مقدمه‌ای پیشرفته بر رمزنگاری ۱۶ آذر ۱۳۹۱

جلسه‌ی ۱۵: امنیت چندپیامی و مقدمه ای بر رمزنگاری نامتقارن

مدرس: شهرام خزائی *نگارنده: سهند جاوید*

۱ امنیت چندپیامی

یادآوری ۱ *سیستم رمز یکبار مصرف*[۱] *یک سه‌تایی* $\mathrm{OTP} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *روی فضای پیام* $\mathcal{M} = \{0,1\}^n$ *است که در آن:*

$$\mathsf{Gen}() : k \leftarrow \{0,1\}^n$$
$$\mathsf{Enc}_k(m) = m \oplus k$$
$$\mathsf{Dec}_k(c) = c \oplus k$$

به یاد داریم که OTP امنیت تک‌پیامی[۲] دارد، یعنی هر مهاجم با قدرت چندجمله ای نمی تواند توزیع هر دو پیام رمز شده را با احتمالی غیر قابل اغماض تمیز دهد. ما پا را از این حد فراتر گذاشتیم و نشان دادیم که OTP، در مقابل هر مهاجم ، با قدرت محاسباتی نامحدود ، امکان موفقیت بیشتر از $\frac{1}{2}$ در تمیز دو توزیع رمز پیام، ندارد.[۳] این امنیت تا زمانی برقرار است که از کلید رمزمان برای رمز کردن بیش از یک پیام استفاده نکنیم. در بازی زیر نشان می‌دهیم که این امنیت زمانی که از کلیدمان برای رمز کردن بیش از یک پیام استفاده کنیم، فرو می‌ریزد :

قضیه ۱ *سیستم رمز* OTP *دارای امنیت چندپیامی*[۴] *نیست.*

برهان. مهاجم $\mathcal{A}$ دو دسته پیام $\bar{m}_0 = (m_0^0, m_0^1) = (0^n, 0^n)$ و $\bar{m}_1 = (m_1^0, m_1^1) = (0^n, 1^n)$ را برای چالشگر[۵] می‌فرستد و در مقابل چالشگر به تصادف بیت b را از $\{0,1\}$ انتخاب می‌کند و $c = \mathsf{Enc}(\bar{m}_b)$ را به مهاجم بر می‌گرداند. مهاجم $\mathcal{A}$ با دریافت $c = c_1 c_2$ به صورت زیر عمل می‌کند:

- اگر $c_1 = c_2$ بود، مقدار $b' = 0$ را بر می‌گرداند.
- اگر $c_1 \neq c_2$ بود، مقدار $b' = 1$ را بر می‌گرداند.

[۱] one-time pad
[۲] single-message secure
[۳] perfect secrecy
[۴] multi-message secure
[۵] challenger

■

با کمی دقت متوجه می‌شویم، هیچ سیستم رمز قطعی[۶] دارای امنیت چندپیامی نیست.

قضیه ۲ *یک سیستم رمز با امنیت چندپیامی، نمی تواند قطعی باشد.*

برهان. بازی قبل را به این صورت در نظر بگیرید که مهاجم دو دسته پیام $\bar{m}_0 = (m_0, m_0)$ و $\bar{m}_1 = (m_0, m_1)$ را تولید می‌کند که $m_0 \neq m_1$ و بر روی مقدار $c = c_1 c_2$ به مانند قبل تصمیم می‌گیرد.

■

این موضوع اهمیت ساخت الگوریتم‌های رمز کردن احتمالاتی[۷] را برای ما روشن می‌سازد. برای این کار، از توابع شبه تصادفی[۸] استفاده می‌کنیم.

یادآوری ۲ *خانواده‌ای از توابع* $\{f_k : \{0,1\}^{|k|} \to \{0,1\}^{|k|}\}_{k \in \{0,1\}^*}$ *را شبه تصادفی گوییم اگر:*

۱. *(محاسبه آن آسان باشد) یک الگوریتم* PPT *وجود داشته باشد که بتواند مقدار* $f_k(x)$ *را با گرفتن ورودی‌های* k *و* x *محاسبه کند.*

۲. *(شبه تصادفی باشد) دو گردایه*[۹] $\{k \leftarrow \{0,1\}^n : f_k\}_n$ *و* $\{F \leftarrow RF_n : F\}_n$ *از دید یک مهاجم* nuPPT *غیر قابل تمایز باشند.*

$$\{k \leftarrow \{0,1\}^n : f_k\}_n \approx \{F \leftarrow RF_n : F\}_n$$

در زیر، سیستم رمزی ارائه می‌کنیم که دارای امنیت چندپیامی است:

تعریف ۱ *فرض کنید* $m \in \{0,1\}^n$ *باشد و* $\{f_k\}$ *یک خانواده از توابع شبه تصادفی باشد. تعریف می‌کنیم:*

- $\mathsf{Gen}(1^n)$*: از میان تمامی رشته‌های به طول* n *یکی را به تصادف به عنوان کلید انتخاب کن* $(k \leftarrow U_n)$
- $\mathsf{Enc}_k(m)$*: رشته* r *را از میان تمامی رشته‌های به طول* n *به تصادف انتخاب کن* $(r \leftarrow U_n)$ *و مقدار* $(r, m \oplus f_k(r))$ *را برگردان.*
- $\mathsf{Dec}_k((r,c))$*: با دریافت ورودی های* k *و* (r,c) *مقدار* $c \oplus f_k(r)$ *را برگردان.*

دقت کنید که در هر بار فراخوانی تابع $\mathsf{Enc}_k(\cdot)$، یک r تصادفی انتخاب می‌شود، که این موجب می‌شود برای هر پیام یکسان m_0 ، در هر بار فراخوانی $\mathsf{Enc}_k(m_0)$ ، خروجی های متفاوتی داشته باشیم. در نتیجه تابع $\mathsf{Enc}(\cdot)$، یک تابع احتمالاتی است.

---

[۶] deterministic encryption scheme
[۷] probabilistic
[۸] pseudorandom functions
[۹] ensemble

قضیه ۳ *سیستم رمز* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *ارائه شده در تعریف ۱ ، دارای امنیت چندپیامی است.*

برهان. فرض خلف کنید که یک تمایزگر nuPPT (که nuPPT است) و مقدار چند جمله‌ای $p(n)$ وجود دارد که برای دو دسته پیام $\bar{m} = (m_0, m_1, ..., m_{q(n)})$ و $\bar{m}' = (m'_0, m'_1, ..., m'_{q(n)})$ و هر مقدار n می‌تواند $\bar{m}$ و $\bar{m}'$ را با احتمال $\frac{1}{p(n)}$ از هم تمیز دهد. توزیع‌های هایبرید H_i را به مانند زیر در نظر بگیرید:

- H_1: توزیع رشته‌های رمز شده $m_0, m_1, ..., m_{q(n)}$ به وسیله سیستم رمز ارائه شده در تعریف ۱:

$$K \leftarrow \{0,1\}^n$$

$$r_0, r_1, ..., r_{q(n)} \leftarrow \{0,1\}^n$$

$$(r_0, m_0 \oplus f_k(r_0)), (r_1, m_1 \oplus f_k(r_1)), ..., (r_{q(n)}, m_{q(n)} \oplus f_k(r_{q(n)}))$$

این دقیقا همان چیزی است که مهاجم با دریافت عبارت رمز شده $m_0, m_1, ..., m_{q(n)}$ ، می‌بیند.

- H_2: تابع f را با تابع واقعا تصادفی F جایگزین کنید.

$$F \leftarrow RF_n$$

$$r_0, r_1, ..., r_{q(n)} \leftarrow \{0,1\}^n$$

$$(r_0, m_0 \oplus F(r_0)), (r_1, m_1 \oplus F(r_1)), ..., (r_{q(n)}, m_{q(n)} \oplus F(r_{q(n)}))$$

- H_3: از OTP برای رمز کردن $m_0, m_1, ..., m_{q(n)}$ استفاده کنید.

$$p_0, p_1, ..., p_{q(n)} \leftarrow \{0,1\}^n$$

$$r_0, r_1, ..., r_{q(n)} \leftarrow \{0,1\}^n$$

$$(r_0, m_0 \oplus p_0), (r_1, m_1 \oplus p_1), ..., (r_{q(n)}, m_{q(n)} \oplus p_{q(n)})$$

- H_4: از OTP برای رمز کردن $m'_0, m'_1, ..., m'_{q(n)}$ استفاده کنید.

$$p_0, p_1, ..., p_{q(n)} \leftarrow \{0,1\}^n$$

$$r_0, r_1, ..., r_{q(n)} \leftarrow \{0,1\}^n$$

$$(r_0, m'_0 \oplus p_0), (r_1, m'_1 \oplus p_1), ..., (r_{q(n)}, m'_{q(n)} \oplus p_{q(n)})$$

- H_5: به جای p_i ها از تابع واقعا تصادفی F استفاده کنید.

$$F \leftarrow \{\{0,1\}^n \rightarrow \{0,1\}^n\}$$

$$r_0, r_1, ..., r_{q(n)} \leftarrow \{0,1\}^n$$

$$(r_0, m'_0 \oplus F(r_0)), (r_1, m'_1 \oplus F(r_1)), ..., (r_{q(n)}, m'_{q(n)} \oplus F(r_{q(n)}))$$

- $H_۶$: توزیع رشته‌های رمز شده $m'_۰, m'_۱, ..., m'_{q(n)}$ به وسیله سیستم رمز ارائه شده در تعریف ۱:

$$K \leftarrow \{۰, ۱\}^n$$

$$r_۰, r_۱, ..., r_{q(n)} \leftarrow \{۰, ۱\}^n$$

$$(r_۰, m'_۰ \oplus f_k(r_۰)), (r_۱, m'_۱ \oplus f_k(r_۱)), ..., (r_{q(n)}, m'_{q(n)} \oplus f_k(r_{q(n)}))$$

این دقیقا همان چیزی است که مهاجم با دریافت عبارت رمز شده $m'_۰, m'_۱, ..., m'_{q(n)}$ ، می‌بیند.

با استفاده از لم هیبرید، $\mathcal{D}$ می‌تواند یکی از دو هیبرید مجاور را با احتمالی غیر قابل اجتناب، تمیز دهد. ما با استفاده از این موضوع به تناقض می‌رسیم:

- $H_۱ \approx H_۲$: توجه کنید که $\mathcal{D}$ تنها می‌تواند با احتمالی ناچیز[۱۰]، $H_۱$ و $H_۲$ را از هم تمیز دهد، در غیر این صورت با توجه به شبه تصادفی بودن $f_s(n)$ به تناقض می‌رسیم.
- $H_۲ \approx H_۳$: این دو توزیع، تقریبا در اکثر مواقع تمایزناپذیر هستند. تنها حالتی که ممکن است مشکل ایجاد کند، زمانی است که i و j مختلفی وجود داشته باشند به گونه که $r_i = r_j$ اما این تنها با احتمال

$$\begin{pmatrix} q(n) \\ ۲ \end{pmatrix} .۲^{-n}$$

اتفاق می‌افتد، که خود ناچیز است. بنابرین $\mathcal{D}$، دو هیبرید $H_۲$ و $H_۳$ را با احتمالی ناچیز، تمیز می‌دهد.

- $H_۳ \approx H_۴$: دو هیبرید $H_۳$ و $H_۴$، با توجه به اینکه OTPدارای امنیت کامل است، دارای توزیع یکسانی هستند.
- $H_۴ \approx H_۵$: استدلالی مشابه با $H_۲ \approx H_۳$
- $H_۵ \approx H_۶$: استدلالی مشابه با $H_۱ \approx H_۲$

■

۲ رمزنگاری نامتقارن

تا به اینجا، مدل ارتباطی، به ما امکان تبادل کلید، برای برقراری یک ارتباط امن را می‌داد. ما این تبادل کلید را تا به اینجا امن و انجام شدنی فرض کردیم، اما این موضوع خود موجب به وجود آمدن دسته‌ای از مسائل و روش‌های حل برای آن می‌شد.
در این بخش، می‌خواهیم روشی ارائه کنیم که نیاز به تبادل کلید را برای برقراری یک ارتباط امن، حذف کند. ایده این روش، استفاده از ۲ کلید، یکی برای رمز کردن و یکی برای رمز گشایی است. به کلیدی که برای رمز کردن استفاده می‌شود، کلید عمومی[۱۱] (pk) و به کلیدی که برای رمز گشایی از آن استفاده می‌شود، کلید خصوصی[۱۲]

[۱۰] negligible
[۱۱] public key
[۱۲] private key

(sk) می‌گوییم.
کلید عمومی همانطور که از نامش پیداست، در دسترس عموم برای رمز کردن پیام‌ها قرار می‌گیرد (این موضوع نشان می‌دهد که امنیت روش ما نباید بر روی کلید عمومی بنا شده باشد). کلید خصوصی تنها در اختیار گیرنده قرار می‌گیرد و برای رمز گشایی پیام‌ها استفاده می‌شود.
سیستم رمز نامتقارن را به صورت زیر تعریف می‌کنیم :

تعریف ۲ *سه‌تایی* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *یک سیستم رمز نامتقارن است اگر :*

- $\mathsf{Gen}(1^n)$ *یک الگوریتم* PPT *است که زوج کلید* (pk, sk) *را تولید می‌کند.*
- $\mathsf{Enc}_{pk}(m)$ *یک الگوریتم* PPT *است که با دریافت کلید* pk *و پیام* $m \in \{0,1\}^n$ *عبارت رمزی* c *را تولید می‌کند.*
- $\mathsf{Dec}_{sk}(c)$ *یک الگوریتم قطعی است که با گرفتن* c *و کلید* sk *، پیام* $m \in \{0,1\}^n$ *یا* $\perp$ *را تولید می‌کند.*
- *برای هر* n *و هر* (pk, sk) *تولید شده توسط* $\mathsf{Gen}(1^n)$ *و هر پیام* m *داریم:*

$$\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m$$

یا به عبارتی دقیق تر

$$\Pr[(pk, sk) \leftarrow \mathsf{Gen}(1^n) : \mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m] = 1$$

ما از نماد $\perp$ *، برای حالتی استفاده می‌کنیم که در آن عبارت رمزی، "غیر قابل رمزگشایی" باشد.*

تعریف ۳ آزمایش شنود[۱۳] $\mathsf{PubK}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$

۱. $\mathsf{Gen}(1^n)$ *برای دریافت جفت کلید* (pk, sk) *، اجرا می‌شود.*
۲. *کلید عمومی* pk *به مهاجم* $\mathcal{A}$ *داده می‌شود.*
۳. *مهاجم* $\mathcal{A}$ *دو پیام* m_0 *و* m_1 *که* $|m_0| = |m_1|$ *است را انتخاب و به خروجی خود می‌دهد.*
۴. *یک بیت تصادفی* $b \in \{0,1\}$ *انتخاب می‌شود و مقدار* $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ *به مهاجم* $\mathcal{A}$ *داده می‌شود.*
۵. *مهاجم* $\mathcal{A}$ *بیت* b' *را باز می‌گرداند.*

تعریف ۴ امنیت تک‌پیامی سیستم رمز نامتقارن

سیستم رمز نامتقارن $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *دارای امنیت تک‌پیامی*[۱۴] *، در حضور مهاجم شنودگر است، اگر برای هر مهاجم کارای احتمالاتی*[۱۵] *تابع ناچیز* $\varepsilon(n)$ *وجود داشته باشد که*

$$\Pr[\mathsf{PubK}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \varepsilon(n)$$

[۱۳] eavasedropping experiment
[۱۴] single-message secure
[۱۵] probabilistic polynomial time

تفاوت تعریف بالا، با تعریف مشابه آن، برای سیستم‌های رمز متقارن، در این است که به مهاجم کلید pk و درنتیجه دسترسی به اوراکل رمز کردن، داده شده است.
برخلاف رمز متقارن، در مورد رمز نامتقارن داریم:

۱. امنیت تک‌پیامی در برابر شنودگر، امنیت چندپیامی در برابر شنودگر را نتیجه می‌دهد.

۲. امنیت در برابر مهاجم شنودگر، امنیت در برابر مهاجم با قابلیت دسترسی به اوراکل رمز کردن را نتیجه می‌دهد.