دانشکده‌ی علوم ریاضی

مقدمه‌ای پیشرفته بر رمزنگاری ۴ مهر ۱۳۹۱

جلسه‌ی ۴: سناریوهای حمله، تابع‌یک‌طرفه

مدرس: شهرام خزائی *نگارنده: محمد قیاسی*

۱ نمادگذاری

PPT[۱]: مجموعه‌ی الگوریتم‌های تصادفی چندجمله‌ای
KPA[۲] : حمله‌ی متن اصلی معلوم
CPA[۳]: حمله‌ی متن اصلی منتخب
CCA[۴]: حمله‌ی متن رمز شده منتخب

۲ سناریوهای حمله

سناریوهای مختلفی را می توان برای حمله به یک سیستم رمز تصور کرد که در زیر چند تا از آنها را آورده ایم:

- KPA: در این نوع حمله دشمن از چند زوج مرتب از متن‌های اصلی[۵] و متن‌های رمز شده[۶] متناظرشان با خبر است (البته با این فرض که همه‌ی اینها با یک کلید ثابت رمز شده اند) و حالا تلاش می‌کند که درباره‌ی یک متن رمز شده‌ی جدید که تحت همان کلید رمز شده است اطلاعات کسب کند.
- CPA: در این حمله دشمن این توانایی را دارد که متن رمز شده‌ی هر متن اصلی دلخواهش را بدست بیاورد و حالا تلاش می‌کند که درباره‌ی یک متن رمز شده‌ی جدید که تحت همان کلید رمز شده است اطلاعات نابدیهی کسب کند.
- CCA: در این مورد دیگر دشمن نه تنها می‌تواند متن رمز شده‌ی متناظر با متن‌های اصلی دلخواهش را بدست آورد، بلکه می‌تواند متنهای اصلی مربوط به متن‌های رمز شده‌ی دلخواهش را نیز بدست آورد یا به عبارت دیگر می‌تواند متن رمز شده بدهد و متن اصلی نظیرش را بگیرد و حالا هدف او این است که درباره‌ی متن اصلی مربوط به یک متن رمز شده اطلاعات بدست آورد. (البته قابل توجه است که او نمی‌تواند به صورت مستقیم عبارت رمز شده را بدهد و ورودی اولیه اش را بگیرد.)

[۱] probabilistic polynomial time
[۲] known-plaintext attack
[۳] chozen-plaintext attack
[۴] chozen-ciphertext attack
[۵] plaintext
[۶] ciphertext

۳ توابع یک‌طرفه

تعریف ۱ *تابع*

$$f : \{0,1\}^* \longrightarrow \{0,1\}^*$$

یک تابع یک‌طرفه‌ی در بدترین حالت[۷] *نامیده می‌شود هرگاه:*

- *(محاسبه‌ی آن آسان باشد*[۸]*) محاسبه‌ی $f(x)$ برای هر x در دامنه‌ی تابع از روی x آسان باشد. یا به عبارت دیگر یک الگوریتم PPT وجود داشته باشد که برای هر x در دامنه‌ی تابع بتواند $f(x)$ را محاسبه کند.*
- *(معکوس کردن آن دشوار باشد*[۹]*) هیچ الگوریتم مهاجم PPT مثل $\mathcal{A}$ وجود نداشته باشد که:*

$$\forall x, \Pr\{\mathcal{A}(f(x)) \in f^{-1}(f(x))\} = 1$$

می‌توان نشان داد که با فرض اینکه کلاس مسائل NP در کلاس مسائلBPP نیست، توابع یک‌طرفه با تعریف بالا می‌بایست وجود داشته باشند. در حقیقت، این دو فرض با هم معادل اند.

تعریف ۲ *(تابع ناچیز) یک تابع $\varepsilon(n)$ ناچیز*[۱۰] *خوانده می‌شود اگر برای هر $c > 0$ ای، یک n_0 ای وجود داشته باشد به طوری که برای هر $n > n_0$ ای*

$$\varepsilon(n) < \frac{1}{n^c}$$

به عبارت دیگر، یک تابع ناچیز به صورت مجانبی کمتر از معکوس هر چند جمله ای است. همچنین، یک تابع $t(n)$ را غیر ناچیز (یا قابل توجه)[۱۱] *می‌گوییم اگر یک ثابت c وجود داشته باشد که برای نامتناهی تا عدد طبیعی n داشته باشیم : $t(n) > n^c$*

حال یک تعریف موجه‌تری برای صورتی از تابع یک‌طرفه ارائه می‌دهیم که برای کاردهای رمزنگاری مناسب است.

تعریف ۳ *(تابع یک‌طرفه‌ی قوی*[۱۲]*) تابع*

$$f : \{0,1\}^* \longrightarrow \{0,1\}^*$$

یک تابع یک‌طرفه‌ی قوی نامیده می‌شود اگر شرایط زیر را برقرار کند:

- *(محاسبه‌ی آن آسان باشد) محاسبه‌ی $f(x)$ برای هر x در دامنه‌ی تابع از روی x آسان باشد. یا به عبارت دیگر یک الگوریتم PPT وجود داشته باشد که برای هر x در دامنه‌ی تابع بتواند $f(x)$ را محاسبه کند.*

---

[۷] worst-case one-way function
[۸] easy to compute
[۹] hard to invert
[۱۰] negligible
[۱۱] non-negligible
[۱۲] strong one-way function

- *(معکوس کردن آن دشوار باشد) برای هر الگوریتم مهاجم* PPT *مانند* $\mathcal{A}$ *یک تابع ناچیز* $\varepsilon(n)$ *وجود دارد به طوری که برای هر عدد طبیعی* n *ای داشته باشیم:*

$$\Pr\{x \longleftarrow \{0,1\}^n; y \longleftarrow f(x) : f(\mathcal{A}(1^n, y)) = y\} \leq \varepsilon(n)$$

توجه کنید که الگوریتم $\mathcal{A}$ *ورودی* 1^n *را می‌گیرد و این برای این است که* $\mathcal{A}$ *اجازه دارد که در زمان چند جمله ای بر حسب طول ورودی کار کند. برای بهتر واضح شدن این مفهوم تابع* $f(x) = \frac{1}{|x|}$ *را در نظر بگیرید. این تابع بدون در نظر گرفتن* 1^n *در تعریف تابع یک‌طرفه محسوب می‌شود.*

به عنوان مثال می‌توان بررسی کرد که تابع $f(x, y)$ که به این صورت تعریف می‌شود که اگر یکی از x یا y یک بود این تابع یک می‌دهد و در غیر این صورت حاصل ضرب x و y را محاسبه می‌کند یک تابع یک‌طرفه‌ی قوی نیست. برای این کار کافی است که یک الگوریتم را در نظر بگیریم که در حالتی که خروجی تابع زوج است مقادیر ۲ و نصف خروجی را به عنوان برگردان تابع می‌دهد و در حالتی هم که خروجی فرد است هر دو را یک می‌دهد. این الگوریتم با احتمال $\frac{3}{4}$ درست کار می‌کند و در نتیجه احتمال موفقیتش ناچیز نیست و این به معنای این است که این تابع یک‌طرفه‌ی قوی نیست. برای در نظر گرفتن چنین توابعی به عنوان تابع یک‌طرفه تعریف ضعیف‌تر زیر را ارائه می‌دهیم.

تعریف ۴ *(تابع یک‌طرفه‌ی ضعیف*[۱۳]*) تابع*

$$f : \{0,1\}^* \longrightarrow \{0,1\}^*$$

یک تابع یک‌طرفه‌ی ضعیف نامیده می‌شود اگر شرایط زیر را برقرار کند:

- *(محاسبه‌ی آن آسان باشد) محاسبه‌ی* $f(x)$ *برای هر* x *در دامنه‌ی تابع از روی* x *آسان باشد. یا به عبارت دیگر یک الگوریتم* PPT *وجود داشته باشد که برای هر* x *در دامنه‌ی تابع بتواند* $f(x)$ *را محاسبه کند.*
- *(معکوس کردن آن دشوار باشد) یک تابع چندجمله ای* $q(n)$ *وجود داشته باشد که برای یک مقدار* n_0 *ای داشته باشیم:*

$$\forall \mathcal{A} \in \mathsf{PPT}, \forall n \geq n_0, \Pr\{x \longleftarrow \{0,1\}^n; y \longleftarrow f(x) : f(\mathcal{A}(1^n, y)) = y\} \leq 1 - \frac{1}{q(n)}$$

یا به عبارتی دیگر

$$\forall \mathcal{A} \in \mathsf{PPT}, \forall n \geq n_0, \Pr\{x \longleftarrow \{0,1\}^n; y \longleftarrow f(x) : f(\mathcal{A}(1^n, y)) \neq y\} > \frac{1}{q(n)}$$

حال که ما دو تعریف قوی و ضعیف را برای توابع یک‌طرفه ارائه دادیم، یک سوال جالب این است که آیا می‌توان از یک تابع یک‌طرفه‌ی ضعیف یک تابع یک‌طرفه‌ی قوی ساخت؟ جواب این سوال مثبت است. در قضیه زیر این موضوع را واضح‌تر بررسی می‌کنیم.

[۱۳] weak one-way function

قضیه ۱ *برای هر تابع یک‌طرفه‌ی ضعیف* $f : \{۰,۱\}^* \longrightarrow \{۰,۱\}^*$ *یک چندجمله ای* $m(\cdot)$ *وجود دارد به گونه ای که تابع*

$$f'(x_۱, x_۲, ..., x_{m(n)}) = (f(x_۱), f(x_۲), ..., f(x_{m(n)}))$$

که $f' : (\{۰,۱\}^n)^{m(n)} \longrightarrow (\{۰,۱\}^n)^{m(n)}$ *یک تابع یک‌طرفه‌ی قوی است.*

ایده‌ی اثبات این قضیه شبیه ایده‌ی کاهش[۱۴] در نظریه‌ی پیچیدگی است. در واقع این ایده را ما در مسائل بسیاری که از این دست هستند به کار می‌بریم. به این صورت که فرض می‌کنیم که یک الگوریتم $\mathcal{A}$ ای وجود دارد که یک‌طرفه‌ی قوی بودن تابع جدید ما را تحدید می‌کند، سپس با استفاده از آن یک الگوریتم ثانویه‌ی $\mathcal{A}'$ ای می‌سازیم که یک‌طرفه‌ی ضعیف بودن تابع اصلی ما را تحدید می‌کند. به عبارت دیگر اثبات می‌کنیم که اگر راهی برای حمله به تابع یک‌طرفه‌ی قوی ای که ساخته ایم وجود داشته باشد، قطعا راهی هم برای حمله به تابع یک‌طرفه‌ی ضعیفی که داشتیم هم وجود دارد.

[۱۴] reduction