به نام خدا

دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری ۲۰ فروردین ۱۳۹۲

جلسه‌ی ۱۲: توابع شبه‌تصادفی و امنیت چندپیامی

مدرس: دکتر شهرام خزائی *نگارنده: احمد فنایی شیخ الاسلامی*

۱ امنیت چندپیامی

تعریف ۱ ***آزمایش امنیت چندپیامی***[۱] $\mathsf{PrivK}^{\mathrm{mult}}_{\Pi,\mathcal{A}}(n)$

۱. *چالشگر کلید k را تولید می‌کند:* $k \leftarrow \mathsf{Gen}(1^n)$

۲. *مهاجم $\mathcal{A}$، دو دسته پیام به صورت* $(m^1_0, ..., m^{p(n)}_0)$ *و* $(m^1_1, ..., m^{p(n)}_1)$ *انتخاب می‌کند که* $|m^i_0| = |m^i_1|$ *و به چالشگر می‌دهد.*

۳. *چالشگر یک بیت تصادفی* $b \in \{0, 1\}$ *انتخاب می‌کند.*

۴. *چالشگر مقادیر* $c_i = \mathsf{Enc}_k(m^i_b)$ *را محاسبه و به مهاجم $\mathcal{A}$ می‌دهد.*

۵. *مهاجم $\mathcal{A}$ بیت $\hat{b}$ را باز می‌گرداند.*

خروجی آزمایش که با $\mathsf{PrivK}^{\mathrm{mult}}_{\Pi,\mathcal{A}}(n)$ *نشان داده می‌شود، برابر یک در نظر گرفته می‌شود اگر $b = \hat{b}$ و صفر است اگر $b \neq \hat{b}$. دقت کنید که خروجی آزمایش یک متغیر تصادفی است که به تمام بیت‌های تصادفی آزمایش و الگوریتم‌ها، بستگی دارد.*

تعریف ۲ *سیستم رمز متقارن* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *دارای امنیت چندپیامی در حضور مهاجم شنودگر است، اگر برای هر مهاجم غیر یکنواخت تصادفی چندجمله‌ای- زمان[۲] مانند $\mathcal{A}$، تابع ناچیز $\varepsilon(n)$ وجود داشته باشد که:*

$$\Pr\{\mathsf{PrivK}^{\mathrm{mult}}_{\Pi,\mathcal{A}}(n) = 1\} \leq \frac{1}{2} + \varepsilon(n)$$

قضیه ۱ *اگر الگوریتم رمزنگاری تصادفی نباشد، امنیت چند پیامی برقرار نیست.*

[۱] multi-message security expriment

[۲] non-uniform problistic polynomial time

برهان. مهاجم $\mathcal{A}$ را به گونه‌ای می‌سازیم که در آزمایش $\mathsf{PrivK}^{\mathrm{mult}}_{\Pi,\mathcal{A}}(n)$ با احتمالی قابل توجه پیروز شود. مهاجم با دریافت ورودی 1^n دو دسته پیام $(m^1_0, m^2_0) = (0^n, 0^n)$ و $(m^1_1, m^2_1) = (0^n, 1^n)$ را انتخاب می‌کند و آن‌ها را برای چالشگر می‌فرستد و متن‌های رمزشده (c_1, c_2) را دریافت می‌کند. با توجه به این‌که این سیستم رمزنگاری قطعی است، اگر دسته اول انتخاب شده باشد، همواره $c_1 = c_2$ و اگر دسته دوم انتخاب شده باشد $c_1 \neq c_2$. بنابراین مهاجم بیت $\hat{b}$ را برابر یک قرار می‌دهد اگر و فقط اگر $c_1 = c_2$ باشد. به وضوح مهاجم با احتمال یک در آزمایش بالا پیروز می‌شود. ■

۲ توابع تصادفی و شبه‌تصادفی

۱.۲ توابع تصادفی

مجموعه‌ی همه توابعی را که n بیت را به n بیت می‌نگارند، با RF_n نشان دهید. تعداد کل چنین توابعی برابر است با 2^{n2^n} زیرا، تعداد ورودی‌های تابع 2^n تاست و برای هر ورودی 2^n امکان وجود دارد. منظور از یک تابع تصادفی تابعی است مانند $f : \{0,1\}^n \to \{0,1\}^n$ که از مجموعه RF_n به تصادف و با توزیع یکنواخت انتخاب شده است. برای ذخیره یک تابع تصادفی به $n2^n$ بیت نیاز است که حتی برای مقادیر کوچک n نیز غیر عملی است.

۲.۲ توابع شبه‌تصادفی

برای حل مشکل ذخیره‌سازی توابع تصادفی، سعی می‌کنیم زیرمجموعه کوچکتری از همه‌ی 2^{n2^n} تابع انتخاب کنیم. مثلا اگر یک خانواده از توابع که شامل 2^n تابع است را در نظر بگیریم، می‌توان هر عضو آنرا با $n-$بیت ذخیره کرد. واضح است که وقتی یک عضو از این خانواده 2^n عضوی از به تصادف انتخاب شود دیگر کاملاً تصادفی نیست. ما در این بخش می‌خواهیم مفهوم تابع شبه‌تصادفی را برای یک خانواده از توابع تعریف کنیم.

تعریف ۳ (خانواده توابع) *مجموعه $\mathcal{F} = \{f_k : \{0,1\}^{|k|} \to \{0,1\}^{|k|}\}_{k\in\{0,1\}^*}$ را یک خانواده از توابع می‌نامیم.*

دقت کنید که به ازای هر کلید $n-$بیتی مانند k تابع $f_k(\cdot)$ تابعی است که $n-$بیت را به $n-$بیت می‌برد. حال سوال این است که چگونه می‌توان مفهوم شبه‌تصافی بودن را برای یک خانواده از توابع تعریف کرد؟
برای جواب دادن به سوال بالا می‌توان از آزمایش زیر که بین یک چالشگر فرضی و مهاجم $\mathcal{A}$ انجام می‌شود استفاده کرد:

تعریف ۴ ***آزمایش*** $\mathsf{ExpPRF}_{\mathcal{F},\mathcal{A}}(n)$

۱. *چالشگر بیت تصادفی b را تولید می‌کند:* $b \leftarrow \{0,1\}$

۲. *چالشگر با استفاده از بیتی که در مرحله ی قبل تولید کرده تابع f را به صورت زیر انتخاب می‌کند.*
اگر $b = 0$، تابع f از بین تمام توابع ممکن به تصادف انتخاب می‌شود: $f \leftarrow RF_n$
اگر $b = 1$، تابع f به تصادف از مجموعه $\{f_k \mid |k| = n\}$ انتخاب می‌شود؛ یعنی چالشگر ابتدا کلید کاملاً تصادفی $n-$بیتی k را تولید می‌کند و سپس قرار می‌دهد: $f \leftarrow f_k$.

۳. مهاجم $\mathcal{A}$ با دریافت ورودی 1^n می‌تواند به تعداد چندجمله‌ای بار پرسمان[۳]های دلخواه خود را به صورت تطبیقی[۴] انتخاب و برای چالشگر ارسال کند. چالشگر هم تابع f را روی همه پرسمان‌ها اعمال و پاسخ را بی‌درنگ برای مهاجم ارسال می‌کند (اصطلاحاً گفته می‌شود مهاجم به تابع f دسترسی اُراکلی[۵] می‌یابد).

۴. مهاجم بیت $\hat{b}$ را تولید می‌کند.

خروجی آزمایش که با $\mathsf{ExpPRF}_{\mathcal{F},\mathcal{A}}(n)$ نشان داده می‌شود برابر یک در نظر گرفته می‌شود اگر $\hat{b} = b$ و صفر است اگر $\hat{b} \neq b$. دقت کنید که خروجی آزمایش یک متغیر تصادفی است که به تمام بیت‌های تصادفی آزمایش و الگوریتم‌ها، بستگی دارد.

تعریف ۵ (تابع شبه‌تصادفی) فرض کنید $\mathcal{F} = \{f_k : \{0,1\}^{|k|} \to \{0,1\}^{|k|}\}_{k\in\{0,1\}^*}$ یک خانواده از توابع باشد که بتوان $f_k(x)$ را از روی k و x در زمان چندجمله‌ای محاسبه کرد. می‌گوییم $\mathcal{F}$ شبه‌تصادفی است اگر برای هر مهاجم غیر یکنواخت تصادفی چندجمله‌ای- زمان مانند $\mathcal{A}$، تابع ناچیز $\varepsilon(n)$ وجود داشته باشد که:

$$\Pr\{\mathsf{ExpPRF}_{\mathcal{F},\mathcal{A}}(n) = 1\} \leq \frac{1}{2} + \varepsilon(n)$$

دسترسی اوراکلی یک الگوریتم مانند $\mathcal{A}$ به تابعی چون f را به صورت $\mathcal{A}^{f(\cdot)}$ نشان می‌دهند. چنین الگوریتمی یک ماشین اوراکلی نامیده می‌شود. می‌توان چنین تصور نمود که ماشین اوراکلی $\mathcal{A}$ به جعبه‌ای که اصطلاحاً جعبه‌سیاه[۶] نامیده می‌شود دسترسی دارد. جعبه سیاه یک نوار ورودی و یک نوار خروجی دارد به طوری‌که به محض اینکه ماشین $\mathcal{A}$ پرسمان خود را بر روی نوار ورودی قرار می‌دهد، خروجی بر روی نوار خروجی ظاهر می‌شود و ماشین $\mathcal{A}$ می‌تواند از آن استفاده کند. در محاسبه‌ی زمان اجرای الگوریتم (ماشین) اوراکلی $\mathcal{A}$، هر پرسمان و دریافت پاسخ آن یک واحد زمانی درنظر گرفته می‌شود. بنابراین یک الگوریتم اوراکلی چندجمله‌ای، فقط به تعداد چندجمله‌ای بار (بر حسب طول ورودی) می‌تواند به اوراکل (یا اوراکل‌های) خود دسترسی پیدا کند. تعریف زیر که با استفاده از ماشین (مهاجم) اوراکلی ارائه شده است معادل تعریف بالا از شبه‌تصادفی بودن یک خانواده از توابع است.

تعریف ۶ (تابع شبه‌تصادفی) یک خانواده از توابع $\{f_k : \{0,1\}^{|k|} \to \{0,1\}^{|k|}\}_{k\in\{0,1\}^*}$ شبه‌تصادفی است، هرگاه:

۱. $f_k(x)$ را بتوان در زمان چندجمله‌ای از روی k و x محاسبه کرد.

۲. برای هر مهاجم تصادفی $\mathcal{A}$ که در زمان چند جمله ای اجرا شود، تابع ناچیز $\varepsilon(n)$ وجود داشته باشد که:
$$|\Pr\{f \leftarrow RF_n : \mathcal{A}^{f(\cdot)}(1^n) = 1\} - \Pr\{k \leftarrow \{0,1\}^n : \mathcal{A}^{f_k(\cdot)}(1^n) = 1\}| \leq \varepsilon(n)$$

مثال ۷ تابع $f_k(x) = k \oplus x$ شبه‌تصادفی نیست. برای نشان دادن این موضوع یک مهاجم $\mathcal{A}$ می‌سازیم که خانواده فوق را از خانواده توابع تصادفی تشخیص دهد. مهاجم با دریافت ورودی 1^n دو پرسمان $x_1 = 0^n$ و $x_2 = 1^n$ را انتخاب می‌کند و آن‌ها را برای اوراکل می‌فرستد و پاسخ‌های y_1 و y_2 را دریافت می‌کند. اگر $y_1 \oplus y_2 = 1^n$ باشد، مهاجم خروجی ۱ را برمی‌گرداند و در غیر این صورت خروجی ۰ را برمی‌گرداند. این مهاجم دارای مزیت غیرناچیز $|2^{-n} - 1|$ است، زیرا:

---

[۳] query
[۴] addaptive
[۵] oracle access
[۶] black box

$$\Pr\{f \leftarrow RF_n : \mathcal{A}^{f(\cdot)}(1^n) = 1\} = \Pr\{f \leftarrow RF_n \mid f(0^n) \oplus f(1^n) = 1^n\} = 2^{-n}$$

$$\Pr\{k \leftarrow \{0,1\}^n : \mathcal{A}^{f_k(\cdot)}(1^n) = 1\} = \Pr\{k \leftarrow \{0,1\}^n : f_k(0^n) \oplus f_k(1^n) = 1^n\} = 1$$

نکته ۱ *برای سادگی تعریف تابع شبه‌تصادفی برای حالتی که طول ورودی، طول کلید و طول خروجی همگی برابر هستند ارائه شد. مفهوم تابع شبه‌تصادفی را می‌توان به توابعی که طول ورودی، طول کلید و طول خروجی آنها متفاوت است تعمیم داد.*

اگر $G : \{0,1\}^n \to \{0,1\}^n$ یک مواد شبه‌تصادفی باشد، تابع $f_k(x) = G(k||x)$ که n بیت را به $2n$ بیت می‌نگارد، لزوماً یک تابع شبه‌تصادفی نیست. اما روش پیچیده‌تری برای ساخت یک تابع شبه‌تصادفی با استفاده از یک مولد شبه‌تصادفی وجود دارد.

قضیه ۲ *مولد شبه‌تصادفی وجود دارد اگر وتنها اگر تابع شبه‌تصادفی وجود داشته باشد.*

قضیه ۳ *فرض کنید $\{f_k : \{0,1\}^{|k|} \to \{0,1\}^{|k|}\}_{k\in\{0,1\}^*}$ یک خانواده از توابع شبه تصادفی باشد. سیستم $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ روی فضای پیام $\mathcal{M} = \{0,1\}^n$ که به صورت زیر تعریف می‌شود، دارای امنیت چندپیامی است.*

- *الگوریتم تولید کلید Gen: ورودی 1^n را می‌گیرد و کلید تصادفی $k \leftarrow \{0,1\}^n$ را تولید می‌کند.*
- *الگوریتم رمزنگاری Enc: پیام $m \in \{0,1\}^n$ و کلید $k \in \{0,1\}^n$ را می‌گیرد و متن رمزی c را به این صورت تولید می‌کند که ابتدا رشته تصادفی $r \leftarrow \{0,1\}^n$ انتخاب و سپس $c = \langle r, m \oplus f_k(r)\rangle$ محاسبه می‌شود.*
- *الگوریتم رمزگشایی Dec: متن رمزی $c \in \{0,1\}^*$ و کلید $k \in \{0,1\}^n$ را دریافت می‌کند؛ سپس اگر $c = \langle r, s\rangle$ که $r, s \in \{0,1\}^n$ متن اصلی $m = s \oplus f_k(r)$ برگردانده می‌شود. در غیر این صورت (اگر $|c| \neq 2n$) نماد $\perp$ به معنای نامعتبر بودن متن رمزی برگردانده می‌شود.*

شهود اثبات. ابتدا ببینیم یک مهاجم غیرچندجمله‌ای چگونه می‌تواند به سیستم حمله کند. با استفاده از تناقض روز تولد یک مهاجم غیر چندجمله‌ای با مزیت قابل توجه برای آزمایش امنیت چندپیامی می‌سازیم. هریک از دسته‌های متن اصلی $(m_0^1, \cdots, m_0^{p(n)})$ و $(m_1^1, \cdots, m_1^{p(n)})$ که مهاجم انتخاب می‌کند شامل $p(n) = 2^{n/2}$ متن است. مهاجم هر یک از $2p(n)$ متن m_j^i را که $j \in \{0,1\}$ و $i = 1, \cdots, p(n)$ به طور کاملاً تصادفی از مجموعه‌ی $\{0^n, 1^n\}$ انتخاب می‌کند. سپس متن‌های رمزی $c_1, \cdots, c_{p(n)}$ را که $c_i = \langle r_i, s_i\rangle$ و $r_i, s_i \in \{0,1\}^n$ دریافت می‌کند. دقت کنید که اگر i و j متمایزی وجود داشته باشند که

$$r_i = r_j, \;\; m_0^i \oplus m_0^j \neq m_1^i \oplus m_1^j$$

مهاجم می‌تواند تشخیص دهد که c_i ها رمزشده‌ی کدام دسته هستند؛ زیرا

$$s_i \oplus s_j = m_b^i \oplus f_k(r_i) \oplus m_b^j \oplus f_k(r_j) = m_b^i \oplus m_b^j$$

در این حالت مهاجم بیت b را می‌تواند به درستی حدس بزند و بیت $\hat{b}$ را متناظراً تولید کند. در غیر این‌صورت بیت $\hat{b}$ به طور کاملاً تصادفی تولید می‌شود. حال احتمال موفقیت مهاجم را محاسبه می‌کنیم. توجه کنید که با احتمال $\frac{۱}{۲}$ پیشامد $m_۰^i \oplus m_۰^j \neq m_۱^i \oplus m_۱^j$ (برای هر i, j دخواه متمایز) اتفاق می‌افتد و طبق قضیه روز تولد با احتمال $\frac{۱}{۲}$ مقادیر متمایز i, j وجود دارد که $r_i = r_j$. بنابراین در حالت اول که با احتمال $\frac{۱}{۴}$ اتفاق می‌افتد مهاجم با احتمال یک موفق می‌شود و در حالت دوم که با احتمال $\frac{۳}{۴}$ اتفاق می‌افتد مهاجم با احتمال $\frac{۱}{۲}$ موفق می‌شود. بنابراین احتمال موفقیت مهاجم برابر است با

$$\frac{۱}{۴} \cdot ۱ + \frac{۳}{۴} \cdot \frac{۱}{۲} = \frac{۵}{۸}$$

که مزیت قابل توجهی برای وی به همراه دارد.

به صورت شهودی مهاجم چندجمله‌ای که $p(n)$ برای وی یک چندجمله‌ای است، تنها زمانی در آزمایش امنیت چند پیامی موفق می‌شود که خوش شانس باشد و دو پیام با مقادیر تصادفی یکسان رمز شوند. احتمال این اتفاق از $\frac{\binom{p(n)}{۲}}{۲^n}$ کمتر است که برای توابع چندجمله‌ای $p(\cdot)$ ناچیز است.