ..... 7
گام سوم: ایجاد 3 نود لوکال اتریوم ..... 7
گام چهارم: ایجاد اکانت و شروع به کار گره ها ..... 8
گام چهارم:Initialize کردن نودها و اجرای آن ..... 8
گام پنجم: وصل شدن به نودها از طریق یک کلاینت ..... 8
گام ششم: اتصال نودها به یکدیگر ..... 9
گام هفتم: ایجاد تراکنش ..... 10
گام هشتم: ماین کردن یک بلاک جهت قرار گیری تراکنش در بلاکچین ..... 10

# بخش اول: تعامل با شبکه بیت کوین

در این تمرین قصد داریم تا با شبکه بیت کوین به صورت عملی تعامل کنید و در آن تراکنش انجام دهید و سایز مکانیزم های بیت‌کوین ها را نیز بررسی کنید. با توجه به اینکه در شبکه اصلی بیت کوین (Main Net) برای انجام تراکنش نیاز به صرف هزینه واقعی و پرداخت بیت کوین (به عنوان کارمزد تراکنش) است، موارد خواسته شده را در شبکه آزمایشی (Test Net) انجام می دهید. شبکه آزمایشی از نظر فنی کاملا مشابه شبکه اصلی بیت کوین است و در صورتی که بتوانید عملیات مد نظر را در این شبکه به طور صحیح به انجام رسانید،این عملیات در شبکه اصلی نیز قابل اجرا خواهد بود. تفاوت جدی شبکه اصلی و شبکه آزمایشی در این است که در شبکه آزمایشی بیت کوین ها ارزشی نداشته و با این هدف طراحی شده که توسعه‌دهندگان بیت کوین برای طراحی، و ساخت و توسعه امکانات جدید و تست آن‌ها هزینه ای صرف نکنند.
برای دریافت پول در شبکه آزمایشی برخی از وب سایت ها که Faucet نامیده می شوند به کار شما می آید، به این صورت که می توانید با ارایه آدرس خود در شبکه آزمایشی به آن ها بیت کوین دریافت کنید
برخی از Faucet های شناخته شده در بیت کوین عبارتند از:

1. https://coinfaucet.en/en/btc-testnet
2. https://bitcoinfaucet.uo1.net/
3. https://testnet-faucet.mempool.co/
4. https://kuttler.eu/en/bitcoin/btc/faucet/

برای مشاهده زنجیره بلوکی شبکه آزمایشی بیت کوین می تواند از مرورگر های زنجیره بلوکی زیر استفاده کنید:

1. https://www.blockchain.com/explorer?view=btc-testnet
2. https://blockchair.com/bitcoin/testnet
3. https://live.blockcypher.com/btc-testnet/
4. https://blockstream.info/testnet/

## قسمت اول: Address Generation

اولین قدم در شروع کار با شبکه آزمایشی بیت کوین تولید یک آدرس معتبر است. زیر برای دریافت پول ابتدا می‌بایست یک آدرس معتبر تولید کنید و سپس از آن استفاده کرده و تراکنش های لازم را انجام دهید.
در بیت کوین آدرس با فرمت مشخصی تولید می گردد که در آن از توابع رمزنگاری در هم‌سازSHA-256 و RIPEMD-160 استفاده می‌شود.

**سوال 1)** کدی به زبان پایتون و در فرمت ipynb که آدرسی برای شبکه آزمایشی تولید کنید.
خروجی این کد می‌بایست یک آدرس در پایه 58 باشد BASE58 و کلید خصوصی متناظر با آن به فرم WIF باشد. همچنین توضیح دهید که میان آدرس های شبکه اصلی و آزمایشی چه تفاوتی وجود دارد.

**سوال 2)** مشابه قسمت قبل کدی بنویسید که یک آدرس خاص فوت نوت تولید کند. این کد با دریافت 3 کاراکتر، باید آدرسی تولید کند که با این 3 کاراکتر شروع می شود ( ا توجه به اینکه کاراکتر اول بیت کوین معمولا دارای مقادیر خاص و محدودی است،حروف داده شده باید در جایگاه دوم تا چهارم آدرس شما ظاهر شوند. **توجه کنید** که فرآیند تولید آدرس باید به طول کامل توسط شما انجام شود و استفاده کردن از کد های آماده package ها مجاز نیست و اگر کدی را از اینترنت استفاده می کنید باید به آن تسلط داشته باشید)

## قسمت دوم: ایجاد تراکنش

برای انجام تراکنش در شبکه آزمایشی بیت کوین از کتابخانه python-bitcoinlib استفاده می کنیم.
این کتابخانه دارای توابع مختلف برای ایجاد انواع تراکنش است. برای انجام این قسمت از تمرین ابتدا با نحوه کار کردن ماشین پشته ای بیت کوین و Script ها و دستورات آن از روی منابعی که در انتها معرفی شده است و یا سایر منابع معتبر آشنایی لازم را پیدا کنید.

با توجه به اینکه برای انجام تراکنش در بیت کوین نیاز به اتصال به شبکه توزیع شده آن است برای راحتی کار در این تمرین تراکنش ها به API هایی که توسط برخی از وب سایت های مرورگر زنجیره بلوکی ارائه شده است ارسال می شود و آن ها به نیابت از شما تراکنش را در شبکه Broadcast می کنند. برای شروع یک کد استارتر در فایل CA2-TODO.ipynb در سکشن util قرار داده شده است که با استفاده از آن می توانید یک تراکنش را با یک ورودی و یک خروجی از نوع P2PKH ایجاد کنید. همچنین در سکشن transaction برخی توابع کاربردی ایجاد شده است که آدرس API مربوط Push TX یکی از وب سایت های معروف نیز در آن قرار داده شده است که در صورت لزوم می توانید آن را تغییر داده و تراکنش ها را به سایر API های موجود ارسال کنید. در توابع مشخص شده در سکشن util قسمتهای مربوط به scriptSig و scriptPubKey را با داده ها و دستورات مناسب در قسمت return کامل کنید تا بتوانید یک تراکنش ساده با آن انجام دهید. با استفاده از کد های قسمت اول تمرین و یا استفاده از تولید کننده های آدرس آنلاین برای شبکه آزمایشی بیت کوین آدرس ایجاد کرده ( آدرس مورد استفاده را به همراه کلید خصوصی آن در گزارش خود ذکر کنید) و

پس از دریافت پول از Faucet ها تراکنش های زیر را انجام دهید ( لازم است برای هر سوال علاوه بر توضیحات لازم  دو سکشن در فایل CA2-TODO.ipynb ایجاد کنید که یکی برای تراکنش ایجاد UTXO های مورد نظر و دیگری برای تراکنش خرج کردن آن ها باشد):

**ترانکش ۱)** تراکنشی با یک ورودی و دو خروجی ایجاد کنید که خروجي اول آن توسط هیچکس قابل خرج شدن نباشد و خروجی دوم آن توسط هر شخصی قابل خرج شدن باشد. در تراکنشی دیگر خروجی قابل خرج این تراکنش را خرج کرده و به آدرس اصلي خود به صورت خروجی P2PKH بازگردانید.

**تراکنش 2)** سه آدرس جدید تولید کنید و مشخصات آن را در گزارش ذکر کنید. تراکنشی ایجاد کنید که یک ورودی و یک خروجی داشته باشد که خروجی آن از نوع P2MS یا Multisig بوده و توسط ۲ نفر از این ۳ آدرس آن قابل خرج شدن باشد. در تراکنشی دیگر این خروجی را خرج کرده و پول آن را به آدرس اصلی خود بازگردانید.

**تراکنش 3)**

در این سوال می خواهیم با زبان برنامه نویسی bitcoin و استک آن آشنا بشویم. برای اینکار میخواهیم تراکنشی با یک ورودی و یک خروجی تولید کنیم به طوری که تنها از دو طریق میتوان آن را خرج کرد:

1. فراهم کردن سال تولد و سال فعلی و بررسی اینکه آیا شخص بزرگتر از 18 میباشد.
2. فراهم کردن یک رمز عبور از قبل تعیین شده که در اینجا شماره دانشجویی هر فرد میباشد. برای اینکه امنیت و محرمانگی رمز عبور بالا برود، script را به گونه ای بنوسید که هش رمزعبور مورد بررسی قرار بگیرد.

دو تراکنش با scriptpubkey یکسان که قابلیت خرج شدن به هریک از این روش‌ها را دارند، تولید کنید و هر دو روش را امتحان کنید.
(راهنمایی: برای پیاده سازی، میتوانید، OP_CODE های استک بیت کوین را از این سایت چک کنید و همچنین برای VALIDATE کردن script ای که نوشتید، قبل از broadcast کردن آن، میتوانید از این سایت استفاده کنید. برای ساخت scriptpubkey از op_if میتوانید استفاده کنید. همچنین نحوه نوشتن بخش scriptSig و تعداد ورودی‌هایی که شامل‌اش میشود، به انتخاب شما میباشد.)

لازم است که در تمامی سوالات مشخصات کامل آدرسهاي استفاده شده شامل کلید خصوصی آنها به فرم WIF و شناسایی تراکنشها را به طور کامل در گزارش خود ذکر کنید.

**سوال 1)** به تراکنش هایی مانند تراکنش unspendable انجام شده در سوال1، Burning Transactions گفته میشود. درباره الگوریتم اجماع Proof of Burn تحقیق کنید. به نظر شما Burn کردن کوین‌ها چگونه می‌تواند از منظر اقتصادی باعث ثبات ارزش یک شبکه رمزارز شود؟

**سوال 2)** درباره زبان scripting بیتکوین تحقیق کنید. آیا زبان فوق Turing Complete است؟ استفاده از یک زبان Turing Complete چه مزایایی می‌تواند داشته باشد؟

برای اطلاعات بیشتر در مورد فرم WIF به لینک زیر مراجعه کنید:

https://learnmeabitcoin.com/technical/wif

برای اطلاعات بیشتر در مورد آدرسهای بیتکوین به لینک زیر مراجعه کنید:

https://en.bitcoin.it/wiki/Address

برای اطلاعات بیشتر در مورد پیشوند آدرس در بیتکوین به لینک زیر مراجعه کنید:

https://en.bitcoin.it/wiki/List_of_address_prefixes

برای آشنایی بیشتر با دستورات بیتکوین و نحوه‌ی کار کردن ماشین پشته ای آن به لینک زیر مراجعه کنید:

https://en.bitcoin.it/wiki/Script

برای اطلاعات بیشتر در مورد ساختار بلوک به لینک زیر مراجعه کنید:

https://en.bitcoin.it/wiki/Block

## بخش دوم: راه اندازی نود لوکال اتریوم

در این بخش از تمرین یک بلاکچین اتریوم را به صورت لوکال در کامپیوتر خود بالا می‌آوریم و با نحوه کارکرد نودهای رمزارزها و نحوه عملیات ماینینگ آشنا می‌شویم. توجه شود که بلاکچین اتریوم در این تمرین تفاوتی با بلاکچین بیتکوین نخواهد داشت و صرفا به دلیل ساده‌تر بودن راه اندازی نود اتریوم نسبت به بیتکوین، از اتریوم استفاده می‌نماییم. بدین منظور ما از geth استفاده خواهیم کرد. geth یک پیاده سازی از execution layer اتریوم به زبان GO است که به ما جهت راه اندازی نود اتریوم کمک می‌کند.

*** در این قسمت از تمرین پس از اجرای هر دستور و هر گام، از صفحه خود اسکرین شات بگیرید و در گزارش قرار دهید.**

### گام اول: نصب geth

برای نصب geth در ubuntu می‌توانید از دستورات زیر استفاده کنید یا اگر سیستم عامل دیگری هستید از این صفحه کمک بگیرید یا آن را از اول build کنید.

```
sudo add-apt-repository -y ppa:ethereum/ethereum
```

```
sudo apt-get update
```

```
sudo apt-get install ethereum
```

پس از نصب می‌توانید لیستی از امکانات آن را با استفاده از دستور فوق مشاهده نمایید. توضیح کوتاهی در مورد هر یک از موارد خروجی این دستور ارائه دهید.

```
geth --help
```

## گام دوم: کانفیگ کردن مشخصات بلوک اولیه[1]

تمام گره ها باید از یک بلوک اولیه شروع به تولید بلاک چین نمایند. بدین منظور اطلاعات زیر را در یک فایل json ذخیره می‌کنیم. قسمت های آدرس را با آدرس های گره های بدست آمده در قسمت های بعدی عوض می‌کنیم.

**نکته! در قسمت های مشخص شده با *********، شماره دانشجویی خود را قرار دهید.**

```
genesis.json file:

{
    "config": {
        "chainId": 15,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "400000",
    "gasLimit": "2100000",
    "alloc": {
        "[Account #1 Address]": { "balance": "1000000000*********" },
        "[Account #2 Address]": { "balance": "2000000000*********" },
        "[Account #3 Address]": { "balance": "1500000000*********" },
    }
}
```

## گام سوم: ایجاد 3 نود لوکال اتریوم

با استفاده از دستور فوق، اقدام به initialize کردن 3 نود اتریوم نمایید.

```
$ mkdir node01 node02 node03
```

**سوال 1:** کارکرد هر نود (Node) در یک شبکه رمزارز را توضیح دهید. درباره مفاهیم Full Node و Light Node تحقیق کرده و به اختصار توضیح دهید.

[1] Genesis block

## گام چهارم: ایجاد اکانت و شروع به کار گره ها

با استفاده از دستور فوق، 3 عدد اکانت بسازید و برای هر یک پسورد انتخاب کنید. سپس آدرس های بدست آمده را در فایل genesis.json که در گام اول به آن پرداختید، قرار دهید.

```
$ geth --datadir "/PATH_TO_NODE01/" account new
```

## گام چهارم:Initialize کردن نودها و اجرای آن

با استفاده از دستور init، تمام نود های ایجاد شده را initialize کنید تا آماده اجرا شوند:

```
$ geth --datadir "/PATH_TO_NODE/" init /PATH_TO/genesis.json
```

حال اقدام به استارت زدن هرکدام از نودها در ترمینالی متفاوت با استفاده از دستور فوق نمایید (جهت سهولت در split کردن یک ترمینال به چند ترمینال می‌توانید از اکستنشن tmux استفاده کنید). توجه شود که برای هر نود میبایست یک port و rpc-port مجزا قرار دهید.

```
$ geth --identity "name_of_your_node" --rpc --rpcport "8000" --rpccorsdomain "*"
--datadir "/PATH_TO_NODE/" --port "30303" --nodiscover --rpcapi
"db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any"
```

* در صورتی که در فلگ rpcport– و rpc و یا –rpccorsdomain به مشکل خوردید، با استفاده از لاگ بدست آمده از فلگ جدیدتر HTTP استفاده کنید.

## گام پنجم: وصل شدن به نودها از طریق یک کلاینت

هم اکنون با استفاده از دستور فوق در یک ترمینال مجزا از ترمینال های قبلی، به عنوان یک کلاینت به یکی از نودها به صورت لوکال متصل شوید. (به پورت نودها در دستور فوق توجه کنید)

```
$ geth attach http://127.0.0.1:8000
```

پس از وصل شدن با استفاده از دستور فوق اطلاعات مربوط به گره را می‌توانید مشاهده کنید:

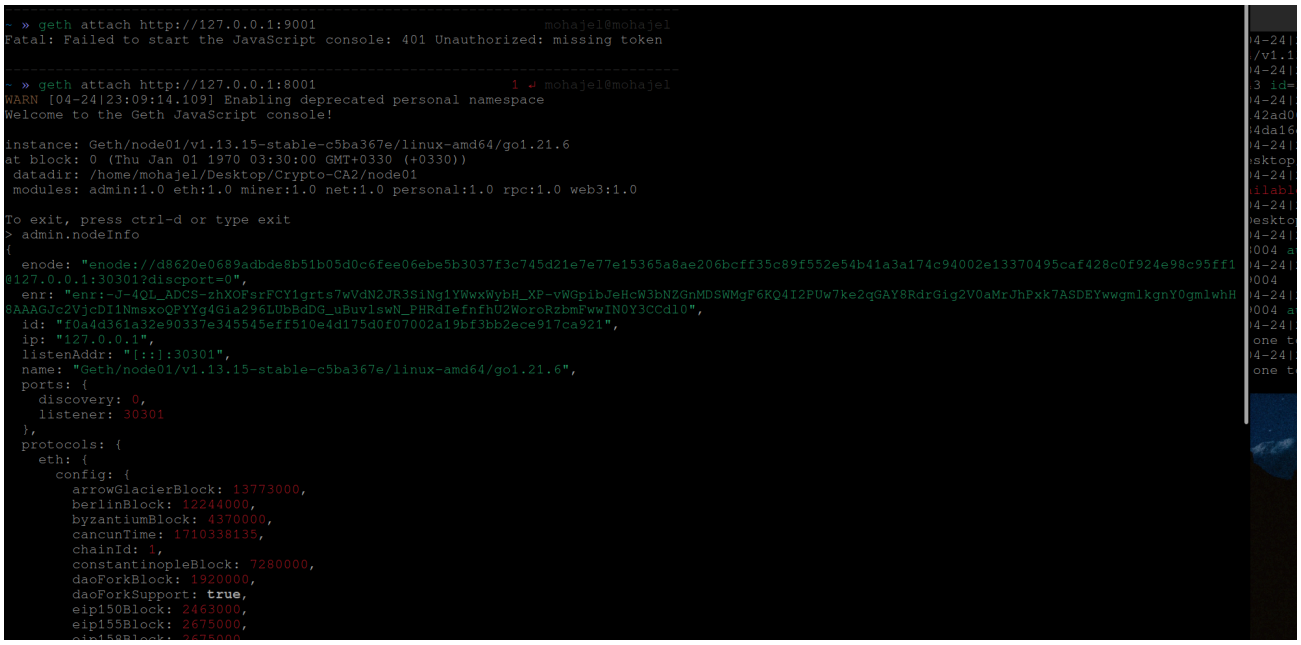
```
> admin.nodeInfo
```

## گام ششم: اتصال نودها به یکدیگر

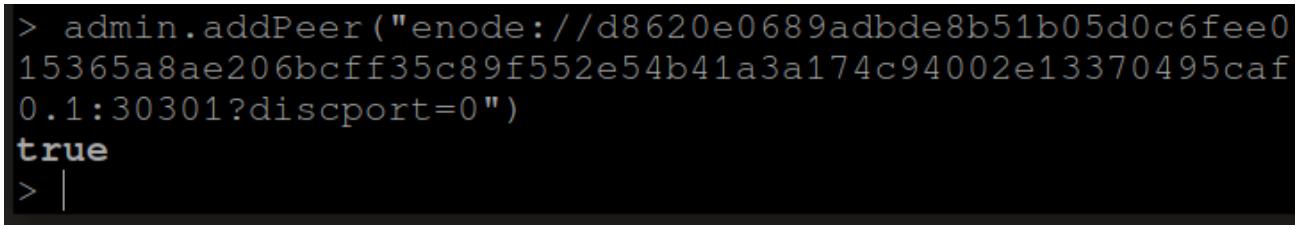
تا قسمت فوق، 3 نود ساخته ایم و با استفاده از یک کلاینت به یکی از نودها وصل شده ایم (نام این نود را نود مرکزی می‌گذاریم)، اما هنوز نودها را به یکدیگر متصل نکرده ایم. با این وجود جهت داشتن یک سیستم Decentralized، نیازمندیم تا نودها به یکدیگر متصل شوند و همگی یک بلاکچین یکسان را maintain نمایند. به همین جهت، ابتدا مقدار enode نود مرکزی را با استفاده از دستور admin.nodeInfo مشاهده و کپی کرده، و سپس با استفاده از دستور فوق، نودها را به نود مرکزی اضافه کنید:

```
> admin.addPeer("enode://26f7b8...92e@[::]:30303?discport=0")
```

نمونه ران کردن نود مرکزی و مشاهده enode:




نمونه اضافه کردن یک نود به نود مرکزی:



حال با استفاده از دستور فوق اقدام به مشاهده تعداد نودهای متصل به یکدیگر نمایید:

```
> net.peerCount
```

## گام هفتم: ایجاد تراکنش

با استفاده از دستور فوق نیز می‌توانید اکانت های موجود را ببینید، و با دستور بعدی نیز بالانس هر اکانت را مشاهده نمایید:

```
> eth.accounts
```

```
> eth.getBalance(eth.accounts[0])
```

```
eth.accounts
"0x922a2025ba30b2b56e9a968cfe8899b610c8acf4"]
eth.getBalance(eth.accounts[0])
```

دقت داشته باشید برای انجام تراکنش توسط http حتما فلگ --allow-insecure-unlock را در قسمت شروع به کار گره ها استفاده نمایید. برای شروع تراکنش با استفاده از دستور زیر و دادن رمز عبور اکانت مورد نظر که می‌خواهد تراکنش انجام دهد را باز کنید:

```
personal.unlockAccount(eth.accounts[0])
```

حال با استفاده از دستور زیر می‌توانید تراکنش مد نظرتان را به نودها ارسال کنید.

```
eth.sendTransaction({from:eth.accounts[0], to:eth.accounts[1], value:1000})
```

## گام هشتم: ماین کردن یک بلاک جهت قرار گیری تراکنش در بلاکچین

در قسمت قبل یک تراکنش را ایجاد کردیم اما تا زمانی که تراکنش ما توسط یک ماینر در یک بلاک قرار نگیرد و ماین نشود، به زنجیره بلاکچین اضافه نشده و ثبت نمیشود. به همین جهت، توسط یکی از نودهای ایجاد شده اقدام به ماین کردن تراکنش ساخته شده در گام قبل می‌نماییم:

برای عمل فوق، ابتدا در ترمینال یکی از نودها (ترجیحا نود اصلی) دستور زیر را وارد کرده و یک آدرس انتخاب کنید تا جایزه ماینینگ (Block Reward) در آن ریخته شود.

```
> miner.setEtherbase(eth.accounts[0])
```

حال با اجرای دستور فوق، اقدام به استارت زدن عملیات Mining نمایید. جزییات عملیات ماینینگ را می‌توانید در ترمینال مشاهده نمایید:

```
> miner.start()
```

**سوال 2:** خروجی مشاهده شده در عملیات ماینینگ را بررسی کنید و هرکدام از لاگ های بدست آمده را به اختصار توضیح دهید.

**سوال 3:** از سال 2015 تا کنون، 19722000 بلاک در زنجیره اصلی شبکه اتریوم ماین شده است. فرض کنید ما در این تمرین به عنوان یک مهاجم، اقدام به ماین کردن سریع در بلاکچین لوکال خود نماییم و سعی کنیم تا 19722000 در بلاکچین لوکال خود ایجاد کنیم و سپس با broadcast کردن آن در شبکه اینترنت، ادعا کنیم که بلاکچین ما بلاکچین اصلی اتریوم است! آیا قادر به انجام چنین کاری می‌باشیم؟ اگر بله، علت اتفاق را توضیح دهید و اگر خیر، به چه دلیلی قادر به انجام این کار نمی‌باشیم؟

## نکات تکمیلی:

- جهت انجام بخش اول پروژه، فایل نوت بوک موجود در سایت را دانلود کرده و قسمت های خالی نوت بوک را تکمیل کنید.
- جهت انجام بخش دوم پروژه، اسکرین شات های خروجی را به همراه پاسخ به سوالات تشریحی را در یک فایل PDF قرار دهید.
- **تقلب نکنید! پروژه جهت یادگیری شما می‌باشد** و در صورت وجود ابهام، با طراحان تمرین در ارتباط باشید. توجه کنید که پروژه فوق به صورت آنلاین تحویل گرفته خواهد شد و لازم است تا بر کد خود تسلط کافی داشته باشید.
- لطفا فایل نوت بوک ژوپیتر نهایی پروژه خود را به صورت فایل **ZIP** به فرمت **CA#2-FamilyName-StudentID.zip** در سامانه بارگزاری نمایید.