مقاله پژوهشی

طراحی یک الگوریتم اجماع امن برای به‌کارگیری در بلاک‌چین

حسین بدری و معصومه صفخانی

چکیده: فناوری بلاک‌چین، شبکه را از لزوم وجود کارساز مرکزی بی‌نیاز می‌نماید. این فناوری از یک دفتر کل توزیع‌شده تشکیل گردیده که تمامی تراکنش‌های شبکه در آن ثبت می‌شود و شامل زنجیره‌ای از بلاک‌هاست. همه گره‌های شبکه، یک رونوشت از این دفتر کل را دارند. برای آنکه وضعیت این دفتر کل در هر لحظه از زمان برای تمام گره‌های شبکه یکسان باشد، به سازوکاری نیاز داریم که حصول توافق را برای کل شبکه فراهم کند که به آن «الگوریتم اجماع» می‌گویند. ما در این مقاله، یک الگوریتم اجماع جدید ارائه خواهیم نمود که در مقابل چهار حمله رایج بر بستر بلاک‌چین ایمن است. این حملات عبارت هستند از حمله سیبل، حمله منع خدمت، حمله ۵۱ درصد و حمله کسوف. با توجه به آنکه الگوریتم پیشنهادی ما دارای ویژگی‌هایی نظیر وجود پارامترهای کنترلی مختلف، ماهیت عمومی و همه‌منظوره، مقاوم‌بودن در برابر حملات مختلف و سرعت اجرای مناسب است، می‌توان از آن در پیاده‌سازی سامانه‌های امن مبتنی بر بلاک‌چین در حوزه‌های مختلف مانند اینترنت اشیا و سلامت الکترونیک استفاده نمود.

کلیدواژه: **بلاک‌چین، الگوریتم اجماع، امنیت بلاک‌چین، الگوریتم اثبات کار، الگوریتم اثبات سهام، الگوریتم تحمل خطای بیزانس.**

۱- مقدمه

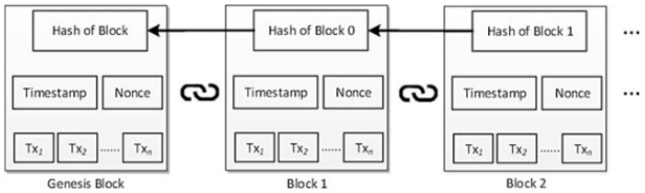
فناوری بلاک‌چین[1] از یک دفتر کل[2] توزیع‌شده تشکیل گردیده که تمامی تراکنش‌های شبکه در آن ثبت می‌شود. این دفتر کل شامل زنجیره‌ای از بلاک‌هاست که اتصال این بلاک‌ها به هم از طریق ارجاع هر بلاک به بلاک قبلی خود امکان‌پذیر است. این زنجیره، یک‌طرفه بوده و از طریق یک مقدار چکیده‌سازی‌شده[3]، بلاک فعلی را به بلاک قبلی مرتبط می‌کند. هر بلاک شامل رکوردها و تراکنش‌هایی است و تا زمانی که توسط کل شبکه مورد اجماع قرار نگیرد، به هیچ وجه نباید دچار تغییر شوند. قوانین خاصی بر روی شبکه حاکم است و به رایانه‌هایی که به این شبکه متصل هستند، گره[4] یا همتا[5] گفته می‌شود. در شکل ۱ طرحواره‌ای از این زنجیره را مشاهده می‌کنید و در ادامه به معرفی اجمالی ساختار بلاک‌چین خواهیم پرداخت.

این مقاله در تاریخ ۹ آذر ماه ۱۴۰۱ دریافت و در تاریخ ۲۰ اردیبهشت ماه ۱۴۰۲ بازنگری شد. این پژوهش با حمایت مالی دانشگاه تربیت دبیر شهید رجائی طبق ابلاغ گرنت شماره ۴۸۹۹ مورخ ۱۴۰۲/۰۳/۰۶ انجام گردیده است.

حسین بدری، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران، (email: hosein.badri@sru.ac.ir).

معصومه صفخانی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران، (email: safkhani@sru.ac.ir).

1. Blockchain
2. Ledger
3. Hash
4. Node
5. Peer



شکل ۱: زنجیره بلاک‌ها در بلاک‌چین [۱].

۱-۱ معرفی ساختار بلاک‌چین

پایگاه داده‌ای را در نظر بگیرید. به‌جای اینکه تمام داده‌های این پایگاه داده در یک کامپیوتر ذخیره شود، آنها را در چندین کامپیوتر ذخیره می‌کنیم. بلاک‌چین از چنین مفهومی استفاده می‌کند؛ با این تفاوت که به‌جای پایگاه داده از روش خاصی به نام دفتر کل استفاده می‌کند. این دفتر کل به‌صورت غیرمتمرکز و توزیع‌شده بر روی تمام گره‌های شبکه ذخیره می‌شود.

دو ویژگی اصلی شبکه بلاک‌چین، تمرکززدایی و غیرقابل تغییربودن آن است. در شبکه‌های سنتی که به شبکه‌های کارخواه- کارساز[6] معروف هستند، هر گرهی که بخواهد اطلاعاتی به‌دست آورد باید درخواست خود را به کارساز ارسال نموده و کارساز بر اساس این درخواست، اطلاعات لازم را ارائه کند. در چنین ساختار شبکه‌ای، کارساز باید همواره در دسترس بوده و به‌طور مداوم کار کند. چنانچه کسی به هر طریقی این کارساز را هک کند می‌تواند به اطلاعات حساس آن دسترسی پیدا کند. در مقابل، فناوری بلاک‌چین بر روی شبکه نظیربه‌نظیر[7] کار می‌کند که در آن، بار کاری شبکه در بین تمام گره‌ها پخش می‌شود و بدین طریق شبکه از وجود کارساز مرکزی بی‌نیاز می‌گردد. اگر یکی از گره‌ها دچار مشکل شود، می‌توان اطلاعات لازم را از گره‌های دیگر به‌دست آورد.

سه نوع اصلی بلاک‌چین وجود دارد که عبارت هستند از بلاک‌چین عمومی، خصوصی و ترکیبی.

بلاک‌چین عمومی: در این نوع بلاک‌چین فرض بر آن است که هر کس می‌تواند با یک کامپیوتر و دسترسی به اینترنت به این شبکه متصل شود و نیازی به واسطه ندارد. استفاده از این نوع بلاک‌چین در شبکه‌هایی که به توزیع‌پذیری کامل و تراکنش‌های غیرمتمرکز در ساختار خود نیاز دارند مناسب‌تر است. به‌طور مثال بیت‌کوین و اتریوم[8] از این نوع بلاک‌چین استفاده می‌کنند. چنین شبکه‌هایی بسیار کند هستند و برای به‌اجماع‌رسیدن و تأیید تراکنش‌ها منابع بسیاری را هدر می‌دهند؛ اما در مقابل بسیار امن هستند [۲].

بلاک‌چین خصوصی: این شکل از بلاک‌چین توسط شرکت‌ها ایجاد

6. Client-Server
7. Peer to Peer
8. Etherium

می‌شود و تمامی گره‌هایی که به این شبکه متصل می‌شوند از قبل شناخته‌شده و قابل اعتماد هستند. دسترسی به این سامانه محدود است و هر گونه دعوت یا اجازه‌ای باید توسط اعضای آن مورد تأیید قرار بگیرد. فرایند تأیید اعتبار نیز توسط گره‌هایی که توسط مدیریت مشخص شده‌اند، انجام می‌پذیرد. این نوع بلاک‌چین مزایای خاص خود را دارد که شامل انجام سریع‌تر تراکنش‌ها، امنیت بالاتر و وجود یک کنترل مرکزی بر شبکه بلاک‌چین است. بلاک‌چین خصوصی برای شرکت‌ها و مدل‌های دولتی بسیار مناسب است. به‌عنوان مثال یک دولت می‌تواند با راه‌اندازی یک بلاک‌چین مخصوص رأی‌گیری به میزان بسیار زیادی در هزینه‌ها صرفه‌جویی نموده و در عین حال از امنیت بالای بلاک‌چین و کنترل خود بر روی آن سود ببرد [۳]. بیشتر بلاک‌چین‌های خصوصی در خدمات مالی مورد استفاده قرار می‌گیرند که در این مورد می‌توان به شرکت‌های NASDAQ، JPMorgan، Bank of America و بسیاری از شرکت‌های دیگر اشاره نمود که بلاک‌چین اختصاصی خود را جایگزین تراکنش‌های کاغذی کرده‌اند [۴].

بلاک‌چین ترکیبی: این نوع بلاک‌چین از خصوصیات مثبت هر دو نوع قبلی سود می‌برد. این سامانه هم شامل یک بلاک‌چین عمومی هستند که همه می‌توانند در آن شرکت کنند و هم شامل یک بلاک‌چین خصوصی هستند که شرکت‌کردن در آن نیازمند تأیید اعتبار است. شرکت‌های بزرگ و دولت‌ها می‌توانند از مزایای بلاک‌چین ترکیبی بهره ببرند؛ به‌طوری که تصمیم بگیرند کدام داده‌ها خصوصی باقی بماند و کدام یک در دفتر کل عمومی ذخیره شود. تا کنون استفاده‌های واقعی از این طرح در شاخه‌هایی نظیر زنجیره تولید، هوانوردی، تجارت بین‌المللی و سازوکارهای مالی مورد استفاده قرار گرفته است [۵].

الگوریتم اجماع[1] به‌عنوان قلب تپنده یک سامانه بلاک‌چین در نظر گرفته می‌شود و امر مهم اعتماد متقابل را در شبکه‌ای که عدم اعتماد و ناشناس‌بودن از ویژگی‌های آن است، فراهم می‌آورد. یک الگوریتم اجماع، سازوکاری با توانایی تحمل خطاست که در سامانه‌های رایانه‌ای و بلاک‌چین مورد استفاده قرار می‌گیرد و وظیفه اصلی آن، ایجاد یک توافق جامع و ضروری بر روی یک داده واحد یا یک وضعیت خاص شبکه است؛ به‌طوری که تمام گره‌های متصل به آن شبکه بر روی آن داده یا وضعیت خاص توافق کنند [۶]. در بخش بعدی به معرفی الگوریتم‌های اجماع بنیادین خواهیم پرداخت.

۱-۲ الگوریتم‌های اجماع بنیادی

تا کنون الگوریتم‌های اجماع بسیار زیادی معرفی شده‌اند. تقریباً اصول اولیه تمام الگوریتم‌های اجماع اخیر بر پایه یک یا ترکیبی از سه الگوریتم اصلی است. این الگوریتم‌ها عبارتند از اثبات کار، اثبات سهام و تحمل خطای بیزانس و به‌دلیل اهمیتشان به معرفی اجمالی آنها خواهیم پرداخت.

۱-۲-۱ الگوریتم اثبات کار

این الگوریتم برای اولین بار به همراه بیت‌کوین، ارائه و پس از آن در اتریوم اولیه به‌کار گرفته شد. سازوکار آن جهت افزودن یک بلاک جدید به زنجیره بلاک‌ها، مبتنی بر حل محاسبات پیچیده‌ای است که به قدرت محاسباتی زیادی نیاز دارد. به این فرایند به اصطلاح «استخراج‌کردن»[2] و به گره‌هایی که در این فرایند شرکت می‌کنند، «استخراج‌گر»[3] گفته می‌شود. انگیزه اصلی استخراج‌گرها برای شرکت در این رقابت محاسباتی، مقدار رمزارز تشویقی است که در صورت تکمیل فرایند به تکمیل‌کننده محاسبات پرداخت می‌شود. علاوه بر آن استخراج‌گرها در صورت تکمیل هر تراکنش و پذیرش و اضافه‌شدن آن به زنجیره، یک مبلغ دیگر به‌عنوان کارمزد دریافت می‌کنند.

جهت افزودن یک بلاک جدید به شبکه بلاک‌چین، استخراج‌گرها باید یک مسئله ریاضی سخت را به روش سعی و خطا حل کنند. اولین استخراج‌گری که به پاسخ برسد، آن را به همراه بلاک به کل شبکه ارسال می‌کند. مشارکت‌کنندگان دیگر با استفاده از این پاسخ می‌توانند بلاک جدید را به بلاک‌های دیگر متصل کنند. یافتن پاسخ، فرایندی زمان‌بر و هزینه‌بر است؛ اما تأیید اعتبار آن بسیار آسان است. از همین رو بقیه گره‌ها می‌توانند به‌سرعت پاسخ را تأیید نموده و بلاک جدید را به زنجیره اضافه کنند [۷].

الگوریتم اثبات کار برای تأیید تراکنش‌ها به تأخیر زمانی زیادی دارد. از زمان ارسال یک تراکنش تا زمانی که شخص مطمئن شود تراکنش وی به‌طور غیرقابل بازگشتی ثبت شده است، تقریباً ۳۰ دقیقه (به اندازه تولید سه بلاک پشت‌سرهم) طول می‌کشد. این زمان در مقایسه با زمان انجام تراکنش‌های مالی واقعی کارت‌های اعتباری، بسیار طولانی است.

مشکل دیگر الگوریتم اثبات کار این است که مستعد حمله ۵۱ درصد می‌باشد (در بخش ۱-۳ انواع حملات معرفی شده‌اند)؛ شرایطی که در آن یک گره یا گروهی از گره‌ها بیش از ۵۰ درصد از قدرت محاسباتی شبکه را به‌دست گرفته و با تأییدکردن تراکنش‌های نامعتبر، توانایی تغییر کل بلاک‌های زنجیره را به‌دست می‌آورند.

۱-۲-۲ الگوریتم اثبات سهام

ایده اصلی این الگوریتم آن است که هرچه سهم بیشتری جهت تأیید اعتبار داشته باشید، به همان میزان شانس بیشتری جهت تأیید بلاک و کسب جایزه خواهید داشت. در الگوریتم اثبات سهام به گره‌های فعال، «اعتباردهنده» می‌گویند. اعتباردهنده‌ها با تأیید یک بلاک، کارمزد آن را دریافت می‌کنند. با چنین سازوکاری به قدرت محاسباتی بالا احتیاجی نیست و تمام سکه‌ها از همان ابتدا در دسترس هستند.

در عمل، گره‌ها جهت اعتبارسنجی به‌صورت تصادفی انتخاب می‌شوند. شانس انتخاب یک گره به تعداد سکه‌هایی که در حساب خود دارد وابسته است. برای این کار، یک گره باید مقدار دلخواهی سکه به‌عنوان سپرده برای دور بعدی اعتبارسنجی در نظر بگیرد.

یکی از مواردی که در اثبات سهام با آن مواجه هستیم، مشکل «سرمایه‌گذاری روی هیچ»[4] [۸] است. این مشکل زمانی اتفاق می‌افتد که که هیچ مشوقی برای رأی‌دادن به بلاک صحیح وجود نداشته باشد. در چنین شرایطی، یک گره خرابکار می‌تواند روی دو زنجیره سرمایه‌گذاری کند؛ بدون آنکه نگرانی بابت ازدست‌رفتن سهام خود داشته باشد. هر کدام از زنجیره‌ها مورد تأیید قرار بگیرند، گره مورد نظر سود کرده و برنده می‌شود و چیزی برای ازدست‌دادن ندارد. راه حل این مشکل آن است در صورتی که یک گره تلاش کند خارج از دستور عمل کند و نقصی در مراحل کار اثبات سهام ایجاد نماید، سریعاً توبیخ شده و تمام سهامش از بین برود. در نتیجه اعتباردهنده‌ها انگیزه خود را برای سرمایه‌گذاری در زنجیره نادرست و اعمال هر گونه خرابکاری از دست می‌دهند.

1. Consensus Algorithm
2. Mining
3. Miner
4. Nothing at Stake

جدول ۱: مقایسه امنیت در الگوریتم‌های اجماع اخیر.

ردیف	نام الگوریتم	نوع حمله			
		سیبل	منع خدمت	۵۱ درصد	کسوف
۱	[۱۱] Medical image sharing	×	×	×	×
۲	[۱۲] Fruitchains	✓	×	×	×
۳	[۱۳] Proof of Luck	×	×	×	×
۴	[۱۴] Cooperative Bargaining	×	×	×	×
۵	[۱۵] Condorcet	✓	×	×	×
۶	[۱۶] Robust Round Robin	×	×	✓	×
۷	[۱۷] Fantˆomette	✓	✓	×	×
۸	[۱۸] CloudPoS	×	×	✓	×
۹	[۱۹] Trust-CP	✓	×	×	×
۱۰	[۲۰] Weighted Voting	×	×	×	×
۱۱	[۲۱] DDPoS	×	×	×	×
۱۲	[۲۲] BIFTS	×	×	✓	×
۱۳	[۲۳] Implicit	×	×	✓	×
۱۴	[۲۴] FastBFT	×	×	×	×
۱۵	[۲۵] YAC	×	×	×	×
۱۶	[۲۶] Tendermint	×	×	✓	×
۱۷	[۲۷] Block-Supply	×	✓	×	✓
۱۸	[۲۸] Proof of Learning	×	✓	×	×
۱۹	[۲۹] Solida	✓	×	×	×
۲۰	[۳۰] Hybrid consensus	✓	×	×	×
۲۱	[۳۱] Panda	✓	✓	✓	×
۲۲	[۳۲] ISCP	×	×	×	×
۲۳	[۳۳] Proof of Vote	×	×	×	×
۲۴	[۳۴] Lightweight	×	×	×	×
۲۵	[۳۵] RDV	×	×	✓	×
۲۶	[۳۶] Proof of Disease	×	×	×	×
۲۷	[۳۷] Proof of Play	✓	×	✓	×
۲۸	[۳۸] PoRX	✓	×	✓	×

۱-۲-۳ الگوریتم تحمل خطای بیزانس

الگوریتم «تحمل خطای بیزانس» بر اساس [۹] توسعه یافته که اصل اساسی این الگوریتم «رسیدن به اجماع با رأی دوسوم مشارکت‌کنندگان» است. در هر سامانه‌ای که تحمل‌پذیر خطا[1] باشد، ممکن است پیام‌ها از دست بروند یا تحریف، دچار تأخیر و یا تکرار شوند. به‌علاوه، ترتیب ارسال پیام‌ها ممکن است با ترتیب دریافت آن در طرف دیگر یکی نباشد. در چنین سامانه‌ای، فعالیت گره‌ها نیز غیرقابل پیش‌بینی است. گره‌ها می‌توانند در هر زمانی به سامانه وارد یا از آن خارج شوند. در چنین محیط پویا و نامنظمی، الگوریتم تحمل خطای بیزانس می‌تواند با درصد تحمل خطایی در حدود یک‌سوم از کل گره‌ها، آنها را به توافق برساند که به آن معناست که حتی اگر یک‌سوم از کل گره‌های شبکه خرابکار باشند یا پاسخ ندهند یا به هر دلیلی در اجماع شرکت نکنند، باز عملکرد و ثبات سامانه تضمین می‌شود و در وضعیت پایدار باقی می‌ماند [۷].

به زبان ساده، بلاک‌چینی که از پروتکل اجماع تحمل خطای بیزانس استفاده کند به شرح زیر است: یک گره کارخواه، بلاک مورد نظر را به گره اصلی[2] ارسال می‌نماید و گره اصلی، بلاک را به چندین گره پشتیبان[3] می‌فرستد. اگر تعداد مشخصی از گره‌ها بلاک مورد نظر را تأیید کنند، آنگاه آن بلاک به زنجیره، افزوده شده و در غیر این صورت، بلاک حذف می‌گردد.

۱-۳ انواع حملات بر روی بلاک‌چین

امروزه به یک دلیل ساده، حمله به بسترهای بلاک‌چین بسیار جذاب شده و اینجا جایی است که پول وجود دارد [۱۰]. تا کنون حملات فراوانی برای این امر طراحی شده‌اند که نقاط ضعف الگوریتم را هدف قرار می‌دهند و موجب فروپاشی یک سامانه مبتنی بر بلاک‌چین می‌گردند. از این رو طراحی الگوریتم اجماعی که هم اشکالات الگوریتم‌های قبل را برطرف کند و هم از امنیت بالایی در برابر حملات مختلف برخوردار باشد، بسیار حائز اهمیت است.

ما در این بخش به‌طور خاص، چهار نوع حمله اساسی بر روی شبکه بلاک‌چین را به‌اختصار معرفی خواهیم نمود که عبارتند از حمله سیبل[4]، حمله منع خدمت (DoS)[5]، حمله ۵۱ درصد و حمله کسوف[6].

۱-۳-۱ حمله سیبل

در حمله سیبل، یک موجودیت تلاش دارد با ایجاد هویت‌های مختلف و کنترل گره‌های مختلف، بر شبکه نظیربه‌نظیر اثر گذارد. دشمن از طریق این حمله، حساب‌های کاربری جعلی مختلفی ایجاد می‌کند و با کنترل این حساب‌ها، قدرت رأی بیشتری در یک ساختار دموکراتیک کسب می‌نماید.

۱-۳-۲ حمله منع خدمت

حمله منع خدمت، نوعی حمله رایج است که باعث می‌شود کاربران نتوانند به یک کارساز خاص دسترسی داشته باشند. در حمله منع خدمت توزیع‌شده، حمله‌کننده به‌جای یک ماشین از چندین ماشین در شبکه استفاده می‌نماید تا به هدف خود به‌صورت هم‌زمان حمله کند.

۱-۳-۳ حمله ۵۱ درصد

یک گره خرابکار در این حمله، کنترل بیشتر از ۵۰ درصد نرخ چکیده‌سازی‌شده شبکه بلاک‌چین را به‌دست می‌گیرد و بدین صورت قادر است بلاک‌ها را دستکاری کند.

۱-۳-۴ حمله کسوف

حمله‌کننده در حمله کسوف، کنترل یک گره را به‌دست می‌گیرد و کاری می‌کند که اطلاعات ردوبدل‌شده گره قربانی فقط از طریق گره حمله‌کننده انجام شود. حمله‌کننده از این طریق می‌تواند سازوکار اجماع و فرایند استخراج را دستکاری کند.

۱-۴ نوآوری مقاله

در جدول ۱ که از [۳۹] استخراج گردیده، مقاومت ۲۸ الگوریتم اجماع اخیر در برابر چهار حمله اساسی بررسی شده است. با یک نگاه اجمالی به جدول درخواهیم یافت که هیچ کدام از الگوریتم‌های ارائه‌شده در برابر هر چهار حمله (بخش ۱-۳) ایمن نیستند. حتی مواردی وجود دارد که یک الگوریتم نسبت به هیچ کدام از حملات مقاوم نیست؛ هرچند که برای هر

1. Fault Tolerant
2. Primary
3. Backup
4. Sybil
5. Denial of Service
6. Eclipse

حمله حداقل یک الگوریتم ایمن در برابر آن حمله ارائه شده است. به طور مثال [۲۷] تنها الگوریتم امن جدول ۱ در برابر حمله کسوف است و یا [۳۱] در برابر سه حمله سیبل، منع خدمت و ۵۱ درصد امن است.

با درنظرگرفتن این موضوع که تا کنون هیچ الگوریتم اجماعی طراحی نشده که در مقابل هر چهار حمله سیبل، منع خدمت، ۵۱ درصد و کسوف امن باشد [۳۹]، متوجه شدیم که با ترکیب خصوصیات برخی از این الگوریتم‌ها می‌توان الگوریتمی پیشنهاد نمود که با ادغام ویژگی‌های مثبت الگوریتم‌های دیگر و برطرف‌نمودن نقاط ضعف آنها در مقابل هر چهار حمله مورد اشاره مقاوم باشد.

در بخش ۲، سه الگوریتم که توانسته‌اند در مقابل برخی از حملات بالا ایمن باشند معرفی خواهند شد که از ایده‌های آنها برای مقابله با هر یک از حملات در طراحی الگوریتم پیشنهادی استفاده خواهیم نمود. در بخش ۳، الگوریتم خود را ارائه داده و نشان می‌دهیم که الگوریتم پیشنهادی برخلاف الگوریتم‌های اجماع پیشین در مقابل هر چهار حمله سیبل، منع خدمت، ۵۱ درصد و کسوف امن است. در بخش ۴ به بررسی و ارزیابی الگوریتم پیشنهادی می‌پردازیم و نهایتاً در بخش ۵ کار را با نتیجه‌گیری به پایان می‌رسانیم.

۲- کارهای مرتبط

از آنجا که الگوریتم اجماع، نقشی حیاتی در هر سامانه مبتنی بر بلاک‌چین ایفا می‌کند، توجه بسیاری را در عرصه پژوهش علمی به خود جلب کرده و مقالات فراوانی، تمرکز خود را صرفاً روی بررسی و طراحی الگوریتم‌های اجماع مختلف معطوف کرده‌اند. ما در این بخش به معرفی تعدادی از این الگوریتم‌های منتخب می‌پردازیم که اخیراً ارائه شده‌اند و هر یک در برابر بعضی از چهار حمله سیبل، منع خدمت، ۵۱ درصد و کسوف امن هستند.

۲-۱ الگوریتم Block-Supply

در این قسمت، الگوریتم Block-Supply شرح داده می‌شود [۲۷]. این الگوریتم از آن جهت انتخاب گردیده که تنها الگوریتم موجود در جدول ۱ است که در برابر حمله کسوف ایمن می‌باشد. لذا برای طراحی الگوریتمی که بتواند در برابر هر چهار حمله ایمن باشد به بررسی و استفاده از راهکار ارائه‌شده در این الگوریتم نیاز داریم. در این الگوریتم، هر بار که یک بلاک جدید ارائه شود، مجموعه‌ای از logn گره به‌صورت تصادفی به‌عنوان اعتباردهنده‌ها[1] انتخاب می‌شوند. بدین طریق هم کار تأیید اعتبار به‌صورت یکسان در شبکه پخش می‌گردد و هم انتخاب عادلانه گره‌ها در نظر گرفته می‌شود.

گره مخرب برای اجرای یک حمله کسوف، نیاز به شناسایی گره اعتباردهنده دارد. از آنجا که انتخاب گره‌های اعتباردهنده به‌صورت تصادفی انجام می‌شود، کار شناسایی گره هدف بسیار پرهزینه بوده و عملاً شبکه را نسبت به حمله کسوف ایمن می‌سازد. همین موضوع در مواجهه با حمله منع خدمت نیز صادق است؛ چرا که در هر دور فرایند اجماع، هم گره انتخاب‌کننده و هم گره‌های اعتباردهنده به‌صورت تصادفی تغییر می‌کنند. گره مخرب برای ضربه‌زدن به شبکه باید تقریباً کل گره‌های شبکه را در دست گیرد که این موضوع در محیط توزیع‌شده‌ای که گره‌های آن به‌صورت ناشناس در حال فعالیت هستند، امری تقریباً غیرممکن و بسیار پرهزینه است.

1. Validator

۲-۲ الگوریتم Panda

الگوریتم Panda [۳۱] توانسته است که در مقابل حمله سیبل، منع خدمت و ۵۱ درصد به امنیت برسد؛ از این رو در این قسمت شرح داده می‌شود. در این الگوریتم، توسعه سامانه به دو مرحله اولیه و آزاد تقسیم می‌گردد. مدیر سامانه در مرحله اولیه، میزان ظرفیت و اعتبار گره‌های اولیه را که قرار است به سامانه ملحق شوند بررسی نموده و میزان مشخصی از واحد پولی سامانه (DLT) را به حساب اجماع هر گره اختصاص می‌دهد. این گره‌های اولیه می‌توانند شرکت‌های بزرگ، مؤسسات تحقیقاتی یا سازمان‌های آموزشی و درمانی مختلف باشند. پس از اختصاص DLT به این گره‌ها، سامانه وارد مرحله آزاد می‌شود که در آن، گره‌های جدید می‌توانند با خرید DLT لازم از حساب‌های اولیه یا سایر حساب‌ها به شبکه ملحق گردند.

باید توجه داشت که تنها فرستنده می‌تواند بلاک تراکنش را ایجاد کند؛ در نتیجه غیرممکن است که توسط شخص ثالث جعل شود. این وضعیت باعث می‌گردد که یک گره خرابکار بتواند چندین بلاک تراکنش با مقدار چکیده‌سازی‌شده قبلی (H_{pre}) یکسان در زنجیره خود بسازد. از آنجا که این بلاک‌های تراکنش باید به تمام شبکه ارسال شوند، گره دیگر متوجه وجود این انشعاب خواهد شد. در چنین وضعیتی با توجه به آنکه زنجیره یک‌طرفه می‌باشد، لازم است گره‌ها از بین انشعاب ایجادشده، تنها یک بلاک را انتخاب و به زنجیره اضافه کنند. اینجاست که الگوریتم اجماع اجرا می‌شود.

فرایند اجماع هم‌زمان با ایجاد حساب‌های اجماع، آغاز و این فرایند به دو مرحله رأی‌گیری[2] و تثبیت[3] تقسیم می‌شود. در مرحله رأی‌گیری، تمام حساب‌ها از بین بلاک‌های تراکنش موجود در انشعاب، یک بلاک را انتخاب نموده و به آن رأی می‌دهند. رأی‌ها به تمام گره‌های دیگر کمیته ارسال می‌شود. در مرحله تثبیت، گره‌ها بر اساس رأی‌های دریافتی شروع به تثبیت رأی نموده و آن را به بقیه ارسال می‌کنند. زمانی که تعداد رأی تثبیت دریافتی یک گره از یک مقدار آستانه عبور کند، اجماع در آن گره اتفاق افتاده است.

از آنجا که گره‌ها در الگوریتم Panda بر اساس سپرده‌ای که در حساب خود دارند قادر به ایجاد حساب اجماع و شرکت در فرایند اجماع هستند، در نتیجه صرفاً با ایجاد گره‌های بیشتر توسط گره خرابکار، امکانی جهت شرکت در اجماع و به‌دست‌آوردن حق رأی بیشتر فراهم نمی‌شود؛ در نتیجه حمله سیبل هیچ سودی برای او ندارد.

۲-۳ الگوریتم PoRX

الگوریتم PoRX [۳۸] به لحاظ امنیت در مقابل حمله سیبل و حمله ۵۱ درصد در این بخش شرح داده می‌شود.

در PoRX یک طرح ایجاد انگیزه به نام «اثبات اعتبار»[4] جهت کسب اعتبار برای گره‌ها ارائه شده که گره‌ها را تشویق می‌کند به‌صورت درست رفتار کنند. رفتار درست با پاداش و رفتار نادرست با تنبیه همراه می‌شود. مزیت اصلی این مدل، آن است که می‌تواند روی هر کدام از الگوریتم‌های PoX (اثبات کار، اثبات سهام[5] (PoS) و ...) که در حال حاضر کاربرد دارند، ساخته شود.

2. Vote
3. Commit
4. Proof of Reputation
5. Proof of Stake

وظیفه یک قرارداد ثبت هویت در پروتکل اجماع، این است که هویت گره‌ها را ثبت نموده و آن را با اطلاعات واقعی آنها تطبیق دهد. قرارداد ثبت‌نام باید اطمینان حاصل کند که هویت اعلام‌کننده، هم معتبر و هم منحصربه‌فرد است. از آنجا که فقط یک قرارداد ثبت‌نام وجود دارد، همه می‌توانند به منطق آن اعتماد کرده و از آن استفاده نمایند. وظایف ویژه یک قرارداد ثبت‌نام به شرح زیر است:

ثبت‌نام: این مرحله شامل ثبت یک حساب به‌عنوان استخراج‌گر است که پس از تأیید اعتبار می‌تواند حق تولید بلاک و ذخیره حساب خود در لیست سراسری حساب‌ها و دریافت لیست تمام حساب‌ها را داشته باشد.

خروج از حساب: وقتی که یک کاربر بخواهد از اجماع خارج شود، حساب او بسته می‌شود. یک حساب خارج‌شده به‌عنوان یک حساب نامعتبر در نظر گرفته شده و نمی‌تواند در اجماع شرکت کند و نیز حساب از لیست سراسری حساب‌ها حذف می‌شود.

تحلیل‌های صورت‌گرفته نشان می‌دهند که مقدار اعتبار گره‌هایی با محیط محاسباتی مشابه، شبیه به هم هستند. نرخ ایجاد بلاک و نرخ افزایش اعتبار تقریباً شبیه به هم است. اگر مقدار اعتبار گره‌هایی که محیط محاسباتی مشابه دارند متفاوت باشد، آنها وارد یک چرخه می‌شوند. نرخ تولید بلاک گرهی با درجه اعتبار بیشتر بالاتر خواهد بود؛ اما نه بیش از اندازه بالاتر؛ ولی اعتبار آن به‌آرامی افزایش پیدا می‌کند. اگر میزان اعتبار گرهی که اعتبار زیادی دارد به حد مشخصی افزایش یابد، نرخ تولید بلاک آن افزایش پیدا می‌کند؛ اما افزایش اعتبار آن سخت‌تر می‌شود. نرخ تولید بلاکی با میزان اعتبار کم کاهش می‌یابد؛ اما اعتبارش به‌صورت پیوسته افزایش می‌یابد. اگر میزان اعتبار دو گره با دو محیط محاسباتی متفاوت برابر باشد، گرهی با قدرت محاسباتی بیشتر، نرخ تولید بلاک بیشتری را از آن خود خواهد کرد؛ اما اعتبارش به‌آرامی افزایش می‌یابد و بالعکس، نرخ تولید بلاک گره ضعیف‌تر کاهش می‌یابد؛ اما میزان اعتبار آن با سرعت بیشتری افزایش می‌یابد تا فاصله بین قدرت محاسباتی آن دو جبران شود.

در واقع با این سازوکار به کاربرهایی با میزان قدرت محاسباتی کمتر، شانس بیشتری برای شرکت در اجماع داده می‌شود. در [۳۸] به‌جای طراحی یک پروتکل کامل، فقط یک واحد اعتبار طراحی شده است. پس میزان بهینگی و تمرکززدایی آن به پروتکل اجماع اصلی بستگی دارد و تغییر چندانی نخواهد کرد. هدف اصلی ارائه واحد اعتبار، تعیین کیفیت گره‌ها بر اساس رفتار آنهاست. حال به بررسی مقاومت این الگوریتم در برابر حمله سیبل و ۵۱ درصد می‌پردازیم.

حمله سیبل: در حمله سیبل یک مهاجم می‌تواند چندین حساب داشته باشد و فعالیت خرابکارانه را به‌صورت هم‌زمان با این حساب‌ها انجام دهد. تعداد کمی از الگوریتم‌های اجماع می‌توانند در مقابل چنین حمله‌ای مقاومت کنند؛ زیرا حساب‌های بلاک‌چین کاملاً ناشناس هستند و نمی‌توان هویت آنها را مشخص کرد. در این الگوریتم یک قرارداد ثبت هویت وجود دارد که تنها به حساب‌های ثبت‌شده امکان شرکت در اجماع را می‌دهد. این قرارداد تضمین می‌کند که یک شخص یا یک گره فقط می‌تواند یک حساب ثبت‌شده داشته باشد؛ مگر اینکه یک مهاجم به تعداد زیادی گره رشوه دهد یا اطلاعات آنها را سرقت کند و ثبت‌نام انجام دهد. در غیر این صورت این الگوریتم می‌تواند در مقابل حمله سیبل مقاومت کند.

حمله ۵۱ درصد: این الگوریتم یک چرخه رقابت معرفی می‌کند. در این چرخه میزان سختی تولید بلاک برای گرهی که تعداد زیادی بلاک تولید کرده است، به‌طور چشمگیری افزایش می‌یابد؛ بنابراین تعداد بلاک‌هایی که می‌تواند تولید کند، محدود می‌شود. معمولاً یک مهاجم نمی‌تواند تعدادی بلاک پشت سر هم ایجاد کند. اگر تعداد زیادی از بلاک‌ها در یک انشعاب توسط یک گره یا تعداد معدودی از گره‌ها تولید شود، تشخیص آن در زنجیره بسیار آسان است. از این رو انجام‌دادن یک حمله ۵۱ درصد موفق بر روی شبکه‌هایی که از الگوریتم PoRX به‌عنوان الگوریتم اجماع استفاده می‌کنند، تقریباً غیرممکن است.

۳- الگوریتم اجماع پیشنهادی

شبکه مبتنی بر الگوریتم پیشنهادی ما از مجموعه‌ای از گره‌ها تشکیل گردیده است که بر اساس وظیفه و سطح دسترسی به دو گروه کلی تقسیم می‌شوند.

گروه اول همانند الگوریتم پاندا شامل تعدادی گره‌های اولیه به‌عنوان گره‌های امن هستند که آنها را «کمیته راهبران» می‌نامیم. از آنجا که وظایفی نظیر انتخاب گره‌های شرکت‌کننده در اجماع، دریافت امضاها، تأیید یا عدم تأیید بلاک جدید و به‌روزرسانی اعتبار سایر گره‌ها بر عهده کمیته راهبران می‌باشد، ضروری است فرض نماییم که این گره‌ها وظایف خود را به شکلی صحیح و ایمن انجام می‌دهند.

گروه دوم شامل گره‌های عادی هستند که می‌توانند به‌صورت ناشناس به شبکه، اضافه یا از آن خارج شوند. در میان این گره‌ها، گره‌های مخرب یا گره‌های ناایمن نیز وجود دارند. وظایف اصلی این گره‌ها، ارائه بلاک جدید و شرکت در فرایند اجماع است. در ادامه مشخصات این گره‌ها را شرح خواهیم داد.

الگوریتم پیشنهادی به دو مرحله اولیه و اجرا تقسیم شده است. در مرحله اولیه به هر گره عادی شبکه به‌صورت ناشناس، یک گره راهبر از کمیته راهبران نسبت می‌دهیم. این انتخاب به‌صورتی است که گره مورد نظر، خود نمی‌داند که کدام گره راهبر به‌عنوان راهبر او انتخاب شده است؛ اما هر کدام از گره‌های راهبر می‌دانند که باید به کدام گره‌ها خدمت‌دهی کنند. فرایند انتخاب گره‌های راهبر و نسبت‌دهی آنها به گره‌های شبکه در هر دور اجماع و به‌صورت تصادفی تکرار می‌شود.

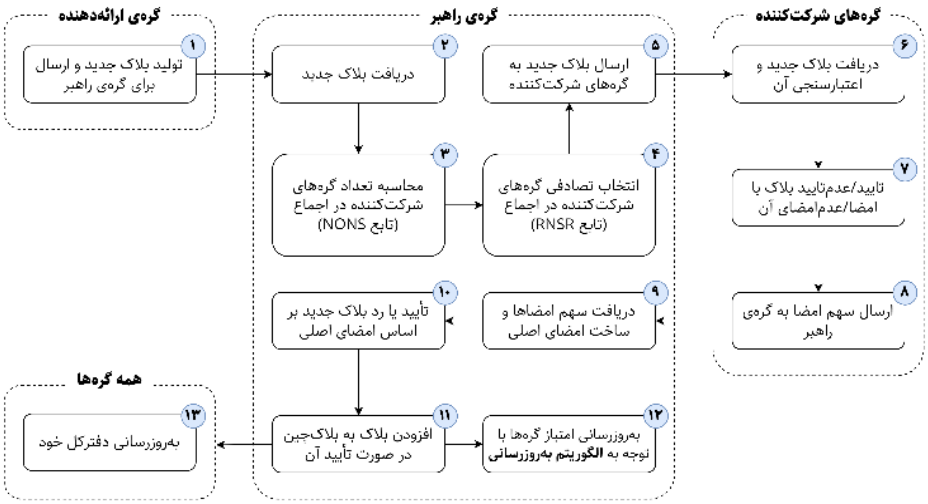
سایر مراحل الگوریتم پیشنهادی در مرحله اجرا قرار دارد و هر مرحله توسط گره یا گروهی از گره‌های خاص انجام می‌پذیرد. در شکل ۲ نمایی کلی از مراحل این الگوریتم را مشاهده می‌نمایید. در ادامه، هر مرحله با جزئیات شرح داده خواهد شد.

۳-۱ ارائه بلاک جدید

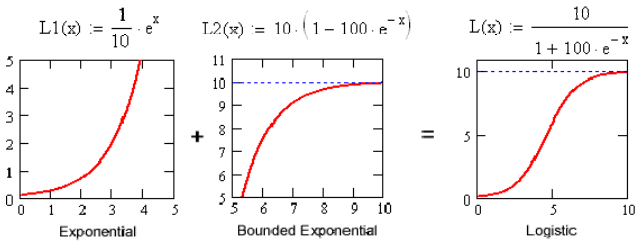
هر گره عادی که آن را «گره ارائه‌دهنده» می‌نامیم، می‌تواند بسته به شرایط خاص، یک بلاک جدید ارائه کند. این بلاک می‌تواند شامل لیستی از تراکنش‌ها، مرحله‌ای از زنجیره تأمین کالا، به‌روزرسانی وضعیت وسیله نقلیه در اینترنت اشیا، به‌روزرسانی پرونده پزشکی بیمار و یا هر مجموعه داده دیگر باشد. گره ارائه‌دهنده، بلاک جدید را به کمیته راهبران ارسال می‌کند. توجه داشته باشید که گره ارائه‌دهنده، گره راهبر متناظر خود را نمی‌شناسد.

۳-۲ آماده‌سازی فرایند اجماع

با دریافت بلاک جدید توسط کمیته راهبران، گره راهبر متناظر با گره ارائه‌دهنده، مراحل زیر را جهت آماده‌سازی فرایند اجماع انجام می‌دهد.



شکل ۲: طرحواره الگوریتم پیشنهادی.



شکل ۳: تابع لجستیک [۴۰].

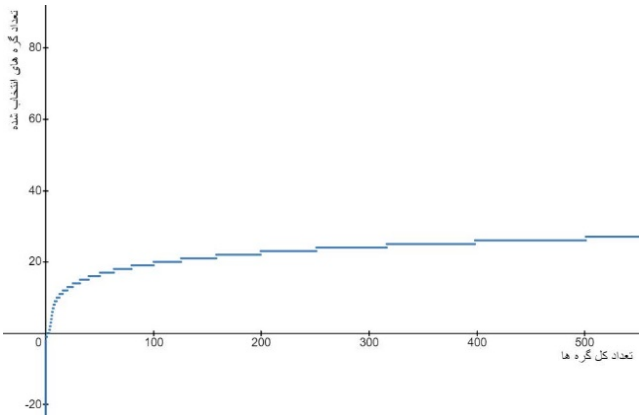
۳-۲-۱ محاسبه تعداد گره‌های شرکت‌کننده در اجماع

ابتدا در این بخش به معرفی تابع لجستیک[1] می‌پردازیم و سپس نحوه محاسبه تعداد گره‌های شرکت‌کننده در اجماع را شرح می‌دهیم.

همان طور که در شکل ۳ مشاهده می‌شود، یک تابع لجستیک ترکیبی از تابع نمایی و تابع نمایی محدود است. در تابع نمایی با افزایش ورودی، خروجی نیز به‌صورت نمایی افزایش می‌یابد. در مقابل در تابع نمایی محدود با توجه به ضریب محدوده با افزایش مقدار ورودی، میزان خروجی به همان نسبت افزایش می‌یابد تا زمانی که به مقدار محدوده برسد. تابع لجستیک دو تابع قبل را با هم ترکیب می‌کند. بر این اساس زمانی که وروی تابع کوچک است، میزان خروجی را به‌صورت نمایی افزایش می‌دهد و زمانی که ورودی افزایش پیدا می‌کند با کاستن آهنگ افزایش خروجی، از عبور آن از محدوده مشخص‌شده جلوگیری می‌نماید. در شکل ۳، تابع لجستیک استفاده‌شده در الگوریتم اجماع پیشنهادی آمده است.

حال باید بر اساس تابع لجستیک نشان‌داده‌شده در شکل ۳، رابطه‌ای را توسعه داد که علاوه بر داشتن خصوصیات بالا بتواند تعداد مناسبی از گره‌ها را بر اساس تعداد کل گره‌ها انتخاب کند. از این رو یک ضریب logn به تابع اضافه می‌نماییم تا همانند الگوریتم Block-Supply تعداد گره‌های انتخابی در محدوده logn گره باشد و از آنجا که به یک عدد صحیح (تعداد گره‌ها) نیاز داریم، جزء صحیح مقدار حاصل را در نظر می‌گیریم (رابطه (۱)). شکل ۴ نمودار خروجی تابع (۱) را نشان می‌دهد.

با بررسی خروجی تابع برای ورودی‌های کمتر از ده متوجه می‌شویم که این مقادیر با مقادیر مورد انتظار که حداکثر $n-۱$ است تفاوت دارد؛ زیرا برای تعداد کم گره‌ها می‌خواهیم تمام گره‌ها به‌جز گره ارائه‌دهنده در اجماع شرکت نمایند. لذا به‌سادگی، خروجی تابع را برای مقادیر کمتر از ده برابر با $n-۱$ در نظر می‌گیریم.



شکل ۴: نمودار خروجی تابع لجستیک جایگزین در الگوریتم اجماع پیشنهادی.

گره راهبر از (۱) که آن را تابع لجستیک nons[2] می‌نامیم برای مشخص‌نمودن تعداد گره‌هایی که باید به‌صورت تصادفی جهت اجماع انتخاب شوند، استفاده می‌کند. در این رابطه n نشان‌دهنده تعداد کل گره‌های شبکه است

$$nons(n)=\begin{cases} n-۱ & \text{if } n<۱۰ \\ \left\lfloor (\log n)\dfrac{۱۰}{۱+۱۰۰\times e^{-n}} \right\rfloor & \text{if } n\geq ۱۰ \end{cases} \qquad (۱)$$

لازم به ذکر است که تابع تعریف‌شده در (۱) به‌صورتی طراحی گردیده تا اطمینان حاصل کند زمانی که تعداد کل گره‌ها کم است، به تعداد کافی (تا حد امکان زیاد برای بالابردن امنیت و تحمل خطای الگوریتم) گره انتخاب کند و زمانی که تعداد کل گره‌ها زیاد می‌شود، به تعداد مناسب (تا حد امکان کم که کارایی الگوریتم را بهبود بخشیده و ردوبدل‌شدن پیام‌ها کم شود؛ اما نه آن قدر کم که امنیت و تحمل خطای الگوریتم را به خطر اندازد) گره انتخاب نماید. در جدول ۲ خروجی تابع (۱) را برای تعداد مختلف گره‌ها مشاهده می‌کنید.

۳-۲-۲ انتخاب تصادفی گره‌های شرکت‌کننده در اجماع

حال با محاسبه تعداد گره‌های شرکت‌کننده در اجماع، باید سازوکاری طراحی کنیم که گره‌ها را به‌صورت تصادفی انتخاب کند. از آنجا که الگوریتم پیشنهادی باید در برابر حمله ۵۱ درصد ایمن باشد، انتخاب تصادفی گره‌ها باید این اطمینان را حاصل کند که هم تمام گره‌های

1. Logistic

2. Number of Node Selection

مولفه‌های هر گره	
احتمال انتخاب گره در اجماع	**RSP**
اعتبار گره	**R**
میزان مشارکت گره در گلیکو	**RD**

شکل ۵: مؤلفه‌های هر گره برای انتخاب در دور بعدی اجماع.

جدول ۲: تعداد گره‌های انتخاب‌شده بر اساس تابع (۱).

تعداد کل گره‌ها	تعداد گره‌های انتخاب‌شده	تعداد کل گره‌ها	تعداد گره‌های انتخاب‌شده
۴	۳	۵۰	۱۶
۶	۵	۱۰۰	۲۰
۱۰	۹	۲۵۰	۲۳
۱۱	۱۰	۵۰۰	۲۶
۱۲	۱۰	۱۰۰۰	۳۰
۱۳	۱۱	۱۰۰۰۰	۴۰
۱۶	۱۲	۱۰۰۰۰۰	۵۰
۲۰	۱۳	۱۰۰۰۰۰۰	۶۰

شبکه، شانس برابری برای شرکت در اجماع داشته باشند و هم با افزایش تعداد دفعاتی که یک گره انتخاب می‌شود، شانس او برای انتخاب‌های بعدی، کمتر و کمتر گردد تا به نوعی از ایجاد تمرکز در سامانه جلوگیری کنیم و عدالت در انتخاب رعایت شود. بدین منظور برای هر گره، سه مؤلفه معرفی می‌کنیم تا بر اساس آنها بتوانیم انتخاب گره‌ها در دور بعدی اجماع را مدیریت کنیم. در شکل ۵ مؤلفه‌های اصلی هر گره نشان داده شده است.

RSP[1]: این مؤلفه، احتمال انتخاب گره در اجماع را نشان می‌دهد. هرچه میزان RSP یک گره بیشتر باشد با احتمال بیشتری در دور بعدی انتخاب می‌شود. این پارامتر در ابتدا برای همه گره‌ها یکسان بوده و برابر با $RSP_{DEFAULT}$ می‌باشد و به آن معناست که همه گره‌ها از شانس برابری برای انتخاب در دور بعدی اجماع برخوردارند. نحوه تغییر RSP در هر دور اجماع به عوامل مختلفی بستگی دارد که در بخش‌های بعدی به آن خواهیم پرداخت.

R[2]: این مؤلفه نشان‌دهنده اعتبار یا امتیاز[3] گره است و بر اساس نحوه رفتار گره و از طریق الگوریتم گلیکو[4] تنظیم می‌شود (الگوریتم گلیکو در بخش‌های بعدی معرفی خواهد شد). هرچه R گره بیشتر باشد به این معناست که گره دارای اعتبار بیشتری بوده و در دورهای قبلی، درست‌تر و قابل اعتمادتر رفتار کرده است.

RD: این مؤلفه نیز میزان مشارکت گره را نشان می‌دهد و برای محاسبه اعتبار گره‌ها در الگوریتم گلیکو استفاده می‌شود. این انتخاب تصادفی، شبکه را در برابر حمله کسوف و حمله منع خدمت ایمن می‌سازد؛ زیرا گره‌هایی که در دور بعدی اجماع باید شرکت کنند تصادفی و ناشناس هستند و از این رو عملاً حمله کسوف و منع خدمت را غیرقابل توجیه و بی‌فایده می‌سازد.

1. Random Selection Probability
2. Reputation
3. Rating
4. Glicko

Algorithm 1: RNSR: Random Node Selection based on Reputation

```
Input: Nods, nons
Output: Returns selected nods
selectedNodes ← [];
i ← 0;
while i <nons do
    selectedNode ← getRandomWeightedElement(nodes);
    if selectedNode is Not in selectedNodes then
        selectedNodes[i] ← selectedNode;
        i++;
    end
end
return selectedNodes
```

شکل ۶: الگوریتم RNSR.

Algorithm 2: getRandomWeightedElement

```
Input: Nodes
Output: a random node
threshold ← 20;
weightedValues ← [];
foreach nodes as node do
    if RSP_node − RSP_MIN < threshold then
        weightedValues[] ← RSP_node + R_node;
    else
        weightedValues[] ← 1;
    end
end
rand ← a random number from 1 to sum of elements in weightedValues;
foreach weightedValues as key => value do
    rand ← rand − value;
    if rand ⩽ 0 then
        return key
    end
end
```

شکل ۷: الگوریتم getRandomWeightedElement.

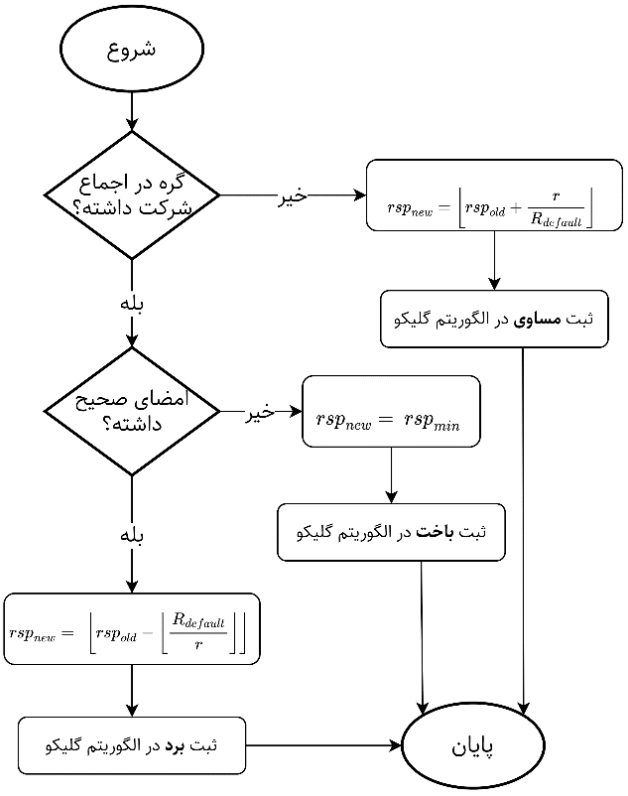
۳-۲-۳ الگوریتم انتخاب تصادفی گره‌ها (RNSR)

الگوریتم انتخاب تصادفی گره‌ها (RNSR)[5] در شکل ۶ نمایش داده شده است. این الگوریتم اطمینان حاصل می‌نماید که هر گره فقط یک بار انتخاب گردد. عمل اصلی انتخاب گره با درنظرگرفتن مؤلفه‌های R و RSP گره، توسط زیرتابع getRandomWeightedElement (شکل ۷) انجام می‌پذیرد. این زیرتابع از [۴۱] استخراج گردیده و تغییراتی در آن به انجام رسیده است. در صورتی که اختلاف RSP گره و RSP_{min} از حد آستانه کمتر باشد، وزن گره یک در نظر گرفته می‌شود و در غیر این صورت، وزن گره برابر با حاصل‌ضرب RSP گره و R گره می‌شود. این زیرتابع بر اساس وزن هر گره، شانس گره را مشخص نموده است و یک گره را به‌صورت تصادفی انتخاب می‌کند. در شکل‌های ۶ و ۷ به‌ترتیب الگوریتم RNSR و getRandomWeightedElement نشان داده شده است.

۳-۳ گره‌های شرکت‌کننده و بلاک جدید

پس از اینکه مجموعه گره‌های شرکت‌کننده در فرایند اجماع انتخاب شد، گره راهبر بلاک جدید را به همه گره‌های این مجموعه ارسال می‌کند. گره‌های شرکت‌کننده با دریافت بلاک جدید، ابتدا امضای گره راهبر را بر روی آن تأیید نموده و سپس محتویات بلاک را اعتبارسنجی می‌کنند. اگر بلاک مورد تأیید قرار گرفت، آن را امضا نموده و برای گره راهبر ارسال می‌کنند و در غیر این صورت، گره کاری انجام نمی‌دهد.

5. Random Node Selection Based on Reputation



شکل ۸: الگوریتم به‌روزرسانی امتیاز گره‌ها در الگوریتم اجماع پیشنهادی.

۳-٤ گره راهبر و بلاک جدید

گره راهبر در این مرحله، امضاهای گره‌های شرکت‌کننده در اجماع را بر روی بلاک جدید، دریافت و هر یک را اعتبارسنجی می‌نماید. چنانچه تعداد امضاهای معتبر از دوسوم تعداد گره‌های مجموعه شرکت‌کننده بیشتر شود، بلاک جدید مورد تأیید قرار گرفته و توسط گره راهبر به دفتر کل افزوده می‌شود. سپس بلاک جدید برای تمام گره‌های شبکه ارسال می‌گردد تا آنها نیز گره را به دفتر کل بلاک‌چین خود اضافه نمایند.

۳-۵ الگوریتم به‌روزرسانی امتیازها

ابتدا در این بخش، الگوریتم گلیکو را معرفی نموده و سپس نحوه به‌روزرسانی امتیازها را شرح می‌دهیم.

۳-۵-۱ الگوریتم گلیکو

الگوریتم گلیکو یکی از مهم‌ترین و پراستفاده‌ترین الگوریتم‌ها برای محاسبه امتیازات بازیکنان شطرنج در سرویس‌های برخط است. پارامتر اصلی این الگوریتم (R) برای محاسبه امتیاز بازیکنان مورد استفاده قرار می‌گیرد. این الگوریتم به‌صورتی عمل می‌کند که اگر بازیکنی با امتیاز بیشتر در برابر بازیکنی با امتیاز کمتر پیروز شود، امتیاز آن به میزان کمی افزایش می‌یابد و اگر بازنده شود، امتیاز آن به میزان زیادی کاهش می‌یابد. در مقابل اگر بازیکنی با امتیاز کمتر در برابر بازیکنی با امتیاز بیشتر پیروز شود، امتیاز آن به میزان زیادی افزایش می‌یابد و اگر بازنده شود امتیاز آن به میزان کمی کاهش می‌یابد. بدین طریق هرچه امتیاز بازیکنی بیشتر باشد، کسب امتیازهای بیشتر برای او سخت‌تر می‌شود و اگر در امتیازهای بالا اشتباه نموده و شکست بخورد، امتیاز او بیشتر کم می‌شود. این روش به نوعی میزان اعتبار بازیکن را با تخمین دقیقی محاسبه می‌کند و در بازی‌هایی با شرایط مختلف، پیشرفت و پسرفت بازیکنان را کنترل می‌نماید.

۳-۵-۲ محاسبه امتیازات گره‌ها

الگوریتم به‌روزرسانی امتیازات گره‌ها سه بخش دارد و مراحل آن پس از پایان اجماع و توسط گره راهبر انجام می‌پذیرد. مراحل اجرای الگوریتم در شکل ۸ آمده است. این الگوریتم برای تک‌تک گره‌ها، اجرا و امتیاز (R، RD و RSP) گره‌ها بر اساس یکی از سه حالت زیر به‌روزرسانی می‌شود.

حالت اول: گره برای اجماع انتخاب شده و عملکرد صحیح داشته است.

عملکرد صحیح می‌تواند به دو صورت اتفاق بیفتد. اول اینکه گره، بلاک را امضا نماید و نهایتاً آن گره در اجماع مورد تأیید قرار گیرد. دوم اینکه گره، بلاک را امضا نکند و بلاک نیز در اجماع مورد تأیید قرار نگیرد. به‌عبارتی چنانچه گره مطابق با تأیید یا عدم تأیید نهایی بلاک توسط گره راهبر عمل کند، عملکرد او صحیح در نظر گرفته می‌شود. در این حالت RSP جدید گره به‌صورت زیر محاسبه می‌شود

$$RSP_{new} = \left\lfloor RSP_{old} - \left\lfloor \frac{R_{default}}{R} \right\rfloor \right\rfloor \qquad (۲)$$

همچنین برای محاسبه اعتبار گره (R) برای او یک برد در الگوریتم گلیکو در نظر گرفته می‌شود؛ اما به دلیل اینکه بازیکنی وجود ندارد که با آن رقابت کند، ما یک بازیکن مجازی را با امتیاز پیش‌فرض برای این موضوع در نظر می‌گیریم.

طبق (۲) در صورتی که گره به‌درستی عمل کند، RSP او کاهش می‌یابد؛ زیرا ما نمی‌خواهیم گره‌ها امکان شرکت متوالی در اجماع را داشته باشند. پس با هر بار شرکت یک گره در اجماع، RSP او و در نتیجه شانس او برای انتخاب‌شدن در دورهای بعدی کمتر می‌گردد. پارامتر R که میزان اعتبار یک گره را نشان می‌دهد نیز در این محاسبه دخیل است. هرچه اعتبار یک گره بیشتر باشد، کاهش RSP او کمتر خواهد بود و به این معناست که RSP گره‌هایی با اعتبار بیشتر، پس از شرکت در اجماع، کمتر کاهش می‌یابد.

حالت دوم: گره برای اجماع انتخاب گردیده و عملکرد نادرست داشته است.

عملکرد نادرست نیز می‌تواند به دو صورت اتفاق بیفتد. اول اینکه گره، بلاک را امضا نماید و نهایتاً آن گره در اجماع مورد تأیید قرار نگیرد. دوم اینکه گره، بلاک را امضا نکند اما بلاک در اجماع مورد تأیید قرار گیرد. به‌عبارتی چنانچه گره مطابق با تأیید یا عدم تأیید نهایی بلاک توسط گره راهبر عمل نکند، عملکرد او نادرست در نظر گرفته می‌شود. در این حالت RSP گره را به RSP_{min} تنظیم می‌کنیم. به نوعی گره تنبیه شده و شانس او به حداقل کاهش می‌یابد. این موضوع باعث می‌شود که گره‌های خرابکار به‌مرور از دور رقابت خارج شوند. ضمن اینکه در الگوریتم گلیکو یک باخت برای او در نظر می‌گیریم تا علاوه بر شانس، اعتبار گره نیز کاهش یابد.

با اینکه RSP_{min} شانس گره را برای شرکت در دورهای بعدی به‌شدت کاهش می‌دهد، اما این شانس صفر نمی‌شود. از این رو اگر گرهی به هر دلیلی به‌صورت سهوی عملکرد نادرست داشت، باز هم بتواند در اجماع‌های بعد انتخاب شود؛ زیرا شانس او کم‌کم افزایش یافته و نهایتاً راهی برای بازگشت به دور رقابت پیدا خواهد کرد.

حالت سوم: گره برای فرایند اجماع انتخاب نشده است. در این حالت RSP گره از (۳) محاسبه می‌شود

$$RSP_{new} = \left\lfloor RSP_{old} + \left\lfloor \frac{R}{R_{default}} \right\rfloor \right\rfloor \qquad (۳)$$

رابطه (۳) به‌گونه‌ای عمل می‌کند که بر اساس نسبت اعتبار گره به اعتبار پیش‌فرض، مقداری به شانس گره افزوده شود؛ بدین معنی که شانس گره‌هایی که در اجماع شرکت نمی‌کنند به‌تدریج برای انتخاب‌شدن در دورهای بعدی اجماع افزایش می‌یابد. گره‌هایی با اعتبار بیشتر، RSP بیشتری برای دور بعدی به‌دست می‌آورند تا شانس بیشتری برای انتخاب در فرایند اجماع داشته باشند. این موضوع عدالت در انتخاب را برآورده نموده و از تمرکزگرایی و به‌دست‌گرفتن حق شرکت در اجماع توسط گروهی خاص جلوگیری می‌کند. نهایتاً برای گره یک تساوی در گلیکو در نظر می‌گیریم. علت تساوی آن است که همه گره‌ها در هر دور اجماع، یک بازی در گلیکو انجام دهند تا محاسبات الگوریتم برای مشخص‌نمودن امتیاز کاربران دقیق باشد.

همان طور که مشاهده می‌شود، الگوریتم به‌روزرسانی امتیازها به‌شدت نسبت به عملکرد نادرست گره‌ها حساس بوده و به‌سرعت گره‌های خرابکار را از دور رقابت خارج می‌کند. همچنین هم شانس گره‌هایی را که سهواً نادرست عمل کرده‌اند و هم شانس گره‌هایی را که در اجماع شرکت نداشته‌اند، تدریجاً افزایش می‌دهد. برای افزایش تدریجی و کاهش ناگهانی مورد استفاده در این روش از الگوریتم معروف AIMD [۴۲] الهام گرفته شده است.

۳-۶ خرید اعتبار اولیه

همان طور که در بخش ۳ اشاره شد، شبکه مبتنی بر الگوریتم پیشنهادی از دو نوع گره راهبر و عادی تشکیل شده است. گره‌های راهبر، کمیته راهبران را تشکیل می‌دهند و بنا بر وظایفی که بر عهده آنهاست، فرض می‌کنیم که این گره‌ها ایمن بوده و دستورات را به‌درستی و در زمان مناسب انجام دهند؛ اما گره‌های عادی می‌توانند به‌صورت ناشناس به شبکه، اضافه یا از آن خارج شوند. ناشناس‌بودن این امکان را فراهم می‌کند که گره‌های خرابکار نیز در شبکه حضور داشته باشند.

پس از ورود گره به شبکه، مقدار RSP و R گره برابر صفر خواهد بود تا از شرکت گره در اجماع جلوگیری شود. برای اینکه گره بتواند در اجماع شرکت نماید باید همانند الگوریتم DLattice، شانس اولیه RSP و اعتبار اولیه R را از کمیته راهبران خریداری نماید. پس از خریداری این مقادیر، گره شانس حضور در دور بعدی اجماع را به‌دست خواهد آورد و چنانچه توسط الگوریتم RNSR برای اجماع انتخاب شود، می‌تواند بلاک جدید را تأیید و یا رد نماید.

زمانی می‌توان حمله سیبل را روی شبکه اعمال کرد که یک ماشین بتواند با ایجاد حساب‌های کاربری فراوان، شانس خود را در رأی‌گیری و شرکت در اجماع بیشتر کند. همانند الگوریتم DLattice جهت شرکت در اجماع، یک گره باید شانس اولیه RSP و اعتبار اولیه R را از گره‌های دیگر یا مدیر سامانه خریداری نماید. اعمال این موضوع در الگوریتم اجماع پیشنهادی این مقاله، ایجاد حساب کاربری که قادر به شرکت در رقابت است را هزینه‌بر نموده و حمله سیبل را غیرعملی و ناممکن می‌سازد.

۴- ارزیابی الگوریتم پیشنهادی

هدف اصلی الگوریتم پیشنهادی ما رسیدن به امنیت در برابر چهار حمله رایج در بستر بلاک‌چین است، اما این موضوع نباید کارایی الگوریتم را زیر سؤال ببرد. لذا در طراحی آن تلاش نمودیم راه‌حلی ارائه دهیم که علاوه بر رسیدن به اهداف اصلی از کارآمدی خوبی نیز برخوردار باشد. اما باید این نکته را نیز در نظر داشت که اهداف مختلف در برخی موارد در تقابل با هم هستند و دستیابی به یک هدف ممکن است سایر اهداف را تحت‌الشعاع خود قرار دهد. ما در الگوریتم پیشنهادی خود تلاش نمودیم تعادل خوبی بین امنیت، کارایی و عملکرد آن برقرار سازیم تا بتوان در محدوده وسیع‌تری از زمینه‌ها از این الگوریتم بهره گرفت. در این بخش، ابتدا یک مثال واقعی را مورد بررسی قرار می‌دهیم و سپس الگوریتم خود را از سه منظر امنیت، عملکرد و کارایی ارزیابی خواهیم کرد.

۴-۱ بررسی یک مثال واقعی

یک سامانه زنجیره تأمین کالای مبتنی بر بلاک‌چین را در نظر بگیرید. یک موجودیت خرابکار تصمیم دارد که یک کالای تقلبی را به سامانه اضافه کند. برای انجام چنین کاری باید یک بلاک جدید با مشخصات کالای تقلبی ایجاد نموده و به شبکه بفرستد و اطمینان حاصل کند که این بلاک جدید تأیید خواهد شد. موجودیت خرابکار برای چنین کاری نیاز دارد که حداقل دوسوم گره‌های منتخب در آن دور اجماع را تحت کنترل خود داشته باشد که بر اساس احتمالات باید حداقل دوسوم از کل گره‌های شبکه را در اختیار داشته باشد که این امر برای او بسیار پرهزینه است.

چنانچه گره خرابکار بتواند تعدادی از گره‌های منتخب را تحت کنترل بگیرد و این تعداد کمتر از دوسوم باشد- از آنجا که نمی‌تواند بلاک را تأیید کند- RSP و R گره‌های تحت کنترل او و به همین نسبت شانس آنها برای دورهای بعدی اجماع کم می‌شود و هزینه‌ای که صرف تحت کنترل قراردادن آن گره‌ها شده است عملاً بی‌نتیجه باقی می‌ماند.

۴-۲ ارزیابی امنیتی

همان طور که در بخش قبل اشاره شد، هدف اصلی الگوریتم پیشنهادی، ایجاد امنیت در برابر چهار حمله رایج بر بستر بلاک‌چین است. در این بخش، هر حمله و نحوه مقاومت الگوریتم اجماع پیشنهادی در برابر آنها را شرح خواهیم داد.

حمله سیبل: همان طور که پیشتر اشاره شد در حمله سیبل، یک موجودیت تلاش می‌کند تا با ایجاد حساب‌های کاربری فراوان در شبکه، شانس و حق رأی بیشتری جهت تغییر در فرایند اجماع و جعل بلاک‌های جدید کسب نماید.

اگرچه در الگوریتم پیشنهادی ما برای ایجاد حساب‌های کاربری جدید محدودیتی وجود ندارد و گره‌ها می‌توانند به‌صورت ناشناس به شبکه اضافه شوند، اما هر گره برای شرکت در فرایند اجماع باید RSP و R اولیه خود را از کمیته راهبران خریداری نماید. این موضوع ایجاد حساب‌های کاربری فراوان را که قابلیت شرکت در اجماع دارند، هزینه‌بر نموده و حمله سیبل را برای یک حمله‌کننده پرهزینه و غیرقابل انجام می‌سازد. ضمن اینکه به واسطه سازوکار NONS تعبیه‌شده در الگوریتم، حتی با ایجاد حساب‌های کاربری فراوان و با بالارفتن تعداد گره‌های موجود در شبکه، نسبت تعداد گره‌های انتخاب‌شده به کل گره‌ها به‌شدت کاهش می‌یابد.

حمله منع خدمت: حمله‌کننده در حمله منع خدمت، تلاش دارد تا با ارسال حجم زیادی از بسته‌ها یا درخواست‌ها به گره هدف، آن را از کار بیندازد. در این حمله، چنانچه گره هدف از قبل قابل شناسایی و رهگیری باشد، حمله راحت‌تر صورت گرفته و گره مورد نظر در معرض خطر بیشتری قرار می‌گیرد.

از آنجا که در الگوریتم پیشنهادی در هر دور اجماع، گروهی از گره‌ها توسط کمیته راهبران به‌صورت تصادفی انتخاب می‌شوند، شناخت گره هدف توسط حمله‌کننده سخت می‌شود. ضمن اینکه گره‌های شرکت‌کننده در اجماع در هر دور تغییر می‌کنند که این موضوع شرایط را برای شناسایی هدف توسط حمله‌کننده به‌مراتب سخت‌تر می‌کند.

جدول ۳: مقادیر پیش‌فرض کلاس NODE.

مؤلفه	مقدار
$R_{default}$	۱۵۰۰
$RD_{default}$	۳۵۰
$RSP_{default}$	۱۵۰۰
RSP_{min}	۱
RSP_{max}	$۲^{۳۲}$

جدول ۴: پارامترهای کلی الگوریتم اجماع پیشنهادی.

پارامتر	توضیح
n	تعداد گره‌ها
itr	تعداد اجرای الگوریتم
devil	شماره گره خرابکار
Fault probability	احتمال خطای هر گره

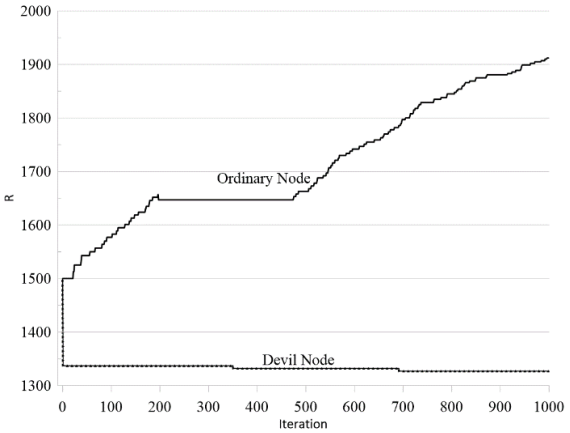
حمله ۵۱ درصد: حمله‌کننده در حمله ۵۱ درصد باید کنترل بیش از ۵۰ درصد از گره‌های شبکه را در اختیار گیرد. با توجه به تأثیری که مؤلفه‌های اعتبار گره (R) و شانس گره (RSP) بر انتخاب گره در دورهای بعدی اجماع می‌گذارند، حمله‌کننده لزوماً با کنترل گره‌ها نمی‌تواند بر روی شبکه تأثیری بگذارد؛ زیرا به‌تدریج گره‌های تحت کنترل حمله‌کننده، اعتبار خود را از دست داده و از چرخه رقابت خارج می‌شوند. ضمناً با توجه به دو الگوریتم NONS و RNSR و انتخاب تصادفی گره‌ها، هیچ تضمینی وجود ندارد که حتی با دردست‌داشتن بیش از ۵۰ درصد گره‌ها، حمله‌کننده بتواند تعداد لازم گره منتخب جهت شرکت در فرایند اجماع و تأثیر در تأیید یا عدم تأیید بلاک بعدی را در کنترل خود درآورد.

حمله کسوف: مهاجم در حمله کسوف، تمام ارتباطات ورودی و خروجی قربانی را در انحصار خود درمی‌آورد؛ بنابراین قربانی را از بقیه همتایان خود در شبکه جدا می‌کند [۴۳]. از آنجا که گره هدف از دفتر کل بلاک‌چین جدا شده است، گره جداشده می‌تواند توسط مهاجم دستکاری شود. حمله کسوف می‌تواند منجر به اختلال در استخراج بلاک و همچنین تأیید تراکنش‌های نامشروع شود [۴۴]. گره مهاجم در این حمله، نیاز به شناسایی گره هدف دارد؛ اما با توجه به آنکه در الگوریتم پیشنهادی ما همانند الگوریتم Block-Supply، گره‌های شرکت‌کننده در اجماع در هر دور به‌صورت تصادفی انتخاب می‌شوند، در نتیجه کار شناسایی گره هدف بسیار سخت شده و عملاً شبکه را در برابر حمله کسوف ایمن می‌سازد.

۴-۳ ارزیابی عملکرد

ابتدا در این بخش، محیط شبیه‌سازی الگوریتم را شرح داده و سپس با توجه به مؤلفه‌های مختلف به بررسی عملکرد سامانه خواهیم پرداخت. ما برای شبیه‌سازی الگوریتم از یک کلاس Node استفاده نمودیم که خصوصیات مختلف هر گره در آن ذخیره خواهد شد. مقادیر پیش‌فرض گره‌ها را که در کلاس Node استفاده شده در جدول ۳ مشاهده می‌کنید.

مقادیر پیش‌فرض R و RD دقیقاً همان مقادیر پیش‌فرض الگوریتم گلیکو هستند و مقدار پیش‌فرض RSP را نیز ۱۵۰۰ در نظر می‌گیریم تا با R همخوانی داشته باشد. RSP_{max} نیز برابر با بزرگ‌ترین عدد قابل ذخیره در حافظه در نظر گرفته شده است. برای هر گره چهار پارامتر R، RD، RSP و $Selection_{count}$ را جهت ذخیره‌سازی وضعیت آن گره در نظر می‌گیریم. پس از آنکه گره به شبکه ملحق شد و اعتبار اولیه را خریداری نمود، مقدار R، RD و RSP گره به مقادیر پیش‌فرض هر کدام تنظیم خواهد شد.



شکل ۹: تغییرات R گره درستکار و خرابکار.

حال خصوصیات کلی الگوریتم را تنظیم می‌کنیم تا بر اساس آنها بتوانیم الگوریتم را در شرایط مختلف، کنترل و آزمایش نماییم. همان طور که در جدول ۴ مشاهده می‌نمایید، *n* تعداد کل گره‌ها و *itr* تعداد اجرای الگوریتم یا به عبارتی تعداد دفعاتی را که فرایند اجماع باید انجام شود نشان می‌دهند. از آنجا که برای آزمودن الگوریتم و نحوه مقابله آن با گره‌های خرابکار باید رفتار الگوریتم با گره خرابکار را زیر نظر بگیریم، یک گره را به‌عنوان خرابکار مشخص می‌کنیم. خصوصیت این گره آن است در هر دور که برای اجماع انتخاب شود امضای صحیح انجام نخواهد داد یا به عبارتی در همه شرایط، خلاف تصمیم نهایی اتخاذشده رأی خواهد داد.

پارامتر عمومی دیگر احتمال خطا[1] است که مشخص می‌کند هر گره درستکار با چه احتمالی دچار خطا می‌شود. در محیط واقعی ممکن است این خطا به دلایل مختلفی نظیر عدم برخط‌بودن گره، عدم پاسخ‌دهی گره در زمان مناسب و یا هر خطای دیگری در زمان دریافت، امضاکردن یا ارسال آن به گره راهبر رخ دهد. با درنظرگرفتن موارد بالا، الگوریتم را برای تعداد گره‌های مختلف و تعداد اجراهای مختلف بررسی می‌نماییم.

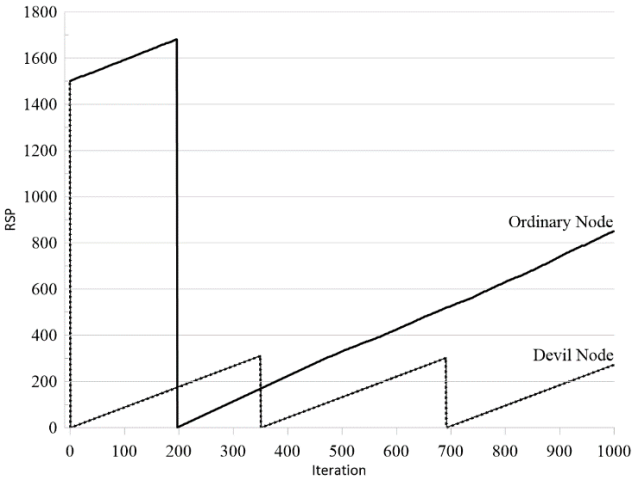
۴-۳-۱ بررسی انتخاب گره خرابکار

با توجه به اینکه گره خرابکار در هر دور انتخاب توسط راهبر، رأی خود را به‌صورت نادرست ارسال می‌کند، در اکثر اجراهای الگوریتم پس از اولین باری که گره خرابکار انتخاب شده و با توجه به دو خصوصیت R و RSP، شانس گره برای انتخاب مجدد به حدی کم می‌شود که عملاً تا انتهای اجرای الگوریتم دیگر انتخاب نمی‌شود. این موضوع به‌خوبی در جدول ۵ آمده است. در این جدول، تعداد دفعاتی که گره خرابکار در دورهای مختلف اجماع انتخاب شده را مشاهده می‌نمایید. حال جدول ۵ را با جدول ۶ که تعداد انتخاب‌های یک گره عادی را نشان می‌دهد، مقایسه کنید. این موضوع نشان می‌دهد که رفتار نادرست در سامانه، چه میزان در انتخاب گره‌ها برای دورهای بعدی اجماع تأثیر می‌گذارد.

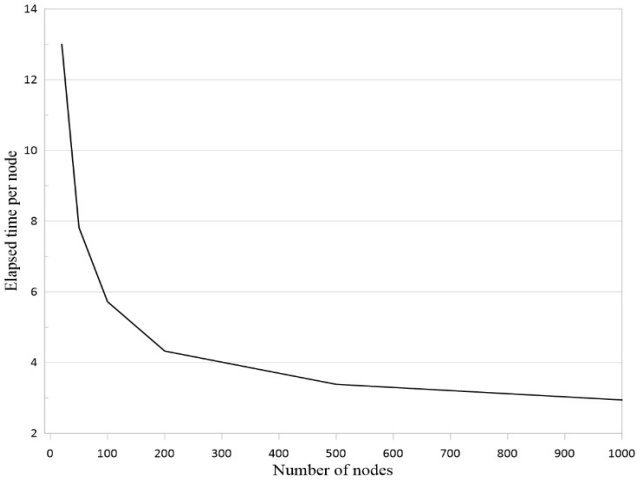
۴-۳-۲ بررسی اعتبار و شانس گره‌ها

با وجود آنکه هر دو مؤلفه R و RSP در انتخاب گره برای دور بعدی اجماع نقش دارند، کارکرد هر کدام و نحوه تغییرات آنها بسته به رفتار گره متفاوت است. مثلاً در ۱۰۰۰ دور اجرای آزمایشی الگوریتم، تغییرات R و RSP دو گره (یکی گره درستکار و دیگری گره خرابکار) را بررسی کردیم. در این ۱۰۰۰ دور اجماع، گره عادی ۹۶ بار و گره خرابکار تنها ۳ بار برای اجماع انتخاب شده است. در شکل ۹ تغییرات R گره درستکار و خرابکار و در شکل ۱۰ تغییرات RSP گره درستکار و خرابکار را مشاهده می‌کنید.

1. Fault Probability



شکل ۱۰: تغییرات RSP گره درستکار و خرابکار.



شکل ۱۱: زمان سپری‌شده اجماع به‌ازای هر گره در الگوریتم اجماع پیشنهادی.

جدول ۵: تعداد انتخاب گره خرابکار برای اجماع در اجراهای مختلف الگوریتم پیشنهادی.

تعداد اجرا \ تعداد گره	۱۰	۲۵	۵۰	۱۰۰	۲۵۰	۵۰۰	۱۰۰۰
۱۰	۱	۱	۱	۰	۰	۱	۰
۲۵	۱	۱	۱	۱	۱	۱	۱
۵۰	۱	۱	۱	۱	۰	۱	۱
۱۰۰	۱	۱	۱	۱	۱	۱	۱
۲۵۰	۱	۱	۱	۱	۱	۱	۱
۵۰۰	۲	۲	۲	۲	۲	۲	۲
۱۰۰۰	۴	۳	۳	۳	۳	۳	۳

جدول ۶: تعداد انتخاب گره درستکار برای اجماع در اجراهای مختلف.

تعداد اجرا \ تعداد گره	۱۰	۲۵	۵۰	۱۰۰	۲۵۰	۵۰۰	۱۰۰۰
۱۰	۸	۵	۵	۱	۰	۱	۰
۲۵	۲۲	۱۶	۱۲	۲	۱	۱	۲
۵۰	۴۴	۳۱	۱۳	۸	۵	۶	۱
۱۰۰	۹۱	۵۷	۳۰	۱۶	۹	۵	۴
۲۵۰	۲۲۵	۱۴۶	۶۷	۴۴	۲۴	۱۵	۸
۵۰۰	۴۷۵	۲۶۴	۱۵۰	۱۱۱	۴۹	۲۹	۱۴
۱۰۰۰	۸۹۵	۵۴۷	۳۲۳	۱۹۸	۱۰۸	۶۰	۳۳

در این شبیه‌سازی، احتمال خطای گره را برابر با ۱٪ در نظر گرفتیم و بدین معناست که یک گره درستکار در هر ۱۰۰۰ دور اجماع، احتمالاً یک بار دچار خطا می‌شود. همان طور که در شکل ۹ مشاهده می‌کنید، اعتبار گره درستکار در ابتدا با هر بار انتخاب‌شدن افزایش می‌یابد. گره انتخابی در اجرای ۱۹۶ام دچار خطا می‌گردد. بنابراین مقداری از اعتبار او کاسته شده و پس از آن تا ۳۰۰ دور آینده، اعتبار گره ثابت باقی می‌ماند که نشان‌دهنده عدم انتخاب گره در اجماع است. از حدود دور ۵۰۰ام به بعد، گره مجدداً شانس شرکت در اجماع را به‌دست آورده و با رفتار درستی که از خود نشان می‌دهد، اعتبار او به‌تدریج افزایش می‌یابد.

همان طور که در شکل ۱۰ مشاهده می‌نمایید، وقوع یک خطا توسط گره درستکار باعث می‌شود شانس گره (RSP) برای انتخاب در دورهای بعدی به‌شدت کاهش یابد. همچنین گره خرابکار در ابتدای اجرای الگوریتم برای اجماع انتخاب شده و با رأی نادرستی که ارائه می‌کند، اعتبار او کاهش می‌یابد. سپس تا حدود دور ۳۵۰، الگوریتم اعتبارش ثابت می‌ماند که نشان‌دهنده عدم انتخاب او توسط الگوریتم است. با توجه به عدم انتخاب گره، RSP او رفته‌رفته افزایش می‌یابد؛ اما با انتخاب مجدد و رأی نادرست گره، شانس او مجدداً صفر می‌شود. این تغییرات از الگوریتم افزایش تدریجی و کاهش ناگهانی (AIMD) که پیشتر در بخش ۳-۵-۲ بدان اشاره شد، تبعیت می‌کند.

٤-٤ ارزیابی کارایی

یکی از مواردی که برای طراحی الگوریتم باید در نظر گرفت، زمان اجرای الگوریتم است. الگوریتم پیشنهادی ما از یک سازوکار انتخاب گره (NONS) برای شرکت در اجماع استفاده می‌کند. این سازوکار طوری طراحی گردیده که با ده برابرشدن تعداد کل گره‌ها به مجموع گره‌های انتخابی برای اجماع تنها ۱۰ عدد اضافه می‌شود. به‌طوری که به ازای صدهزار گره متصل در شبکه ۵۰ گره و به ازای یک میلیون گره فقط ۶۰ گره برای اجماع انتخاب می‌شوند و ادامه فرایند اجماع با این ۵۰ یا ۶۰ گره پی گرفته می‌شود. این موضوع قابلیت مقیاس‌پذیری را برای الگوریتم فراهم آورده و کارایی الگوریتم را افزایش می‌دهد.

برای بررسی این موضوع، الگوریتم را به‌ترتیب با ۲۰، ۵۰، ۱۰۰، ۲۰۰، ۵۰۰ و ۱۰۰۰ گره و برای ۱۰۰ دور اجرا نمودیم. سپس زمان سپری‌شده کل را بر تعداد گره‌های شبکه تقسیم کردیم تا متوسط زمان سپری‌شده به ازای هر گره به‌دست آید. همان طور که در شکل ۱۱ مشاهده می‌نمایید با افزایش تعداد گره‌های شبکه، متوسط زمان سپری‌شده به ازای هر گره به طرز چشم‌گیری کاهش می‌یابد.

۵- نتیجه‌گیری

الگوریتم اجماع پیشنهادی با ترکیبی از ویژگی‌های مثبت الگوریتم‌های دیگر و افزودن خصوصیت‌های منحصربه‌فردی نظیر تابع لجستیک، الگوریتم AIMD و الگوریتم گلیکو طراحی شده است. در این الگوریتم توانستیم با استفاده از روش انتخاب تصادفی گره‌های شرکت‌کننده در اجماع، استفاده از تابع لجستیک، استفاده از الگوریتم گلیکو، الگوریتم اجماع پیشنهادی خود را در مقابل حمله کسوف و منع خدمت ایمن کنیم. همچنین با سازوکار RSP و اعتبار R از ایجاد تمرکز در سامانه و حمله ۵۱ درصد جلوگیری کردیم و نهایتاً برای شرکت در اجماع، تمهیدی اندیشیدیم که گره‌های جدید RSP و R اولیه را خریداری نمایند تا عملاً ایجاد حساب‌های بیشتر و شرکت در اجماع هزینه‌بر بوده و الگوریتم را در برابر حمله سیبل ایمن کند. همچنین نشان داده شد که الگوریتم پیشنهادی از قابلیت مقیاس‌پذیری و کارایی خوبی جهت گسترش شبکه و اضافه‌شدن

گره‌ها به آن برخوردار است.

در کارهای آتی قصد داریم با ارائه راهکاری جدید، الگوریتم را از این فرض که گره‌های کمیته راهبران باید امن باشند، بی‌نیاز و امکان انتخاب هر گره عادی به‌عنوان گره راهبر را فراهم کنیم. ضمن اینکه به‌جای امضای عادی از امضای آستانه‌ای برای تأیید بلاک جدید استفاده نماییم تا فرایند تأیید بلاک و افزودن آن به بلاک‌چین با امنیت و سرعت بالاتری انجام پذیرد.

مراجع

[1] L. Zhu, K. Gai, and M. Li, *Blockchain Technology in Internet of Things*, Springer, 2019.

[2] R. Pinto, *What Role will Blockchains Play in Cybersecurity?* Forbes Technology Council. April 3. 2019. https://www.forbes.com/sites/forbestechcouncil/2019/04/03/what-role-will-blockchains-play-in-cybersecurity/ (Accessed Apr. 15, 2019).

[3] C. Thompson, *How Does the Blockchain Work? (Part 1)*, Medium, 2016. https://medium.com/blockchain-review/how-does-the-blockchain-work-for-dummies-explained-simply-9f94d386e093 (Accessed Jun. 30, 2021).

[4] I. Marco and K. R. Lakhani, "The truth about blockchain," *Harv. Bus. Rev.*, vol. 95, no. 1, pp. 118-127, 2017.

[5] D. Freuden, "Hybrid Blockchains: The Best of Both Public and Private, Brave New Coin, 2018. https://bravenewcoin.com/insights/hybrid-blockchains-the-best-of-both-public-and-private (Accessed Jun. 30, 2021).

[6] The Investopedia Team, *Consensus Mechanism (Cryptocurrency) Definition*, https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp (Accessed Aug. 13, 2021).

[7] D. Hellwig, G. Karlic, and A. Huchzermeier, *Build Your Own Blockchain: A Practical Guide to Distributed Ledger Technology*, Springer, 2019.

[۸] س. ع. بنوفاطمه، *بررسی الگوریتم اجماع اثبات سهام (PoS) و ۳ شبکه محبوب آن*، ۱۳۹۸، https://www.bourseiness.com/41344/proof-of-stake (دسترسی شهریور ۱۵، ۱۴۰۱).

[9] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proc. of the 3rd Symp. on Operating Systems Design and Implementation*, pp. 173-186, New Orleans, LA, USA, 22-22 Feb. 1999.

[10] K. Seifried, *Over 200 Documented Blockchain Attacks, Vulnerabilities and Weaknesses*, CSA| CSA, https://cloudsecurityalliance.org/blog/2020/10/26/blockchain-attacks-vulnerabilities-and-weaknesses/ (Accessed Sept. 6, 2022).

[11] V. Patel, "A framework for secure and decentralized sharing of medical imaging data via blockchain consensus," *Health Informatic Journal*, vol. 25, no. 4, pp. 1398-1411, Dec. 2019.

[12] R. Pass, and E. Shi, "Fruitchains: a fair blockchain," in *Proc. of the ACM Symp. on Principles of Distributed Computing*, pp. 315-324, Washington DC, USA, 25-27 Jul. 2017.

[13] M. Milutinovic, W. He, H. Wu, … M. K. the 1st W. on S., and undefined 2016, "Proof of luck: an efficient blockchain consensus protocol," in *Proc. of the 1st Workshop on System Software for Trusted Execution*, Trento, Italy, 12-16 Dec. 2016.

[14] S. Kim, "Two-phase cooperative bargaining game approach for shard-based blockchain consensus scheme," *IEEE Access*, vol. 7, pp. 127772-127780, 2021.

[15] D. Vangulick, B. Cornélusse, and D. Ernst, "Blockchain: a novel approach for the consensus algorithm using Condorcet voting procedure," in *Proc. of the IEEE Int. Conf. on Decentralized Applications and Infrastructures*, 10 pp. Newark, CA, USA, 4-9 Apr. 2019.

[16] M. Ahmed-Rengers and K. Kostiainen, *Don't* Mine, *Wait in Line: Fair and Efficient Blockchain Consensus with Robust Round Robin*, Apr. 2018, http://arxiv.org/abs/1804.07391 (Accessed Aug. 3, 2021).

[17] S. Azouvi, P. McCorry, and S. Meiklejohn, *Betting on Blockchain Consensus with Fantomette*, May 2018, http://arxiv.org/abs/1805.06786 (Accessed Aug. 3, 2021).

[18] D. Tosh, S. Shetty, P. Foytik, C. Kamhoua, and L. Njilla, "CloudPoS: a proof-of-stake consensus design for blockchain integrated cloud," in *Proc. of the IEEE 11th Int. Conf. on Cloud Computing*, pp. 302-309, San Francisco, CA, USA, 2-7 Jul. 2018.

[19] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Novel trust consensus protocol and blockchain-based trust evaluation system for M2M application services," *Internet of Things*, vol. 7, Article ID: 100058, Sept. 2017.

[20] S. Leonardos, D. Reijsbergen, and G. Piliouras, "Weighted voting on the blockchain: Improving consensus in proof of stake protocols," *Int. J. Netw. Manag.*, vol. 30, no. 5, Article ID: e 2093, Sept./Oct. 2020.

[21] F. Yang, W. Zhou, Q. Wu, R. Long, … N. X.-I., and U. 2019, "Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism," *IEEE Access*, vol. 7, pp. 118541-11855, 2019.

[22] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, "Blockchain-driven IoT for food traceability with an integrated consensus mechanism *Access*, vol. 7, pp. 129000-129017, 2019.

[23] Z. Ren, K. Cong, J. Pouwelse, and Z. Erkin, "Implicit Consensus: Blockchain with Unbounded Throughput," May 2017, http://arxiv.org/abs/1705.11046 (Accessed Aug. 3, 2021)

[24] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *IEEE Trans. on Computers*, vol. 68, no. 1, pp. 139-151, Jan. 2019.

[25] F. Muratov, A. Lebedev, N. Iushkevich, B. Nasrulin, and M. Takemiya, *YAC: BFT Consensus Algorithm for Blockchain*, Sept. 2018, http://arxiv.org/abs/1809.00554 (Accessed Aug. 3, 2021).

[26] E. Buchman, J. Kwon, and Z. Milosevic, *The Latest Gossip on BFT Consensus*, Jul. 2018, http://arxiv.org/abs/1807.04938 (Accessed Aug. 3, 2021).

[27] N. Alzahrani and N. Bulusu, "A new product anti-counterfeiting blockchain using a truly decentralized dynamic consensus protocol," *Concurrency and Computation Practice and Experience*, vol. 32, no. 12, Article ID: e5232, Jun. 2019.

[28] F. Bravo-Marquez, S. Reeves and M. Ugarte, "Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions," in *Proc. of the IEEE Int. Conf. on Decentralized Applications and Infrastructures*, pp. 119-124, Newark, CA, USA, 4-9 Apr. 2019.

[29] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, "Solida: A Blockchain Protocol Based on Reconfigurable Byzantine Consensus," Mar. 2018, https://arxiv.org/abs/1612.02916v1 (Accessed Aug. 3, 2021).

[30] R. Pass, E. Shi, "Hybrid consensus: efficient consensus in the permissionless model," *in Proc. 31st Int. Symp. on Distributed Computing*, vol. 91, pp. 39:1-39:16, Vienna, Austria, 16-20 Oct. 2017.

[31] T. Zhou, X. Li, and H. Zhao, "DLattice: q permission-less blockchain based on DPoS-BA-DAG consensus for data tokenization," *IEEE Access*, vol. 7, pp. 39273-39287, 2019.

[32] Z. –C. Li, J. –H. Huang, D. –Q. Gao, Y. –H. Jiang, and L. Fan, "ISCP: an improved blockchain consensus protocol.," *International Journal of Network Security*, vol. 21, no. 3, PP.359-367, May 2019.

[33] K. Li, H. Li, H. Hou, K. Li and Y. Chen, "Proof of vote: a high-performance consensus protocol based on vote mechanism & consortium blockchain," in *Proc. IEEE 19th Int. Conf. on High Performance Computing and Communications; IEEE 15th Int. Conf. on Smart City; IEEE 3rd Int. Conf. on Data Science and Systems*, pp. 466-473, Bangkok, Thailand, 18-20 Dec. 2017.

[34] K. Finlow-Bates, *A Lightweight Blockchain Consensus Protocol*, Aug. 2017, https://www.chainfrog.com/wp-content/uploads/2017/08/consensus.pdf (Accessed Aug. 3, 2021).

[35] S. Solat, "RDV: An alternative to proof-of-work and a real decentralized consensus for blockchain," in *Proc. the 1st Workshop on Blockchain-enabled Networked Sensor Systems*, pp. 25-31, Shenzhen, China, 4-4 Nov. 2018.

[36] A. K. Talukder, M. Chaitanya, D. Arnold and K. Sakurai, "Proof of disease: a blockchain consensus protocol for accurate medical decisions and reducing the disease burden," in *Proc. of the SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation,* pp. 257-262, Guangzhou, China, 08-12 Oct. 2018.

[37] H. Y. Yuen, *et al.*, "Proof-of-play: A novel consensus model for blockchain-based peer-to-peer gaming system," in *Proc. of the ACM Int. Symp. on Blockchain and Secure Critical Infrastructure*, pp. 19-28, Auckland, New Zealand, 8-8 Jul. 2019.

[38] E. K. Wang, Z. Liang, C. -M. Chen, S. Kumari, and M. Khurram Khan, "PoRX: A reputation incentive scheme for blockchain consensus of IIoT," *Future Generation Computer Systems*, vol. 102, pp. 140-151, Jan. 2020.

[39] S. Bouraga, "A taxonomy of blockchain consensus protocols: A survey and classification framework," *Expert Syst. Appl.*, vol. 168, Article ID: 114384, Apr. 2021.
[40] -, *Logistic Functions*, http://wmueller.com/precalculus/families/1_80.html (Accessed Oct. 01, 2022).
[41] I. Syed, *PHP: Utility Function for Getting Random Values with Weighting*, 2015, https://gist.github.com/irazasyed/f41f8688a2b3b8f7b6df (Accessed Oct. 15, 2022).
[42] D. M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Networks ISDN Syst.*, vol. 17, no. 1, pp. 1-14, Jun. 1989.
[43] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, *Eclipse Attacks on Bitcoin's Peer-to-Peer Network*, Mar. 2015, https://hashingit.com/elements/research-resources/2015-03-20-eclipse-attacks-bitcoin.pdf (Accessed Oct. 15, 2022).
[44] M. Deer, *What Is an Eclipse Attack?* Dec. 2021, https://cointelegraph.com/explained/what-is-an-eclipse-attack (Accessed Oct. 25, 2022).

حسین بدری تحصیلات خود را در رشته مهندسی نرم افزار کامپیوتر در مقطع کارشناسی در سال ۱۳۹۱ در دانشگاه پیام نور رامسر و در مقطع کارشناسی ارشد در سال ۱۴۰۱ در دانشگاه تربیت دبیر شهید رجایی تهران به پایان رسانده است. زمینه¬های تحقیقاتی مورد علاقه ایشان عبارتند از بلاک‌چین، سامانه‌های توزیع‌شده، امنیت شبکه و رمزنگاری.

معصومه صفخانی در سال ۱۳۸۳ مدرک کارشناسی مهندسی برق–الکترونیک خود را از دانشگاه علم و صنعت ایران و در سال‌های ۱۳۸۶ و ۱۳۹۲ به ترتیب مدرک کارشناسی ارشد مهندسی برق–الکترونیک و دکترای مهندسی برق خود را از دانشگاه علم و صنعت ایران دریافت نمود. از سال ۱۳۹۳ نام‌برده به عنوان عضو هیآت علمی دانشکده مهندسی کامپیوتر دانشگاه تربیت دبیر شهید رجایی مشغول فعالیت گردید. زمینه‌های علمی مورد علاقه ایشان متنوع بوده و شامل موضوعاتی مانند ایده‌های نو در طراحی و تحلیل پروتکل‌های امنیتی، طراحی و تحلیل شبکه های مبتنی بر بلاک چین، حفظ حریم خصوصی، رمزنگاری، امنیت شبکه‌های اینترنت اشیا، امنیت انواع شبکه‌های ارتباطاتی و داده کاوی با حفظ حریم خصوصی می‌باشد.