


دانشکده‌ی علوم ریاضی

مقدمه‌ای پیشرفته بر رمزنگاری ۱۹ آذر ۱۳۹۱

جلسه‌ی ۲۰: محاسبه‌ی دو عاملی

مدرس: شهرام خزائی *نگارنده: رحیم رمضانیان*

۱ مقدمه

محاسبه توزیع شده[۱] روشی است که در آن تعدادی از افراد، وسایل و یا گروه‌ها که با هم در ارتباط هستند قصد دارند که یک تابع را به طور مشترک محاسبه کنند. روش محاسبه‌ی چند عاملی امن[۲] نوعی از روش محاسبه توزیع شده است که در آن محاسبه تابع به صورت امن صورت می‌گیرد. در ادامه این جلسه منظور از محاسبه امن توضیح داده می‌شود. در یک محاسبه چند عاملی امن خواص امنیتی باید در حضور عامل‌های بد رفتار[۳] نیز حفظ شود .

مثال ۱ *اداره A که بر روی مباحث عدالت کیفری تحقیق می‌کند درخواستی را به اداره مهاجرت مبنی بر تعداد مهاجران ثبت شده‌ای که در نیویورک مجرم شناخته شده‌اند فرستاده است. اما اداره مهاجرت نباید لیست مهاجران را در اختیار اداره A قرار دهد و اداره A نیز بنا به سیاست امنیتی نمی‌تواند لیست افراد مجرم را در اختیار اداره مهاجرت قرار دهد.*

مثال ۲ *فرآیند انجام یک انتخابات را در نظر بگیرید. از نظر افراد شرکت‌کننده در انتخابات تعدادی از نیازمندی‌ها الزامی است. به عنوان مثال حریم خصوصی بیان می‌دارد که هیچ ائتلافی از عامل‌ها قادر به فهمیدن رای یک عامل خاص نشوند. همچنین صحت انجام انتخابات بیان می‌کند هیچ گروهی از عامل‌ها قادر به تغییر نتیجه رای گیری به جز روش‌های مجاز نباشند.*

۲ محاسبه چند عاملی امن

به منظور بیان و اثبات امن بودن پروتکل‌های محاسبات چند عاملی به تعریف تعدادی از خواص امنیتی می‌پردازیم. در محیط محاسبه حمله‌کننده کنترل تعدادی از عامل‌ها را در دست دارد و قصد دارد به اجرای پروتکل حمله کند.

- حریم خصوصی[۴]: هیچ عاملی نباید بیش از خروجی‌های خودش اطلاع یابد. به عبارت دیگر تنها اطلاعی که در مورد ورودی‌های دیگر عامل‌ها کسب می‌کند باید از خروجی‌های مربوط به خودش استنتاج شود. به عنوان مثال در یک مزایده با مشخص شدن فرد پیروز تنها می‌توان نتیجه گرفت که دیگر پیشنهادات کمتر از پیشنهاد پیروز شده بوده است.

[۱] distributed computing
[۲] secure multi-party computation
[۳] miss-behaving parties
[۴] privacy

- صحت[۵] : به هر عامل تضمین می‌شود که خروجی پروتکل صحیح است به عنوان مثال در یک مزایده کسی که بالاترین پیشنهاد را داده است فرد برنده خواهد بود.
- استقلال ورودی‌ها[۶]: عامل‌های معیوب[۷] که کنترل آن‌ها در اختیار حمله کننده است باید ورودی خود را مستقل از عامل‌های صادق[۸] انتخاب کنند. استقلال ورودی‌ها از حریم خصوصی نتیجه نمی‌شود.
- عادلانه[۹]: گوییم یک پروتکل محاسبه عادلانه است اگر یک عامل خروجی را دریافت کرد همه عامل‌ها نیز خروجی را دریافت کنند.
- ضمانت تحویل خروجی[۱۰]: عامل‌های معیوب نباید قادر به جلوگیری از رسیدن خروجی به عامل‌های صادق باشند. به عنوان مثال عامل‌های معیوب نمی‌توانند حمله منع سرویس‌دهی را اجرا کنند.

۱.۲ تعریف امنیت

یک روش تحلیل امنیت پروتکل لیست کردن تمام ویژگی‌های امنیتی و اثبات برقراری آن‌ها در پروتکل است. اما این روش مناسب نیست زیرا ویژگی‌های امنیتی وابسته به کاربرد هستند و نیاز به تعریف دوباره دارند. تعریفی کلی از امنیت ارائه خواهیم کرد که تمام حملات ممکن را مدل خواهد کرد بدین منظور باید تعریف کنیم که حمله کننده چه قابلیت‌هایی دارد، ساز و کار پروتکل چگونه است و میزان ضمانت امنیتی چه میزان است.

۲.۲ مدل کردن حمله‌کننده

حمله‌کننده را از نظر رفتار، قدرت اجرایی و استراتژی تخریب مدل می‌کنیم.

- از نظر رفتار حمله‌کننده می‌تواند شبه‌صادق[۱۱] (این دسته پروتکل را به درستی اجرا می‌کنند و حمله‌کننده حالت‌های داخلی عامل‌های معیوب را در اختیار دارد و از آن‌ها جهت یادگیری و استنتاج اطلاعات حساس و امن استفاده می‌کند)، بد رفتار[۱۲] (این دسته به طور دلخواه از پروتکل منحرف می‌شوند) و یا پنهان[۱۳] (این دسته رفتاری بین دو دسته قبل دارد و اگر رفتار خرابکارانه داشته باشد آنگاه با احتمالی امکان گیر افتادن آن‌ها وجود دارد) باشد. حمله‌کننده‌های شبه‌صادق را معمولا منفعل[۱۴] و بد رفتار را فعال[۱۵] می نامند.
- از نظر قدرت محاسباتی حمله‌کننده می‌تواند قدرت چند جمله‌ای داشته باشد و یا بدون محدودیت باشد.

---

[۵] correctness
[۶] independence of inputs
[۷] corrupted parties
[۸] honest parties
[۹] fairness
[۱۰] guaranteed output delivery
[۱۱] semi-honest
[۱۲] malicious
[۱۳] covert
[۱۴] passive
[۱۵] active

- از نظر استراتژی تخریب حمله‌کننده می‌تواند ایستا[۱۶] و یا تطابقی[۱۷] باشد. حمله‌کننده ایستا قبل از اجرای پروتکل عامل‌های معیوب را انتخاب می‌کند اما حمله‌کننده تطابقی بعد از دیدن یک سری از پیام‌های پروتکل تصمیم می‌گیرد کدام عامل‌ها را کنترل کند.

۳.۲ ساز و کار پروتکل

پروتکل‌ها می‌توانند به صورت سریال[۱۸] و یا همزمان[۱۹] انجام شوند. معمولا پروتکل‌های تصدیق از پروتکل‌هایی هستند که به صورت همزمان توسط چند عامل اجرا می‌شوند.

۴.۲ آیا محاسبه امن امکان دارد؟

نشان داده شده است که اگر تعداد عامل‌های صادق بیش از نصف عامل‌ها باشد برای هر تابعی که بتوان آن را به صورت کارا محاسبه کرد پروتکلی وجود دارد که می‌تواند ویژگی‌های امنیتی را برآورده کند. اگر اکثر عامل‌ها صادق نباشند تضمینی برای برقراری عدالت وجود ندارد. همچنین ضمانتی برای تحویل خروجی در این حالت وجود ندارد. اما در مورد دیگر خواص امنیتی می‌توان پروتکلی طراحی کرد که آن‌ها را ارضا کند.

۵.۲ نمادگذاری

فرض کنید n شاخص امنیتی باشد. معمولا شاخص امنیتی و طول ورودی از هم مجزا هستند. تابع ϵ را ناچیز نامند اگر برای هر تابع چند جمله ای $p(\cdot)$ عدد طبیعی N وجود داشته باشد که برای هر $n \geq N$ داشته باشیم $\epsilon(n) \leq \frac{1}{p(n)}$. فرض کنید a یک رشته متناهی و n یک عدد طبیعی و $X(a,n)$ یک متغیر تصادفی باشد. به عنوان مثال $X(a,n)$ می‌تواند خروجی اجرای پروتکل با ورودی a و شاخص امنیتی n باشد. یک خانواده[۲۰] از توزیع‌ها $X = \{X(a,n)\}_{n\in\mathbb{N},a\in\{0,1\}^*}$ دنباله‌ای نامتناهی از متغیرهای تصادفی است. دو خانواده X و Y را تمایزناپذیر محاسباتی نامیم (و می‌نویسیم $X \approx^c Y$) هرگاه برای هر تمایزگر چندجمله‌ای غیر یکنواخت $\mathcal{D}$ تابع ناچیز ϵ وجود داشته باشد به طوری که برای هر رشته a و هر n به قدر کافی بزرگ داشته باشیم

$$|\Pr[\mathcal{D}(X(a,n)) = 1] - \Pr[\mathcal{D}(Y(a,n)) = 1]| \leq \epsilon(n)$$

در تعریف بالا اگر $\mathcal{D}$ قدرت نامحدود داشته باشد آنگاه X و Y را از نظر آماری نزدیک نامند.
عاملیت[۲۱]: فرض کنید $f = (f_1, f_2)$ یک تابع دو مولفه‌ای باشد و $x, y \in \{0,1\}^n$. عامل اول ورودی x و عامل دوم ورودی y را دریافت می‌کند. عامل اول قصد دارد $f_1(x,y)$ را و عامل دوم $f_2(x,y)$ را حساب کند. این روش محاسبه را محاسبه دو عاملی[۲۲] نامند.

[۱۶] static
[۱۷] adaptive
[۱۸] serial
[۱۹] concurrent
[۲۰] ensemble
[۲۱] functionality
[۲۲] two party computation

۶.۲ امنیت مبتنی بر شبیه‌سازی

در این قسمت می‌خواهیم امنیت را در حضور حمله‌کننده شبه‌صادق تعریف کنیم. یک شبیه‌ساز با دریافت ورودی‌ها و خروجی‌های عامل‌های معیوب می تواند دید[۲۳] حمله‌کننده از اجرای پروتکل را شبیه‌سازی کند.

تعریف ۳ *(محاسبه‌ی N-عاملی امن) فرض کنید $f = (f_1, \dots, f_N)$ یک عاملیت باشد و π پروتکلی که f را محاسبه می‌کند. گوییم پروتکل π امن است اگر برای هر حمله‌کننده شبه‌صادق $\mathcal{A}$ برای π یک شبیه‌ساز $\mathcal{S}$ وجود داشته باشد که برای هر زیر مجموعه I از عامل‌های معیوب و هر بردار ورودی $x = (x_1, ..., x_N)$ دو توزیع زیر نزدیک باشند.*

۱. *خروجی $\mathcal{A}$ و خروجی همه عامل‌ها بعد از اجرای پروتکل با ورودی x_i برای عامل iام.*

۲. *خروجی شبیه‌ساز $\mathcal{S}$ که $(x_i, f(x_i))$ را برای هر $i \in I$ را به عنوان ورودی می‌گیرد و تمام مقادیر $f_1(x)$، $f_2(x)$، ... و $f_N(x)$.*

در حالت خاص محاسبه‌ی دو عاملی این تعریف به صورت زیر ساده می‌شود.

تعریف ۴ *(محاسبه‌ی دو عاملی امن) فرض کنید $f = (f_1, f_2)$ یک عاملیت باشد می‌گوییم پروتکل π به طور امن f را در برابر حمله‌کننده ایستا و نیمه صادق محاسبه می‌کند اگر الگوریتم‌های PPT، $\mathcal{S}_1$ی و $\mathcal{S}_2$ وجود داشته باشد که دو توزیع زیر تمایزناپذیر باشند.*

$$\{(\mathcal{S}_1(1^n, x, f_1(x,y), f(x,y))\}_{x,y,n} \approx^c \{\mathsf{view}_1^\pi(x,y,n), \mathsf{output}^\pi(x,y,n))\}_{x,y,n}$$

$$\{(\mathcal{S}_2(1^n, y, f_2(x,y), f(x,y))\}_{x,y,n} \approx^c \{\mathsf{view}_2^\pi(x,y,n), \mathsf{output}^\pi(x,y,n))\}_{x,y,n}$$

که در آن view_i پیام هایی است که عامل iام در طول اجرای پروتکل می بیند، output_i خروجی نهایی عامل iام است و

$$\mathsf{output}^\pi(x,y,n)) = (\mathsf{output}_1^\pi(x,n)), \mathsf{output}_2^\pi(y,n)))$$

بنابه تعریف بالا دید یک عامل می‌تواند توسط یک الگوریتم PPT که تنها به ورودی‌ها و خروجی‌های عامل دسترسی دارد تولید شود. بنابراین اگر حمله‌کننده شبه‌صادق بتواند بعد از اجرای پروتکل محاسبه‌ای انجام دهد می‌تواند آن محاسبه را تنها با توجه به ورودی‌ها و خروجی‌ها انجام دهد. نکته قابل توجه در تعریف امنیت محاسبه دو عاملی تمایزناپذیری دو توزیع توام است. ممکن است اگر توزیع‌های بالا توام نباشند از نظر محاسباتی قابل تمایز باشند. برای مثال فرض کنید عاملیت f به گونه‌ای باشد که عامل اول یک بیت تصادفی را در خروجی قرار میدهد و عامل دوم هیچ عملی انجام نمی‌دهد و همچنین از خروجی عامل اول اطلاعی نمی‌یابد. فرض کنید پروتکل چنین باشد که عامل اول یک بیت تصادفی تولید می‌کند و آن را به خروجی می‌دهد و بعلاوه آن را برای عامل دوم می‌فرستد. در مقابل عامل دوم بیت دریافتی را دور می‌اندازد. در این صورت $\{\mathcal{S}_2(1^n, y, f_2(x,y)\}_{x,y,n}$ از $\{\mathsf{view}_2^\pi(x,y,n)\}_{x,y,n}$ تمایزپذیر است در صورتی که این پروتکل به وضوح امن است.

[۲۳] view