بسم الله الرحمن الرحیم

بلاکچین

دکترمحمدعلی مداح علی

پرهام محمدی ۹۶۱۰۲۳۴۲

امیرحسین رستمی ۹۶۱۰۱۶۳۵

گزارش تفصیلی پروژه تحقیقاتی

زمستان 99

در این مقاله به بررسی عمیق تر امنیت شبکه بیت کوین می پردازیم،همانطور که می دانیم در backbone protocol امنیت در شبکه بیت کوین به شرطی برقرار بود که strictly در هرلحظه داشته باشیم که β کمتر از 0.5 باشد اما در این مقاله قرار است یک پله حساسیت را پایین تر بیاوریم و شرط اینکه در **هرلحظه پارامتر** β کمتر مساوی 0.5 باشد به شرایطی تقلیل کند که نیاز باشد که روابط امنیتی برای Expected توان های پردازشی برقرار باشد(اینکه این روابط تغییریافته دقیقا چه باشند را در ادامه اثبات و مطرح می کنیم)و این یعنی مثلا اینکه برای لحظاتی پارامتر بتا از 0.5 بیشتر شود مشکلی برای امنیت شبکه ایجاد نمی کند.مثلا در سال 2014 یک mining pool به نام GHash.io 54 درصد از توان پردازشی شبکه را برای مدتی تصاحب کرد ولی همچنان شبکه امن باقی ماند.

هم چنین علاوه بر فرضیات backbone protocol، در این مقاله حالتی که ما node های sleepy داشته باشیم نیز مورد بررسی و کاوش قرار میگیرد.یک نودِ sleepy در اصل نود های صادقی اند که در حال حاضر توانایی دنبال کردن پروتکل را ندارند و این عدم توانایی می تواند در اثر حمله denial of service نود های متخاصم به نود های صادق باشد.

همانطور که مشخص است در این فرایض برای مساله، نود های متخاصم دو توانمندی جهت تخریب عملکرد سالم شبکه دارند:

1- استفاده از توان پردازشی جهت mine کردن بلوک ها برای حملاتی از قبیل double spend
2- استفاده کردن از توان پردازشی جهت sleepy کردن برخی از نود های سالم.

و نود متخاصم می تواند بسته به شرایط از هر یک از موارد ذکر شده جهت آسیب زدن به شبکه استفاده کنند،زیرا همان نود های sleepy به علت نقص اطلاعشان از وضعیت بلاکچین اگرچه بالقوه صادق اند اما به علت نقص اطلاع می توانند در جهت ارضای امیال متخاصم پیش بروند.

همانطور که می دانید تاخیر در شبکه های بلاکچینی پارامتر خیلی مهمی است و روی سنکرون/آسنکرون بودن مساله تاثیر بسیاری دارد.در این مقاله مساله را در فضای semi آسنکرون حل می کنیم که به این معنا است که تاخیر در شبکه یک حداکثرِ متناهی دارد.

علاوه بر درنظرگرفتن موارد فوق داریم که برخلاف backbone protocol داریم که در این مقاله message loss داریم و این بدین شرح است که همانطور که گفتیم برخی نود های متخاصم می توانند با حمله denial of service برخی از نود های صادق شبکه را به خواب ببرند و به این ترتیب به سبب وجود فرض message loss در شبکه می توانند در دیدِ نود های sleepy از بلاکچین تاثیر داشته باشند و بدین ترتیب از آنها به نوعی به عنوان مهره برای اعمال خصمانه خود استفاده کنند.

در این مقاله به دنبال تعمیق در امنیت بیت کوین با بررسی تاثیرِ فرایض و شرایط فوق ایم.در ابتدا نیاز است تا مدلمان از مساله را مطرح کنیم و سپس در این مدل به حل و فصل بپردازیم.

مدل:

مدل مان در این مقاله بنیادا برپایه مدلی است که در مقاله backbone protocol مطرح گردید،است.حال به معرفی اجزای دقیق مدل می پردازیم تا وارد حل مساله شویم:

1- Execution
2- Sleepy, Alert and corrupted
3- Parametrized Model
4- properties

بخش اول(Execution):

در این فرض میکنیم تعداد نود های شرکت کننده در Backbone protocol برابر n اند(این عدد ثابت است).هر شرکت کننده می تواند یکی از سه حالتِ زیر باشد:

1- alert
نود صادقی که روشن بوده و فعالانه در پروتکل شرکت می کند.
2- Corrupted
نودِ متخاصم.
3- Sleepy
نود صادقی که بنا به دلیلی توانایی دنبال کردن پروتکل را ندارد(مثلا در اثر قربانی حمله denial of service شدن)

Involved programs:

تمام program ها به صورت ماشین های تورینگ تعامل پذیری که ازلحاظ زمان اجرای محدودیت اجرایِ چندجمله ای دارند مدل می شوند.(Polynomial bounded interactive Turing machines).می دانیم که تورینگ ماشین ها به صورت عادی فقط یک نوار دارند که در آن ورودی نوشته می شود (در ابتدا) و سپس روی آن جاروب زده و الگوریتم را اجرا می کند.هم چنین می دانیم که ماشین های تورینگ چند نواره،مطلقا معادل اند با ماشین های تورینگ تک نواره.ماشین های تورینگی که ما در این مدل برای program های درنظر میگیریم،ماشین های تورینگ 3 نواره ای اند که یک نوار ورودی،یک نوار خروجی و یک نوار تعامل دارند.

در این مدل یک ITM به نام Z داریم که در اصل مدل کننده environment program است که اجرای backbone protocol را بر عهده دارد،لذا این Z به تعداد n نمونه(تعداد شرکت کننده ها در پروتکل)ماشین تورینگ تعامل پذیر که بیانگر شرکت کننده مربوطه اند را اجرا می کند.

این n عدد ITI را با نام $p_1, p_2, \dots, p_n$ نشان می دهیم.

واحد کنترلی C که خود یک ITM است،اجرای و عملکرد این ITI ها و تعامل بینشان را کنترل می کند.این واحد،محیطِ Z را مجبور می کند که در ابتدا یک ITI ِ متخاصم به نام ِ A ایجاد کند(نمایانگر واحد های متخاصم).

Z به این صورت عمل می کند که هر کدام از شرکت کننده هارا به صورت یک حلقه (round-robin) فعال می کند و این فعال سازی به صورت نوشتن در نوار ورودی آن ها انجام می گیرد.

نکته1:هرگاه یک نودِ متخاصم فعال شود،به جایِ آن،A فعال می شود(به نوعی A یک wrapper برای نود های متخاصم است).

نکته2:نود های متخاصم به واحد کنترل مرکزی می توانند پیغامِ (Corrupt,Pi) بفرستند که به این معناست که واحد کنترلی Pi را به عنوان یک متخاصم در شبکه register کند.(مادامی که تعداد corrupt ها < تعداد شرکت کننده ها)

نکته3:یک متخاصم می تواند پیغامِ (sleep,Pi) به واحد مرکزی بفرستد که بدین معناست که واحد مرکزی نود Pi را به خواب ببرد برای مرحله بعد(مدل سازی حمله dos توسط فرستنده این پیغام به نود صادق Pi) و واحد مرکزی با احتمال s این کار را انجام می دهد ولی A را از اینکه به خواب رفته است یا نه با خبر نمی کند.توجه کنید که این باخبر نکردن خیلی نکته مهمی است چون مثلا در حالت عادی هم که ما به سرور حمله dos می کنیم داریم که متوجه نمی شویم بالاخره down شده است یا نه(ممکن است به درخواست های ما جواب ندهد ولی این لزوما به معنی down شدن اش نیست) لذا وقتی یک نود متخاصم یک نود دیگر را می خواهد به خواب برود از نتیجه اطلاع قطعی نمی یابد.توجه کنید که این درخواست asleep کردن **فقط از جانب نود های متخاصم** اعلام می گردد.

هر شرکت کننده به دو functionality دسترسی دارد که هر کدام به صورت یک ITM مدل می شوند.

1- Random oracle

نقش این functionality به نوعی همان محاسبه Hash است که در حالت ایده آل به هر ورودی یک عدد رندوم نسبت می دهد.

$$H(\cdot) : \{0,1\}^* \to \{0,1\}^\kappa$$

از backbone،سناریوی PoW را به خاطر دارید،احتمال ارضا کردن y(خروجی تابع هش فوق) به صورت زیر است:(که K همان پارامتر امنیت شبکه است).

$$p = Pr[y < T] = \frac{T}{2^\kappa}$$

همانطور که انتظار می رود با افزایش K این احتمال به صورت نمایی کاهش می یابد.

نکته: هر شرکت کننده در این مدل می تواند تعداد نامتناهی تا درخواست **احراز درستی هش** به RO بدهد و حداکثر q درخواست **محاسبه هش**(طبیعتا هر شرکت کننده از تمامی q سهمیه درخواست هش اش استفاده می کند و می توان گفت که انگار q درخواست هر شرکت کننده در هر round دارد).

2- Diffusion channel

این functionality به نوعی communication بین شرکت کننده هارا مدل می کند.هر شرکت کننده یک string ای دارد که می تواند در هر لحظه آن را بخواند و در اصل این string مدل کننده صندوق دریافت کننده پیغام ها است.

نکته: این امکان به A (واحد متخاصم) در این مدل داده شده است که این واحد تمامی پیغام هایی که توسط نود pi ارسال شده است را بخواند بدون اینکه قادر باشد آن را تغییر بدهد یا drop کند یا دچار تاخیر کند.

نکته:نود ها هنگامی که پیغام empty می فرستند بیان می کنند که کارشان در این round تمام شده است و پس از دریافت این پیغام به عنوان complete نشاندار می شوند.

هنگامی که همه node ها به عنوان complete نشان دارس شدند داریم که این functionality همه پیغام هایی که به اندازه delta مرحله از ایجاد آن ها گذشته است را string می نویسد.

نکته مهم:همانطورکه مشخص است اگر message loss قرار باشد اتفاق بیافتد باید در اثر عملکرد این functionality رخ بدهد و این رخ داد به شرح زیر است:

این functionality یک پارامترِ Boolean دارد به نام B که اگر:

اگر B = 1 باشد داریم که این functionality همه پیغام هارا در رشته ِ Receive() **همه شرکت کننده ها** می نویسد و این یعنی در این حالت message loss نداریم اما اگر B = 0 باشد داریم که این functionality همه پیغام هارا **فقط در رشته Receive() نود های Alert می نویسد**.(و یعنی نود های sleepy دچار message loss می شوند).

بخش دوم(Sleepy, Alert and corrupted):

داریم که از کل تعداد نود های شبکه به تعداد t عدد نود متخاصم داریم و لذا داریم که تعداد کل نود های صادق برابر است با n-t و از این عدد نود صادق داریم که تعدادی sleepy اند و تعدادی هم alert اند.

در هر round به شماره i داریم که:

$$n_{honest,i} = n_{alert,i} + n_{sleepy,i}$$

هر نود صادق هم به احتمال s ، sleepy است و لذا داریم که متغیرتصادفیِ $n_{sleep,i}$ که بیانگر تعداد نود های sleepy در Round شماره i است یک متغیرتصادفی نمایی با پارامترهای n-t و s است.

هم چنین داریم که $n_{alert,i}$ که بیانگر تعداد نود های alert در Round شماره i است یک متغیرتصادفی نمایی است با پارامترهای n-t و 1-s است.

distributed with parameters $(n-t)$ and $(1-s)$. Hence, $E[n_{sleepy}] = s \cdot (n-t)$ and $E[n_{alert}] = (1-s) \cdot (n-t)$.

بخش سوم(Parametrized Model):

مدل ما با توجه به تعاریفی که تا الان داده شده است به صورت $M(q,\Delta,B)$ است که:

- q بیانگر تعداد درخواست هایِ محاسبه hash به RO است در هر round.
- Delta بیانگر همان تاخیر شبکه در انتقال اطلاعات است.
- B همان پارامتر diffuse functionality است.

بخش چهارم(properties):

در این بخش به بیان 6 تعریف مهم که در مقاله مطرح گردیده است می پردازیم:

تعریف اول:

فرض کنید که فرضیه Q و متغیر های طبیعیِ متناهی q,t,n (t<n) را داریم.
زمانی می گوییم که backbone protocol فرضیه Q را در مدل $M(q, \Delta,B)$ با n عدد شرکت کننده و حداکثر t عدد متخاصم،ارضا می کند که داشته باشیم در مدت زمان متناهی احتمال اینکه (این مدل با پارامتر های ذکرشده)Q غلط شود با نرخ K قابل صرف نظر کردن باشد.(یعنی هرچه K بیشتر باشد قابل صرفنظر کردن تر باشد).

تعریف دوم ChainGrowth

پارامتر Q_{cg} با پارامتر های طبیعی c و پارامتر حقیقی Tau درنظر بگیرید.حال یک شرکت کننده صادق P با زنجیره C را درنظر بگیرید،داریم که در هر s+1 مرحله(round)حداقل Tau*s عدد بلاک به زنجیره P اضافه شده است.

تعریف سوم CommonPrefix

پارامتر Qcp با پارامتر طبیعی k بیان می دارد که برای هر جفت شرکت کننده های صادق P_1, P_2 که زنجیره های C_1, C_2 را در round های $r_1 \leq r_2$ دارند،داریم که $C1'(removed\ last\ k\ blocks) \leq C$.

تعریف چهارم ChainQuality

پارامتر Q_{cq} با پارامتر های "میو" که یک عدد حقیقی است و l که یک عدد طبیعی است بیان می دارد که برای هر شرکت کننده صادق P با زنجیره C داریم که برای هر l بلاک متوالی از C داریم که نرخ بلاک های متخاصم حداکثر برابر "میو" است.

تعریف پنجم: prediction & Insertion & copy

یک زنجیره C و دو بلوک متوالی B,B' در این زنجیره را درنظر بگیرید زمانی می گوییم:

تزریق یا insertion انجام گرفته است که یافته شود بلوک B^* که اولاهشِ بلوک ماقبل آن همان هش بلوک B باشد و هش خودش هم همان هش بلوک B باشد که یعنی به نوعی بتواند بین دو بلوک B و B' قرار بگیرد.

کپی زمان اتفاق می افتد که یک بلوک در دو محل مختلف از زنجیره قرار گرفته باشد و prediction زمانی رخ می دهد که ناعلیتی در PoW ِ فرآیندِ mining اتفاق بیوفتد یعنی بلاکی اضافه شود ولی بعدا بها mining اش داده شود.

تعریف ششم: typical execution

یک execution با پارامتر های (η,ε) یک typical execution است هرگاه داشته باشیم که برای هر مجموعه متوالی از round های S که داشته باشیم $|S| \geq \eta\kappa$ و X یک متغیر تصادفی روی S باشد داشته باشیم:

1- $(1-\varepsilon)E[X(s)] < X(S) < (1+\varepsilon)E[X(s)]$
2- Insertion و copy و prediction نداریم.

نکته: یک execution از نوع typical execution است با احتمال $1 - e^{-O(\kappa)}$.

حال که با جزییات تعاریف مورد استفاده در مقاله آشنا شدیم امنیت را در 3 فاز بررسی می کنیم:

1- M (q, 0, 1) <- سنکرون،q درخواست بر round توسط هر node و بدون امکان فقدان پیغام.
2- M (1, Δ, 1) <- نیمه آسنکرون با حداکثر تاخیر دلتا،1 درخواست بر round توسط هر node و بدون امکان فقدان پیغام.
3- M (q, 0, 0) <- سنکرون با q درخواست بر round توسط هر node و وجود امکان فقدان پیغام.

فاز اول(M (q, 0, 1 :

مدل ما به شکل زیر است:

1- اولا q bounded است.
2- دوما مدل سنکرون است.
3- سوما message loss نداریم.

تعریف می کنیم که یک round موفق است اگر حداقل یک node صادق در آن موفق به ارضای PoW شود.متغیر تصادفی Xi را تعریف می کنیم:

اگر round شماره i موفق باشد -> Xi = 1

اگر round شماره i موفق نباشد -> Xi = 0

حال فرض کنید که S یک مجموعه ای از round ها باشد داریم که X(S) را به صورت زیر تعریف می کنیم:

$$S:\ X(S)\ =\ \sum_{i\in S} X_i.$$

لم شماره صفر:

حال توجه کنید که اگر ما هیچ nodeِ به خواب رفته نداشته باشیم داریم که:

$$E[X_i] = Pr[X_i = 1] = 1 - (1-p)^{q(n-t)}$$

رابطه فوق بدین صورت محاسبه می شود که طبق قانون متمم احتمال اینکه round دلخواه i موفق باشد برابر است با یک منهای احتمال اینکه هیچ موفقیتی در PoW های node های صادقِ این round انجام نگیرد و می دانیم که اگر هیچ تعداد sleepy نداشته باشیم داریم که تعداد node های صادق برابر است با n-t و داریم که هر node به تعداد q عدد درخواست محاسبه هش می دهد و می دانیم که احتمال موفقیت هر درخواست هش برابر است با p و لذا داریم:

$$total\ number\ of\ requests = q(n-t)$$

$$possibility\ of\ a\ success\ in\ PoW = p = p[y < T] = \frac{T}{2^K}$$

$$possibility\ of\ no\ success\ during\ all\ requests = (1-p)^{q(n-t)}$$

$$possibiliy\ of\ at\ least\ one\ success\ at\ round\ i = E[X_i] = 1 - (1-p)^{q(n-t)}$$

لم شماره یک:

$$It\ holds\ that\ \frac{pqE[n_{alert}]}{1+pqE[n_{alert}]} \leq E[X_i] \leq pqE[n_{alert}].$$

در اثبات قضیه به ترتیب نکات زیر استفاده شده است:

1- استفاده از قانون احتمال کل و شکستن E[Xi]
2- استفاده از لم صفر جهت محاسبه $E[X_i|n_{alert,i} = k]$
3- می دانیم $n_{alert,i}$ یک متغیرتصادفی binomial است و لذا $\Pr[n_{alert,i} = k]$ به سادگی محاسبه می گردد.
4- استفاده کردن از نامساوی برنولی:

$$bernouli: (1+a)^n \geq 1 + na$$

به کمک نکات گفته شده به طریق زیر اثبات را انجام می دهیم:

$$\begin{aligned}
E[X_i] &= \sum_{k=0}^{n-t} E[X_i|n_{alert,i} = k] \cdot Pr[n_{alert,i} = k] \\
&= \sum_{k=0}^{n-t} \left(1-(1-p)^{qk}\right) \cdot \binom{n-t}{k} (1-s)^k s^{n-t-k} \\
&= 1 - \left(s-(s-1)(1-p)^q\right)^{n-t} \geq 1 - \left(s-(s-1)(1-pq)\right)^{n-t} \\
&\geq 1 - e^{-(1-s)(n-t)pq} = \frac{pqE[n_{alert}]}{1+pqE[n_{alert}]}
\end{aligned}$$

حال تعریف میکنیم که round شماره i یک unique successful است هرگاه در این round **فقط** یک node صادق وجود داشته باشد که در PoW موفق شود.متغیر تصادفی Yi را تعریف می کنیم:

اگر round شماره i unique successful باشد -> Yi = 1

اگر round شماره i unique successful نباشد -> Yi = 0

حال فرض کنید که S یک مجموعه ای از round ها باشد داریم که Y(S) را به صورت زیر تعریف می کنیم:

$$\text{let } Y(S) = \sum_{i \in S} Y_i.$$

لم شماره دو:

$$It\ holds\ E[Y_i] = E[pqn_{alert,i}(1-p)^{q(n_{alert,i}-1)}] \geq E[X_i](1-E[X_i]).$$

اثبات:

ابتدا به کمک قضیه برنولی داریم که:

$$E[Y_i] = E[pqn_{alert,i}(1-p)^{q(n_{alert,i}-1)}] \geq E[pqn_{alert,i}(1-pq(n_{alert,i}-1))]$$

حال نیاز است تا رابطه زیر را جهت تکمیل اثبات انجام دهیم:(*)

$$pqE[n_{alert}](1-pqE[n_{alert}]) \geq E[X_i](1-E[X_i]).$$

به کمک باند بالایِ E[Xi] داریم که می توان نوشت:

$$E[X_i] = pqE[n_{alert}] - b.$$

و داریم که b بزرگتر مساوی صفر است،حال به کمک رابطه فوق سمت راست(*) را باید تکمیل کنیم

$$\begin{aligned} E[X_i](1-E[X_i]) &= (pqE[n_{alert}] - b)(1 - pqE[n_{alert}] + b) \\ &= pqE[n_{alert}](1 - pqE[n_{alert}]) - b^2 - b + 2pqE[n_{alert}]b \end{aligned}$$

حال به کمک روابط نوشته شده جهت ارضا نامساوی لم 2 نیاز است تا داشته باشیم:

$$0 \geq -b^2 - b + 2pqE[n_{alert}]b\ which\ is\ equivalent\ to\ 1 \geq E[X_i] + pqE[n_{alert}]$$

و رابطه نوشته شده درصورتی که $1 \geq 2E[X_i]$ برقرار باشد ارضا می گردد و بنابراین تا به اینجای کار داریم که جهت ارضا شدن لم2 نیاز است تا $1 \geq 2E[X_i]$ و داریم که در بیت کوین نتایج آماری نشان می دهد که مقدار E[Xi] حدودا بین 2 الی 3 درصد است و لذا این شرط ارضا می شود حال جهت جمع بندی داریم که:

$$\begin{aligned} &E[pqn_{alert,i}(1-pq(n_{alert,i}-1))] && \geq pqE[n_{alert}] - (pq)^2 E[n_{alert}]^2 \\ \Leftarrow &E[{n_{alert}}^2] - E[n_{alert}] && \leq E[n_{alert}]^2 \end{aligned}$$

آخرین رابطه فوق معادل است با:

$$E[n_{alert}^2] - E[n_{alert}]^2 \leq E[n_{alert}]$$

و طبق تعریف واریانس داریم که رابطه فوق معادل است با:

$$Var[n_{alert}] \leq E[n_{alert}]$$

همانطورکه می دانید داریم که در توزیع binomial با پارامتر های n,p داریم که:

$$E[X] = np, Var[X] = np(1-p), 1 \geq p \geq 0 \rightarrow Var[X] \leq E[X]$$

هم چنین داریم که n_{alert} هم توزیع binomial دارد و لذا اثبات لم تمام است.

حال که متغیرتصادفی های XوY را تعریف کردیم و لم های مربوطه شان را اثبات کردیم می رویم سراغ متغیر تصادفی Z و لم مربوطه اش.

تعریف می کنیم Z_{ijk} برابر ۱ هست هرگاه PoW در round شماره i ام توسط درخواست j ام از درخواست های node متخاصم شماره k ام رخ بدهد،در غیراینصورت آن را برابر 0 میزنیم.حال از روی این متغیرتصادفی،متغیرِ تصادفی Z_i را می سازیم:(که وابستگی صرفا به i دارد که شماره Round است)

$$Z_i = \sum_{k=1}^{t} \sum_{j=1}^{q} Z_{ijk}$$

و حال داریم برای یک مجموعه ای از round های S که:

$$Z(S) = \sum_{i \in S} Z_i$$

می دانیم که تعداد متخاصم ها برابر است با t و داریم که احتمال موفقیت در PoW برابر است با p.لذا داریم که تعداد block هایی که متخاصم در یک round شماره i می تواند mine کند برابر است با:

$$Z_i \text{ is binomial} \rightarrow E[Z_i] = q_{\frac{||Query||}{party}} \times p_{PoW\ condition\ meet} \times t_{number\ of\ adversaries}$$

$$E[Z_i] = qpt = \frac{t}{E[n_{alert}]} pqE[n_{alert}] \leq \frac{t}{E[n_{alert}]} \cdot \frac{E[X_i]}{1-E[X_i]}$$

همانطورکه پیشتر مطرح گردید،در این مقاله ما فرض اکثریت صادقین را برای حالت Expected آن مطرح می کنیم یعنی برخلاف مقاله backbone که اکثریت دایمِ صادقین را می طلبید در این مقاله امنیت بیت کوین را برای حالت expected آن اثبات می کنیم،در این فرضیه در هر round چنین فرضی برقرار است:

$$t \leq c \cdot (1-\delta) \cdot E[n_{alert}]$$

که c برابر است با نسبت توان متخاصم به توان صادق و داریم که اگر داشته باشیم که δ در شرط زیر صدق کند در این صورت امنیت داریم:

$$\delta \geq 2E[X_i] + 2\varepsilon$$

حال می خواهیم به این ترتیب یک upper bound برای s که احتمال sleepy بودن یک nodeِ متخاصم است بیابیم.

داریم که:

$$numberOfSleepy = (n-t)_{||HonestNodes||} - E\big[||HonestNodes||\big]$$

$$s \le \frac{numberOfSleepy}{||Honest||}$$

حال داریم که طبق نتایج به دست آمده داریم:

$$s \le \frac{n-t-\frac{t}{c(1-\delta)}}{n-t} = 1 - \frac{1}{c(1-\delta)}\frac{t}{n-t}$$

حال برویم سراغ آنالیز امنیت این فاز:

طبق تعریف ۶ داریم که شرایطِ typical execution برای متغیر های تصادفی X(S),Y(S),Z(S) برای حالتی که داشته باشیم $|S| \ge \eta\kappa$ برقرار است.

لم شماره سوم:

بخشک اول:

$$(1-\epsilon)E[X_i]|S| < X(S) < (1+\epsilon)E[X_i]|S|$$

اثبات:طبق برقراری typical execution برای X(s) داریم که:

$$(1-\varepsilon)E[X(s)] < X(S) < (1+\varepsilon)E[X(s)]$$

$$Since\ X_i s\ are\ independent \rightarrow\ E[X(S)] = |S|E[X_i]$$

لذا با جایگذاری موارد فوق داریم که صورت لم به دست می آید و اثبات تمام است.

بخشک دوم:

$$(1-\epsilon)E[X_i](1-E[X_i])|S| < Y(S)$$

اثبات:

$$Since\ Y_i s\ are\ independent \rightarrow E[Y(S)] = |S|E[Y_i] \ge |S|E[X_i](1-E[X_i])$$

و اثبات تمام است.

بخشک سوم:

$$Z(S) < (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \le c(1+\epsilon)(1-\delta)\frac{E[X_i]}{1-E[X_i]}|S|$$

اثبات:

همانطور که به خاطر دارید رابطه زیر را پیشتر اثبات کردیم:(رابطه i)

$$E[Z_i] = qpt = \frac{t}{E[n_{alert}]} pqE[n_{alert}] \leq \frac{t}{E[n_{alert}]} \cdot \frac{E[X_i]}{1 - E[X_i]}$$

داریم که:

$$Since\ Z_i s\ are\ independent \rightarrow\ E[Z(S)] = |S|E[Z_i]$$

$$(1 - \varepsilon)E[Z(s)] < Z(S) < (1 + \varepsilon)E[Z(s)]$$

طبق رابطه i و نکته فوق داریم که:

$$Z(S) < (1 + \varepsilon)\frac{t}{E[n_{alert}]} \frac{E[X_i]}{1 - E[X_i]} |S|$$

هم چنین داشتیم که:

$$t \leq c \cdot (1 - \delta) \cdot E[n_{alert}]$$

و بدین ترتیب نابرابری دوم هم اثبات می گردد و اثبات تمام می شود.

بخشک چهارم:

$$For\ \sigma = (1 - \epsilon)(1 - E[X_i]):$$

$$Z(S) < \left(1 + \frac{\delta}{\sigma}\right)\frac{t}{E[n_{alert}]} X(S) \leq c\left(1 - \frac{\delta^2}{2\sigma}\right)X(S)$$

با استفاده از متغیر معرفی و جایگزینی در بخشک قبلی نامساوی های فوق به سادگی به دست می آید.

بخشک پنجم:

$$Z(S) < Y(S)$$

اثبات: به کمک بخش های b,c,d داریم که سمت چپ نابرابری ای که سمت راست آن Y(S) قرار دارد از سمت راست نابرابری ای که سمت چپ آن Z(S) قرار دارد بیشتر است لذا طبق تعدی داریم که خاصیت فوق ارضا می گردد.

حال که مقاله جزییات لم ها و مطالب مربوط به این متغیر های تصادفی را بیان و اثبات کرد در ادامه گفت که به کمک جزییات property هایی که در مقاله [7] مطرح گردیده است و جزییات ذکرشده تا به الان در این مقاله،امنیت شبکه بیت کوین را تحت temporary Dishonest majority اثبات می کند.

آن سه ویژگی مطرحی در [7] عبارت انداز:

رشد زنجیره(chain growth)،پیش رشته ی مشترک(common prefix) و کیفیت زنجیره(chain quality).برای جزییات بیشتر به نسخه full version مقاله فعلی ارجاع می دهیم.

فاز دوم $M(1, \Delta, 1)$

مدل ما به شکل زیر است :

1. اولاً در هر round تعداد query ، 1 است.
2. مدل دارای تأخیر شبکه Δ واحد است.
3. در این حالت message loss نداریم.

در این فاز می‌خواهیم تأثیر وجود تأخیر در شبکه را در امنیت بررسی کنیم. در حالت عادی تأخیر شبکه باعث بی خبری honest ها از یکدیگر و بوجود آمدن Fork در شبکه می‌شود. حال می‌خواهیم نشان دهیم که در حالت شرایط جدید تعریف شده در این مقاله نیز امنیت با چه شرطی برقرار خواهد بود. از طرفی در این حالت message loss نیز وجود ندارد و نودهای sleepy بعد از alert شدن ، می‌توانند از اتفاقات افتاده قبلی باخبر شوند.

حال به اثبات امنیت در شرایطی که بعداً گفته می‌شود می‌پردازیم. برای این منظور در مرحله‌ی نخست یک متغیر به نام X_i' تعریف می‌کنیم : این متغیر هنگامی 1 است که $X_i = 1$ و به ازای هر

$i - delta + 1 < j < i - 1$ گزاره $X_j = 0$ برقرار باشد. در غیر اینصورت نیز 0 است.

همچنین رو یک مجموعه زمانی S تعریف می‌کنیم : $X'(s) = \sum_S X_i'$

سپس یک متغیر دیگر به نام Y_i' تعریف می‌کنیم : این متغیر هنگامی 1 است که $X_i = 1$ و به ازای هر

$i - delta + 1 < j < i + delta - 1$ گزاره $X_j = 0$ برقرار باشد. در غیر اینصورت نیز 0 است.

همچنین رو یک مجموعه زمانی S تعریف می‌کنیم : $Y'(s) = \sum_S Y_i'$

همان‌طور که از اثبات امنیت در حالت تأخیردار به یاد داریم ، دو نوع بلوک تحت عنوان NT و loner تعریف می‌شدند. همان‌طور که مشاهده می‌شود 1 بودن متغیر X_i' معادل NT بودن آن بلوک است چرا که به اندازه تأخیر شبکه قبل ، هیچ بلوک honest ای mine نشده است. 1 بودن متغیر Y_i' معادل loner بودن آن بلوک است چرا که به اندازه تأخیر شبکه قبل و هنچنین بعد از آن ، هیچ بلوک honest ای mine نشده است. تنها تفاوت این دو ، مدل گسسته‌ای است که در این مقاله در نظر گرفته‌ایم در صورتی که در درس مدل پیوسته داشتیم.

حال بعد از تعریف این دو متغیر به سراغ ادامه اثبات خواهیم رفت :

ابتدا به کمک برنولی یک کران پایین برای $E(X_i')$ بدست می‌آوریم. :

$$E[X_i'] = E[X_i](1 - E[X_i])^{\Delta-1} \geq E[X_i](1 - (\Delta - 1)E[X_i]).$$

توان $\Delta - 1$ به واسطه شرط X_i' منوط به Δ-1 round قبلی است.

سپس به طور مشابه به کمک برنولی یک کران پایین برای $E(Y_i')$ بدست می‌آوریم :

$$E[Y_i'] = E[X_i](1 - E[X_i])^{2\Delta-1} \geq E[X_i](1 - (2\Delta - 1)E[X_i]).$$

توان $2\Delta - 1$ به واسطه شرط X_i' منوط به Δ-1 round قبلی و Δ-1 round بعدی است.

در این جا شرطی که باید برقرار باشد تا امنیت برقرار بماند ، به شرح زیر است. این شرط نشان می‌دهد که همچنان می‌توان اکثریت موقتی Dishonest ها را تحمل کرد :

$$t < c\ (\ 1 - \ \delta\ )E(n_{alert}) \quad ; \quad \delta \geq 2\Delta E(X_i) + 4\epsilon + \frac{4\Delta}{\eta . k}$$

همان‌طور که مشاهده می‌شود ، δ باید مقدار بزرگتری نسبت به حالت قبل داشته باشد. حال term های موجود در آن را آنالیز می‌کنیم :

در Term اول مقدار Δ در $2E(X_i)$ ضرب می‌شود. اینکه چند نود Honest در یک round به بلوک برسند ، موجب ایجاد Fork می‌شود. حال در صورتی که تأخیر شبکه را نیز در نظر بگیریم ، در Δ round متوالی این اثر پخش می‌شود و حساسیت شبکه به fork ، Δ برابر می‌شود.

عبارت دوم نیز دو برابر شده است. یعنی ضریب ϵ از 2 به 4 تغییر کرده است. علت این موضوع این است که اگر نوسانات حول میانگین بتواند دامنه وسیع‌تری داشته باشد ، در این حالت تأثیر بیشتری بزارد. زیرا اثرات round ها به واسطه تأخیر در هم پخش می‌شود و محدودیت بیشتری ایجاد می‌کند.

Term سوم نیز جدید است و تنها در این مدل پدیدار می‌شود. این به واسطه fork هایی است که از یک round تا یک round دیگر پدید می‌آید. فاصله این round ها می‌تواند تا Δ باشد. این به این علت است که بی خبری نودها نسبت به یکدیگر می‌تواند چند round ادامه یابد.

حال به اثبات 5 لم می‌پردازیم تا نشان دهیم ، امنیت با فرض شرط بالا ، برقرار می‌شود.

$$(1 - \epsilon)E[X_i](1 - E[X_i])^{\Delta-1}|S| < X'(S)$$

برای اثبات این رابطه کافی است شرط اول typical Execution را برای $X'(S)$ بنویسیم و سپس کران پایین بدست آمده در قسمت قبل به کمک برنولی را روی آن اعمال کنیم.

$$(1-\epsilon)E[X_i](1-E[X_i])^{2\Delta-1}|S| < Y'(S)$$

برای اثبات این رابطه کافی است شرط اول typical Execution را برای Y′(S) بنویسیم و سپس کران پایین بدست آمده در قسمت قبل به کمک برنولی را روی آن اعمال کنیم.

$$Z(S) < (1+\epsilon)\frac{t}{E[n_{alert}]}\frac{E[X_i]}{1-E[X_i]}|S| \leq c(1+\epsilon)(1-\delta)\frac{E[X_i]}{1-E[X_i]}|S|$$

این عبارت مشابهاً همانند قسمت قبل بدست آمده است.

Let $S' = \{r, \ldots, r'\}$ *with* $|S'| \geq \eta\kappa$. *For* $S = \{r, \ldots, r'+\Delta\}$ *and* $\sigma' = (1-\epsilon)(1-E[X_i])^{\Delta}$:

$$Z(S) < \left(1+\frac{\delta}{2\sigma'}\right)\frac{t}{E[n_{alert}]}X'(S')$$

این عبارت همان حالت گسسته لمی است که در درس برای بلوک‌های NT اثبات کردیم.

Let $S' = \{r, \ldots, r'\}$ *with* $|S'| \geq \eta\kappa$. *For* $S = \{r-\Delta, \ldots, r'+\Delta\}$:

$$Z(S) < Y'(S')$$

این

قضیه (تمرین): اگر $g^2\lambda_h > (1+\delta)\lambda_a$ و ...

$r \leq s$ $\quad G_{r,s} = \{\forall u \leq r, \forall v \geq s \quad Z_{u,v} \leq Y_{u+\Delta, v-\Delta}\}$

آنگاه $G_{r,s}$ درست است مگر با احتمال ناچیز $e^{-\Omega(\delta^2 g^2 \lambda_h (s-r))}$ ∎

عبارت همان حالت گسسته لمی است که در درس برای بلوک‌های loner اثبات کردیم.

نتایج لم پنجم ، نشان می‌دهد که امنیت تحت growth و quality موردنظر برقرار است. اثبات آن مشابه اثبات انجام شده در درس با بلوک‌های loner است با این تفاوت که زمان گسسته شده است.

فاز سوم M(q, 0, 0)

مدل ما به شکل زیر است :

1. اولاً در هر round تعداد query ، q-bounded است.
2. مدل دارای تأخیر شبکه نیست.
3. در این حالت message loss داریم.

در این فاز می‌خواهیم تأثیر وجود Message Loss را در امنیت بررسی کنیم. این به این معنی است که اگر یک node به حالت sleep برود ، اگر در round های بعدی alert شود ، پیام‌های که در round های sleep به او رسیده است ، از دست می‌رود و ممکن است شبکه او قدیمی بماند. این علاوه بر اینکه ممکن است فایده یک honest را از بین ببرد ، ممکن است نود honest طبق تمایل Dishonest ها نیز رفتار کند و متناسب با نیاز آن‌ها هدایت شود.

برای اثبات این فاز ابتدا دو متغیر زیر را تعریف می‌کنیم :

C_i : مجموعه‌ی تمام longest chain های واقعی موجود در i امین round می‌باشد.

L_i^j : local chain مربوط به نود j ام در i امینround می‌باشد.

حال سه لم را مطرح واثبات می‌کنیم و در فرآیند اثبات از آن‌ها بهره می‌گیریم :

لم اول : در هر round به شماره i ، به صورت مانیگین $E[n_{alert}] = (1-s)(n-t)$ نود وجود دارد که L_i^j آن‌ها متعلق به C_i باشد.

اثبات : برای این منظور استقرا می‌زنیم. برای round اول همه دارای بلوک Genesis هستند. در فرض استقرا فرض می‌کنیم که حکم برای round شماره i برقرار باشد. حال باید ثابت کنیم که برای round شماره 1 + i نیز برقرار است. برای اثبات این حکم بسته به مقادیر مختلف مقادیر X_i , Z_i , Y_i حالت بندی می‌کنیم و برای هرکدام حکم را ثابت می‌کنیم. اثبات‌ها ساده‌اند و در مقاله ذکر شده است.

قبل از بیان لم دوم ، سه متغیر جدید تعریف می‌کنیم :

یک متغیر به نام n_{alert}^* تعریف می‌کنیم که منظور تعداد نودهایی است کهhonest هستند و هم در round جاری و هم round اخیر alert بوده‌اند. بنابراین مقدار امید ریاضی آن $(1-s)^2(n-t)$ است.

یک متغیر به نام X_i^* تعریف می‌کنیم : این متغیر هنگامی 1 است که یکی از نودهای honest که local chain آن متعلق به C_i باشد و همچنین PoW حل کرده باشد. و در غیر اینصورت نیز 0 است.

همچنین رو یک مجموعه زمانی S تعریف می‌کنیم : $X^*(s) = \sum_S X_i^*$

یک متغیر به نام Y_i^* تعریف می‌کنیم : این متغیر هنگامی 1 است که دقیقاً یکی از نودهای honest که local chain آن متعلق به C_i باشد و همچنین PoW حل کرده باشد. و در غیر اینصورت نیز 0 است.

همچنین رو یک مجموعه زمانی S تعریف می‌کنیم : $Y^*(s) = \sum_S Y_i^*$

لم دوم :

$$\frac{pqE[n_{alert}^*]}{1+pqE[n_{alert}^*]} \leq E[X_i^*] \leq pqE[n_{alert}^*].$$

اثبات این لم دقیقاً مشابه لم منتاظر با آن در مدل اول است. با این تفاوت که استدلال‌ها برای n_{alert}^{*} به جای n_{alert} بیان می‌شود.

لم سوم :

$$E[Y_i^*] = E[pqn_{alert,i}^{*}(1-p)^{q(n_{alert,i}^{*}-1)}] \geq E[X_i^*](1-E[X_i^*])$$

اثبات این لم دقیقاً مشابه لم منتاظر با آن در مدل اول است. با این تفاوت که استدلال‌ها برای n_{alert}^{*} به جای n_{alert} و X_i^* به جای X_i بیان می‌شود.

در این جا شرطی که باید برقرار باشد تا امنیت برقرار بماند ، به شرح زیر است. این شرط نشان می‌دهد که همچنان می‌توان اکثریت موقتی Dishonest ها را تحمل کرد :

$$t + (1-s)E[n_{alert}] < c\ (1-\ \delta\ )E(n_{alert}{}^{*}) \quad ; \quad \delta \geq 2E(X_i^*) + 3\epsilon$$

همان‌طور که مشاهده می‌شود ، δ باید مقدار متفاوتی نسبت به دو حالت قبل داشته باشد. حال term های موجود در آن را آنالیز می‌کنیم :

در Term اول X_i^* به جای X_i استفاده شده است. زیرا کسانی که شبکه اصلی Honest را پیش می‌برند ، کسانی هستند که شبکه آن‌ها جدید و update است.

عبارت دوم نیز بین حالت اول و دوم است. یعنی ضریب ϵ از 2 و 4 به 3 تغییر کرده است. علت این موضوع این است که با اینکه تأخیر در شبکه نداریم ، ولی message loss موجب می‌شود تا نوسان کمتری را بتوانیم حول میانگین تحمل کنیم.

اما مهم‌تر از همه عبارتی است که در سمت چپ تساوی اضافه شده است که آن $(1-s)E[n_{alert}]$ است. این به این دلیل است که نودهای honest و alert ای که در round قبل sleep بوده‌اند ، رفتار آن‌ها همچون Adversary است و متخاصمان می‌توانند توان پردازشی آن‌ها را در جهت منافع خود به کار بگیرند.

با به کار بردن رابطه درجه نامساوی زیر بدست می‌آید :

$$s \leq \frac{2c(1-\delta) - \sqrt{1 + 4(1 + c(1-\delta)\frac{t}{n-t})}}{2(1 + c(1-\delta))}$$

حال 3 لم را اثبات می‌کنیم که دراثبات به کار خواهد آمد.

Lemma 9. *Suppose that at round r, the chains in C_i have size l. Then by round $s \geq r$, an expected number of $E[n_{alert}] = (1-s)(n-t)$ parties will have adapted a chain of length at least $l + \sum_{i=r}^{s-1} X_i^*$.*

برای اثبات این لم از نتیجه لم قبل استفاده می‌کنیم. در آن لم نشان داده شده بود که در هر round حداقل $E[n_{alert}]$ از اعضا ، دارای شبکه به روز update شده هستند. بنابراین برای محاسبه خروجی موردنظر ، کافی است تعداد round هایی را بشماریم که یکی از این longest chain ها بع اندازه یک واحد extend می‌شوند.

Lemma 10. *The probability that the honest parties j with $L_i^j \notin C_i$ can create a new chain $C' \in C_r$ for some round $r \geq i$, before any chain from C_i gets extended is denoted by ϕ. It holds that:*

$$\phi \leq \frac{s}{1-s}$$

برای اثبات حالت worst case را در نظر می‌گیریم که تمام نودهای alert که در round اخیر sleep بوده‌اند ، همه دارای یک longest chain یکسان هستند. و سپس فرض می‌کنیم که تنها یک بلوک از longest chain در آن round کوتاهتر است. کران بالای احتمال موردنظر را محاسبه می‌کنیم برای این منظور محاسبه می‌کنیم چقدر احتمال دارد که این نودها دو بلوک mine کنند قبل از آنکه نودهای n_{alert}^* یک بلوک mine کنند. برای منظور یک سیگما می‌نویسیم و کران احتمال بالارا محاسبه می‌کنیم. فرآیند محاسبه ساده و در مقاله ذکر شده است.

Lemma 11. *Suppose the k^{th} block B of a chain C was computed at round i, where $Y_i^* = 1$. Then with probability at least $1-\phi$, the k^{th} block in a chain C' will be B or requires at least one adversarial block to replace B.*

طبق نتیجه لم قبل و با توجه به تعریف φ ، این لم فوراً نتیجه می‌شود.

حال 5 لم موردنیاز نهایی را مطرح می‌کنیم :

4 لم اول دقیقاً مشابه مدل اول است با این تفاوت که به جای X_i با X_i^* تعویض می‌شوند.

$$a)\ (1-\epsilon)E[X_i^*]|S| < X^*(S)$$
$$b)\ (1-\epsilon)E[X_i^*](1-E[X_i^*])|S| < Y^*(S)$$
$$c)\ Z(S) < (1+\epsilon)\frac{t}{E[n_{alert}^*]}\frac{E[X_i^*]}{1-E[X_i^*]}|S| < (1+\epsilon)\Big(c(1-\delta)-\frac{s}{1-s}\Big)\frac{E[X_i^*]}{1-E[X_i^*]}|S|$$
$$d)\ For\ \sigma^* = (1-\epsilon)(1-E[X_i^*]):$$

$$Z(S) < \Big(1+\frac{\delta}{\sigma^*}\Big)\frac{t}{E[n_{alert}^*]}X^*(S) \le c\Big(1-\frac{\delta^2}{2\sigma^*}\Big)X^*(S)$$

لم 5 ام به صورت زیر می‌باشد :

$$Z(S) < Y^*(S)(1-\epsilon)(1-\phi)$$

تفاوت نتیجه حاصل با مدل اول در 3 عبارت است :

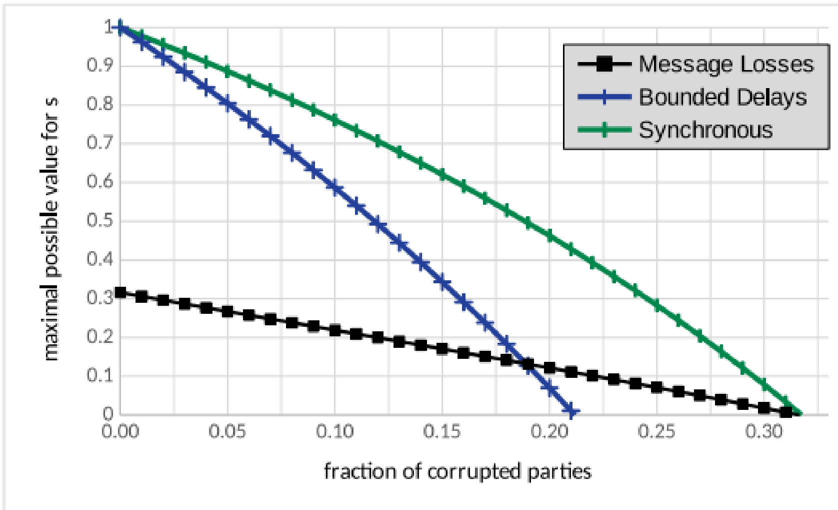
اول به جای $Y(S)$ عبارت $Y^*(S)$ ظاهر شده است. زیرا نودهایی که شبکه honest ها را پیش می‌برند و نتیجه مثبت دارند ، مربوط به n_{alert}^* است و بلوک‌های این گروه بلوک های مفید و پیش‌برنده هستند که این بلوک‌ها در Y_i^* محاسبه می‌شوند.

دوم عبارت $(1-\epsilon)$ در کران راست ظاهر شده است که علت آن تبدیل شدن 2ϵ به 3ϵ در کران δ است.

سوم اینکه عبارت $(1-\varphi)$ در کران راست ظاهر شده است که علت آن این است که در φ درصد مواقع ، بلوک‌های نودهای alert که در round قبل sleep بوده‌اند در شبکه به نفع متخاصمان خرابکاری می‌کند.

نتیجه گیری نهایی :

در 3 مدل ذکر شده در بالا ، نمودار حداکثر s قابل تحمل بر اساس درصد متخاصمان رسم شده است :



تحلیل :

همان‌طور که مشاهده می‌شود در شرایط با درصد کم متخاصمان ، حساسیت حالت دارای message-loss نسبت به دو مدل دیگر بیشتر است ولی در حالتی که درصد متخاصمان بالاست ، افت نمودار در دو مدل اول شدیدتر است و حساسیت به s بالاتر می‌رود.

در کل این نمودار نشان می‌دهد که ترکیب‌های مختلف توان پردازشی و توان حمله DDos توسط متخاصمان ، چه تأثیری بر روی Security شبکه دارد. و طبق این نمودار متخاصم می‌تواند resources خود را به بهینه‌ترین شکل ممکن بین توان پردازشی در شبکه و ddos زدن تقسیم کند تا بالای نمودارهای فوق قرار بگیرد و Security شبکه را از بین ببرد.