علی آراسته ۹۶۱۰۱۱۶۵

مبانی فناوری بلاکچین و رمزارزها – گزارش تمرین عملی سری دوم

سوال ۳:

در این سوال، ابتدا با استفاده از فایل keygen.py، برای فراز، عطا و ۵ سهام‌دار شرکت، هر کدام به طور جداگانه، یک جفت کلید خصوصی و همگانی و متناسب با کلید همگانی، یک آدرس تولید می‌کنیم.

در قسمت‌های 31 و 31، از ساختار OP_IF، OP_ELSE و OP_ENDIF استفاده می‌کنیم.

نحوه عملکرد دستورهای ذکر شده بدین صورت است که اگر مقدار روی stack در هنگام اجرای دستور OP_IF، True باشد، دستورات بین OP_IF و OP_ELSE و در غیر این صورت دستورات بین OP_ELSE و OP_ENDIF انجام می‌شوند و سپس دستور بعد از OP_ENDIF بررسی می‌شود.

البته لازم به ذکر است در شبکه Bitcoin Testnet، Minerها در صورتی که در فرآیند Validate کردن تراکنش با دستور OP_CHECKMULTISIG مواجه شوند، این تراکنش را Mine نمی‌کنند.

۳-۱) در این قسمت در Script_PubKey با استفاده از یک ساختار OP_IF، OP_ELSE و OP_ENDIF ابتدا شرط i با استفاده از OP_CHECKSIG و در صورت برقرار نبودن آن، شرط ii با استفاده از OP_CHECKSIGVERIFY برای امضای عطا و OP_CHECKMULTISIG برای امضای سهام‌داران بررسی می‌شوند.

۳-۲) در این قسمت در Script_PubKey با استفاده از دو ساختار OP_IF، OP_ELSE و OP_ENDIF تو در تو ابتدا شرط i با استفاده از OP_CHECKSIG، سپس در صورت برقرار نبودن آن، شرط iii با استفاده از OP_CHECKMULTISIG و در آخر در صورت برقرار نبودن دو شرط قبلی، شرط ii با استفاده از OP_CHECKSIGVERIFY برای امضای فراز و عطا و OP_CHECKMULTISIG برای امضای سهام‌داران بررسی می‌شوند.

۳-۳) در این قسمت در Script_PubKey از آنجایی که فراز و عطا هیچگاه روی یک تراکنش توافق نمی‌کنند، می‌توان شرط مورد نظر را با استفاده از OP_CHECKSIGVERIFY برای امضای فراز و عطا و OP_CHECKMULTISIG برای امضای سهام‌داران بررسی کرد.

سوال ۴:

در این سوال از دستور OP_RETURN استفاده می‌کنیم.

از زمان bitcoin 0.9، یک روش استاندارد الصاق کردن data به تراکنش، قرار دادن یک Output با مقدار صفر و Script_PubKey به صورت [OP_RETURN, data] در انتهای تراکنش است.

سوال ۵:

در این سوال از دستورات OP_CHECKLOCKTIMEVERIFY و OP_DROP استفاده می‌کنیم.

در این سوال Script_PubKey در واقع همان Script تراکنش استاندارد است که دستورات لازم برای قفل کردن تراکنش تا زمانی مشخص یا ارتفاع Blockchain مشخص به آن اضافه شده است.

در هنگام اجرای دستور OP_CHECKLOCKTIMEVERIFY مقدار روی stack با شرایط فعلی مقایسه می‌شود (اگر این مقدار کوچکتر از 500000000 باشد مقایسه با ارتفاع Blockchain فعلی و در غیر این صورت با UNIX time فعلی بر حسب میلی ثانیه صورت می‌گیرد). اگر شرایط فعلی هنوز به این مقدار نرسیده باشد، Redeem کردن تراکنش Fail می‌شود.

البته لازم به ذکر است در شبکه Bitcoin Testnet، Minerها در صورتی که در فرآیند Validate کردن تراکنش با دستور OP_CHECKLOCKTIMEVERIFY مواجه شوند، این تراکنش را Mine نمی‌کنند.

سوال ۶:

در این سوال از ساختار OP_NOTIF و OP_ENDIF استفاده می‌کنیم.

در این سوال در Script_PubKey ابتدا با استفاده از OP_CHECKSIG وجود امضای حامد و سپس در صورت عدم وجود آن، با استفاده OP_CHECKLOCKTIMEVERIFY فرا رسیدن زمان اتمام قرار بررسی می‌شود. اگر هیچ یک از دو شرط بالا برقرار نباشد، Redeem شدن تراکنش Fail می‌شود و در غیر این صورت با استفاده از OP_CHECKSIG وجود امضای سعید مورد بررسی قرار می‌گیرد.

سوال ۹:

Part 1) در این قسمت، ابتدا اطلاعات موجود در فایل data.hex را به طور کامل می‌خوانیم و سپس با استفاده از کلاس sha256() از پکیج hashlib و دستور digest() از این کلاس، hash اطلاعات فایل data.hex را محاسبه می‌کنیم.

در ادامه از این مقدار به عنوان کلید خصوصی استفاده می‌کنیم و کلید همگانی و آدرس مربوط به آن را بدست می‌آوریم و در آخر در تراکنش ۱ ساتوشی به این آدرس منتقل می‌کنیم.

سالار می‌تواند بعد از منتشر شدن مقاله، از طریق خرج کردن UTXO ایجاد شده با استفاده از hash مقاله، داشتن مقاله در زمان ایجاد UTXO را ثابت کند.

Part 2) در این قسمت، ابتدا با استفاده از درخت Merkle، یک hash به عنوان ریشه درخت Merkle بدست می‌آوریم و سپس بقیه موارد را مانند قسمت قبل انجام می‌دهیم.

سالار می‌تواند بعد از منتشر شدن مقالات، از طریق خرج کردن UTXO ایجاد شده با استفاده از hash ریشه درخت Merkle، داشتن همه مقالات در زمان ایجاد UTXO را ثابت کند.

سوال ۱۰:

۱) در این قسمت از ساختار OP_IF، OP_ELSE و OP_ENDIF استفاده می‌کنیم.

در این سوال در Script_PubKey ابتدا با استفاده از OP_EQUAL وجود secret و سپس در صورت عدم وجود آن، با استفاده OP_CHECKSIGVERIFY وجود امضای فرستنده بررسی می‌شود. اگر هیچ یک از دو شرط بالا برقرار نباشد، Redeem شدن تراکنش Fail می‌شود و در غیر این صورت با استفاده از OP_CHECKSIG وجود امضای گیرنده مورد بررسی قرار می‌گیرد.

۲) Script_Sig در هر قسمت به صورت زیر خواهد بود:

a) [sig_recipient, secret]

b) [sig_recipient, sig_sender]

۳) در صورتی که مقدار alice_redeems، True باشد، تابع coinExchangeScriptSig1 اجرا می‌شود. در این حالت Alice پول خود در شبکه BlockCypher Testnet را آزاد کرده است و در نتیجه مقدار secret مشخص شده است و Bob می‌تواند پول خود در شبکه Bitcoin Testnet را آزاد کند.

در صورتی که مقدار alice_redeems، True باشد، تابع coinExchangeScriptSig2 اجرا می‌شود. در این حالت فرض شده که امضای فرستنده و گیرنده موجود است و در نتیجه Alice می‌تواند پول خود در شبکه BlockCypher Testnet و Bob می‌تواند پول خود در شبکه Bitcoin Testnet را آزاد کند.

در هر دو حالت هر دو تراکنش با موفقیت Redeem می‌شوند و پیغام زیر نمایش داده می‌شود:

Alice swap tx (BTC) created successfully!

Bob swap tx (BCY) created successfully!

Bob return coins (BCY) tx created successfully!

Alice return coins tx (BTC) created successfully!