باسمه تعالی

دانشگاه صنعتی شریف

دانشکده علوم ریاضی

درس مقدّمه‌ای بر رمزنگاری

پروژه‌ی پایانی درس: فرضیات و چالش‌های رمزنگاری

استاد درس: دکتر شهرام خزایی

امیرحسین افشارراد

۹۵۱۰۱۰۷۷

۲۷ مرداد ۱۳۹۹

فهرست مطالب

۱ مقدّمه ۲

۲ تعاریف و مفاهیم اوّلیه ۲

۱.۲ مفروضات ۲

۲.۲ چالش‌ها ۳

۳.۲ ارتباط مفروضات و چالش‌ها ۴

۳ کلاس‌های سختی مفروضات بر مبنای ابطال‌پذیری ۵

۱.۳ تعاریف ۵

۲.۳ مثال‌ها ۶

۴ برخی مسائل باز ۹

۵ جمع‌بندی ۱۱

ضمیمه‌ها ۱۳

الف نمونه‌هایی از مفروضات رمزنگاری ۱۳

الف.۱ فرض تجزیه ۱۳

الف.۲ فرض RSA ۱۳

الف.۳ فرض RSA تفاضلی ۱۳

الف.۴ فرض دانشِ نما ۱۴

ب کلاس‌های پیچیدگی مسائل ۱۴

ج اثبات ابطال‌پذیری کارای فرض تجزیه ۱۵

د توابع و جایگشت‌های یک‌طرفه ۱۷

ه اثبات ابطال‌پذیری فرض شبه‌تصادفی بودن یک مولّد ۱۷

و اثبات دانایی ۱۸

ز اثبات دانایی صفر ۱۹

ح خودکاهش‌پذیری تصادفی ۲۰

مراجع ۲۲

۱ مقدّمه

گزارشی که در اختیار دارید، شرح مختصری بر مقاله‌ی On Cryptographic Assumptions and Challenges نوشته‌ی Moni Noar (مرجع [۱]) می‌باشد. موضوع کلّی این مقاله، بررسی مفروضات رمزنگاری و سختی آن‌ها می‌باشد. امروزه بسیاری از سیستم‌ها و پروتکل‌های رمزنگاری مبتنی بر مفروضاتی عمل می‌کنند که هنوز اثباتی برای درستی آن‌ها ارائه نشده است و در عین حال، با توجّه به دانش امروزی، به صورت عملی می‌توان از آن‌ها استفاده نمود. هدف این مقاله آن است که یک طبقه‌بندی کلّی برای این مفروضات ارائه دهد و به بررسی این موضوع بپردازد که چنین مفروضاتی، اگر درست نباشند، تحت چه شرایطی قابل ابطال خواهند بود. همچنین این مقاله تلاش می‌کند بستری برای مقایسه‌ی قدرت و میزان سنگین‌بودن این مفروضات ایجاد کند تا بتوان در چارچوبی کامل‌تر به بررسی این موضوع پرداخت که امنیت پروتکل‌های رمزنگاری گوناگون، بر مبنای چه میزان از مفروضات تضمین‌شده هستند و آیا می‌توان تا حدّ امکان این مفروضات را سبک‌تر کرد تا امنیت این پروتکل‌ها در گرو فرض‌های کمتری باشند. این طبقه‌بندی به طور خاص مسأله‌ی *سختی ابطال‌پذیری مفروضات* را هدف قرار می‌دهد و این امکان را فراهم می‌کند تا سختی مفروضاتی که در ظاهر ارتباطی به یکدیگر ندارند نیز از این منظر قابل مقایسه باشند. علاوه بر آن، با توجّه به این درجات سختی، مسائل باز جدیدی نیز (ناظر به سختی ابطال‌پذیری این مفروضات) مطرح خواهند شد.

لازم است توجّه داشته باشید که این گزارش ترجمه‌ای از مقاله نیست، بلکه استنباط نگارنده از محتوای آن بوده و در برخی موارد که توضیحات مقاله ناکافی یا نادقیق بوده، نکات یا تعاریفی توسّط نگارنده افزوده شده است. همچنین ممکن است مواردی از مقاله که از اهمّیت چندانی برخوردار نبوده باشند نیز در این گزارش مورد اشاره قرار نگرفته باشند. توجّه کنید که با پایان قسمت اصلی این گزارش، بخشی تحت عنوان ضمیمه‌ها قرار دارد که در آن، برخی مفاهیم و تعاریفی که در مقاله به آن‌ها اشاره شده است معرّفی و تشریح شده‌اند (البته از تشریح مفاهیمی که در درس «مقدّمه‌ای بر رمزنگاری» بیان شده‌اند خودداری شده است و صرفاً مفاهیم جدید مورد بحث قرار گرفته‌اند). به علاوه، به برخی اثبات‌های مرتبط با گزاره‌های مقاله نیز در بخش ضمیمه‌ها اشاره شده است. نهایتاً واضح است که این قسمت از اصل گزارش جداست و بدون مطالعه‌ی آن، خللی در مطالعه‌ی اصل گزارش برای خواننده به وجود نخواهد آمد. همچنین در پایان این گزارش نیز جمع‌بندی نهایی نگارنده از محتوای مقاله ارائه شده است.

۲ تعاریف و مفاهیم اوّلیه

۱.۲ مفروضات

در ادبیات رمزنگاری، فرض‌ها[1] (فرضیات، مفروضات) متعدّدی وجود دارند که بسیاری از سیستم‌ها و پروتکل‌ها بر مبنای درستی آن‌ها عمل می‌کنند. این در حالی است که برقراری بسیاری از این مفروضات به صورت عمومی اثبات نشده است؛ و در عین حال در عمل مشاهده می‌شود که پروتکل‌هایی که بر مبنای درستی آن‌ها عمل می‌کنند موفّق هستند و بدون نقص امنیتی به نظر می‌رسند. در این جا یک تعریف رسمی از یک «فرض» و سختی آن ارائه می‌دهیم و در قسمت‌های بعدی این تعریف را پایه‌ای برای بیان سایر مفاهیم قرار خواهیم داد.

تعریف ۱. (فرض) *یک فرض گزاره‌ای است که از نظر منطقی درست یا غلط باشد، هرچند درستی یا نادرستی آن معلوم نباشد.*

به عنوان مثال، گزاره‌ی $\mathbf{P} = \mathbf{NP}$ (توضیحات در مورد کلاس‌های پیچیدگی در ضمیمه‌ی ب ارائه شده است) یک نمونه فرض است که درستی یا نادرستی آن در حال حاضر مشخّص نیست.

[1] assumptions

تعریف ۲. **(رد کردن فرض)** *منظور از رد کردن یک فرض، ارائه‌ی یک اثبات ریاضی معتبر برای غلط بودن آن فرض است.*

تعریف ۳. **(فرض پارامتردار)** *فرض پارامتردار A، یک چهارتایی مرتّب به صورت (A^*, n, t, ϵ) است که A^* یک فرض (مطابق تعریف ۱) است، n یک عدد است، و t و ϵ به صورت کلّی توابعی از n هستند. همچنین گزاره‌ی منطقی متناظر با A^*، خود می‌تواند تابعی از n باشد.*

تعریف ۴. **(درستی یک فرض پارامتردار)** *می‌گوییم فرض پارامتردار $A(A^*, n, t, \epsilon)$ درست (برقرار) است، هرگاه هیچ الگوریتمی مانند A (که بعضاً یک مهاجم[2] نامیده می‌شود) قادر نباشد در زمان t، با احتمالی بیشتر یا مساوی ϵ فرض A^* را (مطابق با تعریف ۲) رد کند.*

در ادامه‌ی این گزارش (و به تقلید از مرجع [۱])، فرض $A(A^*, n, t, \epsilon)$ را برای سادگی به صورت $A^*(n, t, \epsilon)$ نشان می‌دهیم و می‌گوییم A^* یک فرض پارامتردار با پارامترهای n و t و ϵ است (که به صورت دقیق‌تر و مطابق با تعاریف فوق، معادل با آن است که بگوییم A فرضی پارامتردار با پارامترهای A^* و n و t و ϵ است). همچنین برای سادگی، در ادامه به جای استفاده از عبارت «فرض پارامتردار»، ممکن است از همان لفظ «فرض» استفاده شود.s

به عنوان مثال، برای معرّفی یک فرض پارامتردار مثل A (که متشکّل از فرض ساده‌ی A^* و پارامترهای n و t و ϵ است)، می‌توان A^* را فرض «هیچ عدد n بیتی که حاصل‌ضرب دو عدد اوّل باشد قابل تجزیه به عوامل اوّل خود نیست» در نظر گرفت. در این حالت با در نظر گرفتن انتخاب‌های مختلف برای n و t و ϵ، ممکن است A (مطابق با تعریف ۴) درست یا نادرست باشد.

شایان ذکر است که با وجود آن که تحلیل‌های مجانبی[3] در مورد زمان‌های اجرای الگوریتم‌ها به مراتب رایج‌تر است؛ امّا به دلیل آن که [۱] به جای رویکرد مجانبی از رویکرد ذکر مستقیم پارامتر زمان t استفاده کرده، در این گزارش نیز بر همین مبنا عمل خواهد شد.

مفروضات رمزنگاری متعدّدی وجود دارند و همواره می‌توان مفروضات جدیدی نیز مطرح کرد. سختی یک فرض رمزنگاری تابعی از میزان ϵ و t در تعریف ۳ است. اگرچه مفروضاتی وجود دارند که در مورد برقراری یا عدم برقراری آن‌ها اثباتی در دست نیست، با این حال بسیاری از این مفروضات با فرض $\mathbf{P} \neq \mathbf{NP}$ «سخت» محسوب می‌شوند. توضیحات مختصری در مورد کلاس‌های پیچیدگی مسائل در ضمیمه‌ی ب ارائه شده است. همچنین سختی یک فرض به صورت عمومی را نیز می‌توان به صورت مجانبی و مطابق با تعریف ۵ به صورت رسمی بیان کرد:

تعریف ۵. **(سختی فرض با رویکرد مجانبی)** *فرض A را سخت گوئیم، هر گاه برای هر t که تابعی چندجمله‌ای از n و یک ϵ که تابعی ناچیز از n باشد، $A(n, t, \epsilon)$ برقرار باشد.*

۲.۲ چالش‌ها

یکی از مفاهیمی که در [۱] به کرّات مورد استفاده قرار گرفته و نقشی کلیدی بر عهده دارد، مفهوم *چالش*[4] می‌باشد. با توجّه این که مقاله تعریفی رسمی و دقیق از یک چالش ارائه نداده و صرفاً برخی ویژگی‌های آن را توصیف کرده، ما نیز به ارائه‌ی یک تعریف کیفی بسنده می‌کنیم.

[2] adversary
[3] asymptotic analysis
[4] challenge

تعریف ۶. (چالش) *یک چالش مسأله‌ای است که پاسخ دارد و هر پاسخ آن درست یا نادرست است؛ یعنی برای هر چالش مانند d لازم است حدّاقل یک تصدیق‌گر [۵] مانند V موجود باشد که برای هر پاسخ دلخواه مانند x، خروجی تصدیق‌گر یکی از دو مقدار «قبول» یا «رد» باشد.*

به عبارت دیگر، اگر فضای چالش‌ها را D و فضای پاسخ‌ها را X در نظر بگیریم، هر تصدیق‌گر این چالش تابعی به صورت $V : D \times X \to \{\text{accept}, \text{reject}\}$ خواهد بود.

۳.۲ ارتباط مفروضات و چالش‌ها

یکی از مهم‌ترین مسائلی که در [۱] مورد بررسی قرار گرفته است، ارتباطی است که بین مفروضات و چالش‌های رمزنگاری برقرار می‌شود. برای روشن‌تر شدن این امر، ابتدا گزاره ۱ را در نظر بگیرید.

گزاره ۱. *برای آن که بتوان برقراری یا عدم برقراری یک فرض را سنجید، آن فرض باید **ابطال‌پذیر** [۶] باشد؛ یعنی باید روشی موجود باشد که بتوان ثابت کرد آن فرض غلط است.*

گزاره ۱ بیان می‌کند که مستقل از آن که یک فرض درست باشد یا نباشد، زمانی می‌توان از برقراری یا عدم برقراری آن سخن به میان آورد که روشی برای اثبات غلط بودن آن موجود باشد. مثلاً فرض کنید گزاره‌ی «همه‌ی اعداد طبیعی مضرب ۲ هستند» ابطال‌پذیر است؛ چرا که روشی برای ابطال آن وجود دارد و آن روش این است که یک عدد طبیعی معرّفی کنیم که مضرب ۲ نباشد. مجدّداً تأکید می‌کنیم که ابطال‌پذیری لزوماً به معنی غلط‌بودن فرض نیست. به عنوان مثال، گزاره‌ی «همه‌ی اعداد اوّل بزرگ‌تر از ۲ فرد هستند» یک فرض صحیح و ابطال‌پذیر است؛ یعنی روش صریحی برای ابطال آن وجود دارد و آن روش این است که یک عدد اوّل زوج بزرگ‌تر از ۲ معرّفی کنیم؛ هرچند در حال حاضر می‌دانیم که چنین عددی یافت نمی‌شود و فرض مذکور درست است.

آن‌چه در مقاله‌ی مرجع مورد توجّه قرار گرفته، بررسی *ابطال‌پذیری مفروضات به کمک چالش‌ها* است. به عبارت دیگر، هدف آن است که چارچوبی ترسیم شود که در آن، برای هر فرض مانند A، بتوان مجموعه‌ای از چالش‌ها در نظر گرفت، به گونه‌ای که تناظری یک‌به‌یک بین غلط بودن فرض A و امکان حلّ چالش وجود داشته باشد. به صورت دقیق‌تر، می‌گوییم برای فرض A، مجموعه‌ای از چالش‌ها با یک توزیع تصادفی D_n وجود داشته باشد، به گونه‌ای که برای هر چالش d که به صورت تصادفی از D_n انتخاب شده باشد، حدّاقل یک پاسخ مثل y وجود داشته باشد که مورد قبول تصدیق‌گر V قرار بگیرد (یعنی $V(d,y) = \text{accept}$). در این حالت به دنبال تناظری یک‌به‌یک بین یک مهاجم $\mathcal{A}$ برای شکستن فرض A و یک *ابطال‌گر* [۷] $\mathcal{B}$ برای موفّقیت در چالش d هستیم که اگر $\mathcal{A}$ در زمان t و با احتمال ϵ موفّق به شکستن فرض A می‌شود، آنگاه $\mathcal{B}$ در زمان t' قادر به شکست چالش d باشد که t' تابعی چندجمله‌ای از t و $۱/\epsilon$ باشد.

در این جا یک پارامتر مهمّ دیگر به نام δ نیز به عنوان کران بالایی برای احتمال شکست چالش معرّفی می‌شود؛ بدان معنی که حدّاکثر با احتمال δ ممکن است فرض A غلط باشد و با این حال، چالش d قابل حل نباشد. (فراموش نکنید که به صورت کلّی، تناظر یک‌به‌یک مربوطه بدان معناست که چالش d قابل حل باشد، اگر و تنها اگر فرض A غلط باشد. به عنوان نمونه، در یکی از مثال‌های ساده‌ای که ذکر کردیم، اگر فرض مربوطه را گزاره‌ی «همه‌ی اعداد اوّل بزرگ‌تر از ۲ فرد هستند» معرّفی کرده و چالش d را چنین در نظر بگیریم که «عددی اوّل و بزرگ‌تر از ۲ که زوج باشد معرّفی کنید»؛ آنگاه شرط لازم و کافی برای غلط بودن فرض

[5] verifier
[6] falsifiable
[7] falsifier

آن است که چالش قابل حل باشد. به صورت کلّی‌تر، لازم و کافی بودن این شرط را با یک احتمال خطای δ سست[8] کرده‌ایم).

همچنین باید توجّه داشت که طرف دیگر این تناظر یک‌به‌یک نیز باید برقرار باشد؛ یعنی اگر چالش با احتمال γ قابل حل باشد، فرض A نیز با احتمالی حدّاقل برابر با $\mathsf{poly}(\gamma)$ قابل شکستن باشد.

با توجّه به توضیحات فوق، ایده‌ی کلّی [۱] برای برقراری یک تناظر یک‌به‌یک بین مفروضات و چالش‌ها ترسیم شده است. پیش از به پایان بردن این قسمت، به یک نکته‌ی دیگر نیز اشاره می‌کنیم و آن، تناظر مهاجم[9] و ابطال‌گر است. به صورت کلّی، نقش مهاجم برای فرض مشابه با نقش ابطال‌گر برای چالش است. با این حال دو تفاوت کلیدی قابل تصوّر است:

- به صورت کلّی مفروضات می‌توانند تعاملی[10] باشند؛ بنابراین مهاجم می‌تواند با چالشگر خود تعامل[11] داشته باشد. این در حالی است که طبق تعریف، چالش غیرتعاملی است و پس از مشخّص‌شدن صورت چالش، دیگر تعاملی وجود نخواهد داشت تا آن که ابطال‌گر پاسخ نهایی خود را ارائه کند.
- برای آن که یک فرض با پارامترهای (n, t, ϵ) مثل A غلط باشد، کافی است مهاجمی وجود داشته باشد که با احتمالی بزرگ‌تر یا مساوی ϵ (که خود می‌تواند مقدار کوچکی باشد) در شکستن فرض موفّق شود. از طرف دیگر، لازم است ابطال‌گر با احتمال نسبتاً بالایی (در صورت غلط‌بودن فرض A) در چالش پیروز شود. این احتمال حدّاقل برابر با $۱ - \delta$ است که می‌تواند به شکل قابل‌توجّهی بیشتر از ϵ باشد.

۳ کلاس‌های سختی مفروضات بر مبنای ابطال‌پذیری

۱.۳ تعاریف

با توجّه به مفاهیم مطرح‌شده در قسمت ۲، مهم‌ترین دستاورد [۱] ارائه‌ی یک طبقه‌بندی برای سختی مفروضات است، به گونه‌ای که سختی آن‌ها تنها وابسته به نحوه‌ی ابطال‌پذیری آن‌ها به کمک چالش‌ها باشد. به عبارتی، فرضی سنگین‌تر و قوی‌تر است که ابطال‌پذیری آن به کمک چالش سخت‌تر باشد.

در این طبقه‌بندی، سه کلاس سختی معرّفی شده‌اند که در ادامه تعاریف آن‌ها آورده شده است.

تعریف ۷. (***فرض ابطال‌پذیر کارا***[12]) *فرض A با پارامترهای (n, t, ϵ) را ابطال‌پذیر کارا گوییم، هرگاه توزیعی مانند D_n بر روی چالش‌ها و یک الگوریتم تصدیق $V : \{۰,۱\}^* \times \{۰,۱\}^* \to \{\mathsf{accept}, \mathsf{reject}\}$ موجود باشد، طوری که:*

۱. *نمونه‌برداری از D_n به صورت کارا قابل انجام باشد، یعنی بتوان این کار را در زمانی چندجمله‌ای بر حسب n و $\log ۱/\epsilon$ و $\log ۱/\delta$ صورت بپذیرد.*

۲. *الگوریتم V کارا باشد، یعنی زمان اجرای آن بر حسب n و $\log ۱/\epsilon$ و $\log ۱/\delta$ چندجمله‌ای باشد.*

۳. *در صورتی که فرض A برقرار نباشد، یک ابطال‌گر مانند $\mathcal{B}$ موجود باشد که برای یک $d \in D_n$ تصادفی، پاسخی مانند y را طوری بیابد که با احتمالی حدّاقل برابر با $۱ - \delta$ داشته باشیم $V(d, y) = \mathsf{accept}$. همچنین زمان اجرای $\mathcal{B}$ بر حسب*

[8] relaxed
[9] adversary
[10] interactive
[11] interaction
[12] efficiently falsifiable assumption

زمان اجرای $\mathcal{A}$، n، $۱/\epsilon$، و $\log ۱/\delta$ چندجمله‌ای باشد.

۴. *اگر ابطال‌گری مانند $\mathcal{B}$ موجود باشد که چالش تصادفی $d \in D_n$ را در زمان t و با احتمالی حدّاقل برابر با γ حل می‌کند، آنگاه مهاجمی مانند $\mathcal{A}$ با زمان اجرای t' و احتمال موفّقیتی حدّاقل برابر با ϵ برای شکستن فرض A موجود باشد، طوری که t' و ϵ توابعی چندجمله‌ای از t و γ باشند.*

تعریف ۸. (***فرض ابطال‌پذیر***[۱۳]) *فرض A با پارامترهای (n, t, ϵ) را ابطال‌پذیر گوییم، هرگاه همه‌ی شرایط تعریف ۷ برای آن برقرار باشد، با این تفاوت که زمان‌های نمونه‌برداری از D_n و اجرای V تابعی چندجمله‌ای از $۱/\epsilon$ باشند (و نه تابعی چندجمله‌ای از $\log ۱/\epsilon$).*

تعریف ۹. (***فرض تا حدّی ابطال‌پذیر***[۱۴]) *فرض A با پارامترهای (n, t, ϵ) را تا حدّی ابطال‌پذیر گوییم، هرگاه همه‌ی شرایط تعریف ۸ برای آن برقرار باشد، با این تفاوت که زمان اجرای V ممکن است تابعی چندجمله‌ای از زمان اجرای $\mathcal{B}$ باشد. به طور خاص این بدان معناست که ممکن است V مجبور باشد اجرای $\mathcal{B}$ را شبیه‌سازی کند.*

توجّه مهم: در صورت‌بندی اصلی تعاریفی که در متن [۱] ذکر شده است، در تعریف مفروضات ابطال‌پذیر کارا، ذکر شده است که زمان اجرای ابطال‌گر $\mathcal{B}$ تابعی چندجمله‌ای از $\log ۱/\epsilon$ باشد؛ در حالی که آنچه در تعریف ۷ ذکر کرده‌ایم آن است که این الگوریتم در زمان چندجمله‌ای بر حسب $۱/\epsilon$ اجرا شود. سه دلیل برای این تفاوت وجود دارد:

۱. خود مقاله [۱] این مفهوم را در دو مورد متفاوت ذکر کرده که در یکی، زمان را تابع $\log ۱/\epsilon$ و در دیگری تابع $۱/\epsilon$ ذکر کرده است.

۲. اثبات‌هایی که نگارنده توضیحات آن‌ها را به صورت تکمیلی بر مقاله اضافه کرده و در این گزارش مشاهده می‌شوند (نظیر ضمیمه‌ی ج)، با فرض مذکور قابل قبول خواهند بود.

۳. با توجّه به این که تنها تفاوتی که بین تعاریف مفروضات ابطال‌پذیر و مفروضات تا حدّی ابطال‌پذیر وجود دارد آن است که در مورد دوم، زمان اجرای الگوریتم تصدیق V می‌تواند تابعی چندجمله‌ای از زمان اجرای ابطال‌گر $\mathcal{B}$ باشد، به نظر نمی‌رسد ضرورتی داشته باشد که $\mathcal{B}$ خود را محدود به اجرا در زمان چندجمله‌ای بر حسب $\log ۱/\epsilon$ کرده باشد؛ چرا که مستقل از این موضوع، کلّ الگوریتم V در مفروضات ابطال‌پذیر و تا حدّی ابطال‌پذیر مجاز به تابعیت چندجمله‌ای از $۱/\epsilon$ بوده است.

۲.۳ مثال‌ها

در این قسمت، مثال‌هایی از مفروضات مختلف را در نظر گرفته و در قالب گزاره‌هایی به بیان درجه‌ی سختی آن‌ها را بر حسب ابطال‌پذیری خواهیم پرداخت.

گزاره ۲. *همه‌ی مفروضات خودکاهش‌پذیر تصادفی*[۱۵] *ابطال‌پذیر کارا هستند.*

مفهوم خودکاهش‌پذیری تصادفی در ضمیمه‌ی ح تشریح شده است. به صورت شهودی، مفروضات خودکاهش‌پذیر تصادفی دارای این ویژگی هستند که برای شکستن آن‌ها بر روی یک ورودی مشخّص، می‌توان از مسأله‌ی مشابه با یک ورودی تصادفی کمک گرفت. اگر فرض بر روی این ورودی(های) تصادفی حل شود، حلّ آن برای ورودی اصلی نیز به دست می‌آید. به صورت

[13] falsifiable assumption
[14] somewhat falsifiable assumption
[15] random self-reducible

شهودی اگر فرض A خودکاهش‌پذیر تصادفی باشد، آنگاه کافی است یک نمونه از آن مثل $d \in D_n$ به عنوان چالش داده شود. در این صورت با فرض غلط‌بودن A، کسر ϵ از ورودی‌های ممکن توسّط یک مهاجم $\mathcal{A}$ قابل حل خواهند بود؛ بنابراین کافی است الگوریتم $\mathcal{B}$ به طور متوسّط $\mathcal{A}$ را بر روی $1/\epsilon$ ورودی تصادفی اجرا کند و تا به یک پاسخ مطلوب برسد؛ سپس با استفاده از خاصیت خودکاهش‌پذیری تصادفی حلّ مسأله را برای ورودی اصلی به دست آورد. دقّت کنید که در این فرآیند تنها زمان اجرای $\mathcal{B}$ تابعی از $1/\epsilon$ بوده که این با تعریف ۷ سازگار است.

فرض‌های لگاریتم گسسته و دیفی–هلمن محاسباتی نمونه‌هایی از مفروضات خودکاهش‌پذیر تصادفی هستند که اثبات خودکاهش‌پذیری تصادفی آن‌ها در ضمیمه‌ی ح موجود است.

گزاره ۳. *مسأله‌ی تجزیه*[16] *(که تعریف آن در ضمیمه‌ی الف.۱ موجود است) ابطال‌پذیر کارا است.*

اثبات این گزاره به اندازه‌ی گزاره‌ی قبل سرراست نیست. استنباط و اقتباس نگارنده از اثبات ارائه‌شده برای آن در ضمیمه‌ی ج آورده شده است.

گزاره ۴. *فرض RSA (که تعریف آن در ضمیمه‌ی الف.۲ موجود است) ابطال‌پذیر است.*

به صورت شهودی، ماهیت مسأله‌ی RSA با مسأله‌ی تجزیه مشابه است و به نظر می‌رسد که فرآیند اثبات ارائه‌شده برای مسأله‌ی تجزیه (ضمیمه‌ی ج) برای آن نیز قابل اجرا باشد، امّا می‌توان مشاهده کرد که به نظر روش کارایی برای نمونه‌برداری از $\mathbb{Z}^{(2)}(n)$ (مطابق تعریف ۱) وجود ندارد و تکنیکی که برای حل این مشکل در اثبات موجود در ضمیمه‌ی ج به کار رفته بود دیگر در این مورد قابل اعمال نیست. همین موضوع باعث می‌شود نتوان با روش مشابه ثابت کرد که فرض RSA ابطال‌پذیر کاراست؛ امّا ابطال‌پذیری آن سرراست است.

گزاره ۵. *فرض RSA تفاضلی (که تعریف آن در ضمیمه‌ی الف.۳ موجود است) تا حدّی ابطال‌پذیر است.*

با توجّه به ساختار فرض RSA تفاضلی و ساختار تعاملی آن، به نظر تنها راه تصدیق آن، اجرای الگوریتم $\mathcal{B}$ توسّط تصدیق‌گر V است و مطابق با تعریف ۹، به وضوح در این حالت فرض مربوطه تا حدّی ابطال‌پذیر خواهد بود. البته لازم است دقّت شود که با فرض غلط بودن فرض A، لازم است فرآیند مربوطه به تعداد متوسّط $1/\epsilon$ بار انجام شود تا یک بار موفّقیت در چالش حاصل شود.

گزاره ۶. *مفروضاتی وجود دارند که حتّی تا حدّی ابطال‌پذیر نیز نیستند. مفروضاتی که در آن‌ها، مشخّص نمی‌شود مهاجم $\mathcal{A}$ برنده شده است یا نه در این گروه قرار دارند و نمونه‌ای از آن‌ها، فرض دانشِ نما*[17] *(که در ضمیمه‌ی الف.۴ توضیح داده شده است) می‌باشد.*

با بررسی فرض دانشِ نما مشاهده می‌شود که عملاً روش صریح و سرراستی برای بررسی برنده‌شدن یک مهاجم در شکست فرض وجود ندارد. در واقع زمانی غلط بودن این فرض ثابت می‌شود که نشان داده شود برای یک مهاجم $\mathcal{A}$، هیچ الگوریتم $\mathcal{A}'$ یافت نمی‌شود که خاصیت مشخصی را داشته باشد (برای جزئیات بیشتر در مورد فرض دانشِ نما، ضمیمه‌ی الف.۴ را مطالعه کنید). بررسی چنین موردی از نظر زمان محاسبه، حتّی فراتر از تعریف تا حدّی ابطال‌پذیری است (زیرا برای انجام فرآیند تصدیق، باید به نوعی تمام $\mathcal{A}'$های ممکن بررسی شوند) و بنابراین نمی‌توان این فرض را در هیچ کدام از دسته‌بندی‌های سه‌گانه‌ی ابطال‌پذیری قرار داد.

[16] factoring
[17] knowledge of exponent

گزاره ۷. *برای یک جایگشت $f : \{0,1\}^n \rightarrow \{0,1\}^n$، فرض «$f$ یک جایگشت یک‌طرفه[18] است»، یک فرض ابطال‌پذیر کارا است.*

در مورد تعریف توابع و جایگشت‌های یک‌طرفه در ضمیمه‌ی د توضیحاتی ارائه شده است. به صورت شهودی، اثبات گزاره‌ی فوق نیز مشابه با اثباتی است که برای گزاره‌ی ۳ در ضمیمه‌ی ج ارائه شده است.

گزاره ۸. *برای یک تابع $f : \{0,1\}^n \rightarrow \{0,1\}^n$، فرض «$f$ یک تابع یک‌طرفه[19] است»، یک فرض ابطال‌پذیر است.*

در مورد تعریف توابع و جایگشت‌های یک‌طرفه در ضمیمه‌ی د توضیحاتی ارائه شده است. به تفاوت این گزاره و گزاره‌ی قبلی دقّت کنید. با جمع‌بندی این دو گزاره نتیجه می‌شود که فرض یک‌طرفه بودن یک جایگشت ابطال‌پذیر کاراست، در حالی که فرض یک‌طرفه بودن یک تابع صرفاً ابطال‌پذیر است و در مورد کارایی ابطال‌پذیری آن نمی‌توان به سادگی سخن گفت. به صورت شهودی، علّت این تفاوت در مسأله‌ی نمونه‌برداری از بُرد تابع f است؛ چرا که بر خلاف یک جایگشت دلخواه، برای نمونه‌برداری از بُرد یک تابع دلخواه باید صراحتاً به برد آن دسترسی داشت یا مقادیر آن‌ها را با محاسبه‌ی مستقیم از طریق تابع f به دست آورد (در حالی که برای یک جایشگت، می‌دانیم بُرد آن دقیقاً برابر با $\{0,1\}^n$ است.)

گزاره ۹. *برای یک تابع $G : \{0,1\}^m \rightarrow \{0,1\}^n$، فرض «$G$ یک مولّد شبه‌تصادفی[20] است»، یک فرض ابطال‌پذیر است.*

ایده‌ای که توسّط [۱] برای اثبات ابطال‌پذیری فرض فوق ارائه می‌شود، به صورت شهودی شامل تولید تعدادی زوج مرتّب به صورت (x_i, y_i) است یکی از این دو مؤلّفه خروجی تابع G و دیگری یک عدد کاملاً تصادفی است. توضیحاتی در مورد اثبات این گزاره در ضمیمه‌ی ه ذکر شده است.

دقّت کنید که تفاوت بنیادین این فرض با برخی فرض‌های دیگر که پیش‌تر بررسی شده آن است که در حالت عادی، احتمال تشخیص درست خروجی مولّد شبه‌تصادفی در یک زوج (x_i, y_i) حدّاقل برابر با ۰٫۵ است و این احتمال صرفاً با ارائه‌ی یک انتخاب تصادفی قابل دستیابی است. به همین دلیل، لازم است که چالش مربوطه به شکلی تعریف شود که مزیت ابطال‌گر را نسبت به حالت حدس‌های تصادفی تعیین کند.

گزاره ۱۰. *مفروضاتی که مهاجم آن‌ها امکان عمل‌کردن به صورت تعاملی را داراست، در حالت کلّی تا حدّی ابطال‌پذیر هستند؛ مگر در برخی موارد خاص که از یک پیاده‌سازی[21] خاص استفاده شده باشد.*

با توجّه به غیرتعاملی بودن ذاتی چالش‌ها (بر خلاف فرض‌ها)، اصولاً مفروضاتی که تعاملی هستند (یعنی مهاجم می‌تواند به صورت تعاملی با اوراکل خود در تماس باشد و درخواست‌هایی ارائه دهد و پاسخ‌هایی دریافت کند)، به ناچار در گروه تا حدّی ابطال‌پذیر قرار می‌گیرند. علّت این امر، مطابق با صورت تعریف ۹، آن است که الگوریتم تصدیق‌گر V مجبور است برای بررسی درستی عملکرد، ابطال‌گر $\mathcal{B}$ و به تبع آن مهاجمی مثل $\mathcal{A}$ را صراحتاً اجرا کند. نمونه‌هایی از چنین مفروضاتی عبارتند از شبه‌تصادفی بودن توابع[22]، امنیت رمزهای قالبی[23]، امنیت کدهای اصالت‌سنجی پیام[24]، و امنیت طرح‌های امضا[25].

البته همان طور که در صورت گزاره‌ی ۱۰ بیان شد، پیاده‌سازی‌هایی از چنین ساختارهایی وجود دارد که ابطال‌پذیر یا حتّی

[18] one-way permutation
[19] one-way function
[20] pseudo-random generator
[21] construction
[22] pseudo-randomness of functions
[23] block ciphers
[24] message authentication codes
[25] signature schemes

ابطال‌پذیر کارا باشند. نمونه‌ی آن، مولّد شبه‌تصادفی ارائه‌شده در [۹] است. با توجّه به این که این موضوع تنها ناظر به بررسی یک پیاده‌سازی جدید است، به بررسی بیشتر آن نمی‌پردازیم و خواننده‌ی علاقه‌مند را به [۹] ارجاع می‌دهیم.

گزاره ۱۱. *به صورت کلّی فرض‌های امنیت سیستم‌های رایج رمزنگاری تا حدّی ابطال‌پذیر هستند؛ امّا لزومی ندارد که در کلاس ضعیف‌تری (نظیر ابطال‌پذیر یا ابطال‌پذیر کارا) قرار داشته باشند، مگر در موارد خاص.*

گزاره‌ی اخیر به صورت عمومی در مورد امنیت سیستم‌های رمزنگاری بیان شده است که خود می‌تواند بسیار متنوّع باشد (اعم از کلید عمومی یا خصوصی بودن سیستم رمز و نوع امنیت، نظیر تک‌پیامی، چندپیامی، متن اصلی انتخابی یا متن رمزی انتخابی). علّت تا حدّی ابطال‌پذیری تمامی این مفروضات امنیتی آن است که می‌توان برای تصدیق موفّقیت ابطال‌گر، آن را (و به تبع آن، یک مهاجم برای فرض را) اجرا کرد. با این حال لزومی ندارد که این سیستم‌ها در گروه ضعیف‌تری نظیر ابطال‌پذیر یا ابطال‌پذیر کارا قرار بگیرند.

گزاره ۱۲. *فرض دانایی صفر*[26] *بودن یک سیستم اثبات دانایی*[27] *حتّی تا حدّی ابطال‌پذیر نیز نیست.*

در ضمیمه‌های و و ز، مفاهیم اثبات دانایی و اثبات دانایی صفر مورد توضیح و تشریح قرار گرفته‌اند. به صورت کلّی برای رد کردن فرض دانایی صفر بودن یک اثبات، لازم است نشان داده شود که هیچ شبیه‌ساز[28] مناسبی مانند S وجود ندارد که شرایط اثبات دانایی صفر را برآورده کند. چنین چیزی (مشابه با آن‌چه در مورد گزاره ۶ بیان شد) به صورت کلّی از نظر زمانی پیچیده‌تر از آن است که حتّی چالش مربوطه را در گروه تا حدّی ابطال‌پذیر قرار دهد (زیرا عملاً برای انجام فرآیند تصدیق، لازم است تمامی شبیه‌سازهای ممکن بررسی شوند).

البته می‌توان نشان داد که وجود خاصیت تمایزناپذیری شاهد[29] برای یک اثبات دانایی صفر (مطابق با تعریف ۲۳ در ضمیمه‌ی ز) تا حدّی ابطال‌پذیر است.

۴ برخی مسائل باز

ادبیات رمزنگاری همواره شامل مسائل بازی است که مرتبط با مفروضات رمزنگاری هستند. یک کلاس کلّی از این مسائل آن است که «آیا می‌توان امنیت پروتکل X را بر مبنای فرض سبک‌تری نسبت به فرض فعلی اثبات کرد؟». تعاریفی که در بخش ۱.۳ ارائه شدند، این امکان را فراهم می‌آورند که پرسش‌های باز جالب و متعدّدی در این حوزه مطرح شوند. به صورت کلّی، اگر یک فرض در یکی از کلاس‌های سختی معرّفی‌شده در بخش ۱.۳ قرار داشته باشد، تضمینی وجود ندارد که در کلاس مربوط به یک فرض سبک‌تر نباشد. به عنوان مثال، اگر ثابت شده باشد که فرضی ابطال‌پذیر است، همواره ممکن است شخصی ادّعا کند که این فرض ابطال‌پذیر کاراست. این موضوع به عنوان یک مسأله‌ی باز برای بسیاری از فرض‌ها قابل بررسی است؛ هر چند که در بسیاری از موارد، شهود محقّقین این حوزه بر آن است که ممکن است ابطال‌پذیری کارا برای یک فرض اصلاً قابل اثبات نباشد.

علاوه بر چگونگی ابطال‌پذیری مفروضات، یکی دیگر از پایه‌های مهم در درستی فرض‌ها استفاده از اوراکل‌های تصادفی[30] است. در واقع بسیاری از مفروضات و اثبات‌های موجود در ادبیات رمزنگاری با این فرض برقرارند که یک ساختار در سیستم به صورت اوراکل تصادفی عمل می‌کند؛ و این در حالی است که در عمل چنین چیزی میسّر نیست. در نتیجه اگر به صورت عملی

[26] zero-knowledge
[27] proof of knowledge
[28] simulator
[29] witness indistinguishability
[30] random oracles

بخواهیم فرضی برای امنیت سیستم در نظر بگیریم، آن چه در عمل اتّفاق می‌افتد ممکن است آن باشد که مجموعه‌ی مفروضات لازم برای امنیت یک سیستم، به همان اندازه سنگین باشند که فرض «این سیستم امن است» از ابتدا به عنوان فرض اساسی در نظر گرفته می‌شد. چنین چیزی نامطلوب است و عملاً ارزش گزاره‌ای که با چنین مفروضاتی اثبات شده است را پایین می‌آورد.

در ادامه نمونه‌هایی از مسائل باز موجود در این حوزه را به اختصار بیان می‌کنیم:

۱. رمزهای قالبی کارا

رمزهای قالبی کارای مورد استفاده در دنیای امروز نظیر DES و AES از نظر امنیت پشتوانه‌ی تئوری کاملی ندارند. در واقع فرضی که بر مبنای آن بتوان از امنیت چنین سیستم‌هایی سخن به میان آورد، همان فرض «این سیستم امن است» می‌باشد که در عمل نامطلوب است. از طرفی، سیستم‌هایی که از نظر تئوری پشتوانه‌های مناسبی دارند در عمل کارایی مطلوب را ندارند. یک مسأله‌ی باز ارائه‌ی رمزهای قالبی با پشتوانه‌های تئوری و مفروضات تا حدّ امکان سبک (یا حدّاقل سبک‌تر از فرض «این سیستم امن است») و همچنین با کارایی و بازدهی عملیاتی مناسب است.

۲. طرح‌های امضا

بیشتر طرح‌های امضای کارا و مفید از فرضی مبتنی بر اوراکل‌های تصادفی استفاده می‌کنند. البته طرح‌هایی که مبتنی بر مفروضات ابطال‌پذیر و ابطال‌پذیر کارا عمل کنند نیز ارائه شده‌اند، امّا از نظر کارایی عملیاتی چندان مناسب نیستند. همچنین مواردی نظیر امضاهای کوتاه[۳۱]، امضاهای با سربار کم[۳۲] و امضاهای مبتنی بر فرض لگاریتم گسسته تنها با فرض دسترسی به اوراکل‌های تصادفی جعل‌ناپذیر هستند و ارائه‌ی امضاهای امن جدید مبتنی بر مفروضات ابطال‌پذیر کارا به عنوان یک مسأله‌ی باز قابل بررسی است.

۳. رمزنگاری مبتنی بر هویت[۳۳]

رمزنگاری مبتنی بر هویت، ساختاری است که در آن به صورت شهودی، یک کلید عمومی کوتاه کلیدهای عمومی همه‌ی کاربران را معیّن می‌کند (معرّفی‌شده در [۱۰]). طرّاحی چنین سیستمی، به نحوی که تنها بر یک فرض ابطال‌پذیر کارا اتّکا کند یک مسأله‌ی باز است.

۴. سیستم‌های امضای کور[۳۴] و حفظ ناشناسی[۳۵]

به صورت شهودی یک سیستم امضای کور، یک سیستم امضا است با این قید که امضا کننده از متنی که آن را امضا می‌کند مطّلع نباشد و در فرآیند امضا اطّلاعی از آن به دست نیاورد. طرّاحی چنین سیستم امضایی بر مبنای مفروضات ابطال‌پذیر کارا یک مسأله‌ی باز است.

۵. اثبات دانایی صفر سه‌دوری[۳۶]

ارائه‌ی یک طرح اثبات دانایی صفر متشکّل از سه دور (سه بار تعامل در فرآیند اثبات) و مبتنی بر مفروضات ابطال‌پذیر کارا، یک مسأله‌ی باز است. (توضیحات مربوط به مفهوم اثبات دانایی صفر در ضمیمه‌ی ز بیان شده است.)

۶. توابع غیرقابل فشرده‌سازی[۳۷]

تعریف ۱۰. ***(تابع غیرقابل فشرده‌سازی)*** *تابع $f:\{0,1\}^n \rightarrow \{0,1\}^m$ که $n > m$ را غیرقابل فشرده‌سازی گویند، هر گاه هر پیام $o(m)$بیتی که امکان محاسبه‌ی $f(x)$ را فراهم کند، باعث آشکارشدن مقدار x نیز بشود.*

[31] short signature
[32] low-overhead signature
[33] identity-based encryption
[34] blind signature
[35] anonymity
[36] three-round zero knowledge proof
[37] incompressible functions

ارائه‌ی تابعی که فرض غیرقابل فشرده‌سازی‌بودن آن ابطال‌پذیر کارا باشد یک مسأله‌ی باز است.

نمونه‌های متعدّد دیگری از چنین مسائل بازی قابل ارائه‌کردن می‌باشند. خواننده‌ی علاقه‌مند می‌تواند چنین مواردی را حتّی شخصاً ابداع کند. علاوه بر چنین مسائل بازی، مسأله‌های کلّی‌تری نیز در ادبیات مطرح‌شده در این مقاله قابل بیان هستند. به عنوان مثال، در برخی حوزه‌های این پرسش مطرح است که آیا اصلاً امکان بیان مفروضاتی در ادبیات ابطال‌پذیری میسّر است یا نه. به عنوان مثال، با توجّه به این که در بسیاری از موارد، انسان‌ها ضعیف‌ترین حلقه‌ی امنیتی در یک سیستم هستند (و اگر چه طرّاحی سیستم امن است، امّا نشت اطّلاعات از طریق کاربرهای انسانی صورت می‌گیرد)، این که چگونه می‌توان این پارامتر را نیز در امنیت سیستم گنجاند و به صورت مفروضاتی ابطال‌پذیر از آن‌ها سخن به میان آورد، یک مسأله‌ی باز جالب و متفاوت در این حوزه است.

۵ جمع‌بندی

ابتدا به بیان نقائص طبقه‌بندی ارائه‌شده توسّط [۱] از دیدگاه خود مقاله می‌پردازیم. این مقاله ۴ عنوان نقص را برای طبقه‌بندی خود ذکر کرده است:

۱. ابطال‌پذیری به ارث برده نمی‌شود. در صورتی که فرض A ابطال‌پذیر (کارا) باشد و A فرض B را نتیجه دهد، لزومی ندارد B نیز ابطال‌پذیر (کارا) باشد. علّت این امر آن است که چالش متناظر با فرض B می‌تواند کاملاً متفاوت با چالش نظیر A باشد و ابطال‌گر مربوط به A لزوماً کارکردی برای B نخواهد داشت.

۲. طبقه‌بندی بر مبنای ابطال‌پذیری نمی‌تواند ترتیب قدرتمندی مفروضات را به طور کامل حفظ کند. در واقع ممکن است فرض A فرض B را نتیجه دهد (و این یعنی فرض A قوی‌تر است)، امّا قوی‌تر بودن A ممکن است در طبقه‌بندی مبتنی بر ابطال‌پذیری دیده نشود و هر دو فرض A و B در یک گروه (مثلاً ابطال‌پذیر کارا) قرار گرفته باشند. به این ترتیب این طبقه‌بندی به وضوح قادر نخواهد بود همه‌ی تفاوت‌ها و جزئیات مفروضات را منعکس کند.

۳. تاریخچه‌ی تلاش برای رد کردن مفروضات مدّ نظر قرار نمی‌گیرد. در واقع بسیاری از مفروضات، حتّی اگر اثبات تئوری نداشته باشند، طیّ سال‌ها توسّط افراد مختلف بررسی شده‌اند و تلاش‌هایی برای نقض‌کردن آن‌ها صورت گرفته است. به عنوان نمونه، رمزهای قالبی AES و DES که پیش‌تر به عدم وجود پشتوانه‌ی تئوری برای امنیت آن‌ها اشاره شد، طیّ سال‌ها عملکرد موفّقی از خود نشان داده‌اند و در عمل، حمله‌ای با کارایی بالا برای آن‌ها ارائه نشده است. چنین مواردی در ادبیات این مقاله و طبقه‌بندی ارائه‌شده توسّط آن مدّ نظر قرار نمی‌گیرد.

۴. به صورت کلّی ممکن است توان محاسباتی ابطال‌گر بیشتر از حدّ انتظار باشد. در واقع ممکن است ابطال‌گر بتواند با دسترسی به توان محاسباتی فراتر از حدّ انتظار طرّاح چالش (و مثلاً با بررسی همه‌ی حالت‌ها) چالش را حل کند. البته با در نظر گرفتن این فرض که پارامتر زمانی t برای مهاجم در دسترس نیست، این مورد مشکلی ایجاد نمی‌کند.

موارد فوق، کاستی‌های طبقه‌بندی ارائه‌شده از دیدگاه خود مقاله بودند. علاوه بر این موارد، از منظر نگارنده‌ی این گزارش نکات دیگری نیز به نظر می‌رسد که در ادامه ذکر خواهد شد.

ابتدا لازم است به این نکته اشاره شود که یکی از موارد مهمّی که طبقه‌بندی ارائه‌شده در این مقاله را ارزشمند می‌کند، امکانی است که برای مقایسه‌ی مفروضات نامرتبط فراهم می‌آورد. به عنوان مثال ممکن است یک سیستم امضای دیجیتال هیچ ارتباطی با یک فرض جبری نظیر فرض تجزیه نداشته باشد؛ امّا به کمک این طبقه‌بندی ممکن است بتوان این دو را از جهاتی با هم مقایسه

کرد؛ چنان که بیان شود فرض تجزیه ابطال‌پذیر کارا است امّا فرض امنیت سیستم مذکور ممکن است تا حدّی ابطال‌پذیر باشد و این یعنی استفاده از فرض سبک‌تر (یعنی ابطال‌پذیری کارا) در عمل می‌تواند مفیدتر باشد، چرا که سیستمی که بر مبنای مفروضات سبک‌تر عمل کند نگرانی کمتری از بابت درستی فرض‌های پشتیبان خود را به همراه می‌آورد.

از طرف دیگر آن‌چه از نظر نگارنده‌ی این گزارش در مورد طبقه‌بندی مطرح‌شده جالب نیست آن است که این طبقه‌بندی تا حدّ زیادی مبتنی بر وضعیت طبیعی مفروضات است. در واقع در عموم موارد، چالش مطرح‌شده برای یک فرض عملاً معادل با همان آزمایشی است که برای مهاجمان حمله‌کننده به فرض مطرح می‌شود. در عمل به نظر می‌رسد که در بسیاری از موارد، در نظر گرفتن دوگانه‌ی فرض–مهاجم و چالش–ابطال‌گر تا حدّی غیرضروری است؛ طوری که این دو مفهوم تا حدّ زیادی منطبق هستند. بسیاری از مواردی که چنین نیستند نیز به دلیل آن است که اصلاً امکان تعریف چالش مطابق با پارامترهای مقاله وجود ندارد. به طور خاص ضعف ساختار ارائه‌شده در برخورد با مفروضات تعاملی (که سهم قابل توجّهی در مفروضات مهمّ رمزنگاری دارند) به نظر یک نقص به حساب می‌آید، چرا که اکثر این مفروضات باید به سادگی در گروه تا حدّی ابطال‌پذیر قرار بگیرند (مواردی نظیر گزاره‌های ۵ و ۱۰) و ساختار معرّفی‌شده توسّط مقاله حرف چندانی برای گفتن در مورد آن‌ها ندارد، به جز احتمالاً در موارد خاص.

در مجموع می‌توان مطالب و طبقه‌بندی ارائه‌شده در این مقاله را یک ایده‌ی مناسب برای یکپارچه‌تر کردن ادبیات رمزنگاری در نظر گرفت؛ با این حال به نظر می‌رسد امکان پرورش‌دادن بیشتر این ایده و ارائه‌ی پیشنهادهای بهتر برای تغییر یا تکمیل آن به منظور آن که بتواند مفروضات بیشتری را فرا بگیرد و مقایسه‌ی قوی‌تر و نابدیهی‌تری از آن‌ها ارائه دهد وجود دارد.

ضمیمه‌ها

الف نمونه‌هایی از مفروضات رمزنگاری

الف.۱ فرض تجزیه

ابتدا $\mathbb{Z}^{(۲)}(n)$ را به صورت ۱ تعریف کنید.

$$\mathbb{Z}^{(۲)}(n) = \{N = pq | \text{اعداد اوّل } p \text{ و } q \text{ } n\text{بیتی هستند}\} \quad (۱)$$

می‌گوییم فرض تجزیه[38] با پارامترهای (n, t, ϵ) برقرار است، هرگاه هر مهاجم $\mathcal{A}$ با دریافت عدد تصادفی $N \in \mathbb{Z}^{(۲)}$ نتواند در زمان t با احتمالی بیشتر یا مساوی ϵ اعداد اوّل p و q را خروجی دهد، طوری که $N = pq$:

$$\forall \mathcal{A} \quad \mathbb{P}[\mathcal{A}(N) = (p, q), p \text{ and } q \text{ prime numbers}, pq = N] < \epsilon \quad (۲)$$

الف.۲ فرض RSA

با تعریف $\mathbb{Z}^{(۲)}(n)$ طبق ۱، و انتخاب تصادفی مقادیر $N \in \mathbb{Z}^{(۲)}(n)$، $s \in \mathbb{Z}_N^*$، و عدد e که نسبت به $\phi(N)$ اوّل باشد؛ برقراری فرض RSA با پارامترهای (n, t, ϵ) آن است که هیچ مهاجمی مانند $\mathcal{A}$ موجود نباشد که در زمانی کمتر از t و با احتمالی حدّاقل برابر با ϵ بتواند عدد a را طوری تولید کند که $a^e = s \mod N$. به عبارت دیگر:

$$\forall \mathcal{A} \quad \mathbb{P}[\mathcal{A}(N, e, s) = a \, , \; a^e = s \mod N] < \epsilon \quad (۳)$$

الف.۳ فرض RSA تفاضلی

فرض RSA تفاضلی[39] با سختی یافتن دو پیش‌تصویر مختلف و جدید در مسأله‌ی RSA سروکار دارد، طوری که تفاضل آن‌ها برابر با مقدار مشخّص D باشد و این درحالی است که مهاجم می‌تواند تعدادی از نمونه‌های داری چنین شرطی را (به صورت تعاملی) مشاهده کرده باشد. به عبارت دقیق‌تر، با انتخاب تصادفی $N \in \mathbb{Z}^{(۲)}(n)$ و $D \in \mathbb{Z}_N^*$، مهاجم $\mathcal{A}$ می‌تواند به $m - ۱$ زوج (x_i, y_i) دسترسی داشته باشد که برای هر یک، $x_i^e - y_i^e = D \mod N$ و نهایتاً باید یک زوج متمایز با $m - ۱$ زوج اوّلیه به صورت (x_m, y_m) تولید کند که $x_m^e - y_m^e = D \mod N$.

برقراری فرض RSA تفاضلی با پارامترهای (n, t, ϵ) بدان معناست که برای هر مهاجم $\mathcal{A}$ که در زمان t اجرا می‌شود، با دریافت (N, e, D) و همچنین دریافت مجموعه‌ی $Q = \{(x_i, y_i) | ۱ \leq i \leq m - ۱ \, , \; x_i^e - y_i^e = D \mod N\}$ داریم:

$$\mathbb{P}[\mathcal{A}(N, e, D) = (x_m, y_m) \notin Q \, , \; x_m^e - y_m^e = D \mod N] < \epsilon \quad (۴)$$

[38] factoring
[39] difference RSA

الف.۴ فرض دانشِ نما

فرض دانشِ نما[۴۰] ناظر به آن است که با دانستن اعداد g و g^a (که g مولّد یک گروه دوری مثل $\mathbb{G}$ از مرتبه‌ی q است)، تنها راه کارا برای تولید اعداد Y و C طوری که $Y = C^a$ آن است که عددی مثل c انتخاب شود و قرار داده شود $C = g^c$ (به استناد [۳]).

به عبارت دیگر، با انتخاب یک مقدار تصادفی مانند x و تولید مقدار $h = g^x$، هر مهاجم $\mathcal{A}$ که با دریافت g و h خروجی‌های $h_1 = g^a \mod P$ و $h_2 = h^a \mod P$ را تولید کند ($P = 2q+1$ عددی اوّل است)، باید در صورت برقراری فرض دانشِ نما با پارامترهای (n, t, ϵ)، از مقدار a مطّلع باشد و به کمک آن h_1 و h_2 را تولید کرده باشد. به عبارت دیگر، برای هر مهاجم $\mathcal{A}$ که خروجی‌های مطلوب را در حدّاکثر زمان t تولید کند، الگوریتم $\mathcal{A}'$ که در زمان $t' = \mathsf{poly}(t)$ اجرا می‌شود وجود خواهد داشت، طوری که

$$\left| \sum_a \mathbb{P}[\mathcal{A}(g,h) = (g^a, h^a)] - \sum_a \mathbb{P}[\mathcal{A}'(g,h) = (a, g^a, h^a)] \right| < \epsilon \qquad (۵)$$

توجّه کنید فرآیند فوق شبیه به نوعی اثبات دانایی[۴۱] (که مفهوم آن در ضمیمه‌ی و توضیح داده شده) می‌باشد، با این تفاوت که در آن، هیچ استخراج‌گر دانش[۴۲] وجود ندارد (چرا که تعاملی بین اثبات‌گر[۴۳] و تصدیق‌گر[۴۴] وجود ندارد) و فرض دانشِ نما، یک گزاره‌ی وجودی است که صرفاً بیان می‌دارد همواره می‌توان توزیع احتمالاتی مشابهی با دانستن مقدار a تولید کرد.

ب کلاس‌های پیچیدگی مسائل

تعاریف و مفاهیم این بخش به استناد [۴] بیان می‌شوند.

به صورت کلّی مسائل مختلف را می‌توان بر حسب زمان لازم برای حلّ آن‌ها به کلاس‌های کلّی تقسیم کرد. پیش از معرّفی این کلاس‌ها، مفهوم دیگری را معرّفی می‌کنیم که ناظر به امکان حلّ یک مسأله در صورت حلّ مسأله‌ای دیگر است.

تعریف ۱۱. **(تحویل چندجمله‌ای)** *فرض کنید X و Y دو مسأله باشند، طوری که اگر X در زمان چندجمله‌ای قابل حل باشد، آن‌گاه Y نیز چنین است. در این صورت می‌گوییم Y با یک تحویل چندجمله‌ای قابل تبدیل به X است و می‌نویسیم $Y \leq_P X$.*

تعریف ۱۱ به صورت شهودی بدان معناست که اگر مسأله‌ی X حل شود، آن‌گاه می‌توان از الگوریتم موجود برای حلّ آن کمک گرفت و مسأله‌ی Y را نیز حل کرد.

در ادامه کلاس‌هایی برای سختی مسائل را معرّفی خواهیم کرد.

تعریف ۱۲. **(کلاس پیچیدگی P)** *کلاس $\mathbf{P}$ مجموعه‌ی همه‌ی مسائلی است که یک الگوریتم با زمان اجرای چندجمله‌ای بر حسب طول ورودی برای حلّ آن‌ها موجود است.*

تعریف ۱۳. **(کلاس پیچیدگی NP)** *کلاس $\mathbf{NP}$ مجموعه‌ی همه‌ی مسائلی است که می‌توان درستی یا نادرستی هر پاسخ ارائه‌شده برای آن‌ها را در زمان چندجمله‌ای تحقیق کرد. به عبارت دیگر، $X \in \mathbf{NP}$ است، هرگاه برای X یک تصدیق‌گر[۴۵] کارا (یعنی*

[40] knowledge of exponent
[41] proof of knowledge
[42] knowledge extractor
[43] prover
[44] verifier
[45] certifier

عمل‌کننده در زمان چندجمله‌ای) موجود باشد.

به صورت شهودی، مسائل کلاس $\mathbf{NP}$ ممکن است در زمان چندجمله‌ای قابل حل نباشند؛ در حالی که اگر شخصی ادّعا کند برای چنین مسأله‌ای راه حلّی دارد، می‌توان درستی آن را در زمان چندجمله‌ای تحقیق کرد. به عنوان مثال، مسأله‌ی ۳–رنگ‌آمیزی[46] یک گراف چنین مسأله‌ای است؛ یعنی الگوریتمی چندجمله‌ای برای یافتن چنین رنگ‌آمیزی‌ای برای یک گراف دلخواه موجود نیست؛ با این حال اگر یک رنگ‌آمیزی ارائه شود می‌توان درستی آن (یعنی هم‌رنگ نبودن رئوس دو سر یک یال) را در زمان چندجمله‌ای بررسی کرد. در ادامه یک قضیه در مورد کلاس‌های $\mathbf{P}$ و $\mathbf{NP}$ و همچنین دو کلاس پیچیدگی معروف دیگر را معرّفی خواهیم کرد.

قضیه ۱. *هر مسأله‌ی $\mathbf{P}$ یک مسأله‌ی $\mathbf{NP}$ است؛ یعنی $\mathbf{P} \subseteq \mathbf{NP}$.*

تعریف ۱۴. *(**مسأله‌ی ان‌پی–سخت**[47]) گوئیم مسأله‌ی X ان‌پی–سخت است، هرگاه برای هر مسأله $Y \in \mathbf{NP}$ داشته باشیم $Y \leq_P X$.*

تعریف ۱۵. *(**مسأله‌ی ان‌پی–تمام**[48]) گوئیم مسأله‌ی X ان‌پی–تمام است، هرگاه $X \in \mathbf{NP}$ و X یک مسأله‌ی ان‌پی–سخت باشد.*

نهایتاً پیش از پایان این بخش اشاره می‌کنیم که یکی از مهم‌ترین مسائل باز علوم کامپیوتر، درستی یا نادرستی برابری $\mathbf{P} = \mathbf{NP}$ است. با وجود این که باور عمومی بر عدم برقراری این تساوی است، امّا این گزاره هنوز به صورت رسمی اثبات یا رد نشده است. شایان ذکر است که اگر برقراری $\mathbf{P} = \mathbf{NP}$ اثبات شود؛ بسیاری از مسائل که «سخت» تلقّی می‌شدند، دیگر سخت نخواهند بود و با الگوریتم‌های چندجمله‌ای حل خواهند شد. چنین موضوعی امنیت بسیاری از سیستم‌های رمزنگاری را نیز فاقد اعتبار خواهد کرد.

ج اثبات ابطال‌پذیری کارای فرض تجزیه

برای اثبات ابطال‌پذیری کارای فرض تجزیه (مطابق تعریف بخش الف.۱)، ابتدا فرض می‌کنیم این فرض با پارامترهای (n, t, ϵ) برقرار نباشد. برای این منظور، فرض می‌کنیم کسری به اندازه‌ی 2ϵ از $\mathbb{Z}^{(2)}(n)$ (مطابق تعریف ۱) به سادگی قابل تجزیه‌اند و مسأله‌ی تجزیه برای آن‌ها سخت نیست. این مجموعه از اعداد را $\mathbb{Z}^{(2)}_{\text{easy}}(n)$ می‌نامیم؛ بنابراین

$$\mathbb{Z}^{(2)}_{\text{easy}}(n) \subseteq \mathbb{Z}^{(2)}(n) \quad , \quad \frac{|\mathbb{Z}^{(2)}_{\text{easy}}(n)|}{|\mathbb{Z}^{(2)}(n)|} = \frac{1}{2\epsilon} \tag{6}$$

حال اگر توزیع D_n برای چالش‌ها را انتخاب یک N تصادفی از $\mathbb{Z}^{(2)}(n)$ در نظر بگیریم و چالش آن باشد که عدد N داده‌شده به عوامل اوّل p و q تجزیه شود ($N = p.q$)، آنگاه احتمال آن که یک ابطال‌گر موفّق به پیروزی در این چالش شود برابر با 2ϵ خواهد بود. این در حالی است که طبق تعریف، این احتمال می‌تواند ناچیز باشد، در حالی که لازم است احتمال پیروزی ابطال‌گر در این چالش،احتمال (غیرناچیز) $1 - \delta$ باشد. به عبارت دیگر، لازم است تمهیدی اندیشیده شود تا احتمال پیروزی بالا برود. این مشکل به سادگی حل می‌شود و کافی است چالش را چنین تغییر دهیم که تعداد اعداد تصادفی مانند N که به عنوان چالش داده می‌شوند، از مرتبه‌ی $1/\epsilon$ باشد و لازم باشد ابطال‌گر فقط یکی از این اعداد را به عوامل اوّل تجزیه کند. به این ترتیب، به صورت متوسّط یکی از این اعداد عضوی از $\mathbb{Z}^{(2)}_{\text{easy}}(n)$ خواهد بود و ابطال‌گر با یک احتمال ثابت و بالا به موفّقیت می‌رسد.

[46] 3–coloring
[47] NP–hard
[48] NP–complete

مشکل راهکار فوق آن است که طبق تعریف ۷، برای آن که فرضی ابطال‌پذیر کارا باشد لازم است فرآیند نمونه‌برداری از D_n برای ساخت یک چالش مثل $d \in D_n$ تابعی چندجمله‌ای از n، $\log ۱/\epsilon$، و $\log ۱/\delta$ باشد. فرآیند فوق، مستلزم $۱/\epsilon$ مرتبه نمونه‌برداری است که شرط تابعیت چندجمله‌ای از $۱/\log\epsilon$ را نقض می‌کند. به این ترتیب لازم است تمهید دیگری در این مورد اندیشیده شود. یک مورد قابل توجّه دیگر نیز آن است که دسترسی به $\mathbb{Z}^{(۲)}(n)$ به صورت صریح ممکن است امکان‌پذیر نباشد و بنابراین انتخاب حتّی یک نمونه‌ی تصادفی از آن می‌تواند زمان‌بر باشد.

برای حلّ مورد دوم، فعلاً فرض می‌کنیم که نمونه‌برداری از از میان تمامی اعداد $۲n$ بیتی به صورت یکنواخت انجام می‌شود و خود را محدود به انتخاب از $\mathbb{Z}^{(۲)}(n)$ نمی‌کنیم. می‌دانیم $\mathbb{Z}^{(۲)}(n)$ در مجموعه‌ی همه‌ی اعداد $۲n$ بیتی تُنُک[49] نیست، و اگر تعداد چندجمله‌ای نمونه از اعداد $۲n$ بیتی انتخاب کنیم، کسر قابل توجّهی از آن‌ها عضو $\mathbb{Z}^{(۲)}(n)$ خواهند بود. این که عضو انتخابی واقعاً در $\mathbb{Z}^{(۲)}(n)$ موجود بوده یا نه را در مرحله‌ی تصدیق پاسخ اعلام‌شده توسّط ابطال‌گر بررسی خواهیم کرد.

حال به سراغ حلّ مشکل اوّل (یعنی زیاد بودن تعداد نمونه‌های مورد نیاز) می‌رویم. برای این منظور، از خانواده‌ای از توابع چکیده‌ساز[50] که دارای خاصیت استقلال دوجانبه[51] باشند، استفاده می‌کنیم. تعریف این خاصیت به صورت زیر است:

تعریف ۱۶. **(توابع چکیده‌ساز مستقل دوجانبه)** *یک خانواده از توابع چکیده‌ساز مانند $\mathcal{H} = \{h | h : A \to B\}$ را مستقل دوجانبه گوییم، هرگاه برای هر $\{i, j\} \subseteq A$ (که $i \neq j$) و هر $x \in B$ و $y \in B$ داشته باشیم:*

$$\mathop{\mathbb{P}}_{h \leftarrow \mathcal{H}}[h(i) = x \wedge h(j) = y] = \frac{۱}{|B|^۲} \tag{۷}$$

حال خانواده‌ای از توابع چکیده‌ساز را با خاصیت استقلال دوجانبه در نظر می‌گیریم که توابع موجود در آن به صورت $h : \{۰, ۱\}^n \to \{۰, ۱\}^{n - ۲\log n + \log ۲\epsilon}$ باشند. همچنین لازم است که وارون توابع $\mathcal{H}$ روی یک نقطه‌ی خاص به سادگی قابل محاسبه باشد. حال چالش مربوطه به این صورت تعریف می‌شود که یک تابع h به تصادف از $\mathcal{H}$ و یک عدد تصادفی c از برد h انتخاب شده و زوج (h, c) به عنوان صورت چالش داده می‌شوند. ابطال‌گر باید به عنوان پاسخ دو عدد n بیتی p و q را برگرداند، طوری که $h(p.q) = c$. برقراری این شرط به سادگی توسّط تصدیق‌گر قابل بررسی است. به علاوه، مورد دیگری که توسّط تصدیق‌گر بررسی می‌شود، اوّل بودن p و q است. در صورتی که هر دو شرط برقرار باشند، خروجی تصدیق‌گر accept خواهد بود. لازم است دقّت شود که الگوریتم‌هایی کارا برای بررسی اوّل بودن یا نبودن اعداد وجود دارند، بنابراین بررسی اوّل بودن p و q از نظر مرتبه‌ی زمانی مشکلی ایجاد نخواهد کرد. به این ترتیب در صورت وجود مهاجمی مثل $\mathcal{A}$ برای فرض تجزیه با پارامترهای (n, t, ϵ)، کافی است ابطال‌گر برای هر $N \in h^{-۱}(c)$، الگوریتم $\mathcal{A}$ را بر روی N اجرا کند و خروجی‌های آن، p و q را اعلام کند. با توجّه به این که برد h شامل اعداد $(n - ۲\log n + ۲\log\epsilon)$بیتی بوده و دامنه‌ی آن شامل اعداد n بیتی است، به طور متوسّط به ازای هر عضو از برد، $(n/\epsilon)^۲$ عضو در دامنه وجود دارند، و این یعنی برای یک c انتخابی به عنوان ورودی چالش، $|h^{-۱}(c)|$ از مرتبه $(n/\epsilon)^۲$ عضو خواهد داشت. به این ترتیب مشکل در صرف زمانی از مرتبه $۱/\epsilon$ برای نمونه‌برداری حل شده و تنها با انتخاب یک عدد c، ابطال‌گر با $O(n^۲/\epsilon^۲)$ عدد N روبه‌رو می‌شود که کافی است بتواند یکی از آن‌ها را تجزیه کند (دقّت کنید که محدودیت زمانی برای اجرای خود الگوریتم ابطال‌گر مطابق تعریف ۷ تنها مقیّد به تابعیت چندجمله‌ای از $۱/\epsilon$ است و لذا مشکلی از نظر محدودیت زمانی وجود نخواهد داشت). طبق توضیحاتی که پیش‌تر ارائه شد، در این حالت با احتمال ثابتی ابطال‌گر موفّق می‌شود؛ و برای رسیدن به احتمال خطایی حدّاکثر برابر با $۱ - \delta$ کافی است کلّ این فرآیند از مرتبه $\log ۱/\delta$ بار تکرار شود که مشکلی در محدودیت‌های زمانی ایجاد نخواهد کرد.

به این ترتیب نشان داده‌شده است که مطابق با تعریف ۷، فرض تجزیه ابطال‌پذیر کارا می‌باشد. شایان ذکر است که الگوریتم

[49] sparse
[50] hash function
[51] pair-wise independence

و پارامترهای اثبات مطابق با روش ارائه‌شده در [۱] می‌باشند و نگارنده صرفاً استنباط شخصی خود در مورد درستی این نکات را اضافه کرده است؛ امّا در برخی موارد (نظیر انتخاب تعداد بیت‌های خروجی توابع چکیده‌ساز) می‌توان انتخاب‌های درست دیگری نیز انجام داد.

د　توابع و جایگشت‌های یک‌طرفه

تعاریف این قسمت به استناد [۲] ارائه می‌شوند.

تعریف ۱۷. (***تابع یک‌طرفه***[52]) *تابع* $f : \{0,1\}^* \to \{0,1\}^*$ *یک تابع یک‌طرفه است، اگر دو شرط زیر برقرار باشند:*

۱. *(محاسبه‌پذیری آسان) یک الگوریتم چندجمله‌ای مانند* M_f *برای محاسبه‌ی* f *موجود باشد، یعنی* $M_f(x) = f(x)$ *برای هر* x.

۲. *(وارون‌پذیری دشوار) برای هر الگوریتم چندجمله‌ای احتمالاتی* $\mathcal{A}$*، تابعی ناچیز مثل* negl *وجود داشته باشد، طوری که*

$$\mathbb{P}[\mathsf{Invert}_{\mathcal{A},f}(n) = 1] \le \mathsf{negl}(n) \qquad (8)$$

که منظور از $\mathsf{Invert}_{\mathcal{A},f}(n)$*، خروجی آزمایش وارون‌کردن تابع مطابق تعریف ۱۸ است.*

تعریف ۱۸. (***آزمایش وارون‌کردن تابع***) *آزمایش وارون کردن یک تابع مثل* f *برای مهاجم* $\mathcal{A}$ *مطابق گام‌های زیر تعریف‌شده و خروجی آن با* $\mathsf{Invert}_{\mathcal{A},f}(n)$ *نشان داده می‌شود:*

۱. *مقدار* $x \in \{0,1\}^n$ *به تصادف انتخاب می‌شود و مقدار* $y = f(x)$ *از روی آن محاسبه می‌شود.*

۲. *مهاجم* $\mathcal{A}$ *با دریافت ورودی‌های* 1^n *و* y*، مقدار* x' *را در خروجی اعلام می‌کند.*

۳. *خروجی آزمایش برابر با ۱ است، هرگاه* $f(x') = y$*. در غیر این صورت خروجی آزمایش برابر با صفر خواهد بود.*

همچنین جایگشت‌های یک‌طرفه نیز مشابه با توابع یک‌طرفه تعریف می‌شوند:

تعریف ۱۹. (***جایگشت یک‌طرفه***[53]) *گوییم* $f : \{0,1\}^* \to \{0,1\}^*$ *یک جایگشت یک‌طرفه است، هرگاه:*

۱. f *یک جایگشت (یعنی تابعی یک‌به‌یک و وارون‌پذیر) باشد.*

۲. *تابع* f *مطابق تعریف ۱۷ یک‌طرفه باشد.*

ه　اثبات ابطال‌پذیری فرض شبه‌تصادفی بودن یک مولّد

فرض کنید تابع $G : \{0,1\}^m \to \{0,1\}^n$ داده شده است. برای اثبات ابطال‌پذیری فرض «G یک مولّد شبه‌تصادفی است»، لازم است چالش مناسب تعریف شود.

[52] one-way function
[53] one-way permutation

برای این منظور، چالش به این صورت تعریف می‌شود که تعداد «زیادی» زوج به صورت (x_i, y_i) تولید می‌شود که یکی از آن‌ها خروجی G روی یک ورودی تصادفی است و دیگری یک عدد تصادفی nبیتی می‌باشد. منظور از تعداد زیاد، تعدادی چندجمله‌ای بر حسب n و $۱/\epsilon$ می‌باشد. وظیفه‌ی ابطال‌گر آن است که مشخّص کند در هر یک از این زوج‌ها، کدام مؤلّفه خروجی تابع G روی ورودی تصادفی بوده است. به صورت متوسّط اگر ابطال‌گر به صورت تصادفی عمل کند می‌تواند نیمی از این نمونه‌ها را به درستی تشخیص دهد؛ بنابراین برای پیروزی در چالش (و رد شدن این فرض که تشخیص خروجی‌های G از اعداد تصادفی با احتمالی حدّاقل برابر با ϵ ممکن نیست) لازم است ابطال‌گر بتواند حدّاقل کسری برابر با $۱/۲ + \epsilon/۲$ از زوج‌های داده‌شده را به درستی تشخیص دهد. ضمناً لازم است الگوریتم تصدیق‌گر از پاسخ‌های درست مربوط به هر زوج آگاهی داشته باشد.

دقّت کنید که نکته‌ی مهمّی که این فرض را ابطال‌پذیر (و نه ابطال‌پذیر کارا) می‌کند آن است که فرآیند نمونه‌برداری از D_n و تولید چالش زمانی از مرتبه‌ی $\mathsf{poly}(۱/\epsilon)$ می‌برد و این زمان بر حسب $\log ۱/\epsilon$ چندجمله‌ای نیست. از طرفی تولید این تعداد زوج مرتّب ضروری است؛ چرا که در غیر این صورت به طور متوسّط حتّی در صورت غلط بودن فرض مربوطه، احتمال موفّقیت ابطال‌گر ممکن است کمتر از یک مقدار ثابت باشد.

نهایتاً برای آن که احتمال ثابت موفّقیت ابطال‌گر به حدّ مطلوب (حدّاقل برابر با $۱ - \delta$) برسد، ممکن است نیاز باشد که کلّ این فرآیند از مرتبه‌ی $\log ۱/\delta$ بار اجرا شود که مشکلی از نظر زمانی در تعریف ابطال‌پذیری ایجاد نمی‌کند.

و اثبات دانایی

به صورت کلّی، اثبات دانایی یک فرآیند تعاملی است که در آن، یک اثبات‌گر[۵۴] موفّق می‌شود یک تصدیق‌گر[۵۵] را قانع کند که از چیزی مطّلع است. در ادامه، تعریفی رسمی بر مبنای [۵] و [۱۲] برای اثبات دانایی ارائه می‌دهیم.

تعریف ۲۰. (***اثبات دانایی***[۵۶]) *فرض کنید x گزاره‌ای از یک زبان $\mathbf{NP}$ مانند L باشد و $W(x)$ مجموعه‌ی شواهد x باشند که باید در اثبات مورد پذیرش قرار گیرند. در این صورت می‌توان رابطه‌ای به صورت $R = \{(x, w) | x \in L, w \in W(x)\}$ تعریف کرد. در این صورت یک اثبات دانایی برای رابطه‌ی R با خطای دانایی κ یک پروتکل تعاملی با یک اثبات‌گر P و یک تصدیق‌گر V است که برای آن‌ها، دو خاصیت زیر برقرار باشد:*

۱. *کامل بودن: اگر $(x, w) \in R$، آن‌گاه اثبات‌گر P که از شاهد w برای x مطّلع است، با احتمال ۱ موفّق می‌شود تصدیق‌گر V را به دانش خود قانع سازد. به عبارت دقیق‌تر، با دانستن تعامل بین P و V، احتمال قانع‌شدن V برابر با ۱ است:*

$$\mathbb{P}[P(x, w) \leftrightarrow V(x) \to ۱] = ۱$$

۲. *اعتبار: شرط اعتبار آن است که احتمال موفّقیت هر استخراج‌گر دانش[۵۷] مانند E در استخراج شاهد w از اثبات‌گر P، هنگامی که به آن دسترسی اوراکلی داشته باشد، حدّاقل برابر است با احتمال موفّقیت P در قانع‌کردن تصدیق‌گر V. این خاصیت بدان معناست که هیچ اثبات‌گری بدون دانستن شاهد، با احتمال قابل توجّهی قادر به قانع‌کردن تصدیق‌گر نخواهد بود.*

[54] prover
[55] verifier
[56] knowledge proof
[57] knowledge extractor

ز اثبات دانایی صفر

اثبات دانایی صفر[۵۸] یا پروتکل دانایی صفر[۵۹] به صورت شهودی فرآیندی است که در آن، امکانی فراهم می‌شود تا درستی یک اثبات نشان داده شود، بدون آن که فرآیند اصلی اثبات بیان شود. به عبارت دیگر، در حالت عادی زمانی که فردی قصد داشته باشد نشان دهد که گزاره‌ای درست است، باید اثباتی از آن ارائه دهد به نحوی که مخاطب با بررسی و دانستن آن اثبات، ادّعای شخص اثبات‌گر را بپذیرد؛ با این حال هدف از اثبات دانایی صفر آن است که شخص اثبات‌گر بدون آن که اثبات را صراحتاً بیان کند، صرفاً مخاطب خود را قانع سازد که وی از اثبات درست مطّلع است.

در ادامه دو تعریف رسمی برای روشن‌تر شدن مفهوم ارائه خواهیم کرد. تعریف ۲۱ به نقل از [۶] ارائه می‌شود.

تعریف ۲۱. **(پروتکل دانایی صفر)** *استراتژی تعاملی A را دانایی صفر بر روی (ورودی‌هایی از) مجموعه‌ی S گوئیم، هرگاه برای هر استراتژی (تعاملی) مانند B^*، ساختار محاسباتی غیرتعاملی C^* موجود باشد، طوری که دو توزیع احتمالاتی زیر به صورت محاسباتی تمایزناپذیر باشند:*

۱. $\{(A, B^*)(x)\}_{x\in S}$: *خروجی B^* پس از تعامل با A بر روی ورودی مشترک $x \in S$*

۲. $\{C^*(x)\}_{x\in S}$: *خروجی C^* بر روی ورودی $x \in S$*

به صورت شهودی، منظور از توزیع احتمالاتی اوّل در تعریف فوق، فرآیند تعاملی واقعی صورت‌گرفته بین ارائه‌دهنده‌ی اثبات و پذیرنده‌ی آن است و منظور از توزیع احتمالاتی دوم، فرآیندی غیرتعاملی است که به نوعی وظیفه‌ی شبیه‌سازی را بر عهده دارد. در صورتی که تعامل بین A و B^* دانایی صفر باشد، هر آن‌چه که در این فرآیند به اطّلاعات B افزوده می‌شود باید تنها از طریق x قابل حصول باشد، لذا C^* باید وجود داشته باشد که بتواند همان نتایج را به صورت غیرتعاملی و تنها با دانستن x به دست آورد.

تعریف ۲۱ کمی پیچیده و انتزاعی است. در ادامه، تعریف ۲۲ را به نقل از [۱۱] بیان می‌کنیم تا به صورت روشن‌تری مفهوم مورد نظر را بیان کند.

تعریف ۲۲. **(اثبات دانایی صفر)** *فرض کنید L یک زبان $\mathbf{NP}$ و M یک ماشین تورینگ چندجمله‌ای باشد طوری که $x \in L$ اگر و تنها اگر $u \in \{0,1\}^{p(|x|)}$ موجود باشد، طوری که $M(x,u) = 1$ که در آن، p یک چندجمله‌ای است.*

در این صورت زوج P و V از الگوریتم‌های تصادفی چندجمله‌ای تعاملی را یک اثبات دانایی صفر برای L گویند، هرگاه سه شرط زیر برقرار باشد:

۱. *کامل‌بودن: برای هر $x \in L$ و هر گواه آن مانند u (یعنی $M(x,u) = 1$):*

$$\mathbb{P}[\mathsf{Out}_V\langle P(x,u), V(x)\rangle = 1] \geq \frac{2}{3}$$

که در آن، $\langle P(x,u), V(x)\rangle$ نشان‌دهنده‌ی تعامل P و V است که P ورودی‌های x و u، و V ورودی x را دریافت می‌کند و Out_V نشان‌دهنده‌ی خروجی V در پایان فرآیند تعامل می‌باشد.

۲. *صحّت: اگر $x \notin L$، آن‌گاه برای هر استراتژی P^* و ورودی u*

$$\mathbb{P}[\mathsf{Out}_V\langle P^*(x,u), V(x)\rangle = 1] \leq \frac{1}{3}$$

که نیازی نیست استراتژی P^* لزوماً چندجمله‌ای باشد.

[58] zero knowledge proof
[59] zero knowledge protocol

۳. *دانایی صفر کامل: برای هر استراتژی تصادفی چندجمله‌ای تعاملی V^*، یک الگوریتم چندجمله‌ای تصادفی (متّکی به خود[۶۰]) مانند S^* موجود باشد، طوری که برای هر $x \in L$ و هر گواه آن مانند u (یعنی $M(x,u) = 1$) داشته باشیم:*

$$\mathsf{Out}\langle P(x,u), V^*(x)\rangle \equiv S^*(x)$$

یعنی دو متغیّر تصادفی هم‌توزیع و در نتیجه تمایزناپذیر باشند؛ و این در حالی است که S^* دسترسی‌ای به هیچ گواهی (مانند u) برای x ندارد.

در تعریف فوق، الگوریتم‌های P، V، و S را به ترتیب اثبات‌گر[۶۱]، تصدیق‌گر[۶۲]، و شبیه‌ساز[۶۳] می‌نامند. در واقع P اثبات دانایی صفر را ارائه می‌دهد و V آن را تأیید می‌کند و در صورتی که اثبات به معنای واقعی دانایی صفر باشد، شبیه‌ساز S می‌تواند نتایج مشابه را تنها با دریافت ورودی x تولید کند و این یعنی هیچ اطّلاع اضافه‌ای در فرآیند تعاملی P و V منتقل نشده است.

در ادامه یک تعریف دیگر نیز به نقل از [۷] ارائه می‌دهیم که در آن، یک شرط سبک‌تر نسبت به حالت دانایی صفر برای اثبات‌ها ارائه می‌شود.

تعریف ۲۳. ***(خاصیت تمایزناپذیری شاهد[۶۴])*** *یکی از انواع اثبات‌های دانایی صفر را دارای خاصیت تمایزناپذیری شاهد گوییم، هرگاه الگوریتم اثبات‌گر از یک شاهد[۶۵] به عنوان ورودی پروتکل اثبات استفاده کند؛ و الگوریتم تصدیق‌گر چیزی به جز درستی گزاره‌ی مورد اثبات را متوجّه نشود.*

در واقع تعریف فوق، شکل ساده‌تری از پروتکل دانایی صفر است که در آن، تصدیق‌گر نمی‌تواند تمایزی بین اثبات‌گرهایی که از شواهد مختلف استفاده می‌کنند ایجاد کند؛ امّا ممکن است اطّلاعاتی در مورد مجموعه‌ی همه‌ی شاهدها کسب کند یا در صورتی که تنها یک شاهد وجود داشته باشد، آن را بفهمد.

ح خودکاهش‌پذیری تصادفی

تعریف ۲۴. ***(خودکاهش‌پذیری تصادفی[۶۶])*** *تابع f را خودکاهش‌پذیر تصادفی گویند، هرگاه محاسبه‌ی f بر روی هر ورودی دلخواه را بتوان در زمان چندجمله‌ای به محاسبه‌ی f بر روی تعدادی ورودی تصادفی کاهش داد.*

تعریف فوق که به نقل از [۸] آورده شده است، بیان می‌دارد که خودکاهش‌پذیری تصادفی خاصیتی است که این امکان را فراهم می‌آورد تا برای محاسبه‌ی مقدار یک تابع بر روی یک ورودی، بتوان از محاسبه‌ی مقدار آن بر روی ورودی‌های تصادفی بهره برد. به عبارت دیگر، این خاصیت یعنی در صورتی که بتوان یک مسأله را بر روی درصد بالایی از نمونه‌های آن حل کرد؛ آن‌گاه می‌توان آن را بر روی تمامی نمونه‌ها حل کرد.

قضیه ۲. ***(خودکاهش‌پذیری مسأله‌ی لگاریتم گسسته[۶۷])*** *مسأله‌ی لگاریتم گسسته خودکاهش‌پذیر است؛ یعنی اگر $\mathbb{G}$ یک گروه دوری از مرتبه‌ی q باشد، در این صورت اگر یک الگوریتم قطعی چندجمله‌ای مانند A موجود باشد، طوری که مسأله‌ی لگاریتم*

[60] stand-alone
[61] prover
[62] verifier
[63] simulator
[64] witness indistinguishability
[65] witness
[66] random self-reducibility
[67] discrete logarithm problem

گسسته را برای کسر $۱/\mathsf{poly}(n)$ از کلّ ورودی‌های ممکن حل کند (که در آن، اندازه‌ی ورودی $n = \log q$ است)، آن‌گاه یک الگوریتم تصادفی چندجمله‌ای وجود دارد که مسأله‌ی لگاریتم گسسته را برای هر ورودی دلخواه حل می‌کند.

طرحی از اثبات. فرض کنید g مولّد گروه دوری $\mathbb{G}$ باشد. همچنین فرض کنید $h \in \mathbb{G}$ عضوی از گروه بوده و لگاریتم گسسته‌ی آن در پایه‌ی g برابر با x باشد، یعنی $h = g^x$. حال فرض کنید y به صورت تصادفی و یکنواخت از $\mathbb{G}$ انتخاب شود. در این صورت $hg^y = g^{x+y}$ نیز توزیعی یکنواخت در $\mathbb{G}$ دارد. به این ترتیب توزیع hg^y مستقل از h بوده و با احتمال $۱/\mathsf{poly}(n)$ می‌توان لگاریتم گسسته‌ی آن را در زمان چندجمله‌ای محاسبه کرد. در صورتی که چنین کاری ممکن باشد، لگاریتم گسسته‌ی h، یعنی x نیز محاسبه می‌شود:

$$\log_g h = \log_g(hg^y) - y \mod |\mathbb{G}|$$

به این ترتیب با اجرای تعداد چندجمله‌ای بار از فرآیند فوق، با احتمال بالا در یکی از دفعات امکان حلّ مسأله فراهم می‌شود؛ بنابراین مسأله‌ی لگاریتم گسسته خودکاهش‌پذیر تصادفی است. □

قضیه ۳. (خودکاهش‌پذیری مسأله‌ی دیفی-هلمن محاسباتی[68]**)** *مسأله‌ی دیفی-هلمن محاسباتی خودکاهش‌پذیر است؛ یعنی اگر $\mathbb{G}$ یک گروه دوری از مرتبه‌ی q باشد، در این صورت اگر یک الگوریتم قطعی چندجمله‌ای مانند A موجود باشد، طوری که مسأله‌ی دیفی-هلمن محاسباتی را برای کسر $۱/\mathsf{poly}(n)$ از کلّ ورودی‌های ممکن حل کند، آن‌گاه یک الگوریتم تصادفی چندجمله‌ای وجود دارد که مسأله‌ی لگاریتم گسسته را برای هر ورودی دلخواه حل می‌کند.*

طرحی از اثبات. فرض کنید g مولّد گروه دوری $\mathbb{G}$ باشد. همچنین فرض کنید g^x و g^y با انتخاب تصادفی x و y از $\mathbb{Z}_q$ داده شده‌اند. در این صورت اگر z نیز به صورت تصادفی با توزیع یکنواخت انتخاب شود، g^zg^x و g^zg^y نیز دارای توزیع‌های تصادفی یکنواخت و مستقل از g^x و g^y خواهند بود. حال با احتمال $۱/\mathsf{poly}(n)$ می‌توان از روی g^{x+z} و g^{y+z}، مقدار $g^{(x+z)(y+z)}$ را محاسبه کرد. داریم:

$$g^{(x+z)(y+z)} = g^{xy}g^{xz}g^{yz}g^{z^۲}$$

با توجّه به این که مقادیر g^{xz}، g^{yz}، و $g^{z^۲}$ با داده‌های موجود به صورت مستقیم قابل محاسبه است، مقدار g^{xy} به دست خواهد آمد:

$$g^{xy} = g^{(x+z)(y+z)}/(g^{xz}g^{yz}g^{z^۲})$$

به این ترتیب با اجرای تعداد چندجمله‌ای بار از فرآیند فوق، با احتمال بالا در یکی از دفعات امکان حلّ مسأله فراهم می‌شود؛ بنابراین مسأله‌ی دیفی-هلمن محاسباتی خودکاهش‌پذیر تصادفی است.

□

[68] computational Diffie–Hellman problem

مراجع

[1] Naor, Moni. (2003). *On Cryptographic Assumptions and Challenges*. CRYPTO 2003. LNCS. 2729. 96-109. 10.1007/978-3-540-45146-4_6.

[2] Jonathan Katz and Yehuda Lindell. 2007. *Introduction to Modern Cryptography* (Chapman & Hall/Crc Cryptography and Network Security Series). Chapman & Hall/CRC.

[3] Bellare, Mihir & Palacio, Adriana. (2004). *The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols*. 3152. 273-289. 10.1007/978-3-540-28628-8_17.

[4] Kleinberg, Jon and Tardos, Eva. *Introduction to Algorithms*. 1st ed., 2005.

[5] Bellare, Mihir and Goldreich, Oded. (1999). *On Defining Proofs of Knowledge*. 10.1007/3-540-48071-4_28.

[6] Goldreich, Oded. (2013). *A short tutorial of zero-knowledge*. Cryptology and Information Security Series. 10. 28-60. 10.3233/978-1-61499-169-4-28.

[7] Feige, U.; Shamir, A. (1990). *Witness indistinguishable and witness hiding protocols*. Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC '90. pp. 416–426. doi:10.1145/100216.100272. ISBN 0897913612.

[8] J. Feigenbaum and L. Fortnow. *On the random-self-reducibility of complete sets*. In Structure in Complexity Theory Conference, pages 124–132, 1991

[9] O. Goldreich, S. Goldwasser and S. Micali, *How to Construct Random Functions*, JACM 33(4), 1986, pp. 792–807.

[10] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Advances in Cryptology - CRYPTO'84, Lecture Notes in Computer Science, Springer, 1985, pp. 47–53.

[11] Mehlhorn, Kurn and Sun, He. (2014). *Zero Knowledge Proofs*. Great Ideas in Theoretical Computer Science, Saarland University. http://resources.mpi-inf.mpg.de/departments/d1/teaching/ss14/gitcs/notes6.pdf

[12] *"Proof on knowledge"*, Wikipedia. Available at: https://en.wikipedia.org/wiki/Proof_of_knowledge (Accessed: August 6, 2020).