خواهد بود. این ویژگی‌ها تنها زمانی برقرار خواهند بود- درواقع تنها زمانی معنی خواهند داشت- که H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شده باشد.

قضیه‌ی ۱۱-۲۱

اگر مسئله‌ی CDH نسبت به $\mathcal{G}$ سخت باشد و H به شکل یک پیشگوی تصادفی مدل‌سازی شود، آنگاه ساختار ۱۱-۱۹، CPA-امن است.

اثبات

فرض کنید Π نشان‌دهنده‌ی ساختار ۱۱-۱۹ باشد و فرض کنید $\mathcal{A}$ یک متخاصم PPT باشد. می‌خواهیم نشان دهیم که یک تابع ناچیز $negl$ وجود دارد به صورتی که

$$\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\mathrm{negl}(n).$$

احتمال بالا نیز روی انتخاب یکنواخت تابع H گرفته شده است که دسترسی به آن به $\mathcal{A}$ داده می‌شود.

اجرایی از آزمایش $\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)$ را در نظر بگیرید که در آن، کلید عمومی برابر با $\langle \mathbb{G}, q, g, h\rangle$ و متن رمز برابر با $c=g^y$ باشد و فرض کنید $Query$ رخدادی باشد که در آن، $\mathcal{A}$، $\mathrm{DH}_g(h,c)=h^y$ را از H پرسمان می‌کند. داریم

$$\begin{aligned}\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1] &= \Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\mathrm{Query}}] \\ &\quad + \Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 \wedge \mathrm{Query}] \\ &\le \Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\mathrm{Query}}] + \Pr[\mathrm{Query}]. \quad (11.18)\end{aligned}$$

اگر $\Pr[\overline{\mathrm{Query}}]=0$، آنگاه $\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\mathrm{Query}}]=0$. در غیر این صورت،

$$\begin{aligned}\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 \wedge \overline{\mathrm{Query}}] &= \Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 | \overline{\mathrm{Query}}] \cdot \Pr[\overline{\mathrm{Query}}] \\ &\le \Pr[\mathrm{KM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 | \overline{\mathrm{Query}}].\end{aligned}$$

در آزمایش $\mathrm{KM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)$، متخاصم $\mathcal{A}$، کلید عمومی و متن رمز به‌علاوه‌ی کلید لفافه بندی شده‌ی $k \stackrel{\mathrm{def}}{=} H(h^y)$ یا یک کلید یکنواخت را دریافت می‌کند. اگر $Query$ رخ ندهد، آنگاه k دارای توزیع یکنواخت از دید متخاصم خواهد بود و بنابراین هیچ روشی برای $\mathcal{A}$ وجود ندارد که بین این دو حالت، تمایز قائل شود. این امر به این معنا است که

$$\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1 | \overline{\mathrm{Query}}] = \frac{1}{2}.$$

بنابراین با بازگشت به معادله‌ی (۱۱.۱۸) داریم

$$\Pr[\mathrm{KEM}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\Pr[\mathrm{Query}].$$

در ادامه نشان می‌دهیم که $\Pr[\mathrm{Query}]$ ناچیز است که اثبات را کامل می‌کند.

فرض کنید $t=t(n)$ یک کران بالا (چندجمله‌ای) بر روی تعداد پرسمان‌هایی باشد که $\mathcal{A}$، از پیشگوی تصادفی H می‌پرسد. الگوریتم PPT زیر، $\mathcal{A}'$ را برای مسئله‌ی CDH نسبت به $\mathcal{G}$ تعریف کنید:

الگوریتم $\mathcal{A}'$

این الگوریتم به‌عنوان ورودی، $\mathbb{G}, q, g, h, c$ را دریافت می‌کند.

- قرار می‌دهیم $pk=\langle \mathbb{G}, q, g, h\rangle$ و یک $k \in \{0,1\}^{\ell}$ یکنواخت را انتخاب می‌کنیم.
- $\mathcal{A}(pk, c, k)$ را اجرا می‌کنیم. زمانی که $\mathcal{A}$ یک پرسمان از H می‌پرسد، پاسخ را با انتخاب یک رشته‌ی l-بیتی یکنواخت تازه، ارائه می‌کنیم.
- در انتهای اجرای $\mathcal{A}$، فرض می‌کنیم $y_1, \ldots, y_t$ عبارت باشد از فهرست پرسمان‌هایی که $\mathcal{A}$ از H پرسیده است. یک نمایه‌ی یکنواخت $i \in \{1, \ldots, t\}$ را انتخاب کرده و خروجی y_i را ارائه می‌کنیم.

ملاحظه‌مند به این احتمال هستیم که $\mathcal{A}'$ مسئله‌ی CDH را حل کند، یعنی $\Pr[\mathcal{A}'(\mathbb{G}, q, g, h, c) = \mathrm{DH}_g(h,c)]$، که در آن، احتمال روی $\mathbb{G}, q, g$ خروجی توسط $\mathcal{G}(1^n)$، مقادیر $h, c \in \mathbb{G}$ یکنواخت و مقدار تصادفی $\mathcal{A}'$ گرفته شده است. به‌منظور تحلیل این احتمال، در ابتدا توجه کنید که رخداد $Query$ در اجرای $\mathcal{A}'$ همچنان خوش‌تعریف است هرچند $\mathcal{A}'$ نمی‌تواند رخداد یا عدم رخداد آن را تشخیص دهد. به‌علاوه، احتمال رخداد $Query$ زمانی که $\mathcal{A}$ به‌عنوان یک زیرروال توسط $\mathcal{A}'$ اجرا می‌شود برابر با احتمال رخداد $Query$ در آزمایش $KEM^{cpa}_{\mathcal{A},\Pi}(n)$ است. این نتیجه حاصل می‌شود به این دلیل که دیدگاه $\mathcal{A}$ در هر دو حالت یکسان است تا زمانی که رخداد $Query$ اتفاق می افتد: در هر حالت، $\mathbb{G}, q, g$ به‌عنوان خروجی توسط $\mathcal{G}(1^n)$ ارائه می‌شوند؛ در هر حالت، h و c عناصر یکنواختی از $\mathbb{G}$ هستند و k یک رشته‌ی l-بیتی یکنواخت است و در هر حالت، پرسمان‌ها از H به‌غیراز $H(\mathrm{DH}_g(h,c))$ با یک رشته‌ی l-بیتی یکنواخت پاسخ داده می‌شود. (در $KEM^{cpa}_{\mathcal{A},\Pi}(n)$، پرسمان $H(\mathrm{DH}_g(h,c))$ با کلید لفافه بندی شده‌ی واقعی، پاسخ داده می‌شوند که با احتمال ۱/۲ برابر با k است در حالی که زمانی که $\mathcal{A}$ به‌عنوان یک زیرروال توسط $\mathcal{A}'$ اجرا می‌شود، پرسمان $H(\mathrm{DH}_g(h,c))$ با یک رشته‌ی l-بیتی یکنواخت پاسخ داده می‌شود که مستقل از k است. ولی زمانی که این پرسمان انجام می‌شود، رخداد $Query$ اتفاق می افتد.)

درنهایت، توجه کنید که زمانی که $Query$ رخ می‌دهد، آنگاه بر اساس تعریف، $\mathrm{DH}_g(h,c) \in \{y_1, \ldots, y_t\}$ است و بنابراین، خروجی $\mathcal{A}'$ با احتمال حداقل $1/t$، برابر با نتیجه‌ی صحیح $\mathrm{DH}_g(h,c)$ خواهد بود. بنابراین، نتیجه می‌گیریم که

$$\Pr[\mathcal{A}'(\mathbb{G}, q, g, h, c) = \mathrm{DH}_g(h, c)] \geq \Pr[\text{ Query }]/t,$$

یا $\Pr[\text{ Query }] \leq t \cdot \Pr[\mathcal{A}'(\mathbb{G}, q, g, h, c) = \mathrm{DH}_g(h, c)] = \mathrm{DH}_g(h, c)$. ازآنجاکه مسئله‌ی CDH برای $\mathcal{G}$ سخت است، این احتمال ناچیز است؛ ازآنجاکه t، چندجمله‌ای است، این امر به شکل ضمنی نشان می‌دهد که $\Pr[Query]$ هم ناچیز است. این امر، اثبات را کامل می‌کند. ■

در بخش بعدی مشاهده می‌کنیم که می‌توان نشان داد ساختار ۱۱-۱۹ تحت یک‌گونه‌ی قوی‌تر از فرض CDH دارای امنیت CCA است (اگر همچنان H را به‌عنوان یک پیشگوی تصادفی، مدل‌سازی کنیم).

۱۱-۴-۴- امنیت متن رمز منتخب و *DHIES/ECIES*

طرح رمزگذاری الجمال در برابر حملات متن رمز منتخب، آسیب‌پذیر است. این امر از این واقعیت ریشه می‌گیرد که این طرح، «شکل‌پذیر» است. یادآوری می‌کنیم که یک طرح رمزگذاری، به شکل غیررسمی، شکل‌پذیر است اگر با داشت نیک متن رمز c که یک رمزگذاری از یک پیام نامعلوم مفروض m است، می‌توان یک متن رمز تغییریافته‌ی c' را تولید نمود که یک رمزگذاری از یک پیام m' است که دارای نوعی رابطه با m است. در حالت رمزگذاری الجمال، یک متخاصم $\mathcal{A}$ را در نظر بگیرید که یک متن رمز $c = \langle c_1, c_2 \rangle$ رمز شده با استفاده از کلید عمومی $pk = \langle \mathbb{G}, q, g, h \rangle$ را به دست می‌آورد و در ادامه یک متن رمز تغییریافته‌ی $c' = \langle c_1, c_2' \rangle$ را می‌سازد که در آن، $c_2' = c_2 \cdot \alpha$ برای یک $\alpha \in \mathbb{G}$ مفروض. اگر c یک رمزگذاری از یک پیام $m \in \mathbb{G}$ باشد (که ممکن است برای $\mathcal{A}$ نامعلوم باشد)، داریم $c_1 = g^y$ و $c_2 = h^y \cdot m$ برای یک $y \in \mathbb{Z}_q$ مفروض. ولی آنگاه

$$c_1 = g^y \qquad \text{و} \qquad c_2' = h^y \cdot (\alpha \cdot m)$$

و بنابراین، c' یک رمزگذاری معتبر از پیام $\alpha.m$ است. به‌بیان‌دیگر، $\mathcal{A}$ می‌تواند «یک رمزگذاری از پیام (نامعلوم) m را به یک رمزگذاری از پیام (نامعلوم) $\alpha.m$ تبدیل کند». همان‌طور که در سناریوی سوم در بخش ۱۱-۲-۳ بحث شد، این نوع حمله می‌تواند نتایج وخیمی داشته باشد.

KEM بحث شده در بخش قبلی نیز ممکن است بسته به تابع خاص برگرفتن کلید H مورد استفاده، شکل‌پذیر باشد. بااین‌حال، اگر H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شود، آنگاه چنین حملاتی دیگر ممکن به نظر نمی‌رسند. درواقع، می‌توانیم بر اساس فرض به‌اصطلاح «شکاف-CDH»، اثبات کنیم که در این حالت، ساختار ۱۱-۱۹ دارای امنیت CCA خواهد بود. یادآوری می‌کنیم که فرض CDH بیان می‌کند که با داشتن اعضای گروه g^x . g^y (برای یک مولد g مفروض)، محاسبه‌ی g^{xy} غیرعملی خواهد بود. فرض شکاف-CDH بیان می‌کند که این امر همچنان غیرعملی است حتی با داشتن دسترسی به یک پیشگوی O به گونه ای که $O(U, V)$ دقیقاً هنگامی که $V = U^y$ است، مقدار ۱ را بازگرداند. به‌بیان‌دیگر، مسئله‌ی CDH همچنان سخت خواهد بود حتی با داشتن پیشگویی که

مسئله‌ی DDH را حل می‌کند. (تعریف رسمی را ارائه نمی‌کنیم چراکه از این فرض در مابقی کتاب استفاده نخواهیم کرد.) باور بر این است که این فرض برای دسته‌های گروه‌های مورد بحث در این کتاب، برقرار است. اثبات قضیه‌ی زیر بسیار شبیه به اثبات قضیه‌ی ۱۱-۳۸ است.

قضیه‌ی ۱۱-۲۲

اگر مسئله‌ی شکاف-CDH نسبت به $\mathcal{G}$ سخت باشد و H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شود، آنگاه ساختار ۱۱-۱۹ یک KEM دارای امنیت CCA است.

جالب است توجه کنیم که همین ساختار (یعنی ساختار ۱۱-۱۹) را می‌توان تحت فرضیات مختلف و در مدل‌های مختلف مورد تحلیل قرار داد و نتایج متفاوتی به دست آورد. با این فرض که فقط مسئله‌ی DDH، سخت است (و برای H که به شکل مناسب انتخاب شده است)، این طرح، CPA-امن خواهد بود. اگر ما H را به‌عنوان یک پیشگوی تصادفی مدل‌سازی کنیم (که الزامات شدیدتری برای H ایجاد می‌کند)، آنگاه تحت فرض ضعیف‌تر CDH، امنیت CPA را به دست می‌آوریم و تحت فرض قوی‌تر شکاف-CDH، امنیت CCA را به دست خواهیم آورد.

DHIES/ECIES

ساختار ۱۱-۲۳

فرض کنید $\mathcal{G}$ به شکل تعریف شده در متن باشد. فرض کنید $\Pi_E = (Enc', Dec')$ یک طرح رمزگذاری کلید خصوصی و $\Pi_M = (Mac, Vrfy)$ یک کد احراز اصالت پیام باشد. یک طرح رمزگذاری کلید عمومی را به شکل زیر تعریف کنید:

- *Gen*: با دریافت 1^n به‌عنوان ورودی، $\mathcal{G}(1^n)$ را اجرا می‌کنیم تا $(\mathbb{G}, q, g)$ به دست آید. یک $x \in \mathbb{Z}_q$ یکنواخت را انتخاب می‌کنیم، قرار می‌دهیم $h := g^x$ و یک تابع $H: \mathbb{G} \to \{0,1\}^n$ را مشخص می‌کنیم. کلید عمومی برابر با $\langle \mathbb{G}, q, g, h, H \rangle$ و کلید خصوصی برابر با $\langle \mathbb{G}, q, g, x, H \rangle$ است.
- *Enc*: با دریافت یک کلید عمومی $pk = \langle \mathbb{G}, q, g, h, H \rangle$ به‌عنوان ورودی، یک $y \in \mathbb{Z}_q$ یکنواخت را انتخاب کرده و قرار می‌دهیم $k_E \| k_M := H(h^y)$. $c' \leftarrow \mathrm{Enc}'_{k_E}(m)$ را محاسبه کرده و متن رمز $\langle g^y, c', \mathrm{Mac}_{k_M}(c') \rangle$ را به‌عنوان خروجی ارائه می‌کنیم.
- *Dec*: با دریافت یک کلید خصوصی $sk = \langle \mathbb{G}, q, g, x, H \rangle$ و یک متن رمز $\langle c, c', t \rangle$ به‌عنوان ورودی، خروجی $\perp$ را ارائه می‌کنیم اگر $c \notin \mathbb{G}$. در غیر این صورت، $k_E \| k_M := H(c^x)$ را محاسبه می‌کنیم. اگر $\mathrm{Vrfy}_{k_M}(c', t) \neq 1$، آنگاه خروجی $\perp$ را ارائه می‌کنیم؛ در غیر این صورت، $\mathrm{Dec}'_{k_E}(c')$ را به‌عنوان خروجی ارائه می‌کنیم.

رمزگذاری *CCA*-امن با ساختار ۱۱-۱۹

ترکیب *KEM* در ساختار ۱۱-۱۹ با هر طرح رمزگذاری کلید خصوصی *CCA*-امن، یک طرح رمزگذاری کلید عمومی *CCA*-امن را می‌دهد. (ن.ک. قضیه‌ی ۱۱-۱۴.) نمونه‌سازی این رویکرد با استفاده از ساختار ۴-۱۸ برای مؤلفه‌ی کلید خصوصی منطبق با آن چیزی است که در *DHIES/ECIES* انجام می‌شود که گونه‌هایی از آن‌ها در استاندارد ۲ – ۱۸۰۳۳ *ISO/IES* برای رمزگذاری عمومی، آورده شده است. (ن.ک. ساختار ۱۱-۲۳.) رمزگذاری یک پیام m در این طرح‌ها به شکل زیر خواهد بود

$$\langle g^y, \mathrm{Enc}'_{k_E}(m), \mathrm{Mac}_{k_M}(c') \rangle,$$

که در آن، Enc' نشان‌دهنده‌ی یک طرح رمزگذاری کلید خصوصی *CPA*-امن و c' نشان‌دهنده‌ی $\mathrm{Enc}'_{k_E}(m)$ است. «طرح رمزگذاری ادغامی دیفی-هلمن»، *DHIES*، را می‌توان به‌صورت کلی برای اشاره به هر طرح با این شکل یا برای اشاره‌ی خاص به هر حالتی مورد استفاده قرار داد که در آن، گروه $\mathbb{G}$ یک زیرگروه دوری از یک میدان متناهی باشد. «طرح رمزگذاری ادغامی خم بیضوی»، *ECIES*، به حالتی اشاره می‌کند که در آن، $\mathbb{G}$ یک گروه خم بیضوی است. بیان می‌کنیم که در ساختار ۱۱-۲۳، ضروری است که طی رمزگشایی بررسی کنیم که c، یعنی اولین مؤلفه‌ی متن رمز، در $\mathbb{G}$ باشد. در غیر این صورت، مهاجم ممکن است رمزگشایی یک متن رمز بدریخت $\langle c, c', t \rangle$ را درخواست کند که در آن، $c \notin \mathbb{G}$؛ رمزگشایی چنین متن رمزی (یعنی بدون بازگرداندن $\perp$) ممکن است اطلاعاتی در مورد کلید خصوصی، افشا کند.

بر اساس قضیه‌ی ۴-۱۹، رمزگذاری یک پیام و سپس اعمال یک کد احراز اصالت پیام (قوی) یک طرح رمزگذاری کلید خصوصی *CCA*-امن را می‌دهد. با ترکیب این امر با قضیه‌ی ۱۱-۱۴، نتیجه می‌گیریم که:

نتیجه‌ی فرعی ۱۱-۲۴

فرض کنید Π_E یک طرح رمزگذاری کلید خصوصی *CPA*-امن و Π_M یک کد احراز اصالت پیام دارای امنیت قوی باشد. اگر مسئله‌ی شکاف-*CDH* نسبت به $\mathcal{G}$ سخت باشد و H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شده باشد، آنگاه ساختار ۱۱-۲۳ یک طرح رمزگذاری کلید عمومی *CCA*-امن خواهد بود.

۱۱-۵- رمزگذاری *RSA*

در این بخش، توجه خود را به طرح‌های رمزگذاری مبتنی بر «فرض *RSA*» تعریف‌شده در بخش ۸-۲-۴ معطوف می‌کنیم. بیان می‌کنیم که هرچند رمزگذاری مبتنی بر *RSA* امروزه به‌صورت گسترده مورداستفاده است، در حال حاضر یک تغییر تدریجی از استفاده از *RSA*- به سمت استفاده از

سامانه‌های رمزنگاری مبتنی بر *CDH/DDH* با تکیه بر گروه‌های خم بیضوی- به دلیل طول کلیدهای طولانی‌تر موردنیاز برای طرح‌های مبتنی بر *RSA* وجود دارد. برای بحث بیشتر در خصوص این امر به بخش ۹-۳ مراجعه کنید.

۱۱-۵-۱- *RSA* ساده

با توصیف یک طرح رمزگذاری ساده بر اساس مسئله‌ی *RSA* بحث را آغاز می‌کنیم. هرچند این طرح، ناامن است ولی یک نقطه‌ی شروع مفید برای طرح‌های امن بعدی فراهم می‌کند.

فرض کنید *GenRSA* یک الگوریتم *PPT* باشد که با دریافت 1^n به‌عنوان ورودی، یک پیمانه‌ی N که حاصل‌ضرب دو عدد اول n-بیتی است به همراه اعداد صحیح e, d صادق در $ed = 1 \bmod \phi(N)$ را به‌عنوان خروجی ارائه می‌کند. (مثل همیشه، الگوریتم ممکن است با احتمال ناچیز، شکست بخورد ولی ما این امر را در اینجا نادیده می‌گیریم.) از بخش ۸-۲-۴ به یاد داریم که چنین الگوریتمی را پیاده‌گی می‌توان از روی هر الگوریتم *GenModulus* ساخت که یک پیمانه‌ی مرکب N به همراه تجزیه‌ی آن را می‌دهد؛ ن.ک. الگوریتم ۱۱-۲۵.

الگوریتم ۱۱-۲۵

تولید کلید *RSA*؛ *GenRSA*

ورودی: پارامتر امنیتی 1^n

خروجی: N, e, d به‌صورت توصیف‌شده در متن.

$(N, p, q) \leftarrow$ GenModulus (1^n)

$\phi(N) := (p-1)(q-1)$

$e > 1$ را به‌گونه‌ای انتخاب کنید که $\gcd(e, \phi(N)) = 1$ باشد.

compute $d := [e^{-1} \bmod \phi(N)]$

return N, e, d

فرض کنید N, e, d به‌مانند بالا باشد و قرار دهید $c = m^e \bmod N$ رمزگذاری *RSA* به این واقعیت تکیه دارد که شخصی که d را بداند می‌تواند m را از روی c با محاسبه‌ی $[c^d \bmod N]$ بازیابی کند؛ این امر ممکن است به این دلیل که

$$c^d = (m^e)^d = m^{ed} = m \bmod N,$$

همان‌طور که در بخش ۸-۲-۴ بحث شد. از سوی دیگر، بدون دانستن d (حتی اگر N و e معلوم باشند)، فرض *RSA* (ن.ک. تعریف ۸-۴۶) به شکل ضمنی نشان می‌دهد که بازیابی m از روی c سخت است، حداقل اگر m به‌صورت یکنواخت از $\mathbb{Z}_N^*$ انتخاب شده باشد. این امر به شکل طبیعی، طرح رمزگذاری کلید عمومی نشان داده‌شده در ساختار ۱۱-۲۶ را نشان می‌دهد: گیرنده، *GenRSA*

را برای به دست آوردن N, e, d اجرا می‌کند، N و e را به‌عنوان کلید عمومی خود منتشر می‌کند و d را در کلید خصوصی خود نگه می‌دارد. به‌منظور رمزگذاری یک پیام[۴۲] $m \in \mathbb{Z}_N^*$، گیرنده- که d را می‌داند- می‌تواند c را رمزگشایی کرده و m را بازیابی کند.

طرح رمزگذاری RSA ساده

ساختار ۱۱-۲۶

فرض کنید $GenRSA$ به شکل تعریف‌شده در متن باشد. یک طرح رمزگذاری کلید عمومی را به شکل زیر تعریف کنید:

- *Gen*: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را اجرا می‌کنیم تا N، e و d به دست آید. کلید عمومی برابر با $\langle N, e \rangle$ و کلید خصوصی برابر با $\langle N, d \rangle$ است.
- *Enc*: با دریافت یک کلید عمومی $pk = \langle N, e \rangle$ و یک پیام $m \in \mathbb{Z}_N^*$ به‌عنوان ورودی، متن رمز زیر را محاسبه می‌کنیم:
$$c := [m^e \bmod N]$$
- *Dec*: با دریافت یک کلید خصوصی $sk = \langle N, d \rangle$ و یک متن رمز $c \in \mathbb{Z}_N^*$ به‌عنوان ورودی، پیام زیر را محاسبه می‌کنیم
$$m := [c^d \bmod N].$$

در ادامه‌یک مثال حل‌شده از بحث بالا، ارائه می‌شود (همچنین ن.ک. مثال ۸-۴۹).

مثال ۱۱-۲۷

فرض کنید $GenRSA$ خروجی $(N, e, d) = (۳۹۱, ۳, ۲۳۵)$ را ارائه نماید. (توجه کنید که $۳۹۱ = ۱۷ \cdot ۲۳$ و بنابراین $\phi(۳۹۱) = ۱۶ \cdot ۲۲ = ۳۵۲$. به‌علاوه، $۳ \cdot ۲۳۵ = ۱ \bmod ۳۵۲$.) بنابراین، کلید عمومی برابر با $(۳۹۱, ۳)$ و کلید خصوصی برابر با $(۳۹۱, ۲۳۵)$ است.

به‌منظور رمزگذاری پیام $m = ۱۵۸ \in \mathbb{Z}_{۳۹۱}^*$ با استفاده از کلید عمومی $(۳۹۱, ۳)$، مقدار $c := [۱۵۸^۳ \bmod ۳۹۱] = ۲۹۵$ را محاسبه می‌کنیم؛ این همان متن رمز است. به‌منظور رمزگشایی، گیرنده، $۱۵۸ = [۲۹۵^{۲۳۵} \bmod ۳۹۱]$ را محاسبه می‌کند. ◊

آیا طرح رمزگذاری RSA، امن است؟ فرض تجزیه به شکل ضمنی نشان می‌دهد که برای مهاجمی که کلید عمومی را دریافت می‌کند، به دست آوردن کلید خصوصی متناظر، ازلحاظ محاسباتی

[۴۲] فرض می‌کنیم که $m \in \mathbb{Z}_N^*$. اگر تجزیه‌ی N سخت باشد، یافتن یک $m \in \{۱, \ldots, N-۱\}$ با $m \notin \mathbb{Z}_N^*$ ازلحاظ محاسباتی دشوار خواهد بود (چراکه در این صورت، $\gcd(m, N)$ یک عامل غیربدیهی از N است).

غیرعملی است؛ ن.ک. بخش ۸-۲-۵. این امر برای اینکه یک طرح رمزگذاری کلید عمومی، امن باشد، لازم است ولی کافی نیست. فرض RSA نشان می‌دهد که اگر پیام m به‌صورت یکنواخت از $\mathbb{Z}_N^*$ انتخاب شود، آنگاه یک شنودگر با داشتن N، e و d (یعنی، کلید عمومی و متن رمز) نمی‌تواند m را بازیابی کند. ولی این‌ها تضمین‌های ضعیفی هستند و سطح امنیتی را که ما می‌خواهیم به‌هیچ‌وجه تأمین نمی‌کنند! علی‌الخصوص، این امکان را باقی می‌گذارند که مهاجم بتواند پیام را هنگامی‌که به‌صورت یکنواخت از $\mathbb{Z}_N^*$ انتخاب نشده است، بازیابی نماید (درواقع، زمانی که m از یک دامنه‌ی کوچک انتخاب می‌شود، مشاهده‌ی اینکه مهاجم می‌تواند m را از روی کلید عمومی و متن رمز، محاسبه کند، بسیار ساده است). به‌علاوه، این امر این امکان را از بین نمی‌برد که مهاجم می‌تواند «اطلاعات جزئی» در مورد پیام به دست آورد حتی زمانی که یکنواخت باشد (درواقع، ممکن بودن این امر مشخص شده است). همچنین، رمزگذاری RSA ساده، «قطعی» است و بنابراین باید ناامن باشد، همان‌طور که قبلاً در بخش ۱۱-۲-۱ بحث کردیم.

حملات بیشتر علیه RSA ساده

قبلاً اشاره کردیم که رمزگذاری RSA ساده، CPA-امن نیست. بااین‌حال، ممکن است وسوسه شویم تا از RSA ساده برای رمزگذاری «پیام‌های تصادفی» و/یا در وضعیت‌هایی استفاده کنیم که در آن‌ها، فضای اندک بیت‌های اطلاعاتی در خصوص پیام، قابل‌قبول باشد. به‌طورکلی هشدار می‌دهیم چنین کاری انجام ندهید و در اینجا تنها تعدادی از مشکلات احتمالی این شیوه را ارائه می‌کنیم.

بعضی از حملاتی که در ادامه می‌آیند، فرض می‌کنند که $e = ۳$. در بعضی از موارد، حملات را می‌توان حداقل به‌صورت جزئی به e بزرگ تر تعمیم داد؛ درهرصورت، همان‌طور که در بخش ۸-۲-۴ اشاره شد، قرار دادن $e = ۳$ اغلب در عمل انجام می‌شود. این حملات نشان می‌دهند که ساختار ۱۱-۲۶، ناکافی است و نه اینکه قرار دادن $e = ۳$ ضرورتاً یک انتخاب بد است.)

یک بهبود درجه دوم در بازیابی m

ازآنجاکه رمزگذاری RSA ساده، قطعی است، می‌دانیم که اگر $m < B$ باشد، آنگاه مهاجم می‌تواند m را از روی متن رمز $c = [m^e \bmod N]$ در زمان $O(B)$ با استفاده از حمله‌ی جستجوی فراگیر موردبحث در بخش ۱۱-۲-۱، تعیین کند. بااین‌حال، ممکن است امیدوار باشیم که رمزگذاری RSA ساده را بتواند درصورتی‌که B بزرگ باشد، مورداستفاده قرار داد، یعنی، اگر پیام از یک مجموعه‌ی مطلقاً بزرگ از مقادیر، انتخاب شده باشد. یک سناریوی ممکن که در آن این رخداد ممکن است اتفاق بیفتد مربوط به حالت رمزگذاری ترکیبی (ن.ک. بخش ۱۱-۳) است که در آن، «پیام» یک کلمه n-بیتی تصادفی است و بنابراین $B = ۲^n$. متأسفانه، یک حمله‌ی هوشمندانه برای بازیابی m با احتمال بالا در زمان تقریباً $O(\sqrt{B})$ وجود دارد. این امر می‌تواند یک تفاوت چشم‌گیر را در عمل به وجود آورد: یک حمله‌ی (مثلاً) زمان $۲^{۸۰}$، غیرعملی است ولی حمله‌ی اجراشده در زمان $۲^{۴۰}$ را به شکل نسبتاً ساده می‌توان اجرا نمود.

توصیفی از این حمله به‌عنوان الگوریتم ۱۱-۲۸ ارائه شده است. در توصیف خود، فرض می‌کنیم که $B = 2^n$ و فرض می‌کنیم که $\alpha \in (\frac{1}{3}, 1)$ نشان‌دهنده‌ی یک مقدار ثابت مفروض باشد (ن.ک. ذیل).

پیچیدگی زمانی الگوریتم در سایه‌ی زمان موردنیاز برای مرتب کردن $2^{\alpha n}$ زوج (r, x_r) قرار می‌گیرد؛ این امر را می‌توان در زمان $O(n \cdot 2^{\alpha n})$ انجام داد. جستجوی دودویی در خط یکی به آخر برای بررسی این مسئله استفاده می‌شود که آیا یک r با $x_r = [s^e \bmod N]$ وجود دارد یا خیر.

الگوریتم ۱۱-۲۸

یک حمله علیه رمزگذاری RSA ساده

ورودی: کلید عمومی $\langle N, e\rangle$؛ متن رمز c

خروجی: $m < 2^n$ به صورتی که $m^e = c \bmod N$

$set\ T := 2^{\alpha n}$
$for\ r = 1\ to\ T:$
$x_r := [c/r^e \bmod N]$
زوج‌های $\{(r, x_r)\}_{r=1}^{T}$ را بر اساس مؤلفه‌ی دوم آن‌ها، مرتب می‌کنیم
$for\ s = 1\ to\ T:$
$if\ x_r \stackrel{?}{=} [s^e \bmod N]\ for\ some\ r$
$return\ [r \cdot s \bmod N]$

اکنون مشخص می‌کنیم چرا حمله، m را با احتمال بالا، بازیابی می‌کند. فرض کنید $c = m^e \bmod N$. برای انتخاب مناسب $\alpha > \frac{1}{3}$، می‌توان نشان داد که اگر m یک عدد صحیح n-بیتی یکنواخت باشد، آنگاه با احتمال بالا، r, s با $1 < r \le s \le 2^{\alpha n}$ وجود خواهند داشت که برای آن‌ها $m = r \cdot s$. (برای مثال، اگر $n = 64$ و بنابراین m یک رشته‌ی تصادفی 64-بیتی باشد، آنگاه با احتمال 0.35، مقادیر r, s با طول حداکثر ۳۴ بیت وجود خواهند داشت به صورتی که $m = r \cdot s$. برای جزئیات بیشتر به مراجع انتهای فصل مراجعه کنید.) با فرض اینکه این حالت، برقرار است، الگوریتم بالا، m را خواهد یافت چراکه

$$c = m^e = (r \cdot s)^e = r^e \cdot s^e \bmod N,$$

و بنابراین، $x_r = c/r^e = s^e \bmod N$ با $r, s < T$.

رمزگذاری پیام‌های کوتاه با استفاده از e کوچک

حمله‌ی قبلی نشان می‌دهد که چگونه می‌توان یک پیام m را که می‌دانیم کوچک‌تر از یک کران مفروض B است در زمان تقریباً $O(\sqrt{B})$ بازیابی کرد. در اینجا نشان می‌دهیم که چگونه همین کار را می‌توان در زمان $poly(\|N\|)$ انجام داد اگر $B \le N^{1/e}$ باشد (که در آن، این امر به این معنا است که ریشه‌ی eام عدد N یک عدد حقیقی است).

این حمله به این برداشت تکیه دارد که زمانی که $m < N^{1/e}$ باشد، به توان رساندن m تا توان eام به پیمانه‌ی N نیازمند هیچ کاهش پیمانه‌ای نیست؛ یعنی $[m^e \bmod N]$ برابر با عدد صحیح m^e است. این امر به این معنا است که با داشتن متن رمز $c = [m^e \bmod N]$، مهاجم می‌تواند m را با محاسبه‌ی $m := c^{1/e}$ «روی اعداد صحیح» (یعنی نه در پیمانه‌ی N) تعیین نماید؛ این کار را می توان به‌سادگی در زمان $\text{poly}(\|c\|) = \text{poly}(\|N\|)$ انجام داد چراکه یافتن ریشه‌های eام روی اعداد صحیح، راحت است و تنها زمانی سخت می‌شود که با $\bmod N$ کار کنیم.

برای e کوچک، این مقوله نشان‌دهنده‌ی یک ضعف جدی رمزگذاری RSA ساده است. برای مثال، اگر قرار دهیم $e = 3$ و فرض کنیم که $\|N\| \approx 1024\ bits$، آنگاه حمله مشمر ثمر است حتی زمانی که m یک عدد صحیح ۳۰۰-بیتی یکنواخت باشد؛ این امر یک‌بار دیگر امنیت RSA ساده را نفی می‌کند حتی زمانی که به‌عنوان بخشی از یک طرح رمزگذاری ترکیبی، استفاده شود.

رمزگذاری پیامی که بخشی از آن معلوم است

این حمله را می‌توان به‌عنوان یک تعمیم از حمله‌ی قبلی در نظر گرفت. این حمله، فرستنده‌ای را فرض می‌کند که پیامی را رمزگذاری می‌نماید که بخشی از آن معلوم است (چیزی که نباید منجر به حمله در هنگام استفاده از یک طرح رمزگذاری امن شود). در اینجا ما به یک نتیجه‌ی قوی کاپراسمیت اشاره می‌کنیم که بدون اثبات آن را بیان می‌کنیم:

قضیه‌ی ۱۱-۲۹

فرض کنید $p(x)$ یک چندجمله‌ای با درجه‌ی e باشد. آنگاه در زمان $\text{poly}(\|N\|, e)$، می‌توان تمام ریشه‌های x_0 را یافت به صورتی که $p(x_0) = 0 \bmod N$ و $|x_0| \le N^{1/e}$ باشد.

به دلیل وابستگی زمان اجرا به e، حمله تنها برای e کوچک، عملی خواهد بود. در ادامه برای درستی محاسبات، فرض می‌کنیم که $e = 3$ است.

فرض کنید فرستنده یک پیام $m = m_1 \| m_2$ را برای یک گیرنده با کلید عمومی $\langle N, 3\rangle$ ارسال می‌کند که در آن، بخش اول m_1 پیام، معلوم است ولی بخش دوم m_2 معلوم نیست. برای درستی محاسبات، فرض کنید m_2 دارای طول k بیت است، بنابراین $m = B \cdot m_1 + m_2$ که در آن، فرض می‌کنیم $B = 2^k$ است. با داشتن متن رمز به‌دست‌آمده‌ی $c = [(m_1 \| m_2)^3 \bmod N]$، یک شنودگر می‌تواند چندجمله‌ای مکعب $p(x) \stackrel{\text{def}}{=} (2^k \cdot m_1 + x)^3 - c$ را تعریف کند. این چندجمله‌ای دارای m_2 به‌عنوان یک ریشه (به پیمانه‌ی N) است و $|m_2| < B$. بنابراین، قضیه‌ی ۱۱-۲۹ نشان می‌دهد که مهاجم می‌تواند m_2 را به‌صورت کارا محاسبه کند مادامی که $B \le N^{1/3}$ باشد. یک جمله‌ی مشابه زمانی که m_2 معلوم است ولی m_1 معلوم نیست، قابل‌اجرا است.

رمزگذاری پیام‌های مرتبط[۴۵]

این حمله فرستنده‌ای را فرض می‌کند که دو پیام مرتبط را برای گیرنده‌ی یکسان، رمزگذاری می‌کند (چیزی که نباید منجر به حمله هنگام استفاده از یک طرح رمزگذاری امن شود). فرض کنید فرستنده m و $m+\delta$ را برای یک گیرنده با کلید عمومی $\langle N, e\rangle$ رمزگذاری می‌کند که در آن مقدار اضافی δ معلوم است ولی m معلوم نیست. با داشتن دو متن رمز $c_1 = [m^e \bmod N]$ و $c_2 = [(m+\delta)^e \bmod N]$، شنودگر می‌تواند دو چندجمله‌ای $f_1(x) \stackrel{def}{=} x^e - c_1$ و $f_2(x) \stackrel{def}{=} (x+\delta)^e - c_2$ را تعریف کند. توجه کنید که $x = m$ یکی از ریشه‌های (به پیمانه‌ی N)، هر کدام با درجه‌ی e، را تعریف کند. بنابراین جمله‌ی خطی $(x-m)$ یک عامل برای هر دو آن‌ها است. بنابراین، اگر بزرگ‌ترین مقسوم‌علیه مشترک $f_1(x)$ و $f_2(x)$ (به‌عنوان چندجمله‌ای هایی روی $\mathbb{Z}_N^*$) خطی باشد، m را افشا خواهد نمود. بزرگ‌ترین مقسوم‌علیه مشترک را می‌توان در زمان $poly(\|N\|, e)$ با استفاده از الگوریتمی مشابه الگوریتم موجود در ضمیمه‌ی ب-۱-۲ محاسبه نمود؛ بنابراین، این حمله برای e کوچک، عملی است.

ارسال پیام یکسان به چندین گیرنده[۴۶]

آخرین حمله‌ی ما، فرستنده‌ای را فرض می‌کند که پیام یکسانی را برای چندین گیرنده، رمزگذاری می‌کند (بازهم چیزی که نباید منجر به یک حمله هنگام استفاده از یک طرح رمزگذاری امن شود). فرض کنید $e = 3$ و فرض کنید که پیام یکسان m برای سه طرف متفاوت دارای کلیدهای عمومی به ترتیب $pk_1 = \langle N_1, 3\rangle$، $pk_2 = \langle N_2, 3\rangle$ و $pk_3 = \langle N_3, 3\rangle$ رمزگذاری می‌شود. فرض می‌کنیم برای i, j متمایز، $\gcd(N_i, N_j) = 1$؛ در غیر این صورت، حداقل یکی از پیمانه‌ها را می‌توان فوراً تجزیه نمود و پیام m را می‌توان به‌سادگی بازیابی کرد. شنودگر، مقادیر زیر را می‌بیند:

$$c_1 = [m^3 \bmod N_1], \quad c_2 = [m^3 \bmod N_2] \text{ و } c_3 = [m^3 \bmod N_3]$$

فرض کنید $N^* = N_1 N_2 N_3$. یک ویرایش تعمیم‌یافته از قضیه‌ی باقیمانده‌ی چینی می‌گوید که یک عدد صحیح غیرمنفی منحصربه‌فرد $\hat{c} < N^*$ وجود دارد به صورتی که

$$\hat{c} = c_1 \bmod N_1$$
$$\hat{c} = c_2 \bmod N_2$$
$$\hat{c} = c_3 \bmod N_3.$$

به‌علاوه، با استفاده از روش‌های مشابه با روش‌های نشان داده‌شده در بخش ۸-۱-۵، می‌توان $\hat{c}$ را به‌صورت کارا با داشتن کلیدهای عمومی و متون رمز بالا، محاسبه نمود. درنهایت توجه کنید که m^3

---

[۴۵] این حمله به جبری تکیه دارد که اندکی فراتر از آن چیزی است که در این کتاب بررسی کرده‌ایم.

[۴۶] این حمله به قضیه‌ی باقیمانده‌ی چینی ارائه شده در بخش ۸-۱-۵، تکیه می‌کند.

بر این معادلات صدق می‌کند و $m^3 < N^*$ چراکه $m < min\{N_1, N_2, N_3\}$. به‌مانند حمله‌ی قبلی، این امر به این معنا است که $\hat{c} = m^3$ «روی اعداد صحیح» (یعنی بدون استفاده از کاهش پیمانه‌ای) و بنابراین، پیام m را می‌توان با محاسبه‌ی ریشه‌ی سوم عدد صحیح $\hat{c}$، بازیابی نمود.

۱۱-۵-۲- RSA لایی گذاری شده و PKCS#1 v1.5

هرچند RSA ساده، ناامن است، ولی یک رویکرد کلی برای رمزگذاری کلید عمومی بر اساس مسئله‌ی RSA را می‌دهد: به‌منظور رمزگذاری یک پیام m با استفاده از کلید عمومی $\langle N, e\rangle$، ابتدا m را به یک عنصر $\hat{m} \in \mathbb{Z}_N^*$ نگاشت می‌کنیم؛ سپس، متن رمز $c = [\hat{m}^e \bmod N]$ را محاسبه می‌کنیم. به‌منظور رمزگشایی یک متن رمز c، گیرنده، $\hat{m} = [c^d \bmod N]$ را محاسبه کرده و سپس پیام اولیه‌ی m را بازیابی می‌کند. برای اینکه گیرنده قادر به بازیابی پیام باشد، نگاشت از پیام ها به عناصر $\mathbb{Z}_N^*$ باید (به شکل کارا) وارون‌پذیر باشد. برای اینکه امیدوار باشیم یک طرح ساخته‌شده بر اساس این رویکرد، CPA-امن باشد، نگاشت باید «تصادفی سازی» شود تا رمزگذاری، قطعی نباشد. این امر البته یک شرط لازم است ولی کافی نیست و امنیت طرح رمزگذاری به شکلی حیاتی به نگاشت خاص استفاده‌شده، بستگی خواهد داشت.

یک پیاده‌سازی ساده از ایده‌ی بالا عبارت است از «لایی گذاری تصادفی» پیام قبل از رمزگذاری. یعنی، برای نگاشت یک پیام m (مشاهده‌شده به‌عنوان یک رشته‌ی بیتی) به یک عنصر از $\mathbb{Z}_N^*$، فرستنده یک رشته‌ی بیتی یکنواخت $r \in \{0,1\}^\ell$ (برای l مناسب) انتخاب کرده و قرار می‌دهد $\hat{m} := r \| m$ (مقدار به‌دست‌آمده را به‌صورت طبیعی می‌توان به‌عنوان یک عدد صحیح در $\mathbb{Z}_N^*$ تعبیر نمود و این نگاشت به‌وضوح وارون‌پذیر است. ن.ک. ساختار ۱۱-۳۰. (کران‌ها بر روی $l(n)$ و طول m تضمین می‌کنند که عدد صحیح $\hat{m}$ کوچک‌تر از N باشد.)

این ساختار توسط یک مقدار l، پارامتری می‌شود که طول لایی گذاری تصادفی استفاده‌شده را تعیین می‌کند. امنیت این طرح به l بستگی دارد. یک حمله‌ی واضح جستجوی فراگیر علیه این طرح وجود دارد که در زمان 2^l اجرا می‌شود، بنابراین اگر l بیش‌ازحد کوتاه باشد (علی‌الخصوص اگر $\ell(n) = O(\log n)$)، این طرح، ناامن خواهد بود. در سوی دیگر، در بخش بعدی (اصولاً) نشان می‌دهیم که زمانی که لایی گذاری تا حد ممکن بزرگ باشد و m تنها یک بیت یکتا باشد، آنگاه اثبات امنیت بر اساس فرض RSA ممکن خواهد بود. در حالت‌های میانی، وضعیت تا این حد شفاف نیست: برای مقادیر دامنه‌های خاص از l، نمی‌توانیم امنیت را بر اساس فرض RSA اثبات کنیم ولی هیچ حمله‌ی واقعی چندجمله‌ای هم شناخته نشده است. بحث بیشتر را به بعد از بررسی PKCS□1 v1.5 در ادامه، موکول می‌کنیم.

طرح رمزگذاری RSA لایی گذاری شده

ساختار ۱۱-۳۰

فرض کنید $GenRSA$ به‌مانند قبل باشد و l تابعی با $\ell(n) \leq 2n - 4$ برای تمام nها باشد. یک طرح رمزگذاری کلید عمومی را به‌صورت زیر تعریف می‌کنیم:

- Gen: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را اجرا می‌کنیم تا (N,e,d) به دست آید. کلید عمومی $pk = \langle N,e \rangle$ و کلید خصوصی $sk = \langle N,d \rangle$ را به‌عنوان خروجی ارائه می‌کنیم.
- Enc: با دریافت یک کلید عمومی $pk = \langle N,e \rangle$ و یک پیام $m \in \{0,1\}^{\|N\|-\ell(n)-2}$ به‌عنوان ورودی، یک رشته‌ی یکنواخت $r \in \{0,1\}^{\ell(n)}$ را انتخاب کرده و $\hat{m} := r \| m$ را به‌عنوان عنصری از $\mathbb{Z}_N^*$ تعبیر می‌کنیم. متن رمز زیر را به‌عنوان خروجی ارائه می‌کنیم:
$$c := [\hat{m}^e \bmod N]$$
- Dec: با دریافت یک کلید خصوصی $sk = \langle N,d \rangle$ و یک متن رمز $c \in \mathbb{Z}_N^*$ به‌عنوان ورودی، مقدار زیر را محاسبه می‌کنیم:
$$\hat{m} := [c^d \bmod N]$$
و $\|N\| - \ell(n) - 2$ بیت با پایین‌ترین رتبه از $\hat{m}$ را به‌عنوان خروجی ارائه می‌کنیم.

RSA PKCS#1 v1.5

استاندارد رمزنگاری کلید عمومی آزمایشگاه‌های RSA ($PKCS$) #1 نگارش ۱.۵ صادرشده در سال ۱۹۹۳، از یک گونه از رمزگذاری RSA لایی گذاری شده استفاده می‌کند. برای یک کلید عمومی $pk = \langle N,e \rangle$ با شکل معمول، فرض کنید k نشان‌دهنده‌ی طول N به بایت باشد؛ یعنی k عبارت است از عدد صحیح صادق در $2^{8(k-1)} \leq N < 2^{8k}$. فرض می‌کنیم که طول پیام‌های m که باید رمز شوند برابر با ضریبی از ۸ بیت است و می‌توانند دارای طول‌های از یک تا $k - 11$ بایت باشند. رمزگذاری یک پیام m که دارای طول D بایت است به‌صورت زیر محاسبه می‌شود

$$[(0x00 \| 0x02 \| r \| 0x00 \| m)^e \bmod N],$$

که در آن، r یک رشته‌ی $(k - D - 3)$-بایتی است که به‌صورت تصادفی تولید شده است و هیچ‌کدام از بایت‌های آن برابر با $0x00$ نیستند. (شرط آخر اجازه می‌دهد که پیام هنگام رمزگشایی بدون ابهام بازیابی شود.) توجه کنید که طول ماکزیمم قابل‌قبول برای m تضمین می‌کند که طول r حداقل برابر با ۸ بایت باشد.

متأسفانه، PKCS#1 v1.5 به شکل مشخص شده، CPA-امن نیست به این دلیل که اجازه‌ی استفاده از لایی گذاری تصادفی را می‌کند که بیش‌ازحد کوتاه است. این مسئله را به بهترین شکل با نشان دادن این امر می‌توان واضح نمود که مهاجم می‌تواند بخش ابتدایی یک پیام را که می‌دانیم تعداد زیادی ۰ دنباله‌ای دارد، تعیین کند. برای سادگی، فرض کنید که $m = b \| \underbrace{0\cdots0}_{L}$ که در آن، $b \in$

$\{0,1\}^*$ نامعلوم است و m دارای بالاترین طول ممکن است (بنابراین $L = 8 \cdot (k - 11) - 1$). رمزگذاری m، متن رمز c را می‌دهد که در آن

$$c = (0x00 \| 0x02 \| r \| 0x00 \| b \| 0\cdots0)^e \bmod N.$$

مهاجم می‌تواند $c' = c/(2^L)^e \bmod N$ را محاسبه کند؛ توجه کنید که

$$c' = \left(\frac{0x00 \| 0x02 \| r \| 0x00 \| b \| 0\cdots0}{10\cdots0}\right)^e = (0x00 \| 0x02 \| r \| 0x00 \| b)^e \bmod N.$$

عدد صحیح $0x00 \| r \| 0x02 \| 0x00$ دارای طول ۷۵ بیت است (توجه کنید که $0x02 = 00000010$) و هیچ‌کدام از بیت‌های ۰ درجه بالا، شمارش نمی‌شوند)و بنابراین اکنون مهاجم می‌تواند «حمله‌ی پیام کوتاه» یا حمله‌ای بر اساس رمزگذاری پیامی که بخشی از آن معلوم است، از بخش قبلی، را اعمال کند. برای اجتناب از این حملات، باید r با طول حداقل $\|N\|/e$ را در نظر بگیریم. بااین‌حال، حتی اگر e بزرگ باشد، «حمله‌ی بهبود درجه دوم» از بخش قبلی نشان می‌دهد که r را می‌توان با احتمال بالا در زمان تقریباً $2^{\|r\|/2}$ بازیابی نمود.

اگر r را تقریباً برابر با نیمی از طول N قرار دهیم و به شکل متناظر طول ماکزیمم پیام را کاهش دهیم، آنگاه حدس اینکه طرح رمزگذاری در PKCS#1 v1.5 به‌صورت CPA-امن است، منطقی خواهد بود. (بااین‌حال، تأکید می‌کنیم که هیچ اثبات امنیتی بر اساس فرض RSA شناخته نشده است.) به‌هرحال، به دلیل یک حمله‌ی جدی متن رمز منتخب علیه این طرح، که به‌صورت مختصر در بخش ۱۱-۵-۵ توصیف‌شده است، نسخه‌های جدیدتر استاندارد PKCS#1 معرفی شده‌اند و باید به‌جای این نسخه استفاده شوند.

۱۱-۵-۳ * رمزگذاری CPA-امن بدون پیشگوهای تصادفی

در این بخش، یک طرح رمزگذاری را نشان می‌دهیم که CPA-امن بودن آن را می‌توان بر اساس فرض RSA اثبات نمود. ما با توصیف یک پایه‌ی سخت هسته‌ی خاص (ن.ک. بخش ۷-۱-۳) برای مسئله‌ی RSA بحث را شروع کرده و سپس نشان می‌دهیم که چگونه می‌توان از این پایه‌ی سخت هسته برای رمزگذاری یک بیت یکتا استفاده نمود. در ادامه، این طرح را تعمیم می‌دهیم تا یک KEM به دست آید.

طرح‌های توصیف‌شده در این بخش اصولاً دارای اهمیت نظری هستند و در عمل مورداستفاده قرار نمی‌گیرند. این امر به این دلیل است که این طرح‌ها، کارایی کمتری نسبت به دیگر ساختارهای مبتنی بر RSA دارند که می‌توان امنیت آن‌ها را در مدل پیشگوی تصادفی (ن.ک. بخش ۵-۵) اثبات نمود. ما مثال‌هایی از چنین طرح‌های رمزگذاری را در بخش‌های بعدی مشاهده می‌کنیم.

یک پایه‌ی سخت هسته برای مسئله‌ی RSA

به زبان ساده، فرض RSA بیان می‌کند که با داشتن N، e و $[x^e \bmod N]$ (برای x انتخاب شده به‌صورت یکنواخت از $\mathbb{Z}_N^*$)، بازیابی x، غیرعملی است. به‌خودی‌خود، این مطلب هیچ‌چیزی در مورد دشواری محاسباتی محاسبه‌ی اطلاعاتی خاص در مورد x بیان نمی‌کند. آیا می‌توانیم یک بیت خاص مفروض از اطلاعات پیرامون x را جداسازی نماییم که محاسبه‌ی آن از روی N، e و $[x^e \bmod N]$، سخت باشد؟ مفهوم یک «پایه‌ی سخت هسته» دقیقاً همین الزام را مشخص می‌کند. (پایه‌های سخت هسته در بخش ۷-۱-۳ معرفی شدند. این واقعیت که فرض RSA یک خانواده از جایگشت‌های یک‌طرفه را ارائه می‌کند در بخش ۸-۴-۱ موردبحث قرار گرفت. بااین‌حال، بررسی ما در اینجا به مطالب دیگر بستگی ندارد.) مشخص می‌شود که کم‌اهمیت‌ترین بیت x، نشان داده‌شده توسط $lsb(x)$، یک پایه‌ی سخت هسته برای مسئله‌ی RSA است.

آزمایش زیر را برای یک الگوریتم مفروض $GenRSA$ (با رفتار عادی) و الگوریتم $\mathcal{A}$ تعریف کنید:

آزمایش پایه‌ی سخت هسته‌ی RSA؛ $RSA - lsb_{\mathcal{A},GenRSA}(1^n)$:

۱) $GenRSA(1^n)$ را برای به دست آوردن (N, e, d) اجرا می‌کنیم.
۲) یک $x \in \mathbb{Z}_N^*$ یکنواخت را انتخاب کرده و $y := [x^e \bmod N]$ را محاسبه می‌کنیم.
۳) مقادیر N, e, y به $\mathcal{A}$ داده می‌شود و $\mathcal{A}$ یک بیت b را به‌عنوان خروجی ارائه می‌کند.
۴) خروجی آزمایش برابر ۱ است اگر و تنها اگر $lsb(x) = b$ باشد.

توجه کنید که $lsb(x)$ یک بیت یکنواخت است هنگامی‌که $x \in \mathbb{Z}_N^*$ یکنواخت باشد. $\mathcal{A}$ می‌تواند $lsb(x)$ را با احتمال ۱/۲ حدس بزند به این صورت که یک بیت یکنواخت b را به‌عنوان خروجی ارائه کند. قضیه‌ی زیر بیان می‌کند که اگر مسئله‌ی RSA سخت باشد، آنگاه هیچ الگوریتم کارای $\mathcal{A}$ نمی‌تواند عملکردی داشته باشد که به شکل چشم‌گیری بهتر از این عملکرد باشد؛ یعنی، کم‌اهمیت‌ترین بیت، یک پایه‌ی سخت هسته برای جایگشت RSA است.

قضیه‌ی ۱۱-۳۱

اگر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد، آنگاه برای تمام الگوریتم‌های زمان-چندجمله‌ای تصادفی $\mathcal{A}$، یک تابع ناچیز $negl$ وجود دارد به صورتی که $\Pr[RSA - lsb_{\mathcal{A},GenRSA}(n) = 1] \leq \frac{1}{2} + negl(n)$

یک اثبات کامل برای این قضیه، فراتر از گستره‌ی این کتاب است. بااین‌حال، برداشتی شهودی را برای این قضیه با نشان دادن یک اثبات برای یک نتیجه‌ی ضعیف‌تر به دست می‌دهیم: اینکه فرض RSA نشان می‌دهد که برای همه‌ی $\mathcal{A}$های زمان-چندجمله‌ای تصادفی، داریم $\Pr[RSA - lsb_{\mathcal{A},GenRSA}(n) = 1] < 1$ است. برای اثبات این امر، نشان می‌دهیم که یک الگوریتم کارا که

همواره به‌درستی $lsb(r)$ را از روی N، e و $[r^e \bmod N]$ محاسبه نماید را می‌توان برای بازیابی کارای x (به‌طور کامل) از روی N، e و $[x^e \bmod N]$ مورد استفاده قرار داد.

مقادیر N و e را در نظر گرفته و فرض کنید $\mathcal{A}$ الگوریتمی باشد به صورتی که $\mathcal{A}([r^e \bmod N]) = lsb(r)$ با داشتن N، e و $y = [x^e \bmod N]$، ما بیت‌های x را یک‌به‌یک از کم‌اهمیت‌ترین تا مهم‌ترین، بازیابی می‌کنیم. به‌منظور تعیین $lsb(x)$، الگوریتم $\mathcal{A}(y)$ را اجرا می‌کنیم. دو حالت وجود دارد:

حالت اول: $lsb(x) = 0$

توجه کنید که $y/2^e = (x/2)^e \bmod N$ و به این دلیل که x، زوج است (یعنی $lsb(x) = 0$)، عدد صحیح x بر ۲ بخش‌پذیر است. بنابراین $x/2$ درواقع انتقال بیتی x به سمت راست است و $lsb(x/2)$ برابر با $2sb(x)$ خواهد بود که دومین بیت کم‌اهمیت x است. بنابراین می‌توانیم $2sb(x)$ را با محاسبه‌ی $y' := [y/2^e \bmod N]$ و سپس اجرای $\mathcal{A}(y')$ به دست آوریم.

حالت دوم: $lsb(x) = 1$

در اینجا $[x/2 \bmod N] = (x + N)/2$. بنابراین، $lsb([x/2 \bmod N])$ برابر با $2sb(x + N)$ است؛ مورد دوم برابر با $2sb(N) \oplus 2sb(x) \oplus 1$ است (ما یک بیت حامل در جایگاه دوم خواهیم داشت به این دلیل که هم x و هم N، فرد هستند). بنابراین، اگر $y' := [y/2^e \bmod N]$ را محاسبه کنیم، آنگاه $2sb(x) = \mathcal{A}(y') \oplus 1 \oplus 2sb(N)$ است.

با ادامه‌ی این روند می‌توانیم تمام بیت‌های x را بازیابی کنیم.

رمزگذاری یک بیت

می‌توانیم از پایه‌ی سخت هسته‌ی شناسایی‌شده در بالا برای رمزگذاری یک بیت یکتا استفاده کنیم. این ایده، سرراست است: به‌منظور رمزگذاری یک پیام $m \in \{0,1\}$، فرستنده $r \in \mathbb{Z}_N^*$ یکنواخت را انتخاب می‌کند مشروط به اینکه $lsb(r) = m$ باشد. متن رمز برابر با $c := [r^e \bmod N]$ است. ن.ک. ساختار ۱۱-۳۲.

قضیه‌ی ۱۱-۳۳

اگر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد، آنگاه ساختار ۱۱-۳۲، CPA-امن است.

اثبات

فرض کنید Π نشان‌دهنده‌ی ساختار ۱۱-۳۲ باشد. اثبات می‌کنیم که Π دارای رمزگذاری‌های تمایزناپذیر در حضور یک شنودگر است؛ بر اساس گزاره‌ی ۱۱-۳، این امر به‌صورت ضمنی نشان

رمزگذاری تک‌بیتی با استفاده از یک پایه‌ی سخت هسته برای RSA

ساختار ۱۱-۳۲

فرض کنید $GenRSA$ به شکل معمول باشد و یک طرح رمزگذاری کلید عمومی را به‌صورت زیر تعریف کنید:

- Gen: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را برای به دست آوردن (N, e, d) اجرا می‌کنیم. کلید عمومی $pk = \langle N, e \rangle$ و کلید خصوصی $sk = \langle N, d \rangle$ را به‌عنوان خروجی ارائه می‌کنیم.
- Enc: با دریافت یک کلید عمومی $pk = \langle N, e \rangle$ و یک پیام $m \in \{0,1\}$ به‌عنوان ورودی، یک $r \in \mathbb{Z}_N^*$ یکنواخت را مشروط به $lsb(r) = m$، انتخاب می‌کنیم. متن رمز $c := [r^e \bmod N]$ را به‌عنوان خروجی ارائه می‌دهیم.
- Dec: با دریافت یک کلید خصوصی $sk = \langle N, d \rangle$ و یک متن رمز c به‌عنوان ورودی، $r := [c^d \bmod N]$ را محاسبه کرده و مقدار $lsb(r)$ را به‌عنوان خروجی ارائه می‌کنیم.

می‌دهد که این ساختار، CPA-امن است.

فرض کنید $\mathcal{A}$ یک متخاصم زمان‌چندجمله‌ای تصادفی باشد. بدون کاستن از کلیت، می‌توانیم فرض کنیم $m_0 = 0$ و $m_1 = 1$ در آزمایش $PubK^{eav}_{\mathcal{A},\Pi}(n)$. بنابراین

$$\Pr[PubK^{eav}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}.\Pr[\mathcal{A}(N,e,c) = 0 | c \text{ is an encryption of } 0] + \frac{1}{2}.\Pr[\mathcal{A}(N,e,c) = 1 | c \text{ is an encryption of } 1].$$

اجرای $\mathcal{A}$ در آزمایش $RSA - lsb$ را در نظر بگیرید. بر اساس تعریف،

$$\Pr[RSA - lsb_{\mathcal{A},GenRSA}(n) = 1] = \Pr[\mathcal{A}(N,e,[r^e \bmod N]) = lsb(r)],$$

که در آن، r در $\mathbb{Z}_N^*$ یکنواخت است. ازآنجاکه $\Pr[lsb(r) = 1] = 1/2$، داریم

$$\Pr[RSA - lsb_{\mathcal{A},GenRSA}(n) = 1] = \frac{1}{2}.\Pr[\mathcal{A}(N,e,[r^e \bmod N]) = 0 | lsb(r) = 0] + \frac{1}{2}.\Pr[(N,e,[r^e \bmod N]) = 1 | lsb(r) = 1].$$

با توجه به اینکه رمزگذاری $m \in \{0,1\}$ دقیقاً متناظر با انتخاب r یکنواخت مشروط به $lsb(r) = m$ است، مشاهده می‌کنیم که

$$\Pr[PubK^{eav}_{\mathcal{A},\Pi}(n) = 1] = \Pr[RSA - lsb_{\mathcal{A},GenRSA}(n) = 1].$$

بنابراین، قضیه‌ی ۱۱-۳۱ نشان می‌دهد که یک تابع ناچیز $negl$ وجود دارد به صورتی که

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

همان‌طور که مطلوب است. ■

ساختن یک KEM

اکنون نشان می‌دهیم که چگونه می‌توان ساختار ۱۱-۳۲ را تعمیم داد تا یک KEM با طول کلید n به دست آید. یک روش بیش‌ازحد ساده برای انجام این کار عبارت خواهد بود از انتخاب یک کلید n-بیتی یکنواخت k و سپس رمزگذاری بیت‌های k به‌صورت یک‌به‌یک با استفاده از n فراخوان از ساختار ۱۱-۳۲. این روند منجر به یک متن رمز نسبتاً طولانی شامل n عنصر از $\mathbb{Z}_N$ خواهد شد.

یک رویکرد بهتر این است که فرستنده، جایگشت RSA (یعنی به توان eام رساندن به پیمانه‌ی N) را به‌صورت مکرر با شروع از یک مقدار یکنواخت اولیه‌ی c_1، اعمال نماید. یعنی، فرستنده به‌صورت متوالی، c_2^e، سپس $(c_2^e)^e = c_2^{e^2}$ و الی‌آخر تا $c_1^{e^n}$ (همگی به پیمانه‌ی N) را محاسبه می‌کند. آخرین مقدار $[c_1^{e^n} \bmod N]$ همان متن رمز خواهد بود و دنباله‌ی بیت‌های $lsb(c_1), lsb(c_2^e), \ldots, lsb\left(c_1^{e^{n-1}}\right)$ همان کلید خواهد بود. به‌منظور رمزگشایی یک متن رمز c، گیرنده این روند را وارون می‌کند و به‌صورت متوالی c^d، $(c^d)^d = c^{d^2}$ تا c^{d^n} را محاسبه خواهد نمود (یادآوری، همگی به پیمانه‌ی N) تا مقدار اولیه‌ی $c_1 = c^{d^n}$ استفاده‌شده توسط فرستنده، بازیابی شود. بعد از بازیابی c_1، گیرنده می‌تواند $c_2^e, \ldots, c_1^{e^n}$ را دوباره محاسبه نموده و کلید را به دست آورد.

می‌توان رمزگشایی را به شکلی کاراتر با استفاده از این واقعیت پیاده‌سازی نمود که گیرنده مرتبه‌ی گروه $\mathbb{Z}_N^*$ را می‌داند. در زمان تولید کلید، گیرنده می‌تواند $d' := [d^n \bmod \phi(N)]$ را پیش‌محاسبه نماید و d' را به‌عنوان بخشی از کلید خصوصی خود، ذخیره کند. سپس، به‌جای محاسبه‌ی n توان رسانی پیمانه‌ای متوالی $c^d, c^{d^2}, \ldots$ برای به دست آوردن c_1، گیرنده می‌تواند به‌صورت مستقیم $c_1 := [c^{d'} \bmod N]$ را محاسبه کند. مشخص است که این روش به این دلیل مثمر ثمر است که

$$c^{d^n} \bmod N = c^{[d^n \bmod \phi(N)]} = c^{d'} \bmod N.$$

مطلب بالا به‌صورت رسمی به‌عنوان ساختار ۱۱-۳۴ توصیف‌شده است.

این ساختار مشابه با رویکرد استفاده‌شده برای ساختن یک مولد شبه تصادفی از روی یک جایگشت یک‌طرفه در انتهای بخش ۷-۴-۲ است. اگر فرض کنیم f نشان‌دهنده‌ی جایگشت RSA نسبت به یک کلید عمومی $\langle N, e \rangle$ مفروض باشد (یعنی $f(x) \stackrel{def}{=} [x^e \bmod N]$)، آنگاه CPA-امن بودن ساختار ۱۱-۳۴ برابر با شبه تصادفی بودن $lsb(f^{n-1}(c_1)), \ldots, lsb(c_1)$ است حتی مشروط به

مقدار $c = f^n(c_1)$. به‌نوبه‌ی خود، این مقوله را می‌توان با استفاده از قضیه‌ی ۱۱-۳۱ و روش‌های ارائه شده در بخش ۷-۴-۲، اثبات نمود. (تنها تفاوت آن است که در بخش ۷-۴-۷، خود مقدار $f^n(c_1)$ یک رشته‌ی n-بیتی یکنواخت بود درحالی‌که در اینجا، این مقدار یک عنصر یکنواخت از $\mathbb{Z}_N^*$ است. شبه تصادفی بودن پایه‌های سخت هسته‌ی متوالی، مستقل از دامنه‌ی f است.) به‌طور خلاصه:

قضیه‌ی ۱۱-۳۵

اگر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد، آنگاه ساختار ۱۱-۳۴ یک KEM دارای امنیت CPA است.

یک KEM با استفاده از یک پایه‌ی سخت هسته برای RSA

ساختار ۱۱-۳۴

فرض می‌کنیم $GenRSA$ به شکل عادی باشد و یک KEM را به‌صورت زیر تعریف می‌کنیم:

- *Gen*: با دریافت 1^n به‌عنوان ورودی، $(N, e, d) \leftarrow GenRSA(1^n)$ را اجرا می‌کنیم تا (N, e, d) به دست آید. سپس، $d' := [d^n \bmod \phi(N)]$ را محاسبه می‌کنیم (توجه کنید که $\phi(N)$ را می‌توان از روی (N, e, d) محاسبه نمود یا طی اجرای $GenRSA$، آن را به دست آورد). مقادیر $pk = \langle N, e \rangle$ و $sk = \langle N, d' \rangle$ را به‌عنوان خروجی ارائه می‌دهیم.
- *Encaps*: با دریافت $pk = \langle N, e \rangle$ و 1^n به‌عنوان ورودی، یک $c_1 \in \mathbb{Z}_N^*$ یکنواخت را انتخاب می‌کنیم. سپس برای $i = 1, \ldots, n$، موارد زیر را انجام می‌دهیم:
  (۱) محاسبه‌ی $k_i := \mathrm{lsb}(c_i)$.
  (۲) محاسبه‌ی $c_{i+1} := [c_i^e \bmod N]$.
  متن رمز c_{n+1} و کلید $k = k_1 \cdots k_n$ را به‌عنوان خروجی ارائه می‌کنیم.
- *Decaps*: با دریافت $sk = \langle N, d' \rangle$ و یک متن رمز c به‌عنوان ورودی، $c_1 := [c^{d'} \bmod N]$ را محاسبه می‌کنیم. سپس برای $i = 1, \ldots, n$، موارد زیر را انجام می‌دهیم:
  (۱) محاسبه‌ی $k_i := \mathrm{lsb}(c_i)$.
  (۲) محاسبه‌ی $c_{i+1} := [c_i^e \bmod N]$.
  کلید $k = k_1 \cdots k_n$ را به‌عنوان خروجی ارائه می‌کنیم.

کارایی

ساختار ۱۱-۳۴ به شکل منطقی، کارا است. به بیان عینی، فرض کنید که $n = 128$ باشد، پیمانه‌ی RSA یعنی N دارای طول ۲۰۴۸ بیت باشد و توان عمومی e برابر با ۳ است به صورتی که توان رسانی به توان e به پیمانه‌ی N را بتوان با استفاده از دو ضرب پیمانه‌ای، محاسبه نمود. (ن.ک. ضمیمه‌ی ب-۲-۳.) آنگاه، رمزگذاری نیازمند $2n = 256$ ضرب پیمانه‌ای است. رمزگشایی را می‌توان با استفاده از یک توان رسانی پیمانه‌ای کامل (با هزینه‌ی تقریبی $1.5 \cdot 2048 = 3072$ ضرب پیمانه‌ای)

به‌پیمانه‌ی ۲۵۶ ضرب پیمانه‌ای دیگر، انجام داد. بنابراین، هزینه‌ی رمزگشایی تنها در حدود هشت مرتبه کمتر از این هزینه برای طرح رمزگذاری RSA ساده است. در مقابل، رمزگذاری بسیار گران‌تر از رمزگذاری در RSA ساده است ولی در بسیاری از کاربردها، زمان رمزگشایی بسیار حیاتی‌تر است (از آنجاکه ممکن است توسط کارگذاری پیاده‌سازی شود که در حال اجرای هزاران رمزگشایی به‌صورت هم‌زمان است).

۱۱-۴-۵-۲ $OAEP$ و $RSA\ PKCS\ \#1\ v2.0$

تاکنون طرح‌های رمزگذاری مبتنی بر RSA را که CCA-امن هستند، مدنظر قرار نداده‌ایم. در ابتدا نشان می‌دهیم که تمامی طرح‌های رمزگذاری مبتنی بر RSA که تاکنون مشاهده کردیم در برابر حملات متن رمز منتخب، آسیب‌پذیر هستند.

رمزگذاری RSA ساده

RSA ساده حتی CPA امن هم نیست، ولی تضمین می‌کند که اگر $m \in \mathbb{Z}_N^*$ یکنواخت باشد، آنگاه مهاجمی که رمزگذاری $c = [m^e \bmod N]$ از m را نسبت به کلید عمومی $\langle N, e \rangle$، شنود می‌کند می‌تواند m را بازیابی کند. حتی این تضمین ضعیف هم در موقعیتی که در آن، حملات متن رمز منتخب، ممکن باشند، دیگر برقرار نخواهد بود. به‌مانند وضعیت رمزگذاری الجمال، این امر نتیجه‌ی این واقعیت است که RSA ساده، «شکل‌پذیر» است: با داشتن رمزگذاری $c = [m^e \bmod N]$ از یک پیام نامعلوم m، به‌سادگی می‌توان یک متن رمز c' را تولید نمود که یک رمزگذاری از $[2m \bmod N]$ باشد به این صورت که قرار دهیم

$$c' := [2^e \cdot c \bmod N]$$
$$= 2^e \cdot m^e = (2m)^e \bmod N.$$

$RSA\ PKCS\ \#1\ v1.5$

رمزگذاری RSA لایی گذاری شده که حدس می‌زنیم با تنظیم صحیح پارامترها، CPA-امن باشد، لزوماً در برابر همین شکل از حمله به‌مانند رمزگذاری RSA ساده، آسیب‌پذیر خواهد بود. ولی همچنین یک حمله‌ی جالب‌تر متن رمز منتخب علیه رمزگذاری $PKCS\ \#1\ v1.5$ وجود دارد که برخلاف حمله‌ی ارائه شده در بالا، نیازمند دسترسی کامل به یک پیشگوی رمزگشایی نیست؛ این حمله تنها نیازمند دسترسی به یک پیشگوی رمزگشایی «جزئی» است که مشخص می‌کند آیا رمزگشایی یک متن رمز مفروض، یک خطا را باز خواهد گرداند یا خیر. این امر باعث می‌شود که این حمله بسیار عملی‌تر باشد چراکه این حمله را می‌توان در هرزمانی انجام داد که مهاجم بتواند بین رفتار گیرنده‌ی مد از یک رمزگشایی موفق با رفتار آن بعد از یک رمزگشایی ناموفق، تمایز قائل شود، به‌مانند حالت حمله‌ی مبتنی بر پیشگوی لایی گذاری که در بخش ۳-۷-۲ نشان داده شده است.

یادآوری می‌کنیم که طرح رمزگذاری کلید عمومی تعریف‌شده در استاندارد PKCS #1 v1.5 از یک نوع رمزگذاری RSA لایی گذاری شده استفاده می‌کند که در آن، لایی گذاری به یک شیوه‌ی خاص انجام می‌شود. علی‌الخصوص، دو بایت درجه بالای پیام لایی گذاری شده، همواره برابر با $0x00$ $0x02$ هستند. هنگام رمزگشایی، گیرنده باید بررسی کند که آیا دو بایت درجه بالا با این مقادیر منطبق هستند یا خیر و درصورتی‌که منطبق نباشند باید یک خطا را بازگرداند. در سال ۱۹۹۸، بلایکنباکر (Bleichenbacher) یک حمله‌ی متن رمز منتخب را توسعه داد که متناظر با یک رمزگذاری صادقانه از یک پیام نامعلوم مفروض m نسبت به یک کلید عمومی $\langle N, e \rangle$ است. این حمله به‌صورت مکرر، $s \in \mathbb{Z}_N^*$ یکنواخت را انتخاب کرده و متن رمز $c' := [s^e \cdot c \bmod N]$ را برای گیرنده ارسال می‌کند. فرض کنید $c = [\hat{m}^e \bmod N]$ که در آن

$$\hat{m} = 0x00 \| 0x02 \| r \| 0x00 \| m,$$

همان‌طور که توسط PKCS #1 v1.5 مشخص شده است. سپس، رمزگشایی c' نتیجه‌ی میانی $\hat{m}' = [s.\hat{m} \bmod N]$ را می‌دهد و گیرنده یک خطا را بازمی‌گرداند مگر اینکه دو بایت بالایی $\hat{m}'$ دقیقاً برابر با $0x02 \| 0x00$ باشند. (بررسی‌های دیگری هم انجام می‌شوند ولی آن‌ها را برای سادگی نادیده گرفته‌ایم.) هر بار که رمزگشایی موفق می‌شود، مهاجم می‌فهمد که دو بایت بالایی $s.\hat{m} \bmod N$ برابر با $0x02 \| 0x00$ هستند که در آن، s معلوم است. تعدادی از معادلات ازاین‌دست برای مهاجم کافی خواهد بود تا $\hat{m}$ را بفهمد و کل متن اولیه‌ی m را بازیابی کند.

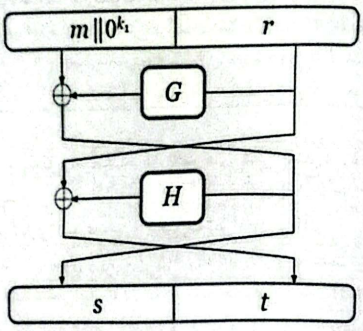
KEM دارای امنیت CPA

در بخش ۱۱-۵-۳، ساختاری از یک KEM را نشان دادیم که می‌توان CPA-امن بودن آن را بر اساس فرض RSA، اثبات نمود. آن ساختار نیز در برابر یک حمله‌ی متن رمز منتخب، ناامن است؛ جزئیات را به‌عنوان یک تمرین باقی می گذاریم.

$RSA - OAEP$

در این بخش، ساختاری از رمزگذاری CCA-امن مبتنی بر RSA را با استفاده از «لایی گذاری رمزگذاری نامتقارن بهینه» (OAEP) بررسی می‌کنیم. طرح $RSA - OAEP$ به‌دست‌آمده این ایده‌ی انتخاب یک پیام m، تبدیل آن به یک عنصر $\hat{m} \in \mathbb{Z}_N^*$ و سپس قرار دادن $c = [\hat{m}^e \bmod N]$ به‌عنوان متن رمز، تبعیت می‌کند (که همچنین در بخش ۱۱-۵-۲ هم استفاده شد). بااین‌حال، تبدیل در اینجا پیچیده‌تر از قبل است. یک نسخه از $RSA - OAEP$ به‌عنوان بخشی از RSA PKCS #1 از نسخه‌ی ۲.۰ استانداردسازی شده است.

فرض کنید $\ell(n), k_0(n), k_1(n) = \Theta(n)$ توابع مقدار صحیح با $k_0(n)$ و به‌گونه‌ای باشند که $\ell(n) + k_0(n) + k_1(n)$ کمتر از طول بیتی مینیمم پیمانه‌های خروجی توسط $GenRSA(1^n)$



تصویر ۱۱-۴: مکانیسم $OAEP$

باشد. n را در نظر گرفته و فرض کنید $\ell = \ell(n)$، $k_0 = k_0(n)$ و $k_1 = k_1(n)$ باشند. فرض می‌کنیم $G: \{0,1\}^{k_0} \to \{0,1\}^{\ell+k_1}$ و $H: \{0,1\}^{\ell+k_1} \to \{0,1\}^{k_0}$ دو تابع چکیده‌سازی باشند که به‌عنوان پیشگوهای تصادفی مستقل، مدل‌سازی خواهند شد. (هرچند استفاده از بیش از یک پیشگوی تصادفی در بخش ۱۱-۵-۵ موردبحث قرار نگرفت، ولی این شیوه به‌صورت طبیعی تعمیر می‌شود.) تبدیل تعریف‌شده توسط $OAEP$ یک شبکه‌ی فیستلی دو دوری با G و H به‌عنوان توابع دوری است؛ ن یک تصویر ۱۱-۴. با جزئیات دقیق، لایی گذاری یک پیام $m \in \{0,1\}^{\ell}$ به‌صورت زیر انجام می‌شود: در ابتدا قرار می‌دهیم $m' := m \| 0^{k_1}$ و یک $r \in \{0,1\}^{k_0}$ یکنواخت را انتخاب می‌کنیم. سپس موارد زیر را محاسبه کرده

$$s := m' \oplus G(r) \in \{0,1\}^{\ell+k_1}, t := r \oplus H(s) \in \{0,1\}^{k_0},$$

و قرار می‌دهیم $\hat{m} := s \| t$.

به‌منظور رمزگذاری یک پیام m با توجه به کلید عمومی $\langle N, e \rangle$، فرستنده $\hat{m}$ را به شکل بالا تولید کرده و متن رمز $c := \hat{m}^e \bmod N$ را به‌عنوان خروجی ارائه می‌کند. (توجه کنید که $\hat{m}$ تعبیر شده به‌عنوان یک عدد صحیح، به دلیل محدودیت‌های لحاظ شده بر ℓ, k_0, k_1، کمتر از N است.) برای رمزگشایی، گیرنده $\hat{m} := [c^d \bmod N]$ را محاسبه کرده و قرار می‌دهد $s \| t = \hat{m}$ که در آن s و t دارای طول‌های مناسب هستند. در ادامه، شبکه‌ی فیستلی را با محاسبه‌ی $r := H(s) \oplus t$ و $m' := G(r) \oplus s$، وارون می‌کند. نکته‌ی مهم این است که در ادامه، گیرنده این مسئله را وارسی می‌کند که k_1 بیت انتهایی m' همگی برابر با ۰ باشند؛ در غیر این صورت، متن رمز، رد خواهد شد و یک

پیام خطا بازگردانده می‌شود. در غیر این صورت، k_1 عدد ۰ کم‌اهمیت از m' حذف می‌شوند و l بیت باقیمانده‌ی m' به‌عنوان پیام، اعلام می‌شوند. این فرآیند در ساختار ۱۱-۳۶ توصیف‌شده است.

طرح رمزگذاری $RSA-OAEP$

ساختار ۱۱-۳۶

فرض کنید $GenRSA$ به شکل تعریف‌شده در بخش‌های قبلی باشد و ℓ, k_0, k_1 به‌صورت تعریف‌شده در متن باشند. فرض کنید $G:\{0,1\}^{k_0} \to \{0,1\}^{\ell+k_1}$ و $H:\{0,1\}^{\ell+k_1} \to \{0,1\}^{k_0}$ دو تابع باشند. یک طرح رمزگذاری کلید عمومی را به‌صورت زیر ایجاد می‌کنیم:

- Gen: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را برای به دست آوردن (N,e,d) اجرا می‌کنیم. کلید عمومی برابر با $\langle N,e\rangle$ و کلید خصوصی برابر با $\langle N,d\rangle$ است.
- Enc: با دریافت یک کلید عمومی $\langle N,e\rangle$ و یک پیام $m \in \{0,1\}^{\ell}$ به‌عنوان ورودی، قرار می‌دهیم $m' := m \| 0^{k_1}$ و یک $r \in \{0,1\}^{k_0}$ یکنواخت را انتخاب می‌کنیم. سپس، موارد زیر را محاسبه کرده
- $s := m' \oplus G(r), t := r \oplus H(s)$
- و قرار می‌دهیم $\hat{m} := s \| t$. متن رمز $c := [\hat{m}^e \bmod N]$ را به‌عنوان خروجی ارائه می‌کنیم.
- Dec: با دریافت یک کلید خصوصی $\langle N,d\rangle$ و یک متن رمز $c \in \mathbb{Z}_N^*$ به‌عنوان ورودی، $\hat{m} = [c^d \bmod N]$ را محاسبه می‌کنیم. اگر $\| \hat{m} \| < \ell + k_0 + k_1$، آنگاه خروجی $\perp$ را ارائه می‌کنیم. در غیر این صورت، $\hat{m}$ را به‌صورت $s \| t$ با $s \in \{0,1\}^{\ell+k_1}$ و $t \in \{0,1\}^{k_0}$ تجزیه می‌کنیم. $r := H(s) \oplus t$ و $m' := G(r) \oplus s$ را محاسبه می‌کنیم. اگر کم‌اهمیت‌ترین k_1 بیت در m' همگی برابر با ۰ نباشند، خروجی $\perp$ را ارائه می‌کنیم. در غیر این صورت، مهم‌ترین l بیت $\hat{m}$ را به‌عنوان خروجی ارائه می‌کنیم.

اثبات CCA-امن بودن برای $RSA-OAEP$ نسبتاً پیچیده است و ما آن را در اینجا ارائه نمی‌کنیم. ما تنها برداشت‌های شهودی را ارائه خواهیم کرد. در ابتدا، امنیت CPA را در نظر بگیرید. طی رمزگذاری، فرستنده موارد زیر

$$m' := m \| 0^{k_1}, s := m' \oplus G(r), t := r \oplus H(s)$$

را برای r یکنواخت محاسبه می‌کند؛ متن رمز برابر با $[(s \| t)^e \bmod N]$ خواهد بود. اگر مهاجم هیچ‌گاه r را از G پرسمان نکند، آنگاه ازآنجاکه G را به‌عنوان یک تابع تصادفی مدل‌سازی می‌کنیم، مقدار $G(r)$ از دیدگاه مهاجم، یکنواخت خواهد بود و بنابراین m با یک رشته‌ی یکنواخت درست مانند حالت طرح رمزگذاری کلید یک‌بارمصرف، پوشانده می‌شود. بنابراین، اگر مهاجم هیچ‌گاه r را از G پرسمان نکند، آنگاه هیچ اطلاعاتی در مورد پیام، افشا نخواهد شد.

مهاجم می‌تواند r را از G پرسمان کند؟ توجه کنید که خود مقدار r توسط $H(s)$ پوشیده می‌شود. بنابراین، مهاجم هیچ اطلاعاتی در مورد r ندارد مگر اینکه ابتدا s را از H پرسمان کند. اگر مهاجم، s را از H پرسمان نکند، آنگاه مهاجم ممکن است خوش‌شانس بوده و r را حدس بزند، ولی اگر ما طول r (یعنی k_0) را به‌اندازه‌ی کافی بلند در نظر بگیریم، آنگاه احتمال این حدس، ناچیز خواهد بود.

بنابراین، تنها راه مهاجم برای فهمیدن چیزی در مورد m آن است که در ابتدا s را از H پرسمان کند. این امر مهاجم را وادار می‌کند که s را از روی متن رمز (یکنواخت) $[(s \| t)^e \bmod N]$ محاسبه کند. توجه کنید که محاسبه‌ی s از روی $[(s \| t)^e \bmod N]$ برابر با مسئله‌ی RSA نیست که درواقع معادل با محاسبه‌ی s و همین طور t است. بااین‌حال، با تنظیم صحیح پارامترها، می‌توانیم از قضیه‌ی ۱۱-۲۹ برای نشان دادن این مسئله استفاده کنیم که s ازلحاظ محاسباتی غیرعملی خواهد بود اگر مسئله‌ی RSA سخت باشد.

ارائه‌ی استدلال برای CCA-امن بودن شامل پیچیدگی‌های بیشتری است ولی ایده‌ی اصلی عبارت است از نشان دادن اینکه هر پرسمان c از پیشگوی رمزگشایی انجام‌شده توسط مهاجم در یکی از این دو دسته قرار می‌گیرد: یا مهاجم c را با رمزگذاری قانونی یک پیام مفروض m به دست می‌آورد (که در این حالت، مهاجم هیچ‌چیزی از پرسمان رمزگشایی نخواهد فهمید) یا اینکه رمزگشایی c باعث برگشت خطا خواهد شد. این امر نتیجه‌ی این واقعیت است که گیرنده طی رمزگشایی، بررسی می‌کند که آیا k_1 بیت درجه پایین در $\hat{m}$ برابر با ۰ هستند یا خیر؛ اگر مهاجم یک متن رمز c مفروض را با رمزگذاری قانونی یک پیام مفروض ایجاد نکند، احتمال اینکه این شرط برقرار باشد، ناچیز است. اثبات رسمی به این دلیل پیچیده می‌شود که پرسمان‌های پیشگوی رمزگشایی مهاجم باید به‌درستی بدون داشتن کلید خصوصی، پاسخ داده شوند که به این معنا است که باید یک روش کارا برای تعیین این مسئله وجود داشته باشد که آیا باید یک خطا را بازگرداند یا خیر و اگر خیر، چه پیامی را باید بازگرداند. این امر با نگاه کردن به پرسمان‌های متخاصم از پیشگوهای تصادفی G, H انجام می‌شود.

حمله‌ی متن رمز منتخب منگر علیه $PKCS\#1\ v2.0$

در سال ۲۰۰۱، جیمز منگر یک حمله‌ی متن رمز منتخب علیه بعضی پیاده‌سازی‌های خاص از طرح رمزگذاری RSA مشخص شده در $PKCS\#1\ v2.0$ نشان داد-هرچند چیزی که مشخص شده بود یک نمونه از $RSA-OAEP$ بود! ازآنجاکه ساختار ۱۱-۳۶، CCA-امن است (با فرض اینکه مسئله‌ی RSA سخت است)، این امر چگونه امکان پذیر است؟

در بررسی الگوریتم رمزگشایی در ساختار ۱۱-۳۶، توجه کنید که دو شیوه برای رخداد خطا وجود دارد یا $\hat{m} \in \mathbb{Z}_N^*$ بیش‌ازحد بزرگ است یا $m' \in \{0,1\}^{l+k_1}$ دارای تعداد کافی ۰ انتهایی نیست. در ساختار ۱۱-۳۶، گیرنده باید در هر حالت، خطای «یکسان» (نشان داده شده با $\perp$) را ارائه کند. بااین‌حال، در بعضی پیاده‌سازی‌ها، گیرنده خطاهای «متفاوتی» را بسته به اینکه کدام گام شکست

می‌خورد، خروجی خواهد داد. این تک بیت اطلاعات اضافی مهاجم را قادر می‌سازد تا یک حمله‌ی متن رمز منتخب را سامان دهد که کل یک پیام m را از روی یک رمزگذاری از آن پیام تنها با استفاده از حدود $N \parallel$ پرسمان از یک پیشگو بازیابی نماید که بعد از رمزگشایی، پیام خطا را افشا می‌کند. این امر نشان‌دهنده‌ی اهمیت پیاده‌سازی طرح‌های رمزنگاری دقیقاً به‌صورت مشخص شده را نشان می‌دهد چراکه اثبات و تحلیل به‌دست‌آمده ممکن است دیگر قابل‌اعمال نباشند اگر بعضی از ابعاد طرح تغییر یافته باشند.

حتی اگر خطای یکسانی در هر دو حالت ارائه شود، مهاجم می‌تواند تعیین کند که خطا در چه جایی رخ داده است اگر «زمان» بازگرداندن پیام خطا، متفاوت باشد. (این مطلب یک مثال عالی از این امر است که مهاجم محدود به بررسی ورودی‌ها/خروجی‌های یک الگوریتم نیست بلکه می‌تواند از «اطلاعات کانال جانبی» برای حمله به یک طرح استفاده کند.) پیاده‌سازی‌ها باید به‌دقت انجام شوند تا تضمین شود که زمان ارائه‌ی یک پیام خطا بدون توجه به محل رخداد خطا، یکسان خواهد بود.

۱۱-۵-۵ یک KEM دارای امنیت CCA در مدل پیشگوی تصادفی

در اینجا ساختاری از یک KEM مبتنی بر RSA را نشان می‌دهیم که در مدل پیشگوی تصادفی، CCA-امن است. (از قضیه‌ی ۱۱-۱۴ به یاد داریم که چنین ساختاری را می‌توان همراه با هر طرح رمزگذاری کلید خصوصی CCA-امنی مورداستفاده قرار داد تا یک طرح رمزگذاری کلید عمومی CCA-امن به دست آید.) در مقایسه با طرح $RSA-OAEP$ از بخش قبلی، اصلی‌ترین مزیت در اینجا عبارت است از سادگی ساختار و سادگی اثبات امنیت آن. اصلی‌ترین عیب آن این است که منجر به متن رمزهای طولانی‌تر هنگام رمزگذاری پیام‌های کوتاه می‌شود چراکه نیازمند الگوی KEM/DEM است درحالی‌که $RSA-OAEP$ این‌گونه نیست. بااین‌حال، برای رمزگذاری پیام‌های طولانی، $RSA-OAEP$ همچنین به‌عنوان بخشی از یک طرح رمزگذاری ترکیبی استفاده می‌شود و منجر به یک طرح رمزگذاری دارای کارایی مشابه با طرح قابل‌دستیابی با استفاده از KEM نشان داده‌شده در اینجا، خواهد شد.

KEM که توصیف می‌کنیم به‌عنوان بخشی از استاندارد ISO/IEC ۱۸۰۳۳ – ۲ برای رمزگذاری کلید عمومی، در نظر گرفته شده است. در این طرح، کلید عمومی شامل $\langle N, e\rangle$ به شکل معمول است و یک تابع $H: \mathbb{Z}_N^* \to \{0,1\}^n$ مشخص می‌شود که به‌عنوان یک پیشگوی تصادفی در تحلیل ما، مدل‌سازی می‌شود. (این تابع می‌تواند بر اساس تعدادی تابع چکیده ساز رمزنگاری مبنا به‌صورت بحث شده در بخش ۵-۵ باشد. ما جزئیات را حذف می‌کنیم.) برای لفافه بندی یک کلید، فرستنده $r \in \mathbb{Z}_N^*$ یکنواخت را انتخاب کرده و سپس متن رمز $c := [r^e \bmod N]$ و کلید $k := H(r)$ را محاسبه می‌کند. به‌منظور رمزگشایی یک متن رمز c، گیرنده تنها r را به شکل معمول بازیابی کرده و سپس، کلید یکسان $k := H(r)$ را دوباره به دست می‌آورد. ن.ک. ساختار ۱۱-۳۷.

ی KEM دارای امنیت CCA (در مدل پیشگوی تصادفی)

ساختار ۱۱-۳۷

روی کنید $GenRSA$ به شکل معمول باشد و یک KEM را به‌صورت زیر بسازید:

- Gen: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را اجرا می‌کنیم تا (N, e, d) به دست آید. کلید عمومی برابر با $\langle N, e\rangle$ و کلید خصوصی برابر با $\langle N, d\rangle$ خواهد بود.
  به‌عنوان بخشی از تولید کلید، یک تابع $H: \mathbb{Z}_N^* \to \{0,1\}^n$ مشخص می‌شود ولی ما این امر را به شکل ضمنی باقی می‌گذاریم.
- Enc: با دریافت کلید عمومی $\langle N, e\rangle$ و 1^n به‌عنوان ورودی، یک $r \in \mathbb{Z}_N^*$ یکنواخت را انتخاب می‌کنیم. متن رمز $c := [r^e \bmod N]$ و کلید $k := H(r)$ را به‌عنوان خروجی ارائه می‌کنیم.
- $Decaps$: با دریافت کلید خصوصی $\langle N, d\rangle$ و یک متن رمز $c \in \mathbb{Z}_N^*$ مقدار $r := [c^d \bmod N]$ را محاسبه کرده و کلید $k := H(r)$ را به‌عنوان خروجی ارائه می‌کنیم.

CPA-امن بودن این طرح، به شکل فوری مشخص می‌شود. درواقع، متن رمز c برای $r \in \mathbb{Z}_N^*$ یکنواخت و برابر با $[r^e \bmod N]$ خواهد بود و بنابراین، فرض RSA نشان می‌دهد که شنودگرای که c را مشاهده می‌کند قادر به محاسبه‌ی r نخواهد بود. این نیز به‌نوبه‌ی خود به این معنا است که شنودگر، r را از پرسمان نمی‌کند و بنابراین مقدار کلید $k \stackrel{\text{def}}{=} H(r)$ از دیدگاه مهاجم، یکنواخت باقی خواهد ماند.

درواقع، مطلب بالا تعمیم می‌یابد تا CCA-امن بودن را نیز نشان دهد. این امر به این دلیل است که پاسخ یک پرسمان پیشگوی لفافه زدایی برای هر متن رمز $\tilde{c} \neq c$ تنها شامل ارزیابی H در یک ورودی مفروض $\tilde{r} = [\tilde{c}^d \bmod N] \neq r$ است. بنابراین، پرسمان‌های مهاجم از پیشگوی لفافه زدایی هیچ اطلاعات اضافی در مورد کلید $H(r)$ لفافه بندی شده توسط متن رمز چالش را افشا نمی‌کند. (یک اثبات رسمی اندکی پیشرفته‌تر است چراکه ما باید نشان دهیم که چگونه می‌توان پاسخ‌ها به پرسمان‌های پیشگوی لفافه زدایی را «شبیه‌سازی» نمود بدون اینکه کلید خصوصی را بدانیم. بااین‌حال، مشخص می‌شود که این امر خیلی دشوار نیست.)

قضیه‌ی ۱۱-۳۸

اگر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد و H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شود آنگاه ساختار ۱۱-۳۷، CCA-امن خواهد بود.

اثبات

فرض کنید Π نشان‌دهنده‌ی ساختار ۱۱-۳۷ بوده و $\mathcal{A}$ یک متخاصم زمان-چندجمله‌ای تصادفی باشد. برای راحتی بیشتر و به این دلیل که این اولین اثباتی است که در آن از قدرت کامل مدل پیشگوی تصادفی بهره می‌بریم، به‌صورت آشکار گام‌های آزمایش $KEM^{cca}_{\mathcal{A},\Pi}(n)$ را توصیف می‌کنیم:

۱) (N,e,d) اجرا می‌شود تا $GenRSA(1^n)$ به دست آید. به‌علاوه، یک تابع تصادفی $H:\mathbb{Z}^*_N \to \{0,1\}^n$ انتخاب می‌شود.

۲) $r \in \mathbb{Z}^*_N$ یکنواخت انتخاب می‌شود و متن رمز $c := [r^e \bmod N]$ و کلید $k := H(r)$ محاسبه می‌شوند.

۳) یک بیت یکنواخت $b \in \{0,1\}$ انتخاب می‌شود. اگر $b = 0$ باشد، قرار می‌دهیم $\hat{k} := k$. اگر $b = 1$ باشد، آنگاه یک $\hat{k} \in \{0,1\}^n$ یکنواخت، انتخاب می‌شود.

۴) $\mathcal{A}$ موارد $pk = \langle N,e\rangle$، c و $\hat{k}$ را دریافت می‌کند و می‌تواند $H(\cdot)$ (بر روی هر ورودی دلخواه) و پیشگوی لفافه زدایی $\text{Decaps}_{(N,d)}(\cdot)$ را بر روی هر متن رمز $\hat{c} \neq c$ مورد پرسمان قرار دهد.

۵) $\mathcal{A}$ یک بیت b' را به‌عنوان خروجی ارائه می‌کند. خروجی آزمایش برابر با ۱ تعریف می‌شود اگر $b = b'$ و در غیر این صورت، برابر با ۰ تعریف خواهد شد.

در یک اجرای آزمایش $KEM^{cca}_{\mathcal{A},\Pi}(n)$، فرض کنید $Query$ رخدادی باشد که در آن، در هر نقطه طی اجرا، $\mathcal{A}$ ،r را از پیشگوی تصادفی H پرسمان می‌کند. فرض می‌کنیم $Success$ نشان‌دهنده‌ی این رخداد باشد که $b' = b$ (یعنی، خروجی آزمایش برابر ۱ باشد). آنگاه

$$\begin{aligned}\Pr[Success] &= \Pr[Success \wedge \overline{Query}] + \Pr[Success \wedge Query] \\ &\leq \Pr[Success \wedge \overline{Query}] + \Pr[Query],\end{aligned}$$

که در آن، تمامی احتمالات روی مقدار تصادفی استفاده‌شده در آزمایش $KEM^{cca}_{\mathcal{A},\Pi}(n)$ گرفته‌شده‌اند. نشان می‌دهیم که $\Pr[Success \wedge \overline{Query}] \leq \frac{1}{2}$ و $\Pr[Query]$ ناچیز است. قضیه در ادامه آورده شده است.

در ابتدا استدلال می‌کنیم که $\Pr[\text{Success} \wedge \overline{\text{Query}}] \leq \frac{1}{2}$. اگر $\Pr[\overline{\text{Query}}] = 0$ باشد، این نتیجه فوراً به دست می‌آید. در غیر این صورت، $\Pr[\text{Success}|\overline{\text{Query}}]$. اکنون مشروط به $\overline{\text{Query}}$، مقدار کلید صحیح $k = H(r)$ یکنواخت است به این دلیل که H یک تابع تصادفی است. اطلاعات $\mathcal{A}$ در مورد k را در آزمایش $KEM^{cca}_{\mathcal{A},\Pi}(n)$ در نظر بگیرید. کلید عمومی pk و متن رمز c، به‌خودی‌خود، شامل هیچ اطلاعاتی در مورد k نیستند. (این موارد به‌صورت منحصربه‌فرد r را تعیین می‌کنند ولی ازآنجاکه H به‌صورت مستقل از هر چیز دیگری انتخاب شده است، این امر هیچ اطلاعاتی در مورد $H(r)$ به دست نمی‌دهد.) پرسمان‌هایی که $\mathcal{A}$ از H می‌پرسد هم هیچ اطلاعاتی در مورد r افشا نمی‌کنند مگر اینکه $\mathcal{A}$ ،r را از H پرسمان کند (که در این حالت، $Query$ رخ

دهد)؛ این امر بازهم به این واقعیت تکیه دارد که H یک تابع تصادفی است. درنهایت، پرسمان‌هایی
$\mathcal{A}$ از پیشگوی لفافه زدایی خود می‌پرسد تنها $H(\tilde{r})$ برای $\tilde{r} \neq r$ افشا می‌کنند. این امر از این
واقعیت نتیجه می‌شود که $\text{Decaps}_{(N,d)}(\tilde{c}) = H(\tilde{r})$ که در آن، $\tilde{r} = [\tilde{c}^d \bmod N]$ ولی $\tilde{c} \neq c$
نشان می‌دهد که $\tilde{r} \neq r$. بازهم این مطلب و این واقعیت که H یک تابع تصادفی است به این معنا
هستند که هیچ اطلاعاتی در مورد $H(r)$ افشا نمی‌شود مگر اینکه $Query$ رخ دهد.

مطلب بالا نشان می‌دهد که مادامی‌که $Query$ رخ ندهد، مقدار کلید صحیح k یکنواخت هستند
دنی با توجه به دیدگاه $\mathcal{A}$ از کلید عمومی، متن رمز و پاسخ‌های ارائه‌شده به پرسمان‌های پیشگوی
ن آنگاه در این حالت، هیچ راهی برای $\mathcal{A}$ وجود ندارد که بتواند تشخیص دهد (به شکلی بهتر از
حی تصادفی) که آیا $\hat{k}$ کلید صحیح است یا یک کلید یکنواخت و مستقل است. بنابراین،
$\Pr[Success|\overline{Query}]$.

یید می‌کنیم که هیچ جایی در استدلال بالا به این واقعیت تکیه نکردیم که $\mathcal{A}$ ازلحاظ محاسباتی
حدود شده است و درواقع $\Pr[Success \wedge \overline{Query}] \leq \frac{1}{2}$ حتی اگر هیچ محدودیت محاسباتی بر
روی $\mathcal{A}$ قرار داده نشده باشد. این امر نشان‌دهنده‌ی بخشی از قدرت مدل پیشگوی تصادفی است.

بمنظور کامل کردن اثبات این قضیه، نشان می‌دهیم که

ادعای ۱۱-۳۹

گر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد و H به‌عنوان یک پیشگوی تصادفی مدل‌سازی
شده باشد، آنگاه $\Pr[Query]$ ناچیز است.

بمنظور اثبات این امر، یک الگوریتم $\mathcal{A}'$ را می‌سازیم که از $\mathcal{A}$ به‌عنوان یک زیرروال استفاده می‌کند.
الگوریتم $\mathcal{A}'$ یک نمونه از N,e,c مسئله‌ی RSA را دریافت می‌کند و هدف آن، محاسبه‌ی r است که
برای آن $r^e = c \bmod N$ باشد. به‌منظور انجام این کار، این الگوریتم، $\mathcal{A}$ را اجرا کرده و پرسمان‌های
آن به H و $Decaps$ را پاسخ می‌دهد. مدیریت پرسمان‌ها از H ساده است چراکه $\mathcal{A}'$ می‌تواند به‌راحتی
یک مقدار تصادفی را بازگرداند. بااین‌حال، پرسمان‌ها از $Decaps$ اندکی پیچیده‌تر هستند چراکه $\mathcal{A}'$
کلید خصوصی مربوط به کلید عمومی مؤثر $\langle N,e\rangle$ را نمی‌داند.

بااین‌حال، با اندکی تأمل درمی‌یابیم که پاسخ به پرسمان‌های لفافه زدایی هم آسان است چراکه $\mathcal{A}'$
می‌تواند در اینجا نیز یک مقدار تصادفی را بازگرداند. یعنی، هرچند انتظار می‌رود پرسمان
$Decaps(\tilde{c})$ با محاسبه‌ی $\tilde{r}$ به صورتی که $\tilde{r}^e = \tilde{c} \bmod N$ و سپس ارزیابی $H(\tilde{r})$ محاسبه شود،
نتیجه درهرصورت تنها یک مقدار یکنواخت خواهد بود. بنابراین، $\mathcal{A}'$ می‌تواند به‌راحتی یک مقدار
تصادفی را ارائه کند بدون اینکه محاسبات میانی را انجام دهد. تنها مسئله آن است که $\mathcal{A}'$ باید از
سازگاری بین پاسخ‌های خود به پرسمان‌های H و پرسمان‌های $Decaps$ اطمینان حاصل کند؛ یعنی

باید تضمین نماید که برای هر $\bar{r}, \bar{c}$ با $H(\bar{r}) = Decaps(\bar{c})$ داشته باشیم $\bar{r}^e = \bar{c} \bmod N$. این امر با استفاده از ساماندهی ساده و فهرست‌های L_H و L_{Decaps} انجام می‌شود که پاسخ‌هایی را که $\mathcal{A}'$ در پاسخ به پرسمان‌های پیشگوی مربوطه ارائه نموده است را ثبت می‌کند. اکنون جزئیات را ارائه می‌کنیم.

الگوریتم $\mathcal{A}'$:

این الگوریتم، (N, e, c) را به‌عنوان ورودی دریافت می‌کند.

۱) فهرست‌های خالی L_H و L_{Decaps} را آغاز می‌کنیم. یک $k \in \{0,1\}^n$ یکنواخت را انتخاب کرده و (c, k) را در L_{Decaps} ذخیره می‌کنیم.

۲) یک بیت یکنواخت $b \in \{0,1\}$ را انتخاب می‌کنیم. اگر $b = 0$ باشد، قرار می‌دهیم $\hat{k} := k$. اگر $b = 1$، آنگاه یک $\hat{k} \in \{0,1\}^n$ یکنواخت را انتخاب می‌کنیم. $\mathcal{A}$ را بر روی $\langle N, e \rangle$، c و $\hat{k}$ اجرا می‌کنیم.

هنگامی‌که $\mathcal{A}$ یک پرسمان $H(\bar{r})$ را انجام می‌دهد، آن را به‌صورت زیر پاسخ می‌دهیم:

- اگر یک مدخل در L_H به شکل $(\bar{r}, k)$ برای یک k وجود داشته باشد، آنگاه k را بازمی‌گردانیم.
- در غیر این صورت، قرار می‌دهیم $\bar{c} := [\bar{r}^e \bmod N]$. اگر یک مدخل در L_{Decaps} به شکل $(\bar{c}, k)$ برای یک k وجود داشته باشد، k را بازگردانده و $(\bar{r}, k)$ را در L_H ذخیره می‌کنیم.
- در غیر این صورت، یک $k \in \{0,1\}^n$ یکنواخت را انتخاب می‌کنیم، k را بازمی‌گردانیم و $(\bar{r}, k)$ را در L_H ذخیره می‌کنیم.

هنگامی‌که $\mathcal{A}$ یک پرسمان $Decaps(\bar{c})$ را ارسال می‌کند، آن را به‌صورت زیر پاسخ می‌دهیم:

- اگر یک مدخل در L_{Decaps} به شکل $(\bar{c}, k)$ برای یک k وجود داشته باشد، k را باز می گردانیم.
- در غیر این صورت، برای هر مدخل $(\bar{r}, k) \in L_H$، بررسی می‌کنیم که آیا $\bar{r}^e = \bar{c} \bmod N$ و اگر این رابطه برقرار بود، k را اعلام می‌کنیم.
- در غیر این صورت، یک $k \in \{0,1\}^n$ یکنواخت را انتخاب می‌کنیم، k را بازمی‌گردانیم و $(\bar{c}, k)$ را در L_{Decaps} ذخیره می‌کنیم.

۳) در انتهای اجرای $\mathcal{A}$، اگر یک مدخل (r, k) در L_H وجود داشته باشد که برای آن، $r^e = c \bmod N$، آنگاه r را باز می گردانیم.

مشخص است که $\mathcal{A}'$ در زمان چندجمله‌ای اجرا می‌شود و دیدگاه $\mathcal{A}$ هنگام اجرا به‌عنوان یک زیرروال توسط $\mathcal{A}'$ در آزمایش $KEM^{cca}_{\mathcal{A},\Pi}(n)$ برابر با دیدگاه $\mathcal{A}$ در آزمایش $inv_{\mathcal{A}',GenRSA}(n) - RSA$ است: ورودی های داده شده به $\mathcal{A}$ به‌وضوح سازگار هستند و پاسخ‌ها به تمامی پرسمان‌های H، یکنواخت و

مستقل هستند. درنهایت، $\mathcal{A}'$ راه‌حل درست را دقیقاً زمانی که $Query$ رخ می‌دهد، خروجی می‌دهد. بنابراین، سختی مسئله‌ی RSA نسبت به $GenRSA$ به شکل ضمنی نشان می‌دهد که $\Pr[Query]$ ناچیز است، همان‌طور که نیاز بود. ■

در ادامه نکاتی در مورد ویژگی‌های مختلف مدل پیشگوی تصادفی (ن.ک. بخش ۵-۵-۱) استفاده‌شده در اثبات بالا، بیان نمود. اول اینکه، ما بر این واقعیت تکیه می‌کنیم که مقدار $H(r)$ یکنواخت است مگر اینکه r از H پرسمان شود–حتی اگر H بر روی چندین مقدار دیگر $r' \neq r$ مورد پرسمان قرار گیرد. همچنین به‌صورت ضمنی از استخراج پذیری برای ارائه‌ی این استدلال استفاده می‌کنیم که مهاجم می‌تواند r را از H پرسمان کند؛ در غیر این صورت، می‌توانیم از این مهاجم برای حل مسئله‌ی RSA استفاده کنیم. درنهایت، این اثبات به قابلیت برنامه‌نویسی به‌منظور شبیه‌سازی پرسمان‌های پیشگوی تابع زدایی متخاصم تکیه می‌کند.

۱۱-۵-۶- مسائل و دام‌های پیاده‌سازی RSA

در پایان این بخش، بحثی مختصر در مورد بعضی از مشکلات مربوط به پیاده‌سازی طرح‌های مبتنی بر RSA و تعدادی از دام‌هایی که باید به آن‌ها توجه کرد ارائه می‌کنیم.

استفاده از قضیه باقیمانده چینی

در پیاده‌سازی‌های رمزگذاری مبتنی بر RSA، گیرنده می‌تواند از قضیه‌ی باقیمانده‌ی چینی (بخش ۸-۱-۵) برای تسریع محاسبه‌ی ریشه‌های eام به پیمانه‌ی N طی رمزگشایی استفاده کند. به‌طور اخص، فرض می‌کنیم $N = pq$ و همین‌طور اینکه گیرنده می‌خواهد ریشه‌ی eام یک مقدار y را با استفاده از $d = [e^{-e} \bmod \phi(N)]$ محاسبه کند. گیرنده می‌تواند از تناظر $[y^d \bmod N] \leftrightarrow ([y^d \bmod p], [y^d \bmod q])$ برای محاسبه‌ی نتایج جزئی زیر استفاده کند

$$x_p := [y^d \bmod p] = [y^{[d \bmod (p-1)]} \bmod p]$$

و

$$x_q := [y^d \bmod q] = [y^{[d \bmod (q-1)]} \bmod q]$$

و سپس این‌ها را باهم ترکیب کند تا $x \leftrightarrow (x_p, x_q)$ را به دست آورد، همان‌طور که در بخش ۸-۱-۵ بحث شد. توجه کنید که $[d \bmod (p-1)]$ و $[d \bmod (q-1)]$ را می‌توان پیش-محاسبه نمود چرا که این موارد، مستقل از y هستند.

چرا این روش بهتر است؟ فرض کنید توان رسانی به پیمانه‌ی یک عدد صحیح l-بیتی نیازمند $\gamma \cdot \ell^3$ عملیات برای یک مقدار ثابت γ است. اگر p, q هرکدام دارای طول n بیت باشند، آنگاه محاسبه‌ی سادی $[y^d \bmod N]$ نیازمند $\gamma \cdot (2n)^3 = 8\gamma \cdot n^3$ گام است (به این دلیل که $\|N\| = 2n$).

استفاده از قضیه باقیمانده چینی باعث می‌شود این مقدار تا تقریباً $(\gamma \cdot n^{\gamma})$ ۲۰ گام (به این دلیل که $\|p\| = \|q\| = n$) یا تقریباً ۱/۴ زمان، کاهش یابد.

مثال ۱۱-۴۰

ما مثال ۸-۴۹ را دوباره بررسی می‌کنیم. یادآوری می‌کنیم که در آنجا، $N = ۱۴۳ = ۱۱.۱۳$ و $d = ۱۰۳$ و همین‌طور $y = ۶۴$ بود. به‌منظور محاسبه‌ی $[۶۴^{۱۰۳} \bmod ۱۴۳]$، عبارت زیر را محاسبه می‌کنیم

$$\begin{aligned}([۶۴ \bmod ۱۱], [۶۴ \bmod ۱۳])^{۱۰۳} &= ([(-۲)^{۱۰۳} \bmod ۱۱], [(-۱)^{۱۰۳} \bmod ۱۳]) \\ &= ([(-۲)^{[۱۰۳ \bmod ۱۰]} \bmod ۱۱], -۱) \\ &= ([-۸ \bmod ۱۱], -۱) = (۳, -۱).\end{aligned}$$

می‌توانیم $(۱,۰) \leftrightarrow ۱_p = ۷۸$ و $(۰,۱) \leftrightarrow ۱_q = ۶۶$ را به‌صورت بحث شده در بخش ۸-۱-۵ محاسبه کنیم. (توجه کنید که این مقادیر را می‌توان پیش-محاسبه نمود چراکه مستقل از y هستند.) آنگاه، $۲۵ \bmod ۱۴۳ = ۱۶۸ = ۶۶ - ۷۸ = ۳۰ \cdot ۱_p - ۱_q \leftrightarrow (۳, -۱)$ که منطبق با پاسخ به‌دست‌آمده‌ی قبلی است. ◊

یک حمله‌ی عیب هنگام استفاده از قضیه باقیمانده چینی

هنگام استفاده از قضیه باقیمانده چینی همان‌طور که هم‌اکنون توصیف شد، باید از یک حمله‌ی بالقوه آگاه باشیم که می‌تواند در صورتی انجام شود که طی محاسبه، «عیبی» رخ دهد (یا توسط مهاجم ایجاد شود مثلاً با دست‌کاری در سخت‌افزار).

این مسئله را در نظر بگیرید که چه اتفاقی خواهد افتاد اگر $[y^d \bmod N]$ دو بار محاسبه شود: اولین بار بدون خطا (که نتیجه‌ی صحیح x را می‌دهد) ولی دومین بار با یک خطا طی محاسبه‌ی معادله‌ی (۱۱.۲۰) ولی نه معادله‌ی (۱۱.۱۹) (همین حمله در حالت عکس اعمال می‌شود). محاسبه‌ی دوم یک نتیجه‌ی نادرست x' را می‌دهد که برای آن، $x' = x \bmod p$ ولی $x' \neq x \bmod q$ این امر به این معنا است که $p|(x' - x)$ ولی $q \nmid (x' - x)$. ولی آنگاه، $\gcd(x' - x, N) = p$ که تجزیه ی N را می‌دهد.

یک اقدام متقابل بالقوه عبارت است از وارسی درستی نتیجه قبل از استفاده از آن با بررسی اینکه $x^e = y \bmod N$ باشد. این مقوله در پیاده‌سازی‌های سخت‌افزاری، توصیه می‌شود.

کلیدهای عمومی وابسته I

وقتی چندین گیرنده می‌خواهند از طرح رمزگذاری یکسانی استفاده کنند، باید از کلیدهای عمومی «مستقل» استفاده نمایند. این حمله و حمله‌ی ذیل نشان می‌دهند که اگر این کار انجام نشود، چه مشکلی ممکن است پیش بیاید.

شرکتی را تصور کنید که می‌خواهد از پیمانه‌ی یکسان N برای هرکدام از کارمندان خود استفاده کند. ازآنجاکه مطلوب نیست که پیام‌های رمز شده برای یک کارمند توسط کارمند دیگری خوانده شوند، هر یک، زوج (e_i, d_i)های متفاوتی را برای هر کارمند، صادر می‌کند. یعنی، کلید عمومی کارمند iام برابر با $pk_i = \langle N, e_i \rangle$ و کلید خصوصی وی برابر با $sk = \langle N, d_i \rangle$ است که در آن، $e_i \cdot d_i = ۱ \bmod \phi(N)$ برای تمام iها.

این رویکرد، ناامن است و به هر کارمندی اجازه می‌دهد تا پیام‌های رمز شده برای همه‌ی دیگر کارمندان را بخواند. دلیل این امر آن است که، همان‌طور که در بخش ۸-۲-۴ اشاره شد، با داشتن N و e_i, d_i یا $e_i \cdot d_i = ۱ \bmod \phi(N)$، تجزیه ی N را می‌توان به‌صورت کارا محاسبه نمود. واضح است که با داشتن تجزیه ی N، می‌توان $d_j := e_j^{-۱} \bmod \phi(N)$ را برای هر j محاسبه نمود.

کلیدهای عمومی وابسته II

حمله‌ای که هم‌اکنون نشان داده شد به هر کارمندی اجازه می‌دهد تا پیام‌های ارسال‌شده برای هر کارمند دیگری را رمزگشایی کند. این امر همچنان این امکان را باقی می‌گذارد که تسهیم پیمانه‌ی N مشکلی نداشته باشد به شرطی که کارمندان به هم اعتماد داشته باشند (یا به شرطی که تنها نیاز باشد که محرمانگی علیه افراد بیرونی و نه علیه دیگر اعضای شرکت، حفظ شود). در اینجا سناریویی را نشان می‌دهیم که مشخص می‌کند تسهیم یک پیمانه همچنان یک ایده‌ی بد است حداقل زمانی که رمزگذاری RSA ساده مورداستفاده قرار گیرد.

فرض کنید پیام یکسان m رمز شده و برای دو کارمند متفاوت (معلوم) با کلیدهای عمومی $(N, e_۱)$ و $(N, e_۲)$، که در آن $e_۱ \neq e_۲$، ارسال می‌شود. همچنین فرض کنید که $\gcd(e_۱, e_۲) = ۱$. آنگاه، یک شنودگر دو متن رمز زیر را مشاهده می‌کند

$$c_۱ = m^{e_۱} \bmod N \quad \text{و} \quad c_۲ = m^{e_۲} \bmod N$$

ازآنجاکه $\gcd(e_۱, e_۲) = ۱$، بر اساس گزاره‌ی ۸-۲، اعداد صحیح X, Y وجود دارند به صورتی که $Xe_۱ + Ye_۲ = ۱$ باشد. به‌علاوه، با داشتن توانهای عمومی $e_۱$ و $e_۲$، می‌توان X و Y را با استفاده از الگوریتم اقلیدسی تعمیم‌یافته (ن.ک. ضمیمه‌ی ب-۱-۲) به‌صورت کارا محاسبه نمود. ادعا می‌کنیم $m = [c_۱^X \cdot c_۲^Y \bmod N]$، که می‌توان آن را به‌سادگی محاسبه نمود. این امر صدق می‌کند چراکه

$$c_۱^X \cdot c_۲^Y = m^{Xe_۱} m^{Ye_۲} = m^{Xe_۱ + Ye_۲} = m^۱ = m \bmod N.$$

یک حمله‌ی مشابه هنگام استفاده از RSA ایی گذاری شده یا $RSA - OAEP$ اعمال خواهد شد اگر فرستنده از پیام تبدیل‌شده‌ی یکسان $\hat{m}$ هنگام رمزگذاری برای دو کاربر، استفاده کند.

کیفیت مقدار تصادفی در تولید کلید RSA

در سراسر این کتاب، همواره فرض کرده‌ایم که طرفین صادق دارای دسترسی به مقدار تصادفی کافی و با کیفیت بالا هستند. هنگامی‌که این فرض نقض می‌شود، آنگاه ممکن است امنیت حفظ نشود. علی‌الخصوص، اگر یک رشته‌ی l-بیتی از یک مجموعه‌ی مفروض $S \subset \{0,1\}^\ell$ انتخاب شود به‌جای اینکه به‌صورت یکنواخت از $\{0,1\}^\ell$ انتخاب شود، آنگاه مهاجم می‌تواند یک جستجوی فراگیر (در زمان $O(|S|)$) را برای حمله به سیستم، اجرا کند.

در بعضی از موارد، وضعیت حتی ممکن است بدتر باشد. علی‌الخصوص، حالت تولید کلید RSA را در نظر بگیرید که در آن، یک نمونه از بیت‌های تصادفی r_p برای انتخاب اولین عدد اول، p، و یک نمونه‌ی ثانویه‌ی r_q برای تولید دومین عدد اول، q، مورداستفاده قرار می‌گیرد. همچنین فرض کنید که تعداد زیادی کلید عمومی/خصوصی با استفاده از همین منبع مقدار تصادفی با کیفیت ضعیف، تولید می‌شوند که در آن، r_p, r_q به‌صورت یکنواخت از یک مجموعه‌ی S با اندازه‌ی 2^s انتخاب می‌شوند. بعد از تولید تقریباً $2^{s/2}$ کلید عمومی (ن.ک. ضمیمه‌ی الف-۴)، انتظار داریم دو پیمانه‌ی متفاوت N, N' را به دست آوریم که با استفاده از مقدار تصادفی یکسان $r'_p = r_p$ تولید شده‌اند. این دو پیمانه دارای یک عامل اول مشترک هستند که می‌توان به‌سادگی آن را با محاسبه‌ی $\gcd(N, N')$، یافت. بنابراین، مهاجم می‌تواند اینترنت را به دنبال یک مجموعه‌ی بزرگ از کلیدهای عمومی RSA بگردد، مقسوم علیه‌های مشترک آن‌ها را محاسبه کند و امیدوار باشد که یک زیرمجموعه از آن‌ها را تجزیه کند. هرچند محاسبه‌ی مقسوم‌علیه‌های‌مشترک $2^{s/2}$ پیمانه به شکل ساده نیازمند زمان $O(2^s)$ است، مشخص شده است که این محاسبه را می‌توان با استفاده از یک رویکرد «تقسیم و حل» به شکل چشم‌گیری بهبود بخشید که فراتر از گستره‌ی این کتاب است. نتیجه آن است که مهاجم می‌تواند یک جستجوی فراگیر را در زمانی بسیار کمتر از 2^s اجرا کند. همچنین این حمله مفید خواهد بود حتی اگر مجموعه‌ی S برای مهاجم، نامعلوم باشد!

سناریوی بالا به‌صورت آزمایشی توسط دو تیم تحقیقاتی که به‌صورت مستقل کار می‌کردند، اعتبارسنجی شده است که این دو تیم دقیقاً همین حمله‌ی بالا را علیه کلیدهای عمومی یافت شده روی اینترنت اجرا نمایند و قادر بودند به‌صورت موفق یک بخش چشم‌گیر از کلیدهای یافت شده را تجزیه کنند.

مراجع و مطالعه‌ی بیشتر

ایده‌ی رمزگذاری کلید عمومی اولین بار در نوشتارهای آزاد توسط دیفی و هلمن [۵۸] پیشنهاد شد. ریوست، شامیر و آلدمن [۱۴۸] فرض RSA را معرفی کرده و یک طرح رمزگذاری کلید عمومی بر اساس این فرض را پیشنهاد دادند. همان‌طور که در فصل قبلی اشاره شد، دیگر پیشگامان رمزنگاری

کلید عمومی شامل مرکل و رابین (در نشریات دانشگاهی) و الیس، کاکس و ویلیامسون (در نشریات طبقه‌بندی شده) هستند.

تعریف ۱۱-۲ ریشه در مطالعه‌ی برجسته‌ی گلدواسر و میکالی [۸۰] دارد که همچنین اولین افرادی بودند که ضرورت رمزگذاری «احتمالاتی» برای برقراری این تعریف را درک کردند. همان‌طور که در فصل چهارم اشاره شد، حملات متن رمز منتخب اولین بار به‌صورت رسمی توسط نائور و یونگ [۱۲۹] و راکوف و سایمون [۱۴۷] تعریف شدند. مقاله‌ی توضیحی ارائه‌شده توسط شوپ [۱۵۶] اهمیت امنیت علیه حملات متن رمز منتخب را موردبحث قرار می‌دهد. بل ایر و همکاران یک بررسی واحد و مدرن را برای مفاهیم مختلف امنیت برای رمزگذاری کلید عمومی ارائه می‌کنند [۱۶].

یک اثبات برای CPA-امن بودن رمزگذاری ترکیبی اولین بار توسط لوم و گلدواسر [۳۶] ارائه شد. حالت امنیت CCA در [۵۶] بررسی شده است.

به شکلی شگفت‌آور، طرح رمزگذاری الجمال [۷۰] تا سال ۱۹۸۴ پیشنهاد نشد هرچند که می‌توان آن را به‌عنوان یک تبدیل مستقیم از پروتکل تبادل کلید دیفی-هلمن (ن.ک. تمرین ۱۱-۴) به شمار آورد. $DHIES$ در [۲] معرفی شد. استاندارد ۲ – ۱۸۰۳۳ ISO/IEC برای رمزگذاری کلید عمومی را می‌توانید در http://www.shoup.net/iso بیابید.

رمزگذاری RSA ساده متناظر با طرح اولیه‌ی معرفی‌شده توسط ریوست، شامیر و آلدمن [۱۴۸] است. حملات علیه رمزگذاری RSA ساده که در بخش ۱۱-۵-۱ توصیف شده‌اند توسط [۱۶۱، ۵۵، ۸۴، ۴۷، ۴۰] ارائه شده‌اند؛ برای حملات دیگر و اطلاعات بیشتر ن.ک. [۱۲۰، فصل هشتم] و [۳۸]. اثبات‌های قضیه‌ی کاپر اسمیت را می‌توانید در مطالعه‌ی اصلی [۴۶] یا چندین مطالعه‌ی بعدی (برای مثال، [۱۱۹]) بیابید.

استانداردهای رمزگذاری RSA $PKCS\,\#1$ (هم نسخه‌ی قبلی و هم نسخه‌ی جاری) در http://www.emc.com/emc-plus/rsa-labs در دسترس هستند. حمله‌ی مبتنی بر متن اصلی انتخابی علیه $PKCS\,\#1$ v۱.۵ توصیف‌شده در اینجا، توسط [۴۹] ارائه شده است. توصیفی از حمله‌ی متن رمز منتخب بلایکنباکر علیه $PKCS\,\#1$ v۱.۵ را می‌توانید در مقاله‌ی اصلی [۳۴] بیابید. برای بهبودهای بعدی ن.ک. [۱۲].

اثبات‌های قضیه‌ی ۱۱-۳۱ و تعمیم‌ها را می‌توانید در [۸، ۸۶، ۶۶، ۷] بیابید. برای بررسی کلی طرح‌های با این شکل، ن.ک. بخش ۱۳-۱-۲. به نظر می‌رسد ساختار ۱۱-۳۷ در ابتدا توسط شوپ [۱۵۱] معرفی و تحلیل شده است. $OAEP$ توسط بل ایر و روگاوای [۲۲] معرفی شد. بعداً مشخص شد که اثبات اصلی $OAEP$ دارای خطا است؛ خوانندگان علاقه‌مند می‌توانند به [۳۹، ۱۵۸، ۶۸] مراجعه کنند. برای جزئیات حمله‌ی متن رمز منتخب منگر علیه پیاده‌سازی‌های $PKCS\,\#1$ v۲.۰ ن.ک. [۱۱۷].

حمله‌ی بزرگترین مقسوم علیه مشترک توصیف‌شده در بخش ۱۱-۵-۶ توسط لنسترا و همکاران [۱۱۲] و هنینگر و همکاران [۸۸] انجام شد.

هنگام استفاده از هر طرح رمزگذاری در عمل، این سؤال به وجود می‌آید که چه طول کلیدی را باید استفاده کرد. این مسئله را نباید دست‌کم گرفت و برای بحث و بررسی عمیق در این خصوص، خواننده می‌تواند به بخش ۹-۳ و مراجع موجود در آن، مراجعه کند.

اولین طرح رمزگذاری کلید عمومی CCA-امن کارا که به مدل پیشگوی تصادفی تکیه نمی‌کند توسط کریمر و شوپ [۵۰] بر اساس فرض DDH نشان داده شد. بعداً، هاف هاینس و کیلتس یک طرح CCA-امن کارا بدون پیشگوهای تصادفی را بر اساس فرض RSA نشان دادند [۹۲].

تمرین‌ها

۱۱-۱: یک طرح رمزگذاری کلید عمومی برای پیام‌های تک‌بیتی بدون خطای رمزگشایی را در نظر بگیرید. نشان دهید که با داشتن pk و یک متن رمز c محاسبه‌شده از طریق $c \leftarrow Enc_{pk}(m)$، برای یک متخاصم نامحدود، تعیین m با احتمال ۱، ممکن است.

۱۱-۲: نشان دهید که برای هر طرح رمزگذاری کلید عمومی CPA-امن برای پیام‌های تک‌بیتی، طول متن رمز باید ابر لگاریتمی بر اساس پارامتر امنیتی باشد.

راهنمایی: اگر این‌گونه نباشد، دامنه‌ی متون رمز ممکن دارای اندازه‌ی چندجمله‌ای خواهد بود.

۱۱-۳: فرض کنید یک طرح رمزگذاری کلید عمومی (Gen, Enc, Dec) برای پیام‌های n-بیتی، «یک‌طرفه» است اگر هر متخاصم PPT، $\mathcal{A}$، دارای احتمال موفقیت ناچیز در آزمایش زیر باشد:

- $Gen(1^n)$ اجرا می‌شود تا کلیدهای (pk, sk) به دست آیند.
- یک پیام $m \in \{0,1\}^n$ به‌صورت یکنواخت و تصادفی انتخاب می‌شود و یک متن رمز $c \leftarrow Enc_{pk}(m)$ محاسبه می‌شود.
- موارد pk و c به $\mathcal{A}$ داده می‌شود و $\mathcal{A}$ یک پیام m' را به‌عنوان خروجی ارائه می‌کند. می‌گوییم $\mathcal{A}$ موفق می‌شود اگر $m' = m$ باشد.

(الف) ساختاری از یک KEM دارای امنیت CPA در مدل پیشگوی تصادفی بر اساس هر طرح رمزگذاری کلید عمومی یک‌طرفه‌ی مفروض، ارائه کنید.

(ب) آیا یک طرح رمزگذاری کلید عمومی «قطعی» می‌تواند یک‌طرفه باشد؟ اگر خیر، عدم امکان این امر را اثبات کنید؛ اگر بله، ساختاری بر اساس یکی از فرضیات معرفی‌شده در این کتاب، ارائه کنید.

۱۱-۴: نشان دهید که هر پروتکل تبادل کلید دو-دور (یعنی، در آن هر طرف یک پیام یکتا را ارسال می‌کند) صادق در تعریف ۱۰-۱ را می‌توان به یک طرح رمزگذاری کلید عمومی CPA-امن تبدیل نمود.

۱۱-۵: نشان دهید که ادعای ۱۱-۷ در موقعیت امنیت CCA، برقرار نیست.

۱۱-۶: طرح رمزگذاری کلید عمومی زیر را در نظر بگیرید. کلید عمومی برابر با $(\mathbb{G}, q, g, h)$ و کلید خصوصی برابر با x است که دقیقاً به‌مانند طرح رمزگذاری الجمال، تولید شده‌اند. به‌منظور رمزگذاری یک بیت b، فرستنده موارد زیر را انجام می‌دهد:

(الف) اگر $b = 0$، آنگاه یک $y \in \mathbb{Z}_q$ یکنواخت را انتخاب کرده و $c_1 := g^y$ و $c_2 := h^y$ را محاسبه می‌کند. متن رمز برابر با $\langle c_1, c_2 \rangle$ است.

(ب) اگر $b = 1$، آنگاه $y, z \in \mathbb{Z}_q$ یکنواخت مستقل را انتخاب می‌کند، $c_1 := g^y$ و $c_2 := g^z$ را محاسبه کرده و متن رمز را برابر با $\langle c_1, c_2 \rangle$ قرار می‌دهد.

نشان دهید که می‌توان با دانستن x، رمزگشایی را به‌صورت کارا انجام داد. اثبات کنید که این طرح رمزگذاری، CPA-امن خواهد بود اگر مسئله‌ی دیفی-هلمن تصمیمی نسبت به $\mathbb{G}$ سخت باشد.

۱۱-۷: گونه‌ی زیر از رمزگذاری الجمال را در نظر بگیرید. فرض کنید $p = 2q + 1$ باشد، $\mathbb{G}$ گروهی از مرتبات به پیمانه‌ی p (بنابراین $\mathbb{G}$ یک زیرگروه از $\mathbb{Z}_p^*$ با مرتبه‌ی q است) و g یک مولد برای $\mathbb{G}$ باشد. کلید خصوصی برابر با $(\mathbb{G}, g, q, x)$ و کلید عمومی برابر با $(\mathbb{G}, g, q, h)$ است که در آن، $h = g^x$ و $x \in \mathbb{Z}_q$ به‌صورت یکنواخت انتخاب شده است. به‌منظور رمزگذاری یک پیام $m \in \mathbb{Z}_q$، یک $r \in \mathbb{Z}_q$ یکنواخت را انتخاب کرده، $c_1 := g^r \bmod p$ و $c_2 := h^r + m \bmod p$ را محاسبه کرده و متن رمز را برابر با $\langle c_1, c_2 \rangle$ قرار می‌دهیم. ایا این طرح، CPA-امن است؟ پاسخ خود را اثبات کنید.

۱۱-۸: پروتکل زیر برای دو طرف A و B برای پرتاب یک سکه‌ی سالم را در نظر بگیرید (نسخه‌های پیچیده‌تر این پروتکل را می‌توان برای قمار اینترنتی استفاده نمود): (۱) یک طرف مورد اعتماد T کلید عمومی pk خود را منتشر می‌کند؛ (۲) سپس، A یک بیت یکنواخت b_A را انتخاب می‌کند. آن را با استفاده از pk رمزگذاری می‌کند و متن رمز c_A را به B و T اعلام می‌کند؛ (۳) در ادامه، B به‌صورت متقارن رفتار کرده و یک متن رمز $c_B \neq c_A$ را اعلام می‌کند؛ (۴) T هم c_A و هم c_B را رمزگشایی می‌کند و طرفین، نتایج را XOR می‌کنند تا مقدار سکه را به دست آورند.

(الف) استدلال کنید که اگر A صادق نباشد (ولی B صادق باشد)، مقدار نهایی سکه دارای توزیع یکنواخت خواهد بود.

(ب) فرض کنید طرفین از رمزگذاری الجمال استفاده می‌کنند (که در آن، بیت b قبل از رمز شدن، به‌عنوان عنصر گروهی g^b کدگذاری می‌شود- توجه کنید که رمزگشایی کارا همچنان ممکن است).

نشان دهید که چگونه یک B غیر صادق می‌تواند باعث اریب شدن سکه به سمت هر مقدار دلخواه خود شود.

(ج) به نظر شما چه نوع طرح رمزگذاری برای استفاده در اینجا، مناسب خواهد بود. آیا می‌توانید یک مفهوم امنیت مناسب را تعریف کرده و اثبات کنید که پیشنهاد شما در این تعریف صدق می‌کند؟

۱۱-۹: به‌صورت رسمی اثبات کنید که طرح رمزگذاری الجمال، CCA-امن نیست.

۱۱-۱۰: در بخش ۱۱-۴-۴، نشان دادیم که رمزگذاری الجمال شکل‌پذیر است و علی‌الخصوص اینکه با داشتن یک متن رمز $\langle c_1, c_2 \rangle$ که رمزگذاری یک پیام نامعلوم مفروض m است، می‌توان یک متن رمز $\langle c_1, c_2' \rangle$ را تولید نمود که رمزگذاری $\alpha . m$ (برای α معلوم) است. گیرنده‌ای که هر دو این متون رمز را دریافت می‌کند ممکن است مشکوک شود چراکه هر دو متن رمز دارای مؤلفه‌ی اول یکسانی هستند. نشان دهید که می‌توان $\langle c_1', c_2' \rangle$ را که رمزگذاری $\alpha . m$ است با $c_1' \neq c_1$ و $c_2' \neq c_2$ تولید نمود.

۱۱-۱۱: قضیه‌ی ۱۱-۲۲ را اثبات کنید.

۱۱-۱۲: یکی از حملات علیه RSA ساده بحث شده در بخش ۱۱-۵-۱ شامل فرستنده‌ای است که دو پیام مرتبط را با استفاده از کلید عمومی یکسان، رمزگذاری می‌کند. یک تعریف مناسب برای امنیت را بیان کنید که چنین حملاتی را منتفی کند و نشان دهید که هر طرح رمزگذاری کلید عمومی CPA-امن، در تعریف شما صدق می‌کند.

۱۱-۱۳: یکی از حملات علیه RSA ساده بحث شده در بخش ۱۱-۵-۱ شامل فرستنده‌ای است که پیام یکسانی را برای سه گیرنده‌ی متفاوت، رمزگذاری می‌کند. یک تعریف مناسب برای امنیت را بیان کنید که چنین حملاتی را منتفی کند و نشان دهید که هر طرح رمزگذاری کلید عمومی CPA-امن، در تعریف شما صدق می‌کند.

۱۱-۱۴: نسخه‌ی تغییریافته‌ی زیر برای رمزگذاری RSA لایی گذاری شده را در نظر بگیرید: فرض کنید پیام‌هایی که باید رمزگذاری شوند دقیقاً دارای طول $\|N\|/2$ هستند. برای رمزگذاری، ابتدا $\hat{m} := 0x00 \| r \| 0x00 \| m$ را محاسبه کنید که در آن، r یک رشته‌ی یکنواخت با طول $\|N\| - 16 - 2$ است. سپس، متن رمز $c := [\hat{m}^e \bmod N]$ را محاسبه کنید. هنگام رمزگشایی یک متن رمز c، گیرنده $\hat{m} := [c^d \bmod N]$ را محاسبه کرده و اگر $\hat{m}$ شامل $0x00$ به اضافه ی $\|N\|/2 - 16$ بیت دلخواه به اضافه ی $0x00$ در ابتدا نباشد، یک خطا را بازمی‌گرداند. نشان دهید که این طرح، CPA-امن نیست. چرا ساختن یک حمله‌ی متن رمز منتخب علیه این طرح ساده‌تر از ساختن چنین حمله‌ای علیه $PKCS\#1$ v1.5 است؟

۱۱-۱۵: طرح رمزگذاری مبتنی بر RSA را در نظر بگیرید که در آن، کاربر، یک پیام $m \in \{0,1\}^\ell$ را بر اساس کلید عمومی $\langle N, e \rangle$ با محاسبه‌ی $\hat{m} := H(m) \| m$ و اعلام متن رمز $[\hat{m}^e \bmod N]$ رمزگذاری می‌کند. (در اینجا، فرض کنید $H: \{0,1\}^\ell \to \{0,1\}^n$ و $\ell + n < \|N\|$.) گیرنده، $\hat{m}$ را به شکل معمول بازیابی کرده و وارسی می‌کند که دارای شکل درست باشد قبل از آنکه l کم‌اهمیت‌ترین بیت را به‌عنوان m خروجی دهد. اثبات یا رد کنید که این طرح، CCA-امن است اگر H به‌عنوان یک پیشگوی تصادفی، مدل‌سازی شده باشد.

۱۱-۱۶: یک حمله‌ی متن رمز منتخب علیه ساختار ۱۱-۳۴ نشان دهید.

۱۱-۱۷: فرض کنید $\Pi = (Gen, Enc, Dec)$ یک طرح رمزگذاری کلید عمومی CPA-امن باشد و فرض کنید $\Pi' = (Gen', Enc', Dec')$ یک طرح رمزگذاری کلید خصوصی CCA-امن باشد. ساختار زیر را در نظر بگیرید:

ساختار ۱۱-۴۱

فرض کنید $H: \{0,1\}^n \to \{0,1\}^n$ یک تابع باشد. یک طرح رمزگذاری کلید عمومی را به‌صورت زیر ایجاد می‌کنیم:

- Gen: با دریافت 1^n به‌عنوان ورودی، $Gen(1^n)$ را اجرا می‌کند تا (pk, sk) را به دست آورد. این موارد را به ترتیب به‌عنوان کلیدهای عمومی و خصوصی، اعلام می‌کنیم.
- Enc: با دریافت یک کلید عمومی pk و یک پیام $m \in \{0,1\}^n$ به‌عنوان ورودی، یک $r \in \{0,1\}^n$ یکنواخت را انتخاب کرده و متن رمز زیر را به‌عنوان خروجی ارائه می‌کنیم:
$$\langle \mathrm{Enc}_{pk}(r), \mathrm{Enc}'_{H(r)}(m) \rangle$$
- Dec: با دریافت یک کلید خصوصی sk و یک متن رمز $\langle c_1, c_2 \rangle$ به‌عنوان ورودی، $r := Dec_{sk}(c_1)$ را محاسبه کرده و قرار می‌دهیم $k := H(r)$. سپس، $Dec'_k(c_2)$ را به‌عنوان خروجی ارائه می‌کنیم.

آیا ساختار بالا دارای رمزگذاری‌های تمایزناپذیر تحت یک حمله‌ی مبتنی بر «متن رمز» انتخابی خواهد بود اگر H به‌عنوان یک پیشگوی تصادفی، مدل‌سازی شود؟ اگر بله، یک اثبات را ارائه کنید. اگر خیر، رویکرد استفاده‌شده برای اثبات قضیه‌ی ۱۱-۳۸ در کجا دچار شکست می‌شود؟

۱۱-۱۸: ساختار ۱۱-۴۲ را بعنوان گونه‌ای از ساختار ۱۱-۳۲ در نظر بگیرید:
اثبات کنید که این طرح، CPA-امن است. مزیت‌ها و معایب آن نسبت به ساختار ۱۱-۳۲ را موردبحث قرار دهید.

ساختار ۱۱-۴۲

فرض کنید $GenRSA$ به شکل معمول باشد و یک طرح رمزگذاری کلید عمومی را به‌صورت زیر تعریف کنید:

- Gen: با دریافت 1^n به‌عنوان ورودی، $GenRSA(1^n)$ را اجرا می‌کنیم تا (N,e,d) را به دست آوریم. کلید عمومی $pk = \langle N,e \rangle$ و کلید خصوصی $sk = \langle N,d \rangle$ را به‌عنوان خروجی ارائه می‌کنیم.
- Enc: با دریافت یک کلید عمومی $pk = \langle N,e \rangle$ و یک پیام $m \in \{0,1\}$ به‌عنوان ورودی، یک $r \in \mathbb{Z}_N^*$ یکنواخت را انتخاب می‌کنیم. متن رمز $\langle [r^e \bmod N], \mathrm{lsb}(r) \oplus m \rangle$ را به‌عنوان خروجی ارائه می‌کنیم.
- Dec: با دریافت یک کلید خصوصی $sk = \langle N,d \rangle$ و یک متن رمز $\langle c,b \rangle$ به‌عنوان ورودی، $r := [c^d \bmod N]$ را محاسبه کرده و $lsb(r) \oplus b$ را به‌عنوان خروجی ارائه می‌کنیم.

۱۱-۱۹: فرض کنید سه کاربر دارای کلیدهای عمومی RSA به‌صورت $\langle N_1, 3 \rangle$، $\langle N_2, 3 \rangle$ و $\langle N_3, 3 \rangle$ (یعنی، همه‌ی آن‌ها از $e = 3$ استفاده می‌کنند) با $N_1 < N_2 < N_3$ هستند. روش زیر برای ارسال پیام یکسان $m \in \{0,1\}^\ell$ برای هرکدام از این طرفین را در نظر بگیرید: یک $r \leftarrow \mathbb{Z}_{N_1}^*$ یکنواخت را انتخاب کرده و متن رمز یکسان زیر را برای همه می فرستیم:

$$\langle [r^3 \bmod N_1], [r^3 \bmod N_2], [r^3 \bmod N_3], H(r) \oplus m \rangle,$$

که در آن، $H: \mathbb{Z}_{N_1}^* \to \{0,1\}^\ell$. فرض کنید که $\ell \gg n$.

(الف) نشان دهید که این روش، CPA-امن نیست و متخاصم می‌تواند m را از روی متن رمز بازیابی کند حتی زمانی که H به‌عنوان یک پیشگوی تصادفی مدل‌سازی شده است.

راهنمایی: ن.ک. بخش ۱۱-۵-۱.

(ب) یک روش ساده برای تصحیح این روش و رسیدن به یک روش CPA-امن نشان دهید که یک متن رمز با طول $3\ell + O(n)$ را ارسال می‌کند.

(ج) یک رویکرد بهتر نشان دهید که همچنان CPA-امن باشد ولی با متن رمزی به طول $\ell + O(n)$.

۱۱-۲۰: یک کلید عمومی RSA به شکل $\langle N,e \rangle$ را مشخص کرده و فرض کنید که یک الگوریتم $\mathcal{A}$ داریم که همواره به‌درستی $lsb(x)$ را با داشتن $[x^e \bmod N]$ محاسبه می‌کند. شبه کد کامل را برای یک الگوریتم $\mathcal{A}'$ بنویسید که x را از روی $[x^e \bmod N]$ محاسبه می‌کند.

۱۱-۲۱: یک کلید عمومی RSA به شکل $\langle N,e \rangle$ را در نظر گرفته و تعریف کنید

$$half(x) = \begin{cases} 0 & if\ 0 < x < N/2 \\ 1 & if\ N/2 < x < N \end{cases}$$

ثابت کنید که $half$ یک پایه‌ی سخت هسته برای مسئله‌ی RSA است.

راهنمایی: به سختی محاسبه‌ی lsb کاهش دهید.