باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس مبانی تکنولوژی بلاکچین و رمزارزها

گزارش پروژه

علی محرابیان96102331

استاد: دکتر مداح علی

زمستان 1398

در این مقاله هدف محدود کردن وکاهش انگیزه برای انجام اعمال سودجویانه در پروتکل proof of stake است.به این منظور دو ویژگی مکملی که هر longest chain در این پروتکل باید یکی از آن هارا داشته باشد، به صورت دقیق و فرمول بندی شده ارائه می شود.سپس حملات سودجویانه ای که با تعریف هرکدام از این ویژگی ها امکان رخ دادن دارد،شرح داده می شود.

مسئله حملات بر اساس انگیزه های سودجویانه و طراحی هوشمندانه پاداش دادن،از ابتدا مورد توجه بوده است. این حملات تاثیر غیرمستقیمی برروی اجماع درشبکه می گذارند.به طور مثال ممکن است با جمع کردن سود اعضای درستکار شبکه،آن ها از شبکه خارج شوند. در بعضی از مقاله ها در زمینه proof of stake،در وهله اول به امنیت شبکه در مبادله پیام ها می پردازندبعضی دیگر اثبات می کنند که با دور شدن از پروتکل اصلی شبکه،تنها مقدار کمی از کل سود به سودجویان می رسد.

در ابتدای مقاله به تعاریف ابتدایی این حوزه می پردازد و دو ویژگی اصلی این شبکه ها که مسئله اجماع و Permissionless(اختیاری بودن ورود و خروج) و pseudonymous(مدیریت اکانت بدون فاش کردن نام واقعی) را بیان می کند.سپس حمله معروف double spend شرح داده می شود ویک راه حل ابتدایی آن انتخاب یک فرد به صورت ((رندوم)) در شبکه است که تراکنش های معتبر را امتیازدهی کند.مشکل دیگری که دراین جا رخ می دهد،مسئله Sybil است که ممکن است یک فرد با توجه به ویژگی های اصلی شبکه،اکانت های بدون استفاده زیادی بسازد و شانس خود را برای انتخاب شدن بیشتر کند.پس انتخاب شدن یک فرد در شبکه،نقطه شروعی برای گسترش کار است.

در ادامه به تعریف proof of work و proof of stake می پردازد.در اولی،هر واحد پردازش معادل یک رای است و در دومی،هر سکه و به طورکلی دارایی،معادل رای دادن است.در اولی چون دور شدن از پروتکل اصلی، توان پردازشی زیادی می خواهد،انگیزه های سودجویانه به حداقل می رسد.در حالی که در دومی،توان پردازشی زیادی نیاز نیست.پس پروتکل باید هوشمندانه طراحی شود تا از حملات جلوگیری شود.

برای وارد شدن به ویژگی های اصلی پروتکل،ابتدا مفاهیم اولیه مورد نیاز مانند coin،block و ازاین دست مفاهیم تعریف می شود.برای این که ویژگی validity را در مسئله اجماع در شبکه داشته باشیم،دو فرض مهم را در اول کار می کنیم.از این جا به بعد،نماد $Pred^{D}(B)$ نماد D بلاک قبلی برای B است.

فرض chain dependence به این معناست که اگر T(B)،گرافی باشد که در راس آن بلاک B باشد و یال ها به عنوان ارتباط بلاک های پشت هم درنظر گرفته شود،validity بلاک B در زمان t،فقط به t،B و $pred\ \ (B)$ بستگی دارد.

فرض monotonicity،به این معناست که اگر بلاک B در زمان t ودرگراف T معتبر باشد،برای تمام $t^* \geq t$ و تمام گراف های T* که T زیرمجموعه آن هااست،بلاک B معتبر است.

به طور مثال اگر validity بلاک به یک منبع خارجی قابل اعتماد که قابلیت های رندوم در شبکه را ایجاد می کند بستگی داشته باشد،بنابراین پروتکل دیگر chain dependence نیست.اگر سازنده،یک بلاک B تولید کند و بلاک B* دیگری را نیز بعد آن تولید کند که با بلاک قبلی در یک هفته ساخته شده باشند،بلاک B برای تمام گراف هایی که فقط شامل B می باشد،valid است ولی برای گراف هایی که شامل هردو می باشد،invalid است.پس فرض monotocity برقرار نیست.

حال ممکن است این نکته به ذهن برسد که برقرار بودن یا نبودن این دوفرض چه مزیتی برای پروتکل دارد؟ در حمله eclipse،نودهای adversary مانع از رسیدن پیام های نودهای درستکار به یکدیگر می شوند که باعث بخش بخش شدن شبکه می شود.بنابراین اگر یک نود به اشتباه باور کند که بلاک B معتبر است،ممکن است بلاک های زیادی را در پی آن بسازد و بعد از یکپارچه شدن شبکه،تازه متوجه شود که بلاک B نامعتبر بوده است. به کمک دو فرض chain dependenceو monotocity،این مشکل به وجود نمی آید.

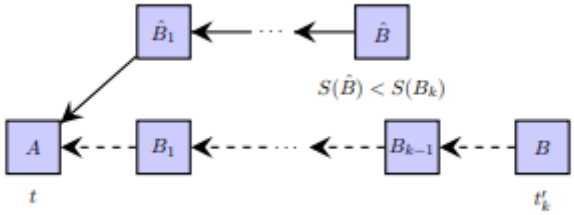
برای شرح پروتکل،دو تابع V(B) به عنوان validating function و M(A,c,t) به عنوان mining function استفاده می شود.تابع V،خروجی 0 یا 1 دارد و مقدار آن برای برای همه افراد شبکه قابل به راحتی قابل محاسبه است.c،نشانگر freezing coin می باشد به این معنی که مقدار آن در F بلاک قبلی نباید استفاده شده باشد و صاحب آن باید miner بلاک B باشد.مقدار t_B هم حتما باید در بازه $[t_{pred(B)}, t]$باشد.اگر خروجی V، 1 بوده و دو شرط قبل برقرار باشد،بلاک B معتبر است.

تابع M،فقط توسط miner که صاحب c می باشد،به راحتی قابل محاسبه است.برای هر c و t،اگر بلاک B وجود داشت به طوری که معتبر بوده و t و c متعلق به آن باشد و B=Pred(A) بوده ،پس خروجی تابع M(A,c,t) حتما بلاک معتبری مانند B* خواهد بود t و c متعلق به آن خواهند بود.

مفهوم longest chain variant به این معناست که تابع scoring برای بلاک تعریف شده که صعودی است. یعنی اگر Pred(B)=A آنگاه S(B)>S(A).اجماع هنگامی اتفاق می افتد که miner،از بین بلاک هایی که از معتبر بودن آن ها آگاه است،بلاکی را انتخاب کند که تابع scoring را ماکزیمم می کند.

حال دو ویژگی مهمی که مدنظر است،تعریف می شود.D_locally predictable به این معناست که تابع M(A,c,t) توسط مالک c به راحتی محاسبه شود به طوری که $Pred^{D}(B) = A$ و V(B)=1 و c,t متعلق به B باشند.اگر امکان محاسبه این تابع برای همه راحت باشد،آن وقت D_globally predictable است. Recency،عکس تعریف بالاست،یعنی اگر تابع بالا برای مالک c به راحتی قابل محاسبه نباشد،D_recent است.بدیهتا دو ویژگی نمی توانند با هم برقرار باشند.

با توجه به تعاریف بالا،حال حمله هایی که در برابر آن آسیب پذیر هستند را معرفی می کنیم.

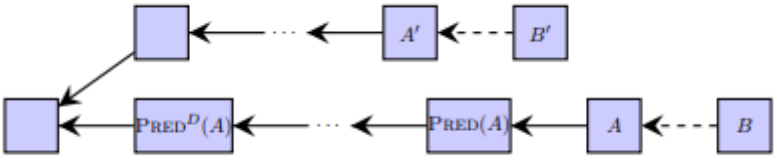


حمله Globally_predictable selfish mining به این معنی است که در در شاخه جداگانه ای شروع به ساخت بلاک B1 کنیم و هنگامی که ساخته شد،آن را منتشر نکنیم.به کار خود ادامه داده و روی بلاک B1،شروع به ساخت بلاک دیگری می کنیم در حالی که بقیه افراد شبکه در حال ساخت بلاک بر روی Pred(B1)=A می باشند.

اگر زودتر از بقیه شبکه موفق به ساخت بلاک B2 شویم،آنگاه longest chain شبکه در دست ماست.به کار خود تا زمانی ادامه می دهیم که بقیه شبکه،chain به اندازه مال ما پیدا کنند.در آن زمان chain خود را منتشر می کنیم.در این صورت chain اصلی متعلق به ما می باشد.حال ممکن است این سوال به وجود آید که ممکن است بقیه شبکه زودتر از ما بلاک معتبر ساخته و ازما جلوزده وما بلاک B1 خود را از دست بدهیم؟
به یاد داریم که فرض کردیم پروتکل ما دارای ویژگی D_globally predictable است.بنابراین برای $D \geq k$ حمله کننده توانایی پیش بینی رفتار chain اصلی را دارد و این حمله برای او ریسکی نخواهد داشت.

برای حمله بعدی ابتدا 2 مفهوم را تعریف می کنیم.honest miner کسی است که برروی بلاک A که تابع Scoring را ماکزیمم می کند،تابع M(A,c,t) را اعمال می کند و بلاک B را در صورت وجود اعلام می کند.
Provable deviation به این معناست که برای دو بلاک B1 valid و B2 با c یکسان، $t_1 = t_2$ یا اگر $t_1 < t_2$ ، S(B1)>S(Pred(B2)).

حمله بعدی،undetectable nothing_at_stake می باشد.undetectable به این معنا که اثباتی برای این که نودی در حمله شرکت کرده،در اختیار نباشد.



ابتدا فرض می کنیم که پروتکل ما دارای ویژگی D_recent است.ابتدا بلاک A را پیدا می کنیم که تابع scoring را ماکزیمم کند.سپس بلاک A* را طوری پیدا می کنیم که تابع scoring را ماکزیمم کرده و هیچ یک از بلاک های بعدی $Pred^{D}(A)$ نباشد.حال هرکدام از توابع M(A,c,t)=B و M(A*,c,t)=B* را حساب می کنیم. اگر B وجود داشت،آن را اعلام می کنیم واگر B* وجود داشت و اعلام آن مصداق provable deviation نبود،آن را اعلام می کنیم.با توجه به D_recent بودن پروتکل،معتبر بودن بلاکی که بعد از A* ساخته می شود، به بلاک های بین A* و $Pred^{D}(A)$ بستگی دارد.پس احتمال ساخته شدن آن وجود دارد.

در انتهای گزارش،بعضی از حوزه ها و مفاهیمی که ادامه کار این مقاله هستند و یا مفاهیمی مرتبط با آن داشته و تحقیقات روی آن ها قابل انجام است را بیان می کنیم.همچنین راه حل هایی برای مشکلات به وجود آمده ارائه می دهیم.

پروتکل های fruitchain و tezaos طوری طراحی شده اند که در برابر predictable selfish mining مقاوم باشند،به این صورت که بلاک ها بعد از mine شدن،نیاز دارند تا توسط اکثریتی از شبکه support شوند.پروتکل snow_white اثباتی را ارائه می دهد مبنی براین که دور شدن از پروتکل،تنها کسر کوچکی از پاداش کلی را به فردسودجو می رساند.برای جلوگیری از حمله predictable double spend،می توانیم زمان تائید شدن یک تراکنش را طولانی تر کنیم،یعنی حتما تعداد معنی بلاک باید بگذرد.برای مثال پروتکل Ouroboros زمان تائید را هر 148 دقیقه پیشنهاد می دهد.

برای جلوگیری از undetectable nothing_at_stake،راه حل اول این است که پروتکل را به گونه طراحی کنیم که D_locally detectable باشد و D،مقدار بزرگی باشد و دارای checkpoint باشد.یکی از موارد تحقیق،آنالیز این راه حل از لحاظ امنیتی است.در پروتکل Algorand،به جای longest chain،از اجماع Byzantine استفاده کنیم.اگرچه که این راه حل در برابر حمله eclipse آسیب پذیر است،ولی می توان نشان داد که تحت فرض هایی،احتمال ایجاد fork قابل اغماض است.راه حل سوم که به dunkles مشهور است،می گوید که miner بلاک هایی که orphan می شوند،باید مجازات شوند.این کار،انگیزه برای فعالیت در chain اصلی را کاهش می دهد.البته ممکن است که به اشتباه بعضی از نودهای درستکار مجازات شوند.

اگرچه که آسیب پذیری های زیادی برای دو ویژگی گفته شده در متن مقاله ارائه شده است،ولی هیچ کدام آنالیز شفافی ندارند.سه حمله ای ذکر شد،به اندازه کافی مهم هستند که مقاله های بعدی در زمینه proof of stake باید به آنالیز دقیق و راه حل های جلوگیری از آن ها بپردازد.