نویسنده: ایوب کریمی

ایمیل نویسنده: ayubkarimi587@gmail.com

منتشر شده توسط

کوین ایران

کوین ایران بزرگترین پایگاه خبری فارسی زبان در حوزه فناوری بلاکچین، ارزهای رمزنگاری شده و پلتفرم های مرتبط با بلاکچین است. این وب سایت در سال ۱۳۹۲ توسط بابک جلیلوند و آرش محبوب، با هدف اطلاع رسانی، آموزش و مشاوره به جامعه فارسی زبان علاقه مند به رمز ارزها راه اندازی شد و اولین مقاله آن در ۱۴ دی ماه همان سال، با عنوان «بیت کوین چیست؟» منتشر گردید . هدف کوین ایران از معرفی این فناوری ایجاد محیطی پژوهشی و آموزشی در راستای استفاده صحیح از این فناوری جهت ارائه تسهیلات و رفاه به جوامع فارسی زبان است.

www.coiniran.com

اپلیکیشن های بیشتر

1- کیف پول های پس انداز

فرض کنید آلیس می‌خواهد پول هایش را امن نگه دارد اما نگران گم کردن و یا هک شدن کلید خصوصی خود می-باشد. او اتر را در یک قرارداد با باب به شکل زیر قرار می‌دهد:

- آلیس به تنهایی می‌تواند حداکثر یک درصد از پولها را در روز برداشت کند.

- باب نیز به تنهایی می‌تواند حداکثر یک درصد از پولها را هر روز برداشت کند، اما آلیس می‌تواند با کلیدش تراکنشی ایجاد کند که این توانایی باب را بلوکه کند.

- آلیس و باب با هم می‌توانند هر مبلغی را برداشت کنند.

به طور عادی یک درصد در روز برای آلیس کافی است و اگر بخواهد بیشتر برداشت کند، می‌تواند از باب درخواست کمک کند. اگر کلید آلیس هک شود، او به سوی باب می شتابد تا پولها را به قرارداد جدیدی انتقال دهد. اگر او کلیدش را گم کند، باب در نهایت پول ها را می تواند خارج کند. اگر باب بداندیش به نظر رسد، آلیس می‌تواند توانایی او برای برداشت را مسدود کند.

2- بیمه محصولات کشاورزی

می توان به آسانی یک قرارداد ماخوذات مالی را ایجاد کرد اما با استفاده از داده های آپدیت پذیر هوا به جای هر شاخص قیمت دیگری. اگر یک کشاورز در Lowa یک ماخوذی را بخرد که سود و پرداختی آن در جهت عکس بارندگی در Lowa باشد، پس اگر خشکسالی شود، این کشاورز به طور خودکار پول دریافت می‌کند و اگر باران کافی هم موجود باشد کشاورز خوشحال خواهد بود، زیرا محصولات او عملکرد خوبی خواهند داشت.

3- داده های آپدیت پذیر غیر متمرکز

در مورد قرارداد های مالی متفاوت، غیر متمرکز سازی ممکن است از طریق پروتکلی بنام SchellingCoin امکان پذیر باشد. این پروتکل اساسا به این صورت کار می‌کند: N طرف هر کدام بهای یک داده مشخص شده را (مثلا قیمت اتریوم به دلار) وارد سیستم می‌کنند. این قیمت ها طبقه بندی شده و هر کس بین بیست و پنجمین و هفتاد و پنجمین بر حسب درصدِ یک توکن را به عنوان پاداش دریافت می‌کند.

هر فرد انگیزه دارد که پاسخی را فراهم کند که افراد دیگر فراهم خواهند کرد و تنها بهایی که تعداد زیادی از بازیگران می‌توانند بر آن به طور واقعی توافق کنند همان پیش فرض شفاف یعنی حقیقت می‌باشد. این یک پروتکل غیر متمرکز را ایجاد می‌کند که می‌تواند در تئوری هر تعداد ارزشی را فراهم کند، از جمله قیمت اتریوم به دلار، دما در برلین و یا حتی نتیجه یک محاسبه مشکل خاص.

4- سند شخص ثالث چند امضایی هوشمند

بیت کوین اجازه قرارداد های تراکنش چند امضایی را در جایی که مثلا سه تا از پنج کلید داده شده می‌توانند پولها را خرج کنند، می‌دهد. اتریوم اجازه دانه دانگی بیشتری را می‌دهد، برای مثال چهار از پنج طرف با هم می‌توانند هر چیزی را خرج کنند، سه طرف از پنج طرف با هم می‌توانند روزانه تا ده درصد خرج کنند و دو تا از پنج تا روزانه می‌توانند تا نیم درصد خرج کنند.

علاوه بر آن، فرآیند چند امضایی اتریوم غیر همزمان است. دو طرف می‌توانند امضا هایشان را در بلاکچین در اوقات متفاوت ثبت کنند و آخرین امضا به طور اتوماتیک تراکنش را می‌فرستد.

5- محاسبه ابری

فناوری EVM می‌تواند همچنین برای ایجاد یک محیط محاسبه تایید پذیر به کار رود و به کاربران اجازه دهد که از دیگران در مورد محاسبات بپرسند و سپس به طور اختیاری مدارکی را درخواست کنند برای این که مطمئن شوند که محاسبات در جاهای مشخصی که به طور تصادفی تعیین شده اند، به درستی انجام شده است.

این اجازه ی ایجاد یک بازار محاسبات ابری را می‌دهد که در آنجا هر کاربری می‌تواند با دسکتاپ، لپتاپ و یا سرور مخصوص خود مشارکت کند. تست هایی که گاها انجام می‌شود همراه با سپرده های امنیتی می‌توانند برای اطمینان از این که سیستم قابل اعتماد است، به کار روند (یعنی گره ها نمی‌توانند تقلب سودآوری داشته باشند.)

اگرچه چنین سیستمی ممکن است برای همه ی کارها مناسب نباشد، برای مثال کارهایی که نیاز به سطح بالایی از ارتباطات درون پروسه ای دارند، نمی‌توانند در گره های ابری عظیم به آسانی انجام گیرند. اما کار های دیگری هستند که برای موازی سازی بسیار آسان تر هستند. پروژه هایی مانند SETI@home، folding@home و الگوریتم های ژنتیکی را می‌توان به آسانی در بالای چنین پلتفرمی پیاده کرد.

6- قمار همتا به همتا

هر تعدادی از پروتکل های قمار همتا به همتا مانند Frank Stajano و طاس سایبری Richard Clayton را می‌توان بر روی بلاکچین اتریوم پیاده سازی کرد. ساده ترین پروتکل قمار در واقع قرارداد متفاوتی در زمینه هش بلوک

بعدی است و پروتکل های پیشرفته تر نیز می توانند از آنجا ساخته شوند و سرویس هایی با هزینه نزدیک به صفر ایجاد کنند که توانایی تقلب نداشته باشند.

7- بازار های پیش بینی

در صورت وجود یک پایگاه داده یا SchellingCoin، پیاده سازی بازار های پیش بینی بسیار آسان می‌باشد و بازار های پیش بینی همراه با SchellingCoin احتمالا اولین اپلیکیشن رایج مدیریتی است که می‌توانند به عنوان یک پروتکل مدیریتی برای سازمان های غیر متمرکز به کار روند.

8- بازار های غیر متمرکز روی زنجیره

این بازار ها از سیستم اعتبار و هویت به عنوان مبنا و اساس خود استفاده می‌کنند.

موارد متفرقه و نگرانی ها

پیاده سازی GHOST اصلاح شده

پروتکل GHOST یک پروتکل نوآورانه است که توسط Yonatan Sompolinsky و Aviv Zohar در دسامبر 2013 معرفی شد. انگیزه پشت این پروتکل این است که بلاکچین هایی با زمان تایید سریع، معمولا از پایین آمدن امنیت به دلیل نرخ stale بالا رنج می‌برند زیرا بلوک ها مدت زمان خاصی طول می‌کشد تا در شبکه منتشر شوند؛ اگر استخراج گر A یک بلوک را استخراج کند و سپس استخراج گر B بلوک دیگری را قبل از این که بلوک استخراج گر A به B منتشر شود، استخراج کند، بلوک استخراج گر B به هدر می‌رود و در امنیت شبکه مشارکت نمی‌کند.

علاوه بر این، یک مسئله تمرکز سازی نیز وجود دارد؛ اگر استخراج گر A یک استخر استخراج با 30 درصد قدرت هش داشته باشد و فرد B نیز ده درصد قدرت هش را، A ریسک ایجاد یک بلوک stale را 70 درصد اوقات خواهد داشت (زیرا در 30 درصد باقی مانده A آخرین بلوک را ایجاد کرده و بنابراین فورا داده های استخراج را دریافت می‌کند) در حالی که B نود درصد اوقات این ریسک را خواهد داشت.

در نتیجه اگر فاصله بلوک به اندازه کافی کوتاه باشد تا درجه stale بالا رود، A در اساس با توجه به خاصیت اندازه اش کارا تر خواهد بود. اگر این دو تاثیر را در نظر بگیریم، بلاکچین هایی که سریعا بلوک تولید می‌کنند، احتمال زیادی دارد که تبدیل به یک استخر استخراج شوند که درصد عظیم مکفی از قدرت هش شبکه را برای کنترل واقعی بر فرآیند استخراج داشته باشند.

آنچنان که توسط Sompolinsky و Zohar توصیف شده، GHOST مسئله اول یعنی فقدان امنیت شبکه را حل می-کند و این را با شامل کردن بلوک های stale در محاسبه اینکه کدام زنجیره طولانی ترین است، حل می‌کند. می-توان گفت که نه فقط پدر و اجداد دیگر یک بلوک بلکه همچنین بچه های stale ایجاد بلوک (در زبان خاص اتریوم عمو ها به کار می‌رود) به محاسبات طولانی ترین بلوک اضافه می‌شوند که این طولانی ترین بلوک، گواه کار آن را پشتیبانی می‌کند.

برای حل مساله دوم که همان تعصب مرکزیت می‌باشد، ما به فراسوی پروتکل توصیف شده توسط دو فرد مذکور می‌رویم و همچنین به stale ها اجازه می‌دهیم که در زنجیره اصلی ثبت شوند تا پاداش بلوک دریافت کنند. یک بلوک stale 93.75 درصد از پاداش مبنا را دریافت و برادر زاده ای که شامل بلوک stale است، 6.25 درصد باقی مانده را دریافت می‌کند، اما هزینه تراکنش به عمو ها داده نمی‌شود.

اتریوم نسخه ساده شده ای از GHOST را پیاده می‌کند که فقط تا پنج رده پایین می‌رود. به طور خاص، یک بلوک stale فقط می‌تواند مشمول یک عمو توسط نسل دوم تا پنجم بچه پدرش باشد و نه بلوک دیگری با رابطه فاصله ای بیشتر (مانند بچه نسل ششم پدر یا بچه نسل سوم پدر بزرگ). این امر به دلایل زیادی انجام می‌شود.

اولا GHOST نامحدود مشمول پیچیدگی های زیادی برای محاسبه این که کدام عمو ها برای بلوک خاصی معتبرند، می‌شود. دوما GHOST نامحدود همراه با قابلیت جبران و پاداش که در اتریوم به کار می‌رود از استخراج گر سلب انگیزه می‌کند که در زنجیره اصلی به استخراج بپردازد و در زنجیره مهاجم عمومی استخراج نکند.

سرانجام محاسبات نشان می‌دهد که GHOST پنج سطحی همراه با انگیزه بخشی، بیش از 95 درصد کارایی دارد و حتی با زمان بلوک 15 ثانیه ای و استخراج گرانی که قدرت هش 25 درصدی دارند، نشان داده شده که کمتر از 3 درصد از منفعت تمرکز برخوردار خواهند بود.

هزینه ها

از آنجا که هر تراکنشی که داخل بلاکچین انتشار می‌یابد، هزینه ی دانلود و تایید را به شبکه تحمیل می‌کند، تا حدودی به یک مکانیزم قانون گذاری نیاز است که معمولا مشمول هزینه های تراکنش است تا از سوء استفاده جلوگیری کند. روش پیش فرضی که در بیت کوین به کار رفته، داشتن هزینه هایی کاملا داوطلبانه است که متکی بر استخراج کنندگان است که به عنوان پاسدار عمل کنند و حداقل های پویایی را تنظیم نمایند.

این روش، مخصوصا در جامعه بیت کوین با محبوبیت زیادی مواجه شده است زیرا رویکردی مبتنی بر بازار است و به عرضه و تقاضا بین استخراج کنندگان و ارسال کنندگان تراکنش ها مجوز می‌دهد و سبب تعیین قیمت می‌شود.

مشکل این شیوه از استدلال این است که پردازش تراکنش را نمی‌توان بازار نامید، اگرچه ذاتا جذاب است که پردازش تراکنش را به عنوان یک سرویسی تفسیر کرد که استخراج گر به ارسال کننده ارائه می‌دهد.

در واقع هر تراکنشی که یک استخراج گر درگیرش می‌شود، لازم است که توسط هر گره در شبکه پردازش شود؛ بنابراین بخش عظیمی از هزینه پردازش تراکنش را طرف های ثالث تقبل می‌کنند، نه استخراج گری که تصمیم می‌گیرد که آیا درگیر آن تراکنش شود یا نه. در نتیجه احتمال زیادی وجود دارد که مشکلات مربوط به تراژدی انبازه ها روی دهد.

اما چون این عیب در مکانیزم مبتنی بر بازار خودش را ظاهر کرده، وقتی فرض ساده شده غیر دقیق خاصی داده می‌شود، به طور اسرارآمیزی خودش را باطل می‌کند. فرض کنید که:

1- یک تراکنش به عملکرد های K منتهی می‌شود و پاداش KR را به هر استخراج گری که درگیر آن شود، می‌دهد. در اینجا R توسط ارسال کننده تنظیم می‌شود و K و R تقریبا از پیش برای استخراج گر قابل مشاهده خواهد بود.

2- یک عملکرد هزینه پردازش C را برای هر گره به همراه دارد (یعنی همه گره ها کارایی مساوی دارند.)

3- تعداد N گره استخراج وجود دارد که هر کدام قدرت پردازش دقیقا مساوی دارند (یعنی 1/N از کل).

4- هیچ گره کامل غیر استخراج گری موجود نیست.

یک استخراج گر، مشتاق به پردازش یک تراکنش خواهد بود اگر پاداش مورد انتظار از هزینه بیشتر باشد. بنابراین پاداش مورد انتظار KR/N است زیرا که استخراج گر 1/N شانس پردازش بلوک بعدی را دارد و هزینه پردازش برای استخراج گر در واقع kC است.

بنابراین استخراج گران درگیر تراکنش هایی می‌شوند که kR/N > kC, or R > NC باشد. توجه داشته باشید که R هزینه قبل از عملکرد است که توسط ارسال کننده فراهم می‌شود و بنابراین در حد پایین تر از سودی است که ارسال کننده از تراکنش استخراج می‌کند. NC هزینه کل شبکه در پردازش عملکرد است. پس استخراج گران انگیزه دارند که فقط درگیر تراکنش هایی شوند که سود عملی آن در کل بیشتر از هزینه باشد.

اما در واقعیت، انحرافات مهم زیادی از این فرضیات وجود دارد:

1- استخراج گر هزینه بالاتری را از دیگر گره های تایید کننده، برای پردازش تراکنش می‌پردازد. از آنجا که این زمان اضافی برای تایید در انتشار بلوک تاخیر ایجاد می‌کند، شانس این که بلوک خراب شود افزایش می‌یابد.

2- در حقیقت گره های کاملی موجودند که غیر استخراج گر هستند.

3- توزیع قدرت استخراج ممکن است نهایتا و در عمل بسیار غیر انسانی شود.

4- نگه دارندگان، دشمنان سیاسی و بداندیشان که عملکرد مفید آنها آسیب رسانی به شبکه است، می‌توانند به طور هوشمندانه ای قرارداد هایی را تنظیم کنند که هزینه اش بسیار پایین تر از هزینه پرداختی توسط دیگر گره های تایید کننده باشد.

نکته 1 در بالا، گرایشی را برای استخراج گر فراهم می‌کند که درگیر تراکنش های کمتری شود و نکته دوم NC را افزایش می‌دهد. لذا این دو تاثیر، حداقل به طور نسبی یکدیگر را خنثی می‌کنند. نکات 3 و 4 مسائل مهمی هستند و برای حل آنها، ما به طور ساده یک پوشش شناور ایجاد می‌کنیم و آن بدین صورت است که هیچ بلوکی نمی‌تواند عملکرد های بیشتری از BLK_LIMIT_FACTOR دفعه میانگین متحرک نمایی درازمدت داشته باشد. به طور خاص:

blk.oplimit = floor((blk.parent.oplimit * (EMAFACTOR - 1) + floor(parent.opcount * BLK_LIMIT_FACTOR))/ EMA_FACTOR)

BLK_LIMIT_FACTOR و EMA_FACTOR ثابت هستند که به مقدار 65536 و 1.5 تنظیم می‌شوند، البته در حال حاضر این گونه است، اما بعد از تحلیل و بررسی بیشتر احتمال دارد که تغییر کنند.

محاسبه و تورینگ کامل

یک نکته مهم این است که ماشین مجازی اتریوم دارای تورینگ کامل است، بدین معنی که کد EVM می‌تواند هر محاسبه ای را که اجرای آن امکان پذیر باشد، از جمله حلقه های نامحدود، رمزدار کند. کد EVM به دو شیوه اجازه حلقه زنی می‌دهد؛ اول با دستورالعمل JUMP که به برنامه اجازه می‌دهد که به نقطه قبلی در کد برگردد و یک دستورالعمل JUMPI که خیزش های مشروط را انجام می‌دهد و اجازه عملیاتی مانند: x = x * 2 :x < 27 را می‌دهد.

دوم این که قرارداد ها می‌توانند با قرارداد های دیگر تماس داشته و به طور بالقوه اجازه حلقه زنی را از طریق بازگشت داشته باشند. این به طور طبیعی منجر به مشکلی می‌شود و آن این که آیا کاربران بداندیش اساسا می‌توانند استخراج گران و گره های کامل را به زور وارد یک حلقه نامحدود کنند؟

این موضوع به دلیل یک نوع مشکل که در علم کامپیوتر موجود است، مطرح می‌شود که این مشکل مکث (halting) نامیده می‌شود. به طور عمومی هیچ شیوه ای وجود ندارد که تعیین کنیم آیا یک مشکل پیش آمده هرگز مکث می‌کند یا نه.

همچنان که در بخش انتقال فازی توصیف شد، راه حل ما به شیوه ای کار می‌کند که یک تراکنش را ملزم کند که بیشترین تعداد مراحل محاسباتی را تا جایی که اجازه هست، تنظیم کند. اگر اجرا زمان بیشتری طول بکشد، محاسبات برگشت داده شده اما با این وجود هزینه ها باز هم پرداخت می‌شوند. پیام ها نیز به همین شیوه عمل می‌کنند. برای مشخص شدن انگیزه ای که پشت راه حل ما هست، مثال های زیر را مورد توجه قرار دهید:

- یک مهاجم قراردادی را ایجاد می‌کند که یک حلقه نامحدود را به اجرا وا می‌دارد و سپس تراکنشی را ارسال می‌کند که حلقه را برای استخراج گر فعال کند. استخراج گر تراکنش را پردازش خواهد کرد و حلقه نامحدود را اجرا می‌کند و منتظر می‌شود تا زمانی که gas تمام شود.

حتی اگر تراکنش gas آن تمام شده و نیمه کاره متوقف شود، تراکنش هنوز هم معتبر است و استخراج گر هنوز هم هزینه را برای هر مرحله محاسباتی از مهاجم مطالبه می‌کند.

- یک مهاجم، حلقه نامحدود خیلی طویلی را ایجاد می‌کند و قصدش این است که استخراج گر را مجبور کند برای مدت طولانی به محاسبه ادامه دهد و در زمانی که محاسبه تمام شود چند بلوک دیگر هم ظاهر شود و در این حالت برای استخراج گر مقدور نخواهد بود که درگیر تراکنش شده و هزینه مطالبه نماید.

اما مهاجم موظف است که بهایی برای STARTGAS تسلیم کند تا تعداد مراحل محاسباتی محدود شود، بنابراین استخراج گر پیشاپیش می‌داند که محاسبه ی مراحلِ بیش از حد معمول را خواهد داشت.

- یک مهاجم قراردادی با کدی مانند send(A,contract.storage[A]);0 = contract.storage[A]را می‌بیند و تراکنشی با gas کافی برای اجرای مرحله اول، و نه فعلا برای مرحله دوم، ارسال می‌کند (یعنی برداشت می کند اما اجازه نمی‌دهد موجودی کم شود). نویسنده ی قرارداد لازم نیست که در مورد محافظت در برابر چنین حملاتی نگران باشد زیرا اگر اجرا نیمه کاره متوقف شود، از طریق تغییراتی برگشت پذیر خواهد بود.

- یک قرارداد مالی با میانگین گیری از نه دسته داده های آپدیت پذیر اختصاصی به منظور به حداقل رساندن ریسک، کار می‌کند. مهاجم یک دسته از داده های آپدیت پذیر را تصاحب می‌کند که این طور طراحی شده اند که از طریق مکانیزم اعلان آدرس متغیر که در بخش DAO ها توصیف شد، قابل اصلاح و تعدیل باشند.

مهاجم آنها را تغییر می‌دهد و سبب به راه اندازی یک حلقه نامحدود می‌شود؛ به موجب آن مهاجم هر تلاشی برای مطالبه وجه از جانب قرارداد مالی را به تمام شدن gas منجر می‌کند. اما قرارداد مالی می‌تواند یک محدوده gas را در پیام مشخص کند تا از این مشکل ممانعت به عمل آورد.

گزینه دیگر در رابطه با تورینگ کامل، تورینگ ناقص است که در آنجا JUMP و JUMPI موجود نیستند و فقط یک کپی از یک قرارداد، مجاز است که در دسته اعلان ها در هر زمان خاصی موجود باشد. با این سیستم می‌توان سیستم هزینه را توصیف کرد و شک و شبهه هایی که پیرامون کارایی راه حل ما وجود دارد ممکن است ضروری نباشد، زیرا هزینه اجرای یک قرارداد می تواند از بالا توسط اندازه اش محدود شود.

علاوه بر این، تورینگ ناقص محدودیت خیلی بزرگی هم نیست و ما در این زمینه قرارداد های زیادی را مورد بررسی قرار داده ایم؛ تا کنون فقط یکی از آنها نیازمند حلقه بوده است و حتی آن حلقه هم می‌توانست با 26 تکرار یک قسمت یک خطی از کد، برداشته شود.

با دادن اشاراتی جدی مبنی بر تورینگ کامل و منفعت محدود، این سوال مطرح می‌شود که چرا باید زبان تورینگ ناقص نباشد؟ اما در واقع زبان تورینگ ناقص از راه حلی مناسب برای این مشکل به دور است. برای متوجه شدن قضیه، قرارداد های زیر را مورد بررسی قرار دهید:

;(C0: call(C1); call(C1

;(C1: call(C2); call(C2

;(C2: call(C3); call(C3

...

;(C49: call(C50); call(C50

C50: (run one step of a program and record the change in storage)

حالا یک تراکنش را به A بفرستید. بنابراین در 51 تراکنش، ما قراردادی داریم که تا 2^{50} مراحل محاسباتی ادامه می‌یابد. استخراج گران می‌توانند چنین خرابی هایی را قبل از موعد تشخیص دهند و با باقی گذاردن یک بها همراه با هر قرارداد، تعداد بیشینه مراحل محاسباتی را تعیین کنند و این را برای قرارداد ها محاسبه کنند تا مکرر با قرارداد های دیگر ارتباط داشته باشند.

اما این امر نیازمند این است که استخراج گران قرارداد هایی را که منجر به ایجاد دیگر قرارداد ها می‌شوند، ممنوع کنند (از آنجا که ایجاد و اجرای همه 50 قرارداد بالا می‌تواند به سادگی در داخل یک قرارداد مجزا قرار گیرد). یک نکته مشکل آفرین دیگر این است که فیلد آدرس یک پیام متغیر است و بنابراین در کل ممکن است غیر ممکن باشد که پیشاپیش بگوییم کدام قرارداد های دیگر مورد اعلان و ارتباط قرار می‌گیرند.

روی هم رفته ما به یک نتیجه ی جالب می‌رسیم: مدیریت تورینگ کامل بسیار آسان است و تورینگ ناقص برای مدیریت بسیار مشکل است، مگر این که کنترل های مشابه دقیقی انجام شود؛ اما در آن صورت چرا اجازه ندهیم که پروتکل تورینگ کامل باشد؟

واحد پول و صدور

شبکه اتریوم شامل واحد پولی است که در داخل خودش ساخته شده و اتر نام دارد. اتر مقصود دوگانه ای را برآورد می‌کند، اول یک لایه نقدینگی اولیه را برای اجازه مبادله مفید بین انواع مختلف دارایی های دیجیتال فراهم می‌کند. دومی که بسیار مهم تر از مورد اول می‌باشد، مکانیزمی را برای پرداخت هزینه تراکنش ها فراهم می‌کند.

برای سادگی و اجتناب از بحث بیشتر، (بحث اخیر در مورد mBTC, uBTC, satoshi را در بیت کوین مورد توجه قرار دهید) پولها از قبل برچسب زده شده و تعیین می‌شوند.

- wei :1
- szabo :12^10
- finney :15^10
- ether :18^10

این باید به عنوان یک نسخه توسعه یافته از دلار و سنت یا بیت کوین و ساتوشی در نظر گرفته شود. در آینده ی نزدیک ما انتظار داریم که اتر برای تراکنش های معمولی، finney برای تراکنش های خرد و szabo و wei برای بحث های فنی در مورد هزینه ها و اجرای پروتکل به کار روند.

مدل صدور به صورت زیر خواهد بود:

- اتر در یک فروش واحد پول به قیمت 1337-2000 اتر در قبال هر بیت کوین، رها خواهد شد و هدف این مکانیزم این است که هزینه های سازمان اتریوم را تامین کند و برای توسعه ای که با موفقیت توسط تعدادی از پلتفرم های رمزنگاری دیگر به کار رفته، هزینه کند.

خریدار های اولیه از تخفیفات بزرگ تری بهره مند می‌شوند. بیت کوینی که از فروش دریافت می‌شود کلا برای پرداخت حقوق ها و بخشش ها به توسعه دهندگان، محققان و پروژه هایی در اکوسیستم رمزارز به کار می‌رود.

- 0.099 از کل مقدار به فروش رفته برای اولین کسانی که در توسعه قبل از دریافت و تامین هزینه از طریق بیت کوین و قبل از به قطعیت رسیدن این تامین هزینه از طریق بیت کوین، مشارکت کردند، اختصاص داده شده و یک 0.099 دیگر هم به پروژه های تحقیقاتی دراز مدت تخصیص داده می‌شود.

- 0.26 از کل مقدار فروخته شده، بعد از آن نقطه تا ابد، هر سال برای استخراج کنندگان خواهد بود.

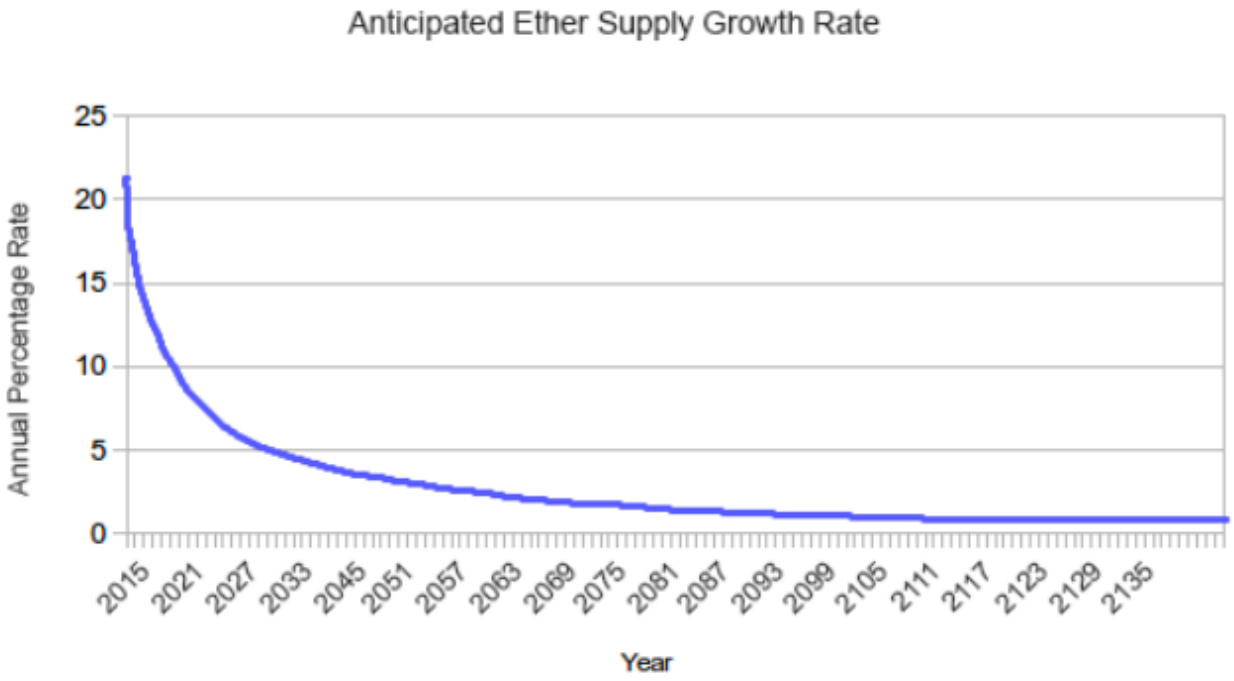
تفکیک صدور

مدل رشد عرضه خطی دائمی، ریسک آنچه که تعدادی به عنوان تمرکز زیاد ثروت در بیت کوین می‌بینند، کاهش پیدا می‌کند و به افرادی که در زمان حال و در دوره های آینده زندگی خواهند کرد، فرصت عادلانه ای داده می‌شود تا واحد های پول را کسب کنند. در این حال همزمان مشکل کاهش بهای اتر هم حل می‌شود زیرا نرخ رشد عرضه اگر به عنوان درصدی در نظر گرفته شود، به مرور زمان تمایل دارد که به سمت صفر سیر کند.

ما به این تئوری می‌رسیم که سکه ها همیشه در طول زمان به دلیل بی دقتی، مرگ و غیره گم می‌شوند و این گم شدگی سکه می‌تواند به صورت درصدی از کل عرضه هر سال، مدل بندی شود. کل عرضه واحد پول در گردش، در واقع سرانجام در ارزشی مساوی با صدور سالیانه تقسیم بر نرخ گم شدگی قرار خواهد گرفت (مثلا در نرخ گم شدگی یک درصد، زمانی که عرضه به 26x برسد، 0.26x استخراج می‌شود و 0.26x هر سال گم می‌شود و این سبب ایجاد یک موازنه می‌گردد).

Group	At launch	After 1 year	After 5 years
Currency units	1.198X	1.458X	2.498X
Purchasers	83.5%	68.6%	40.0%
Early contributor distribution	8.26%	6.79%	3.96%
Long-term endowment	8.26%	6.79%	3.96%
Miners	0%	17.8%	52.0%

با وجود صدور خطی واحد پول، درست مانند بیت کوین به مرور زمان نرخ رشد عرضه تمایل دارد به سمت صفر میل کند.



تمرکز استخراج

الگوریتم استخراج بیت کوین اساسا با استخراج گرانی کار می‌کند که SHA256 را در نسخه های نسبتا اصلاح شده بلوک پیشرو بارها و بارها محاسبه می‌کنند تا سرانجام یک گره به صورت نسخه ای ظاهر شود که هش آن کمتر از هدف باشد (در حال حاضر حدود 2^{190}).

اما این الگوریتم استخراج نسبت به دو شیوه تمرکز آسیب پذیر است. اول این که اکوسیستم استخراج توسط تراشه هایی کامپیوتری که برای این کار طراحی شده اند مغلوب ASIC ها می شود، که هزاران بار کارا تر از استخراج بیت کوین هستند. این بدین معنی است که استخراج بیت کوین هرگز یک پیشه غیر متمرکز و مساوات طلب نیست و نیازمند میلیون ها دلار سرمایه برای مشارکت موثر است.

دوما اکثر استخراج گران بیت کوین در واقع اعتبار بخشی بلوک را به طور محلی انجام نمی‌دهند، در عوض آنها متکی بر استخر های استخراج متمرکز هستند تا پیشرو های بلوک فراهم شوند. این مشکل به طور قابل بحثی بدتر است، چنان که در زمان نوشتن این نوشتار، دو استخر استخراج برجسته تقریبا به طور غیر مستقیم 50 درصد پردازش قدرت در شبکه بیت کوین را کنترل می‌کنند. این حقیقت که استخراج گران اگر یک استخر یا ائتلاف اقدام به حمله 51 درصد کند، می‌توانند به استخر های استخراج دیگری منتقل شوند تا حدودی مسئله را تسکین می-بخشد.

هدف اتریوم در حال حاضر این است که از یک الگوریتم استخراج استفاده کند که مبتنی بر تولید تصادفی یک عملکرد هش بی نظیر برای هر 1000 نانس باشد که از یک دامنه گسترده کافی از محاسبات برای از میان برداشتن منافع سخت افزار خاص استفاده می‌کند.

چنین استراتژیی یقینا تمرکز را به صفر نمی‌رساند و نیاز هم نیست که چنین کند. توجه داشته باشید که هر کاربر فردی، بر روی دسکتاپ و یا لپ تاپ خصوصی خود، می‌تواند مقدار معینی فرآیند استخراج را تقریبا به طور رایگان انجام دهد و فقط هزینه برق را بپردازد. اما بعد از رسیدن به استفاده 100 درصد از سی پی یو کامپیوتر، استخراج اضافی مستلزم پرداخت هزینه برای برق و سخت افزار می‌باشد.

شرکت های استخراج ASIC لازم است که برای پرداخت الکتریسیته و سخت افزار از هش اول اقدام کنند. بنابراین اگر بتوان تمرکز را در زیر نسبت (E+H)/E نگه داشت، حتی اگر ASIC ها هم ساخته شوند، هنوز هم جا برای استخراج گران معمولی خواهد بود.

علاوه بر این با طراحی یک الگوریتم که استخراج نیازمند دسترسی به کل بلاکچین باشد، استخراج کنندگان مجبور می‌شوند که کل بلاکچین را ذخیره کنند و حداقل قادر به تایید هر تراکنش باشند. این مطلب نیاز به استخر های استخراج متمرکز را از میان بر می‌دارد. اگرچه تعدادی از استخر های استخراج هنوز نقش قانونی ایجاد توازن تصادفی بودن توزیع پاداش را ایفا می‌کنند. این عملکرد می‌تواند توسط استخر های همتا به همتا بدون کنترل مرکزی اعمال شود.

این شیوه به جنگ علیه تمرکز نیز کمک می‌کند که این کار با افزایش تعداد گره های کامل در شبکه انجام می‌شود؛ لذا شبکه به طور معقولی غیر متمرکز می‌ماند حتی اگر تعداد زیادی از کاربران معمولی مشتری های سبک را ترجیح دهند.

مقیاس پذیری

یک نگرانی رایج در مورد اتریوم مسئله مقیاس پذیری آن است. اتریوم نیز مانند بیت کوین از این نقص که هر تراکنش لازم است توسط هر گره در شبکه پردازش شود، رنج می‌برد. در مورد بیت کوین، اندازه بلاکچین رایج در حدود 20GB قرار گرفته و هر ساعت حدود 1MB رشد می‌کند.

اگر شبکه بیت کوین مجبور بود مانند ویزا (Visa) در هر ثانیه 2000 تراکنش را پردازش کند، در هر 3 ثانیه 1MB رشد می‌کرد (هر ساعت 1GB و سالی 8TB). اتریوم احتمالا از الگوی رشد مشابهی رنج خواهد برد و این مسئله با توجه به این حقیقت که اپلیکیشن های زیادی در بالای بلاکچین اتریوم موجود خواهد بود، وخیم تر نیز می شود و مانند بیت کوین نیست که فقط یک واحد پول در بالای بلاکچین آن موجود باشد.

اما این قضیه که گره های کامل اتریوم لازم است فقط مرحله را ذخیره کرده و لزومی به ذخیره کل تاریخچه بلاکچین ندارد، سبب بهبود اوضاع می‌شود.

مشکل چنین اندازه بزرگی از بلاکچین، ریسک تمرکز است. اگر اندازه بلاکچین مثلا به اندازه 100 TB افزایش پیدا کند، سپس سناریوی محتمل این خواهد بود که فقط تعداد کمی از کسب و کار های بزرگ، گره های کامل را اجرا و مدیریت می‌کنند، با وجود همه کاربر های عادی که از گره های سبک SPV استفاده می‌کنند.

در چنین موقعیتی، این نگرانی بالقوه مطرح می‌شود که گره های کامل می‌توانند با یکدیگر جمع شوند و همگی توافق کنند که به شیوه ای سودآور تقلب کنند (مثلا با تغییر پاداش بلوک به خودشان بیت کوین بدهند). گره های سبک به هیچ وجه فوری متوجه این موضوع نمی‌شوند، البته حداقل یک گره کامل صادق احتمالا موجود خواهد بود و بعد از گذشت چند ساعتی، اطلاعات در مورد کلاه برداری از طریق کانال هایی مانند Reddit به بیرون درز پیدا می‌کند اما در آن هنگام خیلی دیر خواهد بود.

در این حالت بستگی به کاربران عادی دارد که تلاش کنند بلوک های داده شده را در لیست سیاه قرار دهند. این مشکل نیازمند هماهنگی عظیم و احتمالا غیر ممکن است و در مقیاسی شبیه به حمله 51 درصدی موفقیت آمیز است. در مورد بیت کوین این مشکل اخیرا مورد توجه قرار گرفته است، اما یک فرایند اصلاح بلاکچین وجود دارد که توسط Peter Todd پیشنهاد شده و این مشکل را تسکین می‌بخشد.

اتریوم در کوتاه مدت دو استراتژی اضافی را برای دسته و پنجه نرم کردن با این مشکل، به کار خواهد برد. اولا به دلیل الگوریتم های استخراج مبتنی بر بلاکچین، حداقل هر استخراج گر مجبور خواهد بود که یک گره کامل باشد و محدوده پایین تری را در تعداد گره های کامل ایجاد نماید. دوما که مهمتر از اولی نیز می‌باشد ما یک ریشه درخت فازی واسطه را بعد از پردازش هر تراکنش، در بلاکچین درگیر خواهیم کرد.

حتی اگر اعتبار دهی بلوک متمرکز باشد، به شرطی که گره تایید کننده صادق موجود باشد، مشکل تمرکز می‌تواند از طریق یک پروتکل تایید کنار زده شود. اگر یک استخراج گر بلوک نامعتبری را منتشر کند، ان بلوک باید یا به بدی شکل گرفته باشد و یا حالت S[n] نادرست باشد.

از آنجا که S[0] به عنوان صحیح شناخته شده، باید تعدادی S[i] مرحله اول موجود باشد که نادرست هستند که در آنجا S[i-1] صحیح است. گره تایید کننده شاخص i را فراهم می‌کند، همراه با "یک مدرک عدم اعتماد" که متشکل از زیر مجموعه ای از گره های درخت Patricia می باشد که لازم است APPLY(S[i-1],TX[i]) -> S[i] را پردازش کنند. گره ها قادر خواهند بود که از آن گره ها برای مدیریت آن بخش از محاسبه استفاده کنند و ببینند که S[i] تولید شده با S[i] فراهم شده مطابقت ندارد.

یک حمله پیچیده تر دیگر شامل استخراج گران بداندیشی است که بلوک های ناکامل را منتشر می‌کنند. بنابراین اطلاعات کامل حتی موجود نیست تا با توجه به آن تعیین کرد که آیا بلوک ها معتبرند یا نه. راه حل این مشکل یک پروتکل چالش-جواب است؛ به این ترتیب که گره های تاییدی چالش هایی را صادر می‌کنند که به شکل شاخص های تراکنش هدف می‌باشد و به محض این که یک گره آنها را دریافت می‌کند، یک گره سبک با بلوک به عنوان غیر قابل اعتماد برخورد می‌کند تا یک گره دیگر چه استخراج گر باشد و چه تاییدگر، زیر مجموعه ای از گره های Patricia را به عنوان مدرک اعتبار فراهم کند.

کنار هم گذاشتن تمام موارد: اپلیکیشن های غیر متمرکز

مکانیزم قراردادی که در بالا توصیف شد، به همه اجازه می‌دهد که چیزی را که اساسا یک اپلیکیشن رسته دستور است، بر روی ماشین مجازی مدیریت کند که توسط اجماع بین کل شبکه اجرا می‌شود و به آن اجازه می‌دهد که یک فاز قابل دسترسی جهانی را به عنوان هارد درایو خود تعدیل کند.

اما برای اکثر مردم، واسطه رسته فرمان که یک مکانیزم ارسال تراکنش است، به اندازه کافی خوشایند نیست که عدم تمرکز را به یک جایگزین رایج جذاب، تبدیل کند. برای این مقصود، یک اپلیکیشن غیر متمرکز کامل باید متشکل از هر دو اجزای سطح پایین و با منطق کسب و کاری باشد، چه به طور کامل روی اتریوم پیاده سازی شده باشند، یا از ترکیبی از اتریوم و سیستم های دیگر استفاده کنند (مانند لایه پیام رسانی همتا به همتا که اخیرا برنامه ریزی شده که برای کلاینت اتریوم فراهم گردد)، و یا به طور کلی سیستم های دیگری باشند و با عناصر واسطه کاربری گرافیکی سطح بالا.

طرح کلاینت اتریوم این است که به عنوان یک مرورگر شبکه خدمت کند اما eth شیء API جاوا اسکریپت را پشتیبانی کند که صفحات وب تخصصی شده ای که در کلاینت دیده می‌شوند، قادر خواهند بود که با بلاکچین اتریوم تعامل کنند. از منظر وب سنتی، این وب پیج ها کاملا محتوای ساکنی دارند زیرا که بلاکچین و دیگر پروتکل های غیر متمرکز به عنوان جایگزینی کامل برای سرور و به مقصود مدیریت درخواست هایی که توسط کاربر شروع شده، خدمت می‌کنند. سرانجام پروتکل های غیر متمرکز، خوشبختانه خودشان به نوعی از اتریوم استفاده می‌کنند و ممکن است خودشان نیز برای ذخیره وب پیج ها به کار روند.

نتیجه گیری

پروتکل اتریوم در ابتدا به عنوان یک نسخه پیشرفته ی رمزارز دیده شد که ویژگی های مترقی ای مانند سند شخص ثالث روی بلاکچین، محدوده های برداشت و قرارداد های مالی، بازار قمار و چیز هایی مانند آن را از طریق یک زبان برنامه نویسی بسیار تعمیم یافته، فراهم می‌کند.

پروتکل اتریوم هیچ یک از اپلیکیشن ها را به طور مستقیم پشتیبانی نمی‌کند اما وجود یک زبان برنامه نویسی کامل تورینگ به این معنی است که قرارداد های اختیاری می‌توانند در تئوری برای هر نوع تراکنش و یا اپلیکیشنی ایجاد شوند. اما آنچه که بیشتر در مورد اتریوم جالب است این است که پروتکل اتریوم بسیار فراتر از فقط واحد پول بودن می‌رود.

پروتکل ها و اپلیکیشن های غیر متمرکز در مورد انبارش فایل غیر متمرکز، محاسبات غیر متمرکز و بازار های پیش بینی غیر متمرکز، همراه با تعداد زیادی از مفاهیم دیگر این پتانسیل را دارند که به طور اساسی کارایی صنعت محاسبه را افزایش دهند و تقویت عظیمی برای دیگر پروتکل های همتا به همتا را فراهم کنند که این کار را با اضافه کردن یک لایه اقتصادی برای اولین بار انجام می‌دهند. سرانجام باید گفت که یک رسته اساسی از اپلیکیشن هایی که اصلا در زمینه پول چیزی برای انجام دادن ندارند، موجود است.

مفهوم عملکرد انتقالی فازی اختیاری آنچنان که توسط پروتکل اتریوم پیاده شده برای یک پلتفرم با پتانسیل بی نظیر فراهم می‌گردد و یک پروتکل بن بست و تک منظوره نیست که برای رسته خاصی از اپلیکیشن ها در انبارش داده، قمار یا امور مالی فراهم شده باشد بلکه طراحی اتریوم بی انتها است و ما معتقدیم که بسیار مناسب خدمت به عنوان یک لایه زیربنایی برای تعداد زیادی از پروتکل های هم مالی و هم غیر مالی در سال های آینده است.

یادداشت ها و خواندن بیشتر

یادداشت ها

1- یک خواننده سطح بالا ممکن است متوجه شود که در حقیقت آدرس بیت کوین هش کلید عمومی منحنی بیضوی است و خودش کلید عمومی نیست. اما در واقع یک فناوری رمزنگاری قانونی کامل برای مراجعه به هش کلید عمومی به عنوان یک کلید عمومی است. این بدین دلیل است که رمزنگاری بیت کوین می‌تواند به عنوان یک الگوریتم امضای دیجیتال سفارشی تلقی گردد که در آنجا کلید عمومی متشکل از هش کلید عمومی ECC می‌باشد.

امضا متشکل از یک کلید عمومی ECC است که با امضای ECC الحاق شده و الگوریتم تایید شامل بررسی کلید عمومی ECC در امضا در برابر هش کلید عمومی ECC است که به عنوان یک کلید عمومی فراهم شده و سپس امضای ECC را در برابر کلید عمومی ECC تایید می‌کند.

2- از لحاظ فنی میانگین یازده بلوک قبل

3- ذاتا 2 و CHARLIE هر دو عدد هستند که دومی ارائه 256 بر مبنای بیگ اندیان است. عدد ها می‌توانند حداقل 0 و حداکثر 2^256-1 باشند.

خواندن بیشتر

.1

Intrinsic value: https://tinyurl.com/BitcoinMag-IntrinsicValue

.2

Smart property: https://en.bitcoin.it/wiki/Smart_Property

.3

Smart contracts: https://en.bitcoin.it/wiki/Contracts

.4

B-money: http://www.weidai.com/bmoney.txt

.5

/Reusable proofs of work: http://www.finney.org/~hal/rpow

.6

Secure property titles with owner authority: http://szabo.best.vwh.net/securetitle.html

.7

Bitcoin whitepaper: http://bitcoin.org/bitcoin.pdf

.8

/Namecoin: https://namecoin.org

.9

Zooko's triangle: http://en.wikipedia.org/wiki/Zooko's_triangle

.10

Colored coins whitepaper: https://tinyurl.com/coloredcoin-whitepaper

.11

Mastercoin whitepaper: https://github.com/mastercoin-MSC/spec

.12

Decentralized autonomous corporations, Bitcoin Magazine: https://tinyurl.com/Bootstrapping-DACs

.13

Simplified payment verification:https://en.bitcoin.it/wiki/Scalability#Simplifiedpaymentverification

.14

Merkle trees: http://en.wikipedia.org/wiki/Merkle_tree

.15

Patricia trees: http://en.wikipedia.org/wiki/Patricia_tree

.16

GHOST: http://www.cs.huji.ac.il/~avivz/pubs/13/btc_scalability_full.pdf

.17

StorJ and Autonomous Agents, Jeff Garzik: https://tinyurl.com/storj-agents

.18

Mike Hearn on Smart Property at Turing Festival: http://www.youtube.com/watch?v=Pu4PAMFPo5Y

.19

Ethereum RLP: https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-RLP

.20

Ethereum Merkle Patricia trees: https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-Patricia-Tree

.21

Peter Todd on Merkle sum /trees:http://sourceforge.net/p/bitcoin/mailman/message/31709140