دانشکده مهندسی کامپیوتر

پایان نامه دوره کارشناسی مهندسی کامپیوتر-نرم افزار

پیاده سازی و ایجاد یک ارز رمز جدید مبتنی بر بلاک چین

نگارش:

حامد محمدی

استاد راهنما:

دکتر سعید صدیقیان کاشی

پاییز ۱۳۹۷

چکیده

در دنیای امروز باتوجه به پیشرفته های انجام شده در زمینه تکنولوژی و همچنین نیاز به انجام پرداخت های سریع و مطمئن به صورت اتوماتیک توسط عامل های کامپیوتری و حذف واسطه ها از پرداخت به نظر می‌رسد مدل های پرداختی فعلی دنیا قادر به پاسخگویی به نیاز های این زمینه نباشند. از این رو تکنولوژی بلاک چین و ارز رمز ها به عنوان پاسخی بر نیاز به پرداخت های سریع و امن و بدون واسطه مورد توجه قرار گرفته است. اولین ارز رمز دنیا با نام بیت کوین در سال ۲۰۰۹ توسط فرد یا افرادی ناشناس با نام مستعار ساتوشی ناکاماتو معرفی شده است که از این تاریخ به بعد تلاش هایی زیادی در راستای پیش‌برد تکنولوژی بلاک چین و کشف محدودیت ها قابلیت های آن انجام شده است، یکی از این تلاش ها ارز رمز ParsiCoin می‌باشد که در اینجا به بررسی آن می‌پردازیم.

فهرست مطالب

فهرست تصاویر

فهرست معادلات

فهرست کد ها

۱ فصل اول : مقدمه

در طول تاریخ بشر همواره ردپای معامله به روش های مختلف دیده شده است، این معاملات در ابتدای تاریخ به صورت مبادله کالا با کالا صورت می‌گرفتند که به مرور و با گذر زمان به روش های کارآمد تری نظیر انتخاب یک کالا به عنوان مرجع روی اورده شد، تا اینکه در نهایت با ضرب اولین سکه ها از طلا توسط اقوام باستانی در آسیای صغیر مفهموم پول ایجاد شد.

۱-۱ تاریخچه مختصری از پول

اولین پول ها توسط قوم باستانی لیدی ها ساکن در منطقه آسیاس صغیر تقریبا در قرن ۷ قبل از میلاد ضرب شدند این سکه ها اغلب از جنس فلزات گرانبها مانند طلا ،نقره و یا مس ضرب می‌شدند و حاوی مهر حکومت ضرب کننده آنها به همراه تصویر پادشاه و یا اشخاص برجسته آن قوم بودند که به این سکه ها رسمیت و اعتبار می‌بخشید رفته رفته دیگر اقوام نیز به ضرب سکه روی آوردند و این گونه بود که دوران سکه ها در تاریخ بشر آغاز شد، به طوری که امروزه نیز در اکثر تحقیقات باستان شناسی از سکه های هر قوم و ملیت به عنوان نماد آن قوم و ملیت یاد شده و دارای اهمیت زیادی می‌باشند.

۱-۱-۱ استاندارد طلا [۱]

با گذر زمان و افزایش حجم معاملات تجاری لزوم حمل و نقل مقدار زیادی سکه ایجاد می‌شد که حمل این سکه ها دردسر های زیادی را به همراه داشت، بنابر این سازمان هایی به وجود آمدند که در ازای تعداد مشخصی سکه حواله هایی صادر میکردند که در اکثر مناطق دنیای آن زمان با مراجعه به دفتر آن سازمان در هر شهر قابل تبدیل شدن به آن مقدار سکه بودند بنابر این این حواله ها دارای ارزش معادلی به سکه طلا بودند که به مرور زمان در بین مردمان رواج پیدا کردند به طوری که امروز نیز در قوانین اساسی اکثر کشور ها میزان مشخصی از طلا به ازای هر واحد پول آن کشور اختصاص داده می‌شود [۲] . اما در نهایت با توجه به لزوم وجود نقدینگی در کشور ها و با افزایش جمعیت و برخی عوامل دیگر دولت ها مجبور به چاپ مقادیر پول بیشتر از ذخایر طلای خود شدند که به این ترتیب ارز های بی پشتوانه [۳] ایجاد شدند.

۱-۱-۲ پول بی‌پشتوانه

پول بی‌پشتوانه ،پول حکمی، پول دستوری یا پول اعتباری پولی است که ارزشش ذاتی نبوده و تنها ناشی از دستور دولتی یا قانون باشد. نام آن از واژه لاتین فیات به معنی بگذارید انجام شود گرفته شده‌است. بنابر این در این نوع پول به دلیل عدم وجود یک پشتوانه محکم ارز پول معادل ارز چاپ کننده آن در نظر گرفته می‌شود و با تغییرات شرایط سیاسی یا اقتصادی یک

[1] Gold Standard

[۲] به طور مثال هر یک ریال ایران برابر یکصدوهشت‌هزاروپنجاه‌وپنج‌ده‌میلیونیم (۰،۰۰۱۰۸۰۵۵) گرم طلای خالص است.

[3] Fiat Currency

کشور ارزش پول آن کشور نیز تغییر می‌کند. همچنین دولت ها در زمان های کمبود نقدینگی در کشور هایشان با اقدام به چاپ بی رویه این نوع پول باعث ایجاد تورم و کاهش ارزش پول خود می‌شوند، مجموع عوامل ذکر شده دلیل تغییراتی است که امروزه در قیمت ارز های کشور های مختلف نسبت به یکدیگر شاهد هستیم؛ لازم به ذکر است که تمامی ارز های مرجع امروزی از نوع پول بی پشتوانه می‌باشند.

۲-۱ تراکنش های مالی و نظام بانکی

با گذر زمان و نیاز به انجام تراکنش های مالی با مبالغ سنگین یا انجام پرداخت در زمان مشخصی در آینده بانک های کشور های مختلف اقدام به ایجاد سازکار های مختلفی نظیر دسته چک ها و یا چک های رمز دار بین بانکی و همچنین انتقال از حساب یک شخص به حساب شخص دیگر نمودند که در نهایت با ورود تکنولوژی های جدید مواردی از قبیل کارت های اعتباری [۱] و یا کارت های نقدی [۲] امروزه به عنوان یکی از روش های اصلی انجام تراکنش های مالی شناخته می‌شوند که این کارتها قابلیت انجام انواع تراکنش های مالی مختلف را با کمک دستگاه هایی نظیر خود پرداز ها [۳] و یا پایانه های فروشگاهی [۴] فراهم می‌آوند همچنین به کمک درگاه های پرداخت اینترنتی قابلیت پرداخت با کمک اینترنت نیز فراهم شده است.

۱-۲-۱ اشکالات تراکنش های مالی فعلی

اگرچه انجام تراکنش های مالی امروزه به راحتی از طریق کارت هایی نظیر Master Card و یا VISA Card در سطح جهانی انجام می‌پذیرند اما این سبک انجام تراکنش دارای اشکالاتی است که در ادامه به معرفی آنها میپردازیم.

۱. زمان بین انجام تراکنش و تسویه می‌تواند طولانی باشد.
۲. دوباره کاریها و نیاز به شخص سوم برای تایید اعتبار و حضور واسطه ها.
۳. کلاهبرداری، حمله های سایبری و حتی اشتباه های کوچک به هزینه و پیچیدگی کسب وکار می افزایند و اگر یک سامانهی مرکزی مانند بانک به خطر بیفتد، همگی مشارکت کنندگان در شبکه با مخاطره روبه رو خواهند شد.
۴. برای استفاده از خدمات این کارت های اعتباری اغلب نیاز به پرداخت هزینه های اولیه زیاد می‌باشد.
۵. دریافت این نوع کارت های شامل فرآیند های وقت گیر و کاغذ بازی ها و بررسی سابقه افراد است.

[1] Credit Card
[2] Debit Card
[3] ATM
[4] POS

۶. این نوع سامانه ها دارای شفافیت کافی در برخی موارد نمی‌باشند، در حقیقت هیچ کس نمی‌داند که آن سازمان مرکزی صادر کننده کارت ها و حسابها چگونه به انجام این کارها می‌پردازد.
۷. وجود چنین سامانه های مرکزی با نفوذ افراد و دولت های مختلف رخ می‌دهد که در برخی موارد می‌تواند باعث حذف برخی افراد از سامانه و قطع سرویس دهی به آنها شود نظیر تحریم های اعمال شده علیه کشور ایران که دسترسی این کشور و مردم آن را به بیشتر سامانه های پرداخت جهانی قطع کرده است.

بنابر این و باتوجه به موارد فوق و همچنین گسترش روز افزون اینترنت در جهان برخی دانشمندان حوزه کامپیوتر اقدام به ایجاد نوع خاصی از ارز ها که ارز های الکترونیکی هستند نمودند که بتواند مشکلات ذکر شده در فوق را برطرف نماید و به عنوان یک سبک پرداخت جهانی با شفافیت عملکرد و بدون نیاز به اعتماد به یک شخص ثالث به عنوان مرجع تراکنش های مالی جهانی در نظر گرفته شوند.

۱-۳ ارز های دیجیتال [1]

به طور کلی هر ارزی که نمود فیزیکی نداشته باشد و به عنوان واحد های کامپیوتری در شبکه های خاص مورد استفاده قرار گیرد می‌تواند نوعی ارز دیجیتالی به حساب آید بنابر این باید توجه داشت که مفهوم ارز دیجیتالی به طور کلی برابر با رمز ارز [2] که در ادامه به بررسی آن خواهیم پرداخت نمی‌باشد و هر نوع واحد پولی دیجیتالی حتی سکه ها و یا اعتبارات موجود در اکثر بازی های کامپیوتری را نیز می‌توان نوعی ارز دیجیتال دانست.

۱-۳-۱ تاریخچه ارز های دیجیتال

ابتدایی ترین نوع ارز های دیجیتال که در حقیقت مفاهیم استفاده شده در برخی از آنها مبنای کارکرد رمز ارزهای امروزی است اغلب دارای یکاهایی معادل با یکی از ارز های رایج بودند که کاربران با پرداخت میزان معینی ارز دولتی ، ارز دیجیتال معادل آن را دریافت کرده و می‌توانستند در معاملات اینترنتی خود از آنها استفاده کنند نمونه این ارز ها رزرو آزادی [3] بود که در دهه ۱۹۹۰ میلادی همزمان با حباب دات-کام ایجاد شد. اگرچه اغلب این ارز های دیجیتال از مفاهیمی نظیر امضا دیجیتال [4] و مکانیزم هایی شبیه به گواهی اثبات کار[5] و یا به اختصار POW نیز استفاده می‌کردند و در برخی موارد تا حدی

[1] Digital Currency
[2] Crypto Currency
[3] Liberty Reserve
[4] Digital signature
[5] Proof-of-Work

نیز به صورت نظیر به نظیر [1] کار می‌کردند اما همه آنها در نهایت برای ایجاد توافق[2] به یک سازمان مرکزی وابسته بودند که در حقیقت این بدان معنی بود که در انجام بزرگترین هدف خود که حذف واسطه ها از انجام تراکنش ها بود نا موفق بودند بنابر این اغلب این ارز های دیجیتال به سرعت به فراموشی سپرده شدند با این حال مفاهیم استفاده شده در بسیاری از آنها به عنوان مبنای رمز ارز های امروزی قرار گرفت. در نهایت در سال ۲۰۰۹ میلادی با معرفی بیت کوین [3] مشکل نیاز به واسطه برای انجام تراکنش ها حل شد و برای انجام توافق از یک الگوریتم قوی کامپیوتری به نام بلاک چین [4] استفاده شد.

[1] Peer-To-Peer
[2] consensus
[3] BitCoin
[4] Block Chain

۲ فصل دوم : برخی مفاهیم پایه بکار رفته در رمز ارز ها

برای بررسی بیشتر رمز ارز ها و بلاک چین ابتدا لازم به نظر می‌رسد تا با برخی مفاهیم پایه و الگوریتم های به کار رفته در این سامانه ها آشنا شویم.

۱-۲ رمز نگاری[1]

از ابتدای تاریخ نوع بشر همواره نیاز به رمزکردن پیام های خود را احساس کرده است ،تا به طور مثال در شرایط جنگی و یا موارد خاص بتواند پیامهای خود را بین متحدین خود به گونه ای امن منتقل کند شاید اهمیت رمز نگاری را بتوان با بررسی تاریخ جنگ جهانی دوم مشخص کرد که با شکسته شدن کد های انیگما [2]نوشته شده نازی ها توسط دانشمند انگلیسی و پدر علم کامپیوتر آلن تورینگ [3] سر نوشت جنگ به نفع متحدین تمام شد.

در ابتدا بشر برای رمزنگاری پیام های خود از روش ها مختلفی نظیر قرار دادن سیمبل ها و رمز های خاص استفاده می‌کرد تا اینکه پیوند این رمز ها با ریاضیات و الگوریتم های خاص ریاضی پیدا شد و با بررسی رمزنگاری به عنوان شاخه ای از علم ریاضی بشر موفق به ایجاد رمز های پیچده شد و در نهایت با ورود کامپیوتر ها عرصه رمز نگاری نیز توسط این ماشین های قدرتمند دچار تغییراتی شد و الگوریتم های پیچیده رمزنگاری کامپیوتری ایجاد شدند به طوری که امروزه برخی از آنها نطیر الگوریتم AES [4] به قدری امن در نظر گرفته می‌شوند که شکستن رمز های آنها تقریبا غیر ممکن به نظر می‌رسد.

۱-۱-۲ رمز نگاری متقارن [5]

این نوع رمزنگاری به نوعی خاصی از رمزنگاری گفته می‌شود که در آن برای رمز کردن و رمزگشایی پیام از یک کلید استفاده می‌شود ،کلیدها ممکن است مشابه باشند یا ممکن است رابطه‌ای ساده بین دو کلید وجود داشته باشد. کلید، در عمل، نشان دهنده یک راز مشترک بین دو یا چند طرف است که می‌تواند برای حفظ اطلاعات خصوصی مورد استفاده قرار گیرد. این نیاز که هر دو طرف، دسترسی به کلیدهای مخفی داشته باشند یکی از اشکالات اصلی رمزنگاری کلید متقارن است، چرا که در حقیقت با توجه به نیاز دو طرف برای دانستن یک کلید مشخص این نوع رمزنگاری بین دو طرف که به یکدیگر اعتماد ندارند قابل استفاده نیست و دو طرف باید از قبل یکدیگر را بشناسند و آن رمز مشخص را بین خود پذیرفته باشند. برای این نوع رمزنگاری اغلب رو گروه از الگوریتم های جریانی و بلوکی مورد استفاده قرار می‌گیرند که الگوریتم های جریانی اغلب به رمز کردن یک واحد کوچک داده یا همان بیت به همان صورت که داده در جریان است می‌پردازند و الگوریتم های بلوکی به رمز

[1] cryptography
[2] Enigma Machine
[3] Alan Mathison Turing
[4] Advanced Encryption Standard
[5] Symmetric-key

کردن یک تعداد مشخص از واحد داده ها به عنوان بلوک می‌پردازند ، به طور مثال الگوریتم AES به رمزنگاری داده در بلوک های ۱۲۸ بیتی می‌پردازد.

۲-۱-۲ رمز نگاری نامتقارن [1]

این نوع رمز نگاری در مقابل رمزنگاری متقارن قرار دارد و در آن‌ها برای رمز کردن و رمزگشایی پیام از ۲ کلید متفاوت استفاده می‌شود که یکی از آنها به کلید عمومی [2] و دیگری به کلید خصوصی [3] شهرت دارند و همچنین یک رابطه ریاضی بین این ۲ کلید برقرار است به طوری که همواره از کلید خصوصی می‌توان کلید عمومی را به دست آور اما از کلید عمومی نمی‌توان کلید خصوصی را استخراج کرد.

کارکرد الگوریتم های رمزنگاری نا متقارن به این صورت است است که هر فرد ابتدا با ایجاد یک کلید خصوصی و نگه داشتن آن نظیر خود به صورت امن و به دست آوردن کلید عمومی نظیر آن کلید خصوصی و دراختیار عموم قرار دادن آن کلید عمومی به رد و بدن کردن پیام های رمز شده می‌پردازد، به این صورت که هر پیام توسط کلید عمومی که در اختیار همه هست رمز می‌شود اما فقط با استفاده از کلید خصوصی که در اختیار خود فرد است قابل رمزگشایی خواهد بود ،بنابر این همه پیام های مورد نظر برای یک نفر فقط و فقط توسط همان فرد قابل رمزگشایی خواهند بود در حالی که همه افرادی که کلید عمومی را در اختیار دارند قادر به رمز کردن پیام های خود خواهند بود از الگوریتم های معروف رمزنگاری نامتقارن می‌توان از الگوریتم RSA [4] که بر پایه هم نهشتی و با کمک اعداد اول بزرگ کار می‌کند و ECC [5] بر اساس ساختاری جبری از منحنی‌های بیضوی بر روی میدانهای متناهی طراحی شده‌است نام برد البته اغلب این الگوریتم ها با کمک الگوریتم های دیگر مورد استفاده قرار می‌گیرند.

۲-۱-۲-۱ الگوریتم های تبادل کلید

الگوریتم های رمزنگاری نامقارن اغلب دارای یک مشکل اساسی می‌باشند که محدودیت سایز پیام قابل رمزنگاری توسط آن‌ها می‌باشد به طوری که در صورتی که اندازه پیام از حدی بزرگتر باشد برای رمز نگاری آن نیاز به کلید طولانی تری خواهد بود و طولانی کردن کلید نیز تا حدی ممکن است ،بنابر این برای استفاده از آنها یا باید پیام را به قطعا کوچک شکست و رمز نمود و یا از راهکار دیگری بر مبنای الگوریتم های تبادل کلید استفاده نمود ،در حقیقت همانطور که قبلا اشاره شد الگوریتم

[1] Public-key cryptography, or asymmetric cryptography
[2] Public Key
[3] Private Key
[4] Rivest–Shamir–Adleman
[5] elliptic curve cryptography

های رمزنگاری متقارن دارای مشکل اساسی نیاز به دانستن کلید یکسان توسط هر دو طرف قبل از شروع انتقال پیام می‌باشند اما در بستر های ناامن که نیاز به رمزنگاری هست در صورتی که دو طرف از قبل با یکدیگر در ارتباط نبوده باشند چطور می‌توان کلید مورد نظر را انتقال داد؟ در اینجا با کمک الگوریتم های انتقال کلید و با کمک رمزنگاری نامتقارن می‌توان ابتدا به تبادل کلید الگوریتم رمزنگاری متقارن پرداخت و سپس با کمک آن کلید در ادامه پیام هارا به صورت متقارن رمزنگاری کرد از الگوریتم های معروف این دسته می‌توان از پروتکل تبادل کلید دیفی-هلمن [1] نام برد الگوریتم استفاده شده در پروتکل امن انتقال ابرمتن[2] ویا به اختصار Https نیز تقریبا کارکردی به همین صورت دارد.

۲-۲-۱-۲ امضای دیجیتال

از دیگر کاربرد های رمزنگاری نامتقارن می‌توان به امضا دیجیتال اشاره کرد که در حقیقت این الگوریتم امضا دیجیتال است که در ارز رمز ها نیز بسیار کاربرد دارد و در حقیقت به اثبات هویت ارسال کننده یک پیام و یا تراکنش می‌پردازد به مانند امضا عادی در سیستم های سنتی که امضا هر طرف در انتهای هر نامه و پیامی به معنی تایید فرد مورد نظر است.

کارکرد امضا دیجیتال به این صورت است که ابتدا فرد نویسنده یک پیام و یا تراکنش برای اثبات هویت خود با کمک توابع درهم سازی یک طرفه [3] با طول ثابت به هش کردن پیام خود می‌پردازد و سپس این بار با کمک کلید خصوصی خود رشته به دست آمده را رمز می‌کند حال پیام رمز شده نهایی را به انتهای پیام اصلی می افزاید که به این فرایند امضا کردن پیام گفته می‌شود حال به ارسال این پیام به شخص یا اشخاص مورد نظر آنها با داشتن کلید عمومی طرف می‌توانند امضا فرستاده شده را رمزگشایی کرده و اصل پیام را نیز هش کنند و رشته خروجی را با رشته بدست آمده از رمزگشایی امضا تطبیق دهند در صورت برابر بودن ۲ رشته اثبات هویت ارسال کننده پیام صورت می‌گیرد که در حقیقت فرد متناظر با ان کلید عمومی است.

۲-۲-۱-۳ رمزنگاری ECC

باتوجه به اینکه در اغلب رمز ارزها و سامانه های امروزی از الگوریتم ECC به عنوان الگوریتم رمزنگاری نامتقارن استفاده می‌شود به نظر می‌رسد لازم است که به اختصار نحوه کارکرد آن را توضیح دهیم.

همانطور که پیش تر ذکر شد این الگوریتم بر اساس ساختاری جبری از منحنی‌های بیضوی بر روی میدانهای متناهی طراحی شده‌است. که این امر باعث نیاز به کلید کوچک تری نسبت به دیگر روش های رمزنگاری نا متقارن می‌شود، در حقیقت

[1] Diffie–Hellman key exchange
[2] Hypertext Transfer Protocol Secure
[3] Hash Function

برای اهداف امروزی رمزنگاری، منحنی بیضوی یک منحنی مسطح است که متشکل از نقاط رضایت بخش معادله می‌باشد.

معادله ۲-۱: منحنی های بیضوی در ۲ بعد $Y^2 = X^3 + aX + b$

همراه با یک نقطه برجسته در (مختصات در اینجا از یک حوزه ثابت متناهی از مشخصه که با ۲ یا ۳ برابر نیست انتخاب می‌شوند، یا اینکه معادله منحنی تا حدودی پیچیده‌تر خواهد بود) این مجموعه همراه با عملیات گروهی از نظریه گروه بیضوی از گروه Abelian، با نقطه‌ای در بینهایت به عنوان عنصر هویت می‌باشند. ساختار گروه از گروه مقسوم علیه تنوع جبری زیرین ارث بری می‌کند. همان‌طور که برای دیگر سیستم‌های رمزنگاری کلید عمومی محبوب، بدون اثبات ریاضی برای امنیت ECC از سال ۲۰۰۹ منتشر شد.

درنهایت باید دانست که امنیت کاملECC بستگی به توانایی محاسبه ضرب نقطه‌ای و عدم توانایی برای محاسبه حاصلضرب با توجه به نقاط اصلی و نقاط تولید شده دارد.

۲-۱-۲-۴ تهدید های متوجه رمزنگاری نامتقارن

امنیت شیوه های امروزی رمز نگاری متقارن اغلب بر پایه سخت بودن حل مسائلی نظیر تجزیه اعداد اول و یا مسئله حل لگاریتم گسسته برای کامپیوتر های امروزی مطرح می‌شوند که حل این مسائل به طور عادی از روابط نمایی پیروی می‌کند بنابر این برای حل آنها صرف زمان بسیار بسیار زیادی توسط کامپیوتر های امروزی لازم خواهد بود به طوری که می‌توان به دست آوردن پاسخ آنهارا عملا غیر قابل دست یابی دانست ، اما نوع دیگر کامپیوتر های که کامپیوتر های کوانتمی [1] معروف هستند. با کمک الگوریتم کوانتمی شور [2] قادر به حل این مسائل در زمان معقولی می‌باشند که در حقیقت این مورد این نگرانی را در بین بسیاری از افراد به وجود آورده که این نوع کامپیوتر ها می‌توانند تهدیدی برای اغلب روش های رمزنگری و سامانه های مبتنی بر رمزنگاری نامتقارن امروزی باشند ، که از همین رو تلاش هایی برای ایجاد الگوریتم های رمزنگاری نامتقارن کوانتمی [3] شده است ، البته لازم به ذکر است که هنوز کامپیوتر های کوانتمی قدرتمند که توانایی اجرای الگوریتم گفته شده را داشته باشند ساخته نشده اند ، اما تخمین زده می‌شود که تا سال ۲۰۳۰ میلادی نمونه اولیه چنین کامپیوتر هایی ساخته شوند ، نکته قابل توجه دیگر آن است که کامپیوتر های کوانتمی تهدیدی برای الگوریتم هایی نظیر AES نمی‌باشند.

[1] Quantum computer
[2] Shor's algorithm
[3] Quantum cryptography

۲-۲ توابع درهم سازی یک طرفه

توابع درهم سازی یک طرفه به توابعی گفته می‌شود که اغلب با دریافت یک رشته بیتی با درهم ریختن رشته ورودی به ایجاد یک رشته خروجی با طول ثابت می‌پردازند به طوری که به دست آوردن رشته ورودی از رشته خروجی امکان پذیر نباشد و همچنین به ازای هر رشته منحصر به فرد در ورودی یک رشته منحصر به فرد در خروجی ایجاد شود. به طوری که با داشتن رشته ورودی همواره به توان به یک رشته خروجی رسید و به ازای هیچ دو رشته ورودی غیر یکسانی ،یک رشته خروجی یکسان حاصل نشود. به طور کلی کاربرد این نوع توابع در شماره گذاری رشته ها و جداول داده درهم [۱] کاربردی می‌باشند.

۲-۲-۱ تابع درهم‌سازی رمزنگارانه [۲]

این نوع توابع درهم سازی نوع خاصی از توابع در هم سازی می‌باشند که یک رشته با طول نا مشخص را به یک رشته با طول ثابت درهم ریزی می‌کنند به طوری که رشته خروجی نمایشی از کل محتوای متن یا رشته ورودی است و می‌توان آن را نوعی «اثر انگشت دیجیتالی» برای آن متن به حساب آورد ، این نوع توابع درهم سازی در امضا دیجیتال ، ذخیره اطلاعات حیاتی مانند کلمه عبور کاربران در پایگاه داده ، بلاک چین و بسیاری موارد دیگر کاربرد دارند از معروف ترین توابع درهم سازی رمزنگارانه می‌توان از MD4,MD5,SHA-1,SHA-2,SHA-3 نام برد.

از دیگر خصوصیت این توابع که در سیستم های بلاک چینی بسیار قابل توجه است آن است که در حقیقت امکان پیش بینی خروجی از روی وردی به هیچ عنوان امکان پذیر نیست، در حقیقت کارکرد آنها به این صورت نمی‌باشد که رشته خروجی به ازای تغییرات مشخص ورودی به یک رشته خاص در خروجی میل کند که بتوان آن را حدس زد و با هر تغییر بسایر کوچک در رشته ورودی، رشته خروجی تغییرات قابل توجهی خواهد کرد.

۲-۳ درخت مرکله [۳]

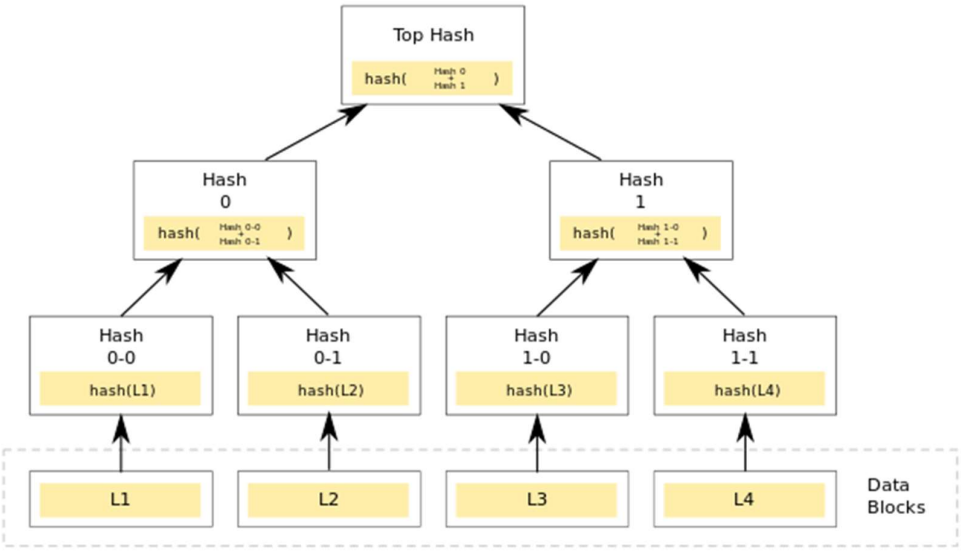
درخت مرکله و یا درخت درهم سازی به طور معمول یک درخت دودویی [۴] و یا یک درخت پیشوندی [۵] است که برگ های آن شامل یک سری داده مشخص می‌باشند ، سپس در مراحل بالاتر مقدار هر گره از مقدار هش فرزندان خود به دست می‌آید به همین صورت از گره های والد برای برگ ها شروع به هش کردن مقادیر میکنیم و سطح به سطح در درخت بالا می‌رویم

[1] Hash Table
[2] Cryptographic hash function
[3] Merkle tree
[4] Binary Tree
[5] Trie

و مقدار هر گره را معادل هش مقدار فرزندان خود می‌گذاریم تا به ریشه درخت برسیم.

معادله ۲-۲: گره های درخت درهم سازی $$f(n) = Hash(f(2n+1) + f(2n+2))$$

کاربرد درخت های مرکز به طور معمول در نوع خاصی از امضا دیجیتال به نام امضای لمپورت [1] و یا تایید اعتبار یک فایل و همچنین به طور گسترده در بلاک چین می‌باشد.



تصویر ۲-۱: نمونه یک درخت درهم سازی

۲-۴ کدبندی نویسه [2]

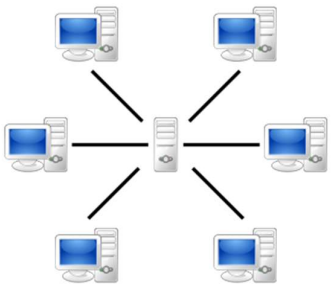
باتوجه به اینکه کوچکترین واحد قابل فهم برای کامپیوتر ها یک بیت است که دارای مقاداری معادل ۰ یا ۱ می‌باشد در نتیجه همه مفاهیم از نظر کامپیوتر ها باید دارای یک مقدار عددی باشند، بنابر این برای نویسه [3] های الفبایی استاندارد هایی در نظر گرفته شده است که هر نویسه را به یک مقدار عددی مشخص نظیر می‌کند از نمونه های این استاندارد ها می‌توان به مواردی از قبیل ASCII [4] که فقط شامل حروف انگلیسی اعداد و برخی نویسه های خاص است و یا UTF-8 که دارای نویسه های موجود در اغلب زبان های می‌باشد اشاره کرد. اما این کدبندی ها اغلب برای داده هایی به کار می‌روند که در زبان های طبیعی دارای معنی و مفهوم می‌باشند و خروجی یک تابع درهم سازی یک طرفه و یا رشته خروجی حاصل از یک الگوریتم

[1] Lamport Signature
[2] Character encoding
[3] Character
[4] American Standard Code for Information Interchange

رمزنگاری اشکال سخت و عجیبی در این نوع از کدبندی ها پیدا می‌کنند بنابر این برای نمایش خروجی این تابع ها به صورت رشته های حرفی عددی نیز استاندارد های خاصی نظیر Hex, Base64, Base58 ایجاد شده است که Hex در حقیقت نمایش هر بایت از داده در مبنای ۱۶ است که باعث طولانی شدن رشته می‌شود بنابر این استاندارد Base64 به طور معمول برای انتقال داده ها استفاده می‌شود که این استاندارد نیز به دلیل وجود نویسه هایی مانند عدد 0 و حرف o لاتین و یا حرف l کوچک لاتین و حرف I بزرگ لاتین و عدد 1 و... که در برخی فونت ها شکل یکسان دارند در صورتی که قرار باشد توسط انسانها و نه ماشین ها مورد بررسی قرار گیرند مشکلاتی را ایجاد می‌کنند که برای پیشگیری از این مشکلات از استاندارد Base58 در اغلب کریپتو کارنسی ها استفاده می‌شود که حروف و اعداد مشابه و غیر قابل تشخیص از یکدیگر در آن مشاهده نمی‌شوند.

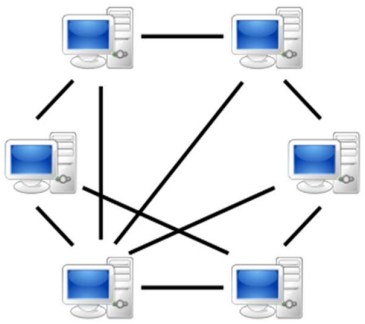
۲-۵ شبکه نظیر به نظیر

این شکبه ها در حقیقت در مقابل شبکه های سرویس دهنده-سرویس گیرنده [1] قرار دارند و به جای آنکه یک سرویس دهنده مرکزی وجود داشته باشد که همه سرویس گیرنده ها به آن متصل باشند و فقط بتوانند با سرویس دهنده انتقال پیام انجام دهند ، همه نظیر ها [2] به صورت مستقیم به هم متصل هستند و در حقیقت هر نظیر حکم یک سرویس دهنده و یک سرویس گیرنده کوچک را دارد ، این شبکه در رمز ارزها اهمیت بالایی دارند چراکه به کمک این شبکه ها امکان حذف شخص ثالث در تراکنش ها حذف شده است.



تصویر ۲-۲ : نمونه یک شبکه سرویس گیرنده-سرویس دهنده

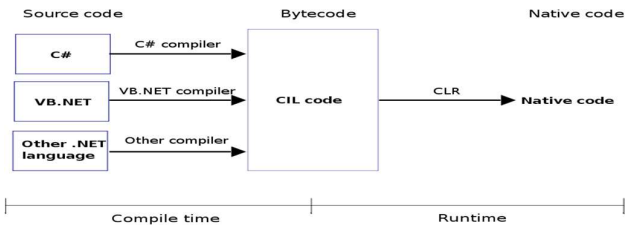
[1] Client-Server
[2] Peer



تصویر ۲-۳ : نمونه یک شبکه نظیر به نظیر

۲-۶ ماشین مجازی [۱]

ماشین مجازی در حقیقت ماشینی است که به طور مجازی روی یک ماشین فیزیکی دیگر اجرا می‌شود و یک سری عملیات خاص انجام می‌دهد ، نوع خاصی این ماشین های مجازی آنهایی هستند که در زبان های برنامه نویسی سطح بالا مانند جاوا [۲] و یا زبان های خانواده دانت نت [۳] به ترتیب با نام های JVM [۴] و CLR [۵] استفاده می‌شوند. این ماشین های مجازی اغلب مانند ریز پردازنده ها [۶] دارای یک زبان اسمبلی مانند [۷] مخصوص به خود می‌باشند که کد های زبان سطح بالاتر به این زبان ها ترجمه می‌شوند و سپس کد های خروجی حاصل در ماشین مجازی مورد نظر اجرا می‌شوند در حقیقت این ماشین های مجازی رفتار شبیه به یک پردازنده را شبیه سازی می‌کنند.



تصویر ۲-۴ : نحوه کارکرد CLR

[1] Virtual Machine
[2] Java
[3] .Net
[4] Jave Virtual Machine
[5] Common Language Runtime
[6] CPU
[7] Assembly

اغلب رمز ارزها نیز برای انجام کارهای خود دارای یک ماشین مجازی می‌باشند که تراکنش ها مورد نظر به صورت کد های این ماشین مجازی در آمده و برای تایید کد های مورد نظر اجرا می‌شوند به طور مثال می‌توان از EVM [1] در این زمینه نامبرد.

[1] Ethereum Virtual Machine

۳ فصل سوم : بلاک چین

همانطور که گفته شد در سال ۲۰۰۹ میلادی با معرفی بیت کوین به عنوان اولین رمز ارز توسط فرد یا افرادی با نام مستعار ساتوشی ناکاماتو [1] مفهوم رمز ارزها پابه عرصه وجود گذاشتند به طوری که امروزه بیش از بیش از ۱۰۰۰ رمز ارز مختلف در دنیا وجود دارد ، بیت کوین برخلاف ارز دیجیتال قبل از خود توانست مشکل نیاز به طرف سوم قابل اعتماد را با کمک یک مکانیزم انقلابی به نام بلاک چین حل کند که در این فصل به بررسی این سیستم می‌پردازیم.

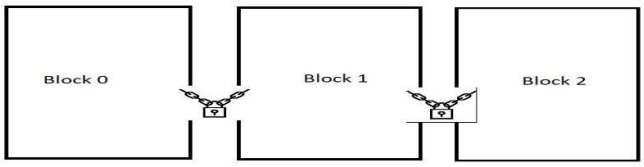
در حقیقت بلاک چین را می‌توان به مانند یک سیستم عامل در نظر گرفت و رمز ارزهارا به عنوان برنامه هایی که بر روی این سیستم عامل بزرگ و توزیع شده اجرا می‌شوند.

۳-۱ تعریف مفهومی بلاک چین

بلاک چین یک دفتر کل بزرگ و توزیع شده می‌باشد که کار ثبت و رهگیری دارایی هارا به صورت تراکنش [2] محور انجام می‌دهد به طوری که امکان تغییر یا حذف اطلاعات درج شده در این دفتر کل وجود ندارد و فقط می‌توان اطلاعات جدیدی را به آن افزود ، بلاک چین کار فرآیند ثبت تراکنشها و ردگیری داراییها را در یک شبکه‌ی کسب‌وکار ساده میکند. یک دارایی میتواند ملموس مانند خانه، خودرو، پول نقد، زمین و یا ناملموس مانند مالکیت معنوی نظیر حق اختراع، حق چاپ یا نام اعتباری باشد. تقریباً هرچیز ارزشمندی میتواند در یک شبکهی بلاک چین ردگیری و معامله شود و مخاطرات و هزینه‌ها را برای همه‌ی طرفهای درگیر کاهش دهد.

۳-۲ تعریف ساختاری بلاک چین

در ساده ترین حالت ساختار بلاک چین را همانطور که از نامش پیداست می‌توان به صورت زنجیره ای از بلاک های داده ای دانست که به طور متوالی پشت یکدیگر قرار می‌گیرند و هر بلاک به طوری به بلاک قبلی خود وابسته است.



تصویر ۳-۱ : شمای کلی یک بلاک چین

[1] Satoshi Nakamoto
[2] Transaction

۳-۳ ساختار هر بلاک

باتوجه به اینکه بلاک چین از زنجیره ای از بلاک ها ساخته شده است بنابر این ضروری به نظر می‌رسد که ابتدا به بررسی ساختار هر بلاک به صورت جز به جز بپردازیم تا در ادامه بتوانیم کاربرد هر یک از این بخش هارا در یک بلاک چین بررسی کنیم.

هر بلاک به طور معمول حداقل دارای ۳ بخش زیر است:

۱. بخش داده ای
۲. نماد بلاک
۳. نماد بلاک قبل از خود

حال به بررسی هر کدام از این بخش ها می‌پردازیم:

۱-۳-۳بخش داده ای بلاک

بخش داده ای یک بلاک در ساده ترین حالت ممکن می‌تواند یک رشته متنی حاوی یک پیام باشد و یا اطلاعات تراکنش های مالی مختلف به همراه اطلاعات دیگری نذیر یک مقدار متغییر ، زمان ساخت بلاک ، شماره بلاک و ... که در رمز ارزها استفاده می‌شود.

۲-۳-۳ نماد یا امضا بلاک

برای هر بلاک نماد یا امضا آن بلاک برابر است با خروجی تابع درهم ریزی یک طرف از همه بخش های به غیر از خود بخش نماد بلاک یا به عبارت دیگر مقدار هش بخش های مختلف بلاک.

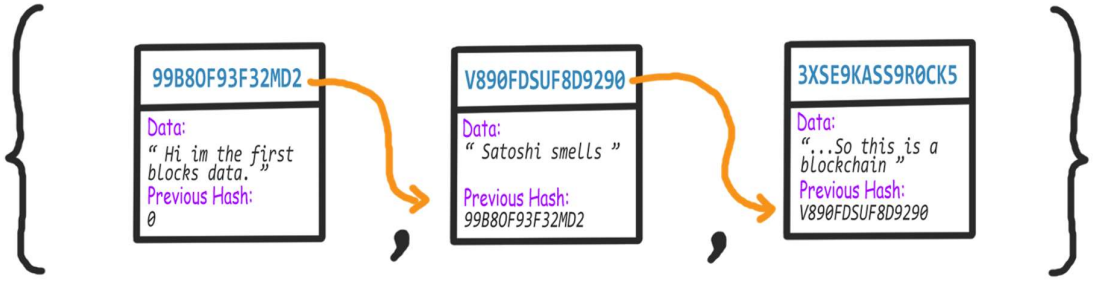
۳-۳-۳ نماد بلاک قبل از خود

هر بلاک شامل نماد بلاک قبلی خود نیز می‌شود. که در حقیقت این بخش باعث ایجاد زنجیره و مرتبط شدن بلاک ها به یکدیگر می‌شود.

۳-۴ بررسی ساختار بلاک چین

همانطور که گفته شد بلاک چین زنجیره ای از بلوک های داده ای به هم وابسته است اما این بلاک ها چگونه به هم وابسته می‌شوند؟

در حقیقت وظیفه ایجاد وابستگی بلاک های یک بلاک چین فارغ از بخش داده ای و بخش های دیگر آن بر عهده بخش شامل نماد بلاک قبلی می‌باشد، به این صورت که با قرار دادن نماد بلاک قبلی در هر بلاک آن بلاک را به بلاک قبلی خود وابسته می‌کنیم و باتوجه به اینکه بخش نماد بلاک قبلی برای محاسبه نماد بلاک جدید استفاده می‌شود در حقیقت با ایجاد هر بلاک ،این بلاک جدید به همه بلاک های قبل از خود وابسته است به طوری که برای ایجاد تغییر در یک بلاک نماد آن بلاک تغییر می‌کند در نتیجه نماد بلاک بعد از آن نیز تغییر می‌کند و این تغییر تا جدید ترین بلاک انتشار می‌یابد ،حال اگر بلاک چین یک بلاک چین پویا باشد به طوری که پیوسته به بلاک های آن افزوده شود عملا ایجاد تغییر در یکی از بلاک های قبلی امکان پذیر نخواهد بود و یا حداقل بسیار بسیار سخت خواهد بود بنابر این همه افرادی که در یک بلاک چین مشارکت می‌کنند می‌توانند بر روی صحیح بودن اطلاعات بلاک چین توافق کنند و به این صورت توافق مورد نظر بدون نیاز به فرد سومی ایجاد می‌شود و در حقیقت همه افراد مشارکت کننده بر روی صحت داده های داخل بلاک چین توافق دارند.



تصویر ۳-۲ : ساختار بلاک چین

۳-۵ خواص بلاک چین

هر بلاک چین باید شامل خواص زیر باشد:

۱. تغییر ناپذیری: تغییر ناپذیری در بلاک چین باتوجه به ساختار آن به وجود می‌آید.
۲. اصل بودن : مشارکت کنندگان میدانند دارایی از کجا میآید و مالکیت آن در طول زمان چگونه تغییر کرده است.
۳. قطعیت : هر بلاک دارای قطعیت است در حقیقت داده های آن قطعی و صحیح و پذیرفته شده هستند.

۴. شفافیت : نحوه ساز و کار بلاک چین دارای شفافیت است و داده های موجود در آن برای همه افراد موجود در شبکه قابل دسترسی است.

۳-۶ بررسی کارکرد شبکه های بلاک چین

اغلب شبکه های مبتنی بر بلاک چین مانند رمز ارز ها بر بستر شبکه های کامپیوتری نظیر به نظیر به فعالیت می‌پردازند به طوری که هر یک از نظیر های موجود در شبکه که اینجا ند [۱] نامیده می‌شوند یک نسخه از بلاک چین را در سیستم خود نگه می‌دارند و قابلیت خواندن اطلاعات بلاک چین و افزودن بلاک جدید در شبکه را دارا می‌باشند و به ازای افزوده شدن هر بلاک جدید ،ندی که آن بلاک را به بلاکچین افزوده است موظف است که این موضوع را به همه نظیر های دیگر اعلام کند تا آنها نیز اطلاعات بلاک چین خود را به روز رسانی [۲] کنند تا شبکه دچار خطا نشود.

۳-۷ انواع بلاک چین

بلاک چین ها دارای انواع مختلفی می‌باشند اما به صورت کلی به دو دسته خصوصی و عمومی تقسیم می‌شوند که بلاک چین های عمومی اطلاعاتشان در اختیار همه افراد قرار دارد و هر کسی می‌تواند در هر زمان به عنوان یک ند جدید وارد این شبکه شود و یا اطلاعات آن‌هارا مورد بررسی قرار دهد نظیر بلاک چین های ارز رمز ها، در مقابل نیز بلاک چین های خصوصی قرار دارند که یک سازمان و یا کسب و کار می‌تواند برای افراد موجود در شبکه خود آن‌را ایجاد کند و بر بستر شبکه های داخلی خود سازمان به اجرای آن بپردازد و یا با رمزگذاری همه داده های آن از دسترسی افراد غیر قابل اعتماد به داده ها جلوگیری به عمل آورد.

۳-۸ اثبات کار

اثبات کار یا POW در حقیقت یک مکانیزم در شبکه های بلاک چین است که ویژگی تغییر ناپذیری بلاک چین را تقویت می‌کند و همچنین از افزوده شدن بلاک های زیاد و غیر ضروری به سادگی در شبکه جلوگیری می‌کند، در حقیقت همان‌طور که از نامش پیداست POW یک مکانیزم است که باعث می‌شود هر فرد برای افزودن هر بلاک در شبکه موظف باشد میزان معینی کار انجام دهد.

[1] Node
[2] Sync

سازکار این مکانیزم به این صورت است که برای هر بلاک چین مقدار مشخصی به نام سختی [1] در نظر گرفته می‌شود و حال یک شرط جدید برای به دست آوردن بلاک هش به سازوکار شبکه افزوده می‌شود به طوری که مقدار عددی هش به دست آمده باید از مقدار سختی کمتر باشد برای ساده شدن کار می‌توان این شرط را به صورت ظاهر شدن تعداد معینی 0 در ابتدای مقدار هش بلاک دانست.

همانطور که گفته شد هش بلاک شامل مقدار هش بخش مختلف بلاک می‌باشد اما این بخش اغلب از داده های ثابتی تشکیل شده اند که قابل تغییر نمی‌باشد بنابر این به ازای داده های بلاک فقط و فقط یک مقدار هش تولید می‌شود که ممکن است شرط مورد نظر برای POW را نداشته باشد، برای حل این مشکل یک مقدار متغییر به نام nonce نیز به داده های بلاک افزوده می‌شود این مقدار متغییر فارغ از دیگر بخش های بلاک است و می‌تواند هر مقداری داشته باشد حال با کمک این مقدار جدید و تغییر دادن آن و محاسبه مجدد هش بلاک می‌توان هش های مختلف برای یک بلاک به دست آورد و درحقیقت مکانیزم POW مشارکت کنندگان را مجبور می‌کند تا مکررا به محاسبه مقدار هش های مختلف برای یک بلاک با nonce های مختلف بپردازند تا یک مقدار هش با شرط کوچکتر بودن از سختی یافته شود؛ به محض یافته شدن این مقدار ،ندی که هش را پیدا کرده است این مقدار جدید را و مقدار nonce ای که به ازای آن مقدار هش را به دست آورده است به دیگر ند ها اعلام می‌کند و دیگر ند ها نیز حال به سادگی با یکبار هش کردن با nonce اعلام شده می‌توانند صحت ادعای ند اعلام کننده را راستی آزمایی کنند و درصورت درست بودن ادعا آن را به عنوان بلاک جدید بپذیرند.

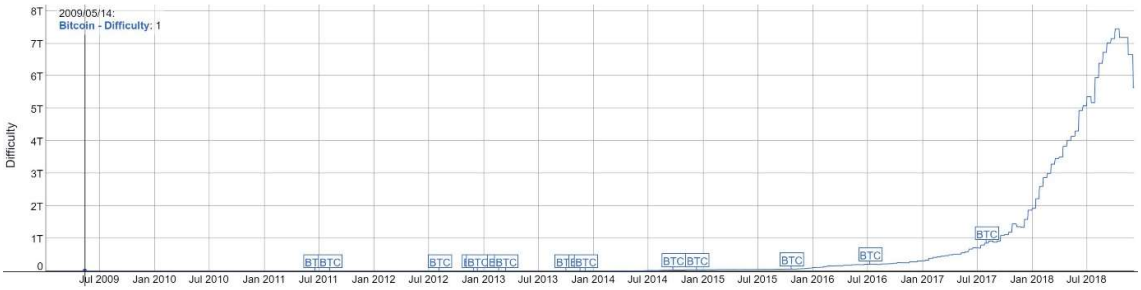
۱-۸-۳ حمله ۵۱ درصد

با افزوده شدن POW به بلاک چین حال تغییر دادن داده های بلاک بیش از بیش سخت می‌شود به طوری که برای تغییر داده های یک بلاک این بار فرد مورد نظر باید به ازای همه بلاک های بعدی نیز الگوریتم POW را نیز اجرا کند و مقدار هشی کمتر از سختی برای همه بلاک ها بیاید، که بسیار زمانگیر خواهد بود همچنین در شبکه های پویا که همواره به بلاک های آن افزوده می‌شود ند خاطی باید بتواند به قدری سریع به محاسبه POW بلاک های قبلی بپردازد که بتواند به آخرین بلاک برسد و بلاک آخر را نیز خود بدست بیاورد و به شبکه اعلام کند تا همه آن‌را بپذیرند که این اتفاق فقط در حالتی ممکن است که ند مورد نظر حداقل ۵۱ درصد قدرت محاسباتی شبکه را داشته باشد یعنی قدرت محاسباتی آن از مجموع قدرت محاسباتی همه ند های دیگر موجود در شبکه بیشتر باشد.

[1] Difficulty

۳-۸-۲ تغییرات سختی

از دیگر کاربرد های POW آن است که شبکه های بلاک چینی شلوغ می‌توانند زمان ایجاد شدن بلاک های خود را کنترل کنند تا به این صورت از رشد بی رویه بلاک چین جلو گیری به عمل آورند به این ترتیب که با افزوده شدن هر ند شانس یافته شدن بلاک جدید بیشتر می‌شود چرا که ند های بیشتری برای بدست آوردن بلاک جدید به رقابت می‌پردازند حال پس از افزوده شدن تعداد زیادی ند می‌توان با افزایش دادن مقدار سختی زمان مورد نیاز برای بدست آوردن بلاک جدید را بیشتر کرد و درحقیقت انجام مکانیزم POW را سخت تر نمود و زمان میانگین برای به دست آمدن تعداد معینی ای بلاک را کنترل نمود.



تصویر ۳-۳ : نمودار تغییرات سختی شبکه بیت کوین

۳-۹ استخراج [1] رمز ارز

در شبکه های بلاک چینی رمز ارزهایی نظیر بیت کوین که تعداد زیادی ند برای بدست آوردن بلاک بعدی رقابت می‌کنند مقداری از توکن های شبکه به عنوان جایزه [2] به هر فردی که موفق به یافتن بلاک بعدی شود اختصاص داده می‌شود تا به این صورت انگیزه برای پیاده کردن بلاک های جدید در بین افراد ایجاد شود چرا که هر فرد با پیدا کردن بلاک جدید و انجام عملیات POW در حقیقت به میزان امنیت شبکه می‌افزاید ،به این ترتیب به تعداد کوین های شبکه افزوده می‌شود و در حقیقت مکانیزمی مانند استخراج کردن ایجاد می‌شود.

به طور مثال برای شبکه بیت کوین در ابتدا جایزه هر بلاک ۵۰ بیت کوین در نظر گرفته شده بود که به ازای ایجاد شدن هر ۲۱۰ هزار بلاک این میزان نصف می‌شود تا در نهایت به ۰ میل کند به طوری که پیش بینی می‌شود در نهایت تعدادی

[1] Mining
[2] Block Reward

نزدیک به ۲۱ ملیون بیت کوین وجود خواهند داشت که این مقدار تا سال 2140 میلادی کاملا استخراج می‌شود ، لازم به ذکر است که با افزایش تعداد ند ها به مرور زمان سرعت به استخراج بیت کوین ها افزایش نمی‌یابد چرا که همانطور که اشاره شد مقدار سختی مکررا افزوده می‌شود به طوری که تقریبا هر ۲۱۰ هزار بلاک زمانی معادل ۴ سال برای ایجاد شدن نیاز خواهند داشت.

۳-۱۰دیگر اثبات ها[1]

علی رغم تمام مزایای اثبات کار این مکانیزم دارای مشکلاتی نیز می‌باشد یکی از بزرگترین ایرادات مکانیزم POW مصرف مقدار زیادی انرژی توسط کامپیوتر های ند هایی است که به اجرای آن می‌پردازند به طوری که گفته می‌شود به طور مثال شبکه بیت کوین در حال حاظر انرژی الکتریکی به اندازه انرژی مورد نیاز برای کل کشور ایرلند می‌باشد، درحالی که این انرژی برای امن کردن شبکه استفاده می‌شود اما تعداد زیادی هش تولید شده برای هر بلاک بی استفاده است و عملا این حجم عظیم انرژی به هدر می‌رود ، همچنین از دیگر اشکالات این مکانیزم آن است که برخی افراد با ایجاد مدارات مجتمع خاص و یا asics قادر به استخراج بیت کوین با سرعت زیاد می‌باشند به طوری که امروزه استخراج با cpu های معمولی عملا امکان پذیر نمی‌باشد بنابر این تعداد اثبات دیگر نیز برای ارز رمز ها مطرح شده است که در اینجا به معرفی برخی از آنها می‌پردازیم.

۳-۱۰-۱ اثبات سهام[2] (POS)

در این روش یکی از ند های موجود در شبکه به طور تصادفی برای ایجاد بلاک جدید توسط شبکه انتخاب می‌شود اما با این شرط که افرادی که دارای سهم بیشتری از توکن های شبکه می‌باشند شانس بیشتری برای انتخاب شدن دارند، که داشتن تعداد زیادی از توکن های شبکه مانع از تقلب فرد مورد نظر می‌شود چرا که اگر این ند اقدام به تقلب کند ارزش توکن های شبکه که خود مالک تعداد زیادی از آنها است کاهش می‌یابد.

۳-۱۰-۲ اثبات حافظه [3]

این اثبات به مانند اثبات کار می‌باشد با این تفاوت که در آن از توابع در هم سازی حافظه سخت می‌باشد که فرد مورد نظر برای استخراج بلاک جدید به جای داشتن قدرت محاسباتی بیشتر باید سیستمی با حافظه بیشتر داشته باشد که باتوجه به سخت بودن ساخت حافظه و گرانتر بودن آن ها شانس استفاده از مدارهای مجتمی محاسباتی درآنها کمتر است.

[1] Proofs
[2] Proof Of Stake
[3] Proof Of Capacity

۳-۱۰-۳ گواهی سوزاندن [1]

این گواه نوعی گواهی است که در آن هر ند برای ایجاد کردن بلاک جدید باید تعدادی از توکن های خود را از بین ببرد بنابر این لزوم از بین بردن مانع از تقلب می‌شود.

همچنین برخی روش های دیگر نیز که گاها ترکیبی از روش های فوق است نیز استفاده می‌شود.

۳-۱۱بلاک چین های نسل دوم [2]

همانطور که پیش‌تر اشاره شد یکی از بخش های ارز رمز ها و برنامه های مبتنی بر بلاک چین یک ماشین مجازی است که به اجرای برخی دستورات شبه اسمبلی مختص آن ارز رمز و بلاک چین می‌پردازد. این زبان شبه اسمبلی در نسخه های اولیه ارز رمز ها شامل دستورات بسیار محدودی می‌شد و به اصطلاح دارای خاصیت کامل بودن تورینگ [3] نبود، به این معنی که دارای دستورات پرش نبود اما با گذر زمان برخی از فعالان حوزه ارز رمز با بررسی پتانسیل های نهفته در این زبان، به استفاده از دستورات کامل تر و زبان های کامل تورینگ روی آوردند که به این ترتیب نسل دوم بلاک چین ها شروع شدند، یکی از نمونه های ارز رمز های با بلاک چین نسل دوم اتریوم [4]می‌باشد.

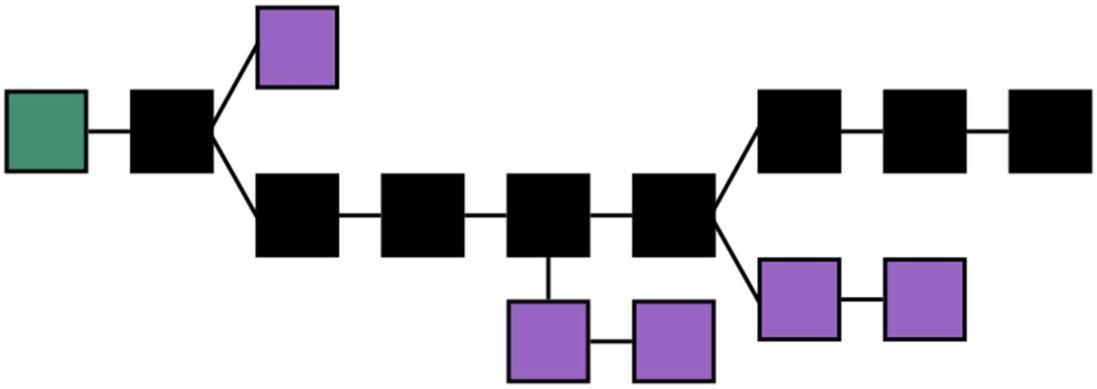
۳-۱۱-۱ قرار داد های هوشمند [5]

قرار داد های هوشمند برخلاف آنچه که ممکن است از نام ظاهری آن‌ها برداشت شود، قطعه کد هایی با زبان های کامل تورینگ ماشین های مجازی ارز رمز ها هستند که به ارسال یک تراکنش به سمت آن‌ها قادر به انجام یک کار مشخص و ثبت نتیجه آن در بلاک چین می‌باشند. بنابر این، این قطعه کد ها در محیط یک شبکه بلاک چین اجرا می‌شوند و همه مشارکت کنندگان شبکه می‌توانند با آن‌ها ارتباط برقرار کرده و انجام سرویس خاصی را مدنظر داشته باشند.

[1] Proof Of Burn
[2] Block chain 2.0
[3] Turing completeness
[4] Ethereum
[5] Smart Contract

۳-۱۲ قاعده طولانی ترین زنجیره [1]

همانطور که گفته شد در بلاک چین های شلوغ مانند بلاک چین های رمز ارز ها تعداد زیادی ند برای به دست آوردن بلاک بعدی با یکدیگر به رقابت می‌پردازند و هر ندی که بتواند زودتر از بقیه ند ها پاسخ مسئله POW را بیابد به عنوان برنده برای استخراج آن بلاک معرفی می‌شود اما اگر ۲ ند همزمان موفق به این کار شوند چه می‌شود؟ در این حالت هر ۲ ند در یک زمان پاسخ درستی برای POW بدست آورده اند و هر دو می‌توانند بلاک خود را به انتهای بلاک چین بی افزایند که این اتفاق در عمل باعث ۲ شاخه شدن بلاک چین می‌شود، اما در نهایت باید یکی از این شاخه به عنوان شاخه صحیح پذیرفته شود و به حیات خود ادامه دهد. راه حل بیت کوین و دیگر رمز ارز ها برای چنین حالتی آن است که ابتدا میگذارند هر ۲ بلاک به انتهای بلاک چین افزوده شود چرا که در مرحله اولیه هر دو بلاک برابر هستند اما با گذر و افزوده شدن بلاک های بعدی در انتهای این ۲ بلاک ۲ شاخه متفاوت از بلاک چین ایجاد می‌شود که در نهایت همه ند ها طولانی ترین شاخه ایجاد شده را به عنوان شاخه درست می‌پذیرند و کار کردن بر روی شاخه دیگر را رها می‌کنند.



تصویر ۳-۴ : قاعده طولانی ترین شاخه

همانطور که در تصویر ۳-۴ دیده می‌شود در هر مرحله پس از ۲ شاخه شدن بلاک چین بلاک چین زنجیره سیاه که طولانی تر بوده است را به عنوان زنجیره اصلی انتخاب کرده و بلاک های بنفش راه شده اند، به این بلاک های رها شده در اصطلاح یتیم شده [2] می‌گویند.

[1] Longest Chain Is Valid
[2] Orphaned Blocks

۳-۱۳ فورک [1] در بلاک چین

شبکه های بلاک چین نیز مانند دیگر برنامه های کامپیوتری همواره دستخور تغییرات و به روز رسانی هایی می‌شوند اما با توجه به این که این شبکه ها کار حساس تری را نسبت به دیگر برنامه ها انجام می‌دهند و همچنین پایگاه داده آنها که بلاک چین می‌باشد همواره در حال افزودن بلاک جدید است در برخی موارد ممکن است این تغییرات باعث ایجاد بلاک هایی با قواعد جدید شوند که با بلاک های قبل از آن تفاوت هایی دارد به چنین حالتی فورک در بلاک چین گفته می‌شود. در چنین حالتی ند هایی که نسخه نرم افزار خود را به روز رسانی [2] کرده اند بلاک هایی با خصوصیات متفاوتی از ند هایی که به روز رسانی را انجام نداده اند تولید می‌کنند.

۳-۱۳-۱ فورک نرم [3]

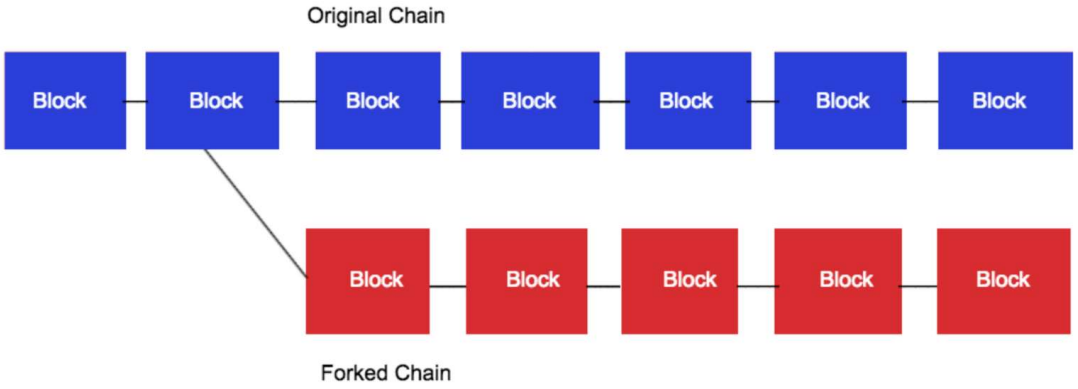
فورک نرم زمانی رخ می‌دهد که تغییرات ایجاد شده اساسی نباشند و بنابر این علی رغم این که ۲ نسخه مختلف از بلاک ها در حال ایجاد شد هستند اما این بلاک ها می‌توانند بدون رخ دادن مشکل خاصی پشت یکدیگر قرار بگیرند و شبکه کلی دچار مشکل نمی‌شود بنابر این ند هایی که به روز رسانی کرده اند و ند هایی که به روز رسانی نکرده اند می‌توانند کار کردن بر روی یک بلاک چین رو ادامه دهند.

۳-۱۳-۲ فورک سخت [4]

این نوع فورک زمانی رخ می‌دهد که تغییرات مذکور بسیار اساسی باشند به طوری که بلاک های تولید شده با هر نسخه نمی‌توانند کنار یکدیگر قرار گیرند و از نقطه ایجاد به روز رسانی عملا بلاک چین به ۲ بلاک چین مجزا تبدیل می‌شود در چنین حالتی ند هایی که به روز رسانی را دریافت کرده اند روی یک بلاک چین و ند هایی که به روز رسانی را دریافت نکرده اند روی یک بلاک چین دیگر کار می‌کنند که به هیچ عنوان سازگار [5] نمی‌باشند و عملا در چنین حالتی ممکن است توکن های ارز رمز به ۲ توکن متفاوت تجزیه شوند، به طور مثال پس از یکی از اپدیت های شبکه بیت کوین در سال ۲۰۱۶ میلادی با ایجاد فورک سخت توکن های شبکه به ۲ نوع بیت کوین معمولی و بیت کوین کلاسیک [6] تقسیم شدند به طوری که دارندگان کیف پول هریک نمی‌توانند توکن های نوع دیگر را داشته باشند.

---

[1] Fork
[2] Update
[3] Soft Fork
[4] Hard Fork
[5] Compatible
[6] Bitcoin Classic



تصویر ۳-۵: نمونه فورک سخت در بلاک چین

همانطور که در تصویر ۳-۵ مشاهده می‌شود پس از یک فورک سخت بلاک چین به ۲ بلاک چین متفاوت تبدیل می‌شود و برخلاف حالت طولانی ترین زنجیره اینجا هر ۲ زنجیره قابل قبول هستند اما نه توسط همه ند ها بلکه برخی ند ها یک زنجیره و برخی ند ها زنجیره دیگر را قبول کرده و روی زنجیره مورد نظر خود مشغول به کار می‌شوند.

۳-۱۴ مشکلات بلاک چین ها

شبکه های بلاک چینی نیز علی رغم همه مزیت هایی که به ارمغان می‌آورند خالی از اشکال نیستند که در اینجا به برخی از اشکالات این شبکه ها اشاره مختصری می‌کنیم.

یکی از ایرادات این شبکه به دلیل مکانیزم POW و مصرف انرژی بیش از حد رخ می‌دهد که تلاش هایی برای حل این مشکل با سعی بر ایجاد کردن اثبات های متفاوت انجام شده است اما هیچکدام به کارآمدی اثبات کار یا POW نمی‌باشند.

در بسیاری از بلاک چین ها امکان پرداخت های با مبلغ بسیار کم وجود ندارد[1] که این خود می‌تواند ناشی از عوامل مختلفی باشد، به طور مثال در شبکه بیت کوین برای هر تراکنش مبلغی نیز به عنوان هزینه تراکنش [2] دریافت می‌شود که اگر این مبلغ کمتر از حد معینی باشد ماینر ها رقبتی به پردازش آن تراکنش نخواهند داشت حال اگر فردی بخواهد تراکنشی با مبلغی کمتر از هزینه آن انجام دهد عملا غیر منطقی می‌باشد.

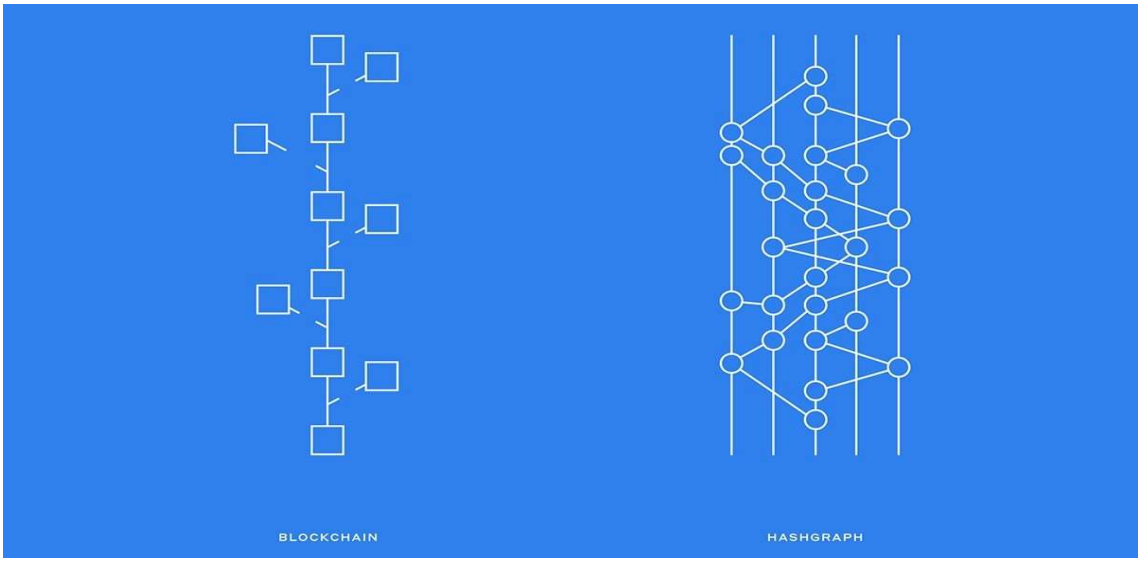
یکی از مهمترین مشکلات نیز مشکل مقیاس پذیری [3] در شبکه های بلاک چینی می‌باشد. که در این حالت باتوجه به

[1] Micro Payment
[2] Transaction Fee
[3] Scalability

نیاز انجام کار مشخصی و سپری شدن زمانی برای افزوده شدن هر بلاک و همچنین محدودیتی که در حجم اطلاعاتی که بلاک می‌تواند ذخیره کند وجود دارد باعث می‌شود عملا در هر زمان بیش از تعداد معینی تراکنش قابل پردازش نباشند به طور مثال در شبکه بیت کوین تقریبا در هر ثانیه فقط ۷ ترنزکشن قابلیت پردازش شدن را دارند که این باعث ایجاد محدودیت بسیاری در ساختار این شبکه و قابلیت های آن می‌شود. همچنین به دلیل محدودیت تعداد تراکنش ها افراد مجبور می‌شوند پیوسته هزینه پردازش تراکنش بیشتری پرداخت کنند تا استخراج کنندگان [1] رغبت بیشتری به پردازش ترکنش آن‌ها داشته باشند.

۳-۱۴-۱ هش گراف [2]

باتوجه به مشکل مقیاس پذیری بسیاری از فعالات حوزه بلاک چین تلاش هایی برای حل این مشکل کردند یکی از راه حل های پیشنهاد شده برای مشکل مقیاس پذیری ایجاد هش گراف ها به جای هش چین ها یا همان بلاک چین ها می‌باشد که در این حالت مشکل مقیاس پذیری تا حد زیادی حل می‌شود و تعداد هزاران تراکنش در ثانیه قابلیت پردازش شدن خواهند داشت همچنین امکان انجام پرداخت های با مقادیر بسیار کم نیز وجود خواهد داشت.



تصویر ۳-۶ : مقایسه هش گراف با بلاکچین

همانطور که در تصویر ۳-۶ مشاهده می‌شود در هش گراف به جای استفاده از قاعده طولانی ترین زنجیره سعی بر آن است تا بلاک های یتیم شده به صورتی مجدادا به شبکه بازگردانده شوند در این حالت به جای یک زنجیره یک گراف ایجاد می‌شود که ند های آن با هش به یکدیگر مرتبط می‌باشند.

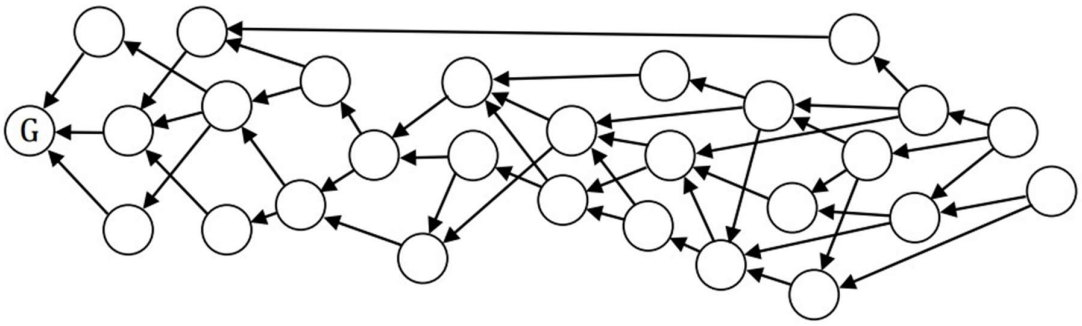
[1] Miners
[2] HashGraph

۳-۱۵ بلاک چین های نسل سوم [1]

بلاک چین های نسل سوم نوع دیگری از بلاک چین های هستند که در پاسخ به مشکل مقیاس پذیری بلاک چین ایجاد شده اند به طوری که با حل این مشکل امکان پردازش تعداد نامحدود تراکنش با هر مبلغ را فراهم می‌آورند. همچنین کاربرد دیگر آنها در موارد مرتبط به اینترنت اشیا [2] می‌باشد یکی از نمونه های موفق پیاده سازی بلاک چین های نسل سوم IOTA می‌باشد.

۳-۱۵-۱ نحوه کارکرد بلاک چین های نسل سوم

در این بلاک چین ها به جای ذخیره کردن تعدادی تراکنش در یک بلاک و سپس پشت هم قرار دادن بلاک ها پشت یکدیگر تلاش می‌شود تا تراکنش های بعدی به صورت مستقیم به تراکنش های قبل از خود متصل شوند، به طور دقیق تر هر تراکنش به ۲ تراکنش تصادفی از تراکنش های قبلی خود متصل می‌شود و بنابر این یک گراف از تراکنش ها ایجاد می‌شود که این گراف در حقیقت یک گراف جهت دار رو به جلو در زمان خواهد بود چرا که هر تراکنش فقط به تراکنش های قبل از خود در زمان می‌تواند متصل شود بنابر این یال های متصل کننده این تراکنش ها در گراف جهت دار هستند و همچنین به دلیل جهت دار بودن امکان ایجاد دور [3] نیز در این گراف موجود نمی‌باشد از این رو به آنها گراف های جهت دار بدون دور [4] ویا به اختصار DAG گفته می‌شود.

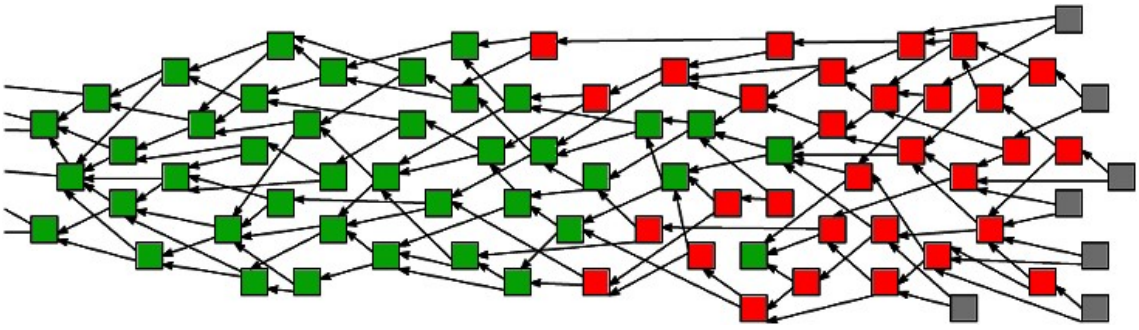


تصویر ۳-۷ : نمونه یک DAG

[1] Block Chain 3.0
[2] IOT
[3] Cycle
[4] Directed Acyclic Graph

۳-۱۵-۲ پردازش تراکنش ها در بلاک چین های نسل سوم

در این نوع بلاک چین ها در حقیقت هر فرد برای ارسال تراکنش خود به جای پرداخت هزینه تراکنش به استخراج کنندگان تا آن‌ها تراکنش فرد را تایید و پردازش و به شبکه اعلام کنند خود اقدام به اعلام کردن تراکنش خود می‌کند با این تفاوت که برای افزودن تراکنش خود به شبکه موظف است ۲ تراکنش قبل از خود را تایید کند و پس از تایید صحت آن ۲ تراکنش، تراکنش خود را به انتهای آن‌ها می‌افزاید به این صورت ساختار DAG گفته شده از اتصال هر تراکنش به ۲ تراکنش قبل از خود به وجود می‌آید به این ترتیب و به دلیل عدم وجود هزینه تراکنش تراکنش های با هر مقدار ناچیز قابلیت افزوده شدن به شبکه را دارا می‌باشند. همچنین به افزایش تعداد تراکنش ها مشکل ایجاد ترافیک برای تراکنش ها رخ نمی‌دهد چرا که افزایش تعداد تراکنش ها به معنی افزایش تعداد ند های فعال در شبکه است که خود موظف به تایید تراکنش های قبل از خود می‌باشند، بنابر این همواره با افزایش تعداد تراکنش ها تعداد تایید کنندگان تراکنش نیز افزایش می‌یابد و شبکه دچار کندی نمی‌شود.



تصویر ۳-۸ : وضعیت تراکنش ها در یک DAG

همانطور که در تصویر ۳-۸ نشان داده شده است تراکنش ها در این نوع شبکه ها به ۳ نوع مختلف تقسیم می‌شوند تراکنش های کاملا تایید شده تراکنش های در دست تایید و تراکنش های جدید که به ترتیب با رنگ های سبز قرمز و خاکستری در تصویر قابل مشاهده می‌باشند.

تراکنش های تایید شده تراکنش هایی اند که از هر تراکنش جدید شبکه پس از طی چند مرحله در گراف می‌توان به آن‌ها رسید، در حقیقت این بدان معنی است که این تراکنش ها سابقیه کاملی از تراکنش های قبلی را در خود دارند، تراکنش های تایید نشده که به رنگ قرمز هستند فقط به تعدادی از تراکنش های جدید متصل اند و تراکنش های جدید تراکنش هایی هستند که در انتظار افزوده شدن تراکنش جدید به انتهای خود و پردازش شدن توسط یک ند می‌باشند.

۳-۱۶ بلاک چین های نسل چهارم [1]

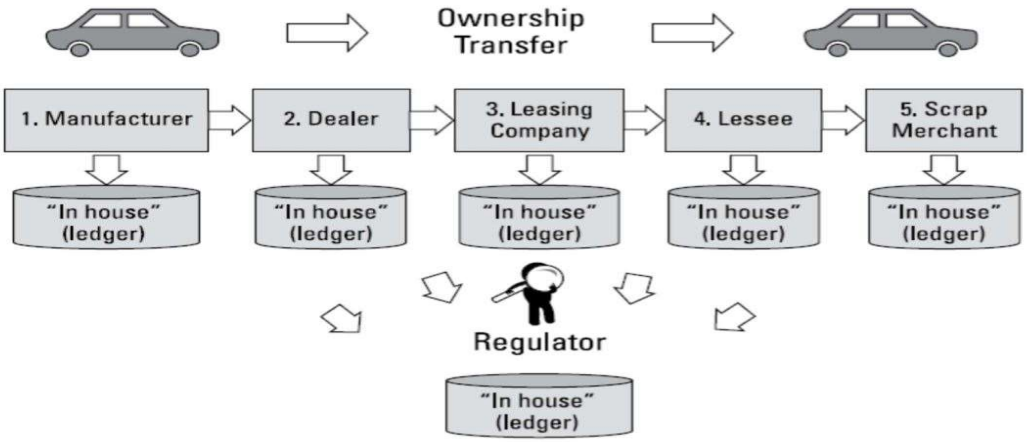
این نوع بلاک چین ها که بلاک چین های هوشمند نیز معروفیت دارند در حقیقت تلاشی برای ایجاد اتصال میان حوزه های هوش مصنوعی [2] و به خصوص یادگیری عمیق [3] با بلاک چین می‌باشند از پروژه های معروف بلاک چین های نسل چهارمی می‌توان به SingularityNet و Deep Brain Chain اشاره کرد.

۳-۱۷ بلاک چین در خارج از رمز ارزها

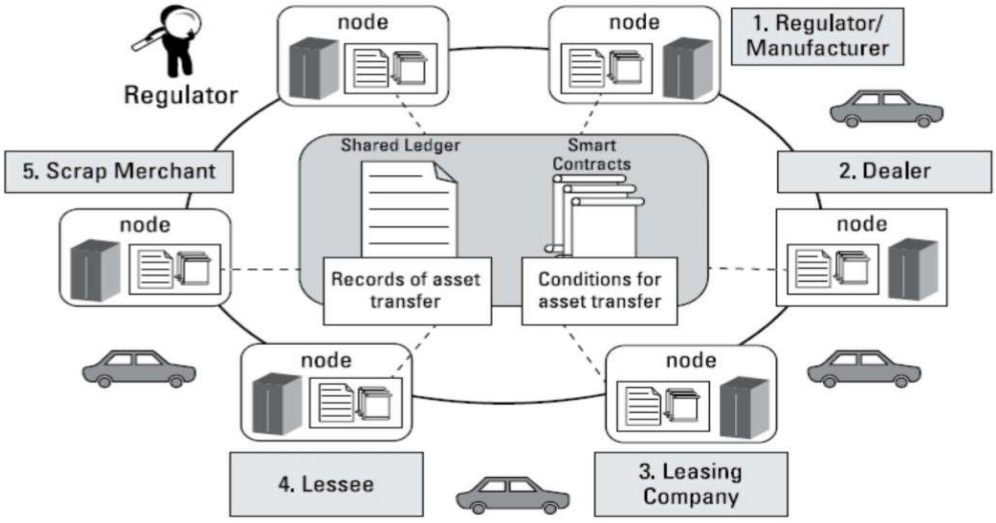
همانطور که پیشتر گفته شد بلاک چین فقط مختص به رمز ارزها نمی‌شود و قابلیت های این تکنولوژی انقلابی می‌تواند در بسیاری از صنایع دیگر نیز مورد استفاده قرار گیرد به طور مثال در اینجا به بررسی کاربرد بلاک چین در یک شبکه فروش اقساطی خودرو می‌پردازیم.

کارخانه‌های سازنده‌ی خودرو خرید قسطی خودرو را ساده جلوه میدهند، اما این کار در واقعیت میتواند کاملا پیچیده باشد. چالش مهمی که این روزها شبکه‌های خرید قسطی خودرو با آن روبهرو هستند این است که هرچند زنجیه تامین فیزیکی معمولا یک‌پارچه است ، اما هر مشارکت کننده در شبکه و سامانه های پشتیبان از هم جدا هستند که هرکدام از این‌ها باید دفتر کل مخصوص به خود را نگه داری کنند همچنین از طرفی نیز مراجع حقوقی که وظیفه ثبت مالکیت و سند را بر عهده دارند دفتر کل خود را دارند و پلیس نیز باید دفتر کل خود را برای شماره گذاری خودرو ها داشته باشد و از طرفی بانک ها نیز دفتر کل خود را برای پرداخت اقساط دارند با وجود این همه سازمان مختلف و دفتر کل های متفاوت هماهنگی میان این مراجعه بسیار سخت صورت می‌گیرد و گاها بسیار دردسر ساز است اما درصورت استفاده از بلاک چین بسیاری از مشکلات حل می‌شوند. به طوری که هریک از مشارکت کنندگان می‌توانند به راحتی بلاک چین را رسد کرده و اطلاعات مورد نظر خود را از آن برداشته و تراکنش های مورد نظر خود را بی افزایند.

[1] Block Chain 4.0
[2] artificial intelligence
[3] Deep Learning



تصویر ۳-۹ : رهگیری مالکیت خودرو بدون بلاک چین



تصویر ۳-۱۰ : رهگیری مالکیت خودرو با بلاک چین

۴ فصل چهارم : بررسی کارکرد چند رمز ارز مهم

باتوجه به این که شبکه های بلاک چین اغلب برای رمز ارز ها استفاده می‌شوند برای درک کامل مدل کارکرد آن‌ها لازم به نظر می‌رسد تا ابتدا به بررسی کارکرد چند رمز ارز مهم دنیای فناوری بپردازیم.

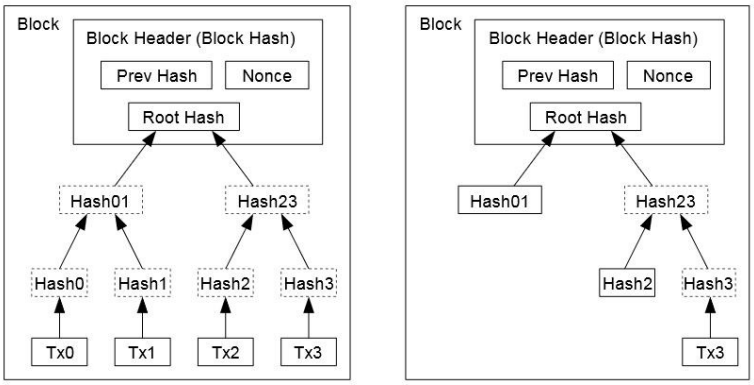
۴-۱ بیت کوین

بررسی تخصصی تر بیت کوین به عنوان اولین ارز رمز دنیا و همچنین به دلیل مشترک بودن الگوریتم های استفاده شده در آن در بسیاری از دیگر ارز رمز ها به عنوان مبنا هر ارز رمزی به نظر ضروری می‌رسد.

بیت کوین برای اولین بار در سال ۲۰۰۹ میلادی برای اولین بار توسط شخص یا اشخاص ناشناسی با نام ساتوشی ناکاماتو معرفی شد، معرفی این ارز رمز به صورت چند برگ مقاله با نام وایت پیپر [۱] معرفی شد در این مقاله به تشریح سازوکار این ارز رمز پرداخته شده است.

۴-۱-۱ ساختار بلاک چین بیت کوین

بلاک چین بیت کوین ساختاری مانند بلاک چین های فصل قبل دارد در اصل در اینجا باید به بررسی دقیق تر بخش داده ای هر بلاک بپردازیم. بخش داده ای هر بلاک بیت کوین حاوی اطلاعات تعدادی از تراکشن های انتقال این رمز ارز می‌باشد به طوری که حجم هر بلاک به ۱۰ کیلو بایت [۲] محدود می‌باشد بنابر این تعداد تراکشن های موجود در هر بلاک محدود می‌باشد. همچنین برای سرعت دادن به محاسبات POW به جای استفاده از کل اطلاعات ترنزکشن در محاسبه هش بلاک فقط از ریشه درخت مرکله تراکنش های بلاک استفاده می‌شود.



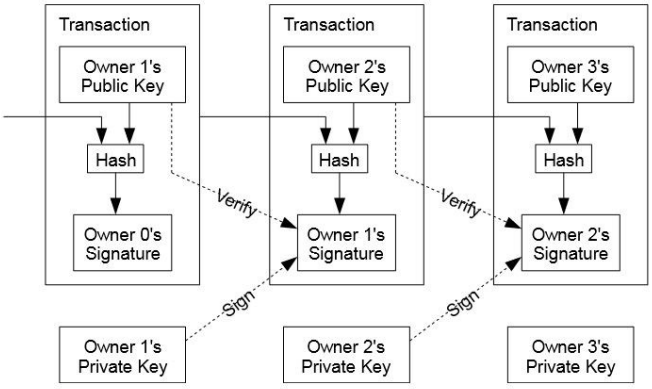
تصویر ۴-۱ : ساختار بلاک های بیت کوین

[1] White Paper
[2] Kilo byte

۴-۱-۲ تراکنش ها در بیت کوین

یکی از مهمترین اجزای تشکیل دهنده شبکه بیت کوین تراکنش ها هستند، در حقیقت خورد ترین بخش سازنده شبکه بیت کوین این تراکنش های می‌باشند که به کمک آن‌ها نقل و انتقالات بیت کوین در شبکه انجام می‌شود.

ساختار هر تراکنش در شبکه بیت کوین شامل اطلاعات کلید عمومی دریافت کننده تراکنش، مقدار تراکنش و امضای دیجیتال صادر کننده می‌باشد که این امضا دیجیتال در حقیقت به صورت یک قطعه کد برای ماشین مجازی بیت کوین به اسم Script Sig می‌باشد.



تصویر ۴-۲ : ساختار تراکنش های بیت کوین

۴-۱-۳ ساختار UTXO [1] در بیت کوین

در بیت کوین برای رهگیری مالکیت توکن های شبکه در طول زمان دارایی هر فرد به صورت یک عدد که با هر بار ارسال توکن مقداری از آن کم می‌شود و با هر بار دریافت به آن افزوده می‌شود درنظر گرفته نمی‌شود، بلکه در این رمز ارز به محض دریافت یک تراکنش توسط یک نفر خروجی های آن تراکنش به عنوان UTXO در کیف پول فرد ذخیره می‌شوند و هنگام ارسال بیت کوین فرد مورد نظر باید تعداد مشخصی از این UTXO ها را به حساب فرد مورد نظر ارسال کند.

۴-۱-۴ کیف پول بیت کوین

شبکه بیت کوین دارای تعداد مشخصی کیف پول می‌باشد که در حقیقت این کیف پول ها وظیفه حفظ و نگه داری

[1] Unspent Transaction Output

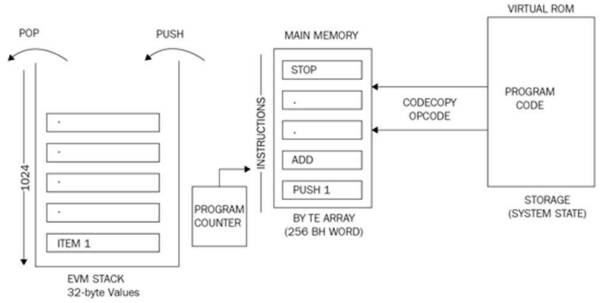
دارایی های هر فرد را دارند، در حقیقت در عمق هرکدام از این کیف پول‌ها کلید خصوصی هر فرد ذخیره می‌شود که با کمک آن و امضا می‌تواند UTXO هایی که به حساب کلید عمومی آن کلید خصوصی ارسال شده است را خرج کند، مهمترین کیف پول بیت کوین BitCoin Core می‌باشد که کیف پول رسمی این ارز رمز می‌باشد، این کیف پول در حقیقت نمونه کامل از پیاده سازی پروتوکل [1] بیت کوین می‌باشد این کیف پول برای کار کردن ابتدا لازم است تا نسخه کاملی از بلاک چین بیت کوین را که در حال حاظر تقریبا حجمی معادل 185GB دارد را دانلود نماید، یکی از قابلیت های این کیف پول امکان پیوستن به شبکه استخراج کنندگان توسط آن می‌باشد.

۴-۲ اتریوم

این ارز رمز نمونه موفق از پیاده سازی بلاک چین های نسل دوم می‌باشد که با کمک ماشین مجازی خود که EVM نام دارد، قادر است یک زبان کامل تورینگ را اجرا کند که به کمک کد های این ماشین شبکه اتریوم امکان اجرای کد هایی به پیچیدگی کد های همه دیگر زبان های برنامه نویسی به صورت توزیع شده روی همه کامپیوتر های اجرا کننده شبکه اتریوم می‌دهد.

۴-۲-۱ بررسی ساختار EVM

ماشین مجازی اتریوم و یا EVM در حقیقت یک ماشین مجازش مبتنی بر پشته [2] می‌باشد به این صورت که همه دستورات آن فقط مقداری را از پشته خوانده و مقداری را به پشته می‌افزایند، سایز این پشته به ازای هر داده برابر با 256 بیت[3] می‌باشد که در حقیقت برابر با خروجی تابع درهم ساز SHA3 (که به نام Keccak 256) نیز شناخته می‌شود می‌باشد همچنین در این ماشین مجازی یک حافظه دیگر نیز برای ذخیره سازی کد ها وجود دارد.



تصویر ۴-۳ : ساختار EVM

[1] Protocol
[2] Stack
[3] Bit

۲-۲-۴زبان Solidity

همانند دیگر زبان های اسمبلی برای زبان اسمبلی EVM نیز چندین زبان سطح بالاتر ایجاد شده است که کد های این زبان ها به کد های اسمبلی EVM ترجمه [۱] می‌شود. یکی از زبان های سطح بالاتر شبکه اتریوم زبان Solidity نام دارد که با کمک این زبان به راحتی می‌توان به نوشتن قرار داد های هوشمند پرداخت.

۳-۲-۴برنامه های توزیع شده [۲]

به برنامه های نوشته شده به زبان سالیدیتی در اصطلاح برنامه های توزیع شده گفته می‌شود چرا که این برنامه ها درحقیقت نه بر روی یک کامپیوتر بلکه برروی همه کامپیوتر هایی که در شبکه اتریوم فعالیت می‌کنند اجرا می‌شوند و هر ند در شبکه می‌تواند توابع موجود در این برنامه هارا فراخوانی و استفاده نماید. البته باید توجه داشت که مفهوم برنامه های توزیع شده مختص شبکه اتریوم نمی‌باشند و این مفهوم از مدت ها قبل نیز وجود داشته است اما یکی از بزرگترین مشکلات این برنامه ها بحث توافق می‌باشد که با کمک شبکه بلاک چین این مشکل حل می‌شود.

۴-۲-۴توکن های ERC20

زبان برنامه نویسی Solidity به قدری قوی می‌باشد که توسط آن افراد مختلف می‌توانند توکن های فرعی دیگر مورد نظر خود را ایجاد و برنامه نویسی کنند و به کمک بلاک چین شبکه اتریوم و کیف پول هایش از آنها استفاده نمایند به چنین توکن هایی ERC20 گفته می‌شود.

۵-۲-۴ساختار کیف پول های اتریوم

در ساختار شبکه اتریوم به جای مکانیزم UTXO استفاده شده در بیت کوین از مکانیزم ماشین حالت[۳] استفاده می‌شود به این ترتیب که هر کیف پول در این شبکه معادل یک حساب کاربری [۴] در نظر گرفته می‌شود که به یک کلید خصوصی متصل می‌باشد و مقدار موجودی نیز به صورت یک عدد به آن‌ها اختصاص داده می‌شود حال به ریشه درخت مرکله ساخته شده از اطلاعات همه این حساب های کاربری حالت سیسم گفته می‌شود که با پردازش هر تراکنش این سیستم از یک حالت به حالت دیگری می‌رود.

---

[1] Compile
[2] Dapps
[3] State Machine
[4] Account

۴-۲-۶ شبکه Ropsten

باتوجه به امکان برنامه نویسی برای شبکه اتریوم و نیاز این برنامه ها به کار کردن به اتریوم علاوه بر شبکه اصلی اتریوم که توکن های آن دارای ارزش می‌باشند، شبکه های دیگیری نیز برای آن ایجاد شده است که در حقیقت این شبکه ها برای تست و خطایابی قرار داد های هوشمند می‌باشند، ساختار این شبکه ها مانند شبکه اصلی اتریوم می‌باشد با این تفاوت که توکن های آن‌ها ارزش ندارند یکی از معروف ترین این شبکه ها Ropsten نام دارد.

۴-۲-۷ Web3.js

زبان Solidity علی رغم داشتن توانایی برای نوشتن هر نوع قرار داد هوشمند و برنامه ای قابلیت ایجاد رابط کاربری گرافیکی [1] را دارا نمی‌باشد در نتیجه کتابخانه های زیادی برای زبان های مختلف برنامه نویسی برای کار با آن ایجاد شده است، یکی از معروف ترین این کتابخانه ها که به زبان جاوا اسکریپت نوشته شده است Web3.js نام دارد که با کمک زبان جاوا اسکریپت و زبان های طراحی Html و Css امکان ایجاد رابط های کاربری تحت وب برای قرار داد های هوشمند را فراهم می‌آورد.

۴-۳ آیوتا

ایوتا یک نمونه از پیاده سازی بلاک چین های نسل سومی است که درسال ۲۰۱۶ میلادی پا به عرصه وجود نهاد این ارز رمز در حقیقت به منظور مناسب بودن برای اینترنت اشیا ایجاد شده است و از پرداخت های ریز به خوبی پشتیبانی می‌کند در این شبکه به DAG ایجاد شده از تراکنش های Tangle گفته می‌شود.

۴-۳-۱ توکن های IOTA

باتوجه به اینکه در IOTA ساختار استخراج وجود ندارد بنابر این همه توکن های شبکه IOTA در ابتدای ایجاد شدن آن به صورت آماده وجود داشتند و توسط یه عرضه عمومی در میان افراد مختلف پخش شده اند.

۴-۳-۲ نظیر یابی در IOTA

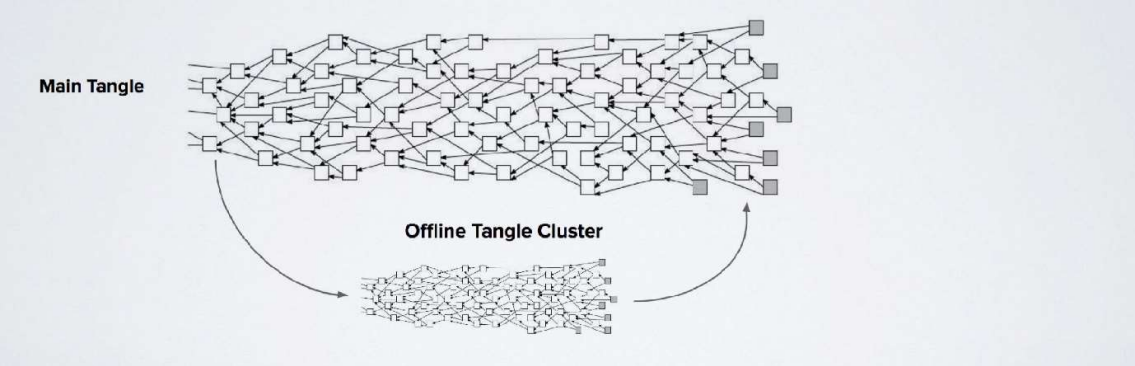
در شبکه IOTA عملیات نظیر یابی برخلاف بیت کوین و اتریوم نه به صورت خودکار بلکه به صورت دستی انجام می‌شود در حقیقت در این شبکه افراد برای متصل شدن به نظیر ها و اتصال به شبکه باید به سایت IOTA مراجعه کرده و از

[1] GUI

پایگاه داده موجود در سایت آدرس نظیر های مورد نظر خود را برداشته و برای اتصال به آن‌ها وارد نمایند، دلیل این ساختار آن است که باتوجه به اینکه در بلاک چین های نسل سوم تراکنش ها به تراکنش های دیگر متصل می‌شوند هر فرد باید به همسایه های مورد اعتماد متصل شود، همچنین باتوجه به ساختار بلاک چین های نسل سومی برخلاف دیگر بلاک چین که قدرت محاسباتی بیشتر به معنی کنترل بیشتر بر شبکه بود تعداد همسایه بیشتر به معنای کنترل بیشتر بر این شبکه است بنابر این با وجود این ساختار نظیر یابی امکان متصل شدن به تعداد زیادی از نظیر ها از بین می‌رود.

۴-۳-۳ امکان جدا شدن و پیوستن تعدادی از ند ها در IOTA

یکی از قابلیت های منحصر به فرد IOTA امکان جدا شدن تعدادی از ندها از شبکه اصلی و مجدادا پیوستن آن‌ها به شبکه می‌باشد در این حالت در مقطعی از زمان تعدادی از ند ها از شبکه جدا می‌شوند و به کار کردن بر روی یک Tangle کوچک تر در شعاع کاری خود می‌پردازند در نهایت به راحتی می‌توانند با متصل کردن تراکنش های شبکه اصلی به تراکنش های خود مجدادا به شبکه بپیوندند.



تصویر ۴-۴ : جدا شدن و اتصال مجدد

۵ فصل پنجم : پارسی کوین

پارسی کوین [1] در حقیقت نمونه یک پیاده سازی از بلاک چین های نسل سومی با زبان C# و با تکنولوژی .Net Standard می‌باشد با علامت اختصاری PIC برای توکن ها. که برای درهم سازی از تابع درهم ساز SHA-512 در آن استفاده شده است. که در ادامه به توضیح بخش های اساسی سامانه می‌پردازیم.

۵-۱ حالت سیستم

در این ارز رمز نیز به مانند ارز رمز اتریوم از ریشه درخت مرکله تمام حساب های کاربری سامانه حالت سیستم به دست می‌آید که با افزوده شدن هر تراکنش جدید سیستم از حالت n-1 به حالت n می‌رود. که این امر مستلزم به روز رسانی درخت مرکله می‌باشد تا ریشه جدید آن بدست آید، نکته قابل توجه آن است که به دلیل خاصیت درخت های دو دویی هر بار به روز رسانی این ریشه به تعداد لگاریتم حساب های کاربری نیاز به انجام عملیات دارد که به مراتب عدد کوچی می‌باشد به طور مثال در صورت وجود یک ملیوم حساب کاربری در سامانه به تعداد تقریبی ۲۰ عملیات برای به روز رسانی حالت سیستم نیاز خواهد بود.

۵-۲ حساب کاربری

هر حساب کاربری در پارسی کوین شامل موارد زیر می‌باشد.

۱. کلید خصوصی دارنده حساب
۲. کلید عمومی دارنده حساب
۳. دارایی حساب
۴. مقدار هش حساب

[1] ParsiCoin

به طوری که مقدار هش حساب از هش کردن موارد دیگر به دست می‌آید و این مقدار هش به عنوان برگهای درخت مرکله ای که ریشه آن حالت سیستم است استفاده می‌شود. حسابها در حقیقت وظیفه ایجاد تراکنش ها و امضای آن‌هارا نیز برعهده دارند.

```
public Transaction TransactionBuilder(string reciepient, double value, string message = "")
 {
    if (value > Balance) throw new Exception("Not enough funds.");
    var t = new Transaction(reciepient, value, signtureProvider, message);
    if (t.ISSigntureVerified()) return t;
    throw new Exception("Something went wrong, cannot sign the transaction");
 }
```

قطعه کد ۵-۱: تابع سازنده تراکنش در حساب

۵-۳ ساختار تراکنش های در پارسی کوین

ساختار تراکنش های در پارسی کوین شامل موراد زیر می‌باشد:

۱. کلید عمومی فرستنده/ایجاد کننده تراکنش
۲. کلید عمومی دریافت کننده تراکنش
۳. میزان تراکنش
۴. زمان ایجاد تراکنش
۵. زمان ثبت تراکنش در شبکه
۶. مقدار هش تراکنش
۷. پیام تراکنش به صورت اختیاری
۸. آدرس تراکنش در DAG
۹. امضا دیجیتال ایجاد کننده تراکنش
۱۰. قطعه کد های لازم برای تایید امضا دیجیتال تراکنش

برای ایجاد هش تراکنش کلید های عمومی فرستنده و دریافت کننده ، میزان تراکنش و زمان ایجاد تراکنش استفاده می‌شوند.

```
public string ComputeObjectHash()
        => $"{TransactionIssuer}-{Reciepient}-{Amount}-{IsuueTime}".ComputeHashString();
```

قطعه کد ۵-۲: محاسبه هش تراکنش

در این سیستم تراکنش ها پس از ایجاد و امضا شدن و قرار گرفتن در انتهای ۲ تراکنش قبلی به ند های همسایه ارسال می‌شوند تا توسط یک الگوریتم به مانند تئوری سخن چینی [1] در میان تمام ند های شبکه پخش شوند. همچنین هر ند پس از دریافت تراکنش ابتدا موظب به تایید هویت امضا آن است که توسط ماشین مجازی این سیستم انجام خواهد شد.

```
public Transaction(string reciepient, double amount, ECDSA ec, string message = "")
{
   TransactionIssuer = ec.ExportPubKey;
   Reciepient = reciepient;
   Amount = amount;
   IsuueTime = DateTime.UtcNow;
   TxHash = ComputeObjectHash();
   Signture = ec.Sign(TxHash);
   ScriptPubKey = $"{Signture};{TransactionIssuer}";
   ScriptSig = $"{ScriptPubKey};CheckSig;IsOne";
}
```

قطعه کد ۵-۳ : ایجاد تراکنش

۵-۴ ساختار ند های DAG در پارسی کوین

همانطور که گفته شد DAG در حقیقت یک گراف می‌باشد بنابر این، این گراف مانند دیگر گراف ها دارای ند هایی می‌باشد، این ندها به منظور نگه داری تراکنش ها ایجاد شده اند اما می‌توانند حاوی اطلاعات دیگیری نیز باشند. در ساختار پارسی کوین این امکان دیده شده است که ندها بتوانند حاوی هیچ تراکنشی نبوده و فقط دارای یک پیام خاص باشند. ساختار ند ها شامل موارد زیر است:

۱. نشانه ند
۲. زمان ایجاد شدن ند
۳. زمان تایید شدن ند
۴. ند قبلی سمت راست
۵. ند قبلی سمت چپ
۶. هش ند قبلی سمت راست
۷. هش ند قبلی سمت چپ

[1] Gossip

۸. پیام

۹. تراکنش

۱۰. هش تراکنش

۱۱. کلید عمومی ایجاد کننده ند

۱۲. هش ند

۱۳. حالت سیسم قبل از ایجاد این ند

۱۴. حالت سیستم بعد از ایجاد این ند

۱۵. مقدارد none برای مسئله POW ند

۱۶. تعداد تایید های ند

که مقدار هش ند در این سیستم برابر با هش شده نشانه ند به همراه زمان ایجاد آن و اطلاعات ند های قبلی و همچنین سازنده و تراکنش موجود در ند می‌باشد. همچنین حل یک مسئله POW نیز برای تایید و ایجاد هر ند درنظر گرفته شده است که این مسئله POW مقدار سختی بسیار کمتر از بلاک چین های نسل اول و دومی دارد به طوری که به طور معمول با یک کامپیوتر معمولی می‌توان در زمان تقریبی حدود ۳۰ ثانیه پاسخ آن‌را به دست آورد.

```
public string Mine()
   {
      byte[] s = null;
      do
      {
         s = ComputeObjectHash().ToByteArray(StringEncoding.Base85Check);
      } while (NodeHash.ToByteArray().CompareDiff());
      return s.ToBase58Check();
   }
```

قطعه کد ۵-۴ : تابع مسئله POW برای هر ند

۵-۵ کیف پول ها در پارسی کوین

هر کیف پول در پارسی کوین شامل تعداد نا محدودی حساب کاربری می‌شود که یکی از آن‌ها به عنوان حساب کاربری اصلی در هر زمان وظیفه ایجاد و امضا تراکنش ها را برعهده خواهد داشت.

۵-۶ نوسیه بندی ها در پارسی کوین

در این سامانه از ۴ نوع نویسه بندی برای انتقال متن ها استفاده شده است:

۱. نویسه بندی ASCII برای متن های انگلیسی
۲. نویسه بندی UTF-8 برای دیگر متن ها
۳. نویسه بندی Base64 برای انتقال متن های بی معنی در دیگر نویسه بندی ها در اجزا داخلی سیستم
۴. نویسه بندی Base58Check برای انتقال متن های بی معنی در دیگر نویسه بندی ها برای مواردی که لازم است توسط افراد خوانده شود مانند کلید های عمومی

تفاوت نویسه بندی Base58Check با نویسه بندی Base58 در قرار داشتن یک CheckSum در آن است.

۵-۷ انواع توابع درهم سازی در پارسی کوین

در این سامانه از ۳ تابع درهم سازی MD5 برای موارد معمولی SHA256 و SHA512 برای محاسبه POW استفاده شده است البته حالت اصلی ۲ بار محاسبه خروجی این توابع درهم سازی برای امنیت بیشتر می‌باشد.

۵-۸ رمز نگاری در پارسی کوین

در این ارزرمز از ۳ سیستم رمزنگاری AES وRSA وECC استفاده شده است که از ترکیب AES وRSA برای نقل و انتقال امن پیام ها استفاده می‌شود و از ECC نیز برای ساخت امضا دیجیتال برای ایجاد تراکنش ها. همچنین اطلاعات خصوصی نظیر کلید عمومی هر فرد که برروی سیستم فرد ذخیره می‌شود بر روی یک فایل که با AES رمزنگاری شده است نوشته می‌شوند تا درصورت به خطر افتادن رایانه فرد این اطلاعات قابل بازگشایی نباشند.

۵-۹ اجزا [1] سامانه

در این سامانه برای کنترل بهتر کد، کد برنامه به قسمت ها و کامپوننت هایی به شرح زیر تقسیم شده است.

۱. کامپوننت Base که کامپوننت پایه برنامه می‌باشد و شامل یک سری توابع اساسی که در همه بخش های

[1] Component

دیگر مورد استفاده قرار می‌گیرد می‌باشد.

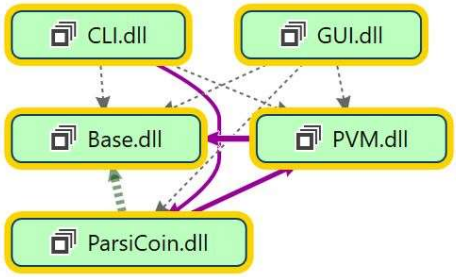
۲. کامپوننت اصلی ParsiCoin که شامل موارد کاری سامانه می‌باشد.

۳. کامپوننت PVM که شامل ماشین مجازی سیستم می‌باشد.

۴. کامپوننت CLI که شامل یک کامند لاین [1] برای کار با سامانه است

۵. کامپوننت GUI که شامل یک رابط کاربری گرافی [2]وب بیس برای سامانه می‌باشد.

ارتباط اجزا برنامه بایکدیگر به صورت زیر می‌باشد. که در ادامه به شرح کارکرد هریک خواهیم پرداخت.



تصویر ۵-۱ : ارتباط بین اجزا در پارسی کوین

۵-۹-۱ کامپوننت Base

این کامپوننت در اصل دربرگیرنده کلاس ها و توابعی است که در همه بخش های دیگر استفاده خواهند شد این توابع شامل تغییرات لازم برای کدبندی ها و محاسبات خروجی توابع درهم سازی می‌باشد.

این کامپوننت از کلاس هایی زیر تشکیل شده است:

۱. کلاس EncoderBase58 [3] که وظیفه کد بندی نویسه ها از حالت آرایه ای از بایت ها به Base58 را دارد.

۲. کلاس AES که شامل توابع مورد نیاز برای رمزنگاری AES در سیستم می‌باشد.

[1] Command Line Interface
[2] Graphic User Interface
[3] Class

۳. کلاس ECDSA که شامل توابع مورد نیاز برای امضا دیجیتال با کمک رمزنگاری ECC می‌باشد.
۴. کلاس های SecureLine که شامل توابع مورد نیاز برای رمزکردن پیام ها و انتقال آن‌ها در بستر اینترنت می‌باشد.
۵. کلاس استاتیک [۱] Util که شامل توابعی برای کد بندی و یا هش کردن می‌باشد.
۶. کلاس Configuration که شامل پیکر بندی های سیستم می‌باشد
۷. اینترفیس [۲] IPICObject که تمام کلاس های کامپوننت اصلی موظف به پیاده سازی آن‌می‌باشند.

که در ادامه به شرح کارکرد برخی از این کلاس‌ها و توابع موجود در آن‌ها خواهیم پرداخت.

۱-۱-۹-۵ کلاس AES

این کلاس که وظیفه انجام رمز نگاری AES را دارا می‌باشد دارای ۴ سازنده می‌باشد که کارکرد آن‌ها به شرح زیر است.

۱. سازنده اول که یک رشته حرفی [۳]از کاربر به عنوان رمز استفاده می‌کند.
۲. سازنده دوم که از GUID به عنوان رمز استفاده می‌کند.
۳. سازنده سوم که به مانند سازنده دوم بوده اما GUID گفته شده را خود تولید می‌کند.
۴. سازنده چهارم که وظیفه انجام دادن کارهای یکسان را داشته و در هر اجرای دیگر سازنده ها فراخوانی می‌شود.

همچنین شامل ۲ تابع برای رمزگذاری و ۲ تابع برای رمزگشایی می‌باشد که به ترتیب رشته های حرفی و یا رشته ای از بایت هارا دریافت می‌کنند.

در پیاده سازی اصلی این کلاس از شی System.Security.Cryptography.Aes خود چهارچوب .Net استفاده شده است که این از یک بردار IV و یک کلید برای رمزنگاری استفاده می‌کند که بردار و کلید گفته شده نیز توسط شی System.Security.Cryptography.Rfc2898DeriveBytes از رمز دریافت شده از کاربر توسط سازنده ها ایجاد می‌شود.

[1] Static
[2] Interface
[3] String

```
public class AES : IDisposable
    {
        private readonly byte[] _password;
        private readonly Aes _aes;
        private readonly byte[] _salt;
        private readonly int _iterationCount;
        public byte[] PassWord { get => _password; }
        public AES(string PassWord) : this(new object())
        {
            _password = Utilities.Util.ToByteArray(PassWord);

            Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(_password, _salt, _iterationCount);
            _aes.Key = pdb.GetBytes(32);
            _aes.IV = pdb.GetBytes(16);
            pdb.Dispose();
        }
        public AES(Guid PassWord) : this(new object())
        {
            _password = PassWord.ToByteArray();

            Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(_password, _salt, _iterationCount);
            _aes.Key = pdb.GetBytes(32);
            _aes.IV = pdb.GetBytes(16);
            pdb.Dispose();
        }
        public AES() : this(Guid.NewGuid())
        {
        }
        private AES(object obj)
        {
            _aes = Aes.Create();
            _salt = new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 };
            _iterationCount = 20_000;
        }
        public byte[] Encrypt(byte[] clearBytes)
        {
            byte[] res = null;
            using (MemoryStream ms = new MemoryStream())
            {
                using (CryptoStream cs = new CryptoStream(ms, _aes.CreateEncryptor(), CryptoStreamMode.Write))
                {
                    cs.Write(clearBytes, 0, clearBytes.Length);
                    cs.Close();
                }
                res = ms.ToArray();
            }
            return res;
        }
    public byte[] Decrypt(byte[] cipherBytes)
        {
            byte[] res = null;
            using (MemoryStream ms = new MemoryStream())
            {
                using (CryptoStream cs = new CryptoStream(ms, _aes.CreateDecryptor(), CryptoStreamMode.Write))
                {
                    cs.Write(cipherBytes, 0, cipherBytes.Length);
                    cs.Close();
                }
                res = ms.ToArray();
            }
            return res;
        }
         public byte[] Encrypt(string clearText, StringEncoding encoding = StringEncoding.UTF8)
        => Encrypt(clearText.ToByteArray(encoding));
        public byte[] Decrypt(string cipherText)
        => Decrypt(cipherText.ToByteArray(StringEncoding.Base64));

    }
```

قطعه کد ۵-۵: کلاس *AES*

۲-۹-۱-۵ کلاس Util

این کلاس شامل توابعی می‌باشد که مهمترین های آن‌ها وظایف مرتبط با محاسبه سختی را دارا می‌باشند، در ساختار پارسی کوین باتوجه به اینکه خروجی توابع هش ۶۴ بایت می‌باشد برای سختی ۲ مقدار در نظر گرفته شده است مقدار اول تعداد بایت های برابر با ۰ می‌باشد و مقدار دوم عدد یک بایت بعدی و بقیه بایت ها برابر با ۱ درنظر گرفته می‌شوند.

```
public static bool CompareDiff(this byte[] targer)
{
   for (int i = 0; i < targer.Length; i++)
   {
      if (targer[i] > Difficulty[i]) return false;
   }
   return true;
}
```

قطعه کد ۵-۶ : مقایسه سختی

۲-۹-۵ کامپوننت اصلی ParsiCoin

این کامپوننت درحقیقت شامل اجزایی از سامانه می‌باشند که وظیفه اجرای الگوریتم ها و پروتکل های اصلی سامانه را دارا می‌باشند. و شامل موارد زیر است:

۱. کلاس Account شمامل حساب های کاربری
۲. کلاس DAG شامل ساختار کلی DAG
۳. کلاس MerkleTree شامل درخت مرکله اکانت ها
۴. کلاس Node شامل اطلاعات ندها
۵. کلاس Services که شامل سرویس های مورد نیاز در سیستم می‌باشد.
۶. کلاس Transaction شامل اطلاعات تراکنش ها
۷. کلاس Wallet شامل اطلاعات کیف پول ها

۵-۹-۲-۱ کلاس سازنده درخت مرکله

این کلاس وظیفه ساخت درخت مرکله از اکانت هارا داری می‌باشد، ساختار داده موردنظر برای ساخت درخت یک ارایه از رشته های حرفی عددی می‌باشد همچنین در این کلاس یک آرایه دیگر نیز به تعداد اکانت ها از کلاس اکانت برای نگه داری اطلاعات کامل اکانت ها درنظر گرفته شده است.

```
public MerkleTree()
{
  Leafs = new MerkleNode[65536];
  Nodes = new string[131071];
  for (int i = 0; i < Leafs.Length; i++)
  {
    Leafs[i] = new MerkleNode();
    Leafs[i].HashString = Leafs[i].ComputeObjectHash();
    Nodes[i + Nodes.Length / 2] = Leafs[i].HashString;
  }
  var Start = Nodes.Length / 2;
  var End = Nodes.Length;
  while (true)
  {
    if (Start == 0) break;
    for (int i = Start; i < End; i += 2)
    {
      Nodes[(i) / 2] = $"{Nodes[i]}{Nodes[i + 1]}".ComputeHashString();
    }
    End = Start;
    Start /= 2;
  }

}
```

قطعه کد ۵-۷ : سازنده درخت مرکله

۵-۹-۲-۲ کلاس مرتبط با پایگاه داده

یکی از کلاس های این بخش کلاسی است که برنامه را به یک پایگاه داده ساده و بدون نیاز به سرور و NoSQL به نام LiteDB متصل می‌کند این پایگاه داده وظیفه ذخیره اطلاعات همه تراکنش ها و اکانت ها را در کامپیوتر هر دارنده سیستم دارا می‌باشد.

۵-۹-۳ کامپوننت PVM

این کامپوننت در حقیقت ماشین مجازی سیستم را دربر می‌گیرد که در ادامه در فصلی جداگانه به بررسی آن خواهیم پرداخت.

۵-۹-۴کامپوننت CLI

۵-۹-۵کامپوننت GUI

۵-۱۰بسته های نرم افزاری استفاده شده

در این سامانه از بسته های نرم افزاری زیر استفاده شده است:

۱. بسته نرم افزاری Json.net که برای سرالایز [۱] کردن اطلاعات از فرمت Json و به فرمت Json است.
۲. بسته LiteDB برای کار کردن با api های LiteDB
۳. بسته NbitCoin برای استفاده از توابع مرتبط با امضا دیجیتال ECC موجود در این بسته.
۴. بسته NetStandardLibrary به عنوان بسته اصلی دربرگیرنده چهارچوب دات نت

۵-۱۱سورس کنترل [۲]

نسخه کاملی از کد برنامه در به صورت بسته های سورس کنترل گیت [۳] در سایت گیت هاب و در مسیر https://github.com/FIVIL/ParsiCoin موجود می‌باشد.

[1] Serialize
[2] Source Control
[3] Git