

بخش سوم: رمزنگاری کلید عمومی (نامتقارن)

فصل هشتم: نظریه‌ی اعداد و فرضیه‌های سخت رمزنگاری

بسیارهای رمزنگاری مدرن معمولاً مبتنی بر یک فرض هستند مبنی بر اینکه یک مسئله‌ی مفروض، سخت است. برای مثال، در فصول سوم و چهارم، مشاهده کردیم که رمزنگاری کلید خصوصی-هم طرح‌های رمزگذاری و هم کدهای احراز اصالت پیام-می‌توانند بر اساس این فرض باشند که جایگشت‌های شبه تصادفی (که به‌عنوان رمزهای قالبی هم شناخته می‌شوند) وجود دارند. یادآوری می‌کنیم که این امر تقریباً به این معنا است که یک جایگشت کلیددار مفروض F وجود دارد که برای آن، در زمان چندجمله‌ای، تشخیص دادن تعاملات با F (برای یک کلید یک‌باراخت مجهول k) و تعاملات با یک جایگشت کاملاً تصادفی، سخت است.

اِلمَظ ظاهری، این فرض که جایگشت‌های شبه تصادفی وجود دارند به نظر تا حد زیادی قوی و غیرطبیعی است و منطقی است بپرسیم آیا این فرض درست است یا خیر یا آیا شواهدی برای پشتیبانی از آن وجود دارد یا خیر. در فصل ششم، این مسئله را بررسی کردیم که رمزهای قالبی به چه صورت در عمل ساخته می‌شوند. این واقعیت که ساختارهای موجود در برابر حمله مقاوم هستند به‌عنوان تمثالی از این مفهوم است که وجود جایگشت‌های شبه تصادفی، قابل‌قبول است. بااین‌حال، ممکن است بار به این امر دشوار باشد که هیچ حمله‌ی تمایزگر کارایی علیه رمزهای قالبی موجود، وجود ندارد. همچنین، وضعیت کنونی نظریه‌ی ما به این صورت است که نمی‌دانیم به چه صورت شبه تصادفی بودن هیچ کدام از ساختارهای عملی موجود را نسبت به یک فرض «ساده‌تر» یا «منطقی‌تر» اثبات کنیم، به‌طور کلی، وضعیت کنونی یک وضعیت مطلوب نیست.

در مثال، همان‌طور که در فصل سوم اشاره شد (و در فصل هفتم به‌دقت بررسی شد) این امکان وجود دارد که وجود جایگشت‌های شبه تصادفی را بر اساس این فرض بسیار ملایم‌تر اثبات کنیم که تابع یک‌طرفه وجود دارند. (به شکل غیررسمی، یک تابع، یک‌طرفه است اگر محاسبه‌ی آن آسان باشد ولی «برعکس» آن سخت باشد).^{۱۰-۸-۱} بخش ۸-۴-۱ بااین‌حال، به‌غیر از یک بحث مختصر در بخش ۷-۱۰-۱، مثال‌های عینی از توانایی که ماور داریم یک‌طرفه هستند، مشاهده نکرده‌ایم.

۲۲-۷: فرض کنید $\{X_n\}_{n \in \mathbb{N}}$ و $\mathcal{X} = \{X_n\}_{n \in \mathbb{N}}$ و $\mathcal{Y}$ خانواده‌های احتمال تمایزناپذیر محاسباتی باشند. اثبات کنید که برای هر الگوریتم زمان-چندجمله‌ای تصادفی M مجموعه‌های $\{M(X_n)\}_{n \in \mathbb{N}}$ و $\{M(Y_n)\}_{n \in \mathbb{N}}$ تمایزناپذیر محاسباتی هستند.

مثال خاصی که بررسی می‌کنیم دارای ماهیت نظریه‌ای اعداد هستند و بنابراین با یک مقدمه‌ی کوتاه برای نظریه‌ی اعداد و نظریه‌ی گروه‌ها، بحث را آغاز می‌کنیم. به این دلیل که همچنین به مسائلی علاقه داریم که می‌توان آن‌ها را به‌صورت کارا حل نمود (حتی یک تابع یک‌طرفه هم باید قابل محاسبه‌ی ساده در یک‌جهت باشد و یک طرح رمزنگاری باید الگوریتمی کارا را برای طرفین حاکم پذیرد)، همچنین مطالعه‌ی نظریه‌ی اعداد الگوریتمی را آغاز می‌کنیم. حتی به خوانندگانی که با نظریه‌ی اعداد یا نظریه‌ی گروه‌ها آشنایی دارد پیشنهاد می‌شود که این فصل را مطالعه کنند چراکه ابعاد الگوریتمی معمولاً در بررسی کاملاً ریاضیاتی این موضوعات، نادیده گرفته می‌شوند.

هدف دوم این فصل، توسعه‌ی مطالب مورد نیاز برای رمزنگاری کلید عمومی است که مطالعه در خصوص آن را در فصل دهم شروع می‌کنیم. چالش انحصاری که علیرغم اینکه در حالت کلیه خصوصی، ساختارهای کاری پایه‌های ضروری (دروهای قاتی و توابع یک‌جهت ساده) بدون فراخوانی نظریه‌ی اعداد وجود دارند، در حالت کلید عمومی، تمایلی ساختارهای شناخته‌شده به مسائل نظریه اعداد سخت، تکمیل دارند. بنابراین، مطالب این فصل هم به‌عنوان پرابنده مطالعه‌ی ما تاکنون در رابطه با رمزنگاری کلید خصوصی به شمار می‌رود و هم به‌عنوان پایه و اساسی برای رمزنگاری کلید عمومی.

۱-۸- مفاهیم و نظریه‌ی اعداد پایه

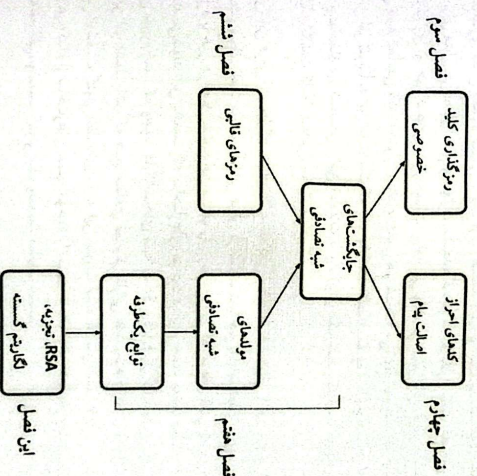
بازآوری بر اعداد اول و حساب پیمانه‌ای پایه کار را آغاز می‌کنیم. حتی خوانندگانی که این موضوعات را قبلاً مشاهده کرده است هم باید دو بخش بعدی را مطالعه کنند به این دلیل که بخشی از مطالب ممکن است جدید باشند و ما اثبات‌هایی را برای اغلب نتایج بیان ندهیم، می‌آوریم.

۱-۸-۱- اعداد اول و بخش پذیری

مجموعه‌ی اعداد صحیح توسط $\mathbb{Z}$ نشان داده می‌شود. برای $a, b \in \mathbb{Z}$ می‌گوییم b بر a بخش‌پذیر است که به‌صورت $a|b$ نوشته می‌شود. اگر یک عدد صحیح c وجود داشته باشد به صورتی که $ac = b$ و a بر b بخش‌پذیر نباشد، می‌نویسیم $a \nmid b$ (ما اصولاً علاقه‌مند به حالتی هستیم که a و b همدیگر شریک باشند، هرچند این تعریف متناظر است حتی زمانی که یک یا چند مورد از این اعداد، منفی یا صفر باشند). یک برداشت ساده این است که اگر $a|b$ و $a|c$ آنگاه $a|(b+c)$ برای هر $x, y \in \mathbb{Z}$.

اگر $a|a$ و a شریک باشد، ما a را یک مقسوم‌علیه، یا یک عامل ضرب (فاکتور)، غیربدیهی a می‌نامیم. یک عدد صحیح مثبت $1 < a$ یک عدد اول است اگر هیچ عامل ضرب (فاکتور) نداشته باشد. یعنی، تنها دو مقسوم‌علیه داشته باشد: 1 و خودش. یک عدد صحیح مثبت بزرگ‌تر از یک که عدد اول نباشد، عدد مرکب نامیده می‌شود. بر اساس قرارداد، عدد 1 به اول است و نه مرکب.

یکی از اهداف این فصل، معرفی مسائل مختلفی است که باور بر آن است که «سخت» هستند و همین‌طور آرائی توابع یک‌طرفه‌ی جدیدی زده‌شده بر اساس این مسائل است.^{۳۳} بنابراین، این فصل را می‌توان به‌عنوان جمع‌بندی یک رویکرد «بالا به پایین» برای رمزنگاری کلید خصوصی در نظر گرفت. از آن‌جا که تصویر (۱-۸) یعنی در فصل سوم و چهارم نشان دادیم که رمزنگاری کلید خصوصی می‌تواند بر اساس توابع و جایگشت‌های شبه تصادفی باشد در ادامه مشاهده کردیم که مورد دوم را می‌توان در عمل با استفاده از دروهای قاتی نمونه‌سازی نمود. همان‌طور که در فصل ششم بررسی شد، یا می‌توان آن‌ها را به شکلی استوار از روی هر تابع یک‌طرفه‌ای ایجاد نمود. همان‌طور که در فصل ششم نشان داده شد، در اینجا بحث را یک گام به جلو برده و نشان می‌دهیم که چگونه توابع یک‌طرفه می‌توانند بر اساس بعضی مسائل ریاضیاتی سخت باشند.



تصویر ۱-۸: رمزنگاری کلید خصوصی: یک رویکرد بالا به پایین

^{۳۳} یادآوری می‌کنیم که در حال حاضر نمی‌توانیم چگونه اثبات کنیم که توابع یک‌طرفه وجود ندارد، بنابراین بهترین کاری که می‌توانیم بکنیم آن است که توابع یک‌طرفه را بر اساس فرض‌های مربوط به بعضی مسائل عامی قرار دهیم.

یک قضیه‌ی بنیادی رئاییات این است که هر عدد صحیح بزرگتر از ۱ را می‌توان به شکل منحصر به فرد (بر اساس درجه) به عنوان ضربی از اعداد اول بیان نمود. یعنی، هر عدد صحیح مثبت $N > 1$ را می‌توان به صورت $N = \prod_{i=1}^r p_i^{a_i}$ نوشت که در آن، $\{p_i\}$ اعداد اول متمایز و $a_i \geq 1$ برای تمام i ها است. به علاوه، $\{p_i\}$ و $\{a_i\}$ به شکل منحصر به فرد بر اساس درجه، تعیین می‌شوند.

ما با فرآیند تقسیم با باقی مانده از دوران مدرسه آشنایی داریم. گزاره‌ی زیر، این مفهوم را رسمی می‌کند.

گزاره‌ی ۸-۱

فرض کنید a یک عدد صحیح و b یک عدد صحیح مثبت باشند. آنگاه اعداد صحیح منحصر به فرد q, r وجود خواهند داشت که برای آن‌ها، $a = qb + r$ و $0 \leq r < b$.

به علاوه، با دانش اعداد صحیح a و b به صورت تعریف شده در گزاره، می‌توان q و r را در زمان چندجمله‌ای محاسبه نمود.^۱ به ضمیمه‌ی ب-۱ (زمان اجرای یک الگوریتم به عنوان تابعی از طول (های) ورودی (های) آن، اندازه‌گیری می‌شود. یک نکته‌ی مهم در رابطه با نظریه‌ی اعداد الگوریتمی آن است که همیشه فرض می‌شود که اعداد به صورت دودویی نشان داده می‌شوند. بنابراین، زمان اجرای یک الگوریتم در یافت‌کننده‌ی یک عدد صحیح N به عنوان ورودی بر اساس $\|N\|$ اندازه‌گیری می‌شود، که طول نمایش دودویی N است. توجه کنید که $1 + \log N = \lceil \log N \rceil$ است.)

بزرگترین مقسوم علیه مشترک دو عدد صحیح، a, b ، نوشته شده به صورت $\gcd(a, b)$ عبارت است از بزرگترین عدد صحیح c به صورتی که $a|c$ و $b|c$ (ما $\gcd(0, 0)$ را تعریف نشده در نظر می‌گیریم). مفهوم بزرگترین مقسوم علیه مشترک همچنان معنا دار خواهد بود اگر یکی یا هر دو a, b منفی باشند. ولی ما معمولاً داریم $a, b \geq 1$ ، به هر حال، $\gcd(a, |b|) = \gcd(a, b)$ توجه کنید که $\gcd(0, b) = b$ و $\gcd(b, 0) = b$ ، همچنین، اگر p عدد اول باشد، آنگاه $\gcd(a, p) = \gcd(a, p)$ برابر با ۱ یا p خواهد بود. اگر $1 = \gcd(a, b)$ می‌گوئیم a و b نسبت به هم اول هستند. گزاره‌ی زیر یک نتیجه‌ی مفید است:

گزاره‌ی ۸-۲

فرض کنید a, b اعداد صحیح مثبت باشند. آنگاه اعداد صحیح X, Y وجود خواهند داشت به صورتی که $\gcd(a, b) = Xa + Yb$ به علاوه، $Xa + Yb = \gcd(a, b)$ کوچک‌ترین عدد صحیح مثبتی است که می‌توان آن را به این صورت نشان داد.

اثبات

مجموعه‌ی $\{Xa + Yb \mid X, Y \in \mathbb{Z}\}$ def $I = \{Xa + Yb \mid X, Y \in \mathbb{Z}\}$ را در نظر بگیرید. توجه کنید که $a, b \in I$ و بنابراین، I مطمئناً شامل تعدادی عدد صحیح مثبت خواهد بود. فرض کنید d کوچک‌ترین عدد صحیح مثبت

در ابتدا نشان می‌دهیم که $\gcd(a, b) = d$ از آنجاکه d را می‌توان به صورت $d = Xa + Yb$ برای $X, Y \in \mathbb{Z}$ نوشت (به این دلیل که $d \in I$ ، این امر قضیه را اثبات می‌کند).

به منظور نشان دادن این امر، باید اثبات کنیم که $d|a$ و $d|b$ و اینکه d بزرگترین عدد صحیح با این ویژگی است. در واقع، می‌توانیم نشان دهیم که d تمام عناصر موجود در I را تقسیم می‌کند (تمام عناصر موجود در I بر d بخش پذیر هستند). برای مشاهده‌ی این امر، یک $c \in I$ دلخواه را در نظر گرفته و می‌نویسیم $c = X'a + Y'b$ و $c = Xa + Yb$ با استفاده از تقسیم با باقی مانده (توزیدی) داریم $c = qd + r$ و $0 \leq r < d$. آنگاه

$$(X' - qX)a + (Y' - qY)b \in I.$$

اگر $r \neq 0$ ، این امر نقض کننده‌ی انتخاب d به عنوان کوچک‌ترین عدد صحیح مثبت در I خواهد بود (چون $d < r$ ، بنابراین، r و در نتیجه، c که این امر نشان می‌دهد که d تمام عناصر موجود در I را تقسیم می‌کند).

از آنجاکه $a \in I$ و $b \in I$ ، نتیجه‌ی بالا نشان می‌دهد که $d|a$ و $d|b$ و بنابراین d یک مقسوم علیه مشترک a و b است. اکنون باید نشان دهیم که این عدد، بزرگترین مقسوم علیه مشترک است. فرض کنید یک عدد صحیح $d' > d$ وجود دارد به صورتی که $d'|a$ و $d'|b$ و در این صورت، بر اساس پرورش قلی، $d' | Xa + Yb$ از آنجاکه مورد دوم برابر با d است، این امر به این معناست که $d'|d$ ولی، این غیرممکن خواهد بود اگر d' بزرگتر از d باشد. نتیجه می‌گیریم که d بزرگترین عدد صحیح تقسیم کننده‌ی a و b است و بنابراین $d = \gcd(a, b)$. ■

با دانش a و b «الگوریتم اقلیدسی» را می‌توان برای محاسبه‌ی $\gcd(a, b)$ و زمان چندجمله‌ای مورد استفاده قرار داد. «الگوریتم اقلیدسی تعمیم یافته» را می‌توان همچنین برای محاسبه‌ی X, Y (توسعه شده در گزاره‌ی ۸-۲) در زمان چندجمله‌ای استفاده کرد برای جزییات بیشتر، نک ضمیمه‌ی ب-۱-۲.

گزاره‌ی قلی برای اثبات نتایج اضافی در خصوص بخش پذیری بسیار مفید است. اکنون دو مثال را نشان می‌دهیم.

گزاره‌ی ۸-۳

اگر a و $p = \gcd(a, c)$ باشند، آنگاه $a|c$ ، بنابراین، اگر p عدد اول باشد و a و p آنگاه با $a|p$ یا

می‌دانیم که هر علامت مساوی در این دنباله (و نه فقط آخرین علامت مساوی) به همبستگی به پیمانه‌ی N اشاره دارد.

b وارون‌پذیر است می‌توانیم b^{-1} را نشان دهیم وارون ضربی منحصر به فرد برای b تعریف کنیم که در دامنه $\{1, \dots, N-1\}$ قرار می‌گیرد.

هنگامی که b وارون‌پذیر به پیمانه N است، تقسیم‌پذیر b به پیمانه N را به عنوان ضرب‌پذیر b تعریف می‌کنیم (یعنی، تعریف می‌شود که b وارون‌پذیر باشد، اگر $ab \equiv 1 \pmod N$)، تأکید می‌کنیم که تقسیم‌پذیر b تنها زمانی تعریف می‌شود که b وارون‌پذیر باشد، اگر $ab \equiv cb \pmod N$ و $a \neq c$ و b وارون‌پذیر باشد، آنگاه می‌توانیم هر سمت معادله را بر b تقسیم کنیم (با در واقع هر سمت را در b^{-1} ضرب کنیم) تا رابطه‌ی بعد به دست آید:

$$(ab) \cdot b^{-1} = (cb) \cdot b^{-1} \pmod N \Rightarrow a = c \pmod N.$$

مشاهده می‌کنیم که در این حالت، تقسیم به صورت همورد انتشار عمل می‌کند بنابراین، کار کردن با اعداد صحیح وارون‌پذیر به پیمانه N را اضافی «هزینه» است.

سوال طبیعی این است: کدام اعداد صحیح وارون‌پذیر به پیمانه N یک پیمانه مفروض N هستند؟ با استفاده از گزاره $۲-۸$ می‌توانیم پاسخ کاملی به این سؤال بدهیم:

گزاره ۷-۸

فرض کنید b اعداد صحیحی با $1 \leq b < N$ باشند. آنگاه b وارون‌پذیر به پیمانه N خواهد بود اگر و تنها اگر $\gcd(b, N) = 1$ باشد.

اثبات

فرض کنید b وارون‌پذیر به پیمانه N است و فرض کنید c نشان دهنده وارون آن باشد. از اینجا که $bc \equiv 1 \pmod N$ این امر نشان می‌دهد که $bc - 1 = rN$ برای یک $r \in \mathbb{Z}$ به معنای: به شکل برابر، $bc - rN = 1$ یا $bc - rN$ از اینجا بر اساس گزاره $۲-۸$ ، $\gcd(b, N)$ کوچک‌ترین عدد صحیح مثبتی است که می‌تواند به این شیوه نشان داده شود و هیچ عدد صحیح مثبت کوچک‌تر از ۱ وجود ندارد، این امر نشان می‌دهد که $\gcd(b, N) = 1$.

در مقابل اگر $\gcd(b, N) = 1$ باشد، گزاره $۲-۸$ اعداد صحیح X, Y وجود دارند به صورتی که $1 = Xb + YN$ و کاهش هر سمت این معادله به پیمانه N می‌دهد $Xb \equiv 1 \pmod N$ و مشاهده می‌کنیم که X یک وارون ضربی b است. (در واقع، این مطلب یک الگوریتم کارا برای محاسبه وارون‌ها را می‌دهد) ■

مثال ۸-۸

فرض کنید $b = 11$ و $N = 17$ ، آنگاه $1 = 2 \cdot 17 + (-3) \cdot 11$ و بنابراین، $[-3 \pmod{17}] = 14$ وارون ۱۱ است، می‌توان نشان داد که $14 \cdot 11 \equiv 1 \pmod{17}$.

چون، تقریباً، ضرب و محاسبه وارون‌ها (هنگامی که وجود دارند) به پیمانه N را می‌توان در زمان چندجمله‌ای انجام داد، آن‌ها ضمیمه $۲-۳$ توان رسمی (یعنی محاسبه $[a^b \pmod N]$ برای $b > 0$ یک عدد صحیح) را نیز می‌توان در زمان چندجمله‌ای محاسبه نمود، آن‌ها ضمیمه $۲-۳$.

۲-۱۹- گروه‌ها

فرض کنید G یک مجموعه باشد. یک عمل دوتایی $\circ$ بر روی G در واقع یک تابع $(\circ): G \times G \rightarrow G$ است که دو عنصر از G را به عنوان ورودی دریافت می‌کند اگر $a, b \in G$ ، آنگاه به جای استفاده از نماد سنگین $(a \circ b)$ می‌نویسیم ab .

اکنون مفهوم مهم یک گروه را معرفی می‌کنیم.

تعریف ۸-۸

یک گروه عبارت است از یک مجموعه G به همراه عمل دوتایی $\circ$ که برای آن، شرایط زیر برقرار هستند:

- (بسته بودن) برای تمام $a, b \in G$ داریم $ab \in G$.
- (وجود یک هویتی) یک هویتی $e \in G$ وجود دارد به گونه‌ای که برای تمام $a \in G$ داریم $ea = a$ و $ae = a$.
- (وجود وارون‌ها) برای تمام $a \in G$ یک عنصر $b \in G$ وجود دارد به صورتی که $ab = ba = e$.
- (شریک‌پذیری) برای تمام $a, b, c \in G$ داریم $(a \circ b) \circ c = a \circ (b \circ c)$.

هنگامی که G دارای تعداد عناصر متناهی است، می‌گوییم G متناهی است و فرض می‌کنیم $|G|$ نشان دهنده همرتبه G گروه است (یعنی تعداد عناصر در G).

یک گروه G با عمل «آبجکت» است اگر شرط زیر برقرار باشد:

- (جابه‌جایی) برای تمام $a, b \in G$ داریم $ab = ba$.

هنگامی که عمل دوتایی در G بسته، مجموعه G را تنها یک گروه می‌نامیم.

موزار با گروه‌های آبدی متناهی سروکار داریم. باین حال، دقت داریم که زمانی که یک نتیجه این قضیات را ملزم می‌کند، این مسئله را مشخص کنیم.

شریک‌پذیری به شکل ضمنی نشان می‌دهد که نیازی به آوردن برترتس هنگام نوشتن عبارت‌های توانی نداریم؛ یعنی، نماد $a_1 a_2 \dots a_n$ واضح است چرا که مهم نیست عمل $\circ$ را به چه ترتیبی (آبجکت) می‌کنیم.

مثال ۱۲-۸. یک $N > ۱$ یک عدد صحیح باشد. مجموعه $\{۰, \dots, N-1\}$ نسبت به جمع به پیمانه‌ی N (یعنی زمانی که $[a+b \bmod N] \stackrel{\text{def}}{=} a+b$ است) یک گروه اَبلی از مرتبه‌ی N است. بسته بودن و انیچ هستند. هر یک از این دو خاصیت نتیجه می‌شوند که اعداد صحیح در این ویژگی‌ها حلق هستند. همانی برابر با ۰ است و از آنجا که $0 \bmod N = ۰$ ، $a + (N-1) = a$ نتیجه می‌شود که وارن هر عنصر a برابر با $(N-a) \bmod N$ است. این گروه را به صورت $\mathbb{Z}_N$ نشان می‌دهیم (همچنین گاهی از $\mathbb{Z}_N$ برای نشان دادن مجموعه‌ی $\{۰, \dots, N-1\}$ بدون توجه به عمل گروهی خاص، استفاده می‌کنیم). $\square$

۱۲-۸.۴

برای کنید $\mathbb{Z}_N$ یک گروه باشد و $a, b, c \in \mathbb{Z}_N$ اگر $ac = bc$ ، آنگاه $a = b$ علی‌الخصوص، اگر $ac = c$ آنگاه a همانی در $\mathbb{Z}_N$ است.

اثبات

می‌بینیم که $ac = bc$ با ضرب دو سمت در وارن معکوس c^{-1} از c ، به دست می‌آوریم $a = b$ با جزئیات دقیق:

$$ac = bc \Rightarrow a(cc^{-1}) = b(cc^{-1}) = b(c^{-1} \cdot c) = b \cdot ۱ = b \cdot ۰ = b.$$

■

اثبات بالا را با بحث (قبل از گزاره‌ی ۷-۸) مرتبط با یک قانون حذف برای تقسیم به پیمانه‌ی N مقایسه کنید. همان‌طور که به دلیل شباهت مشخص است، عناصر وارن‌پذیر به پیمانه‌ی N یک گروه را تشکیل می‌دهند. به‌زودی جزئیات بیشتری در مورد این مثال بیان خواهیم کرد.

نشان‌دهی گروهی

تغییر واثباتی توصیف عمل گروهی اعمال‌شده m بار به یک عنصر ثابت g که در آن، m یک عدد صحیح مثبت است، مفید است. هنگام استفاده از نماد جمع، این امر را به صورت $m \cdot g$ یا mg نشان می‌دهیم. یعنی:

$$mg \stackrel{\text{def}}{=} \underbrace{g + \dots + g}_m.$$

برای کنید m یک عدد صحیح است درحالی‌که g یک عنصر گروهی است. بنابراین، mg نشان‌دهنده‌ی عمل گروه اعمال‌شده بر m و g نیست (درواقع، ما در حال کار در گروهی هستیم که

می‌توان نشان داد که عنصر همانی در یک گروه G منحصر به فرد است و بنابراین می‌توانیم به همانی یک گروه اشاره کنیم. همچنین می‌توان نشان داد که هر عنصر g یک گروه دارای یک وارن منحصربه‌فرد است. ن.ک. تمرین ۱۰-۸.

اگر G یک گروه باشد، یک مجموعه‌ی $H \subseteq G$ یک زیرگروه از G است اگر خود H یک گروه را تحت همان عمل مرتب با G ایجاد نماید. به‌منظور بررسی اینکه آیا H یک زیرگروه است، باید بسته بودن، وجود همانی و وارن‌ها و شرکت‌پذیری بر اساسی تعریف ۹-۸ را بررسی کنیم. (درواقع، شرکت‌پذیری- و همین‌طور جابجایی اگر G یک اَبلی باشد- به صورت خودکار از G به ارث برده می‌شود). هر گروه G همواره دارای زیرگروه‌های بدیهی $\{1\}$ و H است، ما H را یک زیرگروه «کوچک» از G می‌نامیم اگر $H \neq G$.

به‌طورکلی، ما از نماد e برای نشان دادن عمل گروهی استفاده نمی‌کنیم، به‌جای آن، ما از نماد جمع یا نماد ضرب بسته به گروه موردبحث، استفاده می‌کنیم. این امر به این معنا نیست که عمل گروه متناظر با جمع یا ضرب عدد صحیح است، این امر تنها یک نماد مفید است. هنگام استفاده از نماد جمع، عمل گروهی اعمال‌شده برای دو عنصر h, g به صورت $h + g$ نشان داده می‌شود و به‌جای $(g+h)$ نشان داده می‌شود. وارن یک عنصر g به صورت g^{-1} نشان داده می‌شود و به‌جای g^{-1} می‌نویسیم g^{-1} . هنگام استفاده از نماد ضرب، عمل گروه اعمال‌شده برای h, g به صورت hg یا فقط gh نشان داده می‌شود و گاهی به‌جای gh^{-1} می‌نویسیم h/g .

در این نقطه، مشاهده‌ی نمادی مثال می‌تواند مفید باشد.

مثال ۱۰-۸

یک مجموعه ممکن است تحت یک عمل، یک گروه باشد ولی تحت عمل دیگر این‌گونه نباشد. برای مثال، مجموعه‌ی اعداد صحیح $\mathbb{Z}$ یک گروه اَبلی جمعی است. همانی عبارت است از عنصر ۰ و هر عدد صحیح g دارای یک وارن $-g$ است. از سوی دیگر، تحت ضرب، گروه نیست به این دلیل که برای مثال، عدد صحیح ۲ دارای یک وارن ضری در میان اعداد صحیح نیست. $\square$

مثال ۱۱-۸

مجموعه‌ی اعداد حقیقی $\mathbb{R}$ یک گروه ضری نیست به این دلیل که ۰ دارای یک وارن ضری نیست. باین‌حال، مجموعه‌ی اعداد حقیقی غیرصفر یک گروه اَبلی ضری با همانی ۱ است.

مثال بعد گروه $\mathbb{Z}_N$ را معرفی می‌کند که به شکل مکرر از آن استفاده خواهیم کرد.

اثبات
این قضیه را تنها برای حالتی اثبات می‌کنیم که g ، اَبلی است (هرچند برای هر گروه مستطقی برقرار است). $g \in G$ را در نظر گرفته و فرض کنید $g_1, \dots, g_m$ عناصر G باشند لذا می‌کنیم که

$$g_1 \cdot g_2 \cdots g_m = (g_1) \cdot (g_2) \cdots (g_m).$$

برای ساده‌ی این امر، توجه کنید که $g_1 g_2 = g_2 g_1$ بر اساس لم ۱۳-۸ نشان می‌دهد که $g_i = g_j$ برای هر کدام از m عنصر در پرانتزها در سمت راست، متناظر است. به این دلیل که دقیقاً m عنصر داریم و وجود دارد m عنصر ضرب شده در هم در سمت راست، دوواقی نامتغی عناصر G به ترتیب در G جایگزین هستند. از آنجا که G اَبلی است، ترتیب ضرب عناصر، اهمیتی ندارد و بنابراین، سمت راست هر دو یکسان است.

پس با سمت چپ است.
پس با استفاده از این واقعیت که G اَبلی است، می‌توانیم تمام رخدادهای g را شیرون کشیده و به دست آوریم:

$$g_1 \cdot g_2 \cdots g_m = (g_1) \cdot (g_2) \cdots (g_m) = g^m \cdot (g_1 \cdot g_2 \cdots g_m).$$

پس با استفاده از لم ۱۳-۸، این امر نشان می‌دهد که $g^m = 1$. ■

یک نتیجه فرعی مهم مبحث بالا این است که می‌توانیم همه یسنادی مرتبگی گروه در توان، کار کنیم:

نتیجه فرعی ۱۵-۸

فرض کنید G یک گروه مستطقی با $|G| = m$ باشد. آنگاه برای هر $g \in G$ و هر عدد صحیح x داریم $g^x = g^{x \bmod m}$

اثبات

$$\begin{aligned} x &= qm + r \quad \text{که در آن، } q, r \text{ اعداد صحیح بوده و } r \text{ آنگاه} \\ g^x &= g^{qm+r} = g^{qm} \cdot g^r = (g^m)^q \cdot g^r = 1^q \cdot g^r = g^r \end{aligned}$$

(با استفاده از قضیه ۱۴-۸، همان‌طور که ادعا شد. ■)

مثال ۱۶-۸

در صورت نوشتن نتیجه فرعی بالا به صورت جمع، این نتیجه بیان می‌کند که اگر g یک عنصر در یک گروه با مرتبگی m باشد، آنگاه $g = [x \bmod m]$ به معنای مثال، گروه با مرتبگی m با مرتبگی m و $g = 11$ را در نظر بگیرید. این نتیجه فرعی بیان می‌کند که

در آن، عمل گروه به صورت جمع نوشته شده است). با این حال، خوشبختانه، نماد «همان‌طور که باید رفتار می‌کنه» بنابراین، برای مثال، اگر $g \in G$ و $m' = m$ اعداد صحیح باشند، آنگاه $(mg) + (m'g) = (m + m')g = (m + m')g = (mm')g = (mm')g = (m + m')g$ در یک گروه اَبلی G با $h \in G$ داریم $(mg) + (mh) = m(g + h)$.

همگام استفاده از نماد ضرب، اعمال عمل گروه به تعداد دفعات m بار بر روی یک عنصر g را با g^m نشان می‌دهیم. یعنی،

$$g^m \stackrel{\text{def}}{=} \underbrace{g \cdot g \cdots g}_{m \text{ times}}$$

تواند آسانی توان رسانی، برقرار هستند: $g^m \cdot g^n = g^{m+n}$ ، آنگاه $(gh)^m = g^m \cdot h^m$ و $g^m = g$ همچنین، اگر G یک گروه اَبلی باشد و $g, h \in G$ ، آنگاه $(gh)^m = g^m \cdot h^m$.

همه این‌ها تنها «ترجمه‌ای» نتایج به دست آمده از پاراگراف قبلی برای حالت گروه‌ها است که به صورت ضربی به جای جمعی نوشته شده‌اند.

نماد بالا به صورت طبیعی به حالتی تقسیم می‌باید که در آن، m برابر با صفر یا یک عدد صحیح منفی است. هنگام استفاده از نماد جمع، $0 \cdot g = 0$ و $(-m) \cdot g = m \cdot (-g)$ را برای m که یک عدد صحیح مثبت است، تعریف می‌کنیم (توجه کنید که در معادله $0 \cdot g = 0$ ، سمت چپ عدد صحیح 0 است در حالی که 0 سمت راست، عنصر همانی در گروه است) توجه کنید که $g - h$ و $h - g$ است و همان‌طور که انتظار می‌رود $(-m)g = -(mg)$ ، هنگام استفاده از نماد ضرب، $1 \cdot g = g$ و $g^{-1} = g^{-1}$ داریم.

فرض کنید $g \in G$ و $0 \leq h < m$ یک عدد صحیح باشد. آنگاه توان رسانی g^h را می‌توان با استفاده از نماد چندجمله‌ای از عمل اصلی گروه در G محاسبه نمود. بنابراین، اگر عمل گروه را بتوان در زمان چندجمله‌ای محاسبه نمود، آنگاه توان رسانی را هم می‌توان این‌گونه محاسبه کرد. این امر در ضمیمه ب-۲-۲ مورد بحث قرار گرفته است.

اکنون موارد کافی برای اثبات نتیجه‌ی شکفت‌آور زیر داریم:

قضیه ۱۴-۸

فرض کنید G یک گروه مستطقی با $|G| = m$ مرتبگی گروه باشد. آنگاه برای هر عنصر $g \in G$ داریم $g^m = 1$.

$15 \cdot 7 \bmod 23 = 11 + 11 = 2 \cdot 11 = 22 \equiv 15 \bmod 23$.
 مطلوب بالا مطابق با این واقعیت است (ن.ک. مثال ۵-۸) که می‌توانیم «گامش داده و سپس ضرب کنیم» به‌جای اینکه مجبور باشیم «ضرب کرده و سپس گامش دهیم».

یک نتیجهی فرعی دیگر که به‌شدت برای کاربردهای رمزنگاری مفید خواهد بود به‌صورت زیر است:

۱۷-۸ نتیجهی فرعی

فرض کنید G یک گروه متناهی با $|G| > 1$ باشد. فرض کنید $e > 0$ یک عدد صحیح باشد و تابع $f_e^G(q) = q^e$ به G به G به G را تعریف کنید. اگر $m = \gcd(e, m) = 1$ است (یعنی یک تک‌گشت دوسویی، به‌علاوه اگر $m \bmod e^{-1} = 1$ باشد، $d = e^{-1} \bmod m$ ، d ، e ، m و d وارون f_e خواهد بود (ترجیح کنید که بر اساس گزاره ۱۷-۸، $\gcd(e, m) = 1$ ، $d = e^{-1} \bmod m$ ، d ، e ، m و d وارون‌پذیر به پیمانی است).

اثبات

از آنجا که G متناهی است، برخی دو این ادعا به‌صورت ضمنی نشان‌دهندهی برخی اول است، بنابراین، فقط باید نشان دهیم که f_e وارون f_d است. این امر صدق می‌کند به این دلیل که برای هر G و داریم:

$$g = f_d(f_e(q)) = f_e(f_d(q)) = f_e(q^d) = (q^d)^e = q^{ed} = q^{e \gcd(e, m)} = q^e.$$

که در آن، تساوی چهارم از نتیجهی فرعی ۱۵-۸ به دست می‌آید. ■

۴-۱۸-۲ گروه $\mathbb{Z}_N^*$

همان‌طور که در مثال ۱۲-۸ بحث شد، مجموعهی $\{1, \dots, N-1\}$ یک گروه جمعی به پیمانی N است. اما می‌توانیم گروهی بر اساس ضرب به پیمانی N تعریف کنیم؟ برای انجام این کار، ما مجبور به حذف آن عناصری در $\mathbb{Z}_N$ خواهیم بود که وارون‌پذیر نیستند، برای مثال، باید 0 را حذف کنیم به این دلیل که هیچ وارون ضربی ندارد. عناصر غیر صفر هم ممکن است وارون‌پذیر نباشند (ن.ک. گزاره ۸-۸).

کدام عناصر $\{1, \dots, N-1\}$ $b \in \{1, \dots, N-1\}$ وارون‌پذیر به پیمانی N هستند؟ گزاره ۷-۸ می‌گوید که این عناصر دقیقاً آن عناصر b هستند که برای آن‌ها $\gcd(b, N) = 1$ است. همچنین در بخش ۸-۱-۲ مشاهده کردیم که هرگاه b وارون‌پذیر است، دارای یک وارون در دامنه‌ی $\{1, \dots, N-1\}$ است. این امر ما را راضی می‌کند که برای هر $N > 1$ ، قورل دهم

$$\mathbb{Z}_N^* \stackrel{\text{def}}{=} \{b \in \{1, \dots, N-1\} \mid \gcd(b, N) = 1\}.$$

یعنی $\mathbb{Z}_N^*$ شامل اعداد صحیح در مجموعه‌ی $\{1, \dots, N-1\}$ است که نسبت به N اول هستند. عمل گروه عبارت است از ضرب به پیمانی N ، یعنی،

$$ab \stackrel{\text{def}}{=} [ab \bmod N]$$

انجام می‌کنیم که $\mathbb{Z}_N^*$ یک گروه ابلی نسبت به این عمل است. از آنجا که 1 همواره در $\mathbb{Z}_N^*$ قرار دارد، متغی است که این مجموعه شامل یک عنصر هوایی است. بحث بالا نشان می‌دهد که هر عنصر در $\mathbb{Z}_N^*$ دارای یک وارون ضربی در همین مجموعه است. جابه‌جایی و شرکت‌پذیری از این واقعیت نتیجه می‌شود این ویژگی‌ها روی اعداد صحیح، برقرار هستند. برای نشان دادن اینکه بسته بودن برقرار است، فرض کنید $a, b \in \mathbb{Z}_N^*$ ، $a, b \in [ab \bmod N]$ دارای وارون $a^{-1} \bmod N$ است، که به این معناست که $1 = \gcd(a, N) = \gcd(ab \bmod N, N)$ و بنابراین، $ab \in \mathbb{Z}_N^*$ به‌طور خلاصه.

گزاره ۱۸-۸

فرض کنید $N > 1$ یک عدد صحیح باشد. آنگاه $\mathbb{Z}_N^*$ یک گروه ابلی ضربی به پیمانی N است.

$\mathbb{Z}_N^*$ را به‌عنوان ترتیبی گروه $\mathbb{Z}_N^*$ تعریف کنید. $\phi(N)$ را تابع فی اولی می‌نامند. مقدار $\phi(N)$ چیست؟ ابتدا حالتی را در نظر بگیرید که در آن $N = p$ عدد اول است. آنگاه، تمام عناصر در $\{1, \dots, p-1\}$ نسبت به p اول هستند و بنابراین، $\phi(p) = |\mathbb{Z}_p^*| = p-1$ در ادامه حالتی را در نظر بگیرید که $N = pq$ که در آن، p, q اعداد اول متمایز هستند. اگر یک عدد صحیح $a \in \{1, \dots, N-1\}$ نسبت به N اول نباشد، آنگاه یا a یا q یا a نمی‌تواند هم بر p و هم بر q بخش‌پذیر باشد به این دلیل که این امر نشان می‌دهد $pqa \equiv 0 \pmod{N}$ (عناصر در $\{1, \dots, N-1\}$ که بر p بخش‌پذیر هستند دقیقاً $(q-1)$ عنصر $p, \dots, p(q-1)$ هستند و عناصر بخش‌پذیر بر q دقیقاً $(p-1)$ عنصر $p, \dots, p(p-1)$ هستند. بنابراین، تعداد عناصر باقی‌مانده (یعنی آن‌هایی که نه بر p و نه بر q بخش‌پذیر هستند) توسط رابطه‌ی زیر داده می‌شود:

$$(N-1) - (q-1) - (p-1) = pq - p - q + 1 = (p-1)(q-1).$$

بنابراین، اثبات کردیم که $\phi(N) = (p-1)(q-1)$ زمانی که N حاصل‌ضرب دو عدد اول متمایز p و q باشد.

در ترمین ۴-۸، از شما خواسته می‌شود که نتیجهی کلی زیر (که پذیرت در ماتی کتاب مورد استفاده قرار می‌گیرد) را اثبات کنید.

قضیه ۱۹-۸

فرض کنید $N = \prod_{i=1}^r p_i^{e_i}$ ، که در آن، $\{p_i\}$ اعداد اول متمایز بوده و $e_i \geq 1$ ، آنگاه $\phi(N) = \prod_{i=1}^r p_i^{e_i-1} (p_i - 1)$.

در اصل، یک پیکریخی از G به H در واقع تغییر نام عناصر G به عنوان عناصر H است توجه کنید که اگر G متناهی و $H = G$ باشد، آنگاه H باید متناهی و دارای اندازه یکسان با G باشد همچنین، اگر یک پیکریخی f از G به H وجود داشته باشد، آنگاه f^{-1} یک پیکریخی از H به G است. باین حال، ممکن است f قابل معکوسی کارا باشد در حالی که f^{-1} این گونه نباشد (با همگی).

هدف این بخش، استفاده از زبان پیکریخی ها برای درک بهتر ساختار گروهی Z_N و Z_N^* در حالتی است که $N = pq$ حاصل ضرب دو عدد اول متمایز باشد در ابتدا باید مفهوم ضرب مستقیم $G \times H$ را معرفی کنیم. با داشتن گروه های H و G ، با عمل گروهی G و H به ترتیب یک گروه جدید $G \times H$ (ضرب مستقیم G و H) را به صورت زیر تعریف می کنیم. عناصر $G \times H$ عبارتند از زوج های مرتب (g, h) با $g \in G$ و $h \in H$ و g و h به ترتیب n عنصر و m دارای H و G هستند. $G \times H$ دارای $n \cdot m$ عنصر خواهد بود. عمل گروه $G \times H$ به صورت مولفه ای عمل می شود، یعنی:

$$(g_1, h_1) \circ (g_2, h_2) = (g_1 \circ g_2, h_1 \circ h_2).$$

نتیج دانی اینکه $G \times H$ واقعا یک گروه است را به عنوان تمرین ۸-۸، باقی می گذاریم. نکته بالا را می توان به ضرب های مستقیم بیش از دو گروه به شکل طبیعی تعمیم داد. هر چند در ادامه نیازی به این نداریم.

الگوریتمی برای تعریف پیکریخی های چینی را بیان و اثبات کنیم.

قضیه ۸-۲۴ (قضیه ی باقیمانده ی چینی)

فرض کنید p, q که در آن $1 < pq$ نسبت به هم اول هستند. آنگاه

$$Z_N = Z_p \times Z_q \quad \text{و} \quad Z_N^* = Z_p^* \times Z_q^*$$

به علاوه، فرض کنید f تابع نگاشت عناصر $\{1, \dots, N-1\}$ به زوج های (x_p, x_q) با $x_p \in Z_p$ و $x_q \in Z_q$ و $\{1, \dots, q-1\}$ و $\{1, \dots, p-1\}$ است.

$$f(x) \stackrel{\text{def}}{=} (x \bmod p, x \bmod q).$$

آنگاه، یک پیکریخی از Z_N به $Z_p \times Z_q$ است و محدودیت f به Z_N^* یک پیکریخی از Z_N^* به $Z_p^* \times Z_q^*$ است.

اثبات

برای هر $x \in Z_N$ خروجی $f(x)$ یک زوج عنصر (x_p, x_q) با $x_p \in Z_p$ و $x_q \in Z_q$ است. اما می کنیم که اگر $x \in Z_N^*$ آنگاه $x \in Z_p \times Z_q$ خواهد بود در واقع، اگر $x_p \in Z_p^*$ آنگاه این امر به $\gcd(x, p) = 1$ و $\gcd(x, q) = 1$ در این حالت، $\gcd(x, pq) = 1$ و $x \in Z_N^*$ است.

مثال ۸-۲۰
فرض دهید $50 \cdot 2 = 100$ ، آنگاه $N = 100$ ، $Z_N = \{1, 2, \dots, 99\}$ و $Z_N^* = \{1, 3, 7, 9, 11, 13, \dots, 99\}$ و $\phi(100) = 40$ و $\phi(50) = 20$ و $\phi(2) = 1$ است به این دلیل که $100 \bmod 15 = 10$ و $50 \bmod 15 = 5$ و $2 \bmod 15 = 2$ و $1 \bmod 15 = 1$ است.

نتیج دانی که Z_N^* یک گروه با مرتبه $\phi(N)$ است، نتایج فرعی زیر به سادگی از قضیه ی ۸-۱۴ و نتیجه ی فرعی ۸-۱۷ به دست می آیند:

نتیجه ی فرعی ۸-۲۱

عدد صحیح دلخواه $1 < N$ و $a \in Z_N^*$ را در نظر بگیرید. آنگاه

$$a^{\phi(N)} = 1 \bmod N.$$

برای حالت خاصی که $N = p$ اول است و $1 < p$ داریم:

$$a^{p-1} = 1 \bmod p$$

نتیجه فرعی ۸-۲۲

$1 < N$ را در نظر بگیرید. برای عدد صحیح $0 < e$ ، $Z_N^* \rightarrow Z_N^*$ ، $f: Z_N^* \rightarrow Z_N^*$ را توسط $f(x) = [x^e \bmod N]$ تعریف کنید. اگر e نسبت به $\phi(N)$ اول باشد، آنگاه f یک جایگشت است. به علاوه، اگر $d = \phi(N)$ و $e^{-1} \bmod \phi(N)$ آنگاه f^{-1} وارون f است.

۸-۲۳ تعریف

دو گروه، یک ریخت هستند اگر دارای ساختار مشابه یکسانی باشند. از دیدگاه ریاضیاتی، یک پیکریخی از یک گروه G یک روش فرعی ولی معادل برای تفکر در مورد G را می دهد. از دیدگاه محاسباتی، یک پیکریخی روش متفاوتی برای نمایش عناصر در G فراهم می کند که اغلب می تواند یک تأثیر چشم گیر بر روی الزامات الگوریتمی داشته باشد.

تعریف ۸-۲۳

فرض کنید H ، H گروه ای بر اساس عمل $\circ$ و H به ترتیب باشند. یک تابع $f: G \rightarrow H$ یک پیکریخی از G به H است اگر:

$$(1) \quad f(g_1 \circ g_2) = f(g_1) \circ f(g_2) \quad \text{و} \quad (2) \quad \text{برای تمام } g_1, g_2 \in G, f(g_1) \circ f(g_2) = f(g_1 \circ g_2)$$

اگر یک پیکریخی از G به H وجود داشته باشد، آنگاه می گوئیم که این گروه ها پیکریخت هستند و می نویسیم $G \cong H$.

$$\text{Pr}[W - \text{Factor}_n(n) = 1] \leq \text{negl}(n)$$

برای هر الگوریتم p, PPT ، δ ، بازیگر است ابتدا اینکه عدد N در آزمایش بالا به احتمال $2^{-\delta}$ درج است (این اتفاق زمانی می افتد که x_1 یا x_2 زوج باشد)، البته برای δ تجربی N در این حالت ساده است. با این حال می توانیم کار δ را دشوارتر کنیم به این صورت که δ را ملزم کنیم که اندازه صحیح 2^{δ} یا $2^{\delta+1}$ یا طول n را به عنوان خروجی بدهد. همچنین این حالت یوزر است که x_1 یا x_2 (و بنابراین N) ممکن است دارای عوامل اول کوچکی باشند که همچنان مسئله ای بتوان آن را یافت برای کاربردهای رمزنگاری، یا باید منابع این امر شوم.

همان طور که این بحث نشان می دهد، سخت ترین اعداد برای تجزیه اعداد هستند که تنها عوامل اول بزرگ دارند. این مسئله نشان می دهد که باید آزمایش بالا را بازتعریف کنیم به صورتی که x_1 یا x_2 اعداد اول n تجربی تصادفی باشند و نه اعداد صحیح n تجربی تصادفی و در واقع چنین آزمایشی هنگام تعریف رسمی فرضی تجزیه در بخش ۳-۲۸ مورد استفاده قرار می گیرد. با این حال، برای اینکه این آزمایش در زمینه رمزنگاری مفید باشد، ضروری است که قادر به تولید اعداد اول n تجربی تصادفی به صورت کارا باشیم. این مسئله موضوع دو بخش بعدی است.

۳-۲۸- تولید اعداد اول تصادفی

یک روش و طبیعی برای تولید یک عدد اول n تجربی تصادفی عبارت است از انتخاب مکرر اعداد صحیح n تجربی تصادفی تا زمانی که عددی بیابیم که اول باشد، این کار را حداکثر t بار یا تا زمانی که به موفقیت برسیم، تکرار می کنیم. برای یک توصیف سطح بالا از این فرایند یک الگوریتم δ -

۲۱

```

الگوریتم ۳۱-۸
تولید یک عدد اول تصادفی - جازوب سطح بالا
ورودی: طول  $n$  بازاست  $t$ 
خروجی: یک عدد اول  $n$  تجربی یکواخت
for  $i = 1$  to  $t$ :
     $p \leftarrow (0, 1)^{n-1}$ 
    if  $p$  is prime return  $p$ 
return fail

```

نوع کند که این الگوریتم، خروجی را وادار می کند که یک عدد صحیح با طول دقیقاً n باشد (به جای طول حداکثر n) به این صورت که بیت درجه بالایی p را برابر با «۱» قرار می دهد. قرار داده ما در سر

بنظر این، $21 = [-2 \cdot 7 \bmod 25]$ و $15 = [3 \cdot 5 \bmod 25]$ ، $1_q = 1$ (می توانیم بررسی کنیم که این ها درست هستند برای مثال، برای $21 = 1_p$ می توانیم بررسی کنیم که $1 = [5 \bmod 5]$ و $21 = [21 \bmod 5]$) با استفاده از این مقادیر، در ادامه می توانیم محاسبه کنیم

$$\begin{aligned} (4^2) &= 4 \cdot (1, 0) + 3 \cdot (0, 1) \\ &\Rightarrow [4 \cdot 1_p + 3 \cdot 1_q \bmod 25] \\ &= [4 \cdot 1 + 3 \cdot 0 \bmod 25] = 4. \end{aligned}$$

از اینجا $4 \bmod 5 = 4$ و $24 = 3 \bmod 7$ و $24 = 4 \bmod 7$ ، این در واقع نتیجه صحیح است. ۵

۳-۲۸- اعداد اول تجزیه و RSA

در این بخش، اولین مثال های مسائل نظریه اعداد را نشان می دهیم که حدس زده می شود، مسئله هستند، یا بعضی در خصوص یکی از قدیمی ترین مسائل شروع می کنیم: تجزیه عدد صحیح یا فقط تجزیه.

با داشتن یک عدد صحیح مرکب N ، مسئله تجزیه این است که اعداد صحیح $1 < p, q$ را بیابیم به صورتی که $pq = N$ باشد. تجزیه یک مثال کلاسیک از یک مسئله سخت است، هم به این دلیل که توصیف آن بسیار ساده است و هم به این دلیل که به عنوان یک مسئله محاسباتی سخت به مدت طولانی شناخته شده است (حتی قبل از استفاده از آن در رمزنگاری). این مسئله را می توان در زمان همایی $O(\sqrt{N} \cdot \text{polylog}(N))$ یا بیشتر پیچیدگی یا خیر برای $p = 2, \dots, \lfloor \sqrt{N} \rfloor$ (این روش نیازمند $\sqrt{N}$ فراگیر است که هر کدام $\text{polylog}(N) = \log(N)$ نیاز دارد) این روش همواره موفق می شود به این دلیل که هر چند بزرگ ترین عامل اول N ممکن است به بزرگی $N^{1/2}$ باشد، کوچک ترین عامل اول N می تواند حداکثر $\lfloor \sqrt{N} \rfloor$ باشد. هر چند الگوریتم های با زمان اجرای پیاپی، شناخته شده اند (ن ک فصل نهم)، هیچ الگوریتم زمان چندجمله ای برای تجزیه نشان داده نشده است هر چند سال ها در این زمینه تلاش شده است.

آمایش زیر را برای یک الگوریتم مفروض δ و بازاست t در نظر بگیرید:

- ۱) انتخاب دو عدد صحیح n تجربی یکواخت x_1, x_2 .
- ۲) δ مقدار N را دریافت کرده و خروجی $1 < x_1, x_2$ را از آن ها می کند.
- ۳) خروجی آزمایش برابر با ۱ تعریف می شود اگر $x_1 \cdot x_2 = N$ ، در غیر این صورت، خروجی ۰ است.

هم اکنون بکنیم که باور بر آن است که مسئله تجزیه، سخت است، یا این امر به این معنا است که

بعد اول) را ارائه می‌کرد. از سوی دیگر، اگر p مرکب بود، آنگاه الگوریتم تقریباً همواره خروجی $composite$ (عدد مرکب) را ارائه می‌کرد ولی ممکن بود پاسخ اشتباه $(prime)$ را با احتمال ناچیز بر اساس طول p ارائه نماید. بهین‌الدینگر، اگر الگوریتم خروجی $composite$ را ارائه کند، آنگاه p قطعاً مرکب است ولی اگر خروجی $prime$ را ارائه کند، آنگاه به احتمال بالا p اول است ولی این امکان هم وجود دارد که اشتباهی رخ دهد و p درواقع مرکب باشد.^{۳۲}

مکالم استفاده از آزمون اول بودن تصادفی شدنی از این دست در الگوریتم 2^{n-1} الگوریتم تولید عدد اول که قبلاً نشان داده شد، خروجی الگوریتم یک عدد اول بگوشاید با طول مطلوب خواهد بود ولی که الگوریتم خروجی $fail$ را ارائه کند و آزمون اول بودن تصادفی شده هم طی اجرای مابقی که الگوریتم دچار خطا نشود این امر به این معناست که یک منبع خطای اضافی (علاوه بر احتمال الگوریتم) افزوده می‌شود و اکنون الگوریتم ممکن است یک عدد مرکب را به اشتباه به عنوان خروجی ارائه کند. از آنجاکه می‌توانیم تضمین کنیم که این اتفاق تنها با احتمال ناچیز رخ دهد، این احتمال بعید باعث نگرانی در عمل نمی‌شود و می‌توان بدون دقت، آن را نادیده گرفت.

یک الگوریتم زمان چندجمله‌ای قطعی برای آزمون اول بودن در یک نتیجه‌ی یونین در سال 2002 نشان داده شد. این الگوریتم، هرچند در زمان چندجمله‌ای اجرا می‌شود، کندتر از آزمون‌های احتمالاتی مورد اشاره در بالا است. به همین دلیل، آزمون‌های احتمالاتی اول بودن همچنان به صورت انحصاری در عمل برای تولید اعداد اول بزرگ مورد استفاده قرار می‌گیرند.

در بخش $2-8$ ، یکی از معمول‌ترین آزمون‌های احتمالاتی اول بودن را توصیف و تحلیل می‌کنیم. الگوریتم میلر-رابین، این الگوریتم، دو ورودی را می‌پذیرد: یک عدد صحیح p و یک پارامتر t (به صورت یک عدد) که احتمال خطا را تعیین می‌کند. الگوریتم میلر-رابین در زمان چندجمله‌ای در $O(p^2 \log t)$ اجرا می‌شود و در قفسه‌ی زیر صدق می‌کند:

قضیه $23-8$

اگر p اول باشد، آنگاه آزمون میلر-رابین همواره خروجی $prime$ را می‌دهد. اگر p مرکب باشد، الگوریتم خروجی $composite$ را ارائه می‌کند مگر با احتمال حداکثر 3^{-t} .

جمع‌بندی

با توجه به بحث بالا اکنون می‌توانیم یک الگوریتم تولید عدد اول زمان چندجمله‌ای را توصیف کنیم که با دریافت ورودی n ، یک عدد اول n یابری را خروجی می‌دهد مگر با احتمال ناچیز در n به علاوه

^{۳۲} همچنین آزمون‌های اول بودن احتمالاتی وجود دارند که به تئوری محلی عمل می‌کنند. آن‌ها همواره به درستی اعداد مرکب را شناسایی می‌کنند ولی گاهی هنگام دریافت یک عدد اول به عنوان ورودی، دچار اشتباه می‌شوند. ما الگوریتم‌های از این دست را مدنظر قرار نمی‌دهیم.

این کتاب این است که یک «عدد صحیح با طول n » به معنای عدد صحیحی است که نمایش دودویی آن با هر چه کمترین بیت برابر با «۱» دقیقاً دارای طول n بیت است.

با داشتن روشی برای تعیین اینکه آیا یک عدد صحیح p مفروض، اول هست یا خیر، الگوریتم بالا یک عدد اول n یابری بگوشاید را مشروط به این رخداد ارائه می‌کند که $fail$ را به عنوان خروجی ارائه نکند. احتمال اینکه این الگوریتم خروجی $fail$ را ارائه کند به t بستگی دارد و برای اهداف ما، می‌خواهیم t را به صورتی تعیین کنیم که یک احتمال شکست به دست آوریم که بر اساس n ناچیز باشد. برای نشان دادن اینکه الگوریتم $23-8$ منجر به یک الگوریتم کارا (یعنی، زمان چندجمله‌ای در n) برای تولید اعداد اول می‌شود، نیازمند درک بهتری از دو مسئله هستیم: (۱) احتمال اینکه یک عدد صحیح n یابری بگوشاید، یک عدد اول باشد و (۲) چگونگی آزمون کارا اینکه آیا یک عدد صحیح مفروض p اول هست یا خیر. ما این مسائل را اکنون به صورت مختصر مورد بحث قرار می‌دهیم و بررسی دقیق‌تر دومین موضوع را در بخش بعدی انجام خواهیم داد.

توزیع اعداد اول

هفتصدی عدد اول^{۳۳}، یک نتیجه‌ی مهم در ریاضیات، کران‌های نسبتاً دقیقی برای نسبت اعداد صحیح با طول مفروضی که عدد اول هستند، می‌دهد. برای اهداف ما، تنها به یک گونه‌ی ضعیف و یکسویه از این نتیجه نیاز داریم که در اینجا آن را اثبات نمی‌کنیم:

قضیه $23-8$ (اصل بزرگانه)

برای هر $n > 1$ ، نسبت عدد صحیح n یابری که اول هستند حداقل $1/2n$ است.

با بازگشت به رویکرد مورد استفاده برای تولید اعداد اول که در بالا توصیف شد، این امر نشان می‌دهد که اگر قرار دهیم $t = 2n^2$ ، آنگاه احتمال اینکه یک عدد اول در تمام t تکرار الگوریتم، انتخاب نشود حداکثر برابر است با

$$e^{-n} = (e^{-1})^n \leq \left(1 - \frac{1}{2n}\right)^n \leq \left(1 - \frac{1}{2n}\right)^{2n} = \left(1 - \frac{1}{2n}\right)^{t}$$

(با استفاده از نامساوی القیاس^{۳۴})، که در n ناچیز است. بنابراین، با استفاده از $poly(n)$ تکرار الگوریتم به دست می‌آوریم که برای آن، احتمال خروجی $fail$ ناچیز در n است. (نتایج دقیق‌تر از قضیه $23-8$ وجود دارند و بنابراین در عمل، حتی تکرارهای کمتری هم مورد نیاز است.)

آزمون اول بودن

مسئله‌ی تعیین کارا اینکه آیا یک عدد مفروض، اول هست یا خیر، تاریخچه‌ی طولانی دارد. در دهه‌ی ۱۹۷۰، اولین الگوریتم‌های کارا برای آزمون اول بودن ایجاد شدند. این الگوریتم‌ها تصادفی بوده و دارای تضمین زیر بودند: اگر ورودی p یک عدد اول بوده، الگوریتم همواره خروجی $prime$

مشروط به اینکه خروجی p اول باشد. p یک عدد اول n نسبتی دارای توزیع یکنواخت خواهد بود. کل ربه در الگوریتم ۳۴-۸ توصیف شده است.

الگوریتم ۳۴-۸
تولید یک عدد اول تصادفی
ورودی: طول n
خروجی: یک عدد اول n نسبتی یکنواخت
 $for\ t = 1\ to\ 2^n$
 $\quad p' \leftarrow \{0, 1\}^{n-1}$
 $\quad p = 1 || p'$
اجرای آزمون میلر-رابین بر روی ورودی p و پارامتر 2^n
اگر خروجی «prime» است، p را بازگردان.
return fail

تولید اعداد اول با شکل خاص

گاهی مطلوب است تولید یک عدد اول n نسبتی تصادفی با یک شکل خاص، مثلاً، صافی در شرط $4 \bmod 3 = p$ یا به صورتی که $1 + 2q = p$ باشد که در آن، q هم اول است (دهی از نوع دوم را اعداد اول فوی می‌نامند). در این حالت، تغییرات مناسب در الگوریتم تولید عدد اول نشان داده‌اند. در بالا را می‌توان استفاده کرد. برای مثال، به‌منظور به دست آوردن یک عدد اول به شکل $p = 1 + 2q$ الگوریتم را تغییر می‌دهیم تا یک عدد اول تصادفی q را تولید کند. $p = 1 + 2q$ و سپس، p را به‌عنوان خروجی ارائه دهد. اگر p هم یک عدد اول بود، در حالی که این الگوریتم‌های تغییر یافته در عمل به‌خوبی کار می‌کنند، اثبات‌های استوار برای اینکه این الگوریتم‌ها در زمان چند-جمله ای اجرا می‌شوند و تنها با احتمال ناچیز، شکست می‌خورند، پیچیده‌تر هستند و در بعضی از موارد به حسابات اثبات نشده‌ی نظریه اعداد در رابطه با توکم اعداد اول با شکل خاص، تکیه دارند، بررسی دقیق این مسائل خارج از گستره‌ی این کتاب است و تنها وجود الگوریتم‌های تولید عدد اول مناسب را در هنگام نیاز، فرض می‌کنیم.

۳۴-۸-۵ آزمون اول بودن

اکنون آزمون اول بودن میلر-رابین را توصیف کرده و قضیه‌ی ۳۴-۸ را اثبات می‌کنیم. ما به مطالب ارائه‌شده در بخش ۳۴-۸-۱ تکیه می‌کنیم. این مطالب به‌صورت مستقیم در مابقی کتاب مورد استفاده قرار نمی‌گیرد.

تکلیف الگوریتم میلر-رابین باقی‌مان یک ویژگی است که اعداد اول و مرکب را از هم متمایز نماید. فرض کنید N نشان دهنده‌ی عدد ورودی برای آزمون باشد. با مشاهده‌ی زیر شروع می‌کنیم. اگر N اول باشد، $1 = N - 1$ و برای هر $a \in \{1, \dots, N - 1\}$ بر اساس قضیه‌ی ۱۴-۸ داریم $a^{N-1} \equiv 1 \bmod N$. این امر نشان می‌دهد که آزمون اینکه آیا N اول است یا خیر با انتخاب یک عنصر یکنواخت a و بررسی اینکه $1 \equiv a^{N-1} \bmod N$ امکان‌پذیر است. اگر $a^{N-1} \not\equiv 1 \bmod N$ ، آنگاه N نمی‌تواند اول باشد. در مقابل، ممکن است امیدوار باشیم که اگر N اول نیست، پس یک شانس منفی وجود دارد که a را با $a \bmod N \neq 1$ انتخاب کنیم که $a^{N-1} \equiv 1 \bmod N$ باشد. با تکرار این آزمون به دفعات زیاد، می‌توانیم با اطمینان بالا تعیین کنیم که آیا N اول هست یا خیر. رویکرد بالا در الگوریتم ۳۵-۸ نشان داده شده است. (خاوری می‌کنیم که توان رفتی به پیمانی N و محاسبه‌ی بزرگ‌ترین به‌صورتی‌های مشترک را می‌توان در زمان چندجمله‌ای انجام داد. انتخاب یک عنصر یکنواخت از $\{1, \dots, N - 1\}$ را نیز می‌توان در زمان چندجمله‌ای انجام داد. یک ضمیمه‌ی ب-۲)

الگوریتم ۳۵-۸
آزمون اول بودن-تلاش اول
ورودی: عدد صحیح N و پارامتر 2^t
خروجی: یک تصمیم در خصوص اینکه N اول است یا مرکب
 $for\ t = 1\ to\ t:$
 $\quad a \leftarrow \{1, \dots, N - 1\}$
 $\quad if\ a^{N-1} \not\equiv 1 \bmod N\ return\ "composite"$
return prime

اگر N اول باشد، الگوریتم همواره خروجی «prime» ارائه می‌کند. اگر N مرکب باشد، الگوریتم خروجی «composite» را ارائه می‌کند. اگر در یکی از تکرارها یک $a \in \{1, \dots, N - 1\}$ پیدا به صورتی که $a^{N-1} \not\equiv 1 \bmod N$ ، توجه کنید که اگر $a \in \mathbb{Z}_N^*$ ، آنگاه $a \bmod N \neq 1$ می‌تواند برابر یا ۱ باشد. بنابراین، $\gcd(a, N) \neq 1$ و $\gcd(a^{N-1}, N) \neq 1$ و بنابراین $a \in \mathbb{Z}_N$ محدود می‌کنیم. ما هر a اینجایی که $a \bmod N \neq 1$ را یک شاهد برای مرکب بودن N یا تنها یک شاهد «مشاهده» می‌نامیم. ممکن است امید داشته باشیم که زمانی که N مرکب است، تعداد زیادی شاهد وجود داشته باشد و بنابراین، الگوریتم چنین شاهده‌ی را با احتمال بالا «پیدا» کند. این برداشت شهودی، صحیح است به شرطی که حداقل یک شاهد وجود داشته باشد قبل از اثبات این امر، ما به دو لم نظریه‌ی گروهی نیاز داریم.

گزاره ۳۶-۸

فرض کنید G یک گروه متناهی و $H \subseteq G$ باشد. فرض کنید H غیر تهی است و برای همی $a, b \in H$ داریم $ab \in H$ ، آنکه H یک زیر گروه از G است.

اثبات

باید نشان دهیم که H در تمام شرایط تعریف ۱-۸ صدق می کند. بر اساس فرض، H تحت عمل گروه بسته است. شرکت پذیری در H به صورت خودکار از G به ارث می رود. فرض کنید $m = |G|$ در اینجا از این واقعیت استفاده می کنیم که $a \in H$ متناهی است و یک عنصر داخل خواه $a \in H$ را در نظر بگیریم. بسته بودن H به این معنا است که H شامل $a^{-1} = a^{m-1}$ و همین طور $a^m = 1$ است. بنابراین، H شامل وارن هر کدام از عناصر خود و همین طور عنصر همانی است. ■

۳۷-۸

فرض کنید H یک زیر گروه اکید از یک گروه متناهی G باشد (یعنی $G \neq H$). آنکه $|H| \leq |G|/2$

اثبات

فرض کنید $\bar{H}$ یک عنصر از G باشد که در H نیست، از اینجا که $H \neq G$ می دانیم چنین $\bar{H}$ وجود دارد. مجموعه $\{h\bar{h} \mid h \in H\} = \bar{H}$ را در نظر بگیرید. نشان می دهیم که $(1) \quad |\bar{H}| = |H|$ و (2) هر عنصر از $\bar{H}$ خارج از H قرار می گیرد. یعنی، اشتراک H و $\bar{H}$ تهی است. از اینجا که H و $\bar{H}$ زیر مجموعه های G هستند، این موارد نشان می دهند که $|\bar{H}| = |H|$ و $|\bar{H}| \geq |G|$ که کم را اثبات می کند.

برای هر $h_1, h_2 \in H$ ، اگر $h_1 \bar{h}_2 = \bar{h}$ ، آنکه، با ضرب $\bar{h}^{-1}$ در هر طرف داریم $h_1 = \bar{h} h_2$. این امر نشان می دهد که هر عنصر متناظر $h \in H$ متناظر با یک عنصر متناظر $\bar{h} \in H$ است، که (1) را اثبات می کند.

برای رسیدن به حالت نقیض، فرض کنید $\bar{h} h_1 \in H$ برای یک h_1 مفروض. این امر به این معناست که $\bar{h} h_1 = h_2$ برای یک $h_2 \in H$ و بنابراین، $\bar{h} = h_2 h_1^{-1} \in H$ چرا که $h_2 h_1^{-1}$ یک زیر گروه است و $\bar{h} \in H$ و $h_1^{-1} \in H$ ولی این امر به این معناست که $\bar{h} \in H$ که نقیض کندهی شوی انتخاب $\bar{h}$ است. این امر، (2) را اثبات کرده و اثبات این کم را کامل می کند. ■

نقصی زیر را را قادر می سازد تا الگوریتم ارائه شده ی قبلی را مورد تحلیل قرار دهیم.

تفسیر ۳۸-۸

این بخش، کتید فرض کنید یک شاهد برای مرکب بودن N وجود دارد. آنکه، حداقل نمی از عناصر $\bar{a}$ شاهد مرکب بودن N هستند.

اثبات

فرض کنید Bad مجموعه ای از عناصر در $\bar{a}_N$ باشد که شاهد نیستند، یعنی، $a \in Bad$ به این معنا است که $a^{N-1} \equiv 1 \pmod{N}$ مشخص است که $a \in Bad$ اگر $1 \in Bad$ ، $a, b \in Bad$ ، آنکه $(ab)^{N-1} \equiv 1 \pmod{N}$ ، $a^{N-1} \equiv 1 \pmod{N}$ و $b^{N-1} \equiv 1 \pmod{N}$ ، از اینجا که $ab \in Bad$ بر اساس کم ۳۶-۸، نتیجه می گیریم که Bad یک زیر گروه از $\bar{a}_N$ است. از اینجا که 1 در اساس فرضی، حداقل، یک شاهد وجود دارد Bad یک زیر گروه اکید از $\bar{a}_N$ است. آنکه کم ۳۷-۸ نشان می دهد که $|Bad| \leq |\bar{a}_N|/2$ که نشان می دهد حداقل نمی از عناصر $\bar{a}_N$ در Bad نیستند (و بنابراین، شاهد هستند). ■

فرض کنید N مرکب باشد. اگر یک شاهد برای مرکب بودن N وجود داشته باشد، آنکه حداقل $|\bar{a}_N|/2$ شاهد وجود خواهد داشت. بنابراین، احتمال اینکه یک شاهد یا یک عنصر که در $\bar{a}_N$ باشد را در هر تکرار مفروض الگوریتم بیایم حداقل برابر با $1/2$ است و بنابراین، احتمال اینکه الگوریتم مع شاددی در هیچ کدام از t تکرار نیابد (و بنابراین احتمال اینکه الگوریتم به اشتباه خروجی prime را ارائه کند) حداکثر 2^{-t} است.

مشاهده بحث بالا یک راه حل کامل را به دست نمی دهد چرا که تعداد بی نهایت عدد مرکب N وجود دارد که دارای هیچ شاددی برای مرکب بودن خود نیستند. چنین مقادیر N به عنوان «اعداد کارمایکل» شناخته می شوند. بحث دقیق در خصوص این اعداد خارج از گسترده ی این کتاب است.

خوبیخانه می توان نشان داد یک تبدیل در ازمون بالا برای تمام ah عمل می کند. فرض کنید $N = 2^u$ که در آن، u مفرد بوده و $1 \leq r \leq N$ (به سادگی می توان 3 و u را با دانستن N مطابقت نمود). همچنین، محدود کردن به $1 \leq r \leq N$ مفرد است که N مفرد است، ولی هنگامی که N زوج باشد، ازمون اول بودن آن ساده است (الگوریتم نشان داده شده در بالا تنها این مسئله را ازمون می کند که آیا $1 \pmod{N} = a^{N-1} = a^{2^u-1}$ ، یک الگوریتم دقیق تر به دنباله ی $1, 2, \dots, u$ (همگی به بیانه ی N) نگاه می کند. هر جمله در این دنباله عبارت است از مربع جملی قبلی. بنابراین، اگر یک مقدار مفروض برابر با ± 1 باشد، آنکه تمام مقادیر بعدی برابر با 1 خواهد بود.

فرض کنید که $a \in \bar{a}_N$ یک «شاهد قوی برای مرکب بودن N است (یا به شکل مشابه، یک شاهد قوی است) اگر $(1) \pmod{N} = a^{\pm 1}$ و $(2) \pmod{N} = a^{\pm 1} \neq a^{2^u}$ برای تمام $1 \leq i \leq u$.

توجه کنید که زمانی که یک عنصر a یک شاهد قوی نباشد، آنگاه دنباله $(a^u, a^{2u}, \dots, a^{r^u})$ (همگی به پیمانی N) یکی از اشکال زیر را به خود می‌گیرد:

$$(1, \dots, 1) \quad \text{یا} \quad (1, \dots, 1, -1, \dots, -1, \dots, 1)$$

که در آن، $*$ یک جملی دلخواه است. اگر a یک شاهد قوی نباشد، آنگاه داریم $a^{r^u} = \pm 1 \pmod N$

$$a^{r^u} = a^{r^{u-1}} = a^{r^{u-2}} = \dots = a^{r^0} = 1 \pmod N.$$

بنابراین، a نیز یک شاهد برای مرکب بودن N نیست. به بیان دیگر، اگر a یک شاهد باشد، آنگاه یک شاهد قوی هم هست و بنابراین در واقع تنها ممکن است که a r شود قوی بیشتری نسبت به a شود و a داشته باشد.

در ابتدا نشان می‌دهیم که اگر N اول باشد، آنگاه یک شاهد قوی برای مرکب بودن N وجود نخواهد داشت. برای انجام این کار، به a ساده‌ی زیر (که یک حالت خاص از گزاره‌ی $1 \leq a < N$ است که در فصل سیزدهم اثبات خواهد شد) نگاه می‌کنیم:

۲۹-۸-۱

فرض کنید $x \in \mathbb{Z}_N^*$ یک ریشی دوم 1 به پیمانی N است اگر $x^2 = 1 \pmod N$ یک عدد اول فرد باشد. آنگاه تنها ریشه‌های دوم 1 به پیمانی N عبارتند از $[\pm 1 \pmod N]$.

اثبات

فرض کنید $x^2 = 1 \pmod N$ یا $x \in \{1, \dots, N-1\}$ آنگاه $x - 1 = (x+1)(x-1) = x^2 - 1 = 0 \pmod N$ که نشان می‌دهد $N \mid (x+1)(x-1)$ بر اساس گزاره‌ی ۸-۳-۸ این امر تنها زمانی می‌تواند رخ دهد که $x = \pm 1 \pmod N$ ■

فرض کنید N یک عدد اول فرد باشد و $a \in \mathbb{Z}_N^*$ دلخواه را مشخص کنید. فرض کنید $a^2 = 1$ عبارت باشد از مقدار مینیمی که برای آن، $a^{2u} = 1 \pmod N$ آنگاه $a^{2u} = a^{N-1} = 1 \pmod N$ می‌دهیم که چنین $r \leq 2$ وجود دارد. اگر $a^2 = 1 \pmod N$ و $a \neq \pm 1$ یک شاهد قوی نیست. در غیر این صورت،

$$(a^{2^{u-1}})^2 = a^{2^u} = 1 \pmod N$$

و $a^{2^{u-1}}$ یک ریشی دوم از 1 است. اگر N یک عدد اول فرد باشد، تنها ریشه‌های دوم 1 عبارتند از ± 1 باین حال، بر اساس انتخاب $1 < a < N$ داریم $a^{2^{u-1}} \neq \pm 1 \pmod N$ بنابراین $a^{2^{u-1}} = -1 \pmod N$

و a یک شاهد قوی نیست. نتیجه می‌گیریم که زمانی که N یک عدد اول فرد است، هیچ شاهد قوی برای مرکب بودن N وجود ندارد.

یک عدد صحیح مرکب N یک «توان اول» است اگر $N = p^r$ برای یک p اول و عدد صحیح $r \geq 1$ است. اکنون نشان می‌دهیم که هر N مرکب مفرد که یک تولید اول نباشد، دارای تعداد زیادی شاهد قوی است.

۴۰-۸-۱

فرض کنید N یک عدد فرد باشد که یک توان اول نیست. آنگاه حداقل نمی از عناصر $\mathbb{Z}_N^*$ شاهد قوی برای مرکب بودن N هستند.

اثبات

فرض کنید $\mathbb{Z}_N^* \subseteq \text{Bad} \subseteq \mathbb{Z}_N^*$ نشان‌دهنده‌ی مجموعه‌ی عناصری باشد که شاهد قوی نیستند. ما یک مجموعه‌ی Bad' را تعریف کرده و نشان می‌دهیم که $\text{Bad}' \cap \text{Bad} = \emptyset$ یک زیرمجموعه از Bad' است و $\text{Bad}' \cap \text{Bad} = \emptyset$ یک زیرمجموعه‌ی Bad' است. این امر کفایت می‌کند به این دلیل که با ترکیب (۳) و (۴) داریم $|\text{Bad}'| \leq |\mathbb{Z}_N^*|/2$ ، به علاوه، بر اساس (۱) این رابطه برقرار است که $\text{Bad}' \cap \text{Bad} = \emptyset$ بنابراین $|\mathbb{Z}_N^*|/2 \leq |\text{Bad}'| \leq |\mathbb{Z}_N^*|/2$ به شکل قسمتی ۸-۳-۸. بنابراین، حداقل نمی از عناصر $\mathbb{Z}_N^*$ شاهد قوی هستند. (توجه می‌کنیم که ادعا داریم Bad یک زیرمجموعه از $\mathbb{Z}_N^*$ است.)

در ابتدا توجه کنید که $-1 \in \text{Bad}$ چرا که $-1 \pmod N = (-1)^u = -1$ (یادآوری می‌کنیم که u فرد است). فرض کنید $\{1, \dots, r-1\} \subseteq \{0, \dots, r-1\}$ بزرگ‌ترین عدد صحیحی باشد که برای آن، یک $a \in \text{Bad}$ یا $a^{2^i} = -1 \pmod N$ وجود دارد؛ متناوباً، بزرگ‌ترین عدد صحیحی است که برای آن $a \in \text{Bad}$ یا $a^{2^i} = -1 \pmod N$

$$(a^{2^0}, a^{2^1}, \dots, a^{2^r}) = \left(a, a^2, \dots, a^{2^{r-1}}, -1, \dots, -1, \dots, 1 \right).$$

وجود دارد. از آنجا که $-1 \in \text{Bad}$ و $-1 \pmod N = (-1)^u = -1$ چنین i وجود دارد. از به شکل بالا مشخص کنید و تعریف کنید

$$\text{Bad}' = \{a \mid a^{2^i} = \pm 1 \pmod N\}.$$

اکنون آنچه در بالا ادعا کردیم را اثبات می‌کنیم.

۴۱-۸-۱

ادعای $\text{Bad}' \subseteq \text{Bad}'$

یک عدد صحیح N یک «توان کامل» است اگر برای اعداد صحیح N و $2 \leq a$ داشته باشیم $N \equiv 0 \pmod{a}$ (در اینجا، لازم نیست که N اول باشد، هر چند البته هر توان اولی یک توان کامل هم هست). الگوریتم ۴۴-۸ آزمون اول بودن میلر-رابین را ارائه می‌کند. تمرین‌های ۱۲-۸ و ۱۳-۸ از شما می‌خواهند نشان دهید که آزمون اینکه N یک توان کامل است و آزمون اینکه آیا یک a خفیه یک شاهد قوی هست یا خیر را می‌توان در زمان چندجمله‌ای انجام داد. با داشتن این نتایج این الگوریتم به‌ویژه در زمان چندجمله‌ای در $N \parallel 2$ اجرا می‌شود اکنون می‌توانیم اثبات قضیه ۳۳-۸ را کامل کنیم.

اثبات

اگر N یک عدد اول فرد باشد، هیچ شاهد قوی وجود ندارد و بنابراین الگوریتم میلر-رابین همواره خروجی «prime» را ارائه می‌کند. اگر N زوج یا یک توان اول باشد، الگوریتم همواره خروجی «composite» را ارائه می‌کند. حالت چالش‌زایی است که N یک عدد صحیح مرکب فرد باشد که یک توان اول نیست. هر تکراری از حلقه‌ی داخلی را در نظر بگیریم.

الگوریتم ۴۴-۸

آزمون اول بودن میلر-رابین
 ورودی: عدد صحیح $N > 2$ و پارامتر t
 خروجی: یک تصمیم در خصوص اینکه آیا N اول است یا مرکب.
 اگر N زوج است، «composite» را بازگردان.
 اگر N یک توان کامل است، «composite» را بازگردان.
 $N - 1 = 2^s u$ که u فرد را به‌صورت محاسبه کن که $2 \nmid u$
 $for \ i = 1 \text{ to } t$
 $a \leftarrow (1, \dots, N-1)$
 اگر $a^u \not\equiv \pm 1 \pmod{N}$ و $a^N \not\equiv \pm 1 \pmod{N}$ برای $a \in \{1, \dots, N-1\}$
 return «prime»
 «composite» را بازگردان.

در اینجا توجه کنید که اگر $a \in \mathbb{Z}_N^*$ ، $a \not\equiv \pm 1 \pmod{N}$ و $a^N \not\equiv \pm 1 \pmod{N}$ برای $a \in \{1, \dots, N-1\}$ احتمال یافتن یک شاهد قوی یا یک عنصر که در $\mathbb{Z}_N^*$ باشد، حداقل برابر با $1/2$ است (بر اساس قضیه ۴۰-۸). بنابراین، احتمال اینکه الگوریتم هیچگاه خروجی «composite» را در هیچ کدام از t تکرار ارائه نکند حداکثر برابر با 2^{-t} است. ■

ادعای ۴۲-۸

یک زیرمجموعه از $\mathbb{Z}_N^*$ است.

منحنی است که $a, b \in Bad'$ ، به‌علاوه، اگر a, b آنکه

و بنابراین $ab \in Bad'$ به سلسله Bad' یک زیرگروه است.

ادعای ۴۳-۸

یک زیرگروه اکید از $\mathbb{Z}_N^*$ است.

اگر N یک عدد صحیح مرکب فرد باشد که یک توان اول نیست، آنکه N را می‌توان به‌صورت $N = N_1 N_2$ یا $1 < N_1, N_2 = 1$ و $\gcd(N_1, N_2) = 1$ نوشت. با استفاده از قضیه‌ی پایتادی، چنین فرض کنید $(a, a) \in \mathbb{Z}_N^* \times \mathbb{Z}_N^*$ به‌عنوان عنصری از $\mathbb{Z}_N^* \times \mathbb{Z}_N^*$ باشد. یعنی، $a_1 = [a \pmod{N}]$ و $a_2 = [a \pmod{N_1}]$ را در نظر بگیرید به صورتی که $a_1 \equiv -1 \pmod{N}$ (چنین a باید بر سلسله‌ی تعریف i وجود داشته باشد) و فرض کنید $a_2 \equiv -1 \pmod{N_1}$ از آنجمله $(a_1, a_2) \mapsto (-1, -1)$ داریم

$$(a_1, a_2) = (a_1^{t_1}, a_2^{t_2}) = (-1, -1),$$

و بنابراین

$$a_1^{t_1} = -1 \pmod{N_1} \quad \text{و} \quad a_2^{t_2} = -1 \pmod{N_2}$$

عنصر $b \in \mathbb{Z}_N^*$ را با $(a_1, 1) \mapsto (a_1, 1)$ در نظر بگیرید. آنکه

$$b^{t_1} \mapsto (a_1, 1)^{t_1} = (a_1^{t_1} \pmod{N_1}, 1) = (-1, 1) \mapsto \pm 1.$$

یعنی، $b^{t_1} \equiv \pm 1 \pmod{N}$ و $b^{t_2} \equiv \pm 1 \pmod{N}$ و بنابراین یک عنصر $b \in Bad'$ را یافته‌ایم. این امر اثبات می‌کند که Bad' یک زیرگروه اکید از $\mathbb{Z}_N^*$ است و بنابراین، بر سلسله Bad' اندازه‌ی $3\sqrt{N}$ است. ■

۲-۳-۸ فرض تجزیه

فرض کنید $GenModulus$ یک الگوریتم زمان چندجمله‌ای باشد که با دریافت ورودی 1^n خروجی (N, p, q) را ارائه می‌دهد که در آن $N = pq$ و p و q اعداد اول n -جیبی هستند مگر با احتمال ناچیز در n (روش طبیعی برای انجام این کار، تولید دو عدد اول n -جیبی، یکخواخت، به صورتی که قبلاً بحث شد و سپس ضرب آن‌ها برای به دست آوردن N است) سپس، آزمایش زیر را برای یک الگوریتم مفروض h و پارامتر n در نظر بگیرید:

آزمایش تجزیه $Factor_{GenModulus}(n)$

- (۱) اجرای $GenModulus(1^n)$ برای به دست آوردن (N, p, q)
- (۲) h مقدار N را دریافت کرده و $1 > p', q'$ را خروجی می‌دهد.
- (۳) خروجی آزمایش برابر با 1 تیرف می‌شود اگر $N = p'q'$ در غیر این صورت، خروجی برابر با 0 تیرف خواهد شد.

توجه کنید که اگر خروجی آزمایش برابر با 1 باشد، آنگاه $\{p', q'\} = \{p, q\}$ مگر اینکه p یا q مرکب باشند (که تنها با احتمال ناچیز رخ می‌دهد).

اکنون به شکل فرض تجزیه را تعریف می‌کنیم:

تعریف ۴-۵-۸

تجزیه نسبت به $GenModulus$ سخت است اگر برای تمام الگوریتم‌های زمان چندجمله‌ای تعدادی h یک تابع ناچیز $negl$ وجود داشته باشد به صورتی که

$$\Pr[Factor_{GenModulus}(n) = 1] \leq negl(n).$$

فرض تجزیه «این فرض است که یک $GenModulus$ وجود دارد که تجزیه نسبت به آن سخت است».

۴-۳-۸ فرض RSA

مسئله تجزیه به مدت مدتها سال بدون یافتن یک الگوریتم کارا، مورد مطالعه قرار گرفته است هرچند فرض تجزیه یک تابع یک‌طرفه را می‌دهد (ن.ک. بخش ۴-۸). متأسفانه به صورت مستقیم، سادگی‌های رمزنگاری عملی را ارائه نمی‌کند (زاین حال، در بخش ۱۳-۵-۲ نشان می‌دهیم که چگونه می‌توان سادگی‌های رمزنگاری کارا را بر اساس مسئله‌ای ساخت که سختی آن برابر با سختی تجزیه است) این امر باعث جستجو به دنبال دیگر مسائلی شده است که دشواری آن‌ها مربوط به سختی تجزیه است. شناخته‌شده‌ترین مورد از این مسائل، مسئله‌ای معرفی شده در سال ۱۹۷۸ توسط روسته شلایر و آدامس است که اکنون به احترام آن‌ها «مسئله RSA» نامیده می‌شود.

با داشتن یک پیمانه‌ی N و یک عدد صحیح $2 < e$ که نسبت به $\phi(N)$ اول باشد، نتیجه‌ی فرضی ۸-۲۲ نشان می‌دهد که توان رسانی به توان e ام به پیمانه‌ی N یک «چالاکت» است؛ بنابراین، می‌توانیم به $x^{1/e} \bmod N$ (برای هر $y \in \mathbb{Z}_N^*$) را به عنوان عنصر منحصر به فرد از $\mathbb{Z}_N^*$ تعریف کنیم که هنگام به توان رساندن تا توان e ام به پیمانه‌ی N ، از y می‌دهد؛ یعنی، $x = y^e \bmod N$ اگر و تنها اگر $x = y^e \bmod N$ به شکل غیر رسمی، مسئله RSA عبارت است از محاسبه‌ی $x^{1/e} \bmod N$ برای یک پیمانه‌ی N یا تجزیه‌ی مجهول.

به شکل رسمی، فرض کنید $GenRSA$ یک الگوریتم زمان چندجمله‌ای تعدادی باشد که با دریافت ورودی 1^n ، یک پیمانه‌ی N را خروجی می‌دهد که حاصل ضرب دو عدد اول n -جیبی و همین‌طور اعداد صحیح $0 < e, d \leq 1$ و $\gcd(e, \phi(N)) = 1$ است. (چنین d وجود دارد چرا که e و d وزن-پذیر به پیمانه‌ی $\phi(N)$ است، هدف d ، معکوس واضح می‌شود) این الگوریتم ممکن است با احتمال ناچیز در n شکست بخورد. آزمایش زیر را برای یک الگوریتم مفروض h و پارامتر n در نظر بگیرید:

آزمایش $RSA - inv_{GenRSA}(n)$

- (۱) $GenRSA(1^n)$ را اجرا کنید تا (N, e, d) به دست آید.
- (۲) یک $y \in \mathbb{Z}_N^*$ را یک‌خواخت را انتخاب کنید.
- (۳) h ورودی‌های N, e, y را دریافت کرده و $x^e = y \bmod N$ و برابر با 0 در غیر این صورت، تیرف خروجی آزمایش برابر با 1 اگر $x^e = y \bmod N$ و برابر با 0 در غیر این صورت، تیرف می‌شود.

تعریف ۴-۶-۸

مسئله RSA نسبت به $GenRSA$ سخت است اگر برای تمام الگوریتم‌های زمان چندجمله‌ای تعدادی h یک تابع ناچیز $negl$ وجود داشته باشد به صورتی که $\Pr[RSA - inv_{GenRSA}(n) = 1] \leq negl(n)$.

فرض RSA «آن است که یک الگوریتم $GenRSA$ وجود دارد که نسبت به آن، مسئله RSA سخت است. یک الگوریتم $GenRSA$ مناسب را می‌توان از روی هر الگوریتم $GenModulus$ ساخت که یک پیمانه‌ی مرکب را به همراه تجزیه‌ی آن، تولید می‌کند. یک چارچوب سطح بالا به عنوان الگوریتم ۸-۲۲ ارائه شده است که در آن، تنها چیزی که مشخص نشده است آن است که e دقیقاً چگونه انتخاب می‌شود. در واقع، باور بر آن است که مسئله RSA برای هر e که نسبت به $\phi(N)$ اول است، سخت خواهد بود. تعدادی انتخاب معمول برای e را در ذیل، مورد بحث قرار می‌دهیم.

در خصوص انتخاب e

به نظر می‌رسد که تفاوتی در سختی مسئله RSA برای نمادهای متفاوت e وجود داشته باشد و به نظر می‌رسد که تفاوتی برای انتخاب آن پیشنهاد شده‌اند. یک انتخاب محبوب عبارت است از پرایم، روش‌های مختلفی برای انتخاب آن پیشنهاد شده‌اند. N تنها نیازمند دو ضرب است و $e = 3$ چرا که در این حالت، محاسبه توان e ام به بیانی N تنها نیازمند دو ضرب است و $e = 3$ به $2^{16} + 1$ اگر e قرار باشد برابر با ۳ قرار داده شود، آنگاه p و q باید $p, q \ne 1 \bmod 3$ انتخاب شوند تا $\gcd(e, \phi(N)) = 1$ به دلایل مشابه، یک انتخاب محبوب دیگر برابر با $65537 = 2^{16} + 1$ است. یک عدد اول با وزن همیگ پایین (در ضمیمه ب-۲-۱ توضیح می‌دهم) که چرا چنین توان‌هایی ترجیح داده می‌شوند، در مقایسه با انتخاب $e = 3$ این انتخاب این می‌شود که توان رسانی اندکی گران‌تر شود ولی محدودیتها بر p و q را کاهش داده و از بعضی حملات مبتنی بر «توصیف‌شده در انتهای بخش ۱۱-۵» (۱) اجتناب می‌کند که می‌توانند ناشی از رابطه‌های رمزنگاری با پیاده‌سازی ضعیف مبتنی بر مسئله RSA باشند.

توجه کنید که انتخاب d کوچک (یعنی، تغییر GenRSA برای انتخاب d کوچک و سپس محاسبه $d' = [d^{-1} \bmod \phi(N)]$ فکر اشتباهی است. اگر d در یک دامنه بسیار کوچک قرار داشته باشد، آنگاه یک جستجوی فرآیند به دنبال d را می‌توان اجرا نمود و آن‌طور که بیان شد هنگامی که d معلوم باشد، مسئله RSA را به سادگی می‌توان حل کرد، حتی اگر d به‌گونه‌ای انتخاب شود که $N \nmid d$ و بنابراین حملات جستجوی فرآیند امکان‌پذیر نباشند، الگوریتم‌های شناخته‌شده‌ای وجود دارند که می‌توان از آنها برای پازایی d از روی N و e در این حالت استفاده نمود.

۲-۸-۵ مرتبط کردن فرضیات RSA و تجزیه

فرض کنید GenRSA به شیوه‌ی توصیف‌شده در الگوریتم ۸-۴۷ ساخته شود. اگر N را بتوان تجزیه کرد آنگاه می‌توانیم $\phi(N)$ را محاسبه کرده و از آن برای محاسبه $d' = [e^{-1} \bmod \phi(N)]$ برای $e: dt = 1$ به روشی (با استفاده از الگوریتم ب-۱) استفاده کنیم. بنابراین، برای اینکه مسئله RSA نسبت به GenRSA سخت باشد، مسئله تجزیه باید نسبت به GenModulus سخت باشد به بیان دیگر، مسئله RSA نمی‌تواند دشوارتر از تجزیه باشد. سختی تجزیه (نسبت به GenModulus) تنها به شکل بالقوه می‌تواند یک فرض ضعیف تر نسبت به سختی مسئله RSA (نسبت به GenRSA) باشد. مگر مطلب جلوتر یعنی «آیا سختی مسئله RSA توسط سختی تجزیه نشان داده می‌شود؟» این سوال همچنان بی‌پاسخ است. بهترین چیزی که می‌توانیم نشان دهیم آن است که محاسبه d از روی N و e به سختی تجزیه است.

الگوریتم ۸-۴۷
GenRSA-چارچوب سطح بالا
ورودی: پارامتر امنیتی 1^3
خروجی: N, e, d به شکل توصیف‌شده در متن.
 $(N, p, q) \leftarrow \text{GenModulus}(1^3)$
 $\phi(N) := (p-1)(q-1)$
 $e > 1$ را انتخاب کنید به صورتی که $1 = \gcd(e, \phi(N))$
 $d := [e^{-1} \bmod \phi(N)]$
return N, e, d

مثال ۸-۴۸

فرض کنید GenModulus خروجی $(N, p, q) = (143, 11, 13)$ را ارائه می‌کند. آنگاه $\phi(N) = 120$ پس باید یک e را انتخاب کنیم که نسبت به $\phi(N)$ اول باشد. فرض کنید قرار می‌دهیم $e = 7$ که گام بعدی محاسبه d به صورتی است که $d = [e^{-1} \bmod \phi(N)]$ باشد. این امر را می‌توان به صورت نشان داده‌شده در ضمیمه ب-۲-۳ برای به دست آوردن $d = 103$ انجام داد (می‌توان بررسی نمود که $120 \bmod 103 = 17$ ، بنابراین، الگوریتم GenRSA ما، خروجی $(143, 7, 103)$ را ارائه می‌کند.

به عنوان مثال از مسئله RSA نسبت به این پارامترها، $y = 7$ را قرار دهید و بنابراین مسئله عبارت است از محاسبه رشتهی هفتم 4^7 به بیانی 143 بدون دانستن d یا تجزیه N .

محاسبه رشته‌های e ام به بیانی N ساده خواهد شد اگر d ، $\phi(N)$ یا عوامل N معلوم باشند. (همان‌طور که در بخش بعدی نشان می‌دهیم، هر کدام از این موارد را می‌توان برای محاسبه کارایی مورد دیگر، مورد استفاده قرار داد). این امر از نتیجه‌ی فرعی ۸-۲۳ به دست می‌آید که نشان می‌دهد $e = 7$ ، رشته e ام y به بیانی N است. این تقارن - یعنی اینکه به نظر می‌رسد مسئله RSA سخت است زمانی که d یا عوامل تجزیه‌ای N مجهول باشند ولی ساده می‌شود زمانی که d معلوم باشد - به عنوان پایه و اساسی برای کاردهای مسئله RSA در رمزنگاری کلید عمومی، عمل خواهد کرد.

مثال ۸-۴۹

در ادامه مثال قبل، می‌توانیم رشته 10^7 را به بیانی 143 با استفاده از مقدار $d = 103$ محاسبه کنیم. پاسخ برابر است با $10^7 \bmod 143 = 64^3 = 264$ می‌توانیم بررسی کنیم که این مراحل درستی است چرا که $143 \bmod 264 = 70^3 = 26^3$.

قضیه ۵-۸-۵
یک الگوریتم زمان چندجمله‌ای تصادفی وجود دارد که با دریافت یک عدد صحیح مرکب N و اعداد صحیح d, e با $ed = 1 \bmod \phi(N)$ به عنوان ورودی، یک عامل از N را مگر با احتمال ناچیز در $\ll N$ به عنوان خروجی، ارائه می‌کند.

اثبات

این قضیه برای هر N برقرار است ولی برای ساده‌گی و به این دلیل که حالتی است که بیشترین اهمیت را برای رمزنگاری دارد، در اینجا بر حالتی تمرکز می‌کنیم که در آن، N حاصل ضرب دو عدد اول (فرد) متناظر است، از گزاردی 2^6-8 و 2^6-8 و همین‌طور واقعیت‌های زیر استفاده می‌کنیم (که از نتایج کمی‌تر اثبات‌شده در بخش‌های ۱۲-۴-۲ و ۱۳-۵-۲ به دست می‌آیند):

- اگر N حاصل ضرب دو عدد اول متناظر فرد باشد، آنگاه ۱ دارای دقیقاً چهار ریشه‌ی دوم به پیمانه‌ی N است، دو مورد از آن‌ها، ریشه‌های دوم «بدیعی» ± 1 هستند و دو مورد از آن‌ها، ریشه‌های دوم «غیربدیعی» هستند.

- هر ریشه‌ی دوم غیربدیعی 1 را می‌توان برای محاسبه‌ی (کزاری) یک عامل از N مورد استفاده قرار داد. این امر به دلیل این واقعیت است که $1 \bmod N = x^2 = 1$ در نشان می‌دهد که

$$x^2 - 1 = (x-1)(x+1) \bmod N,$$

و بنابراین، $(x+1) \mid N$ یا $(x-1) \mid N$ یا هر دو. حال، $N \nmid (x+1)$ و $N \nmid (x-1)$ به این دلیل که $x^2 \equiv 1 \bmod N$ و $x \not\equiv \pm 1 \bmod N$ زیرا بنابراین، باید این حالت برقرار باشد که $\gcd(x-1, N) = 1$ و $\gcd(x+1, N) = 1$ و عمل اول N است.

فرض کنید $1 - ed = k$ توجه کنید که $k \mid \phi(N)$ با استفاده از نتیجه‌ی فرضی ۸-۲-۸ داریم $x^k = 1 \bmod N$ برای تمامی $x \in \mathbb{Z}_N^*$ فرض کنید $k = 2^t u$ برای u یک عدد صحیح فرد، توجه کنید که $t \geq 1$ چرا که $\phi(N)$ و بنابراین، k زوج است. استراتژی ما برای تجزیه‌ی N عبارت خواهد بود از انتخاب مکرر $x \in \mathbb{Z}_N^*$ و یکپارخت و محاسبه‌ی دنباله‌ی

$$x^k, x^{2k}, x^{4k}, \dots, x^{2^{t-1}k},$$

همگی به پیمانه‌ی N هر جمله در این دنباله برابر است با مربع جمله‌ی قبلی و همان‌طور که هم‌اکنون اشاره کردیم، جمله‌ی آخر در این دنباله برابر با ۱ است. بزرگ‌ترین i در صورت وجود i را در نظر بگیرید که برای آن، $1 \neq [x^{2^i k} \bmod N] \stackrel{\text{def}}{=} z$ بر اساس انتخاب i داریم $1 \bmod N = z^2 = 1$ اگر $z \neq -1$ بر یک ریشه‌ی دوم غیربدیعی برای N یافتیم و سپس می‌توانیم N را به صورت بحث‌شده در بالا تجزیه کنیم.

تمام گام‌های بالا را می‌توان در زمان چندجمله‌ای انجام داد و بنابراین، تنها سوال عبارت است از تعیین احتمال اینکه، روی انتخاب x یک ریشه‌ی دوم غیربدیعی برای N باشد. فرض کنید i

$(-1)^{i-1} \neq [x^{2^{i-1}k} \bmod N]$ باشد که برای آن، یک $z \in \mathbb{Z}_N^*$ وجود دارد به صورتی که $z^2 = x^{2^i k} \bmod N$ ، (از اینجا که u فرد است، $1 \bmod N = -1 \neq (-1)^u = -1$) و بنابراین، این تعریف بدیعی نیست. آنگاه برای همگی $x \in \mathbb{Z}_N^*$ داریم $x^{2^i k} = 1 \bmod N$ و بنابراین $[x^{2^i k} \bmod N]$ یک ریشه‌ی دوم برای ۱ است. تعریف کنید

$$\text{Bad} \stackrel{\text{def}}{=} \{x \mid x^{2^i k} = \pm 1 \bmod N\}$$

توجه کنید که اگر الگوریتم ما Bad را انتخاب کند، آنگاه یک ریشه‌ی دوم غیربدیعی برای ۱ می‌باشد. نشان می‌دهیم که Bad یک زیرگروه اکید از $\mathbb{Z}_N^*$ است، بر اساسی لم ۸-۸-۲۷ این امر نشان می‌دهد که $|\text{Bad}| \leq |\mathbb{Z}_N^*|/2$. این بدان معناست که با احتمال حداقل $1/2$ در هر تکرار، $x \in \text{Bad}$ می‌دهد (و الگوریتم یک ریشه‌ی دوم غیربدیعی برای ۱ می‌یابد). استفاده از تعداد تکرارهای کافی، نتیجه‌ی این قضیه به دست می‌آید.

اکنون اثبات می‌کنیم که Bad یک زیرگروه اکید از $\mathbb{Z}_N^*$ است. ابتدا توجه داشته باشید که Bad بی‌نهایت چرا که $1 \in \text{Bad}$ ، به علاوه، اگر $x, x' \in \text{Bad}$ آنگاه

$$(xx')^{2^i k} = x^{2^i k} x'^{2^i k} = (\pm 1) \cdot (\pm 1) = \pm 1 \bmod N,$$

و بنابراین، $xx' \in \text{Bad}$ و $xx' \in \text{Bad}$ یک زیرگروه است. برای مشاهده‌ی اینکه Bad یک زیرگروه اکید است، فرض کنید $x \in \mathbb{Z}_N^*$ به گونه‌ای باشد که $1 \bmod N \neq x^{2^i k}$ (چنین x باید بر اساس تعریف ما از وجود داشته باشد). اگر $1 \bmod N \neq x^{2^i k}$ و $1 \bmod N \neq x^{2^i k}$ و کار ما تمام است. در غیر این صورت فرض کنید $1 \bmod N \neq x^{2^i k}$ و $1 \bmod N \neq x^{2^i k}$ و فرض کنید $x^{2^i k} \equiv \pm 1 \bmod N$ و نمایش باقیمانده‌ی چینی x باید از اینجا که $1 \bmod N \neq x^{2^i k} = -1 \bmod N$ می‌دانیم که

$$(x_p, x_q)^{2^i k} = (-1, -1) = (-1, -1).$$

پس آنگاه $(x_p, x_q)^{2^i k} = (-1, -1) = (-1, -1)$ در Bad نیست چرا که

$$(x_p, x_q)^{2^i k} = \left([x_p^{2^i k} \bmod p], [x_q^{2^i k} \bmod q] \right) = (-1, 1) \neq (-1, -1).$$

این اثبات را کامل می‌کند. ■

با فرض اینکه تجزیه سخت است، نتیجه‌ی بالا امکان حل کارای مسئله‌ی RSA به این صورت که ابتدا d را از روی N و e محاسبه کنیم، حذف می‌کند. باین‌حال، این امکان را از بین می‌برد که ممکن است یک روش کاملاً متفاوت برای حمله به مسئله‌ی RSA وجود داشته باشد که شامل (یا تلاش‌دهنده‌ی تجزیه‌ی N نباشد. بنابراین، بر اساس دانش کنونی خود، فرض RSA قوی‌تر از فرض تجزیه است - یعنی، ممکن است مسئله‌ی RSA را بتوان در زمان چندجمله‌ای حل کرد هر چند که

تجزیه را نمی‌توان، باین حال، هنگامی که GenRSA بر اساس GenModulus ساخته می‌شود، مثل حالت موجود در الگوریتم ۱۴-۸، بر داشت عمومی آن است که مسطلی RSA نسبت به GenRSA سخت خواهد بود هرگاه تجزیه نسبت به GenModulus سخت باشد.

۳-۸- فرضیات رمزنگاری در گروه‌های دوری

در این بخش، یک دسته از فرضیه‌های سخت رمزنگاری را در گروه‌های دوری «معرفی می‌کنیم. بعضی کلی در خصوص گروه‌های دوری شروع کرده و تعاریف مجرد فرضیات مربوطه را ارائه می‌کنیم. در ادامه، دو مثال عینی از گروه‌های دوری را بررسی می‌کنیم که به صورت گسترده مورد استفاده قرار می‌گیرند که در آن‌ها، باور بر آن است که این فرضیات، برقرار هستند.

۱-۳-۸- گروه‌های دوری و مولدها

فرض کنید G یک گروه متناهی با مرتبه m باشد. برای $G \in \mathcal{G}$ دلخواه، مجموعه‌ای زیر را در نظر بگیرید:

$$\langle g \rangle = \{g^0, g^1, \dots, g^{m-1}\}.$$

(به خواننده هشدار می‌دهیم که اگر G یک گروه نامتناهی باشد، $\langle g \rangle$ به شکلی متفاوت تعریف می‌شود.) بر اساس قضیه ۱۴-۸، داریم $1 = g^m$ فرض کنید $m \leq i$ کوچک‌ترین عدد صحیح مثبتی باشد که برای آن $1 = g^i$ انگاره دنباله‌ای بالا بعد از i جمله، تکرار می‌شود (یعنی $= 1$ ، $g^0 = 1, g^1, g^2, \dots, g^{i-1}$) و بنابراین

$$\langle g \rangle = \{g^0, g^1, \dots, g^{i-1}\}.$$

مطالعه می‌کنیم که $\langle g \rangle$ شامل حداقل i عنصر است، در واقع، دقیقاً شامل i عنصر است به این دلیل که اگر $g^k = g^j$ با $0 \leq j < k < i$ و $g^{k-j} = 1$ ، که نقض کننده انتخاب ما برای i به عنوان کوچک‌ترین عدد صحیح مثبتی خواهد بود که برای آن، $1 = g^i$ است.

بررسی اینکه $\langle g \rangle$ یک زیرگروه از G برای هر G است، سخت نیست. کمترین ۱۴-۸، ما $\langle g \rangle$ را جزیرگروه تولیدشده توسط g می‌نامیم، اگر مرتبه‌ای زیرگروه $\langle g \rangle$ برابر با i باشد، انگاره i را مرتبه‌ای g می‌نامیم؛ یعنی:

تعریف ۵۳-۸

فرض کنید G یک گروه متناهی و $G \in \mathcal{G}$ باشد. مرتبه‌ای g عبارت است از کوچک‌ترین عدد صحیح مثبت i با $1 = g^i$

گزاره‌ای زیر یک مشاهده مفید از نتیجه‌ی فرعی ۱۵-۸ است (اثبات یکسان است):

گزاره ۵۳-۸ فرض کنید G یک گروه متناهی بوده و $G \in \mathcal{G}$ یک عنصر با مرتبه‌ای i است. انگاره برای هر عدد صحیح x داریم $[x \bmod i] = g^x$.

می‌توانیم چیزی قوی‌تر را اثبات کنیم:

گزاره ۵۳-۸

فرض کنید G یک گروه متناهی بوده و $G \in \mathcal{G}$ یک عنصر با مرتبه‌ای i باشد. انگاره $g^x = g^{x \bmod i}$ تنها اگر $x \bmod i = y \bmod i$ باشد.

اثبات

اگر $x \bmod i = y \bmod i$ ، انگاره $[x \bmod i] = [y \bmod i]$ و گزاره قبلی می‌گوید که $g^x = g^y$.

برای طرف چپ، فرض کنید $g^x = g^y$ ، انگاره $[x \bmod i] = [y \bmod i]$ (با استفاده از گزاره قبلی)؛ زیرا که $i < x - y \bmod i$ ولی i کوچک‌ترین عدد صحیح مثبت با $1 = g^i$ است، باید داشته باشیم $0 = [x - y \bmod i]$.

عصر همدلی هر گروه G تنها عنصر از مرتبه‌ای 1 است و گروه $\{1\} = \langle 1 \rangle$ را تولید می‌کند. در سویی دیگر، اگر یک عنصر $G \in \mathcal{G}$ وجود داشته باشد که دارای مرتبه‌ای m باشد (در آن، m مرتبه‌ای G است)، انگاره $G = \langle g \rangle$ در این حالت، ما G را یک «گروه دوری» می‌نامیم و می‌گوییم که G یک مولد G است. (یک گروه دوری ممکن است دارای مولدهای متعدد باشد و بنابراین نمی‌توانیم در خصوص یک مولد صحبت کنیم) اگر G یک مولد برای G باشد، انگاره بر اساس تعریف هر عنصر $G \in \mathcal{G}$ برابر با g^x برای $x \in \{0, \dots, m-1\}$ مفروض خواهد بود که گنگامی است که در بخش بعد به آن باز می‌گردیم.

عناصر مختلف گروه یکسان G ممکن است دارای مرتبه‌های مختلف باشند. باین حال، می‌توانیم محدودیت‌هایی را برای این درجعات بالقوه، قائل شویم.

گزاره ۵۴-۸

فرض کنید G یک گروه متناهی با مرتبه‌ای m باشد و $G \in \mathcal{G}$ دارای مرتبه‌ای i باشد. انگاره $i \mid m$.

اثبات

بر اساس قضیه ۱۴-۸، می‌دانیم که $g^m = 1$ و $g^i = 1$ گزاره ۵۳-۸ نشان می‌دهد که $m \mid i$.

نتیجه فرعی بدی قدرت این نتیجه را نشان می دهد:

نتیجه فرعی ۵۵-۸

اگر G یک گروه با مرتبه اول p باشد، آنگاه G دوری است. به علاوه، تمام عناصر G به غیر از عنصر هملی، مولدهای G هستند.

اثبات

بر اساس گزاره ۵۴-۸، تنها درجات ممکن برای عناصر در G عبارتند از ۱ و p تنها عنصر هملی دارای مرتبه ۱ است و بنابراین تمامی دیگر عناصر دارای مرتبه p هستند و G را تولید می کنند. ■

گروههای با مرتبه اول، یک دسته از گروههای دوری را تشکیل می دهند. گروه جمعی $\mathbb{Z}_p$ برای $N > 1$ مثال دیگری از یک گروه دوری را می دهد (عصر ۱ همواره یک مولد است). قضیه بدی-که یک حالت خاص از قضیه الف-۲۱ است- یک دستهی مهم دیگر از گروههای دوری را می دهد، اثبات خارج از گسترده این کتاب است ولی می توان آن را در هر متن جبر مجرد استاندارد یافت.

قضیه ۵۵-۸

اگر p اول باشد، آنگاه $\mathbb{Z}_p$ یک گروه دوری از مرتبه $p-1$ خواهد بود.

برای $p > 2$ اول، $\mathbb{Z}_p$ دارای مرتبه اول نیست و بنابراین، قضیه بالا از نتیجه فرعی قبلی به دست نمی آید.

مثال ۵۲-۸

گروه (جمعی) $\mathbb{Z}_p$ را در نظر بگیرید. همان طور که اشاره کردیم، $\mathbb{Z}_{p,0}$ دوری است و عنصر ۱ یک مولد است چرا که $1 \bmod p = 1$ و $1 \bmod p \neq 0$. برای هر $i < p$ ، $i \bmod p = i$ (یادآوری می کنیم که در این گروه، عنصر هملی برابر با ۰ است).

در $\mathbb{Z}_p$ دارای مولدهای دیگری است. برای مثال، $\{1, 2, \dots, p-1\}$ دوری است و بنابراین ۲ هم یک مولد است.

همه عناصر مولد $\mathbb{Z}_p$ نیستند. برای مثال، عنصر ۲ دارای مرتبه ۵ است چرا که $2^5 \bmod p = 2$ و بنابراین، ۲ مولد $\mathbb{Z}_p$ نیست. زیر گروه $\langle 2 \rangle$ شامل ۵ عنصر $\{1, 2, 4, 8, 16\}$ است و این زیر گروه در واقع یک زیر گروه جمعی به پیمانه ۱۵ است. عنصر ۱۰ دارای مرتبه ۳ است چرا که $10^3 \bmod p = 10$ و زیر گروه $\langle 10 \rangle$ شامل ۳ عنصر $\{1, 10, 100\}$ است. درجات زیر گروهها (یعنی ۵ و ۳ و ۱۵) را عاده می کنند همان طور که توسط گزاره ۵۴-۸ اعلام شده است. ۵

مثال ۵۸-۸

گروه (صری) $\mathbb{Z}_{15}$ با مرتبه $15 = (3-1)(5-1)$ را در نظر بگیرید. داریم $\{1, 2, 4, 8\} = \langle 2 \rangle$ و بنابراین مرتبه ۲ برابر با ۴ است. همان طور که گزاره ۵۴-۸ می کند، ۸ بر ۴ بخش پذیر است. ۵

مثال ۵۹-۸

گروه (جمعی) $\mathbb{Z}_p$ با مرتبه اول p را در نظر بگیرید. می دانیم که این گروه دوری است ولی نتیجه فرعی ۵۵-۸ مطالب بیشتری به ما می گوید: یعنی، هر عنصر به غیر از ۰، یک مولد است. در واقع، برای هر $h \in \{1, \dots, p-1\}$ و عدد صحیح $i > 0$ داریم $hi \bmod p = i$ اگر و تنها اگر $hi \bmod p \neq 0$ و این حالت، گزاره ۳-۸ می گوید که یا $hi \bmod p = 0$ یا $hi \bmod p$ مورد اول نمی تواند رخ دهد (چون $h < p$ و کوچک ترین عدد صحیحی که برای آن، مورد دوم می تواند رخ دهد عبارت است از p ، $i = p$ و بنابراین، نشان دادیم که هر عنصر غیر صفر h دارای مرتبه p است (و بنابراین مولد $\mathbb{Z}_p$ است)، که منطبق با نتیجه فرعی ۵۵-۸ است. ۵

مثال ۶۰-۸

گروه (صری) $\mathbb{Z}_p$ را در نظر بگیرید که بر اساس قضیه ۵۵-۸، دوری است. داریم $\{1, 2, 4\} = \langle 2 \rangle$ و بنابراین یک مولد نیست. باین حال،

$$\langle 2 \rangle = \{1, 2, 4, 8\} = \mathbb{Z}_8^*$$

و بنابراین، ۲ یک مولد $\mathbb{Z}_8^*$ است. ۵

مثال زیر به مطالب بخش ۸-۵-۱ نکیه دارد.

مثال ۶۱-۸

فرض کنید G یک گروه دوری با مرتبه n باشد و فرض کنید u یک مولد برای G باشد. آنگاه، ثابت $G \rightarrow \mathbb{Z}_n$: داده شده توسط $f(u) = 1$ یک یکپارچگی بین $\mathbb{Z}_n$ و G است. در واقع، برای $a \in \mathbb{Z}_n$ ، $a \cdot u \in G$ داریم

$$f(a \cdot u) = f(a) \cdot f(u) = f(a) \cdot 1 = f(a).$$

دومین بون f را می توان با استفاده از این واقعیت اثبات نمود که n یک مرتبه برای G است. ۵

مثال قبلی نشان می دهد که تمام گروههای دوری با مرتبه n یکسان، یکپارچگی هستند و بنابراین از دیدگاه جبری، یکسان هستند. تأکید می کنیم که این امر از دیدگاه محاسباتی، صقل نمی کند و علی الخصوص نیازی نیست که یک یکپارچگی $\mathbb{Z}_n \rightarrow G$: f (که می دانیم باید وجود داشته باشد)

قابل‌محاسبی کارا باشد. این نکته بعد از بحث موجود در بخش زیر و همین‌طور فصل نهم، واضح‌تر می‌شود.

۳-۳-۸. فوژیات لگاریتم گسسته-دیفنی-حلمن

اکنون چندین مسئله محاسباتی را معرفی می‌کنیم که می‌توانند برای هر دسته‌ای از گروه‌های دوری، تعریف شوند. بحث این بخش را کوتاه می‌کنیم و مثال‌های خاصی گروه‌هایی که در آن‌ها بلور بر این است این مسائل، سخت هستند را در بخش‌های ۳-۳-۸ و ۴-۳-۸ مدنظر قرار می‌دهیم.

فرض می‌کنیم G نشان‌دهنده یک الگوریتم تولید گروه زمان-چندجمله‌ای عمومی باشد. این الگوریتم، الگوریتمی است که با دریافت ورودی 1^n توصیفی از یک گروه دوری G ، مرتبه‌ای آن q را $|G| = n$ و یک مولد $g \in G$ را به‌عنوان خروجی، ارائه می‌کند. توصیف یک گروه دوری مشخص می‌کند که چگونه عناصر گروه به‌صورت رشته‌ای یکتا نمایش داده می‌شوند؛ فرض می‌کنیم که هر عنصر گروه توسط یک رشته یکتا مشخص‌شده نمایش داده می‌شود. ملزم می‌کنیم که الگوریتم‌های کارایی (یعنی الگوریتم‌های اجراشده در زمان چندجمله‌ای بر مبنای n) برای محاسبی عملیات گروهی در G و همین‌طور برای ازمدن این امر وجود دارند که آیا یک رشته یکتا یکتا فروش نشان‌دهنده یک عنصر از G هست یا خیر. محاسبی کارایی عملیات گروهی به‌صورت ضمنی نشان‌دهنده الگوریتم‌های کارا برای توان‌رسانی در G (ن.ک. ضمیمه ب-۳-۲) و برای نمودارهای یک عنصر یک‌جواب $g \in G$ در واقع، انتخاب $x \in \mathbb{Z}_q$ و قرار دادن $g^x = h$ است.

همان‌طور که در انتهای پیش فنی بحث شد، هرچند تمام گروه‌های دوری با یک مرتبه ی مفروض، یک‌ریخت هستند، همانند گروه پیچیدگی محاسباتی عملیات ریاضی در آن گروه را تعیین می‌کند. اگر G یک گروه دوری با مرتبه q با مولد g باشد، آنگاه $\{g^1, g^2, \dots, g^{q-1}\}$ تمام G است، به شکل برزبر، برای هر $h \in G$ یک $h \in \mathbb{Z}_q$ به صورتی که $h = g^h$ هنگامی که گروه مبنای G روی بستر درک شد، این x را «لگاریتم گسسته h نسبت به g » می‌نامیم و می‌نویسیم $h = \log_g h$. $x = \log_g h$ در این حالت، «گسسته» نامیده می‌شوند چراکه مقادیری در یک دامنه‌ای از متناهی را می‌بخشند در مقابل لگاریتم‌های «استاندارد» در حسابان که مقادیر آن‌ها در دامنه‌ای از یک مجموعه نامتناهی از اعداد حقیقی است) توجه کنید که اگر $h = g^x$ برای یک عدد صحیح دلخواه x آنگاه $h = \log_g h \equiv x' \pmod{q}$.

لگاریتم‌های گسسته از بسیاری از قواعد مشابه با لگاریتم‌های «استاندارد» تبعیت می‌کنند. برای مثال، $1 = \log_g 1$ که در آن، 1 همانی G است؛ برای هر عدد صحیح x داریم $h^x = \log_g h^x$.

$$\log_g (h_1 h_2) = (\log_g h_1 + \log_g h_2) \pmod{q} \text{ و } \log_g h \pmod{q}.$$

مسئله لگاریتم گسسته در یک گروه دوری G با مولد g عبارت است از محاسبی $h = \log_g h$ برای یک عنصر یک‌جواب $h \in G$. آزمایش زیر برای یک الگوریتم تولید گروه K الگوریتم h و پارامتر n را در نظر بگیرید.

آزمایش لگاریتم گسسته $DLog_g(n)$:

- (۱) $G(1^n)$ را اجرا کنید تا (G, g, q) به دست آید که در آن، G یک گروه دوری با مرتبه q (یا $q = n$) بوده و g یک مولد G است.
- (۲) یک $h \in G$ را یکجواب و انتخاب کنید.
- (۳) الگوریتم h ورودی‌های h, g, q را دریافت کرده و $x \in \mathbb{Z}_q$ را به‌عنوان خروجی ارائه می‌کند.
- (۴) خروجی آزمایش برابر با 1 است اگر $h = g^x$ و در غیر این صورت، برابر 0 خواهد بود.

تعریف ۳-۸-۱

می‌گوییم مسئله لگاریتم گسسته نسبت به G سخت است اگر برای تمام الگوریتم‌های زمان-چندجمله‌ای تصادفی A ، یک تابع ناچیز $negl$ وجود داشته باشد به صورتی که

$$\Pr[DLog_g(n) = 1] \leq negl(n)$$

فرض لگاریتم گسسته درواقع این است که یک G وجود دارد که برای آن، مسئله لگاریتم گسسته سخت است. دو پیش‌ذیل تمثالی از الگوریتم‌های تولید گروه K مطلوب را موردبحث قرار می‌دهند که برای آن‌ها، بلور بر آن است که این امر صدق می‌کند.

مسائل دیفی-حلمن

مسائل معروف به مسائل دیفی-حلمن مرتبط با مسئله محاسبی لگاریتم‌های گسسته هستند ولی با آن‌ها برابر نیستند. دو گونه‌ی مهم وجود دارد: مسئله دیفی-حلمن محاسباتی (CDH) و مسئله دیفی-حلمن تصمیمی (DDH).

یک گروه دوری G و یک مولد $g \in G$ را در نظر بگیرید. با داشتن عناصر $h_1, h_2 \in G$ ، تعریف کنید

$$DH_g(h_1, h_2) = g^{x_1 x_2} = h_1^{x_2} = h_2^{x_1}.$$

مسئله CDH عبارت است از محاسبی $DH_g(h_1, h_2)$ برای h_1 و h_2 یکجواب. سختی این مسئله را می‌توان از طریق آزمایش طبیعی، به صورت رسمی تعریف نمود؛ جزئیات این امر را به‌عنوان یک تمرین باقی می‌گذاریم.

اگر مسئله‌ی لگاریتم گسسته نسبت به یک G مفروض، ساده باشد، آنگاه مسئله‌ی CDH هم همین‌طور خواهد بود: با داشتن h_1 و h_2 در ابتدا $x_1 = \log_g h_1$ را محاسبه کرده و سپس پاسخ $h_1^{x_1}$ را به‌عنوان خروجی ارائه کنید. در مقابل، مشخص نیست که آیا سختی مسئله‌ی لگاریتم گسسته به این معنا است که مسئله‌ی CDH هم سخت است یا خیر.

به‌صورت تقریبی، مسئله‌ی DDH عبارت است از متمایز نمودن $DH_g(h_1, h_2)$ از یک عنصر گروهی یکنواخت زمانی که h_1, h_2 یکنواخت باشند. یعنی، با داشتن h_1, h_2 یکنواخت و یک عنصر گروهی سوم h' مسئله عبارت است از تصمیم‌گیری در خصوص اینکه آیا $h' = DH_g(h_1, h_2)$ است یا خیر یا آیا h' به‌صورت یکنواخت از G انتخاب شده است. به شکل رسمی:

تعریف ۸-۶۳

می‌گوییم مسئله‌ی DDH نسبت به G سخت است اگر برای تمام الگوریتم‌های زمان-چندجمله‌ای تصادفی A یک تابع ناچیز $negl$ وجود داشته باشد به صورتی که

$$|\Pr[A(G, q, g, g^x, g^y, g^z) = 1] - \Pr[A(G, q, g, g^x, g^y, g^{xy}) = 1]| \leq negl(n),$$

که در آن، در هر حالت، احتمالات روی آزمایشی گرفته شده‌اند که در آن، $G(1^n)$ خروجی (G, q, g) را ارائه داده و سپس، $x, y, z \in \mathbb{Z}_q$ یکنواخت، انتخاب می‌شوند. (توجه کنید که هنگامی که z در $\mathbb{Z}_q$ یکنواخت است، آنگاه g^z دارای توزیع یکنواخت در G است.)

قبلاً مشاهده کردیم که اگر مسئله‌ی لگاریتم گسسته نسبت به یک G مفروض، ساده باشد، آنگاه مسئله‌ی CDH هم به همین صورت است. به شکل مشابه، اگر مسئله‌ی CDH نسبت به G ساده باشد، آنگاه مسئله‌ی DDH هم به همین صورت خواهد بود؛ در تمرین ۸-۱۵ از شما خواسته می‌شود تا این امر را نشان دهید. باین‌حال، به نظر نمی‌رسد که عکس این مطلب صادق باشد و مثال‌هایی از گروه‌هایی وجود دارد که در آن‌ها، باور بر آن است که مسائل لگاریتم گسسته و CDH سخت هستند هرچند که مسئله‌ی DDH ساده است؛ ن.ک. تمرین ۱۳-۱۵.

استفاده از گروه‌های مرتبه اول

(دسته‌های) گروه‌های دوری متفاوتی وجود دارند که در آن‌ها، باور بر آن است که مسائل لگاریتم گسسته و دیفی-هلمن، سخت هستند. باین‌حال، گروه‌های دوری دارای مرتبه‌ی اول به دلایلی که اکنون توضیح می‌دهیم، ترجیح داده می‌شوند.

یک دلیل برای ترجیح دادن گروه‌های با مرتبه‌ی اول این است که مسئله‌ی لگاریتم گسسته در چنین گروه‌هایی، سخت‌ترین حالت خود را دارد. این امر نتیجه‌ی الگوریتم پولیگ-هلمن، توصیف‌شده در فصل نهم، است که نشان می‌دهد مسئله‌ی لگاریتم گسسته در یک گروه با مرتبه‌ی q ساده‌تر خواهد شد اگر q دارای عوامل اول (کوچک) باشد. این امر ضرورتاً به این معنا نیست که مسئله‌ی لگاریتم

گسسته در گروه‌های با مرتبه‌ی غیر اول، ساده است؛ تنها به این معنا است که مسئله، «ساده‌تر» می‌شود.

این واقعیت به مطلب بالا مرتبط است که مسئله‌ی DDH ساده خواهد بود اگر مرتبه‌ی گروه، q ، دارای عوامل اول کوچک باشد. به‌عنوان مثالی از این پدیده می‌توانید به تمرین ۱۳-۱۵ مراجعه کنید.

یک دلیل دیگر برای استفاده از گروه‌های مرتبه اول آن است که یافتن یک مولد در چنین گروه‌هایی، بدیهی است. این امر ناشی از نتیجه‌ی فرعی ۸-۵۵ است که می‌گوید هر عنصر یک گروه مرتبه اول (به‌غیر از عنصر همانی) یک مولد است. در مقابل، یافتن کارای یک مولد برای یک گروه دوری دلخواه نیازمند آن است که تجزیه‌ی مرتبه‌ی گروه، معلوم باشد (ن.ک. ضمیمه‌ی ب-۳).

اثبات‌های امنیت برای بعضی از ساختارهای رمزنگاری نیازمند محاسبه‌ی وارون‌های ضربی بعضی نماهای خاص است (مثالی از این‌دست را در بخش ۸-۴ مشاهده خواهیم کرد). هنگامی که مرتبه‌ی گروه، اول باشد، هر نمای غیر صفر، وارون‌پذیر خواهد بود که باعث می‌شود این محاسبه، ممکن شود.

یک دلیل نهایی برای کار کردن با گروه‌های مرتبه اول در وضعیت‌هایی اعمال می‌شود که در آن‌ها، مسئله‌ی دیفی-هلمن تصمیمی باید سخت باشد. با در نظر گرفتن یک گروه G با مولد g ، مسئله‌ی DDH تبدیل می‌شود به تشخیص دادن چندگانه‌های با شکل $(h_1, h_2, DH_g(h_1, h_2))$ برای h_1, h_2 یکنواخت و چندگانه‌های با شکل (h_1, h_2, y) برای h_1, h_2, y یکنواخت. یک شرط لازم برای سخت بودن مسئله‌ی DDH این است که خود $DH_g(h_1, h_2)$ باید تمایزناپذیر از یک عنصر گروهی یکنواخت باشد. می‌توان نشان داد که $DH_g(h_1, h_2)$ «نزدیک» به یکنواخت خواهد بود (به معنایی که اینجا تعریف نمی‌کنیم) هنگامی که مرتبه‌ی گروه، q ، اول باشد، چیزی که در غیر این صورت صدق نخواهد کرد.

۸-۳-۳- کار کردن با (زیرگروه‌های) $\mathbb{Z}_p^*$

گروه‌های با شکل $\mathbb{Z}_p^*$ ، برای p که اول باشد، یک دسته از گروه‌های دوری را به دست می‌دهند که در آن‌ها، باور بر آن است که مسئله‌ی لگاریتم گسسته، سخت است. به شکل عینی، فرض کنید G یک الگوریتم باشد که با دریافت ورودی 1^n ، یک p اول n -بیتی یکنواخت را انتخاب کرده و به‌عنوان خروجی، p و مرتبه‌ی گروه $q = p - 1$ را به همراه یک مولد g برای $\mathbb{Z}_p^*$ می‌دهد. (بخش ۸-۲-۱ الگوریتم‌های کارا برای انتخاب یک عدد اول تصادفی را موردبحث قرار می‌دهد و ضمیمه‌ی ب-۳ نشان می‌دهد که چگونه می‌توان به شکل کارا یک مولد برای $\mathbb{Z}_p^*$ با داشتن تجزیه‌ی $p - 1$ یافت.) نمایش $\mathbb{Z}_p^*$ در اینجا، یک نمایش بدیهی است که در آن، عناصر به‌صورت اعداد صحیح بین 1 و $p - 1$ نشان داده می‌شوند. حدس می‌زنیم که مسئله‌ی لگاریتم گسسته نسبت به G از این نوع، سخت باشد.

با این حال، گروه دوری $\mathbb{Z}_p^*$ (برای $p > 3$ اول) دارای مرتبه‌ی اول نیست. (ترجیح برای گروه‌های با مرتبه‌ی اول در بخش قبلی مورد بحث قرار گرفت.) مسئله‌ی مشکل‌سازتر آن است که مسئله‌ی دیفی-هلمن تصمیمی به طول کلی در چنین گروه‌هایی، سخت نیست (ن.ک. تمرین ۱۳-۱۵) و بنابراین، این گروه‌ها برای کاربردهای رمزنگاری بر اساس فرض DDH که در فصول بعدی مورد بررسی قرار می‌گیرد، قابل قبول نیستند.

این مسائل را می‌توان با استفاده از یک زیرگروه مرتبه اول از $\mathbb{Z}_p^*$ حل نمود. فرض کنید $p = rq + 1$ که در آن، p و q هم اول هستند. اثبات می‌کنیم که $\mathbb{Z}_p^*$ دارای یک زیرگروه $\mathbb{G}$ با مرتبه‌ی q داده شده توسط مجموعه‌ی «باقیمانده‌های $\mathbb{Z}_p^*$ به پیمانه‌ی p »، یعنی مجموعه عناصر $\{[h^r \bmod p] | h \in \mathbb{Z}_p^*\}$ است که برابر با توان $\mathbb{Z}_p^*$ یک $h \in \mathbb{Z}_p^*$ هستند.

قضیه ۸-۶۴

فرض کنید $p = rq + 1$ با p, q اول. آنگاه

$$\mathbb{G} \stackrel{\text{def}}{=} \{[h^r \bmod p] | h \in \mathbb{Z}_p^*\}$$

یک زیرگروه از $\mathbb{Z}_p^*$ با مرتبه‌ی q است.

اثبات

اثبات اینکه $\mathbb{G}$ یک زیرگروه است، سراسر است و حذف شده است. اثبات می‌کنیم که $\mathbb{G}$ دارای مرتبه‌ی q است به این صورت که نشان می‌دهیم تابع $f_r: \mathbb{Z}_p^* \rightarrow \mathbb{G}$ تعریف شده توسط $f_r(g) = [g^r \bmod p]$ یک تابع 1 -به- 1 است. (از آنجاکه $|\mathbb{Z}_p^*| = p - 1$ ، این امر نشان می‌دهد که $|\mathbb{G}| = (p - 1)/r = q$.) برای مشاهده‌ی این امر، فرض کنید g یک مولد برای $\mathbb{Z}_p^*$ باشد به صورتی که $g^0, \dots, g^{p-2}$ همگی عناصر $\mathbb{Z}_p^*$ باشند. بر اساس گزاره‌ی ۸-۵۳، داریم $(g^i)^r = (g^j)^r$ اگر و تنها اگر $ir = jr \bmod (p - 1)$ یا به شکل برابر، $p - 1 \mid (i - j)r$ باشد. از آنجاکه $p - 1 = rq$ ، این امر برابر است با $q \mid (i - j)$ برای هر $i, j \in \{0, \dots, p - 2\}$ ثابت، این ارم به این معنا است که مجموعه‌ی مقادیر $\{0, \dots, p - 2\}$ که برای آن‌ها، $(g^i)^r = (g^j)^r$ دقیقاً برابر با مجموعه‌ای از r مقدار متمایز

$$\{j, j + q, j + 2q, \dots, j + (r - 1)q\},$$

همگی کاهش یافته به پیمانه‌ی $1 - p$ است. (توجه کنید که $(j + rq = j \bmod (p - 1))$ این امر اثبات می‌کند که f_r یک تابع 1 -به- 1 است. ■

علاوه بر نشان دادن وجود یک زیرگروه مناسب، این قضیه همچنین به شکل ضمنی نشان می‌دهد که تولید یک عنصر یکنواخت از $\mathbb{G}$ و آزمودن اینکه آیا یک عنصر مفروض از $\mathbb{Z}_p^*$ درون $\mathbb{G}$ قرار دارد یا خیر

هم ساده است. به صورت خاص، انتخاب یک عنصر یکنواخت از $\mathbb{G}$ را می‌توان با انتخاب $h \in \mathbb{Z}_p^*$ یکنواخت و محاسبه‌ی $[h^r \bmod p]$ انجام داد. از آنجاکه $\mathbb{G}$ دارای مرتبه‌ی اول است، هر عنصری در $\mathbb{G}$ به‌غیر از عنصر همانی، یک مولد برای $\mathbb{G}$ است. در نهایت، می‌توان تعیین نمود که آیا هر $h \in \mathbb{Z}_p^*$ همچنین در $\mathbb{G}$ نیز هست یا خیر به این صورت که بررسی کنیم آیا $h^q \stackrel{?}{=} 1 \bmod p$ برای مشاهده‌ی عملکرد این روش، فرض کنید $h = g^i$ برای g که یک مولد برای $\mathbb{Z}_p^*$ است و $i \in \{0, \dots, p - 2\}$ باشد. آنگاه با استفاده از گزاره‌ی ۸-۵۳ داریم:

$$\begin{aligned} h^q &= 1 \bmod p \Leftrightarrow g^{iq} = 1 \bmod p \\ &\Leftrightarrow iq = 0 \bmod (p - 1) \Leftrightarrow r \mid iq \Leftrightarrow r \mid i, \end{aligned}$$

بنابراین، $h \in \mathbb{G}$ برای یک c مفروض و $h = g^i = g^{cr} = (g^c)^r$

الگوریتم ۸-۶۵ بحث بالا را در برمی‌گیرد. در این الگوریتم، فرض می‌کنیم n نشان‌دهنده‌ی طول q مرتبه‌ی گروه باشد و فرض می‌کنیم که l نشان‌دهنده‌ی طول p ، پیمانه‌ی مورد استفاده، باشد. رابطه‌ی بین این پارامترها در ادامه مورد بحث قرار گرفته است.

الگوریتم ۸-۶۵

یک الگوریتم تولید گروه $\mathbb{G}$

ورودی: پارامتر امنیتی 1^n ، پارامتر $l = l(n)$

خروجی: گروه دوری $\mathbb{G}$ ، مرتبه‌ی (اول) آن q و یک مولد g

یک عدد اول n -بیتی یکنواخت q را تولید کنید.

یک عدد اول l -بیتی را انتخاب کنید به صورتی که $q \mid (p - 1)$

// جزئیات چگونگی انجام این کار را حذف می‌کنیم

یک $h \in \mathbb{Z}_p^*$ یکنواخت با $h \neq 1$ را انتخاب کنید.

قرار دهید $g := [h^{(p-1)/q} \bmod p]$

p, q, g را بازگردانید.

// $\mathbb{G}$ زیرگروه مرتبه- q از $\mathbb{Z}_p^*$ است.

انتخاب l

فرض کنید $q \parallel n$ و $l \parallel p$. دو نوع از الگوریتم‌ها برای محاسبه‌ی لگاریتم‌های گسسته در زیرگروه‌های مرتبه- q از $\mathbb{Z}_p^*$ شناخته شده‌اند (ن.ک. بخش ۹-۲): آن‌هایی که در زمان $O(\sqrt{q})$ اجرا می‌شوند و آن‌هایی که در زمان $O(2^{n/4})$ اجرا می‌شوند. (ن.ک. $O(2^{n/4}) = O(2^{(l/4)}) = O(2^{(l/4) \cdot (\log p)^{1/4}})$) می‌شوند. با مشخص کردن یک پارامتر امنیتی دلخواه n ، پارامتر l باید به صورتی انتخاب شود که این زمان‌ها را متعادل نماید. (اگر l اندکی کوچک‌تر باشد، امنیت کاهش می‌یابد؛ اگر l اندکی بزرگ‌تر

باشد، عملیات در G کارایی کمتری خواهند داشت درحالی که افزایش امنیتی هم ارائه نخواهد شد. همچنین ن.ک. بخش ۹-۳.

در عمل، مقادیر استاندارد شده (برای مثال، پیشنهادی توسط $NIST$) برای p و q و یک مولد g استفاده می‌شوند و نیازی به تولید پارامترها نیست.

مثال ۸-۶۶

گروه $\mathbb{Z}_{11}^*$ با مرتبه ۱۰ را در نظر بگیرید. اجازه دهید سعی کنیم یک مولد برای این گروه بیابیم. ۲ را امتحان می‌کنیم:

توان‌های ۲	۲ ^۰	۲ ^۱	۲ ^۲	۲ ^۳	۲ ^۴	۲ ^۵	۲ ^۶	۲ ^۷	۲ ^۸	۲ ^۹
مقادیر	۱	۲	۴	۸	۵	۱۰	۹	۷	۳	۶

(تمام مقادیر بالا به پیمانه ۱۱ محاسبه می‌شوند). بار اول شانس آوردیم- عدد ۲ یک مولد است! اجازه دهید ۳ را امتحان کنیم:

توان‌های ۳	۳ ^۰	۳ ^۱	۳ ^۲	۳ ^۳	۳ ^۴	۳ ^۵	۳ ^۶	۳ ^۷	۳ ^۸	۳ ^۹
مقادیر	۱	۳	۹	۵	۴	۱	۳	۹	۵	۴

مشاهده می‌کنیم که ۳ یک مولد برای کل گروه نیست. درواقع، یک زیرگروه $G = \{1, 3, 9, 5\}$ با مرتبه ۵ را تولید می‌کند. اکنون، اجازه دهید ببینیم با ۱۰ چه اتفاقی می‌افتد.

توان‌های ۱۰	۱۰ ^۰	۱۰ ^۱	۱۰ ^۲	۱۰ ^۳	۱۰ ^۴	۱۰ ^۵	۱۰ ^۶	۱۰ ^۷	۱۰ ^۸	۱۰ ^۹
مقادیر	۱	۱۰	۱	۱۰	۱	۱۰	۱	۱۰	۱	۱۰

در این حالت، ما یک زیرگروه با مرتبه ۲ را تولید می‌کنیم.

برای اهداف رمزنگاری، می‌خواهیم در یک گروه مرتبه اول کار کنیم. از آنجاکه $11 = 2 \cdot 5 + 1$ ، می‌توانیم قضیه ۸-۶۴ را با $q = 5$ و $r = 2$ یا با $q = 2$ و $r = 5$ اعمال کنیم. در حالت اول، این قضیه به ما می‌گوید که مربعات تمام عناصر $\mathbb{Z}_{11}^*$ باید یک زیرگروه با مرتبه ۵ را به دست دهند. این امر را به سادگی می‌توان بررسی کرد:

عنصر:	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
مربع:	۱	۴	۹	۵	۳	۳	۵	۹	۴	۱

در بالا مشاهده کردیم که ۳ یک مولد برای این زیرگروه است. (درواقع، از آنجاکه این زیرگروه، اول است، هر عنصر در زیرگروه به‌غیر از ۱ یک مولد برای زیرگروه است). با قرار دادن $q = 2$ و $r = 5$ ، قضیه ۸-۶۴ به ما می‌گوید که محاسبه توان‌های ۵ام، یک زیرگروه با مرتبه ۲ را می‌دهد. می‌توان

بررسی کرد که این امر منجر به یک زیرگروه مرتبه ۲ تولید شده توسط ۱۰ می‌شود که قبلاً مشاهده کردیم.

زیرگروه‌های میدان‌های متناهی

این باور نیز وجود دارد که مسئله‌ی لگاریتم گسسته در گروه ضربی از یک میدان متناهی با مشخصه بزرگ هنگام استفاده از نمایش چندجمله‌ای، سخت است. (ضمیمه الف-۵ اطلاعات مختصری در خصوص میدان‌های متناهی ارائه می‌کند). یادآوری می‌کنیم که برای هر عدد اول p و عدد صحیح k ، یک میدان (منحصربه‌فرد) $\mathbb{F}_{p^k}$ با مرتبه p^k وجود دارد (ن.ک. قضیه الف-۲۱). اگر q یک عامل اول بزرگ برای $p^k - 1$ باشد، آنگاه قضیه ۸-۶۴ نشان می‌دهد که $\mathbb{F}_{p^k}^*$ دارای یک زیرگروه دوری با مرتبه q است. (تنها ویژگی $\mathbb{Z}_p^*$ که در اثبات آن قضیه استفاده می‌کنیم این است که $\mathbb{Z}_p^*$ دوری است). این امر یک انتخاب دیگر برای گروه‌های مرتبه اول می‌دهد که در آن‌ها باور بر این است که مسائل لگاریتم گسسته و دیفی-هلمن، سخت هستند. استفاده‌ی ما از $\mathbb{Z}_p^*$ در این بخش، متناظر با حالت خاص $k = 1$ است.

۴-۳-۸- خم‌های بیضوی

گروه‌هایی که تاکنون بر آن‌ها تمرکز کرده‌ایم همگی به‌صورت مستقیم مبتنی بر حساب پیمانه‌ای بودند. یک دسته‌ی دیگر از گروه‌هایی که برای رمزنگاری مهم است توسط گروه‌های شامل «قطعاتی بر روی خم‌های بیضوی» به دست داده می‌شوند. چنین گروه‌هایی علی‌الخصوص از دیدگاه رمزنگاری جالب‌توجه هستند به این دلیل که برخلاف $\mathbb{Z}_p^*$ یا گروه ضربی یک میدان متناهی، در حال حاضر هیچ الگوریتم زمان زیر نمایی برای حل مسئله‌ی لگاریتم گسسته در گروه‌های خم بیضوی هنگام انتخاب درست آن‌ها، وجود ندارد. (برای بحث گسترده‌تر، ن.ک. بخش ۹-۳). برای سامانه‌های رمزنگاری مبتنی بر فرضیات لگاریتم گسسته یا دیفی-هلمن، این امر به این معنا است که پیاده‌سازی‌های مبتنی بر گروه‌های خم بیضوی دارای کارایی بالاتری نسبت به پیاده‌سازی‌های مبتنی بر زیرگروه‌های مرتبه اول از $\mathbb{Z}_p^*$ در هر سطح مفروض امنیتی خواهند بود. در این بخش، تنها یک مقدمه‌ی مختصر در خصوص این حوزه ارائه می‌کنیم. درک عمیق‌تر مباحث بیان شده در اینجا نیازمند ریاضیات پیچیده‌تری نسبت به آن چیزی است که مایل به فرض کردن برای خواننده هستیم. افراد علاقه‌مند به مطالعه‌ی بیشتر در خصوص این موضوع می‌توانند به مراجع مورد اشاره در انتهای این فصل، مراجعه کنند.

فرض کنید $p \geq 5$ یک عدد اول باشد. یک معادله‌ی E را بر اساس متغیرهای x و y به شکل ذیل در نظر بگیرید

^{۲۹} این نظریه را می‌توان تطبیق داد تا با حالت $p = 2$ or 3 کار کند ولی این امر باعث پیچیدگی‌های بیشتر خواهد شد. درواقع، خم‌های بیضوی می‌توانند در سطح میدان‌های متناهی یا نامتناهی دلخواه، تعریف شوند (ن.ک. بخش الف-۵) و بحث ما تا حد

$$y^2 = x^2 + Ax + B \pmod{p} \quad (A.1)$$

که در آن، $A, B \in \mathbb{Z}_p$ اعداد ثابت با $4A^2 + 27B^2 \not\equiv 0 \pmod{p}$ هستند. (این امر تضمین می‌کند که معادله $x^2 + Ax + B = 0 \pmod{p}$ هیچ ریشه‌ی تکراری نداشته باشد.) فرض کنید $E(\mathbb{Z}_p)$ نشان‌دهنده‌ی مجموعه زوج‌های $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ باشد که در معادله‌ی بالا صدق می‌کنند، به همراه یک مقدار خاص O که هدف آن را به‌زودی مورد بحث قرار می‌دهیم؛ یعنی،

$$E(\mathbb{Z}_p) \stackrel{\text{def}}{=} \{(x, y) | x, y \in \mathbb{Z}_p \text{ and } y^2 = x^2 + Ax + B \pmod{p}\} \cup \{O\}.$$

عناصر $E(\mathbb{Z}_p)$ را «نقاط» بر روی خم بیضوی E تعریف‌شده توسط معادله‌ی (A.1) نامیده و O نیز «نقطه در بی‌نهایت» نامیده می‌شود.

مثال ۸-۶۷

یک عنصر $y \in \mathbb{Z}_p^*$ یک «مانده‌ی مرتبه دوم به پیمانه‌ی p » است اگر یک $x \in \mathbb{Z}_p^*$ وجود داشته باشد به صورتی که $x^2 = y \pmod{p}$ در این حالت، می‌گوییم x یک جذر (ریشه دوم) از y است. برای $p > 2$ اول، نیمی از عناصر در $\mathbb{Z}_p^*$ مانده‌ی مرتبه دوم هستند و هر مانده‌ی مرتبه دوم دقیقاً دارای دو جذر است. (ن.ک. بخش ۱۳-۴-۱.)

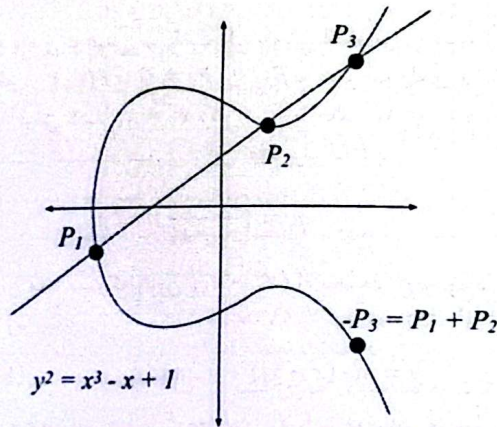
فرض کنید $f(x) \stackrel{\text{def}}{=} x^2 + 2x + 3$ و خم $E: y^2 = f(x) \pmod{7}$ را در نظر بگیرید. هر مقدار x که برای آن، $f(x)$ یک مانده‌ی مرتبه دوم به پیمانه‌ی ۷ باشد، دو نقطه بر روی خم می‌دهد؛ مقادیر x که برای آن‌ها، $f(x)$ یک مانده‌ی غیر مرتبه دوم باشد، بر روی خم نیستند؛ مقادیر x که برای آن‌ها، $f(x) = 0 \pmod{7}$ یک نقطه بر روی خم به دست می‌دهند. این امر به ما اجازه می‌دهد تا نقاط روی خم را تعیین کنیم:

- $f(0) = 3 \pmod{7}$ ، یک نامانده‌ی مرتبه دوم به پیمانه‌ی ۷.
- $f(1) = 0 \pmod{7}$ ، بنابراین ما نقطه‌ی $(1, 0) \in E(\mathbb{Z}_7)$ را به دست می‌آوریم.
- $f(2) = 3 \pmod{7}$ ، یک نامانده‌ی مرتبه دوم به پیمانه‌ی ۷.
- $f(3) = 4 \pmod{7}$ ، یک مانده‌ی مرتبه دوم به پیمانه‌ی ۷ با جذرهای ۲ و ۵. این امر نقاط $(3, 2), (3, 5) \in E(\mathbb{Z}_7)$ را می‌دهد.
- $f(4) = 2 \pmod{7}$ ، یک مانده‌ی مرتبه دوم به پیمانه‌ی ۷ با جذرهای ۳ و ۴. این امر نقاط $(4, 3), (4, 4) \in E(\mathbb{Z}_7)$ را می‌دهد.
- $f(5) = 3 \pmod{7}$ ، یک نامانده‌ی مرتبه دوم به پیمانه‌ی ۷.
- $f(6) = 6 \pmod{7}$ ، یک نامانده‌ی مرتبه دوم به پیمانه‌ی ۷.

زیادی به میدان‌های ویژگی ۲ یا ۳ برابر نیستند، قابل تعمیم است. خم‌های دودویی (یعنی خم‌های در سطح میدان‌های ویژگی ۲) علی‌الخصوص در رمزنگاری مهم هستند ولی ما آن‌ها را در اینجا مورد بحث قرار نمی‌دهیم.

با در نظر گرفتن نقطه در بی‌نهایت، ۶ نقطه در $E(\mathbb{Z}_7)$ وجود دارد.

یک روش مفید برای در نظر گرفتن $E(\mathbb{Z}_p)$ نگاه کردن به نمودار معادله‌ی (A.1) روی اعداد حقیقی (یعنی معادله $y^2 = x^2 + Ax + B$ بدون پیمانه‌ی کاهشی p) به‌مانند تصویر ۸-۲ است. این تصویر دقیقاً منطبق با $E(\mathbb{Z}_p)$ نیست به این دلیل که برای مثال $E(\mathbb{Z}_p)$ دارای یک تعداد متناهی نقطه است (به‌هر حال $\mathbb{Z}_p$ یک مجموعه‌ی متناهی است) درحالی‌که تعداد راه‌حل‌های نامتناهی برای همین معادله وجود خواهد داشت اگر اجازه دهیم x و y در دامنه‌ی کل اعداد حقیقی باشند. با این حال، این تصویر، برداشت مفیدی را می‌دهد. در چنین تصویری، می‌توانیم «نقطه‌ی در بی‌نهایت» O را قرار گرفته در بالای محور y و بر روی تمام خطوط عمودی، در نظر بگیریم.



تصویر ۸-۲: یک خم بیضوی روی اعداد حقیقی

می‌توان نشان داد که هر خط قطع کننده $E(\mathbb{Z}_p)$ آن را دقیقاً در ۳ نقطه قطع می‌کند که در آن‌ها، (۱) یک نقطه‌ی P دو بار شمرده می‌شود اگر خط، مماس با خم در نقطه‌ی P باشد و (۲) نقطه در بی‌نهایت نیز شمرده خواهد شد اگر خط، عمودی باشد. این واقعیت برای تعریف یک عمل دوتایی با نام «جمع» و نشان داده‌شده به‌صورت $+$ بر روی نقاط $E(\mathbb{Z}_p)$ به شکل زیر، مورد استفاده قرار می‌گیرد:

- نقطه‌ی O به‌عنوان یک همانی (جمعی) تعریف می‌شود؛ یعنی، برای تمام $P \in E(\mathbb{Z}_p)$ ، $P + O = O + P = P$ تعریف می‌کنیم.
- برای دو نقطه‌ی $P_1, P_2 \neq O$ بر روی E ، جمع آن‌ها، $P_1 + P_2$ را با ترسیم یک خط گذرا از P_1, P_2 (اگر $P_1 = P_2$ ، آنگاه خط را به‌صورت مماس بر خم در P_1 ترسیم می‌کنیم) و یافتن یک نقطه تقاطع سوم P_3 بین این خط و $E(\mathbb{Z}_p)$ ارزیابی می‌کنیم؛ سومین نقطه‌ی

نقاط ممکن است $P_r = 0$ باشد اگر خط، عمودی باشد. اگر $P_r = (x, y) \neq 0$ ، آنگاه $P_1 + P_r \stackrel{\text{def}}{=} (x, -y)$ را تعریف می‌کنیم. (به شکل تصویری، این امر متناظر با بازتاب دادن P_r در محور x است.) اگر $P_r = 0$ باشد، آنگاه $P_1 + P_r \stackrel{\text{def}}{=} 0$.

اگر $P = (x, y) \neq 0$ یک نقطه از $E(\mathbb{Z}_p)$ باشد، آنگاه $-P \stackrel{\text{def}}{=} (x, -y)$ (که به‌وضوح همچنین نقطه‌ای از $E(\mathbb{Z}_p)$ نیز هست) وارون منحصربه‌فرد P است. درواقع، خط عبوری از (x, y) و $(x, -y)$ ، عمودی است و بنابراین، قاعده‌ی جمع نشان می‌دهد که $P + (-P) = 0$. (اگر $y = 0$ ، آنگاه $P = -P$ و $(x, y) = (x, -y) = -P$ ولی در این صورت، خط مماس در P عمودی خواهد بود و بنابراین در اینجا هم $0 = 0$ ، البته، $(P + (-P)) = 0$.)

مشخص کردن قانون جمع به‌صورت عینی، سراسر است ولی طولانی است. فرض کنید $P_1 = (x_1, y_1)$ و $P_r = (x_r, y_r) \neq 0$ با $P_1, P_r \neq 0$ به شکل معادله‌ی (۸.۱) باشند. برای ساده نگه‌داشتن محاسبات، فرض کنید $x_1 \neq x_r$ (کار کردن با حالت $x_1 = x_r$ همچنان سراسر است ولی طولانی‌تر هم هست). شیب خط عبوری از این نقاط برابر است با:

$$m \stackrel{\text{def}}{=} \left[\frac{y_r - y_1}{x_r - x_1} \bmod p \right];$$

فرض ما که $x_1 \neq x_r$ به این معنا است که وارون $(x_r - x_1)$ به پیمانه‌ی p وجود دارد. خط عبوری از P_1 و P_r دارای معادله‌ی زیر است:

$$y = m \cdot (x - x_1) + y_1 \bmod p. \quad (8.2)$$

برای یافتن سومین نقطه‌ی برخورد این خط با E ، معادله‌ی بالا را در معادله‌ی مربوط به E قرار دهید تا به دست آوریم:

$$(m \cdot (x - x_1) + y_1)^2 = x^2 + Ax + B \bmod p.$$

مقادیر x که در این معادله صدق می‌کنند عبارت‌اند از x_1, x_r و

$$x_r \stackrel{\text{def}}{=} [m^2 - x_1 - x_r \bmod p].$$

دو راه‌حل اول متناظر با نقاط اولیه‌ی P_1 و P_r هستند درحالی‌که راه‌حل سوم عبارت است از مختصه‌ی x سومین نقطه‌ی برخورد، P_r با قرار دادن x_r در معادله‌ی (۸.۲)، درمی‌یابیم که مختصه‌ی y متناظر با x_r برابر با $y_r = [m \cdot (x_r - x_1) + y_1 \bmod p]$ است. برای به دست آوردن پاسخ مطلوب $P_1 + P_r$ ، علامت y را عوض می‌کنیم تا به دست آوریم:

$$(x_1, y_1) + (x_r, y_r) = ([m^2 - x_1 - x_r \bmod p], [m \cdot (x_1 - x_r) - y_1 \bmod p]).$$

در گزاره‌ی ذیل، این مطلب را خلاصه نموده و تعمیم می‌دهیم.

گزاره‌ی ۸-۶۸

فرض کنید $p \geq 5$ عددی اول باشد و فرض کنید E عبارت باشد از خم بیضوی داده‌شده توسط $y^2 = x^3 + Ax + B \bmod p$ که در آن، $4A^3 + 27B^2 \not\equiv 0 \bmod p$. فرض کنید $P_1, P_r \neq 0$ نقاط روی E با $P_1 = (x_1, y_1)$ و $P_r = (x_r, y_r)$ باشند.

- (۱) اگر $x_1 \neq x_r$ ، آنگاه $P_1 + P_r = (x_r, y_r)$ با $y_r = [m \cdot (x_1 - x_r) - y_1 \bmod p]$ و $x_r = [m^2 - x_1 - x_r \bmod p]$ که در آن، $m = \left[\frac{y_r - y_1}{x_r - x_1} \bmod p \right]$.
- (۲) اگر $x_1 = x_r$ ولی $y_1 \neq y_r$ ، آنگاه $P_1 = -P_r$ و بنابراین، $P_1 + P_r = 0$.
- (۳) اگر $P_1 = P_r$ و $y_1 = 0$ ، آنگاه $P_1 + P_r = 2P_1 = 0$.
- (۴) اگر $P_1 = P_r$ و $y_1 \neq 0$ ، آنگاه $P_1 + P_r = 2P_1 = (x_r, y_r)$ و $y_r = [m \cdot (x_1 - x_r) - y_1 \bmod p]$ و $x_r = [m^2 - 2x_1 \bmod p]$ که در آن، $m = \left[\frac{2y_1}{x_1} \bmod p \right]$.

به شکلی عجیب، مجموعه نقاط $E(\mathbb{Z}_p)$ به همراه قاعده‌ی جمع تعریف‌شده در بالا، یک گروه اَبلی را شکل می‌دهند که گروه خم بیضوی E نامیده می‌شود. جابه‌جایی از شیوه‌ی تعریف جمع، نتیجه می‌شود، 0 به‌عنوان همانی عمل می‌کند و قبلاً مشاهده کردیم که هر نقطه بر روی $E(\mathbb{Z}_p)$ دارای یک وارون در $E(\mathbb{Z}_p)$ است. ویژگی که واریسی آن دشوار است، شرکت‌پذیری است که خواننده‌ی شکاک می‌تواند آن را از طریق محاسبات خسته‌کننده، بررسی کند. یک اثبات روشن‌گر تر که شامل محاسبات آشکار نیست به هندسه‌ی جبری تکیه می‌کند.

مثال ۸-۶۹

خم مثال ۸-۶۷ را در نظر بگیرید. شرکت‌پذیری را برای سه نقطه‌ی خاص نشان می‌دهیم. فرض کنید $P_1 = (1, 0)$ و $P_r = Q_r = (4, 3)$. هنگام محاسبه‌ی $P_1 + P_r$ به دست می‌آوریم $m = [(3 - 0) \cdot (4 - 1)^{-1} \bmod 7] = 3$ و $[1^2 - 1 - 4 \bmod 7] = 3$. بنابراین،

$$P_r \stackrel{\text{def}}{=} P_1 + P_r = (3, [1 \cdot (1 - 3) - 0 \bmod 7]) = (3, 5);$$

نوجه کنید که این، درواقع یک نقطه بر روی $E(\mathbb{Z}_7)$ است. در ادامه اگر $P_r + Q_r$ را محاسبه کنیم، به دست می‌آوریم: $m = [(5 - 3) \cdot (4 - 3)^{-1} \bmod 7] = 5$ و $[5^2 - 3 - 4 \bmod 7] = 4$. بنابراین،

$$(P_1 + P_r) + Q_r = P_r + Q_r = (4, [5 \cdot (3 - 4) - 5 \bmod 7]) = (4, 4).$$

اگر $P_r + Q_r = 2P_r$ را محاسبه کنیم به دست می‌آوریم

$$m = [(3 \cdot 4^2 + 3) \cdot (2 \cdot 3)^{-1} \bmod 7] = 5 \text{ و } [5^2 - 2 \cdot 4 \bmod 7] = 3 \text{ بنابراین،}$$

$$P_r' \stackrel{\text{def}}{=} P_r + Q_r = (3, [5 \cdot (4 - 3) - 3 \bmod 7]) = (3, 2).$$

اگر در ادامه، $P_1 + P_r'$ را محاسبه کنیم، می‌یابیم $m = [2 \cdot (3 - 1)^{-1} \bmod 7] = 1$ و $[1^2 - m = 4 \bmod 7] = 4$ بنابراین

$$P_1 + (P_r + Q_r) = P_1 + P_r' = (4, [1 - 4] - 0 \bmod 7) = (4, 4),$$

$$\diamond P_1 + (P_r + Q_r) = (P_1 + P_r) + Q_r \text{ و}$$

یادآوری می‌کنیم که زمانی که یک گروه به صورت جمعی نوشته شده باشد، «توان رسانی» متناظر با جمع مکرر است. بنابراین، اگر یک نقطه‌ی مفروض P را در یک گروه خم بیضوی در نظر بگیریم، مسئله‌ی لگاریتم گسسته (به صورت غیررسمی) به مسئله‌ی محاسبه‌ی x از روی xP تبدیل می‌شود درحالی‌که مسئله‌ی دیفی-هلمن تصمیمی (به صورت غیررسمی) به مسئله‌ی تشخیص دادن چندگانه‌های با شکل (aP, bP, abP) از چندگانه‌های به شکل (aP, bP, cP) تبدیل می‌شود. باور بر آن است که این مسائل در گروه‌های خم بیضوی (یا زیرگروه‌های آن‌ها) که از مرتبه‌ی اول بالا هستند، سخت است مشروط به تعدادی شرط فنی که در ادامه به شکل مختصر به آن‌ها اشاره می‌کنیم.

اگر یک گروه خم بیضوی (یا زیرگروه آن) با مرتبه‌ی اول بالا را بخواهیم، اولین سؤالی که باید در نظر بگیریم این است که گروه‌های خم بیضوی، چقدر بزرگ هستند؟ همان‌طور که در مثال ۸-۶۷ اشاره شد، معادله‌ی $y^2 = f(x) \bmod p$ دارای دو راه حل است هرگاه که $f(x)$ یک مانده‌ی مرتبه دوم باشد و دارای یک راه حل خواهد بود زمانی که $f(x) = 0$ باشد. از آنجاکه نیمی از عناصر در $\mathbb{Z}_p^*$ مانده‌های مرتبه دوم هستند، به صورت تجربی انتظار داریم که $p = (p-1)/2 + 1$ نقطه بر روی خم بیابیم. به همراه نقطه در بی‌نهایت، این امر به این معنا است که باید در حدود $p+1$ نقطه در یک گروه خم بیضوی روی $\mathbb{Z}_p$ وجود داشته باشد. کران‌های $(Hasse)$ بیان می‌کند که این تخمین تجربی، صحیح است به این معنا که هر گروه خم بیضوی دارای «تقریباً» همین تعداد نقطه است.

قضیه ۸-۷۰ (کران هاس)

فرض کنید p یک عدد اول باشد و فرض کنید E یک خم بیضوی روی $\mathbb{Z}_p$ باشد. آنگاه، $p+1 - 2\sqrt{p} \leq |E(\mathbb{Z}_p)| \leq p+1 + 2\sqrt{p}$

این کران نشان می‌دهد که یافتن یک نقطه بر روی یک خم بیضوی مفروض $y^2 = f(x) \bmod p$ همواره آسان است: درواقع، $x \in \mathbb{Z}_p$ یکنواخت را انتخاب می‌کنیم، بررسی می‌کنیم که آیا $f(x)$ برابر با ۰ است یا یک مانده‌ی مرتبه دوم است. اگر این‌گونه باشد- فرض می‌کنیم y جذر $f(x)$ باشد.

(الگوریتم‌های مورد استفاده برای تعیین مانده‌ی مرتبه دوم بودن و محاسبه‌ی جذرها به پیمانه‌ی یک عدد اول در فصل سیزدهم مورد بحث قرار گرفته‌اند.) از آنجاکه نقاط بر روی خم بیضوی، متعدد هستند، نیازی نیست تعداد زیادی مقدار x را برای یافتن یک نقطه، امتحان کنیم.

کران هاس تنها یک دامنه برای اندازه‌ی یک گروه خم بیضوی ارائه می‌کند. با این حال، برای یک عدد اول ثابت p ، مشخص شده است که مرتبه‌ی یک خم بیضوی تصادفی روی $\mathbb{Z}_p$ (یعنی، یک خم تعریف شده توسط معادله‌ی (۸.۱)) که در آن، A, B به صورت یکنواخت در $\mathbb{Z}_p$ مشروط به محدودیت $4A^2 + 27B^2 \neq 0 \bmod p$ انتخاب می‌شوند «تزدیک» به توزیع یکنواخت در دامنه‌ی هاس است. همچنین، الگوریتم‌های کارایی- که توصیف و تحلیل آن‌ها بسیار فراتر از گستره‌ی این کتاب است- برای شمارش تعداد نقاط بر روی یک خم بیضوی وجود دارند. این امر نشان‌دهنده‌ی یک رویکرد برای تولید پارامتر خم بیضوی به صورت مشخص شده در الگوریتم ۸-۷۱ است.

الگوریتم ۸-۷۱

الگوریتم تولید گروه خم بیضوی، G

ورودی: پارامتر امنیتی 1^n

خروجی: گروه دوری G ، مرتبه‌ی (اول) آن q و یک مولد g

یک عدد اول n -بیتی یکنواخت p را تولید کنید.

تا زمانی که q یک عدد اول n -بیتی است، انجام بده:

انتخاب $A, B \leftarrow \mathbb{Z}_p$ با $4A^2 + 27B^2 \neq 0 \bmod p$

تعریف خم بیضوی E به صورت مشخص شده در معادله‌ی (۸.۱)

قرار دادن q به عنوان تعداد نقاط بر روی $E(\mathbb{Z}_p)$

انتخاب $g \in E(\mathbb{Z}_p) \setminus \{O\}$

بازگرداندن $(A, B, p), q, g$

// گروه خم بیضوی E است.

بعضی دسته‌های خاص از خم‌ها از لحاظ رمزنگاری ضعیف محسوب شده و باید از استفاده از آن‌ها اجتناب نمود. این موارد شامل گروه‌های خم بیضوی روی $\mathbb{Z}_p$ می‌شود که مرتبه‌ی آن‌ها برابر با p (خم‌های ناهنجار) یا $p+1$ (خم‌های ابرمنفرد) است یا مرتبه‌ی آن‌ها، $p^k - 1$ را برای k «کوچک»، تقسیم می‌کند. بحث کامل در این رابطه فراتر از گستره‌ی این کتاب است. در عمل، منحنی‌های (خم‌های) استاندارد شده (از قبیل موارد پیشنهاد شده توسط $NIST$) مورد استفاده قرار می‌گیرند و تولید یک منحنی (خم) جدید پیشنهاد نمی‌شود.

ملاحظات کارایی

این بخش را با بحثی مختصر در خصوص تعداد بهبود استاندارد کارایی هنگام استفاده از خم‌های بیضی، به پایان می‌بریم.

فشرده‌سازی نقطه

یک دیدگاه مفید آن است که تعداد بیت‌های موردنیاز برای نشان دادن یک نقطه بر روی یک خم بیضی را می‌توان تقریباً به اندازه‌ی نصف کاهش داد. برای مشاهده‌ی این امر، توجه کنید که برای هر نقطه‌ی (x, y) بر روی یک خم بیضی $E: y^2 = f(x) \mod p$ ، حداکثر دو نقطه بر روی خم وجود دارند که دارای مختصه‌ی x برابر با x هستند: یعنی، (x, y) و $(x, -y)$. (ممکن است $y = 0$ باشد که در این حالت، این دو نقطه، یکی هستند.) بنابراین، می‌توانیم هر نقطه‌ی $P = (x, y)$ را از طریق مختصه‌ی x آن و یک بیت b مشخص کنیم که بین (حداکثر) دو امکان برای مقدار مختصه‌ی y آن، تمایز ایجاد می‌کند. با داشتن x و b ، می‌توانیم P را با محاسبه‌ی دو ریشه‌ی دوم y_1, y_2 معادله‌ی $y^2 = f(x) \mod p$ باز یابی کنیم؛ از آنجاکه $y_1 = -y_2 \mod p$ و بنابراین $y_1 = p - y_2$ ، دقیقاً یکی از y_1, y_2 کمتر از $p/2$ خواهد بود.

مختصات تصویری

نمایش نقاط خم بیضی به صورتی که تاکنون مشاهده کردیم - که در آن، یک نقطه‌ی P بر روی یک خم بیضی از طریق یک زوج عنصر میدانی (x, y) توصیف می‌شود - استفاده از مختصات آفین نامیده می‌شود. روش‌های دیگری برای نشان دادن نقاط با استفاده از مختصات تصویری وجود دارد که می‌توانند بهبودهایی در کارایی را به دست دهند. درحالی‌که این نمایش‌های جایگزین ممکن است دارای دلایل ریاضیاتی باشند، ما تنها آن‌ها را به عنوان کمک‌های محاسباتی مفید در نظر می‌گیریم.

نقاط در مختصات تصویری با استفاده از سه عنصر از $\mathbb{Z}_p$ نمایش داده می‌شوند. یک ویژگی جذاب آن است که یک نقطه دارای چندین نمایش است. هنگام استفاده از مختصات تصویری استاندارد، یک نقطه‌ی $P \neq O$ با نمایش (x, y) در مختصات آفین توسط هر سه‌گانه‌ی $(X, Y, Z) \in \mathbb{Z}_p^3$ نمایش داده می‌شود که برای آن، $X/Z = x \mod p$ و $Y/Z = y \mod p$ است. نقطه‌ی O توسط هر سه‌گانه‌ی $(0, Y, 0)$ با $Y \neq 0$ نمایش داده می‌شود و اینها تنها نقاط (X, Y, Z) با $Z = 0$ هستند. به سادگی می‌توانیم بین سیستم‌های مختصاتی، ترجمه کنیم: (x, y) در مختصات آفین را می‌توان به $(x, y, 1)$ در مختصات تصویری نگاشت نمود و (X, Y, Z) (با $Z \neq 0$) در مختصات تصویری به نمایش $([X/Z \mod p], [Y/Z \mod p])$ در مختصات آفین، نگاشت می‌شود.

اصلی‌ترین مزیت استفاده از مختصات تصویری آن است که می‌توانیم نقاط را بدون نیاز به محاسبه‌ی وارون‌ها به پیمانه‌ی p ، اضافه کنیم. (افزودن نقاط در مختصات آفین نیازمند محاسبه‌ی وارون‌ها است؛ نک. گزاره‌ی ۸-۶۸.) این امر را با استفاده از این واقعیت انجام می‌دهیم که نقاط دارای نمایش‌های

چندگانه هستند. برای مشاهده‌ی این امر، اجازه دهید قانون جمع را برای دو نقطه‌ی $P_1 = (X_1, Y_1, Z_1)$ و $P_2 = (X_2, Y_2, Z_2)$ با $P_1, P_2 \neq O$ (بنابراین، $Z_1, Z_2 \neq 0$) و $P_1 \neq \pm P_2$ (بنابراین، $X_1/Z_1 \not\equiv X_2/Z_2 \mod p$) به دست آوریم. (اگر P_1 یا P_2 برابر با O باشد، جمع بدیهی است.) حالت $P_1 = \pm P_2$ را نیز می‌توان مدیریت کرد، ولی ما جزئیات را در اینجا نمی‌آوریم.) می‌توانیم P_1 و P_2 را به صورت $(X_1/Z_1, Y_1/Z_1)$ و $(X_2/Z_2, Y_2/Z_2)$ در مختصات آفین نمایش دهیم، بنابراین

$$P_+ \stackrel{\text{def}}{=} P_1 + P_2 = (m^x - X_1/Z_1 - X_2/Z_2, m \cdot (X_1/Z_1 - m^x + X_1/Z_1 + X_2/Z_2) - Y_1/Z_1, 1),$$

که در آن

$$m = (Y_2/Z_2 - Y_1/Z_1)(X_2/Z_2 - X_1/Z_1)^{-1} = (Y_2Z_1 - Y_1Z_2)(X_2Z_1 - X_1Z_2)^{-1}$$

و تمام محاسبات به پیمانه‌ی p انجام می‌شوند. توجه کنید که برای نمایش P_+ از مختصات تصویری استفاده کرده و در بالا قرار می‌دهیم $Z_+ = 1$. ولی استفاده از مختصات تصویری به این معنا است که ما محدود به $Z_+ = 1$ نیستیم. با ضرب کردن هر مختصه در $Z_1Z_2(X_2Z_1 - X_1Z_2)^2 \neq 0 \mod p$ درمی‌یابیم که P_+ را می‌توان به این صورت هم نمایش داد:

$$P_+ = (vw, u(v^2X_1Z_2 - w) - v^2Y_1Z_2, Z_1Z_2v^2) \quad (۸.۳)$$

که در آن

$$u = Y_2Z_1 - Y_1Z_2, \quad v = X_2Z_1 - X_1Z_2, \\ w = u^2Z_1Z_2 - v^2 - 2v^2X_1Z_2 \quad (۸.۴)$$

نکته‌ای که باید دریافت آن است که محاسبات در معادلات (۸.۳) و (۸.۴) را می‌توان بدون اجبار برای انجام هیچ وارون سازی پیمانه‌ای، انجام داد.

دقیقاً به این دلیل که نقاط دارای نمایش‌های متعدد در مختصات تصویری هستند، تعدادی نکته‌ی ظریف ممکن است هنگام استفاده از مختصات تصویری به وجود آیند. (ما تاکنون به صورت آشکار فرض کرده‌ایم که اعضای گروه دارای نمایش‌های منحصر به فرد به صورت رشته‌های بیتی هستند.) علی‌الخصوص، یک نقطه‌ی نمایش داده شده در مختصات تصویری ممکن است اطلاعاتی را در مورد چگونگی به دست آمدن این نقطه به دست آورد که ممکن است به اطلاعات محرمانه وابسته باشد. برای حل این مسئله - و همین‌طور به دلایل کارایی - مختصات آفین باید برای ارسال و ذخیره‌ی نقاط استفاده شود و مختصات تصویری تنها به عنوان یک نمایش میانی طی انجام یک محاسبه مورد استفاده قرار گیرد (که نقاط به‌لازم مختصات تصویری در ابتدای انتهای محاسبه، تبدیل می‌شوند).

۴-۸- کاربردهای رمزنگاری

مدت زمانی را صرف بحث کردن در مورد نظریه‌ی اعداد و نظریه‌ی گروه و معرفی فرضیه‌های که به صورت عمومی باور بر این است که از لحاظ محاسباتی سخت هستند نمودیم. کاربردهای این فرضیات در مابقی کتاب مورد تمرکز خواهند بود ولی تعدادی مثال مختصر را در اینجا ارائه می‌کنیم.

۴-۸-۱- توابع و جایگشت‌های یک‌طرفه

توابع یک‌طرفه، پایه‌ی حداقلی در رمزنگاری هستند و برای رمزگذاری کلید خصوصی و کدهای احراز اصالت پیام، لازم و کافی‌اند. یک بحث کامل‌تر در خصوص نقش توابع یک‌طرفه در رمزنگاری در فصل هفتم آورده شده است؛ در اینجا تنها تعریفی از توابع یک‌طرفه را ارائه داده و نشان می‌دهیم که وجود آن‌ها از فرضیه‌های سخت نظریه‌ی اعدادی که در این فصل مشاهده کردیم، نتیجه خواهد شد.

به شکل غیررسمی، یک تابع f ، یک‌طرفه است اگر محاسبه‌ی آن، ساده باشد ولی وارون سازی آن، سخت باشد. آزمایش و تعریف ذیل، بیان تعریف ۷-۱ به صورتی دیگر، این موضوع را به صورت رسمی بیان می‌کند:

آزمایش وارون سازی $Invert_{A,f}(n)$:

- (۱) $x \in \{0,1\}^n$ یکنواخت را انتخاب کرده و $y = f(x)$ را محاسبه کنید.
- (۲) A به عنوان ورودی، 1^n و y را دریافت کرده و به عنوان خروجی، x' را ارائه می‌کند.
- (۳) خروجی آزمایش برابر با x است اگر و تنها اگر $f(x') = y$ باشد.

تعریف ۴-۸-۲

یک تابع $f: \{0,1\}^* \rightarrow \{0,1\}^*$ یک‌طرفه است اگر دو شرط زیر برقرار باشد:

- (۱) (سادگی محاسبه): یک الگوریتم زمان چندجمله‌ای وجود دارد که با دریافت ورودی x خروجی $f(x)$ را ارائه می‌کند.
- (۲) (سختی وارون سازی): برای تمام الگوریتم‌های PPT ، A یک تابع ناچیز $negl$ وجود دارد به صورتی که $\Pr[Invert_{A,f}(n) = 1] \leq negl(n)$.

اکنون به صورت رسمی نشان می‌دهیم که فرض تجزیه نشان دهنده‌ی وجود یک تابع یک‌طرفه است. فرض کنید Gen یک الگوریتم زمان چندجمله‌ای باشد که با دریافت ورودی 1^n ، خروجی (N, p, q) را ارائه می‌کند که در آن، $N = pq$ و p و q اعداد اول n -بیتی هستند مگر با احتمال ناچیز بر مبنای n . (ما از Gen به جای $GenModulus$ در اینجا تنها برای راحتی نوشتن استفاده می‌کنیم.) از آنجاکه در Gen زمان چندجمله‌ای اجرا می‌شود، یک کران بالای چندجمله‌ای بر روی تعداد بیت‌های تصادفی وجود دارد که الگوریتم مورد استفاده قرار می‌دهد. برای سادگی و به منظور بیان ایده‌های اصلی، فرض

می‌کنیم Gen همواره از حداکثر n بیت تصادفی بر روی ورودی 1^n استفاده می‌کند. در الگوریتم ۸-۷۳، یک تابع f_{Gen} را تعریف می‌کنیم که از ورودی خود به عنوان بیت‌های تصادفی برای اجرای Gen استفاده می‌کند. بنابراین، f_{Gen} یک تابع قطعی به صورت ملزم شده، است.

الگوریتم ۸-۷۳

الگوریتم محاسبه‌ی f_{Gen}

ورودی: رشته‌ی x با طول n

خروجی: عدد صحیح N

$Gen(1^n; x) = (N, p, q)$ را محاسبه کنید.

// یعنی، $Gen(1^n)$ را با استفاده از x به عنوان نوار تصادفی، اجرا کنید.
 N را بازگردانید.

اگر مسئله‌ی تجزیه نسبت به Gen ، سخت باشد، آنگاه به صورت شهودی، f_{Gen} یک تابع یک‌طرفه خواهد بود. مشخص است که محاسبه‌ی f_{Gen} ساده است. در رابطه با سختی وارون نمودن این تابع، توجه کنید که توزیع‌های زیر، یکسان هستند:

- (۱) پیمانه‌ی N خروجی از f_{Gen} هنگامی که $x \in \{0,1\}^n$ به صورت یکنواخت انتخاب می‌شود.
- (۲) پیمانه‌ی N خروجی توسط (الگوریتم تصادفی شده‌ی) $Gen(1^n)$.

اگر تجزیه‌ی پیمانه‌های N تولیدشده بر اساس توزیع دوم، سخت باشد، آنگاه همین امر برای پیمانه‌های N تولیدشده بر اساس اولین توزیع هم صدق می‌کند. به علاوه، با داشتن هر پیش تصویر x' از N نسبت به f_{Gen} (یعنی، یک x' که برای آن، $f_{Gen}(x') = N$) توجه کنید که الزام نمی‌کنیم که $x' = x$ باشد، به سادگی می‌توان یک عامل از N را با اجرای $Gen(1^n; x')$ برای به دست آوردن (N, p, q) و اعلام عوامل p و q بازیابی نمود. بنابراین، یافتن یک پیش تصویر از N نسبت به f_{Gen} به سختی تجزیه‌ی N است. به سادگی می‌توان این امر را به یک اثبات رسمی برای قضیه‌ی ذیل، تبدیل نمود:

قضیه ۴-۸-۷۴

اگر مسئله‌ی تجزیه نسبت به Gen سخت باشد، آنگاه f_{Gen} یک تابع یک‌طرفه است.

جایگشت‌های یک‌طرفه

همچنین می‌توانیم از فرضیات نظریه‌ی اعداد برای ساختن یک خانواده از جایگشت‌های یک‌طرفه استفاده کنیم. با بیان دوباره‌ی تعاریف ۷-۲ و ۷-۳ شروع می‌کنیم که برای حالت جایگشت‌ها، اختصاصی شده‌اند:

تعریف ۷۵-۸

یک سه‌گانه‌ی $\Pi = (Gen, Samp, f)$ از الگوریتم‌های زمان-چندجمله‌ای تصادفی یک خانواده از جایگشت‌ها است اگر موارد زیر برقرار باشد:

- (۱) الگوریتم تولید پارامتر Gen با دریافت ورودی 1^n ، پارامترهای I با $|I| \geq n$ را به‌عنوان خروجی ارائه می‌کند. هر مقدار I یک مجموعه‌ی $\mathcal{D}_I$ را تعریف می‌کند که دامنه و برد یک جایگشت را تشکیل می‌دهد (یعنی نگاشت دوسویی) $f_I: \mathcal{D}_I \rightarrow \mathcal{D}_I$.
- (۲) الگوریتم نمونه‌گیری $Samp$ با دریافت ورودی I ، یک عنصر از $\mathcal{D}_I$ دارای توزیع یکنواخت را به‌عنوان خروجی ارائه می‌کند.
- (۳) الگوریتم ارزیابی قطعی f با دریافت I و $x \in \mathcal{D}_I$ به‌عنوان ورودی، یک عنصر $y \in \mathcal{D}_I$ را به‌عنوان خروجی ارائه می‌کند. این امر را به‌صورت $y = f_I(x)$ می‌نویسیم.

با داشتن یک خانواده از توابع Π ، آزمایش زیر را برای هر الگوریتم $\mathcal{A}$ و پارامتر n در نظر بگیرید:

آزمایش وارون سازی $Invert_{\mathcal{A}, \Pi}(n)$:

- (۱) $Gen(1^n)$ برای به دست آوردن I اجرا می‌شود و سپس، $Samp(I)$ برای انتخاب یک $x \in \mathcal{D}_I$ یکنواخت، اجرا می‌شود. درنهایت، $y = f_I(x)$ محاسبه می‌شود.
- (۲) $\mathcal{A}$ ورودی‌های I و y را دریافت کرده و خروجی x' را ارائه می‌دهد.
- (۳) خروجی آزمایش برابر با ۱ است اگر و تنها اگر $y = f_I(x')$.

یک خانواده از جایگشت‌ها بر اساس مسئله‌ی RSA

ساختار ۷۷-۸

فرض کنید $GenRSA$ به‌مانند قبل باشد. یک خانواده از جایگشت‌ها را به‌صورت زیر تعریف کنید:

- Gen : با دریافت ورودی 1^n ، $GenRSA(1^n)$ را اجرا کنید تا (N, e, d) به دست آید و $I = \langle N, e \rangle$ را به‌عنوان خروجی ارائه دهید. قرار دهید $\mathcal{D}_I = \mathbb{Z}_N^*$.
- $Samp$: با دریافت $I = \langle N, e \rangle$ به‌عنوان ورودی، یک عنصر یکنواخت از $\mathbb{Z}_N^*$ انتخاب کنید.
- f : با دریافت $I = \langle N, e \rangle$ و $x \in \mathbb{Z}_N^*$ به‌عنوان ورودی، خروجی $[x^e \bmod N]$ را ارائه کنید.

تعریف ۷۶-۸

خانواده‌ی جایگشت‌های $\Pi = (Gen, Samp, f)$ یک‌طرفه است اگر برای تمام الگوریتم‌های زمان-چندجمله‌ای تصادفی $\mathcal{A}$ یک تابع ناچیز $negl$ وجود داشته باشد به‌صورتی که

$$Pr[Invert_{\mathcal{A}, \Pi}(n) = 1] \leq negl(n).$$

با داشتن $GenRSA$ به شکل بیان‌شده در بخش ۴-۲-۸، ساختار ۷۷-۸ یک خانواده از جایگشت‌ها را تعریف می‌کند. فوراً مشخص است که اگر مسئله‌ی RSA نسبت به $GenRSA$ سخت باشد، آنگاه این خانواده، یک‌طرفه خواهد بود. به شکل مشابه می‌توان نشان داد که سختی مسئله‌ی لگاریتم گسسته در $\mathbb{Z}_p^*$ با p اول، نشان‌دهنده‌ی وجود یک خانواده‌ی یک‌طرفه از جایگشت‌ها است؛ ن.ک. بخش ۷-۲-۱.

۴-۲-۸ ساختن توابع چکیده ساز مقاوم در برابر برخورد

توابع چکیده ساز مقاوم در برابر برخورد در بخش ۵-۱ معرفی شدند. هرچند ساختارهای توابع چکیده ساز مقاوم در برخورد مورد استفاده در عمل را در بخش ۶-۳ مورد بحث قرار دادیم، هنوز ساختارهایی ندیده‌ایم که بتوانند به شکل استوار بر اساس فرضیات ساده‌تر باشند. در اینجا یک ساختار مبتنی بر فرض لگاریتم گسسته در گروه‌های مرتبه اول را نشان می‌دهیم. (یک ساختار مبتنی بر مسئله‌ی RSA در تمرین ۸-۲۰ توصیف شده است.) هرچند این ساختارها نسبت به توابع چکیده ساز مورد استفاده در عمل، کارایی کمتری دارند، ولی مهم هستند به این دلیل که قابلیت عملی رسیدن به مقاومت در برابر برخورد بر اساس فرضیات نظریه اعداد استاندارد و مورد مطالعه‌ی کافی را نشان می‌دهند.

فرض کنید $\mathcal{G}$ یک الگوریتم زمان-چندجمله‌ای باشد که با دریافت ورودی 1^n ، یک (توصیف از) گروه دوری G ، مرتبه‌ی آن q (با $\|q\| = n$) و یک مولد g را می‌دهد. در اینجا همچنین ملزم می‌کنیم که q عدد اول باشد مگر با احتمال ناچیز. یک تابع چکیده ساز طول ثابت بر اساس $\mathcal{G}$ در ساختار ۸-۷۸ ارائه شده است.

یک تابع چکیده ساز با طول ثابت

ساختار ۷۸-۸

فرض کنید $\mathcal{G}$ به‌صورت توصیف‌شده در متن باشد. یک تابع چکیده ساز طول ثابت (Gen, H) را به‌صورت زیر تعریف کنید:

- Gen : با دریافت ورودی 1^n ، $\mathcal{G}(1^n)$ را اجرا کنید تا (G, q, g) به دست آید و سپس یک $h \in G$ یکنواخت را انتخاب کنید. خروجی $s = \langle G, q, g, h \rangle$ را به‌عنوان کلید، ارائه دهید.
- H : با داشتن یک کلید $s = \langle G, q, g, h \rangle$ و ورودی $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$ ، خروجی $H^s(x_1, x_2) = g^{x_1} h^{x_2} \in G$ را ارائه کنید.

توجه کنید که Gen و H را می‌توان در زمان چندجمله‌ای محاسبه نمود. قبل از ادامه دادن به تحلیل این ساختار، چند نکته‌ی فنی را بیان می‌کنیم:

(ب) در غیر این صورت $(1 \neq h, x)$ را به صورت (x_1, x_1') تجزیه کنید و x' را به صورت (x_1, x_1') تجزیه کنید که در آن، $x_1, x_1', x_1, x_1' \in \mathbb{Z}_q$ و نتیجه $(x_1 - x_1')^{-1} \bmod q$ را باز گردانید.

منحی است که اگر در زمان چندجمله‌ای اجرا می‌شود به علاوه، ورودی s داده شده به h هنگام تجزیه به عنوان یک زیررول توسط h دارای توری دقیقاً برابر با توزیع در $\text{Hash} - \text{Hash}$ coll_h برای یک مقدار مفروض برای پارامتر نسبتی n است. (نمودی برای h با اجرای $g^{(n)}$ برای به دست آوردن q, q, q و سپس انتخاب $h \in G$ به شکل یکپارخت و تصادفی تولید می‌شود. این دقیقاً همان شیوهی تولید s توسط $\text{gen}^{(n)}$ است) بنابراین، با احتمال دقیقاً $e(n)$ یک برخورد وجود دارد یعنی، $x' \neq x$ و $H^s(x) = H^s(x')$.

اما می‌کنیم که هر بار که یک برخورد وجود دارد، h پاسخ صحیح h را ارائه می‌دهد اگر h انگاه مشخصی است که این امر صحت می‌کند (از اینجا که در این حالت، $h = \log h$ است) با فرض اینکه $1 \neq h$ وجود یک برخورد به این معنا است که

$$\begin{aligned} H^s(x_1, x_1') &= H^s(x_1', x_1) \Rightarrow g^{x_1} h^{x_1'} = g^{x_1'} h^{x_1} \\ &\Rightarrow g^{x_1 - x_1'} = h^{x_1 - x_1'} \end{aligned} \quad (A.5)$$

توجه کنید که $q \bmod q = 0$ در $x_1 - x_1' \neq 0$ در این صورت، خواهیم داشت $q \bmod q = x_1$ ولی در این حالت، $x = x'$ و برخورد نخواهیم داشت. از اینجا که q اول است، و از آن $\Delta \equiv (x_1' - x_1) \bmod q^{-1}$ وجود دارد به توان رساندن دو طرف معادله (A.5) به این توان، می‌دهد:

$$g^{(x_1 - x_1') \Delta} = (h^{x_1 - x_1'})^\Delta = h^1 = h.$$

و بنابراین، خروجی ارائه شده توسط h عبارت است از

$$\log_g h = [(x_1 - x_1') \cdot \Delta \bmod q] = [(x_1 - x_1') \cdot (x_1' - x_1)^{-1} \bmod q].$$

مانده می‌کنیم که h به قدرستی مسئلهی لگاریتم گسسته را با احتمال دقیقاً $e(n)$ حل می‌کند (از اینجا که بر اساس فرض، مسئلهی لگاریتم گسسته نسبت به q سخت است، نتیجه می‌گیریم که $e(n)$ ناچیز است. ■

با استفاده از تئوریم ۸-۲۱ به همراه تبدیل موکل-جمنگارد (ن. ک. بخش ۵-۴۲، به دست می‌آوریم:

قضیه ۸-۸

اگر مسئلهی لگاریتم گسسته سخت باشد، انگاه تولید یکجمله ساز مقاوم در برابر برخورد وجود دارند.

- برای یک (g, q, h) مفروض با $q \parallel n$ تابع H^s به صورتی تعریف می‌شود که عناصر $g, g \in \mathbb{Z}_q$ را به عنوان ورودی می‌پذیرد. باین حال، H^s را می‌توان دریافت‌کنندهی رشته‌ای بتنی با طول $(n-1)$ به عنوان ورودی در صورتی در نظر گرفت که یک ورودی رشته‌ای بتنی $x \in \{0, 1\}^{(n-1)}$ را به شکل دو رشته‌ای x_1, x_2 هر کدام با طول $n-1$ تجزیه کنیم و سپس x_1, x_2 را به عنوان عناصر g_1 به شکل طبیعی، در نظر بگیریم.
- به شکل مشابه، خروجی H^s به عنوان عنصری از G مشخص می‌شود، ولی می‌توانیم آن را به عنوان یک رشتهی بتنی در نظر بگیریم. اگر یک نمایش مفروض از G را در نظر بگیریم برای برقراری الزامات تعریف ۲-۴ (که بیان می‌کند طول خروجی باید به عنوان بتنی از n مشخص شود) می‌توانیم خروجی را در صورت نیاز، لایه‌گذاری کنیم.

- با داشتن موارد بالا، این ساختار تنها ورودی خود را برای گروه‌های G فشرده‌سازی می‌کند که در آن‌ها عناصر G را می‌توان با تعداد کمتر از $2n - 2$ بیت، نمایش داد. این امر هم برای گروه‌های ارائه شده توسط لگاریتم ۸-۴۵ (یا فرض $E \ll n$ ، که معمولاً برقرار است) و همین‌طور برای گروه‌های خم بنحوی هنگام استفاده از فشرده‌سازی نقطه، برقرار است. یک تمهید از ساختار ۸-۷۸ را می‌توان برای به دست آوردن فشرده‌سازی از هر G مورد استفاده قرار داد که برای آن، مسئلهی لگاریتم گسسته، سخت است بدون توجه به تعداد بیت‌های مورد نیاز برای نمایش اشیای گروه ن. ک. تئوریم ۸-۲۱.

۷۹-۸. قضیه‌ی

اگر مسئلهی لگاریتم گسسته نسبت به q سخت باشد، انگاه ساختار ۸-۷۸ یک تابع یکجمله ساز مقاوم در برابر برخورد با طول ثابت خواهد بود (موضوع به بحث مربوط به فشرده‌سازی در ۸-۹۱).

اثبات

فرض کنید $\Pi = (gen, H)$ به شکل مشخص شده در ساختار ۸-۷۸ باشد و فرض کنید h یک الگوریتم زمان چندجمله‌ای تصادفی، با

$$e(n) = \Pr[\text{Hash} - \text{coll}_h(n) = 1]$$

باشد (ن. ک. تعریف ۵-۴۲). نشان می‌دهیم که چگونه h می‌تواند توسط یک الگوریتم h' برای حل مسئلهی لگاریتم گسسته با احتمال موقعیت $e(n)$ مورد استفاده قرار گیرد:

الگوریتم h'

این الگوریتم، به عنوان ورودی، g, q, h را دریافت می‌کند.

۱) فرض کنید (g, h) $s = (g, q, h)$ را اجرا کنید و خروجی x و x' را به دست آورید.

۲) اگر $x' \neq x$ و $H^s(x) = H^s(x')$ باشد، انگاه:

۳) (الف) اگر $1 \neq h$ باشد، 0 را بازگردانید.

تمرین‌ها

- ۱-۸ فرض کنید G یک گروه اَبلی باشد. اثبات کنید که یک همانی منحصر به فرد در G وجود دارد و اینکه هر عنصر $e \in G$ دارای یک وارزن منحصر به فرد است.
- ۲-۸ نشان دهید زمانی که G نامتناهی است، گزاردی $2^{\aleph_0}$ ضرورتاً یزر قرار نیست.

راهنمایی: مجموعه‌ای $\mathbb{R} \subset \{2^{\aleph_0}, 2^{\aleph_1}, \dots\}$ را در نظر بگیرید.
- ۳-۸ فرض کنید G یک گروه متناهی باشد و $e \in G$ نشان دهد که $\langle e \rangle$ یک زیر مجموعه از G است. [ا همگانی که G نامتناهی باشد، مجموعه‌ای $\{e, e^2, \dots\}$ ضرورتاً یک زیر گروه از G است.
- ۴-۸ این سؤال مربوط به تابع فی اویلر است.

(الف) فرض کنید p عددی اول و $1 \leq e$ عددی صحیح باشد. نشان دهید که $\phi(p^2) = p^2 - p$ است.

(ب) فرض کنید p, q نسبت به هم اول باشند. نشان دهید که $\phi(pq) = \phi(p) \cdot \phi(q)$ می‌باشد.

(ج) قضیه ۱۹-۸ را اثبات کنید.
- ۵-۸ دو رقم (اعشاری) آخر 3^{1000} را (به صورت دستی) محاسبه کنید.

راهنمایی: پاسخ برابر است با $10^{1000} \bmod 100$.
- ۶-۸ $2^{\aleph_0} \bmod 10$ را (به صورت دستی) با استفاده از قضیه‌ای، اقلیمادی چینی، حساب کنید.
- ۸-۸ اثبات کنید که اگر H, H_1 گروه باشند، آنگاه $H \times H_1$ یک گروه است.
- ۹-۸ فرض کنید N, p اعداد صحیح با $p \nmid N$ باشند. اثبات کنید که برای هر عدد صحیح x ، $[x \bmod N] \bmod p = [x \bmod p]$ باشد.
- ۱۰-۸ نتایجی فوری ۲۱-۸ نشان می‌دهد که اگر $N = pq$ و $N \nmid \phi(N)$ ، آنگاه برای همه $x \in \mathbb{Z}_N^*$ داریم $x \bmod N = x^{\phi(N)} \bmod N$. نشان دهید که این امر برای همه $y \in \{1, \dots, N-1\}$ برقرار است.

مراجع و مطالعه بیشتر

کتاب نوشته شده توسط چاپلر [۴۴] به شکلی عالی به موضوع نظریه‌ی گروه‌ی مودیخت در این فصل (و بیشتر) به شکلی عمیق‌تر ولی با سطح تفسیر مشابه، می‌پردازد. شوب [۱۵۹] یک بررسی بیشتر و فزاینده‌ی حال قابل درک را برای بخش اعظم مطالب با تمرکز بر ایجاد الگوریتمی ارائه می‌کند (تجربی بیان اصل برترانه توسط ما از [۱۵۹]، قضیه‌ی ۸-۵ گرفته شده است). مقدمات نسبتاً ملایم برای جبر مجرد و نظریه‌ی گروهی که ویرالتر از فضای موجود در این کتاب است در کتابهای پیشنهادی نوشته شده توسط فولی [۴۶] و هورستاین [۸۹] وجود دارد. خوانندگی علاقه‌مند به سادگی می‌تواند متون جبری پیشرفته‌تر را در صورت تمایل ببیند.

اولین آزمون کار برای سختی اول بودن توسط سولوی و استراسن ارائه شد [۱۶۴]. آزمون سیر-رلین از سیر [۱۲۷] و رلین [۱۶۴] گرفته شده است. یک آزمون اول بودن قطعی توسط آگراوال و همکاران [۵۹] یافت شد. برای بررسی جامع این حوزه به دیتابیس‌نگر [۵۷] مراجعه کنید.

مسئله‌ی RSA توسط ریست، شامیر و آدلش [۱۲۸] معرفی شد. هر چند در سال ۱۹۹۷ مشخص شد که الیس، کاسی و ویلیامسون، سه عضو $CCHQ$ (یک سازمان اطلاعاتی بریتانیایی)، ایده‌ی مشابه را - بدون اینکه به‌طور کامل متوجه اهمیت آن‌ها شوند - در یک وضعیت محرمانه چندین سال قبل از بررسی کرده‌اند. مسائل نظریه‌ی گسترده و دینی حل‌شدن در اینجا حداقل به صورت ضمنی توسط دینی و هلمن [۵۸] مورد بررسی قرار گرفتند.

اغلب بررسی‌های مهم بغیر از این‌ها به پیش‌زمینه‌ی ریاضیاتی پیشرفته‌ی خواننده است. کتب نگارش شده توسط سیرلین و تیت [۱۶۰] شاید یک استثنا بر این امر باشد. با این حال، به مانند بسیاری از کتابهای بیرون این موضوع که برای دانشمندان نوشته شده‌اند، این کتاب تنها اندکی به جهای پیشروی روی میدان‌های متناهی می‌پردازد. در مقاله‌ای است که بیشترین اهمیت را برای رونمایی دارد. متن نوشته شده توسط واشنگتن [۱۷۶]، هر چند اندکی پیشرفته‌تر است، به صورت گسترده روی نه به صورت اختصاری به حالت میان متناهی می‌پردازد. مسائل پیاده‌سازی مرتبط با رونمایی هم پیشروی توسط هاکوسون و همکاران [۱۲۱] بررسی شده‌اند. پارامترهای پیشنهادی برای جهای پیشروی و همین‌طور زیر گروه‌های به پیمانه‌ی یک عدد اول توسط $MIST$ [۱۲۲] ارائه شده‌اند.

ساختار یک تابع چکیده ساز مقاوم در برابر برخورد بر اساس مسئله‌ی نگاریم گسترده از [۴۲] گرفته شده است و یک ساختار قطعی بر اساس سختی تجزیه در [۸۱] ارائه شده است (همچنین ن.ک. تمرین ۲۰-۸).

۱۶-۸: نقاط روی خم بیضی $1 + x^2 + y^2 = 0$ روی $\mathbb{Z}_{11}$ را تعیین کنید. چه تعداد نقطه بر روی این منحنی وجود دارد؟

۱۷-۸: گروه خم بیضی مثال ۶-۸ را در نظر بگیرید. (همچنین ن. ک. مثال ۱-۸، $(0, 1)$ و $(1, 0)$ را در این گروه به این صورت حساب کنید که ابتدا به مختصات تصویر تبدیل کرده و سپس از معادلات $(8, 1)$ و $(8, 1)$ استفاده کنید.

۱۸-۸: جمله‌ی چهارم در گزاردی $8-8$ را اثبات کنید.

۱۹-۸: آیا مسئله زیر را می‌توان در زمان چندجمله‌ای حل کرده؟ با دلالتن یک عدد اول p ، یک مثال $x \in \mathbb{F}_p$ و $y \in \mathbb{F}_p$ که $y = [x^2 \bmod p]$ ده در آن، h یک مقدار یکنواخت در $\mathbb{F}_p$ است، h را بیابید. یعنی $y \bmod p$ را محاسبه کنید. اگر پاسخ شما «آری» است، یک الگوریتم زمان چندجمله‌ای ارائه کنید. اگر پاسخ شما «خیر» است، یک کاهش به یکی از فرضیات معرفی‌شده در این فصل نشان دهید.

۲۰-۸: فرض کنید GenRSA به‌صورت تعریف‌شده در بخش ۲-۸ باشد. اثبات کنید که اگر مسئله RSA نسبت به GenRSA سخت باشد، انگانه ساختار $8-8$ یک تابع چکیده ساز مقاوم در برابر برخورد با طول ثابت است.

ساختار ۸-۸

را به‌صورت زیر تعریف کنید:

- Gen : با دریافت ورودی 1^n ، GenRSA(1^n) را اجرا کنید تا N, e, d به دست آید و y را انتخاب کنید. کلید برابر است با (N, e, y) .
- Dec : اگر $s = (N, e, y)$ باشد، انگانه H^e ورودی‌ها در $\{0, 1\}^n$ را به خروجی‌ها در $\mathbb{Z}_N$ نگاشت می‌کند. فرض کنید $[x^e \bmod N]$ و $f^e(x) = [y \cdot x^e \bmod N]$ تعریف کنید. برای یک رشته‌ی با طول $2n$ یعنی $x_1 \dots x_{2n}$ ، Dec تعریف کنید:

$$H^e(x) = f^e(y_1 \dots y_n)$$

۲۱-۸: تسمه ذیل برای ساختار $8-8$ را در نظر بگیرید:

رابطه‌های از قضیه‌ی بقایایندمی چینی استفاده کنید.

۱۱-۸: جزییات اثبات قضیه‌ی بقایایندمی چینی را کامل کنید. یعنی نشان دهید که $\mathbb{Z}_m \times \mathbb{Z}_n$ یکریخت است.

۱۲-۸: این تمرین یک الگوریتم کارا برای بررسی این مسئله ایجاد می‌کند که آیا یک عدد صحیح یک توان کامل است یا خیر.

(الف) نشان دهید که اگر برای تعدادی عدد صحیح $1 < e < N$ ، N مفروض، $N = e$ باشد، N انگانه e است.

(ب) با دلالتن N و e با $1 < e \leq N$ نشان دهید که چگونه می‌توان در زمان $\text{poly}(\log N)$ مشخص نمود که آیا یک عدد صحیح N یا $N = e$ وجود دارد یا خیر.

رابطه‌های از جستجوی دودویی استفاده کنید.

(ج) با دلالتن N نشان دهید که چگونه می‌توان در زمان $\text{poly}(\log N)$ این مسئله را ازبود که آیا N یک توان کامل هست یا خیر.

۱۳-۸: با دلالتن N و e نشان دهید که چگونه می‌توان در زمان چندجمله‌ای این مسئله را ازبود که آیا e یک شاهد قوی برای مرکب بودن N هست یا خیر.

۱۴-۸: N, e را با $1 < e < N$ و $\gcd(e, \phi(N)) = 1$ نشان دهید و فرض کنید که یک متخاصم m در حال اجرا در زمان e وجود دارد که برای آن:

$$\text{Pr}[m^e \bmod N] = x]$$

که در آن، احتمال روی انتخاب یکنواخت $x \in \mathbb{Z}_N$ گرفته شده است. نشان دهید که امکان ساختن یک متخاصم m وجود دارد که برای آن:

$$\text{Pr}[m^e \bmod N] = x]$$

برای تمام x ها زمان اجرای m^e برابر m^e باشد. $\|N\|$ باشد. $r = (y \cdot m)^{1/e} \bmod N$ را به‌دست آورید.

۱۵-۸: به‌صورت رسمی فرض CDH را تعریف کنید. اثبات کنید که سختی مسئله CDH نسبت به g نشان‌دهندگی سختی مسئله‌ی الگوریتم گسترده نسبت به g است و اینکه سختی مسئله DDH نسبت به g نشان‌دهندگی سختی مسئله CDH نسبت به g است.