(الف) چگونگی محاسبه‌ی $[x \bmod p]$ در زمان $O(polylog(q).\sqrt{p})$ را نشان دهید.

راهنمایی: معادله‌ی

$$(g^{p^{e-1}})^{x_0} = h^{p^{e-1}}$$

را حل کنید و از ایده‌های موجود در الگوریتم پولیگ-هلمن استفاده کنید.

(ب) فرض کنید $x = x_0 + x_1 \cdot p + \cdots + x_{e-1} \cdot p^{e-1}$ با $0 \le x_i < p$ در گام قبلی، x_0 را تعیین کردیم. نشان دهید که چگونه می‌توان در زمان $polylog(q)$ یک مقدار h_1 را محاسبه نمود به صورتی که $h_1 = (g^p)^{x_1 + x_2 \cdot p + \cdots + x_{e-1} \cdot p^{e-2}}$ باشد.

(ج) از بازگشت برای به دست آوردن زمان اجرای ادعاشده برای مسئله‌ی اولیه، استفاده کنید. (توجه کنید که $e = O(\log q)$ است.)

۹-۵: فرض کنید q دارای تجزیه‌ی اول $q = \prod_{i=1}^{k} p_i^{e_i}$ است. با استفاده از نتیجه‌ی به‌دست‌آمده از تمرین قبلی، یک تغییر برای الگوریتم پولیگ-هلمن نشان دهید که مسئله‌ی لگاریتم گسسته را در یک گروه با مرتبه‌ی q در زمان $O(\text{polylog}(q) \cdot \sum_{i=1}^{k} e_i e_i \sqrt{p_i}) = O(polylog(q) \cdot max_i\{\sqrt{p_i}\})$ حل می‌کند.

۹-۶: شبه کدی برای الگوریتم فضای کوچک برای محاسبه‌ی لگاریتم‌های گسسته‌ی توصیف‌شده در بخش ۹-۲-۳ ارائه کرده و یک تحلیل تجربی بر روی احتمال موفقیت آن، ارائه نمایید.

فصل دهم: مدیریت کلید و انقلاب کلید عمومی

۱۰-۱- توزیع کلید و مدیریت کلید

در فصول اول تا هفتم مشاهده کردیم که چگونه رمزنگاری کلید خصوصی را می‌توان برای تضمین محرمانگی و یکپارچگی برای دو طرف در حال ارتباط روی یک کانال ناامن مورداستفاده قرار داد با این فرض که دو طرف دارای یک کلید محرمانه‌ی مشترک هستند. بااین‌حال، سؤالی که از فصل اول تاکنون به آن نپرداخته‌ایم، این است:

اصلاً چگونه طرفین می‌توانند در ابتدا یک کلید محرمانه را به اشتراک بگذارند؟

مشخص است که کلید را نمی‌توان به‌سادگی روی یک کانال ارتباطی عمومی ارسال نمود به این دلیل که یک متخاصم شنودگر در آن حالت قادر به مشاهده‌ی آن در طول مسیر خواهد بود. یک مکانیسم دیگر را باید به‌جای این کار مورد استفاده قرار داد.

در بعضی از مواقع، طرفین ممکن است دارای دسترسی به یک کانال امن باشند که بتوانند برای تسهیم یک کلید محرمانه به شکل قابل‌اطمینان از آن بهره ببرند. یک مثال معمول ازاین‌دست زمانی است که دو طرف در یک زمان در یک مکان قرار می‌گیرند که در آنجا می‌توانند یک کلید را تسهیم کنند. همچنین به‌جای این کار، طرفین می‌توانند یک حامل مورد اعتماد را به‌عنوان یک کانال امن مورد استفاده قرار دهند. تأکید می‌کنیم که این واقعیت که طرفین می‌توانند یک کلید را تسهیم کنند و بنابراین باید دارای دسترسی به یک کانال امن در یک نقطه‌ی زمانی باشند- باعث نمی‌شود که رمزنگاری کلید خصوصی، بی‌فایده شود: در مثال اول، طرفین دارای یک کانال امن در نقطه‌ای از زمان ولی نه به‌صورت بی‌نهایت هستند؛ در مثال دوم، استفاده از یک کانال امن ممکن است کندتر و پرهزینه‌تر از برقراری ارتباط بر روی یک کانال ناامن باشد.

رویکردهای بالا برای تسهیم کلیدها در بسترهای دولتی، دیپلماتیک و نظامی مورد استفاده قرارگرفته‌اند. برای مثال، «خط تلفن قرمز» متصل‌کننده‌ی مسکو و واشنگتن در دهه‌ی ۱۹۶۰ یا

استفاده از یک کلید یک‌بارمصرف رمز شده بود و کلیدها توسط حاملانی تسهیم می‌شدند که از یک کشور به کشور دیگر پرواز می‌کردند درحالی‌که کیف‌های پر از برگه‌های چاپ‌شده با خود حمل می‌کردند. چنین رویکردهایی را می‌توان در شرکت‌ها هم استفاده کرد مثلاً برای تعیین یک کلید مشترک بین یک پایگاه داده‌ی مرکزی و یک کارمند جدید در اولین روز کاری وی. (در بخش بعدی به این مثال بازمی‌گردیم.)

بااین‌حال، تکیه به یک کانال امن برای توزیع کلیدها در بسیاری از دیگر موقعیت‌ها، بیهوده است. برای مثال، یک شرکت عظیم چندملیتی را در نظر بگیرید که در آن، هر زوجی از کارمندان ممکن است نیازمند به توانایی برقراری ارتباط امن باشند و ارتباط آن‌ها باید در برابر دیگر کارمندان هم حفاظت شود.

دست‌کم، اینکه هر زوجی از کارمندان باهم ملاقات نمایند تا بتوانند یک کلید را به شکل امن تسهیم کنند، خیلی راحت نخواهد بود؛ برای کارمندانی که در شهرهای متفاوت کار می‌کنند، این امر حتی ممکن است غیرممکن باشد. حتی اگر مجموعه‌ی کنونی کارمند بتوانند به شیوه‌ای کلیدها را با یکدیگر تسهیم کنند، تسهیم کلیدها با کارمندان جدیدی که بعد از تسهیم اولیه به شرکت می‌پیوندند، غیرعملی خواهد بود.

با فرض اینکه این N کارمند به شیوه‌ای قادر به تسهیم امن کلیدها با یکدیگر هستند، یک نقطه‌ضعف چشم‌گیر دیگر، آن است که هر کارمند مجبور به مدیریت و ذخیره‌ی $N-1$ کلید رمز خواهد بود (یعنی یک کلید برای هرکدام از دیگر کارمندان). درواقع، این امر ممکن است تخمین بسیار پایینی برای تعداد کلیدهای ذخیره‌شده توسط هر کاربر باشد چراکه هر کارمند ممکن است همچنین نیازمند کلیدهایی برای برقراری ارتباط امن با منابع از راه دور از قبیل پایگاه‌های داده، چاپگرها و غیره باشد. تکثیر این‌همه کلید رمز یک مشکل لجستیکی بسیار برجسته است. به‌علاوه، همه‌ی این کلیدها باید به‌صورت امن ذخیره شوند. هرچه تعداد کلیدها بیشتر باشد، حفاظت از آن‌ها دشوارتر بوده و احتمال اینکه تعدادی از کلیدها توسط یک مهاجم سرقت شوند، بیشتر خواهد شد. سامانه‌های رایانه‌ای معمولاً دچار ویروس، کرم و دیگر اشکال نرم‌افزارهای مخرب می‌شوند که می‌توانند کلیدهای رمز را سرقت و به شکل مخفیانه و از طریق شبکه آن‌ها را برای مهاجم ارسال کنند. بنابراین، ذخیره‌ی کلیدها بر روی رایانه‌های شخصی کارکنان همواره یک راه‌حل امن نیست.

اگر بخواهیم شفاف صحبت کنیم، خطر بالقوه‌ی کشف کلیدهای رمز همواره باعث نگرانی است بدون توجه به تعداد کلیدهایی که در اختیار هر یک از طرفین است. بااین‌حال، هنگامی‌که تنها تعداد اندکی کلید باید ذخیره شوند، راه‌حل‌های خوبی برای مدیریت این تهدید وجود دارد. امروزه، یک راه‌حل معمول عبارت است از ذخیره‌ی کلیدها بر روی سخت‌افزارهای امنی از قبیل کارت‌های هوشمند. یک کارت هوشمند می‌تواند محاسبات رمزنگاری را با استفاده از کلیدهای رمز ذخیره‌شده انجام دهد که باعث تضمین این امر می‌شود که این کلیدها هیچ‌گاه وارد رایانه‌های شخصی کاربران نخواهند شد.

اگر کارت‌های هوشمند به‌درستی طراحی شده باشند می‌توانند مقاومت بسیار بالاتری علیه حملات در مقایسه با یک رایانه‌ی شخصی داشته باشند- برای مثال، معمولاً این کارت‌ها قابل ویروسی شدن نیستند- و بنابراین، روش خوبی برای حفاظت از کلیدهای رمز کاربران به دست می‌دهند. متأسفانه، کارت‌های هوشمند معمولاً حافظه‌ی بسیار محدودی دارند و بنابراین نمی‌توانند صدها (یا هزاران) کلید را ذخیره کنند.

همه‌ی نگرانی‌های بیان‌شده در بالا را می‌توان در سازمان‌های «بسته» که شامل یک جمعیت تعریف‌شده از کاربران هستند و همه‌ی آن‌ها مایل به تبعیت از سیاست‌های یکسان برای توزیع و ذخیره‌ی کلیدها هستند- در اصل و شاید هم در عمل- حل نمود. بااین‌حال، این روش‌ها در «سیستم‌های باز» شکست می‌خورند که در آن‌ها کاربران دارای تعاملات مقطعی هستند، نمی‌توانند ملاقات فیزیکی ترتیب دهند و ممکن است حتی از وجود هم بی‌خبر باشند تا زمانی که بخواهند باهم ارتباط برقرار کنند. این وضعیت درواقع یک وضعیت معمول‌تر نسبت به چیزی است که تصور می‌شود: استفاده از رمزگذاری برای ارسال اطلاعات کارت اعتباری به یک فروشنده‌ی اینترنتی را در نظر بگیرید که تاکنون از وی هیچ‌چیزی نخریده‌اید یا ارسال ایمیل به کسی را در نظر بگیرید که هیچ‌گاه به‌صورت حضوری او را ملاقات نکرده‌اید. در چنین وضعیت‌هایی، رمزنگاری کلید خصوصی به‌تنهایی یک راه‌حل محسوب نمی‌شود و ما باید به دنبال راه‌حل‌های کافی بگردیم.

به‌صورت خلاصه، حداقل سه مشکل متمایز مربوط به استفاده از رمزنگاری کلید خصوصی وجود دارد. اولین مشکل مربوط به توزیع کلید است؛ دومین مشکل مربوط به ذخیره و مدیریت تعداد زیادی کلید رمز است؛ سومین مشکل مربوط به عدم امکان اعمال رمزنگاری کلید خصوصی برای سیستم‌های باز است.

۱۰-۲- یک راه‌حل جزئی: مراکز توزیع کلید

یک روش برای حل بعضی از نگرانی‌های بیان‌شده در بخش قبلی، استفاده از یک مرکز توزیع کلید (*KDC*) برای تعیین کلیدهای تسهیمی است. یک‌بار دیگر یک شرکت عظیم را در نظر بگیرید که در آن، هر دو کارمند باید قادر به برقراری ارتباط به‌صورت امن باشند. در چنین وضعیتی، می‌توانیم از این واقعیت بهره ببریم که همه‌ی کارمندان ممکن است به یک نهاد- مثلاً، مدیر سیستم- حداقل در رابطه با امنیت اطلاعات کاری، اعتماد داشته باشند. این نهاد امین می‌تواند به‌عنوان یک *KDC* عمل کرده و به همه‌ی کارمندان کمک کند تا زوج کلیدها را تسهیم نمایند.

هنگامی‌که یک کارمند جدید وارد شرکت می‌شود، *KDC* می‌تواند یک کلید را با آن کارمند (به‌صورت حضوری، در یک مکان امن) به‌عنوان بخشی از روز اول کاری آن کارمند، تسهیم نماید. درعین‌حال، *KDC* می‌تواند همچنین کلیدهای تسهیم شده را بین آن کارمند و تمام کارمندان موجود شرکت، توزیع نماید. یعنی، هنگامی‌که کارمند *i*ام وارد شرکت می‌شود، *KDC* می‌تواند (علاوه بر تسهیم یک

کلید بین خودش و این کارمند جدید) $i-1$ کلید $k_{i-1}, \ldots, k_1$ را تولید کند، این کلیدها را به کارمند جدید بدهد و سپس کلید k_j را به کارمند کنونی jام بدهد به این صورت که آن را با استفاده از کلیدی که آن کارمند از قبل با KDC تسهیم کرده، رمزگذاری کند. در ادامه، کارمند جدید یک کلید را با هرکدام زا دیگر کارمندان (و همین طور با KDC) تسهیم می‌کند.

یک رویکرد بهتر، که نیاز به ذخیره و مدیریت چندین کلید توسط کارمندان را از بین می‌برد، عبارت است از استفاده از KDC به‌صورت آنلاین برای تولید کلیدها «بر اساس نیاز» هرگاه که دو کارمند بخواهند به‌صورت امن باهم ارتباط برقرار کنند. به‌مانند قبل، KDC یک کلید (متفاوت) را باهم کارمند تسهیم می‌کند، چیزی که می‌توان به‌صورت امن در اولین روز کاری یک کارمند انجام داد. فرض کنید KDC کلید k_A را با آلیس (یکی از کارمندان) و k_B را با باب (کارمند دیگر) تسهیم می‌کند. مدتی بعد، هنگامی‌که آلیس می‌خواهد به‌صورت امن با باب ارتباط برقرار کند، می‌تواند پیام «من، آلیس می‌خواهم با باب صحبت کنم» (I, Alice, want to talk to Bob) را برای KDC ارسال کند. (در صورت دلخواه، این پیام را می‌توان با استفاده از کلید تسهیم شده توسط آلیس و KDC، احراز اصالت نمود.) در ادامه، KDC یک کلید تصادفی جدید – با نام کلید «نشست»- را انتخاب کرده و این کلید k رمز شده با استفاده از k_A را برای آلیس و رمز شده با استفاده از k_B را برای باب می‌فرستد. (این پروتکل بیش‌ازحد ساده است که در عمل مورد استفاده قرار گیرد؛ ن.ک. بحث ذیل.) هنگامی‌که هم آلیس و هم باب، این کلید نشست را بازیابی کردند، می‌توانند از آن برای برقراری ارتباط امن استفاده کنند. وقتی صحبت کردن آن‌ها تمام شد، آن‌ها می‌توانند (و باید) کلید نشست را حذف کنند به این دلیل که می‌توانند همواره دوباره با KDC تماس بگیرند اگر بخواهند بعداً باهم ارتباط دیگری داشته باشند.

مزیت‌های این رویکرد را در نظر بگیرید:

۱) هر کارمند تنها باید یک کلید رمز بلندمدت (یعنی، کلیدی که با KDC تسهیم می‌کنند) را ذخیره کند. کارمندان همچنان نیازمند مدیریت و ذخیره‌ی کلیدهای نشست هستند ولی این کلیدها، کلیدهای کوتاه‌مدتی هستند که بعد از اتمام نشست ارتباطی، پاک می‌شوند.
KDC باید تعداد زیادی کلید بلندمدت را ذخیره کند. بااین‌حال، KDC را می‌توان در یک مکان امن قرار داد و بالاترین حد حفاظت ممکن علیه حملات شبکه‌ای را برای آن در نظر گرفت.

۲) هنگامی‌که یک کارمند جدید وارد سازمان می‌شود، تنها کاری که باید انجام شود ایجاد یک کلید بین این کارمند و KDC است. نیازی نیست که هیچ کارمند دیگری مجموعه کلیدهای در اختیار خود را به‌روزرسانی کند.

بنابراین، KDCها می‌توانند دو مورد از مشکلات مشاهده‌شده در ارتباط با رمزنگاری کلید خصوصی را رفع کنند: می‌توانند توزیع کلید را ساده کنند (چراکه تنها یک کلید جدید باید هنگام ورود یک

کارمند جدید، تسهیم شود و منطقی است که یک کانال امن بین KDC و آن کارمند در اولین روز کار وی را فرض کنیم) و می‌توانند پیچیدگی ذخیره‌ی کلید را کاهش دهند (چراکه هر کارمند تنها نیازمند ذخیره‌ی یک کلید یکتا است). KDCها تأثیر زیادی در عملی سازی رمزنگاری کلید خصوصی در سازمان‌های بزرگی دارند که در آن‌ها یک نهاد یکتا وجود دارد که مورد اطمینان و اعتماد همه است.

بااین‌حال، تعدادی نقطه‌ضعف در تکیه کردن به KDCها وجود دارد:

۱) یک حمله‌ی موفق علیه KDC منجر به شکستن کامل سیستم می‌شود: مهاجم می‌تواند تمام کلیدها را به دست آورده و در ادامه تمام ترافیک شبکه را مورد شنود قرار دهد. این امر باعث می‌شود که KDC یک هدف باارزش باشد. توجه کنید که حتی اگر KDC به‌خوبی علیه حملات بیرونی، حفاظت شده باشد، همواره امکان حملات داخلی توسط کارمندانی که دارای دسترسی به KDC هستند (برای مثال، مدیر IT) وجود خواهد داشت.

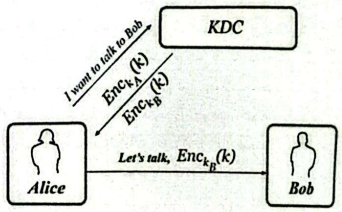
۲) KDC یک نقطه شکست یکتا است: اگر KDC از کار بیفتد، ارتباط امن موقتاً غیرممکن خواهد شد. اگر کارمندان همواره در تماس با KDC و ارائه‌ی درخواست برای ایجاد کلیدهای نشست باشند، بار وارد بر KDC می‌توان بسیار بالا برود که این امر باعث افزایش احتمال شکست آن یا کندی پاسخ‌دهی آن خواهد شد.

یک راه‌حل ساده برای مسئله‌ی دوم عبارت است از تکثیر KDC. این روش عمل می‌کند (و در عمل هم انجام می‌شود) ولی به این معنا نیز هستند که اکنون نقاط حمله‌ی جدیدی برای حمله به سیستم وجود دارد. افزودن تعداد بیشتری KDC همچنین باعث می‌شود افزودن کارمندان جدید، دشوارتر شود چراکه به‌روزرسانی‌ها باید به‌صورت امن برای همه‌ی KDCها، تکثیر شوند.

پروتکل‌ها برای توزیع کلید با استفاده از یک KDC

تعدادی پروتکل در ادبیات برای توزیع کلید امن با استفاده از یک KDC وجود دارد. علی‌الخصوص به پروتکل نیدهام-شرودر اشاره می‌کنیم که هسته‌ی اصلی کربروس را تشکیل می‌دهد که یکی از خدمات مهم و مورداستفاده‌ی گسترده برای انجام احراز اصالت و پشتیبانی از ارتباط امن است. (کربروس در بسیاری از دانشگاه‌ها و شرکت‌ها مورد استفاده قرار گرفته و مکانیسم پیش‌فرض برای پشتیبانی از احراز اصالت و ارتباط شبکه‌ای امن در ویندوز و بسیاری از سیستم‌های یونیکس است.) ما تنها یکی از ویژگی‌های این پروتکل را موردبحث قرار می‌دهیم. وقتی‌که آلیس با KDC تماس می‌گیرد و می‌خواهد با باب ارتباط برقرار کند، KDC کلید نشست رمز شده را برای آلیس و باب به صورتی که قبلاً توصیف کردیم، ارسال نمی‌کند. به‌جای این کار، KDC علاوه بر کلید نشست رمز شده با کلید باب، کلید نشست رمز شده با کلید آلیس را هم برای آلیس ارسال می‌کند. سپس، آلیس متن رمز ثانویه را به شکل نشان داده شده در تصویر ۱۰-۱ برای باب ارسال می‌کند. متن رمز ثانویه را گاهی

یک «بلیت» می‌نامند و می‌توان آن را به‌عنوان یک اعتبارنامه در نظر گرفت که به آلیس اجازه می‌دهد با باب صحبت کند (و به باب اجازه می‌دهد که اطمینان حاصل کند که در حال صحبت با آلیس است). درواقع، هرچند این نکته را در بحث خود مورد تأکید قرار نداده‌ایم، یک رویکرد مبتنی بر KDC می‌تواند روش مفیدی را برای اجرای احراز اصالت نیز فراهم کند. همچنین توجه کنید که نیازی نیست آلیس و باب هر دو کاربر باشند؛ آلیس ممکن است یک کاربر باشد و باب می‌توان یک منبع از قبیل یک کارگذار، یک دیسک از راه دور یا یک چاپگر باشد.



تصویر ۱۰-۱: یک قالب عمومی برای پروتکل‌های توزیع کلید

این پروتکل به این شیوه طراحی شده است تا بار وارد بر KDC را کاهش دهد. در این پروتکل توصیف‌شده، نیازی نیست که KDC یک ارتباط دوم را با باب برقرار کند و زمانی که آلیس پروتکل را آغاز می‌کند، نیازی نیست KDC نگران باشد که آیا باب آنلاین هست یا خیر. به‌علاوه، اگر آلیس بلیت (و کپی خود از کلید نشست) را حفظ نماید، آنگاه می‌تواند ارتباط امن با باب را با ارسال دوباره‌ی بلیت به باب، از نو آغاز نماید بدون اینکه نیازی به دخالت KDC وجود داشته باشد. (در عمل، بلیت‌ها منقضی می‌شوند و درنهایت باید تجدید شوند. ولی یک نشست را می‌توان طی یک دوره‌ی زمانی قابل‌قبول، از نو ایجاد نمود.)

در پایان اشاره می‌کنیم که در عمل، کلیدی که آلیس با KDC تسهیم می‌کند ممکن است یک گذرواژه‌ی کوتاه باشد که به‌سادگی قابل حفظ کردن است. در این حالت، بسیاری مشکلات امنیتی دیگری به وجود می‌آیند که باید حل شوند. همچنین به‌صورت ضمنی مهاجمی را فرض کرده‌ایم که تنها به‌صورت غیرفعال دست به شنود می‌زند و نه مهاجمی که به‌صورت فعال سعی می‌کند در پروتکل مداخله کند. خواننده‌ی علاقه‌مند می‌تواند به مراجع انتهای فصل مراجعه کند تا اطلاعات بیشتری در خصوص چگونگی حل این مسائل به دست آورد.

۱۰-۳- تبادل کلید و پروتکل دیفی-هلمن

KDCها و پروتکل‌هایی از قبیل کربروس به‌صورت معمول در عمل مورد استفاده قرار می‌گیرند. ولی این رویکردها برای مسئله‌ی توزیع کلید همچنان نیازمند یک کانال «خصوصی» و «احراز اصالت شده» در نقطه‌ای از زمان هستند که می‌تواند برای تسهیم کلیدها استفاده شود. (علی‌الخصوص، وجود چنین کانالی بین KDC و کارمند را در روز اول کاری وی، فرض کردیم.) بنابراین، آن‌ها همچنان قادر به حل مشکل توزیع کلید در سیستم‌های باز از قبیل اینترنت نیستند که در آن‌ها، ممکن است هیچ کانال خصوصی بین دو طرفی که می‌خواهند ارتباط برقرار کنند، وجود نداشته باشد.

برای رسیدن به ارتباط خصوصی بدون برقراری ارتباط بر روی یک کانال خصوصی، یک رویکرد کاملاً متفاوت موردنیاز است. در سال ۱۹۷۶، رایتفیلد دیفی و مارتین هلمن مقاله‌ای با عنوانی ساده به نام «جهت‌گیری‌های جدید در رمزنگاری» منتشر کردند. در این مطالعه، آن‌ها بیان کردند که در جهان اطلب شاهد «عدم تقارن» هستیم؛ علی‌الخصوص، بعضی فعالیت‌های خاص وجود دارند که می‌توان به‌سادگی آن‌ها را انجام داد ولی به‌سادگی قابل وارون سازی نیستند. برای مثال، قفل‌های کلیدی را با یک کلید می‌توان قفل کرد (یعنی به‌سادگی) ولی نمی‌توان آن‌ها را دوباره باز کرد. به‌عنوان مثالی واضح‌تر، به‌سادگی می‌توان یک ظرف شیشه‌ای را شکست ولی چسباندن آن به هم بسیار دشوار است. (لحاظ الگوریتمی (که اهمیت بیشتری برای اهداف ما دارد)، به‌سادگی می‌توان دو عدد اول بزرگ را در هم ضرب کرد ولی بازیابی آن اعداد اول از روی حاصل‌ضرب آن‌ها، دشوار است. (این دقیقاً همان مسئله‌ی تجزیه است که در فصول قبلی به آن پرداختیم.) دیفی و هلمن دریافتند که چنین پدیده‌هایی را می‌توان برای به دست آوردن پروتکل‌های تعاملی برای «تبادل کلید امن» مورد استفاده قرار داد که به طرفین اجازه می‌دهند یک کلید رمز را از طریق ارتباط بر روی یک کانال عمومی، تسهیم کنند به این صورت که طرفین عملیاتی را اجرا نماید که بتوانند آن را وارون کنند ولی یک شنودگر قادر به وارون کردن آن‌ها نباشد.

وجود پروتکل‌های تبادل کلید امن بسیار شگفت‌انگیز است. این امر به این معنا است که شما و دوستان می‌توانید در مورد یک کلید به توافق برسید به این صورت که در یک اتاق با فریاد آن را به هم برسانید (و مقداری محاسبات محلی انجام دهید)؛ رمز برای هیچ‌کس دیگری معلوم نخواهد بود حتی اگر به تمام حرف‌های شما گوش داده باشند. درواقع، تا سال ۱۹۷۶، باور عمومی بر آن بود که ارتباط امن را نمی‌توان به دست آورد مگر اول نوعی اطلاعات محرمانه را با استفاده از یک کانال خصوصی، تسهیم نمود.

تأثیر مقاله‌ی دیفی و هلمن بسیار عظیم بود. علاوه بر معرفی یک شیوه‌ی بنیادی جدید برای در نظر گرفتن رمزنگاری، این مقاله یکی از اولین گام‌ها برای انتقال رمزنگاری به بیرون از حوزه‌ی خصوصی و واردکردن آن به درون حوزه‌ی عمومی بود. ما دو بند اول مقاله‌ی آن‌ها را نقل می‌کنیم:

«امروز در لبه‌ی یک انقلاب در رمزنگاری ایستاده‌ایم. توسعه‌ی سخت‌افزارهای دیجیتالی ارزان باعث آزاد شدن آن از دست محدودیت‌های طراحی محاسبات مکانیکی شده و هزینه‌ی دستگاه‌های رمزنگاری سطح بالا را به حدی کاهش داده است که می‌توان آن‌ها را در کاربردهای تجاری از قبیل پرداخت مبالغ از راه دور و پایانه‌های رایانه‌ای مورد استفاده قرار داد.

در ادامه، چنین کاربردهایی باعث ایجاد نیاز به انواع جدید سیستم‌های رمزنگاری شده‌اند که ضرورت وجود کانال‌های توزیع کلید امن را به حداقل می‌رسانند... درعین‌حال، پیشرفت‌های نظری در نظریه‌ی اطلاعات و علوم رایانه نشان‌دهنده‌ی امید برای ارائه‌ی سامانه‌های رمزنگاری با امنیت قابل‌اثبات هستند که این هنر باستانی را به یک علم بدل می‌کند.»

دیفی و هلمن در حال مبالغه نبودند و انقلابی که درباره‌ی آن صحبت می‌کردند تا حد زیادی ناشی از مطالعه‌ی آن‌ها بود.

در این بخش، ما پروتکل تبادل کلید دیفی-هلمن را ارائه می‌کنیم. امنیت آن علیه متخاصمین شنودگر یا به شکل برابر تحت این فرض اثبات می‌کنیم که طرفین بر روی یک کانال عمومی ولی «احراز اصالت شده» ارتباط برقرار می‌کنند (بنابراین مهاجم نمی‌تواند در ارتباط آن‌ها مداخله نماید). امنیت علیه یک متخاصم شنودگر یک تضمین نسبتاً ضعیف است و در عمل، پروتکل‌های تبادل کلید باید در مفاهیم قوی‌تر امنیت صدق نمایند که فراتر از گستره‌ی کنونی ما است. (به‌علاوه، ما در اینجا علاقه‌مند به موقعیت‌های هستیم که در آن‌ها، طرفین ارتباط هیچ اطلاعات تسهیم شده‌ی قبلی ندارند، که در این حالت، هیچ کاری نمی‌تواند برای جلوگیری از متخاصم برای جعل هویت یکی از طرفین، انجام داد. بعداً به این نکته بازمی‌گردیم.)

موقعیت و تعریف امنیت

ما موقعیتی با دو طرف را در نظر می‌گیریم- که به‌صورت سنتی آلیس و باب نامیده می‌شوند- که یک پروتکل احتمالاتی Π را اجرا می‌کنند تا یک کلید رمز تسهیم شده را تولید کنند؛ Π را می‌توان به‌عنوان مجموعه‌ای از دستورالعمل‌ها برای آلیس و باب در پروتکل در نظر گرفت. آلیس و باب با در نظر گرفتن پارامتر امنیتی 1^n شروع می‌کنند؛ سپس Π را با استفاده از بیت‌های تصادفی (مستقل) اجرا می‌کنند. در انتهای پروتکل، آلیس و باب، کلیدهای $k_A, k_B \in \{0,1\}^n$ را به ترتیب، اعلام می‌کنند. الزام اساسی درستی آن است که $k_A = k_B$ باشد. ازآنجاکه تنها با پروتکل‌هایی سروکار داریم که در این شرط صدق می‌کنند، تنها از تک کلید $k = k_A = k_B$ صحبت می‌کنیم که از طریق اجرای صادق Π تولید می‌شود.

اکنون به تعریف امنیت می‌پردازیم. به شکل شهودی، یک پروتکل تبادل کلید، امن است اگر کلید خروجی توسط آلیس و باب برای یک متخاصم شنودگر، کاملاً غیرقابل حدس زدن باشد. این امر به‌صورت رسمی با این الزام تعریف می‌شود که متخاصمی که بر یک اجرای پروتکل، شنود کرده است

باید قادر به تشخیص دادن کلید k تولیدشده توسط آن اجرا (و اکنون تسهیم شده بین آلیس و باب) از یک کلید یکنواخت با طول n نباشد. این امر بسیار قوی‌تر از این الزام ساده است که متخاصم قادر به محاسبه‌ی دقیق k نباشد و این مفهوم قوی‌تر ضروری خواهد بود اگر طرفین در ادامه از k برای یک کاربرد رمزنگاری مفروض، استفاده نمایند (برای مثال، به‌عنوان کلیدی برای یک طرح رمزنگاری کلید خصوصی).

برای ارایه یک تعریف رسمی از مطلب بالا، فرض کنید Π یک پروتکل تبادل کلید، $\mathcal{A}$ یک متخاصم و n پارامتر امنیتی باشد. آزمایش زیر را داریم:

آزمایش تبادل کلید $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$:

۱) دو طرف مالک 1^n، پروتکل Π را اجرا می‌کنند. این امر منجر به یک رونوشت $trans$ شامل تمام پیام‌های ارسال‌شده توسط طرفین و یک کلید k خروجی توسط هرکدام از طرفین می‌شود.

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

۳) $\mathcal{A}$ مقادیر $trans$ و $\hat{k}$ را دریافت کرده و یک بیت b' را به‌عنوان خروجی ارائه می‌کند.

۴) خروجی آزمایش برابر با ۱ است اگر $b' = b$ و در غیر این صورت برابر با ۰ است. (در حالتی که $\mathrm{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1$، می‌گوییم $\mathcal{A}$ موفق شده است.)

$\mathcal{A}$ مقدار $trans$ را دریافت می‌کند تا این واقعیت را نشان دهیم که $\mathcal{A}$ بر روی کل اجرای پروتکل، شنود می‌کند و بنابراین تمام پیام‌های مبادله شده توسط طرفین را می‌بیند. در جهان واقعی، $\mathcal{A}$ هیچ کلیدی دریافت نخواهد کرد؛ در این آزمایش، متخاصم، $\hat{k}$ را تنها به‌عنوان روشی برای تعریف معنای اینکه $\mathcal{A}$ امنیت Π را «شکسته است» دریافت می‌کند. یعنی، متخاصم موفق به شکستن Π خواهد شد اگر بتواند به‌درستی این امر را تعیین کند که آیا کلید $\hat{k}$ یک کلید واقعی منطبق با اجرای مفروض پروتکل هست یا خیر یا اینکه آیا $\hat{k}$ یک کلید یکنواخت مستقل از رونوشت هست یا خیر. به شکل مورد انتظار، می‌گوییم Π امن است اگر متخاصم با احتمالی که حداکثر به شکل ناچیز از ۱/۲ بهتر است، بتواند موفق شود. یعنی:

تعریف ۱۰-۱

یک پروتکل تبادل کلید Π در حضور یک شنودگر، امن است اگر برای تمام متخاصمین زمان-چندجمله ای احتمالاتی $\mathcal{A}$، یک تابع ناچیز $negl$ وجود داشته باشد به صورتی که

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

هدف یک پروتکل تبادل کلید، تقریباً همواره عبارت است از تولید یک کلید تسهیم شده‌ی k که توسط طرفین برای یک هدف رمزنگاری دیگر مورد استفاده قرار گیرد مثلاً برای رمزگذاری و احراز اصالت ارتباط بعدی آن‌ها با استفاده از مثلاً یک طرح رمزگذاری احراز اصالت شده. به شکل شهودی، استفاده از یک کلید تسهیم شده‌ی تولیدشده توسط یک پروتکل تبادل کلید امن باید «به‌خوبی» استفاده از یک کلید تسهیم شده بر روی یک کانال خصوصی باشد. اثبات این امر به‌صورت رسمی، امکان‌پذیر است؛ ن.ک. تمرین ۱۰-۱.

پروتکل تبادل کلید دیفی-هلمن

اکنون پروتکل تبادل کلیدی را توصیف می‌کنیم که در مقاله‌ی اصلی دیفی و هلمن، آورده شده است (هرچند در مقایسه با اینجا، بیان آن‌ها کمتر رسمی بود). فرض کنید $\mathcal{G}$ یک الگوریتم زمان-چندجمله ای احتمالاتی باشد که با دریافت ورودی 1^n، توصیفی از یک گروه دوری $\mathbb{G}$، مرتبه‌ی آن، q، (با $q \| = n$) و یک مولد $g \in \mathbb{G}$ را به‌عنوان خروجی ارائه دهد. (ن.ک. بخش ۸-۳-۲.) پروتکل تبادل کلید دیفی-هلمن به‌صورت رسمی به شکل ساختار ۱۰-۲ توصیف شده و در تصویر ۱۰-۲ نشان داده شده است.

پروتکل تبادل کلید دیفی-هلمن

ساختار ۱۰-۲

ورودی مشترک: پارامتر امنیتی 1^n

پروتکل:

۱) آلیس $\mathcal{G}(1^n)$ را برای به دست آوردن $(\mathbb{G}, q, g)$ اجرا می‌کند.

۲) آلیس یک $x \in \mathbb{Z}_q$ یکنواخت را انتخاب کرده و $h_A := g^x$ را محاسبه می‌کند.

۳) آلیس، $(\mathbb{G}, q, g, h_A)$ را برای باب ارسال می‌کند.

۴) باب $(\mathbb{G}, q, g, h_A)$ را دریافت می‌کند. وی یک $y \in \mathbb{Z}_q$ یکنواخت را انتخاب کرده و $h_B := g^y$ را محاسبه می‌کند. باب h_B را برای آلیس ارسال کرده و کلید $k_B := h_A^y$ را به‌عنوان خروجی ارائه می‌کند.

۵) آلیس h_B را دریافت کرده و کلید $k_A := h_B^x$ را به‌عنوان خروجی ارائه می‌کند.

در توصیف ما، فرض کرده‌ایم که آلیس $(\mathbb{G}, q, g)$ را تولید کرده و این پارامترها را برای باب به‌عنوان بخشی از اولین پیام خود، ارسال می‌کند. در عمل، این پارامترها، «استانداردسازی شده» هستند و ثابت و معلوم برای هر دو طرف قبل از آغاز پروتکل خواهند بود. در این حالت، تنها نیاز است که آلیس h_A را ارسال کند و نیازی نیست باب منتظر دریافت پیام آلیس قبل از محاسبه و ارسال h_B بماند.

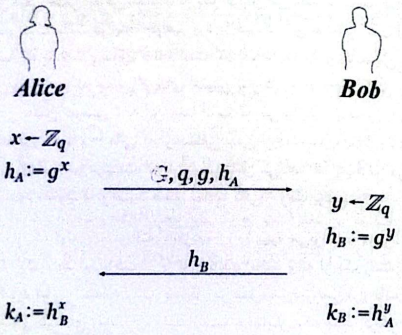
مشاهده‌ی این امر که این پروتکل، صحیح است، دشوار نیست: باب کلید زیر را محاسبه می‌کند

$$k_B = h_A^y = (g^x)^y = g^{xy}$$

و آلیس کلید زیر را محاسبه می‌کند

$$k_A = h_B^x = (g^y)^x = g^{xy}$$

و بنابراین $k_A = k_B$. (خواننده‌ی دقیق متوجه می‌شود که کلید تسهیم شده یک عنصر گروهی است و نه یک رشته‌ی بیتی. بعداً به این نکته بازمی‌گردیم.)



تصویر ۱۰-۲: پروتکل تبادل کلید دیفی-هلمن

دیفی و هلمن، امنیت پروتکل خود را اثبات نکردند؛ درواقع، مفاهیم مناسب (هم چارچوب تعریفی و همین ایده‌ی فرمول‌بندی فرضیات دقیق) هنوز وجود نداشتند. اجازه دهید ببینیم که چه نوع فرضی به‌منظور امن بودن این پروتکل موردنیاز است. یک مشاهده‌ی اولیه، بیان‌شده توسط دیفی و هلمن، این است که یک الزام حداقلی برای امنیت در اینجا عبارت است از اینکه مسئله‌ی لگاریتم گسسته نسبت به $\mathcal{G}$ سخت باشد. اگر سخت نباشد، آنگاه متخاصم دارای رونوشت مفروض (که به‌صورت خاص شامل h_A است) می‌تواند مقدار رمز یکی از طرفین (یعنی x) را محاسبه کرده و سپس به‌سادگی، کلید تسهیم شده را با استفاده از آن مقدار، محاسبه کند. بنابراین، سختی مسئله‌ی لگاریتم گسسته

برای امن بودن این پروتکل، الزامی است. بااین‌حال، این امر کافی نیست به این دلیل که ممکن است شیوه‌های دیگری برای محاسبه‌ی کلید $k_A = k_B$ بدون محاسبه‌ی آشکار x یا y وجود داشته باشد. فرض دیفی-هلمن محاسباتی- که تنها تضمین می‌کند که محاسبه‌ی کل کلید g^{xy} از روی رونوشت، سخت است- نیز کافی نخواهد بود. آنچه توسط تعریف ۱۰-۱ الزام می‌شود آن است که کلید تسهیم شده‌ی g^{xy} باید برای یک متخاصم با داشتن g، g^x و g^y از مقدار یکنواخت، تمایزناپذیر باشد. این امر دقیقاً همان فرض دیفی-هلمن تصمیمی معرفی‌شده در بخش ۸-۳-۲ است.

همان‌طور که خواهیم دید، یک اثبات امنیت برای این پروتکل تقریباً به شکل فوری از فرض دیفی-هلمن تصمیمی، نتیجه می‌شود. این امر نباید تعجب‌آور باشد به این دلیل که فرضیات دیفی-هلمن- بسیار بعد از انتشار مقاله‌ی دیفی و هلمن- به‌عنوان شیوه‌ای برای مجرد نمودن ویژگی‌های مبنای امنیت (حدسی) پروتکل دیفی-هلمن معرفی شدند. با در نظر گرفتن این مطلب، منطقی است بپرسیم که آیا بهره‌ای از تعریف و اثبات امنیت در اینجا خواهیم برد یا خیر. در این نقطه از کتاب، امید بر آن است که قانع شده باشید که پاسخ این سؤال، «بله» است. تعریف دقیق تبادل کلید امن ما را وادار می‌کند در این خصوص فکر کنیم که چه ویژگی‌های امنیتی نیاز داریم؛ مشخص کردن یک فرض دقیق (یعنی، فرض دیفی-هلمن تصمیمی) به این معنا است که می‌توانید این فرض را مستقل از کاربرد خاص موردمطالعه قرار دهیم و -زمانی که از منطقی بودن آن اطمینان حاصل کردیم- دیگر پروتکل‌ها را بر اساس آن، ایجاد کنیم؛ درنهایت، اثبات امنیت نشان می‌دهد که این فرض درواقع برای اینکه این پروتکل، در مفهوم مطلوب ما از امنیت صدق نماید، کافی خواهد بود.

در اثبات ما برای امنیت، از یک گونه‌ی تغییریافته‌ی تعریف ۱۰-۱ بهره می‌بریم که در آن، لازم است که کلید تسهیم شده از یک عنصر یکنواخت از $\mathbb{G}$ تمایزناپذیر باشد و نه از یک رشته‌ی n-بیتی یکنواخت. این تفاوت را باید قبل از استفاده از این پروتکل در عمل، موردبررسی قرار داد- به‌هرحال، اعضای گروه معمولاً به‌عنوان کلیدهای رمزنگاری، مفید نیستند و نمایش یک عنصر گروهی یکنواخت به‌طورکلی معادل با یک رشته‌ی بیتی یکنواخت نخواهد بود- و ما به‌صورت مختصر یکی از روش‌های استاندارد برای انجام این کار را بعد از اثبات، موردبحث قرار خواهیم داد. فعلاً، فرض می‌کنیم که $\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n)$ نشان‌دهنده‌ی یک آزمایش تغییر یافته باشد که در آن، اگر $b = 1$، متخاصم $\mathcal{A}$ انتخاب‌شده به‌صورت یکنواخت از $\mathbb{G}$ را به‌جای یک رشته‌ی n-بیتی یکنواخت، دریافت می‌کند.

قضیه‌ی ۱۰-۳

اگر مسئله‌ی دیفی-هلمن تصمیمی نسبت به $\mathcal{G}$ سخت باشد، آنگاه پروتکل تبادل کلید دیفی-هلمن Π در حضور یک شنودگر (نسبت به آزمایش تغییریافته‌ی $\widetilde{KE}^{eav}_{\mathcal{A},\Pi}$) امن خواهد بود.

اثبات

فرض کنید $\mathcal{A}$ یک متخاصم PPT باشد. ازآنجاکه $\Pr[b = 0] = \Pr[b = 1] = 1/2$، داریم

$$\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}.\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1|b = 0] + \frac{1}{2}.\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1|b = 1].$$

در آزمایش $\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n)$، متخاصم $\mathcal{A}$ مقادیر $(\mathbb{G}, q, g, h_A, h_B, \hat{k})$ را دریافت می‌کند که در آن، $(\mathbb{G}, q, g, h_A, h_B)$ نشان‌دهنده‌ی رونوشت اجرای پروتکل و $\hat{k}$ با کلید واقعی محاسبه‌شده توسط طرفین (اگر $b = 0$) یا یک عنصر گروهی یکنواخت (اگر $b = 1$) خواهد بود. تمایز بین این دو حالت، دقیقاً معادل با حل مسئله‌ی دیفی-هلمن تصمیمی است. یعنی

$$\begin{aligned}
&\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1] \\
&= \frac{1}{2}.\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1|b = 0] + \frac{1}{2}.\Pr[\widetilde{KE}^{eav}_{\mathcal{A},\Pi}(n) = 1|b = 1] \\
&= \frac{1}{2}.\Pr[\mathcal{A}((\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 0] + \frac{1}{2}.\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] \\
&= \frac{1}{2}.(1 - \Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^{xy}) = 1]) + \frac{1}{2}.\Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] \\
&= \frac{1}{2} + \frac{1}{2}.(\Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]) \\
&\leq \frac{1}{2} + \frac{1}{2}.|\Pr[\mathcal{A}(\mathbb{G}, g, q, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]|,
\end{aligned}$$

که در آن، همه‌ی احتمال‌ها بر روی $(\mathbb{G}, q, g)$ خروجی توسط $\mathcal{G}(1^n)$ و انتخاب یکنواخت $x, y, z \in \mathbb{Z}_q$ گرفته‌شده‌اند. (توجه کنید که ازآنجاکه g یک مولد است، g^z یک عنصر یکنواخت از $\mathbb{G}$ است زمانی که z دارای توزیع یکنواخت در $\mathbb{Z}_q$ باشد.) اگر فرض دیفی-هلمن تصمیمی نسبت به $\mathcal{G}$ سخت باشد، این امر دقیقاً به این معنا است که یک تابع ناچیز $negl$ وجود دارد که برای آن

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

نتیجه می‌گیریم که

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

که اثبات را کامل می‌کند. ■

اعضای یکنواخت گروه در مقابل رشته‌های بیتی یکنواخت

قضیه‌ی قبلی نشان می‌دهد که کلید خروجی توسط آلیس و باب در پروتکل دیفی-هلمن، (برای یک شنودگری زمان-چندجمله‌ای) از یک عنصر گروهی یکنواخت، تمایزناپذیر است. به‌منظور استفاده از کلید برای کاربردهای رمزنگاری بعدی-و همین‌طور برقراری تعریف ۱۰-۱- کلید خروجی توسط طرفین باید از یک رشته‌ی بیتی یکنواخت با طول مناسب، تمایزناپذیر باشد. پروتکل دیفی-هلمن را می‌توان تغییر داد تا به این امر رسید به این صورت که طرفین، یک «تابع برگرفتن کلید» مناسب (ر.ک. بخش ۵-۶-۴) را بر عنصر گروهی تسهیم شده‌ی g^{xy} که هرکدام محاسبه می‌کنند، اعمال می‌کنند.

متخاصمین فعال

تاکنون تنها یک متخاصم شنودگر را در نظر گرفته‌ایم. هرچند حملات شنود بسیار معمول هستند (چراکه اجرای آن‌ها ساده است)، به‌هیچ‌وجه تنها حملات ممکن نیستند. حملات فعال، که در آن‌ها، متخاصم پیام‌های خود را به یکی یا هر دو طرفین ارسال می‌کند نیز یک دغدغه هستند و هر پروتکل استفاده‌شده در عمل باید در مقابل چنین حملاتی هم مقاوم باشد. هنگام در نظر گرفتن حملات فعال، مفید است ک هبه‌صورت غیررسمی بین حملات «جعل هویت»، که در آن‌ها متخاصم هویت یکی از طرفین را در هنگام تعامل با طرف دیگر، جعل می‌کند و حملات فرد در میانه، که در آن‌ها هر دو طرف صادق در حال اجرای پروتکل هستند و متخاصم در حال مشاهده و تغییر پیام‌های ارسالی از یک طرف به طرف دیگر است، تمایز قائل شویم. به‌صورت رسمی، امنیت علیه این دو دسته از حملات را تعریف نخواهیم کرد چراکه چنین تعاریفی نسبتاً پیشرفته هستند و نمی‌توان بدون اینکه طرفین نوعی اطلاعات را از قبل باهم تسهیم نمایند، به آن‌ها دست‌یافت. بااین‌حال، شایان توجه است که پروتکل دیفی-هلمن علیه حملات فرد در میانه، کاملاً ناامن است. درواقع، یک متخاصم فرد در میانه می‌تواند به شیوه‌ای عمل کند که آلیس و باب پروتکل را با کلیدهای متفاوت k_A و k_B به پایان ببرند که هر دو آن‌ها برای متخاصم، معلوم هستند بااین‌حال نه آلیس و نه باب نمی‌توانند تشخیص دهند که حمله‌ای اجرا شده است. جزئیات این حمله را به‌عنوان یک تمرین باقی می‌گذاریم.

تبادل کلید دیفی-هلمن در عمل

پروتکل دیفی-هلمن در بنیادی‌ترین شکل خود معمولاً در عمل مورداستفاده قرار نمی‌گیرد به‌این‌علت که در برابر حملات فرد در میانه، ناامن است، همان‌طور که در بالا بحث شد. این امر به‌هیچ‌وجه باعث کاهش اهمیت آن نخواهد شد. پروتکل دیفی-هلمن اولین ابزاری بود که نشان داد روش‌های نامتقارن (و مسائل نظریه‌ی اعداد) را می‌توان برای تقلیل مشکلات توزیع کلید در رمزنگاری مورداستفاده قرار داد. به‌علاوه، پروتکل دیفی-هلمن در هسته‌ی مرکزی پروتکل‌های تبادل کلید استانداردشده‌ای قرار دارد که در برابر حملات فرد در میانه، مقاوم هستند و امروزه به‌صورت گسترده مورداستفاده قرار می‌گیرند. یک مثال برجسته ازاین‌دست، *TLS* است؛ ن.ک. بخش ۱۲-۸.

۱۰-۴- انقلاب کلید عمومی

علاوه بر تبادل کلید، دیفی و هلمن همچنین در اثر برجسته‌ی خود، مفهوم رمزنگاری «کلید عمومی» (یا «نامتقارن») را معرفی کردند. در موقعیت کلید عمومی (برخلاف موقعیت کلید خصوصی که در فصول اول تا هفتم موردمطالعه قراردادیم)، طرفی که می‌خواهد ارتباط امن داشته باشد، یک زوج کلید را تولید می‌کند: یک کلید عمومی که به‌صورت گسترده توزیع می‌شود و یک کلید خصوصی که محرمانه باقی می‌ماند. (این واقعیت که اکنون دو کلید متفاوت وجود دارد چیزی است که باعث می‌شود این طرح، نامتقارن باشد.) بعد از تولید این کلیدها، طرف می‌توان از آن‌ها برای تضمین

محرمانگی برای پیام‌های دریافتی با استفاده از یک طرح رمزگذاری کلید عمومی یا یکپارچگی پیام‌های ارسالی با استفاده زا یک طرح امضای دیجیتال، استفاده کند. (ن.ک. تصویر ۱۰-۳.) ما مقدمه‌ی مختصری بر این پایه‌ها در اینجا ارائه می‌کنیم و جزئیات گسترده‌ی آن‌ها را به ترتیب در فصول یازدهم و دوازدهم موردبحث قرار می‌دهیم.

در یک طرح رمزگذاری کلید عمومی، کلید عمومی تولیدشده توسط یکی از طرفین به‌عنوان یک «کلید رمزگذاری» عمل می‌کند؛ هرشخصی که آن کلید عمومی را بداند می‌تواند از آن برای رمزگذاری پیام‌ها و تولید متون رمز متناظر استفاده کند. کلید خصوصی به‌عنوان یک «کلید رمزگشایی» عمل می‌کند و توسط طرفی که آن را می‌داند برای بازیابی پیام اولیه از روی هر متن رمز تولیدشده توسط کلید عمومی متناظر، استفاده می‌شود. به‌علاوه- و شگفت‌انگیز است که چنین چیزی وجود دارد!- محرمانگی پیام‌های رمز شده ، حفظ خواهد شد حتی علیه متخاصمی که کلید رمزگذاری را می‌داند (ولی کلید رمزگشایی را نمی‌داند). به‌عبارت‌دیگر، کلید رمزگذاری (عمومی) هیچ استفاده‌ای برای مهاجمی ندارد که در تلاش برای رمزگشایی متون رمزی است که با استفاده از آن کلید، رمز شده اند. بنابراین، برای ممکن نمودن ارتباط محرمانه، گیرنده می‌تواند کلید عمومی خود را به یک فرستنده‌ی بالقوه ارسال کند (بدون اینکه نگران متخاصم شنودگرای باشد که کلید عمومی وی را مشاهده می‌کند) یا کلید عمومی خود را بر روی وب‌سایت خود یا در یک پایگاه داده‌ی مرکزی در دید همگان قرار دهد. بنابراین، یک طرح رمزگذاری کلید عمومی، امکان ارتباط خصوصی بدون نکیه به یک کانال خصوصی برای توزیع کلید را فراهم می‌کند[۲۰].

	موقعیت کلید خصوصی	موقعیت کلید عمومی
محرمانگی	رمزگذاری کلید خصوصی	رمزگذاری کلید عمومی
یکپارچگی	کدهای احراز اصالت پیام	طرح‌های امضای دیجیتال

تصویر ۱۰-۳: پایه‌های رمزنگاری در موقعیت‌های کلید خصوصی و کلید عمومی

یک طرح امضای دیجیتال، یک مشابه کلید عمومی از کدهای احراز اصالت پیام (*MAC*ها) است. در اینجا، کلید خصوصی به‌عنوان یک «کلید احراز اصالت» عمل می‌کند (که معمولاً یک «کلید امضا» نامیده می‌شود) که به طرفی که این کلید را می‌داند اجازه می‌دهد تا «برچسب‌های احراز اصالت»

---

[۲۰] بااین‌حال، فعلاً یک کانال «احراز اصالت شده» را فرض می‌کنیم که به فرستنده اجازه می‌دهد یک کپی قانونی از کلید عمومی گیرنده را به دست آورد. در بخش ۱۲-۷ نشان می‌دهیم که چگونه رمزنگاری کلید خصوصی را می‌توان برای کاهش این فرض هم مورداستفاده قرار داد.

(یعنی، امضاها) را برای پیام‌هایی که ارسال می‌کند، تولید کند. کلید عمومی به‌عنوان یک «کلید راست آزمایی» عمل کرده و به هرشخصی که آن را می‌داند اجازه می‌دهد تا امضاهای صادرشده توسط فرستنده را راست آزمایی کند. مشابه با *MAC*ها، یک طرح امضای دیجیتال را می‌توان برای جلوگیری از دست‌کاری نامحسوس یک پیام مورداستفاده قرار داد. بااین‌حال، مشخص شده است که این واقعیت که راست آزمایی می‌تواند توسط هرفردی که کلید عمومی فرستنده را می‌داند، انجام شود دارای نتایج بسیار گسترده‌ای است. علی‌الخصوص، این امر این امکان را به وجود می‌آورد که یک سند امضاشده توسط آلیس را انتخاب کرده و آن را به یک طرف ثالث (مثلاً یک قاضی) به‌عنوان اثبات این امر نشان داد که آلیس واقعاً آن سند را امضا کرده است. این ویژگی را «انکارناپذیری» می‌نامند که کاربردهای گسترده‌ای در تجارت الکترونیک دارد. برای مثال، می‌توان قراردادها را به‌صورت دیجیتال امضا کرد، سفارش‌های خرید الکترونیکی یا تعهدهای پرداخت امضاشده را ارسال نمود و غیره. امضاهای دیجیتال همچنین برای توزیع امن کلیدهای عمومی به‌عنوان بخشی از یک «زیرساخت کلید عمومی» مورداستفاده قرار می‌گیرند که با جزئیات بیشتر در بخش ۱۲-۷ موردبحث قرار می‌گیرد.

در مقاله‌ی خود، دیفی و هلمن، مفهوم رمزنگاری کلید عمومی را ارائه کردند ولی هیچ نمونه ساختاری را پیشنهاد ندادند. یک سال بعد، ران ریوست، آدی شامیر و لن ادلمن، «مسئله‌ی *RSA*» را پیشنهاد داده و اولین طرح‌های رمزگذاری کلید عمومی و امضای دیجیتال را بر اساس سختی این مسئله، ارائه کردند. گونه‌های متفاوت طرح‌های آن‌ها اکنون در میان پایه‌های رمزنگاری قرار دارند که به‌صورت گسترده مورداستفاده قرار می‌گیرند. در سال ۱۹۸۵، طاهر الجمل یک طرح رمزگذاری را ارائه داد که اصولاً یک چرخش جزئی از پروتکل تبادل کلید دیفی-هلمن است و اکنون به‌صورت گسترده مورداستفاده قرار دارد. بنابراین، هرچند دیفی و هلمن موفق به ساختن یک طرح رمزگذاری کلید عمومی (غیرتعاملی) نشدند، ولی به این مقصود بسیار نزدیک شدند.

در پایان، چگونگی حل محدودیت‌های موقعیت کلید خصوصی موردبحث در بخش ۱۰-۱ توسط رمزنگاری کلید عمومی را به‌صورت خلاصه بیان می‌کنیم:

۱) رمزگذاری کلید عمومی اجازه می‌دهد توزیع کلید بر روی کانال‌های عمومی (ولی احراز اصالت شده) انجام شود. این امر می‌تواند توزیع و به‌روزرسانی محتوای کلید را ساده نماید.
۲) رمزنگاری کلید عمومی باعث کاهش نیاز کاربران برای ذخیره‌ی تعداد زیادی کلید رمز می‌شود. بازهم موقعیت یک شرکت بزرگ را در نظر بگیرید که در آن، هر زوجی از کارمندان نیازمند توانایی برقراری ارتباط امن هستند. با استفاده از رمزنگاری کلید عمومی، کافی است هر کارمند تنها یک کلید خصوصی یکتا (متعلق به خود) و کلیدهای عمومی تمامی دیگر کارمندان را ذخیره نماید. نکته‌ی مهم این است که نیازی نیست این کلیدهای عمومی به‌صورت محرمانه ذخیره شوند؛ این کلیدها را حتی می‌توان در یک مخزن مرکزی (عمومی) ذخیره نمود.

۳) درنهایت، رمزنگاری کلید عمومی برای محیط‌های بازی مناسب (تر) است که در آن، طرفینی که هیچ‌گاه قبلاً باهم تعامل نداشته‌اند می‌خواهند قادر به برقراری ارتباط امن باشند. به‌عنوان یک مثال ساده، یک فروشنده‌ی اینترنتی می‌تواند کلید عمومی خود را به‌صورت آنلاین منتشر کند؛ آنگاه کاربری که خرید می‌کند می‌تواند هنگام نیاز برای رمزگذاری اطلاعات کارت اعتباری خود، کلید عمومی فروشنده را به دست آورد.

اختراع رمزگذاری کلید عمومی یک انقلاب در رمزنگاری بود. تصادفی نیست که تا اواخر دهه‌ی ۱۹۷۰ و اوایل دهه‌ی ۱۹۸۰، رمزگذاری و رمزنگاری به‌صورت کلی متعلق به حوزه‌ی سازمان‌های اطلاعاتی و نظامی بود و تنها با ظهور روش‌های کلید عمومی بود که استفاده از رمزنگاری در میان عموم مردم گسترش یافت.

مراجع و مطالعه‌ی بیشتر

ما مسائل توزیع کلید و مدیریت کلید را تنها به‌صورت مختصر موردبحث قرار داده‌ایم. برای اطلاعات بیشتر، پیشنهاد می‌کنیم به کتاب‌های پیرامون امنیت شبکه مراجعه کنید. کافمن و همکاران [۱۰۲] بررسی مناسبی از پروتکل‌های مورداستفاده برای توزیع کلید امن، هدف آن‌ها و چگونگی عملکرد آن‌ها، ارائه می‌کنند.

هیچ تلاشی برای بیان کل تاریخچه‌ی توسعه‌ی رمزنگاری کلید عمومی نکردیم. افراد دیگری به‌غیراز دیفی و هلمن هم بر روی ایده‌های مشابه در دهه‌ی ۱۹۷۰ کار می‌کردند. یک محقق که به‌صورت خاص در حال انجام مطالعات مشابه و مستقل بود، رالف مرکل است که بسیاری او را مخترع هم‌زمان رمزنگاری کلید عمومی می‌دانند (هرچند وی اثر خود را بعد از دیفی و هلمن منتشر نمود). همچنین به مایکل رابین اشاره می‌کنیم که ساختارهایی از طرح‌های امضا و طرح‌های رمزگذاری کلید عمومی بر اساس سختی تجزیه در حدود یک سال بعد از مطالعه‌ی ریوست، شامیر و ادلمن، توسعه داد [۱۴۸].

مطالعه‌ی مقاله‌ی اصلی دیفی و هلمن [۵۸] را به‌شدت توصیه می‌کنیم و خواننده را به کتاب نوشته‌شده توسط لوی [۱۱۴] برای اطلاعات بیشتر پیرامون ابعاد سیاسی و تاریخی انقلاب کلید عمومی، ارجاع می‌دهیم.

جالب اینجاست که ابعاد رمزنگاری کلید عمومی در جامعه‌ی اطلاعاتی قبل از منتشر شدن در ادبیات علمی عمومی، کشف شده بودند. در اوایل دهه‌ی ۱۹۷۰، جیمز الیس، کلیفورد کاکس و ملکوم ویلیامسون از سازمان اطلاعات بریتانیا، *GCHQ*، مفهوم رمزنگاری کلید عمومی، یک گونه از رمزگذاری *RSA* و یک گونه از پروتکل تبادل کلید دیفی-هلمن را اختراع کردند. اثر آن‌ها تا سال ۱۹۹۷ از حالت محرمانه خارج نشد. هرچند ریاضیات بنیادی رمزنگاری کلید عمومی ممکن است قبل از سال ۱۹۷۶