دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۱، جلد ۲۰، شماره ۲ (۲۲–۳۲)
https://monadi.isc.org.ir

ارائه‌شده در نوزدهمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه گیلان، ۹ و ۱۰ شهریور ۱۴۰۱

روش اجماع با امنیت بالا و بار مخابراتی کم★

علیرضا هدیه‌لو[۱،*]، جواد مهاجری[۲] و محمدرضا عارف[۱]

[۱] دانشکده‌ی مهندسی برق، دانشگاه صنعتی شریف، تهران، ایران
[۲] پژوهشکده‌ی الکترونیک، دانشگاه صنعتی شریف، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
اجماع
شبکه‌ی همگام
گره بیزانسی
امضای تجمیعی

dor: 20.1001.1.24763047.1401.11.2.3.2

نوع مقاله: پژوهشی

چکیده

برای ایجاد هماهنگی بین گره‌های شبکه و افزایش مقاومت سامانه‌های توزیع شده در برابر خطا از پروتکل‌های اجماع استفاده می‌شود. در این مقاله، یک روش اجماع همگام جدید معرفی می‌شود. روش پیشنهادی، نسخه‌ای بهبود یافته از پروتکل Abraham است که در آن از امضای تجمیعی برای کاهش بار مخابراتی استفاده کرده‌ایم. در پروتکل Abraham بار مخابراتی و بار محاسباتی پروتکل، به ترتیب از مرتبه‌ی $\mathcal{O}(n^3 s_s)$ و $\mathcal{O}(n^3)$ است در حالی که در پروتکل پیشنهادی بار مخابراتی و بار محاسباتی به ترتیب از مرتبه‌ی $\mathcal{O}(m_a n^2 \lceil \log k_t \rceil + m_a n s_s)$ و $\mathcal{O}(m_a n^2)$ است، که در آن n تعداد گره‌های شبکه، s_s اندازه‌ی یک امضای دیجیتال، k_t حداکثر تعداد تکرارهای پروتکل و m_a یک پارامتر امنیتی است که می‌تواند بسیار کوچکتر از n باشد. بنابراین در شبکه‌هایی با تعداد گره زیاد، کاهش بار مخابراتی و بار محاسباتی محسوس خواهد بود. همچنین در این پروتکل برای مقاومت در برابر f گره‌ی بیزانسی به حداقل $n = 2f + 1$ گره نیاز داریم و با احتمال حداقل $1 - \frac{1}{2^{m_a}}$ روند اجماع به درستی انجام می‌شود.

© ۱۴۰۱ انجمن رمز ایران

۱ مقدمه

در سامانه‌های توزیع شده، برای هماهنگی بین گره‌های شبکه و مقاومت سامانه در برابر خطا از اجماع استفاده می‌شود. از کاربردهای اجماع می‌توان به زنجیره‌ی قالب‌ها [۶–۸] و انتخاب رهبر [۹] اشاره کرد. در پروتکل‌های Random Beacon Protocol [۴، ۵]، نیز در برخی مراحل، از اجماع استفاده می‌شود. در زنجیره‌ی قالب‌ها، برای جلوگیری از حمله‌ی دوبار خرج کردن، باید گره‌های شبکه، بعد از تشکیل هر قالب، به اجماع برسند. در زنجیره‌ی قالب‌هایی که مبتنی بر اثبات سهم هستند، برای اجماع، بیشتر از روش‌های عیب‌تاب در برابر عوامل بیزانسی مانند PBFT [۱] [1] استفاده می‌شود.

مسئله‌ی رسیدن به اجماع به معنی توافق روی یک مقدار، در یک شبکه‌ی مخابراتی شامل n گره است. در میان این n گره، f گره بیزانسی هستند و می‌توانند رفتار دلخواه خود را داشته باشند. اما بقیه‌ی گره‌ها که گره‌های صادق هستند از یک پروتکل مشخص پیروی می‌کنند. پس از انجام پروتکل اجماع، همه‌ی گره‌های صادق پس از مدت محدودی باید روی یک مقدار یکسان به توافق برسند. روش‌های اجماع به دو دسته‌ی روش‌های همگام و جزئاً همگام تقسیم می‌شوند. در روش‌های اجماع همگام، پیام‌های گره‌ها در بازه‌ی زمانی مشخص به مقصد می‌رسند و مهاجم فقط در این بازه‌ی زمانی می‌تواند زمان رسیدن پیام‌ها را جابجا کند. اما در روش‌های جزئاً همگام، این بازه‌ی زمانی مشخص نیست

★از کمیته علمی نوزدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول
آدرس‌های رایانامه: alireza.hediehloo@ee.sharif.edu (علیرضا هدیه‌لو)، mohajer@sharif.ir (جواد مهاجری)، aref@sharif.edu (محمدرضا عارف)
© ۱۴۰۱ تمامی حقوق متعلق به انجمن رمز ایران است.

[1] Practical Byzantine Fault Tolerance

[۱۳]. یکی از روش‌های مطرح اجماع که در کارهای گذشته ارائه شده، PBFT است [۱]. در این روش، شبکه جزئاً همگام در نظر گرفته می‌شود و برای مقاومت در برابر f گره‌ی بیزانسی حداقل به $n = ۳f + ۱$ گره نیاز است. بار مخابراتی این روش از $\mathcal{O}(n^۲ s_s)$ است که s_s اندازه‌ی یک امضای دیجیتال است. در نتیجه با افزایش n کارایی روش کاهش می‌یابد. در [۷]، یک روش مبتنی بر PBFT معرفی شده است که با استفاده از امضای تجمیعی CoSi [۱۰]، بار مخابراتی به $\mathcal{O}(n^۲ + ns_s)$ کاهش یافته است، اما زمان اجرای پروتکل به $\mathcal{O}(\log n)$ افزایش پیدا کرده است. در [۱۱]، روش FBFT [1] معرفی شده که با تغییراتی در PBFT، با استفاده از امضای تجمیعی، بار مخابراتی به $\mathcal{O}(n^۲ + ns_s)$ کاهش یافته است و زمان اجرای پروتکل هم از $\mathcal{O}(۱)$ است. در پروتکل Abraham [۲]، یک طرح بر مبنای شبکه‌ی همگام ارائه شده و تا حداکثر ۴۹ درصد گره‌ها می‌توانند بیزانسی باشند. در این طرح، بار مخابراتی از $\mathcal{O}(n^۳ s_s)$ است.

در این مقاله، ما یک پروتکل اجماع که یک نسخه‌ی بهبود یافته از پروتکل Abraham است، ارائه می‌کنیم. پروتکل پیشنهادی، نیز مانند طرح Abraham تا ۴۹ درصد در برابر عوامل مخرب مقاومت دارد. همچنین بار مخابراتی آن از $\mathcal{O}(m_a n^۲ \lceil \log k_t \rceil + m_a n s_s)$ است که k_t حداکثر تعداد تکرارهای پروتکل است و m_a یک پارامتر امنیتی مستقل از n است. بار محاسباتی پروتکل نیز از $\mathcal{O}(n^۳)$ است. در این پروتکل نیز مانند طرح پایه، شبکه همگام در نظر گرفته می‌شود. در این پروتکل از دو راهکار برای کاهش بار مخابراتی استفاده شده است: ۱) استفاده از امضای تجمیعی ۲) ارسال و دریافت پیام‌ها برای دسته‌ی کوچکی از گره‌ها، به جای ارسال و دریافت پیام‌ها برای تمام گره‌ها.

پروتکل پیشنهادی همانند پروتکل Abraham تا ۴۹ درصد در برابر عوامل بیزانسی مقاوم است. در حالی که روش‌های مبتنی بر PBFT، از مقاومت ۳۳ درصدی در برابر این عوامل برخوردارند. در عوض، در پروتکل پیشنهادی و پروتکل پایه، از فرض همگام بودن شبکه استفاده شده است. همچنین پروتکل پیشنهادی، در مقایسه با پروتکل Abraham، بار مخابراتی کمتری دارد.

در ادامه، در بخش ۲، امضای تجمیعی معرفی می‌شود. از آن‌جا که پروتکل ارائه شده، یک نسخه‌ی بهبود یافته از پروتکل Abraham [۲] است، در بخش ۳ به معرفی پروتکل Abraham می‌پردازیم. در بخش ۴، پروتکل بهبودیافته پیشنهادی معرفی می‌شود. در بخش ۵ به ارزیابی امنیت، بار مخابراتی و بار محاسباتی طرح پیشنهادی اختصاص می‌پردازیم. بخش ۶ به جمع‌بندی و نتیجه‌گیری مقاله اختصاص یافته است.

۲ امضای تجمیعی

در این بخش، طرح امضای تجمیعی مقاله‌ی [۳]، معرفی می‌شود. طرح امضای تجمیعی، یک امضای دیجیتال است که برای وارسی دارای قابلیت تجمیع است. فرض کنید n امضا روی n پیام مجزا از n گره داده شده است. در این امضا این امکان فراهم شده است که تمام امضاهای دیجیتال را در یک امضا خلاصه کرد. این امضا (و n پیام اصلی) می‌تواند شخص وارسی‌کننده را در مورد صحت کلیه امضاها متقاعد سازد.

۱.۲ طرح امضای تجمیعی

طرح Boneh [۳] بر مبنای استفاده از نگاشت دو خطی است. اگر $G_۱$ و $G_۲$ دو گروه ضربی از مرتبه‌ی یک عدد اول بزرگ و $g_۱$ و $g_۲$ به ترتیب مولدهای این دو گروه باشند. $e: G_۱ \times G_۲ \to G_T$ یک نگاشت دوخطی باشد و با داشتن $h \in G_۱$ و $g_۲, g_۲^a \in G_۲$ محاسبه‌ی h^a به لحاظ محاسباتی کار سختی باشد. همچنین اگر $H: \{۰,۱\}^* \to G_۱$ یک تابع چکیده‌ساز باشد. مراحل امضای تجمیعی Boneh، به صورت زیر است:

۱) **تولید کلید:** هر کدام از گره‌ها مانند گره‌ی iام، عدد تصادفی $a_i \in \mathbb{Z}_p$ را به عنوان کلید خصوصی و $A_i = g_۲^{a_i}$ را به عنوان کلید عمومی در انتخاب می‌کند.

۲) **تولید امضا:** امضای هر گره مانند گره‌ی iام روی پیام M_i به صورت $\sigma_i = (H(M_i))^{a_i}$ محاسبه می‌شود.

۳) **وارسی امضا:** اگر پیام M_i، امضای σ_i و کلید عمومی شخص امضاکننده A_i داده شده باشد. در صورت صحت امضا، تساوی زیر برقرار خواهد بود:

$$e(\sigma_i, g_۲) = e(H(M_i), A_i)$$

۴) **تجمیع:** در این الگوریتم امضاهای $\{\sigma_i\}_{i=۱}^n$ دریافت می‌شود که این امضاها توسط گره‌هایی با کلید عمومی $\{A_i\}_{i=۱}^n$ بر روی پیام‌های $\{M_i\}_{i=۱}^n$ تولید شده اند و امضای تجمیعی به شکل زیر تولید می‌شود:

$$\sigma = \prod_{i=۱}^{n} \sigma_i$$

۵) **وارسی امضای تجمیعی:** در این الگوریتم پیام‌های $\{M_i\}_{i=۱}^n$ و امضای تجمیعی σ دریافت می‌شود. در صورت صحت کلیه امضاها، تساوی زیر برقرار خواهد شد:

$$e(\sigma, g_۲) = \prod_{i=۱}^{n} e(H(M_i), A_i)$$

در این طرح ، برای جلوگیری از حمله‌ی «کلید مرتبط» [۱۲]، همراه پیام M_i، کلید عمومی نیز امضا می‌شود ($\sigma_i = (H(M_i \| A_i))^{a_i}$).

۲.۲ امضای تجمیعی روی تعداد محدودی پیام

فرض کنید در یک سامانه‌ی توزیع شده شامل n گره، برخی از گره‌ها، یکی از پیام‌های مجموعه‌ی $\{M_۱, M_۲, \ldots, M_{k_t}\}$ را امضا می‌کنند و آن را به گره رهبر ارسال می‌کنند تا او امضاها را تجمیع و نتیجه را به تمام گره‌ها ارسال کند. برای وارسی امضای تجمیع شده، باید مشخص شود

[1] Fast Byzantine Fault Tolerance

کدام گره‌ها، امضا ارسال کرده‌اند و هر کدام در صورت امضا، چه پیامی را امضا کرده‌اند. بنابراین $k_t + ۱$ حالت برای هر گره وجود دارد که آن را با $\lceil \log_۲ (k_t + ۱) \rceil$ بیت می‌توان نشان داد. در نتیجه گره تجمیع کننده، باید همراه امضای تجمیعی، یک رشته بیت به طول $n\lceil \log_۲ (k_t + ۱) \rceil$ را ارسال کند تا وضعیت تمام گره‌ها را مشخص کند.

۳ پروتکل Abraham

از آن‌جا که پروتکل پیشنهادی، یک نسخه‌ی بهبود یافته‌ی پروتکل Abraham [۲] است، در این بخش به معرفی این پروتکل می‌پردازیم.

فرض کنید که $n = ۲f + ۱$ گره داریم و حداکثر f گره تحت کنترل مهاجم است. هدف رسیدن به اجماع بین گره‌ها روی یک مقدار است. تمام پیام‌های ارسالی امضا می‌شوند و گره‌ها، کلیدهای عمومی یکدیگر را دارند. پیام‌ها در شبکه‌ی همگام ارسال می‌شود. گره‌های بیزانسی با یکدیگر متحد هستند و کلید خصوصی یکدیگر را دارند. پروتکل شامل تعدادی تکرار است و در هر تکرار، یکی از گره‌ها به عنوان رهبر انتخاب شده است. انتخاب رهبر توسط یک پروتکل سطح بالاتر، انجام می‌شود. مثلاً این کار می‌تواند با تولید اعداد تصادفی به صورت نامتمرکز انجام شود [۴، ۵]. گره‌ی رهبر در تکرار kام را با L_k نشان می‌دهیم. فرض کنید $\mathrm{sign}_i(x)$ نشان‌دهنده‌ی امضای پیام x توسط گره‌ی iام است. «تصمیم گرفتن روی مقدار v» برای یک گره، به معنای نهایی شدن و پذیرفتن مقدار v برای آن گره است.

۱.۳ جزئیات پروتکل

پروتکل Abraham، یک پروتکل با تکرار است که هر تکرار آن، ۴ دور دارد. دورها به ترتیب «وضعیت»، «پیشنهاد»، «تعهد» و «اعلام» نام دارند. در دور «وضعیت»، گره‌ها وضعیت فعلی سامانه را به گره رهبر ارسال می‌کنند. گره رهبر در دور «پیشنهاد» با توجه به وضعیت سامانه باید یک مقدار v را انتخاب و به تمام گره‌ها ارسال کند. در دور «تعهد»، گره‌ها مقدار دریافت‌شده از رهبر را بازارسال می‌کنند. در دور «اعلام»، هر گره، در صورتی که فقط یک مقدار پیشنهادی از سوی گره رهبر مشاهده کند و حداقل $f + ۱$ پیام «تعهد» دریافت کرده باشد، روی مقدار v تصمیم می‌گیرد و یک پیام «اعلام» به سایر گره‌ها ارسال می‌کند.

فرض کنید شماره‌ی تکرار فعلی را با k و شماره‌ی دور فعلی را با r نشان دهیم که یکی از اعداد ۰، ۱، ۲ و ۳ است. هر گره مثل گره‌ی iام، سه متغیر v_i، k_i و C_i دارد. v_i را «مقدار پذیرفته شده» می‌نامیم. k_i شماره‌ی تکراری است که مقدار v_i در آن پذیرفته شده است. C_i یک گواهی برای مقدار v_i است و در زمان پذیرش v_i، گواهی C_i مشخص می‌شود. در شروع پروتکل مقداردهی اولیه به صورت زیر انجام می‌شود:

$$v_i = \perp, \quad k_i = ۰, \quad C_i = \perp$$

ابتدا وضعیت پروتکل را برای حالتی که هنوز هیچ گره‌ای تصمیم نگرفته است توضیح خواهیم داد.

دور ۰ (وضعیت): گره‌ها پیام زیر را به رهبر (L_k) ارسال می‌کنند.

$$\left\{ \begin{array}{c} k, r, v_i, k_i, \mathrm{sign}_i(k, r, v_i, k_i), C_i, \\ \mathrm{sign}_i(k, r, v_i, k_i, \mathrm{sign}_i(k, r, v_i, k_i), C_i) \end{array} \right\}$$

دور ۱ (پیشنهاد): رهبر پیام‌های دریافتی را وارسی می‌کند. s_i را بخشی از پیام دریافتی از سوی گره‌ی iام به صورت زیر تعریف می‌کنیم:

$$s_i = \{k, r = ۰, v_i, k_i, \mathrm{sign}_i(k, r = ۰, v_i, k_i)\}$$

فرض کنید P شامل $f + ۱$ تا از s_iها باشد:

$$P = \{s_۱, s_۲, \ldots, s_(f + ۱)\}$$

گره رهبر باید یک «مقدار امن با توجه به P» را پیشنهاد کند که آن را v می‌نامیم. به این معنی که او در بین s_iها، گزینه‌ای که بیشترین k_i را دارد انتخاب می‌کند (در صورت وجود چندین گزینه، یکی را انتخاب می‌کند). فرض کنید $s_{\max}$ انتخاب شده باشد. اگر مقدار $k_{\max}$ برابر صفر باشد، گره رهبر می‌تواند هر مقداری را به عنوان v انتخاب کند و C را برابر با $C = \perp$ قرار دهد. در غیر این صورت باید v را برابر $v_{\max}$ قرار داده و $C = C_{\max}$ باشد. در نهایت پیام زیر به تمام گره‌ها ارسال می‌شود:

$$\left\{ \begin{array}{c} k, r, v, \mathrm{sign}_(L_k)(k, r, v), P, C \\ \mathrm{sign}_{L_k}(k, r, v, \mathrm{sign}_{L_k}(k, r, v), P, C) \end{array} \right\}$$

دور ۲ (تعهد): در این مرحله گره‌ها، پیام گره رهبر را وارسی می‌کنند، در صورت معتبر بودن پیام، پیام «تعهد» را به تمام گره‌ها ارسال کنند که به صورت زیر است:

$$\left\{ \begin{array}{c} k, r = ۱, v, \mathrm{sign}_{L_k}(k, r = ۱, v) \\ k, r = ۲, v, \mathrm{sign}_i(k, r = ۲, v) \end{array} \right\}$$

دور ۳ (اعلام): هر یک از گره‌ها پیام‌های «تعهد» دریافت شده را وارسی کرده و درصورتی که مقدار v در همه‌ی این پیام‌ها، یکسان باشد و حداقل $f + ۱$ پیام «تعهد» دریافت شده باشد، روی مقدار v تصمیم گرفته و پیام «اعلام» به تمام گره‌های دیگر ارسال می‌کند. این پیام به صورت زیر است:

$$\left\{ \begin{array}{c} k, r, v, \mathrm{sign}_i(k, r, v), C \\ \mathrm{sign}_i(k, r, v, \mathrm{sign}(k, r, v), C) \end{array} \right\}$$

در این پیام C برابر حداقل $f + ۱$ پیام «تعهد» است که از سوی گره‌ها ارسال شده است. در پایان این دور، گره‌ها پس از وارسی پیام‌ها، مقادیر v_i، k_i و C_i خود را مطابق یکی از پیام‌های «اعلام» دریافت شده، قرار می‌دهند.

وضعیت پروتکل پس از پیام‌های «اعلام»: پس از آن که برخی از گره‌ها در دور «اعلام»، بر روی مقداری تصمیم گرفتند، در تکرارهای بعدی

در پروتکل شرکت نمی‌کنند. بنابراین باید پیام «اعلام» آن‌ها در مراحل پروتکل در نظر گرفته شود.

در دور وضعیت: گره رهبر، پیام «اعلام» از سوی گره‌ی iام را مانند یک پیام «وضعیت» در نظر می‌گیرد. حتی در اثبات P و گواهی C نیز می‌تواند از آن استفاده کند.

در دور پیشنهاد: اگر L_k رهبر تکرار فعلی باشد و در تکرارهای قبلی پیام «اعلام» را ارسال کرده باشد، بقیه‌ی گره‌ها، پیام «اعلام» او را مانند پیام «پیشنهاد» در نظر می‌گیرند. در صورت برابری مقدار گره رهبر با مقدار v_i، گره‌ی iام پیام «تعهد» را ارسال می‌کند.

در دور تعهد: فرض کنید گره‌ی iام در یکی از تکرارهای قبلی پیام «اعلام» را ارسال کرده باشد. در صورتی که مقدار v در پیام «اعلام» برابر مقدار پیشنهادی رهبر باشد، بقیه‌ی گره‌ها آن پیام را مانند یک پیام «تعهد» در نظر می‌گیرند. همچنین پیام «اعلام» می‌تواند برای ارائه‌ی گواهی C نیز استفاده شود.

اگر در یک تکرار، گره رهبر از میان گره‌های صادق انتخاب شود، در همان تکرار، تمام گره‌های صادق به اجماع می‌رسند. چون گره رهبر در تکرارهای مختلف به صورت تصادفی انتخاب می‌شود، با احتمال بالا، گره رهبر در یکی از تکرارها گره‌ای صادق خواهد بود. اگر حداکثر ۵۰ درصد گره‌ها، مهاجم باشند، پروتکل اجماع به صورت میانگین ۲ تکرار (۸ دور) طول می‌کشد.

۴ پروتکل پیشنهادی

در این بخش جزئیات پروتکل پیشنهادی معرفی می‌شود. در این پروتکل بار مخابراتی از $\mathcal{O}(m_a n^2 \lceil \log k_t \rceil + m_a n s_s)$ است که در آن m_a یک پارامتر امنیتی و مستقل از n است و می‌تواند بسیار کوچکتر از n باشد و s_s اندازه‌ی امضای دیجیتال است.

فرضیات پروتکل، همانند پروتکل Abraham است. پروتکل دارای تعدادی تکرار است که شماره‌ی آن را با «k» نشان می‌دهیم. همچنین هر تکرار دارای ۶ دور هست و دور را با «r» نشان می‌دهیم. هر گره، دارای یک شماره‌ی یکتا «uid» است که برای تمام گره‌ها شناخته شده است. «uid» عددی بین ۰ تا $n-1$ است. در تمام پیام‌های امضا شده در پروتکل، uid، k و r حضور دارند. همچنین نماد $S_i(x)$ به صورت زیر تعریف می‌شود:

$$S_i(x) = \mathrm{sign}_i(uid, k, r, x)$$

m_a گره به عنوان گره‌های مجموعه‌ی «تجمیع»، به صورت تصادفی انتخاب می‌شوند. انتخاب این گره‌ها می‌تواند با استفاده Random Beacon Protocol، انجام شود [۴، ۵]. امنیت پروتکل بر این پایه است که با احتمال بسیار بالا، حداقل یکی از گره‌های مجموعه‌ی «تجمیع»، صادق است. در واقع اگر این گره‌ها به صورت تصادفی انتخاب شوند و مهاجم حداکثر ۵۰ درصد گره‌ها را کنترل کند، احتمال آن که تمام m_a گره، از گره‌های مهاجم باشند، $\frac{1}{2^{m_a}}$ است. با انتخاب معقول m_a می‌توان این احتمال را کم کرد.

هر تکرار شامل ۶ دور است در دور «وضعیت» گره‌ها وضعیت فعلی سامانه را به رهبر ارسال می‌کنند. در دور «پیشنهاد»، گره رهبر یک مقدار امن با توجه به وضعیت سامانه انتخاب می‌کند و به تمام گره‌ها ارسال می‌کند. در دور «تعهد» گره‌ها مقدار پیشنهادی گره رهبر را به گره‌های مجموعه‌ی «تجمیع» بازارسال می‌کنند. در دور «تجمیع اولیه» گره‌های مجموعه‌ی «تجمیع»، در صورتی که تمام پیام‌های دریافتی از گره رهبر شامل مقدار یکسانی باشند ، یک امضای تجمیعی روی مقدار ارسال شده‌ی رهبر، ارائه و به تمام گره‌ها ارسال می‌کنند. اما در صورتی که گره رهبر مقادیر متفاوتی ارسال کرده باشد، دو مقدار از آن‌ها را انتخاب و آن را به همراه امضای گره رهبر به تمام گره‌ها ارسال می‌کنند. در دور «اعلام»، گره‌ها، در صورتی که رهبر در تمام پیام‌ها ، فقط یک مقدار را ارسال کرده باشد و حداقل به اندازه‌ی $f+1$ گره، پیام «تعهد» را ارسال کرده باشند، آن‌گاه روی مقدار گره رهبر تصمیم گرفته می‌شود و یک پیام «اعلام» به گره‌های مجموعه‌ی «تجمیع» ارسال می‌شود. در دور «تجمیع ثانویه» گره‌های مجموعه‌ی «تجمیع»، پیام‌های دریافتی را وارسی کرده و در صورتی که تمامی مقادیر یکسان باشند، یک امضای تجمیعی تولید و ارسال می‌کنند. اما در صورتی که پیام‌های «اعلام» با مقادیر مختلف دریافت کرده باشند، دو پیام دریافتی که دارای مقادیر متفاوت هستند را به تمام گره‌ها ارسال می‌کنند.

۱.۴ جزئیات پروتکل

هر گره مانند گره‌ی iام متغیرهای $(v_i, k_i, C_i, \{N_{i,j}\}_{j=1}^{m_a})$ را در اختیار دارد. در ابتدای کار مقدار v_i برابر با $v_i = \perp$ است. ولی پس از دریافت پیام‌های «تجمیع ثانویه»، این مقدار تغییر می‌کند. k_i شماره‌ی تکراری است که گره‌ی iام، مقدار v_i را تغییر داده است. در ابتدای پروتکل $k_i = 0$ است. C_i نشان دهنده گواهی مربوط به مقدار v_i است که در تکرار k_i به دست آمده است. در شروع پروتکل $C_i = \perp$ است. $N_{i,j}$ فهرستی از پیام‌های «تجمیع ثانویه» است که از گره‌ی jام از مجموعه‌ی «تجمیع»، دریافت شده است. در ابتدای کار $N_{i,j}$ تهی است. ساختار کلی پیام‌ها به صورت زیر است:

$$(uid, k, r, m, S_{uid}(m))$$

که هر پیام ارسالی، باید از ساختار فوق پیروی کند. m برحسب این که در چه دوری پیام ارسال شود، تعیین می‌شود. در حقیقت m می‌تواند شامل چندین متغیر باشد و بسته به این که در چه دوری هستیم، ساختاری متفاوت داشته باشد. در ادامه ساختار m توضیح داده می‌شود. در این مقاله از علامت "." برای نشان دادن جزئی از یک ساختار استفاده می‌شود. به عنوان مثال فرض کنید s_t یک پیام با ساختار مذکور، باشد، در این صورت $s_t \cdot uid$ نشان دهنده‌ی uid موجود در پیام s_t است. در ابتدا پروتکل را برای حالتی که هنوز هیچ گره‌ای به تصمیم نرسیده است، توضیح می‌دهیم.

دور ۰ (وضعیت): ساختار m در این دور به صورت زیر است:

$$m = (v_i, k_i, sigIt, sigVal, C_i)$$

که در آن

$$sigIt = S_i(k_i), \quad sigVal = S_i(k_i, v_i)$$

در این دور، گره‌ها این پیام را به گره رهبر L_k، ارسال می‌کنند. با این کار گره رهبر از وضعیت فعلی سامانه آگاه می‌شود.

دور ۱ (پیشنهاد): در این دور گره رهبر باید مقدار v را انتخاب و به همه‌ی گره‌ها ارسال کند. گره رهبر، $f+1$ پیام از پیام‌های دریافتی را انتخاب می‌کند. اگر این پیام‌ها به صورت $\{s_1, \ldots, s_{(f+1)}\}$ باشند و اگر $\max\left(\{s_t \cdot m \cdot k_i\}_{t=1}^{f+1}\right)$ برابر صفر باشد، رهبر هر مقداری را می‌تواند پیشنهاد داده و C را برابر $\perp$ قرار دهد؛ در غیر این صورت فرض کنید:

$$t_{\max} = \text{argmax}_{1 \le t \le f+1} s_t \cdot m \cdot k_i$$

در این صورت گره رهبر باید $s_{t_{\max}} \cdot m \cdot v_i$ را به عنوان مقدار پیشنهادی و $s_{t_{\max}} \cdot m \cdot C_i$ را به عنوان گواهی C در نظر گیرد. در این حالت، گره رهبر باید ثابت کند مقدار پیشنهادی او از روی پیام‌های دریافتی‌اش به درستی انتخاب شده است. برای این کار sp و P را ارسال می‌کند. sp بر اساس پیام $s_{t_{\max}}$ مقداردهی می‌شود و ساختار آن به صورت زیر است:

$$sp = (uid_{\max}, k_{\max}, r_{\max}, S_{uid_{\max}}(k_{\max}, v_{\max}))$$

که در آن:

$$uid_{\max} = s_{t_{\max}} \cdot uid, \; r_{\max} = 0, \; k_{\max} = s_{t_{\max}} \cdot m \cdot k_i$$

P نشان می‌دهد که مقدار $sp \cdot k_{\max}$، برابر بزرگترین عدد مجموعه‌ی $\{s_t \cdot m \cdot k_i\}_{t=1}^{f+1}$ است. یک راه ساده برای چنین اثباتی، ارسال تمام مجموعه‌ی $\{s_t\}_{t=1}^{f+1}$ است. اما این کار بار مخابراتی زیادی دارد. به جای آن، از فشرده‌سازی و امضای تجمیعی استفاده می‌کنیم. ساختار P به صورت زیر است:

$$P = (iterations, iterationIdx, roundIdx, aggSig)$$

$iterations$ یک مجموعه‌ی بدون تکرار از اعداد $\{s_t \cdot m \cdot k_i\}_{t=1}^{f+1}$ است و $iterationIdx$ یک آرایه به طول n است. اگر گره‌ی iام در بین $f+1$ پیام انتخاب شده از سوی گره رهبر نباشد، مؤلفه‌ی iام $iterationIdx$ برابر صفر است. در غیر این صورت مؤلفه‌ی iام نشان می‌دهد در پیام «وضعیت» ، مقدار k_i برابر چندمین عضو از مجموعه‌ی $iterations$ است. در صورتی که هیچ گره‌ای تا این زمان به تصمیم نرسیده باشد، همه‌ی مؤلفه‌های $iterationIdx$ برابر صفر یا یک هستند. $roundIdx$ نیز یک رشته بیت به طول n است. همه‌ی مؤلفه‌های این آرایه نیز در صورتی که هیچ گره‌ای تا به حال تصمیم نگرفته باشند برابر صفر است. $aggSig$ یک امضای تجمیعی از $\{s_t \cdot m \cdot sigIt\}_{t=1}^{f+1}$ است.

ساختار m در دور وضعیت به صورت زیر است:

$$m = (v, S_{L_k}(v), P, sp, C)$$

دور ۲ (تعهد): گره‌ها پیام گره رهبر را وارسی می‌کنند و در صورت درستی، پیامی را به گره‌های مجموعه‌ی «تجمیع» ارسال می‌کنند که قسمت m دارای ساختار زیر است:

$$m = (v, k_{Leader}, idx, S_{L_k}, S_i(v))$$

در این ساختار v همان مقدار پیشنهادی گره رهبر است. k_{Leader} برابر شماره‌ی تکرار فعلی است و idx یک رشته بیت به طول n است. در صورتی که k_{Leader} برابر k باشد، مقدار idx اهمیتی ندارد. S_{L_k} امضای گره رهبر روی مقدار v است که دریافت شده است. گره‌ها این پیام را به گره‌های مجموعه‌ی «تجمیع»، ارسال می‌کنند.

دور ۳ (تجمیع اولیه): در این دور، گره‌های مجموعه‌ی «تجمیع»، یکسان بودن $v, k_{L_k}, r_{L_k}, S_{L_k}$ را در همه‌ی پیام‌های دریافتی، بررسی می‌کنند. در صورتی که بیش از یک مقدار وجود داشته باشد، دو مقدار را به همراه امضای گره رهبر، به تمام گره‌ها ارسال می‌کنند. ساختار m به صورت زیر است:

$$m = (1, v_1, k_{L_1}, r_{L_1}, idx_{L_1}, S_{L_1}, v_2, k_{L_2}, r_{L_2}, idx_{L_2}, S_{L_2})$$

این یک پیام از نوع یک است. در صورتی که در همه‌ی پیام‌های دریافتی، (v, k_L, r_L, S_L) یکسان باشد ساختار m به صورت نوع صفر است:

$$m = (0, v, k_L, r_L, S_L, idx, aggSig)$$

که در آن $aggSig$ نشان‌دهنده‌ی امضای تجمیعی امضاهایی است که از دور «تعهد» ارسال شده است ($S_i(v)$). متغیر idx یک رشته بیت به طول n است که مؤلفه‌ی iام نشان می‌دهد که گره‌ی iام در این امضای تجمیعی شرکت کرده یا خیر. در نهایت پیام به تمام گره‌ها ارسال می‌شود.

دور ۴ (اعلام): گره‌ها، در بین پیام‌های دریافتی از مجموعه‌ی «تجمیع»، در صورتی که پیامی از نوع یک مشاهده نکردند و در حداقل یکی از پیام‌های دریافتی (پیام c) در رشته بیت idx بیش از $f+1$ بیت برابر یک باشد، یک پیام به گره‌های مجموعه‌ی «تجمیع» ارسال می‌کنند و تصمیم خود را روی مقدار دریافت شده اخذ می‌کنند. گره‌ها پیام خود را بر مبنای پیام c تولید می‌کنند. ساختار m در این پیام به صورت زیر است:

$$m = (v, S_i(k), S_i(k, v), C)$$

مقدار v برابر $c \cdot m \cdot v$ است. C یک گواهی است که نشان می‌دهد، $f+1$ گره، پیامی را با مقدار v در دور «تعهد» ارسال کرده‌اند. در حقیقت C با استفاده از امضای تجمیعی، از امضای $f+1$ گره ارائه می‌شود.

$$C = (iterations, iterationIdx, roundIdx, aggSig)$$

در صورتی که تا به حال هیچ گره‌ای تصمیم نگرفته باشد، $iterations$ برابر یک مجموعه‌ی یک عضوی است که فقط شامل شماره‌ی تکرار فعلی است و $iterationIdx$ برابر $c \cdot m \cdot idx$ است. $roundIdx$ نیز

یک آرایه به طول n است که هر مؤلفه‌ی آن یک بیت است. در صورتی که هیچ گره‌ای تا به حال تصمیم نگرفته باشد همه‌ی مؤلفه‌های این آرایه برابر صفر است و مقدار $aggSig$ برابر $c \cdot m \cdot aggSig$ است.

دور ۵ (تجمیع ثانویه): گره‌های مجموعه‌ی «تجمیع»، اگر دو پیام «اعلام» با مقادیر v متفاوت، دریافت کنند، این دو پیام را به تمام گره‌ها ارسال می‌کنند. در این صورت ساختار m به صورت زیر است:

$$m = (۱, rr_۱, rr_۲)$$

اما در صورتی که همه‌ی پیام‌ها، v یکسان داشته باشد، یک امضای تجمیعی از پیام‌های دریافت شده ارائه می‌شود. ساختار m در این دور به صورت زیر است:

$$m = (۰, rr, idx, aggSigIt, aggSigVal)$$
$$rr = (uid, v, S_{uid}(k), S_{uid}(k, v), C)$$

در این ساختار rr یکی از پیام‌های «اعلام» دریافتی است و uid در آن، نشان‌دهنده‌ی شماره‌ی گره‌ای است که پیام «اعلام» را ارسال کرده است. idx یک رشته بیت به طول n است و نشان می‌دهد گره‌ی iام پیام «اعلام» را ارسال کرده یا خیر. $aggSigIt$ یک امضای تجمیعی از امضاهای $S_i(k)$ است که در دور قبل ارسال شده است. $aggSigVal$ نیز یک امضای تجمیعی از امضاهای $S_i(k, v)$ است که در دور قبل ارسال شده است.

در پایان این دور، گره‌ها پیام‌های دریافتی را وارسی می‌کنند. در صورتی که پیام از نوع یک در بین آن‌ها وجود داشت، یا دو پیام دریافتی از دو گره مختلف از مجموعه‌ی «تجمیع»، دارای مقدار v مختلف باشند، همه‌ی پیام‌های دریافتی را دور می‌ریزند. اما در غیر این صورت، پیام «تجمیع ثانویه» که از گره‌ی jام دریافت کرده‌اند را به $N_{i,j}$ اضافه می‌کنند. برای افزودن یک پیام جدید «تجمیع ثانویه» به فهرست $N_{i,j}$، باید بررسی شود که با اضافه کردن پیام «تجمیع ثانویه» جدید، هیچ گره‌ای دو پیام «اعلام» نخواهند داشت. به عبارتی برای هر دو عضو $a_۱, a_۲ \in N_{i,j}$ نباید اندیسی مثل q وجود داشته باشد که

$$a_۱ \cdot m \cdot idx[q] = a_۲ \cdot m \cdot idx[q] = ۱$$

در پایان دور «تجمیع ثانویه»، مقدار (v_i, C_i) را برابر یکی از مقادیر دریافتی قرار می‌دهیم. k_i نیز برابر شماره تکرار فعلی می‌شود.

وضعیت پروتکل پس از دریافت پیام‌های «تجمیع ثانویه»: پیام‌های «تجمیع ثانویه»، خود به نوعی یک پیام فشرده شده از تعداد زیادی پیام «اعلام» است. با توجه به آن که گره‌ها پس از ارسال پیام «اعلام» در تکرارهای بعدی، در دورهای ۰، ۱، ۲ و ۴ دیگر شرکت نمی‌کنند، باید پروتکل را کمی توسعه دهیم تا حضور آن‌ها به صورت مجازی ادامه داشته باشد.

دور ۰ (وضعیت): رهبر پیام‌های «تجمیع ثانویه» را مانند فشرده شده‌ی چند پیام «وضعیت» در نظر می‌گیرد و سعی می‌کند فقط با یکی از فهرست‌های پیام‌های «تجمیع ثانویه» مثل $N_{L_k,j}$، پیام «پیشنهاد» را بسازد. پیام گره رهبر به صورت زیر است:

$$m = (v, S_L(v), P, sp, C)$$

سپس گره رهبر، تعدادی از پیام‌های «وضعیت» دریافت شده مثل $\{s_۱, \ldots, s_u\}$ انتخاب می‌کند. همچنین تعدادی از پیام‌های فهرست $N_{L_k,j}$ مثل $\{a_۱, \ldots, a_w\} \subseteq N_{L_k,j}$ را انتخاب می‌کند. هر یک از a_g ها ساختاری به شکل زیر دارد:

$$a_g = (uid, k, r, m, S_{uid}(m))$$
$$m = (۰, rr, idx, aggSigIt, aggSigVal)$$
$$rr = (uid, v, S_{uid}(k), S_{uid}(k, v), C)$$

اگر $sum(a_g \cdot m \cdot idx)$ نشان دهنده‌ی تعداد بیت‌های یک در آرایه‌ی $a_g \cdot m \cdot idx$ باشد. باید داشته باشیم:

$$u + \sum_{i=۱}^{w} \mathrm{sum}(a_i \cdot m \cdot idx) \geq f + ۱$$

همچنین هیچ q وجود نداشته باشد که:

$$\exists\, g_۱, g_۲ \colon a_{g_۱} \cdot m \cdot idx[q] = ۱,\ a_{g_۲} \cdot m \cdot idx[q] = ۱$$

یا

$$\exists\, g_۱, g_۲ \colon a_{g_۱} \cdot m \cdot idx[q] = ۱,\ s_{g_۲} \cdot uid = q$$

سپس گره رهبر v متناظر با بزرگترین شماره‌ی تکرار را انتخاب می‌کند. بزرگترین شماره‌ی تکرار به صورت زیر به دست می‌آید:

$$k_{\max} = \max\ \left(\{a_g \cdot k\}_{g=۱}^{w}, \{s_t \cdot m \cdot k_i\}_{t=۱}^{u}\right)$$

اگر $k_{\max}$ از مجموعه‌ی $\{s_t \cdot m \cdot k_i\}_{t=۱}^{u}$ باشد، مقادیر C و sp مانند قبل از پیام متناظر انتخاب می‌شود. اما اگر $k_{\max}$ از مجموعه‌ی $\{a_g \cdot k\}_{g=۱}^{w}$ باشد، و $a_b \cdot k = k_{\max}$ در این صورت مقادیر C و sp به صورت زیر تعیین می‌شوند:

$$C = a_b \cdot m \cdot rr \cdot C$$
$$sp \cdot uid_{\max} = a_b \cdot m \cdot rr \cdot uid$$
$$sp \cdot k_{\max} = a_b \cdot k$$
$$sp \cdot r_{\max} = ۱$$

مقدار امضا در ساختار sp از روی امضای موجود در پیام $a_b \cdot m \cdot rr$ برداشته می‌شود تا ثابت کند مقدار v توسط $sp \cdot uid_{\max}$ امضا شده است.

P دارای ساختار زیر است:

$$(iterations, iterationIdx, roundIdx, aggSig)$$

مجموعه‌ی $iteration$ برابر $\{s_t \cdot m \cdot k_i\}_{t=۱}^{u} \cup \{a_g \cdot k\}_{g=۱}^{w}$ است. $roundIdx$ یک رشته بیت به طول n است. اگر یک g وجود داشته باشد که $a_g \cdot m \cdot idx[q] = ۱$ آنگاه $roundIdx[q] = ۱$، در غیر این صورت

$roundIdx[q] = 0$ است. مؤلفه‌ی qام $iterationIdx$ اندیس شماره‌ی تکرار پیام «اعلام» $(a_g \cdot k)$ یا پیام «وضعیت» $(s_t \cdot m \cdot k_i)$ در آرایه‌ی $iterations$ را نشان می‌دهد.

$aggSig$ امضای تجمیعی است و برای «تجمیع ثانویه» از $aggSigIt$ استفاده می‌شود تا امضای تجمیعی $aggSig$ تولید شود.

$$P \cdot aggSig = \prod_{t=1}^{u} s_t \cdot m \cdot sigIt \times \prod_{g=1}^{w} a_g \cdot m \cdot aggSigIt$$

نکته: اگر چند امضای تجمیعی داشته باشیم، تنها با ضرب کردن آن‌ها، می‌توان همه‌ی آن‌ها را به یک امضای تجمیعی تبدیل کرد.

دور ۱ (پیشنهاد): اگر در یکی از فهرست‌های پیام‌های «تجمیع ثانویه» مثل $N_{i,j}$، پیامی مثل a_g وجود داشت که $a_g \cdot m \cdot idx[L_k] = 1$، مقدار $a_g \cdot m \cdot v$ را مقدار پیشنهاد شده‌ی گره رهبر در نظر گرفته می‌شود. در صورتی که $a_g \cdot m \cdot v$ برابر v_i باشد، گره‌ها پیام «تعهد» را ارسال می‌کنند. پیام «تعهد» بر اساس a_g ارائه می‌شود.

در پیام «تعهد»، $k_{Leader} = a_g \cdot k$، $idx = a_g \cdot m \cdot idx$ و $S_{L_k} = a_g \cdot m \cdot aggSigVal$.

دور ۴ (اعلام): در این دور، در بین پیام‌های دریافتی، در صورتی که پیام از نوع یک مشاهده نشود، در مرحله‌ی بعد سعی می‌شود، با استفاده از فقط یکی از فهرست‌های پیام‌های «تجمیع ثانویه» مثل $N_{i,j}$، پیام «اعلام» ارائه شود. اگر c_j پیام دریافتی از گره‌ی jام باشد و $\{a_1, \ldots, a_w\}$ بزرگترین زیر مجموعه از مجموعه‌ی $N_{i,j}$ باشد به طوری که مقدار $a_g \cdot m \cdot v$ برای تمام $1 \le g \le w$ برابر $c_j \cdot m \cdot v$ باشد، در صورتی که نامساوی زیر برقرار بوده:

$$\mathrm{sum}(c_j \cdot m \cdot idx) + \sum_{g=1}^{w} \mathrm{sum}(a_g \cdot m \cdot idx) \ge f + 1$$

و همچنین هیچ q وجود نداشته باشد که:

$$\exists\, g_1, g_2 : a_{g_1} \cdot m \cdot idx[q] = 1,\ a_(g_2) \cdot m \cdot idx[q] = 1$$

یا

$$\exists\, g_1 : a_{g_1} \cdot m \cdot idx[q] = 1,\ c_j \cdot m \cdot idx[q] = 1$$

آنگاه پیام «اعلام» ارسال می‌شود. همان طور که قبلاً ذکر شد، ساختار m در پیام «اعلام» به صورت زیر است:

$$(v, S_i(k), S_i(k, v), C)$$
$$C = (iterations, iterationIdx, roundIdx, aggSig)$$

باید در گواهی C، پیام‌های «تجمیع ثانویه» نیز در نظر گرفته شود. $iterations$ یک مجموعه شامل شماره تکرار فعلی و تمام اعداد $\{a_g \cdot k\}_{g=1}^{w}$ است. $iterationIdx$ یک آرایه به طول n است. اگر در مجموعه‌ی $\{a_i \cdot m \cdot idx[q]\}_{i=1}^{w} \cup \{c_j \cdot m \cdot idx[q]\}$، همه صفر باشند، $iterationIdx[q]$ برابر صفر است. اگر $c_j \cdot m \cdot idx[q]$ برابر یک باشد، $iterationIdx[q]$ اندیس شماره تکرار فعلی در مجموعه‌ی $iterations$ را نشان می‌دهد. اگر $a_i \cdot m \cdot idx[q]$ برابر یک باشد، $iterationIdx[q]$ اندیس $a_i \cdot k$ را در مجموعه‌ی $iterations$ نشان می‌دهد. برای هر $0 \le q < n$ اگر g وجود داشته باشد که

$$a_g \cdot m \cdot idx[q] = 1,$$

آنگاه $roundIdx[q]$ برابر یک است. در غیر این صورت $roundIdx[q]$ برابر صفر است. $aggSig$ نیز که یک امضای تجمیعی است، با حاصل ضرب تمام امضاهای تجمیعی به دست می‌آید.

$$C \cdot aggSig = c_j \cdot m \cdot aggSig \times \prod_{g=1}^{w} a_g \cdot m \cdot aggSigVal$$

۲.۴ نکات

- گره‌هایی که در گروه «تجمیع» قرار دارند، حتی پس از ارسال پیام «اعلام»، باید در تکرارهای بعدی در دور ۳ و ۵ شرکت کنند. ولی در بقیه‌ی دورها لازم نیست شرکت کنند.
- در دور «پیشنهاد» و «اعلام» گره‌ها ممکن است با یکی از فهرست‌ها مثل $N_{i,j}$ موفق به ارائه پیام ارسالی نشوند. در این صورت باید سعی کنند با فهرست دیگری پیام «پیشنهاد» یا «اعلام» را تولید کنند.

۵ ارزیابی پروتکل پیشنهادی

در این بخش به تحلیل امنیت و سربار مخابراتی و بار محاسباتی می‌پردازیم. در بخش امنیت دو ویژگی «ایمنی» و «مانائی» بررسی می‌شوند. «ایمنی» به معنای آن است که اگر یک گره‌ی صادق مثل N در تکرار k^*، روی مقدار v^* به تصمیم برسد، بقیه‌ی گره‌های صادق نیز روی همین مقدار به تصمیم خواهند رسید. «مانائی» به این معنی است که گره‌های صادق، در زمانی، روی یک مقدار تصمیم می‌گیرند. اثبات این دو ویژگی با فرض وجود یک گره‌ی صادق در مجموعه‌ی «تجمیع» انجام می‌شود.

۱.۵ ایمنی

لم ۱. فرض کنید تا به حال هیچ گره‌ی صادقی، تصمیم نگرفته باشد. اگر در یک تکرار، دو پیام «اعلام» ارائه شود که روی دو مقدار مختلف هستند، آن‌گاه، هر دو پیام «اعلام» از سوی گره‌های بیزانسی ارائه شده است.

اثبات. فرض کنید دو پیام «اعلام» از سوی گره‌های N_1 و N_2 ارائه می‌شود. در هر دو پیام «اعلام»، یک گواهی وجود دارد که نشان می‌دهد $f + 1$ گره، برای یک مقدار، پیام «تعهد» یا پیام «اعلام» ارسال کرده‌اند. در پیام «اعلام» مربوط به گره‌ی N_1 در بین این $f + 1$ گره، قطعاً یک گره‌ی صادق مثل N_3 وجود دارد. برای N_3 دو حالت را می‌توان در نظر گرفت. او یا پیام «تعهد» را ارسال کرده است، یا پیام «اعلام» را ارسال

کرده است. از آن‌جا که فرض کردیم تا به حال هیچ گره‌ی صادقی تصمیم نگرفته است و از آن‌جا که N_3 یک گره‌ی صادق است، پس N_3 فقط می‌تواند پیام «تعهد» را ارسال کرده باشد.

N_3، پیام «تعهد» خود را به تمام گره‌های مجموعه‌ی «تجمیع» ارسال کرده است و در بین آن‌ها حتماً یک گره‌ی صادق مثل N_4 وجود دارد. برای N_4 دو حالت داریم.

حالت اول: گره‌ی N_4 دو پیام «تعهد» با مقادیر مختلف دریافت کرده باشد، در این صورت این پیام را به همه‌ی گره‌ها از جمله N_2 ارسال کرده است. پس اگر N_2 گره‌ی صادقی باشد، او نباید پیام «اعلام» را ارسال کند.

حالت دوم: گره‌ی N_4 در همه‌ی پیام‌های «تعهد» دریافتی، فقط یک مقدار v یکسان را ببینند. در این صورت گره‌ی N_4 پیام «تعهد» گره‌ی N_3 را به همه‌ی گره‌ها از جمله N_2 ارسال کرده باشد. پس اگر N_2 صادق باشد، نباید پیام «اعلام» را ارسال کند.

در این اثبات، می‌توان جای N_1 و N_2 را عوض کرد و به این نتیجه رسید که N_1 نیز نمی‌تواند یک گره‌ی صادق باشد. □

حال با استفاده از لم ۱.۵، «ایمنی» را اثبات می‌کنیم. فرض کنید N_1 اولین گره‌ی صادقی باشد که پیام «اعلام» را ارسال می‌کند. N_1 در تکرار k^*، روی مقدار v^* پیام «اعلام» را ارسال می‌کند. طبق لم ۱.۵، در تکرار k^*، پیام «اعلام» روی مقدار دیگری جز v^* نمی‌تواند ارائه شده باشد. این پیام «اعلام» به تمام گره‌های مجموعه‌ی «تجمیع» ارسال می‌شود و چون حداقل یکی از آن‌ها یک گره‌ی صادق است، پیام «اعلام» را در قالب پیام «تجمیع ثانویه»، به تمام گره‌های دیگر ارسال خواهد کرد. در این صورت همه‌ی گره‌های صادق مقدار v_i را به v^* تغییر می‌دهند. به این ترتیب دو حالت زیر متصور است:

۱- گره رهبر تکرار بعدی، قبلاً پیام «اعلام» را ارسال کرده باشد: در این صورت مقدار پیشنهادی مجازی او، در صورتی منجر به ارسال پیام «تعهد» در گره‌های صادق می‌شود که مقداری که قبلاً در پیام «اعلام» آورده، برابر v^* باشد.

۲- گره رهبر تکرار بعدی، هنوز پیام «اعلام» ارسال نکرده باشد: طبق استقرا همه‌ی گره‌های صادق، تا این تکرار روی مقدار v^* پیام «اعلام» را ارسال کرده‌اند و مقدار v_i برای گره‌های صادق، برابر v^* است. در این صورت برای آن‌که گره رهبر بعدی بتواند اثبات P را بسازد، باید حتماً از یکی از پیام‌های «اعلام» یا «وضعیت» گره‌های صادق استفاده کند. بنابراین او چاره‌ای جز پیشنهاد مقدار v^* نخواهد داشت.

در نتیجه در تکرار بعد گره‌های صادق تنها برای مقدار v^* پیام «تعهد» ارسال می‌کنند و بنابراین نمی‌تواند پیام «اعلام» برای مقدار $v_2 \neq v^*$ ارائه شود. زیرا باید شامل یک گواهی C باشد که وجود حداقل $f+1$ پیام «تعهد» یا «اعلام» را برای مقدار v_2 نشان دهد. حداقل یکی از آن‌ها، یک گره‌ی صادق است و او نیز پیام «تعهد» یا «اعلام»، برای مقدار v_2 ارسال نکرده است.

جدول ۱. مقایسه‌ی بار مخابراتی پروتکل پیشنهادی و Abraham

پروتکل	بار مخابراتی پروتکل
پروتکل پیشنهادی	$\mathcal{O}(m_a n^2 \lceil \log k_t \rceil + m_a n s_s)$
Abraham	$\mathcal{O}(n^3 s_s)$

۲.۵ مانائی

در هر کدام از تکرارها، اگر گره‌ی رهبر، صادق باشد، در همان تکرار، تمام گره‌های صادق به اجماع می‌رسند. فرض کنید حداکثر ۵۰ درصد گره‌ها تحت کنترل مهاجم باشد. از آن‌جا که رهبر تکرارهای مختلف، توسط یک پروتکل دیگر، به صورت تصادفی انتخاب می‌شوند، احتمال آن که پس از k_t تکرار، رهبر در همه‌ی تکرارها، از گره‌های بیزانسی باشد، حداکثر برابر $\frac{1}{2^{k_t}}$ است. همچنین پروتکل اجماع به صورت میانگین پس از دو تکرار (۱۲ دور)، به پایان می‌رسد.

۳.۵ بار مخابراتی

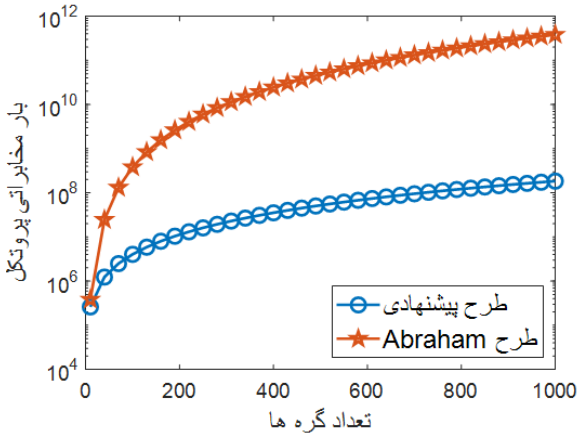
در این بخش به بررسی بار مخابراتی پروتکلی پیشنهادی می‌پردازیم و آن را با پروتکل Abraham [۲] مقایسه می‌کنیم. فرض کنید اندازه‌ی یک امضای دیجیتال s_s بیت باشد و پروتکل را حداکثر تا k_t تکرار انجام دهیم. قسمتی از پروتکل که بیشترین بار مخابراتی را دارد مربوط به ارسال پیام‌های «تعهد» و «اعلام» به مجموعه‌ی «تجمیع» است. در این پیام‌ها، آرایه‌ای به نام $iterationIdx$ وجود دارد که هر مؤلفه‌ی آن اندیس شماره‌ی تکرار را نشان می‌دهد. از آن‌جا که حداکثر k_t حالت مختلف برای شماره‌ی تکرار داریم، پس هر مؤلفه را با $\lceil \log k_t \rceil$ بیت می‌توانیم نشان دهیم. بنابراین می‌توان آرایه $iterationIdx$ را با $n\lceil \log k_t \rceil$ بیت نشان داد. آرایه‌ی $iterations$ نیز حداکثر k_t عضو دارد و چون $k_t \ll n$، آن را در نظر نمی‌گیریم. هر کدام از پیام‌های «تعهد» و «اعلام» به ترتیب ۳ و ۴ امضای دیجیتال را شامل می‌شوند. این تعداد از $\mathcal{O}(1)$ است. بنابراین بار مخابراتی پروتکل از $\mathcal{O}(m_a n^2 \lceil \log k_t \rceil + m_a n s_s)$ است.

در پروتکل Abraham، قسمت اعظم بار مخابرتی مربوط به پیام «اعلام» است که در آن باید یک گواهی شامل امضای حداقل $f+1$ گره، به تمام گره‌ها ارسال شود.

از آن‌جا که m_a، k_t و s_s مقدارشان مستقل از n است، این مقادیر می‌تواند نسبت به n بسیار کوچکتر باشد و پروتکل پیشنهادی، بار مخابراتی کمتری داشته باشد.

۴.۵ بار محاسباتی

در پروتکل پیشنهادی، در دور «تجمیع ثانویه» زمانی که گره‌های مجموعه‌ی «تجمیع» پیام‌های «اعلام» را دریافت می‌کنند، برای هر پیام لازم است



شکل ۱. نمودار بار مخابراتی بر حسب تعداد گره، برای حالتی که $m_a = ۳۲$، $s_s = ۳۸۴$ و $k_t = ۳۲$

جدول ۲. مقایسه‌ی بار محاسباتی پروتکل پیشنهادی و Abraham

پروتکل	بار محاسباتی پروتکل
پروتکل پیشنهادی	$\mathcal{O}(m_a n^۲)$
Abraham	$\mathcal{O}(n^۳)$

امضای تجمیعی را وارسی کنند. بار محاسباتی برای وارسی یک امضای تجمیعی توسط n گره، از مرتبه‌ی $\mathcal{O}(n)$ است. همچنین چون حداکثر n پیام دریافت می‌شود و m_a گره در مجموعه‌ی «تجمیع» وجود دارد، مجموع بار محاسباتی از $\mathcal{O}(n^۲ m_a)$ است. بار محاسباتی بقیه‌ی دورها نیز کمتر یا مساوی دور «تجمیع ثانویه» است. اما در پروتکل Abraham قسمت اعظم بار محاسباتی مربوط به وارسی پیام‌های «اعلام» است. برای وارسی هر پیام «اعلام»، باید حداکثر n امضای دیجیتال وارسی شود. چون حداکثر n پیام «اعلام» دریافت می‌شود و n گره وجود دارد، در نتیجه مجموع بار محاسباتی از $\mathcal{O}(n^۳)$ است.

۶ نتیجه

در این مقاله، ما یک روش اجماع جدید که نسخه‌ی بهبود یافته پروتکل Abraham است معرفی و نشان داده شد که پروتکل پیشنهادی ویژگی‌های «ایمنی» و «مانائی» را برآورده می‌کند. بار مخابراتی پروتکل پیشنهادی از $\mathcal{O}(m_a n^۲ \lceil \log k_t \rceil + m_a n s_s)$ و بار محاسباتی آن از $\mathcal{O}(m_a n^۲)$ است و در مقایسه با پروتکل Abraham بار مخابراتی و بار محاسباتی نسبت به پارامتر n، یک درجه، کاهش یافته است. در بهبود پروتکل پیشنهادی از امضای تجمیعی استفاده شده و پیام‌ها به دسته‌ی کوچکی از گره‌ها ارسال می‌شود. پروتکل پیشنهادی همانند پروتکل پایه تا ۵۰ درصد در برابر بیزانس‌ها مقاومت دارد. این پروتکل همگام بوده و دارای ۶ دور با تکرار است. در شرایطی که نزدیک ۵۰ درصد از گره‌ها بیزانسی باشند با توجه به انتخاب تصادفی رهبر در هر تکرار، به صورت میانگین، پس از دو تکرار (۱۲ دور) اجماع حاصل می‌شود. پروتکل پیشنهادی به دلیل نیاز به صادق بودن یکی از گره‌های «تجمیع»، با احتمال $۱ - \frac{۱}{۲^{m_a}}$، به درستی کار می‌کند.

جدول ۳. نمادها

n	تعداد گره‌ها
f	تعداد گره‌های بیزانسی
v	مقدار پیشنهادی رهبر
C	یک گواهی برای مقدار v که در آن $f + ۱$ پیام «تعهد» یا «اعلام» فشرده شده است.
v_i	مقدار پذیرفته شده‌ی گره‌ی iام
k_i	شماره تکرار مربوط به v_i
C_i	گواهی مربوط به v_i
$N_{i,j}$	لیستی از پیام‌های «تجمیع ثانویه» که گره‌ی iام از گره‌ی jام در مجموعه‌ی «تجمیع» دریافت کرده است.
k	شماره تکرار فعلی
r	شماره دور فعلی
uid	شماره‌ی گره که عددی بین ۰ تا $n - ۱$ است
$\mathrm{sign}_i(x)$	امضای x توسط گره‌ی iام
$S_i(x)$	$\mathrm{sign}_i(uid, k, r, x)$
m_a	تعداد گره‌های مجموعه‌ی «تجمیع»
s_s	تعداد بیت‌های امضای دیجیتال
P	اثباتی که رهبر ارائه می‌دهد تا نشان دهد شماره‌ی تکرار پیشنهادی، در بین $f + ۱$ پیام دریافتی، بیشترین مقدار است.
sp	پیام «وضعیت» یا «تجمیع ثانویه» دریافت شده‌ی رهبر که متناظر با بیشترین شماره‌ی تکرار است.
idx	یک رشته بیت به طول n که مشارکت گره‌ها در امضای تجمیعی را نشان می‌دهد.
$iterationIdx$	یک آرایه به طول n که مؤلفه‌ی iام نشان می‌دهد که گره‌ی iام، آیا در امضای تجمیعی شرکت داشته یا خیر. اگر شرکت داشته، امضایش مربوط به کدام تکرار است.
$roundIdx$	یک رشته بیت به طول n که مؤلفه‌ی iام نشان می‌دهد امضای گره‌ی iام، مربوط به دور «اعلام» بوده است یا خیر

مراجع

[1] Castro, Miguel, Liskov, Barbara, et al. Practical byzantine fault tolerance. 99(1999):173–186, 1999.

[2] Abraham, Ittai, Devadas, Srinivas, Dolev, Danny, Nayak, Kartik, and Ren, Ling. Efficient synchronous byzantine consensus. *arXiv preprint arXiv:1704.02397*, 2017.

[3] Boneh, Dan, Gentry, Craig, Lynn, Ben, and Shacham, Hovav. Aggregate and verifiably encrypted signatures from bilinear maps. in *Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Appli-*

cations of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22, pp. 416–432. Springer, 2003.

[4] Schindler, Philipp, Judmayer, Aljosha, Hittmeir, Markus, Stifter, Nicholas, and Weippl, Edgar. Randrunner: Distributed randomness from trapdoor vdfs with strong uniqueness. 2021.

[5] Bhat, Adithya, Shrestha, Nibesh, Luo, Zhongtang, Kate, Aniket, and Nayak, Kartik. Randpiper–reconfiguration-friendly random beacons with quadratic communication. in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3502–3524, 2021.

[6] Zamani, Mahdi, Movahedi, Mahnush, and Raykova, Mariana. Rapidchain: Scaling blockchain via full sharding. in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 931–948, 2018.

[7] Kokoris Kogias, Eleftherios, Jovanovic, Philipp, Gailly, Nicolas, Khoffi, Ismail, Gasser, Linus, and Ford, Bryan. Enhancing bitcoin security and performance with strong consistency via collective signing. USENIX Association, 2016.

[8] Buterin, Vitalik and Griffith, Virgil. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017.

[9] Aguilera, Marcos K, Delporte-Gallet, Carole, Fauconnier, Hugues, and Toueg, Sam. Communication-efficient leader election and consensus with limited link synchrony. in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pp. 328–337, 2004.

[10] Syta, Ewa, Tamas, Iulia, Visher, Dylan, Wolinsky, David Isaac, Jovanovic, Philipp, Gasser, Linus, Gailly, Nicolas, Khoffi, Ismail, and Ford, Bryan. Keeping authorities" honest or bust" with decentralized witness cosigning. in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 526–545. Ieee, 2016.

[11] Harmony Team. Harmony technical whitepaper, version 2.0, [Accessed 2022]. [Online]. Available: https://harmony.one/whitepaper.pdf.

[12] Michels, Markus and Horster, Patrick. On the risk of disruption in several multiparty signature schemes. in *Advances in Cryptology—ASIACRYPT'96: International Conference on the Theory and Applications of Cryptology and Information Security Kyongju, Korea, November 3–7, 1996 Proceedings*, pp. 334–345. Springer, 1996.

[13] Dwork, Cynthia, Lynch, Nancy, and Stockmeyer, Larry. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)*, 35(2):288–323, 1988.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2022, Volume 20, Number 2 (pp. 22–32)
https://monadi.isc.org.ir

Presented at the ISCISC 2022 in University of Giulan, Rasht, Iran

Consensus Method with High Security and Low Communication Load★

Alireza Hediehloo *,1, Javad Mohajery 2 and Mohammadreza Aref 1

1 *Electrical enginearing, Sharif University of Technology, Tehran, Iran*
2 *Electronics Research Institute, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Keywords:
Consensus
Synchronous network
Byzantine node
Collective signature

dor: 20.1001.1.24763047.1401.11.2.3.2

Type: research paper

ABSTRACT

Consensus protocols are used to establish coordination between network nodes and increase the resistance of distributed systems against errors. In this paper, a new synchronous consensus protocol is introduced. The proposed protocol is an improved version of Abraham protocol in which we have used aggregate signature to reduce the communication load. In the Abraham protocol, communication load and calculation load of the protocol are of the order $\mathcal{O}(n^3 s_s)$ and $\mathcal{O}(n^3)$, respectively, while in the proposed protocol, communication load and calculation load are of the order $\mathcal{O}(m_a n^2 \log k_t)$ and $\mathcal{O}(m_a n^2)$, respectively, where n is the number of network nodes, s_s is the size of a digital signature, k_t is the maximum number of protocol iterations, and m_a is a security parameter that can be much smaller than n. Therefore, in networks with a large number of nodes, the reduction of communication load and computing load will be noticeable. Also, in this protocol, we need at least $n = 2f + 1$ nodes to resist f Byzantine node, and the consensus process is done correctly with a probability of at least $1 - \frac{1}{2^{m_a}}$.

© 2022 ISC

★ The ISCISC 2022 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: alireza.hediehloo@ee.sharif.edu (Alireza Hediehloo), mohajer@sharif.ir (Javad Mohajery), aref@sharif.edu (Mohammadreza Aref)
© 2022 ISC. All rights reserved.