بسم الله الرحمن الرحیم

گزارش تمرین عملی سری دوم

پوریا دادخواه - ۹۶۱۰۶۴۸۵

سوال ۳-۱)

الف) باید چک کنیم کدام یک از حالت های ۱ یا ۲ اتفاق افتاده است. در حالت ۱ که امضای دو صاحب اصلی را داریم ، sig شامل دو امضا است و با احتساب فیلد dummy (برای باگ مربوط به multisig) در مجموع ۳ عدد دارد ولی در حالت ۲ ، ۴ امضا به همراه dummy داریم . بنابرین اگر

```
OP_DEPTH, OP_3, OP_EQUAL
```

برقرار باشد ، که کد مولتی سیگ عادی مربوط به حالت ۱ را میزنیم که دو ورودی sig و دو وردی pub است در غیر این صورت حالت ۲ است که ابتدا یک sig با دو pub داریم(یکی از دو صاحب اصلی) و سپس ۳ sig و ۵ pub است.

```
txout_scriptPubKey = [OP_DEPTH, OP_3, OP_EQUAL, OP_IF, OP_2, my_public_key,
Faraz_public_key, OP_2, OP_CHECKMULTISIG, OP_ELSE, OP_1, my_public_key,
Faraz_public_key, OP_2, OP_CHECKMULTISIGVERIFY, OP_3,
Shareholder1_public_key, Shareholder2_public_key, Shareholder3_public_key,
Shareholder4_public_key, Shareholder5_public_key, OP_5, OP_CHECKMULTISIG,
OP_ENDIF]
```

ب) برای آزاد سازی نیز همان sig هایی که در بخش قبل توضیح داده شد ، که در اول آن ها یک فیلد اضافی(OP_0) اضافه می کنیم تا باگ مربوط به OP_CHECKMULTISIG که استک را خالی میگذارد برطرف شود.

Sig 1 :

```
[OP_0, create_OP_CHECKSIG_signature(txin, txout, txin_scriptPubKey,
my_private_key),
create_OP_CHECKSIG_signature(txin, txout, txin_scriptPubKey,
Faraz_private_key)]
```

Sig2 :

```
[OP_0, create_OP_CHECKSIG_signature(txin, txout, txin_scriptPubKey,
Shareholder1_private_key),create_OP_CHECKSIG_signature(txin, txout,
txin_scriptPubKey,
Shareholder2_private_key),create_OP_CHECKSIG_signature(txin, txout,
txin_scriptPubKey, Shareholder3_private_key), OP_0,
create_OP_CHECKSIG_signature(txin, txout, txin_scriptPubKey, my_private_key)]
```

۳-۲) این قسمت تنها یک حالت داریم و نیازی به if نیست و درواقع کافیست قسمت بعد از else قسمت قبل را بزنیم.

سوال ۴)

برای لاک کردن یک تراکنش از

```
[signature pubkey]
[timelock,OP_CHECKLOCKTIMEVERIFY,OP_DROP,OP_DUP,OP_HASH160,my_address,OP_EQUA
LVERIFY,
OP_CHECKSIG]
```

استفاده میکنیم . تنها نکته این است که برای آزاد کردن تراکنش باید فیلد nSequence که در تابع

```
create_txin
```

از ویژگی های txin است را از حالت دیفالت که ماکس است تغییر داده و صفر کنیم. این ویژگی به این خاطر است که از مدتی قبل ویژگی تایم لاک در حالت عادی برداشته شده و یعنی اگر nSequence ماکس باشد اصلا تابع CHECKLOCKTIMEVERIFY چک نخواهد شد.

همچنین باید nLocktime که مربوط به منقضی شدن تراکنش است را هم تغییر دستی داده تا علنا مضوح پیدا کند که تراکنش معتبر است وگرنه ماینر ها آن را ماین نمیکنند. این پارامتر مربوط به تابع

```
CMutableTransaction
```

است.

۴-۲) این خصوصیت که یک دیتا به تراکنش اضافه کنیم در سوال ۵ آمده است که ۲راه برای آن ارایه داده شده. اولین راه که در این سوال هم استفاده میشود استفاده از آپکد OP_RETURN است. همانطور که در سایت bitcoin wiki/ scripts توضیح داده شده ، یکی از کاربرد های این دستور این است که فیلدی که پس از آن در اسکریپت قرار میگیرد را در ترکنش قرار می دهد.

در این سوال برخلاف ۵ نیازی به هش گیری نیست و فقط باید string (HAPPY BIRTHHDAY) را به byte (utf-8) تبدیل کنیم.

سوال ۵)

۵-۱)

راه اول : در قسمت قبل توضیح داده شد.

راه دوم : ایده کلی که در صورت سوال گفته شده. برای پیاده سازی ، ابتدا با استفاده از تابع

```
sha256sum
```

در تمرین قبل ، هش فایل را گرفته و با تابع

```
CBitcoinSecret.from_secret_bytes
```

که در config.py بود ، یک private key تولید میکنیم و آدرس آن پرایوت هم که از روی آن می سازیم.

۵-۲)

این قسمت تنها کار اضافه ای که دارد این است که با استفاده از merkleroot.py که در تمرین قبل با آن مرکل روت n هش را به دست می آوردیم ، مرکل روت فایل های موجود(در فولدر resource) به دست می آوریم و کار قبل را روی آن انجام می دهیم.

سوال ۶)

پروتکل این راه که هم در کد و هم در اسلاید ها توضیح داده شده است. تنها وظیفه ما برای تکمیل پیاده سازی تعیین کردن pubkey و sig keyاست.

Sig1) در حالتی که معامله انجام شود هر کس باید بتواند هویت خود را اثبات کند و اینکه secret را نیز میداند:

```
[sig_recipient, secret]
```

Sig2) وقتی پول به حساب خود شخص باز گردد با توجه به توافق اولیه یعنی فرد مقصد نیز این nLocktime را از اول قبول کرده بود و تراکنشی با این تاریخ فعال سازی را امضا نموده. حالا فرد مبدا کافیست امضای خود را در کنار آن امضا قرار دهد تا تراکنش را آزاد کند: (از آن جایی که در pubkey از مولتی سیگ استفاده کردیم باید یک فیلد dummy نیز قرار دهیم:

```
OP_0, sig_sender, sig_recipient
```

Pubkey) اینجا نیز دو حالت داریم ؛ ابتدا دیفالت فرض میکنیم حالت ۱ اتفاق می افتد و سیگ۱ را دریافت میکنیم. آن راDUP میکنیم (زیرا اگر حالت ۱ نبوده باشد فیلد از بین نرود) هش آن را گرفته و با hash_of_secret مقایسه می کنیم. اگر اوکی بود فیلد اضافی secret را DROP میکنیم و پابلیک گیرنده را با امضای آن چک میکنیمOP_CHECKSIG. در غیر این صورت که یک امضای مولتی سیگ عادی را verify می کنیم.

```
OP_DUP, OP_HASH160, hash_of_secret, OP_EQUAL, OP_IF, OP_DROP,
public_key_recipient, OP_CHECKSIG, OP_ELSE, OP_2,public_key_recipient,
public_key_sender, OP_2, OP_CHECKMULTISIG, OP_ENDIF]
```