به نام خدا

دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری ۳۰ اردیبهشت ۹۲

جلسه‌ی بیست‌وچهارم: رأی‌گیری الکترونیکی

مدرس: دکتر شهرام خزائی *نگارنده: شفیع قلیزاده*

۱ مقدمه

برگزاری انتخابات از مهمترین وظایف هر دولت مردم‌سالار است. از سوی دیگر اما به درستی انجام دادن این وظیفه، برای هر دولتی می‌تواند یك چالش مهم نیز محسوب شود؛ چه اینکه برآورده ساختن ملزومات و رعایت محدودیت‌های آن به سادگی میسر نیست. ممکن است در نگاه اول چنین به نظر آید که برگزاری رأی‌گیری با دستگاه‌های رأی‌گیری مستقیم الکترونیکی، DRE[۱]، مشکلات و محدودیت‌های مورد بحث را به سادگی مرتفع می‌سازد و در عین حال سرعت انجام عملیات را نیز بهبود می‌بخشد. اما رأی‌گیری الکترونیکی نیز ملزومات امنیتی خاص خود را دارد که بسیار متنوع و گاهی حتی متناقض هستند. برای نمونه، هر رأی‌دهنده باید این امکان را داشته باشد تا مطمئن شود رأی‌اش به درستی و چنان که مورد نظرش بوده ثبت شده است. اما در عین حال برای جلوگیری از خرید و فروش رأی، همین رأی‌دهنده نباید بتواند به دیگران ثابت کند که چه رأیی داده است. به‌کارگیری تکنیك‌های رمزنگاری در رأی‌گیری الکترونیکی برای پاسخ‌گویی به چنین دغدغه‌هایی است.

در ادامه به بررسی اصول کلی رأی‌گیری الکترونیکی خواهیم پرداخت. در حالت کلی در این بحث دو موضوع در کانون توجه قرار می‌گیرد: صحت[۲] رأی‌گیری و حفظ حریم خصوصی[۳].

۲ معرفی یك سیستم رأی‌گیری مقدماتی

در اینجا قصد داریم تا با استفاده از ساده‌ترین فرض‌های ممکن، طرحی مقدماتی برای انجام یك رأی‌گیری الکترونیکی ارائه کنیم. در ادامه به بررسی ملاحظات امنیتی مهمی خواهیم پرداخت که می‌تواند این طرح مقدماتی را مورد چالش قرار دهد. فرض می‌کنیم در سیستم رأی‌گیری مورد نظر، N رأی‌دهنده داریم و رأی هر یك از آنان نیز تنها می‌تواند صفر یا یك باشد. چنین فرضی به طور خاص در همه‌پرسی‌هایی برقرار است که به صورت بلی و خیر برگزار می‌گردند. رأی شخص iام را با v_i نشان می‌دهیم که $v_i \in \{۰, ۱\}$ برای $i = ۱, ۲, \ldots, N$.

[۱] Direct Record Electronic
[۲] Correctness
[۳] Privacy

می‌خواهیم از سیستم رمز الگمال[۴] در طرح رأی‌گیری استفاده نماییم. با این سیستم رمز در جلسات گذشته آشنا شده‌ایم. به طور مختصر این سیستم رمز را مرور می‌کنیم.

الگوریتم تولید کلید:

$$(\mathbb{G}, q, g) \leftarrow \mathsf{GroupGen}(1^n)$$ (یك گروه دوری $\mathbb{G}$ با مرتبه q و مولد g تولید می‌کند.)

$$x \leftarrow \mathbb{Z}_q$$
$$h = g^x$$
$$pk = (\mathbb{G}, q, g, h)$$
$$sk = (\mathbb{G}, q, g, x)$$

الگوریتم رمزگذاری:

$$r \leftarrow \mathbb{Z}_q$$
$$\langle g^r, m \cdot h^r \rangle = \mathsf{Enc}_{pk}(m)$$

الگوریتم رمزگشایی:

$$\frac{c_2}{c_1^x} = \mathsf{Dec}_{sk}(\langle c_1, c_2 \rangle)$$

در این سیستم رمز، فرض بر این است که مساله دیفی‌ـهلمن تصمیمی[۵] برای گروه مورد استفاده سخت است.
برای استفاده از سیستم الگمال در این طرح باید تغییر مختصری در سیستم الگمال اعمال کنیم؛ بدین صورت که به جای آنکه هر پیام (m_i) را برابر با رأی متناظر (v_i) قرار دهیم، آن را برابر با g^{v_i} قرار می‌دهیم. یعنی:

$$\begin{aligned} \langle g^{r_i}, g^{v_i} \cdot h^{r_i} \rangle &= \mathsf{Enc}_{pk}(m_i) \\ &= \mathsf{Enc}_{pk}(g^{v_i}) \\ &= \mathsf{Enc}'_{pk}(v_i) \end{aligned}$$

بدین ترتیب رأی‌دهنده‌ها متن‌های رمزشده زیر را تولید و منتشر می‌کنند، به طوری که عموم افراد می‌توانند بررسی کنند که هر شخص چه رأی رمزشده‌ای را ارسال کرده است.

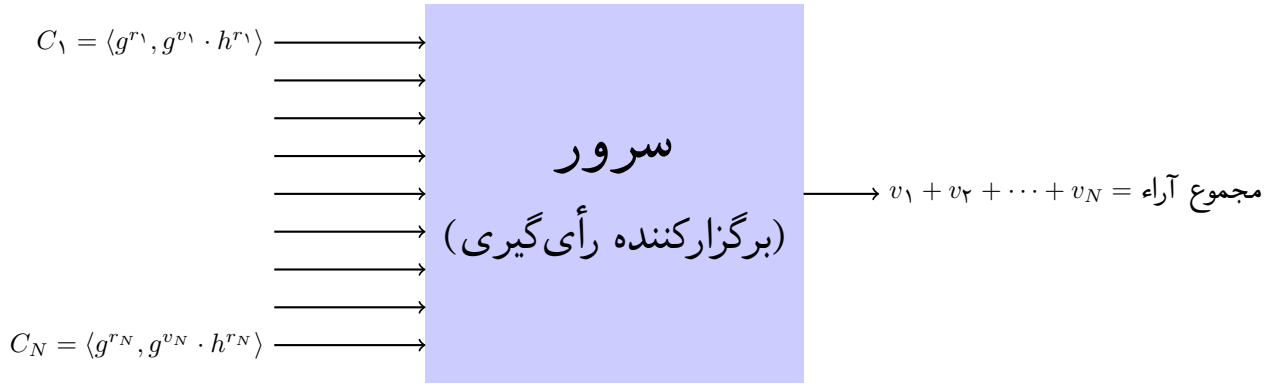
$$\begin{aligned} C_1 &= \mathsf{Enc}'_{pk}(v_1) \\ C_2 &= \mathsf{Enc}'_{pk}(v_2) \\ \vdots &\quad\quad \vdots \\ C_N &= \mathsf{Enc}'_{pk}(v_N) \end{aligned}$$

دقت کنید که با توجه به امنیت سیستم رمز الگمال، رأی‌های رمزشده اطلاعاتی در مورد مقدار رأی نشت نمی‌دهند. همچنین با تغییر کوچکی که در سیستم رمز الگمال حاصل کرده‌ایم، خاصیت هوموورفیك ضربی آن را به خاصیت هوموورفیك جمعی تبدیل کردیم. زیرا:

$$\begin{aligned} C_1 \cdot C_2 \cdots C_N &= \mathsf{Enc}(m_1 \cdot m_2 \cdots m_N) \\ &= \mathsf{Enc}(g^{v_1+v_2+\cdots+v_N}) \\ &= \mathsf{Enc}'(v_1 + v_2 + \cdots + v_N) \end{aligned}$$

بنابراین با دانستن آرای رمزشده خواهیم توانست حاصل‌ضرب آن‌ها را محاسبه و رمزگشایی کنیم تا به مجموع آراء موافق (برابر با یك) دست پیدا کنیم. در عمل این کار توسط یك نهاد معتمد که سرور[۶] نیز نامیده می‌شود انجام می‌گردد.

[۴] El Gamal
[۵] Decisional Diffie-Hellman
[۶] Server



۳ مشکلات سیستم رأی‌گیری معرفی شده

در طرح معرفی شده برای رأی‌گیری مشکلات متعددی وجود دارد که باید برطرف گردند. فهرست برخی از این مشکلات در ذیل آمده است.

- **امکان تقلب نهاد برگزارکننده:** در این سیستم شاید نهاد برگزارکننده رأی‌گیری، آراء را به درستی رمزگشایی نکند.
- **امکان تقلب رأی‌دهنده با ارسال رأی غیر صفر یا یک:** ممکن است رأی‌دهنده به جای صفر یا یك عدد دیگری را رمز کند. در این حالت اگر شخصی عدد مثبت n را رمز کند، در حقیقت n بار رأی موافق (برابر یك) داده و اگر عدد منفی n را رمز کند، در حقیقت n تا از آرأی موافق را حذف کرده است. این درحالی‌است که تعداد کل رأی‌دهنده‌ها ثابت مانده است.
- **امکان افشای حریم خصوصی رأی‌دهندگان توسط نهاد برگزارکننده:** در این سیستم نهاد برگزارکننده عملاً به محتوای یکایك آراء دسترسی خواهد داشت و تواند تشخیص دهد که هر رأی‌دهنده چه رأیی داده است.
- **امکان ارسال رأی مشابه از طرف رأی‌دهنده:** چنانچه رأی‌گیری مطابق سیستم معرفی شده برگزار گردد، افراد می‌توانند رأی شخصی دیگر (مثلا شخصی مهم یا معروف) را عیناً تقلید کنند بدون آنکه از محتوای آن آگاهی داشته باشند. دلیل این کار می‌تواند سرسپردگی[۷]، علاقه شخصی و یا هر مورد دیگری باشد. قابل توجه است که در حالت خاصی که تنها سه رأی‌دهنده داشته باشیم، این امر می‌تواند به فاش شدن آرای دو شخص دیگر برای هر یك از رأی‌دهنده‌ها منجر شود.

از میان مشکل‌های یادشده، موارد اول و دوم به مساله صحت رأی‌گیری مربوط هستند. سه مورد دیگر اما به مساله حفظ حریم خصوصی رأی‌دهندگان باز می‌گردد. این سیستم یک مشکل دیگر نیز دارد که البته ویژگی کلی سیستم‌های رأی‌گیری الکترونیکی است:

- **امکان بروز اجبار، تهدید و تطمیع رأی‌دهنده:** در این سیستم جلوی اجبار و تهدید و فروش رأی گرفته نمی‌شود؛ چرا که عدد تصادفی r که هنگام رمزکردن رأی هر رأی‌دهنده انتخاب می‌گردد، عملاً ممکن است به عنوان یك

[۷] Devotion

رسید از سوی همان رأی‌دهنده به کار رود. در حقیقت از این طریق، هر رأی‌دهنده می‌تواند ثابت کند که به چه کسی رأی داده است. با این حساب، ممکن است رأی‌دهنده مورد تهدید قرار گرفته و یا پیشاپیش اقدام به فروش رأی خود بنماید.

آنچه در پی می‌آید راه‌حل‌هایی کلی هستند که برای برطرف ساختن مشکلات مربوط به صحت و حریم خصوصی، می‌توانند مورد استفاده قرار گیرند. مشکل مربوط به امکان بروز اجبار، تهدید و تطمیع رأی‌دهنده باید به نحو دیگری حل شود که موضوع این بحث نیست.

۱.۳ اثبات دانش در مورد محتوای رأی

چنان‌که گفته شد ممکن است رأی‌دهنده‌ای بخواهد رأی شخص مهمی را مستقیماً کپی کند بدون آنکه از محتوای آن آگاهی داشته باشد. ممکن است به نظر برسد که با حذف آرای رمزشده تکراری، می‌توان جلوی این مساله را گرفت. اما حتی با اضافه کردن این مرحله نیز باز در سیستم یادشده چنین عملی امکان‌پذیر است؛ چرا که سیستم رمز الگمال قابلیت رمزگذاری مجدد[۸] را دارا است و در صورتی که رأی‌دهنده‌ای متن رمزشده یك رأی‌دهنده دیگر را مجدداً رمزگذاری کند، متن رمزشده تغییر می‌کند، اما رأی داده شده عوض نمی‌گردد.

برای حل مشکل امکان ارسال رأی مشابه، رأی‌دهنده باید بتواند دانش خود را در مورد محتوای رأی داده شده ثابت کند. این اثبات باید ناتراوا[۹] باشد. یعنی رأی‌دهنده در عین حال که دانش خود را درباره رأیی که داده ثابت می‌کند، نباید هیچ اطلاعات اضافی در مورد محتوای رأی بروز دهد. در این مورد کافی است تا رأی‌دهنده iام به جای اثبات دانش در مورد m_i، دانش خود را در مورد r_i ثابت کند. چرا که با توجه به رابطه زیر، دانش در مورد r_i بیانگر دانش در مورد m_i است.

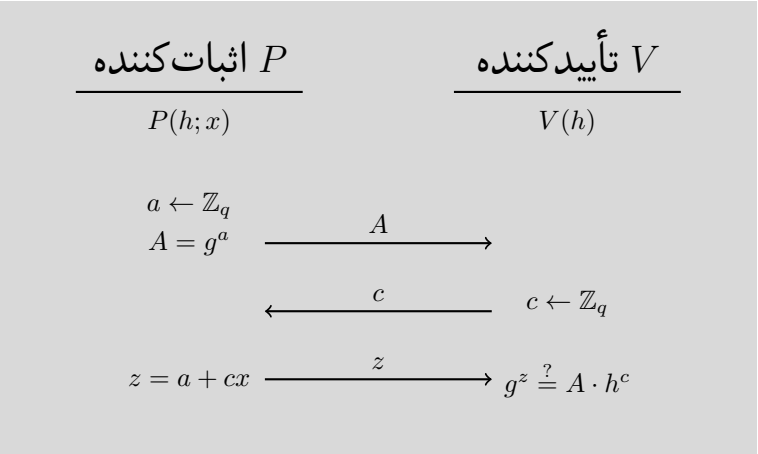
$$m_i = \frac{g^{r_i}}{h^{r_i}}$$

برای این منظور با فرض $h = g^x$ پروتکل زیر را بین اثبات‌کننده[۱۰] با ورودی x و تأییدکننده[۱۱] با ورودی h در نظر بگیرید.

اثبات‌کننده P		تأییدکننده V
$P(h;x)$		$V(h)$
$a \leftarrow \mathbb{Z}_q$ $A = g^a$	$\xrightarrow{A}$	
	$\xleftarrow{b}$	$b \leftarrow \{0,1\}$
$z = \begin{cases} a & b=0 \\ a+x & b=1 \end{cases}$	$\xrightarrow{z}$	$g^z \stackrel{?}{=} A \cdot h^b$

[۸] Re-encryption
[۹] Zero-knowledge
[۱۰] Prover
[۱۱] Verifier

در این آزمون، اثبات‌کننده قرار است نشان دهد که مقدار x را می‌داند. برای این منظور عدد تصادفی a را از $\mathbb{Z}_q$ انتخاب می‌کند و سپس مقدار g^a را برای تأییدکننده می‌فرستد. تأییدکننده بیت تصادفی b را تولید کرده و برای اثبات‌کننده ارسال می‌کند. بر اساس مقدار بیت b، اثبات‌کننده مقدار z را محاسبه کرده و آن را به تأییدکننده بازمی‌گرداند. نهایتاً کافی است تأییدکننده که مقادیر A، h، b و z را می‌داند، بر اساس این مقادیر، $A \cdot h^b$ را محاسبه کرده و با مقدار g^z مقایسه کند.
در این آزمایش حالت $b = 0$ به این دلیل تعبیه شده است که اگر تأییدکننده همواره قرار باشد بیت یك را برای اثبات‌کننده بفرستد، اثبات‌کننده می‌تواند دو مقدار z و A را طوری انتخاب کند که $g^z = A \cdot h^b$ و در مرحله نخست آزمایش همان مقدار A را برای تأییدکننده بفرستد. قابل توجه است که این آزمایش با احتمال $\frac{1}{2}$ دانش اثبات‌کننده را می‌آزماید. یعنی اگر اثبات‌کننده مقدار x را هم نداند باز هم با احتمال $\frac{1}{2}$ شانس پیروزی دارد. اما اگر این آزمایش چند بار تکرار شود به پروتکلی خواهیم رسید که امکان تقلب از سوی تأییدکننده را به اندازه دلخواه ناچیز می‌سازد. پروتکل زیر، که به پروتکل اشنور[۱۲] معروف است، همین امکان را در یك اجرا فراهم می‌کند.



توجیهی که کارآمدی این پروتکل را نشان می‌دهد آن است که اگر اثبات‌کننده به ازای دو مقدار متفاوت c بتواند آزمایش را با موفقیت بگذراند، حتما مقدار x را می‌داند. هر چند باید توجه داشت که عملا اثبات‌کننده به ازای دو مقدار متفاوت c آزمایش را تکرار نمی‌کند، زیرا در غیر این صورت تأییدکننده می‌تواند مستقیما مقدار x را محاسبه نماید.
فرض کنید اثبات‌کننده پس از ارسال مقدار A در مرحله اول و دریافت مقادیر c و c' در مرحله دوم، بتواند در مرحله سوم پروتکل را به ترتیب با ارسال مقادیر z و z' با موفقیت به اتمام برساند. یعنی:

$$g^z = A \cdot h^c$$
$$g^{z'} = A \cdot h^{c'}$$

[۱۲] Schnorr Protocol

بدین ترتیب، مقدار x با استفاده از این دو اجرای پروتکل به راحتی قابل محاسبه است:

$$g^{z-z'} = h^{c-c'} \qquad \Rightarrow h = g^{(z-z')(c-c')^{-1}} \qquad \Rightarrow x = \frac{z-z'}{c-c'} \mod q$$

توجه کنید که در پروتکل اشنور، تأییدکننده نمی‌تواند اطلاعات خاصی را در حین آزمایش کسب کند، چرا که به تنهایی نیز می‌توانسته از ابتدا با شبیه‌سازی، یک سه‌تایی (A, c, z) را طوری تولید کند که توزیع آن با آنچه که در یک تعامل با اثبات‌کننده می‌بیند، یکسان باشد. برای این کار، کافی است ابتدا z و c را به صورت تصادفی تولید و سپس مقدار A را بر حسب آنها محاسبه نماید.

$$c \leftarrow \mathbb{Z}_q$$
$$z \leftarrow \mathbb{Z}_q$$
$$A \leftarrow \frac{g^z}{h^c}$$

این (A, c, z) تولید شده توزیعی عیناً برابر با آنچه تأییدکننده صادق[۱۳] در یک تعامل با اثبات‌کننده به دست می‌آورد، دارد. دقت کنید که تأییدکننده ناصادق[۱۴] می‌تواند مقدار c را کاملا تصادفی انتخاب نکند و مثلا آن را وابسته به مقدار دریافتی A کند. با این وجود، هنوز هیچ شاهدی مبنی بر این که این نکته می‌تواند یک ضعف در پروتکل یادشده باشد، گزارش نشده است.

۲.۳ اثبات صحت رمزگشایی

چنان که پیشتر اشاره شد یکی از مشکلاتی که ممکن است در سیستم رمزگذاری معرفی شده به وقوع بپیوندد این است که نهاد برگزارکننده رأی‌گیری آراء را به درستی رمزگشایی نکند. در این مورد برگزارکننده باید ثابت کند که آراء را به درستی رمزگشایی نموده است.
فرض کنید $\langle u, v \rangle$ حاصل ضرب کلیه آراء رمزشده ارسالی باشد. دقت کنید که هر فردی می‌تواند با ضرب کردن همه آراء که در دسترس عموم قرار دارند، مقادیر u و v را محاسبه کند. بنابراین چهارتایی زیر اطلاعاتی است که در اختیار عموم است، اما تنها سرور برگزارکننده کلید خصوصی x را در می‌داند.

$$(g, h, u, v) = (g, g^x, g^r, m \cdot g^{rx})$$

که

$$m = g^{\sum_{i=1}^{N} v_i}$$

$$r = \sum_{i=1}^{N} r_i$$

همچنین پس از انتشار آرای تجمیع شده، یعنی $v_1 + \cdots + v_N$، توسط برگزار کننده، هر فردی می‌تواند مقدار m را محاسبه کند. بنابراین نهاد برگزارکننده در واقع باید ثابت کند که متن رمزی $\langle u, v \rangle$ به درستی به m رمزگشایی می‌شود. چهارتایی زیر را در نظر بگیرید:

$$(g_1, h_1, g_2, h_2) = (g, h, u, \frac{v}{m}) = (g, g^x, g^r, g^{rx})$$

[۱۳] Honest
[۱۴] Corrupt

توجه کنید که دو لگاریتم گسسته زیر برابر اند.

$$\log_{g_1} h_1 = \log_{g_2} h_2 = x$$

بنابراین مساله اثبات صحت رمزگشایی به مساله اثبات برابری لگاریتم گسسته تبدیل می‌گردد. یعنی برگزارکننده می‌تواند به طور معادل ثابت کند دو لگاریتم فوق گسسته برابرند. دقت کنید که این کار بدون افشای هیچ‌گونه اطلاعاتی در مورد مقدار x باید صورت گیرد.
برای اثبات برابری لگاریتم گسسته نیز پروتکل زیر را بین اثبات‌کننده با ورودی x و تأییدکننده با ورودی‌های (g_1, h_1, g_2, h_2) معرفی می‌کنیم. در این پروتکل، روابط زیر بین ورودی‌ها صادق است.

$$h_1 = g_1^x$$
$$h_2 = g_2^x$$

اثبات‌کننده P		تأییدکننده V
$P(g_1, h_1, g_2, h_2; x)$		$V(g_1, h_1, g_2, h_2)$
$a \leftarrow \mathbb{Z}_q$		
$A_1 = g_1^a$ $A_2 = g_2^a$	$\xrightarrow{A_1, A_2}$	
	$\xleftarrow{c}$	$c \leftarrow \mathbb{Z}_q$
$z = a + cx$	$\xrightarrow{z}$	$g_1^z \stackrel{?}{=} A_1 \cdot h_1^c$ $g_2^z \stackrel{?}{=} A_2 \cdot h_2^c$

عملکرد این پروتکل همانند اجرای هم‌زمان دو نسخه پروتکل اشنور است. قابل توجه است که در این پروتکل تأییدکننده علاوه بر این که ثابت می‌کند دو لگاریتم گسسته برابر اند، دانش خود به مقدار x را نیز اثبات می‌کند. نکته مهم دیگر این است که چنانچه رمزگشایی آراء به صورت توزیع شده نیز انجام یابد، همچنان می‌توان با استفاده از سیستمی مبتنی بر پروتکل یادشده، صحت رمزگشایی در هر قسمت را احراز نمود. در مورد رمزگشایی توزیع شده، در ادامه توضیحات لازم داده خواهد شد.

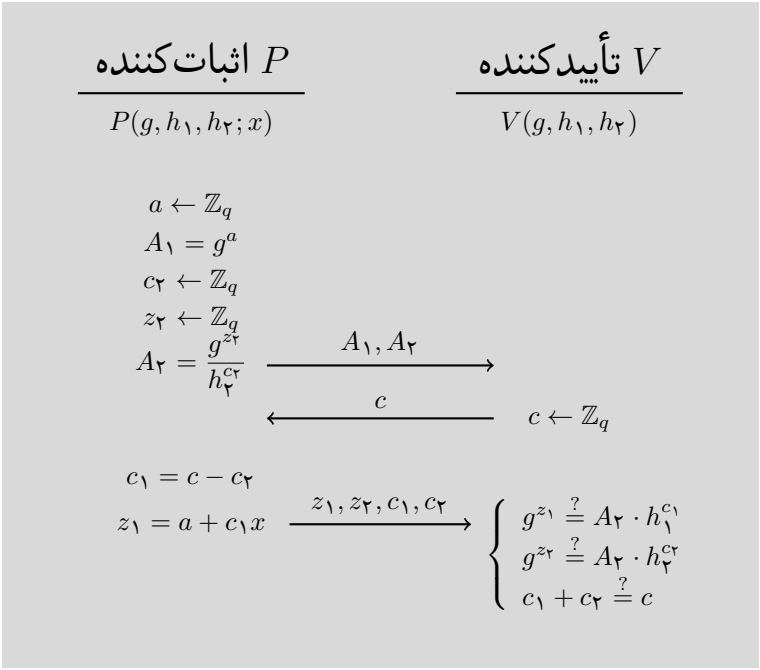
۳.۳ اثبات صفر یا یك بودن رأی داده شده

برای اینکه صحت فرآیند رأی‌گیری زیر سوال نرود، باید مطمئن باشیم که هیچ رأی‌دهنده‌ای مقداری غیر از صفر یا یك را رمز نمی‌کند. رأی‌دهنده‌ای را در نظر بگیرید که مقدار g^0 یا g^1 را با استفاده از مقدار تصادفی r توسط الگمال، رمز نموده و رأی رمز شده را به صورت $\langle u, v \rangle$ ارسال نموده است. یعنی:

$$\langle u, v\rangle = \langle g^r, g^{\circ} \cdot h^r\rangle \qquad \text{یا} \qquad \langle u, v\rangle = \langle g^r, g^{۱} \cdot h^r\rangle$$

در این مورد کافی است رأی‌دهنده ثابت نماید که لگاریتم گسسته $\log_g u$ برابر است با یکی از دو لگاریتم گسسته $\log_h v$ یا $\log_h v/g$. با استفاده از شگرد OR می‌توان پروتکل اثبات برابری لگاریتم گسسته را به پروتکلی برای این منظور تبدیل کرد.
در اینجا شگرد OR را برای پروتکل اثبات دانش لگاریتم گسسته مطرح می‌کنیم. ولی این شگرد OR کلی است و به طور مشابهی برای مسأله برابری لگاریتم نیز قابل تطبیق است.

شگرد OR. فرض کنید تأییدکننده مقادیر g، $h_۱$ و $h_۲$ را می‌داند و اثبات‌کننده در پی آن است که ثابت کند یکی از دو لگاریتم گسسته $\log_g h_۱$ یا $\log_g h_۲$ را می‌داند، بدون اینکه اطلاعاتی بدهد در مورد اینکه کدام یك را می‌داند و یا اطلاعاتی در مورد مقدار لگاریتم گسسته افشا کند. برای حل این مساله شگرد OR از پروتکل اثبات دانش لگاریتم گسسته به نحو مطلوبی استفاده می‌کند. در ذیل، این پروتکل برای حالتی توصیف شده است که اثبات‌کننده $\log_g h_۱$ را می‌داند. حالت دیگر که اثبات‌کننده $\log_g h_۲$ را می‌داند، به صورت مشابه با جابجا کردن اندیس‌ها به دست می‌آید.



در پروتکلی که توصیف شد، در حقیقت اثبات‌کننده سه‌تایی $(A_۲, c_۲, z_۲)$ را با شبیه‌سازی تولید می‌کند به طوری که رابطه $A_۲ = \frac{g^{z_۲}}{h_۲^{c_۲}}$ برقرار باشد. به همین دلیل اثبات‌کننده بی‌آنکه $x = \log_g h_۲$ را بداند، سه‌تایی معتبری تولید کرده است که شرط دوم تأییدکننده را ارضا می‌کند. داریم:

$$A_۲ = \frac{g^{z_۲}}{h_۲^{c_۲}} \Longrightarrow g^{z_۲} = A_۲ \cdot h_۲^{c_۲}$$

اما از طرف دیگر فرض کردیم که اثبات‌کننده $\log_g h_1$ را می‌داند. یعنی در واقع می‌داند که $x = \log_g h_1$. داریم:

$$\begin{aligned} x = \log_g h_1 &\Longrightarrow g^x = h_1 \\ &\Longrightarrow g^{c_1 x} = {h_1}^{c_1} \\ &\Longrightarrow g^a \cdot g^{c_1 x} = g^a \cdot {h_1}^{c_1} \\ &\Longrightarrow g^{a+c_1 x} = g^a \cdot {h_1}^{c_1} \\ &\Longrightarrow g^{z_1} = A_1 \cdot {h_1}^{c_1} \end{aligned}$$

بنابراین اثبات‌کننده هر سه رابطه تساوی مد نظر تأییدکننده را برآورده می‌سازد. دقت کنید که شرط سوم یعنی تساوی $c_1 + c_2 \stackrel{?}{=} c$ تنها به این دلیل در پروتکل تعبیه شده که اثبات کننده مجبور باشد مقدار c_1 را حتماً از طریق رابطه $c_1 = c - c_2$ محاسبه کند. به این ترتیب اثبات‌کننده نخواهد توانست سه‌تایی (A_1, c_1, z_1) را شبیه‌سازی کند و تنها در صورتی قادر به گذراندن آزمون نهایی پروتکل خواهد بود که واقعاً مقدار x یعنی لگاریتم گسسته h_1 در مبنای g را بداند.

۴.۳ رمزگشایی توزیع شده

پیشتر اشاره شد که در سیستم رأی‌گیری معرفی شده، نهاد برگزارکننده به آرای یکایک رأی‌دهندگان دسترسی دارد و می‌تواند مشخص سازد که هر فرد چه رأیی داده است. راه حل این مشکل اما شباهت چندانی به موارد گفته شده قبلی ندارد. برای حل این مشکل از رمزگشایی توزیع‌شده استفاده می‌کنیم.
در اینجا ابتدا مساله تسهیم راز[۱۵] را مطرح می‌کنیم و سپس به مساله اصلی بازخواهیم گشت.

مساله تسهیم راز

می‌خواهیم رازی مانند عدد $S \in \mathbb{Z}_p$ را بین n نفر تقسیم کنیم به طوری که این n نفر به همراه یکدیگر بتوانند عدد S را پیدا کنند، اما هیچ $(n-1)$ نفری نتوانند راز را کشف کنند. برای این منظور کافی است به هر یك از $(n-1)$ نفر اول، عددی تصادفی از $\mathbb{Z}_p$ را بدهیم و به نفر آخر مقدار $s_n = S + s_1 + \ldots + s_{n-1}$ را نسبت بدهیم.

$$\begin{array}{cc} p_1 & s_1 \leftarrow \mathbb{Z}_p \\ p_2 & s_2 \leftarrow \mathbb{Z}_p \\ \vdots & \vdots \\ p_{n-1} & s_{n-1} \leftarrow \mathbb{Z}_p \\ p_n & s_n = S + s_1 + \ldots + s_{n-1} \end{array}$$

چنان که مشخص است هیچ یك از افراد به تنهایی راز را نمی‌دانند و ظاهراً شرایط مساله برآورده شده است؛ چرا که داریم:

[۱۵] Secret Sharing

$$S = -s_1 + \ldots + s_n$$

اما یك مشکل که در این راه حل به چشم می‌آید آن است که ممکن است یکی از افراد نخواهد مقدار s_i خود را به درستی اعلام کند. بدین منظور لازم است مقادیر $Y_i = g^{s_i}$ را نیز در اختیار عموم قرار دهیم تا در صورت بروز چنین مشکلی، فرد متقلب شناسایی گردد.
اما مشکل دیگری نیز وجود دارد. ممکن است یکی از افراد سهیم، اساساً سهم خود از راز را اعلام نکند. در این صورت چه باید کرد؟
مساله‌ای که مطرح شد در واقع بیانگر حالتی خاص از تسهیم راز است (تسهیم راز (n, n)). حالت کلی‌تر مساله، تسهیم راز (t, n) است که در آن کشف راز با همراهی هر زیرمجموعه حداقل t نفری از مجموع n نفر میسر خواهد بود. در ادامه پروتکل تسهیم راز شامیر[۱۶] را معرفی خواهیم نمود که جوابی به همین حالت کلی‌تر مساله است.

پروتکل تسهیم راز شامیر

[۱۶] Shamir