بسم الله الرحمن الرحیم

دانشگاه جامع انقلاب اسلامی

دانشکده فناوری‌های نوین

فصل۱۱

موضوع :مدیریت کلید و انقلاب کلید عمومی

استاد:جناب آقای دکتر لاجوردی

ارائه‌دهنده : مهسا یوسفی

۱۴۰۴/۱۰/۰۸

۱۱.۱ توزیع کلید و مدیریت کلید

❖ در فصل‌های پیشین دیدیم که چگونه **رمزنگاری با کلید خصوصی** می‌تواند برای **تضمین محرمانگی** و **یکپارچگی میان دو طرفی که از طریق یک کانال ناامن با یکدیگر ارتباط برقرار می‌کنند** به‌کار رود، مشروط بر آنکه بپذیریم این دو طرف **یک کلید محرمانه مشترک** در اختیار دارند.

❖ با این حال، پرسشی که از فصل ۱ تاکنون به تعویق انداخته‌ایم این است:

چگونه طرفین در وهله نخست می‌توانند یک کلید محرمانه را با یکدیگر به اشتراک بگذارند؟

❖ بدیهی است که **کلید را نمی‌توان به‌سادگی از طریق کانال ارتباطی ناامن ارسال کرد**، زیرا در این صورت یک مهاجمِ شنودکننده می‌تواند کلید را مشاهده کند و دیگر محرمانه نخواهد بود. بنابراین باید از سازوکار دیگری استفاده شود.

❖ در برخی موقعیت‌ها، **طرفین** ممکن است به یک **کانال امن** دسترسی داشته باشند که بتوانند از آن برای **به اشتراک‌گذاری مطمئن یک کلید محرمانه** استفاده کنند.

- یک نمونه رایج زمانی است که دو طرف در مقطعی از زمان از نظر فیزیکی در یک مکان حضور دارند و در آن زمان می‌توانند یک کلید را با هم به اشتراک بگذارند. به‌طور جایگزین، ممکن است طرفین بتوانند از **یک سرویس پیک مورد اعتماد** به‌عنوان یک **کانال امن** استفاده کنند.
- تأکید می‌کنیم که حتی اگر طرفین در مقطعی به یک کانال امن دسترسی داشته باشند، این موضوع رمزنگاری با کلید خصوصی را بی‌فایده **نمی‌کند**:
  - ✓ در مثال اول، طرفین در یک زمان خاص کانال امن دارند ولی بعداً ندارند،
  - ✓ در مثال دوم، استفاده از کانال امن ممکن است کندتر و پرهزینه‌تر از ارتباط از طریق یک کانال ناامن باشد.
- روش‌های فوق برای توزیع کلید در محیط‌های دولتی، دیپلماتیک و نظامی به‌کار رفته‌اند.
  - ✓ به‌عنوان نمونه، «تلفن قرمز» که در دهه ۱۹۶۰ مسکو و واشنگتن را به هم متصل می‌کرد، با استفاده از رمز یک‌بارمصرف (One-Time Pad) رمزنگاری شده بود و کلیدها توسط پیک‌هایی که بین دو کشور پرواز می‌کردند و چمدان‌هایی پر از نسخه‌های چاپی را حمل می‌کردند، مبادله می‌شد.

- چنین روش‌هایی همچنین می‌توانند در شرکت‌ها نیز استفاده شوند،
  - ✓ برای مثال، برای برقراری یک کلید مشترک میان یک پایگاه داده مرکزی و یک کارمند جدید در نخستین روز کاری او

    (در بخش بعدی دوباره به این مثال بازمی‌گردیم.)
- با این حال، **اتکا به یک کانال امن برای توزیع کلید** در بسیاری از موقعیت‌های دیگر به‌خوبی کار **نمی‌کند**.
  - ✓ برای مثال، یک شرکت بزرگ و چندملیتی را در نظر بگیرید که در آن هر جفت از کارمندان ممکن است نیاز داشته باشند به‌طور امن با یکدیگر ارتباط برقرار کنند، به‌گونه‌ای که ارتباطشان حتی از سایر کارمندان نیز محافظت شود. دست‌کم بسیار ناخوشایند خواهد بود که هر جفت از کارمندان مجبور شوند با هم ملاقات کنند تا بتوانند به‌صورت امن یک کلید را به اشتراک بگذارند، برای کارمندانی که در شهرهای مختلف کار می‌کنند، این کار حتی ممکن است غیرممکن باشد.

    حتی اگر مجموعه فعلی کارمندان به‌نحوی بتوانند با یکدیگر کلیدها را به اشتراک بگذارند، در عمل غیرواقعی و ناممکن خواهد بود که بتوانند با کارمندانی که پس از انجام این اشتراک اولیه به شرکت می‌پیوندند نیز کلید به اشتراک بگذارند.

- حتی با فرض اینکه این **N کارمند** به نحوی قادر به اشتراک‌گذاری ایمن کلیدها با یکدیگر باشند، یکی دیگر از معایب قابل توجه این است که **هر کارمند باید N-1 کلید مخفی** (یکی برای هر کارمند دیگر در شرکت) را مدیریت و ذخیره کند.
- **در واقع،** این ممکن است تعداد کلیدهای ذخیره شده توسط هر کاربر را به طور قابل توجهی کمتر از تعداد واقعی نشان دهد، زیرا کارمندان ممکن است برای برقراری ارتباط ایمن با منابع از راه دور مانند **پایگاه‌های داده، سرورها، چاپگرها** و غیره نیز به کلید نیاز داشته باشند.
- **افزایش این همه کلید مخفی یک مشکل لجستیکی قابل توجه است.**
- علاوه بر این، همه این کلیدها باید به طور ایمن ذخیره شوند. **هرچه تعداد کلیدها بیشتر باشد، محافظت از آنها دشوارتر** است و احتمال سرقت برخی از کلیدها توسط یک مهاجم بیشتر است.
- سیستم‌های کامپیوتری اغلب توسط **ویروس‌ها، کرم‌ها** و **سایر اشکال نرم‌افزارهای مخرب** که می‌توانند **کلیدهای مخفی** را بدزدند و آنها را بی‌سروصدا از طریق شبکه برای یک مهاجم ارسال کنند، آلوده می‌شوند.
- بنابراین، ذخیره کلیدها در رایانه‌های شخصی کارمندان همیشه یک راه حل ایمن **نیست**.

- برای روشن شدن موضوع، صرف نظر از تعداد کلیدهایی که هر طرف در اختیار دارد، **احتمال لو رفتن کلیدهای مخفی** همیشه یک نگرانی است.
- با این حال، هنگامی که فقط چند کلید نیاز به ذخیره دارند، راه‌حل‌های خوبی برای مقابله با این تهدید وجود دارد.
- امروزه یک راه‌حل معمول، ذخیره کلیدها روی سخت‌افزار امن مانند کارت هوشمند است.
- یک کارت هوشمند می‌تواند محاسبات رمزنگاری را با استفاده از کلیدهای مخفی ذخیره شده انجام دهد و اطمینان حاصل کند که این کلیدها هرگز به رایانه‌های شخصی کاربران راه پیدا نمی‌کنند.
- اگر کارت هوشمند به درستی طراحی شود، می‌تواند در برابر حمله بسیار مقاوم‌تر از یک رایانه شخصی باشد (به عنوان مثال، معمولاً نمی‌تواند توسط بدافزار آلوده شود) و بنابراین وسیله خوبی برای محافظت از کلیدهای مخفی کاربران ارائه می‌دهد.

- متأسفانه، کارت‌های هوشمند معمولاً از نظر **حافظه بسیار محدود** هستند و بنابراین **نمی‌توانند** صدها (یا هزاران) کلید را ذخیره کنند.
- همچنین ممکن است **در صورت گم شدن** تا حدودی گران باشند و جایگزینی آنها دشوار باشد.
- نگرانی‌های ذکر شده در بالا، در اصل، حتی اگر در عمل قابل حل نباشند، می‌توانند در سازمان‌های «بسته» متشکل از جمعیتی کاملاً مشخص از کاربران که همگی مایل به پیروی از سیاست‌های یکسان برای توزیع و ذخیره کلیدها هستند، مورد توجه قرار گیرند.
- با این حال، این نگرانی‌ها در «سیستم‌های باز» که کاربران تعاملات گذرا دارند، نمی‌توانند جلسه حضوری ترتیب دهند و حتی ممکن است تا زمانی که برای اولین بار می‌خواهند ارتباط برقرار کنند، از وجود یکدیگر آگاه نباشند، **از بین می‌روند**.

- در واقع، این وضعیت رایج‌تر از آن چیزی است که در ابتدا ممکن است تصور شود: ارسال اطلاعات کارت اعتباری به یک فروشنده اینترنتی که قبلاً هرگز چیزی از او خریداری نکرده‌اید، یا ارسال ایمیل به کسی که هرگز حضوری او را ملاقات نکرده‌اید را در نظر بگیرید.
- در چنین مواردی، رمزنگاری کلید خصوصی به تنهایی راه‌حلی ارائه نمی‌دهد و ما باید بیشتر به دنبال راه‌حل‌های مناسب باشیم.
- **به طور خلاصه**، حداقل سه مشکل متمایز مربوط به استفاده از رمزنگاری کلید خصوصی وجود دارد.
  - اولین مورد، توزیع کلید،
  - دومی ذخیره و مدیریت تعداد زیادی از کلیدهای مخفی و
  - سومی عدم کاربرد رمزنگاری کلید خصوصی در سیستم‌های باز است.

۱۱.۲ یک راه حل جزئی: مراکز توزیع کلید

- یک راه برای پرداختن به برخی از نگرانی‌های بخش قبلی، استفاده از یک مرکز توزیع کلید (KDC) برای ایجاد کلیدهای مشترک است.
- دوباره مورد یک شرکت بزرگ را در نظر بگیرید که در آن همه جفت‌های کارمند باید بتوانند به طور ایمن ارتباط برقرار کنند.

در چنین شرایطی، می‌توانیم از این واقعیت استفاده کنیم که همه کارمندان ممکن است به یک نهاد (مثلاً مدیر سیستم) حداقل در رابطه با امنیت اطلاعات مربوط به کار اعتماد کنند.

- این نهاد مورد اعتمادمی‌تواند به عنوان یک KDC عمل کند و به همه کارمندان کمک کند تا **کلیدهای جفتی** را به اشتراک بگذارند.
- وقتی یک کارمند جدید ملحق می‌شود، KDC می‌تواند یک کلید را با آن کارمند(**حضوری، در یک مکان امن**) به عنوان بخشی از اولین روز کاری آن کارمند به اشتراک بگذارد.

- در عین حال، KDC همچنین می‌تواند **کلیدهای مشترک** را بین **آن کارمند و همه کارمندان موجود** توزیع کند.
- یعنی وقتی کارمند iام ملحق می‌شود، KDC می‌تواند (علاوه بر به اشتراک گذاشتن یک کلید بین خود و این کارمند جدید) i-1کلید $k_1, \ldots, k_{i-1}$ تولید کند.
- k_{i-1}، این کلیدها را به کارمند جدید بدهید و سپس کلید k_j را با رمزگذاری آن با استفاده از کلیدی که کارمند از قبل با KDC به اشتراک گذاشته است، برای jامین کارمند موجود ارسال کنید.
- پس از این، کارمند جدید یک کلید را با هر کارمند دیگر (و همچنین با KDC) به اشتراک می‌گذارد.
- یک رویکرد بهتر، که از نیاز کارمندان به ذخیره و مدیریت کلیدهای چندگانه جلوگیری می‌کند، استفاده از KDC به صورت **آنلاین** برای تولید کلیدها **"در صورت تقاضا"** در هر زمان که **دو کارمند مایل به برقراری ارتباط ایمن** باشند، است.
- مانند قبل، KDCیک کلید (**متفاوت**) را با هر کارمند به اشتراک می‌گذارد، کاری که می‌تواند در اولین روز کاری هر کارمند به صورت ایمن انجام شود.

- فرض کنید KDC کلید k_A را با کارمند آلیس و کلید k_B را با کارمند باب به اشتراک می‌گذارد.
- در زمان دیگری، وقتی آلیس می‌خواهد به صورت ایمن با باب ارتباط برقرار کند، می‌تواند به سادگی پیام "من، آلیس، می‌خواهم با باب صحبت کنم" را به KDCارسال کند. (در صورت تمایل، این پیام را می‌توان با استفاده از کلید مشترک آلیس و KDCاحراز هویت کرد.)
- سپس KDCیک کلید تصادفی جدید به نام کلید جلسه انتخاب می‌کند و این کلید k را که با استفاده از k_A رمزگذاری شده است، برای آلیس و با استفاده از k_B برای باب ارسال می‌کند. (این پروتکل برای استفاده در عمل بسیار ساده است، به بحث بیشتر در زیر مراجعه کنید.)
- هنگامی که آلیس و باب هر دو این کلید جلسه را بازیابی می‌کنند، می‌توانند از آن برای برقراری ارتباط ایمن استفاده کنند.
- وقتی مکالمه‌شان تمام شد، می‌توانند (**و باید**) **کلید جلسه را پاک کنند** زیرا اگر بعداً بخواهند دوباره با KDC ارتباط برقرار کنند، همیشه می‌توانند دوباره با آن تماس بگیرند.

❖ **مزایای** این رویکرد را در نظر بگیرید:

۱. هر کارمند فقط باید **یک کلید مخفی بلندمدت** (یعنی همان کلیدی که با KDC به اشتراک می‌گذارد) را ذخیره کند.

- **کارمندان هنوز هم باید کلیدهای جلسه را مدیریت و ذخیره کنند**، اما این کلیدهای کوتاه‌مدت هستند که پس از **پایان جلسه ارتباط پاک می‌شوند.**
- KDC باید تعداد زیادی کلید بلندمدت را ذخیره کند. با این حال، **KDC** می‌تواند **در مکانی امن نگه‌داری شود** و **بالاترین سطح ممکن از حفاظت** در برابر حملات شبکه‌ای به آن اختصاص داده شود.

۲. وقتی یک کارمند به سازمان می‌پیوندد، تنها کاری که باید انجام شود این است که یک کلید بین این کارمند و KDC تنظیم شود. هیچ کارمند دیگری نیازی به به‌روزرسانی مجموعه کلیدهایی که در اختیار دارد، **ندارد**.

❖ بنابراین، KDC ها می‌توانند دو مورد از مشکلاتی را که در رابطه با رمزنگاری کلید خصوصی دیده‌ایم، **کاهش دهند**:

- آن‌ها می‌توانند توزیع کلید را ساده کنند (زیرا فقط یک کلید جدید باید هنگام پیوستن یک کارمند به اشتراک گذاشته شود و منطقی است که فرض کنیم یک کانال امن بین KDCو آن کارمند در اولین روز کاری‌شان وجود دارد) و می‌توانند پیچیدگی ذخیره‌سازی کلید را کاهش دهند (زیرا هر کارمند فقط باید یک کلید واحد را ذخیره کند).
- KDC ها در عملی کردن رمزنگاری کلید خصوصی در سازمان‌های بزرگ که در آن‌ها یک نهاد واحد وجود دارد که مورد اعتماد همه است، گام‌های بلندی برمی‌دارند.

❖ با این حال، برخی از معایب تکیه بر KDC ها وجود دارد:

۱. یک **حمله موفقیت‌آمیز به KDC** منجر به از **کار افتادن کامل سیستم** می‌شود:

- یک مهاجم می‌تواند تمام **کلیدها را به خطر بیندازد** و متعاقباً تمام ترافیک شبکه را **شنود** کند.
- این امر KDC را به یک هدف با ارزش بالا تبدیل می‌کند.
- توجه داشته باشید که حتی اگر KDCدر برابر حملات خارجی به خوبی محافظت شده باشد، **همیشه احتمال حمله داخلی توسط کارمندی** که به KDCدسترسی دارد (مثلاً مدیر فناوری اطلاعات) وجود دارد

۲. **KDCیک نقطه شکست واحد** است:

- اگر KDC از کار بیفتد، ارتباط امن موقتاً غیرممکن می‌شود.
- اگر **کارمندان دائماً با KDCتماس بگیرند** و **درخواست ایجاد کلیدهای جلسه** کنند، بار روی KDC می‌تواند بسیار زیاد باشد و در نتیجه احتمال **خرابی یا کندی پاسخ** آن افزایش می‌یابد.

❖ با این حال، برخی از معایب تکیه بر KDC ها وجود دارد:

۱. یک **حمله موفقیت‌آمیز به KDC** منجر به از **کار افتادن کامل سیستم** می‌شود:

- یک مهاجم می‌تواند تمام **کلیدها را به خطر بیندازد** و متعاقباً تمام ترافیک شبکه را **شنود** کند.
- این امر KDC را به یک هدف با ارزش بالا تبدیل می‌کند.
- توجه داشته باشید که حتی اگر KDCدر برابر حملات خارجی به خوبی محافظت شده باشد، **همیشه احتمال حمله داخلی توسط کارمندی** که به KDCدسترسی دارد (مثلاً مدیر فناوری اطلاعات) وجود دارد

۲. **KDCیک نقطه شکست واحد** است:

- اگر KDC از کار بیفتد، ارتباط امن موقتاً غیرممکن می‌شود.
- اگر **کارمندان دائماً با KDCتماس بگیرند** و **درخواست ایجاد کلیدهای جلسه** کنند، بار روی KDC می‌تواند بسیار زیاد باشد و در نتیجه احتمال **خرابی یا کندی پاسخ** آن افزایش می‌یابد.

- یک **راه حل ساده برای مشکل دوم**، تکثیر KDC است. این کار جواب می‌دهد (و در عمل انجام می‌شود)، اما همچنین به این معنی است که اکنون نقاط حمله بیشتری روی سیستم وجود دارد.
- اضافه کردن KDC های بیشتر، اضافه کردن کارمندان جدید را نیز دشوارتر می‌کند، زیرا به‌روزرسانی‌ها باید به طور ایمن به هر KDC منتشر شوند.

پروتکل‌های توزیع کلید با استفاده از KDC

- در مقالات علمی، تعدادی پروتکل برای توزیع کلید امن با استفاده از KDC وجود دارد.
- ما به طور خاص به پروتکل **نیدهم–شرودر (Needham–Schroeder)** اشاره می‌کنیم که هسته Kerberos را تشکیل می‌دهد، **یک سرویس مهم و پرکاربرد برای انجام احراز هویت و پشتیبانی از ارتباطات امن**.

(Kerberosدر بسیاری از دانشگاه‌ها و شرکت‌ها استفاده می‌شود و مکانیسم **پیش‌فرض** برای پشتیبانی از احراز هویت و ارتباطات شبکه‌ای امن در ویندوز و بسیاری از سیستم‌های یونیکس است.)

- ما فقط یک ویژگی از این پروتکل را برجسته می‌کنیم.
- هنگامی که آلیس با KDC تماس می‌گیرد و درخواست برقراری ارتباط با باب را می‌کند، KDC همانطور که قبلاً توضیح دادیم، کلید جلسه رمزگذاری شده را برای آلیس و باب ارسال نمی‌کند.
- در عوض، KDC علاوه بر کلید جلسه رمزگذاری شده با کلید باب، کلید جلسه رمزگذاری شده با کلید آلیس را نیز برای آلیس ارسال می‌کند. سپس آلیس متن رمز شده دوم را مانند **شکل ۱۱.۱** برای باب ارسال می‌کند.
- **متن رمز دوم** گاهی اوقات **بلیط** نامیده می‌شود و می‌تواند به عنوان یک **اعتبارنامه** در نظر گرفته شود که به آلیس اجازه می‌دهد با باب صحبت کند (و به باب این اطمینان را می‌دهد که با آلیس صحبت می‌کند).

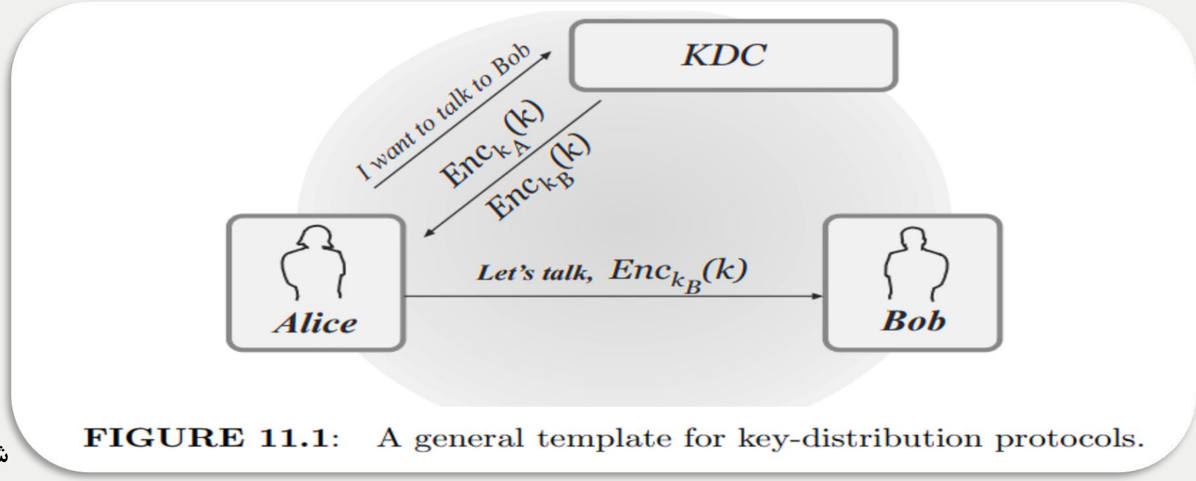

FIGURE 11.1: A general template for key-distribution protocols.

شکل ۱۱.۱: یک الگوی کلی برای پروتکل‌های توزیع کلید

- در واقع، اگرچه ما در بحث خود بر این نکته **تأکید نکرده‌ایم**، اما یک **رویکرد مبتنی بر KDC** می‌تواند وسیله‌ای مفید برای انجام **احراز هویت** نیز فراهم کند.
- همچنین توجه داشته باشید که آلیس و باب لزوماً هر دو **کاربر نیستند**، آلیس ممکن است یک کاربر و باب منبعی مانند یک سرور از راه دور، یک پایگاه داده یا یک چاپگر باشد.
- این پروتکل به این روش طراحی شده است تا بار روی KDC را **کاهش دهد**.
  - در پروتکلی که شرح داده شد، KDC نیازی به **شروع اتصال دوم به باب ندارد** و نیازی به نگرانی در مورد **آنلاین** بودن باب در **هنگام شروع پروتکل** توسط آلیس **ندارد**.
  - علاوه بر این، اگر آلیس بلیط (و نسخه خود از کلید جلسه) را حفظ کند، می‌تواند با ارسال مجدد بلیط به باب، بدون دخالت KDC، ارتباط امن با باب را دوباره آغاز کند. (در عمل، بلیط‌ها منقضی می‌شوند و در نهایت نیاز به تمدید دارند. اما یک جلسه می‌تواند در یک دوره زمانی قابل قبول دوباره برقرار شود.)

- در پایان، اشاره می‌کنیم که در عمل، کلیدی که آلیس با KDC به اشتراک می‌گذارد، ممکن است یک **رمز عبور کوتاه** و آسان برای به خاطر سپردن باشد. در این مورد، مشکلات امنیتی **اضافی زیادی ایجاد می‌شود** که باید به آنها رسیدگی شود.
- ما همچنین به طور ضمنی مهاجمی را فرض کرده‌ایم که فقط به صورت غیرفعال استراق سمع می‌کند، نه کسی که ممکن است به طور فعال سعی در تداخل با پروتکل داشته باشد.
- برای اطلاعات بیشتر درباره چگونگی رسیدگی به چنین مسائل، خواننده علاقه‌مند را به منابع ذکرشده در انتهای این فصل ارجاع می‌دهیم.

۱۱.۳ تبادل کلید و پروتکل دیفی_هلمن

- KDC ها و پروتکل‌هایی مانند Kerberos در عمل مورد استفاده قرار می‌گیرند.
- **اما** این رویکردها برای حل مسئلهٔ توزیع کلید همچنان مستلزم آن هستند که در مقطعی، یک **کانال خصوصی و احرازشده وجود داشته باشد** که بتوان از آن **برای به‌اشتراک‌گذاری کلیدها** استفاده کرد.

(به‌طور خاص، ما وجود چنین کانالی را میان KDC و یک کارمند در نخستین روز کاری او فرض کردیم.)

- بنابراین، آنها هنوز نمی‌توانند مشکل توزیع کلید را در **سیستم‌های باز** مانند اینترنت حل کنند، جایی که ممکن است **هیچ کانال خصوصی** بین دو کاربری که مایل به برقراری ارتباط هستند، **وجود نداشته باشد**.
- برای دستیابی به ارتباط خصوصی **بدون برقراری ارتباط از طریق کانال خصوصی**، رویکردی کاملاً متفاوت مورد نیاز است.

- در سال ۱۹۷۶، ویتفیلد **دیفی** و مارتین **هلمن مقاله‌ای با عنوان** ظاهراً ساده (معصومانه) **"مسیرهای جدید در رمزنگاری"** منتشر کردند.
- در آن اثر، آن‌ها مشاهده کردند که در جهان اغلب نوعی عدم تقارن وجود دارد، به ویژه، اقدامات خاصی وجود دارد که می‌توان به راحتی انجام داد اما به راحتی معکوس نکرد.
  - ✓ به عنوان مثال، قفل‌ها را می‌توان بدون کلید قفل کرد (یعنی به راحتی)، اما نمی‌توان دوباره آنها را باز کرد.
  - ✓ جالب‌تر اینکه، شکستن یک گلدان شیشه‌ای آسان است اما دوباره سرهم کردن آن بسیار دشوار است.
  - ✓ از نظر الگوریتمی (و برای اهداف ما مرتبط‌تر)، ضرب دو عدد اول بزرگ آسان است اما بازیابی آن اعداد اول از حاصل‌ضرب آنها دشوار است.

(این دقیقاً همان مسئله فاکتورگیری است که در فصل‌های قبلی مورد بحث قرار گرفت.)

- دیفی و هلمن متوجه شدند که چنین پدیده‌هایی را می‌توان برای **استخراج پروتکل‌های تعاملی** برای **تبادل کلید امن** استفاده کرد که به دو طرف اجازه می‌دهد از طریق ارتباط از طریق یک **کانال عمومی**، یک **کلید مخفی** را به اشتراک بگذارند، با این کار طرفین عملیاتی را انجام می‌دهند که یک استراق سمع‌کننده نمی‌تواند آن را معکوس کند.
- وجود پروتکل‌های تبادل کلید امن بسیار شگفت‌انگیز است.
- یعنی شما و یک دوست می‌توانید با فریاد زدن در یک اتاق (و انجام برخی محاسبات محلی) بر سر یک راز به توافق برسید؛ این راز برای هر کس دیگری ناشناخته خواهد بود، **حتی اگر آنها به تمام گفته‌ها گوش داده باشند**.
- در واقع، تا سال ۱۹۷۶ عموماً اعتقاد بر این بود که ارتباط امن بدون به اشتراک گذاشتن برخی اطلاعات مخفی با استفاده از یک کانال خصوصی امکان‌پذیر نیست.

- تأثیر مقاله دیفی و هلمن بسیار زیاد بود. علاوه بر معرفی یک روش اساساً جدید برای نگاه به رمزنگاری،
- این مقاله یکی از اولین گام‌ها برای انتقال رمزنگاری از **حوزه خصوصی** به **حوزه عمومی** بود.
- **ما دو پاراگراف اول مقاله آنها را نقل می‌کنیم:**

ما امروز در آستانه انقلابی در رمزنگاری هستیم. توسعه سخت‌افزار دیجیتال ارزان، آن را از محدودیت‌های طراحی محاسبات مکانیکی رها کرده و هزینه دستگاه‌های رمزنگاری درجه بالا را تا جایی کاهش داده است که می‌توان از آنها در کاربردهای تجاری مانند دستگاه‌های خودپرداز از راه دور و پایانه‌های رایانه‌ای استفاده کرد.

به نوبه خود، چنین کاربردهایی نیاز به انواع جدیدی از سیستم‌های رمزنگاری را ایجاد می‌کنند که ضرورت کانال‌های توزیع کلید امن را به حداقل برساند... در عین حال، پیشرفت‌های نظری در نظریه اطلاعات و علوم کامپیوتر نوید ارائه سیستم‌های رمزنگاری امن قابل اثبات را می‌دهد و این هنر باستانی را به یک علم تبدیل می‌کند.

دیفی و هلمن اغراق نمی‌کردند و انقلابی که از آن صحبت می‌کردند تا حد زیادی مدیون کار آنها بود.

- **در این بخش، پروتکل تبادل کلید دیفی-هلمن را ارائه می‌دهیم.**
- امنیت این پروتکل را در برابر مهاجمانِ شنودکننده اثبات می‌کنیم، یا به‌طور معادل، تحت این فرض که طرفین از طریق یک کانال عمومی اما **احرازشده** با یکدیگر ارتباط برقرار می‌کنند (به‌گونه‌ای که مهاجم نتواند در ارتباط آن‌ها دخالت کند).
- امنیت در برابر یک مهاجم شنودکننده تضمینی نسبتاً ضعیف است و در عمل، پروتکل‌های تبادل کلید باید مفاهیم قوی‌تری از امنیت را برآورده کنند که خارج از دامنه بحث فعلی ما هستند.

(علاوه بر این، ما در اینجا به شرایطی علاقه‌مند هستیم که طرفین ارتباط هیچ اطلاعات مشترک قبلی ندارند، که در این صورت هیچ کاری نمی‌توان برای جلوگیری از جعل هویت یکی از طرفین توسط مهاجم انجام داد. بعداً به این نکته بازخواهیم گشت.)

❖چارچوب و تعریف امنیت

چارچوبی را در نظر می‌گیریم که شامل دو طرف است (که به‌طور سنتی آلیس و باب نامیده می‌شوند) و این دو طرف یک پروتکل تصادفی Π را اجرا می‌کنند تا یک کلید محرمانه مشترک تولید کنند.

- ❖ Π را می‌توان مجموعه‌ای از دستورالعمل‌ها برای آلیس و باب در طول پروتکل دانست.
- ❖ آلیس و باب کار خود را با در اختیار داشتن پارامتر امنیتی 1^n آغاز می‌کنند، سپس Πرا با استفاده از بیت‌های تصادفی (مستقل) اجرا می‌کنند.
- ❖ در پایان پروتکل، آلیس و باب به‌ترتیب کلیدهایی $k_A, k_B \in \{0,1\}^n$ خروجی می‌دهند.
- ❖ شرط اساسی درستی این است که $k_A = k_B$.
- ❖ از آنجا که در ادامه فقط با پروتکل‌هایی سروکار داریم که این شرط را برآورده می‌کنند، صرفاً از کلید $k = k_A = k_B$که در یک اجرای صادقانه Πتولید می‌شود سخن می‌گوییم.

(از آنجا که Πتصادفی است، کلید تولیدشده عموماً در هر بار اجرای Πمتفاوت خواهد بود.)

❖ **اکنون به تعریف امنیت می‌پردازیم:**

به‌طور شهودی، یک پروتکل تبادل کلید **زمانی امن است** که کلید خروجیِ آلیس و **باب به‌طور کامل** از **دید یک مهاجم شنودکننده پنهان بماند.**

❖ این مفهوم به‌صورت صوری چنین تعریف می‌شود:

مهاجمی که اجرای پروتکل را شنود کرده است، نباید بتواند کلید kتولیدشده در آن اجرا (که اکنون میان آلیس و باب مشترک است) را از یک کلید یکنواخت با طول n تشخیص دهد.

❖ این شرط به‌مراتب قوی‌تر از این است که صرفاً بخواهیم مهاجم نتواند k را دقیقاً حدس بزند، ، و این مفهوم قوی‌تر در صورتی ضروری است که طرفین متعاقباً از k برای برخی کاربردهای رمزنگاری (مثلاً به عنوان کلید برای یک طرح رمزگذاری کلید خصوصی) استفاده کنند.

❖ **با فرمول‌بندی موارد فوق**، فرض کنید **Πیک پروتکل تبادل کلید**، **A یک دشمن(مهاجم)** و **n پارامتر امنیتی** باشد. آزمایش زیر را داریم:

✓ **آزمایش تبادل کلید** $\mathsf{KE}^{eav}_{A,\Pi}(n)$:

۱. دو طرف با داشتن 1^n پروتکل Π را اجرا می‌کنند. این منجر به تولید یک **رونوشت** $trans$ می‌شود که شامل تمام پیام‌های ارسالی طرفین است و همچنین کلیدی k که توسط هر یک از طرفین خروجی داده می‌شود.

۲. یک بیت یکنواخت $b \in \{0,1\}$ انتخاب می‌شود. اگر $b = 0$ آنگاه $\hat{k} := k$ و اگر $b = 1$ آنگاه $\hat{k}$ به‌طور یکنواخت از $\{0,1\}^n$ انتخاب می‌شود ($\hat{k} \in \{0,1\}^n$)

۳. مهاجم A رونوشت $trans$ و $\hat{k}$ را دریافت می‌کند و یک بیت b' خروجی می‌دهد.

۴. خروجی آزمایش به این صورت تعریف می‌شود که برابر با ۱ است اگر $b' = b$ و در غیر این صورت برابر با 0 .(در صورتی که $\mathsf{KE}^{eav}_{A,\Pi}(n) = 1$، گفته می‌شود که A موفق شده است.)

❖ به A مقدار trans داده می‌شود تا این واقعیت را نشان دهد که:

A کل اجرای پروتکل را شنود می‌کند و بنابراین تمام پیام‌های رد و بدل شده توسط طرفین را می‌بیند. در دنیای واقعی، به A **هیچ کلیدی داده نمی‌شود**، در این آزمایش، $\hat{k}$ فقط به عنوان وسیله‌ای برای تعریف معنای **شکستن امنیت** Π**توسط** A، به دشمن(مهاجم) داده می‌شود.

❖ به عبارت دیگر، مهاجم در «شکستن» Πموفق است **اگر بتواند به‌طور صحیح تعیین کند** که آیا $\hat{k}$ همان **کلید واقعی** مربوط به اجرای داده‌شده پروتکل است **یا** $\hat{k}$یک **کلید یکنواخت** است که مستقل از رونوشت می‌باشد.

❖ همان‌طور که انتظار می‌رود، می‌گوییم پروتکل Π **امن** است اگر احتمال موفقیت مهاجم تنها اندکی بیش از $1/2$باشد.

به عبارت دیگر:

❖ تعریف ۱۱.۱

یک پروتکل تبادل کلید Π در حضور یک شنودگر امن است اگر برای همه مهاجمان تصادفی با زمان چندجمله‌ای A، یک تابع جزئی‌ناپذیر $negl$وجود داشته باشد به‌طوری که:

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

❖ هدف یک پروتکل تبادل کلید تقریباً همیشه تولید یک کلید مشترک kاست که طرفین از آن برای یک هدف رمزنگاری بعدی استفاده کنند،

- ✓ برای مثال، رمزگذاری و احراز هویت ارتباطات بعدی آن‌ها با استفاده از، مثلاً، یک طرح رمزگذاری احراز هویت شده، استفاده خواهد شد.
- ✓ به طور شهودی، استفاده از یک کلید مشترک تولید شده توسط یک پروتکل تبادل کلید امن باید به خوبی استفاده از یک کلید مشترک از طریق یک کانال خصوصی باشد. (امکان اثبات رسمی این موضوع نیز وجود دارد؛ به تمرین ۱۱,۱ مراجعه کنید.)

❖ پروتکل تبادل کلید دیفی-هلمن

اکنون پروتکل تبادل کلید را که در مقاله اصلی دیفی و هلمن آمده است، شرح می‌دهیم (اگرچه آنها از آنچه ما در اینجا خواهیم بود، غیررسمی‌تر بودند).

❖ فرض کنید G یک الگوریتم تصادفی با زمان چندجمله‌ای باشد که با دریافت ورودی 1^n، یک **گروه چرخشی** G، **مرتبه آن** q (به‌طوری که $\|q\| = n$) و یک **مولد** $g \in G$ را خروجی می‌دهد. (به بخش ۹.۳.۲ مراجعه کنید.)

❖ پروتکل تبادل کلید دیفی-هلمن به طور رسمی به عنوان ساختار ۱۱.۲ توصیف شده و در شکل ۱۱.۲ نشان داده شده است.

❖ساختار ۱۱.۲: پروتکل تبادل کلید دیفی–هلمن

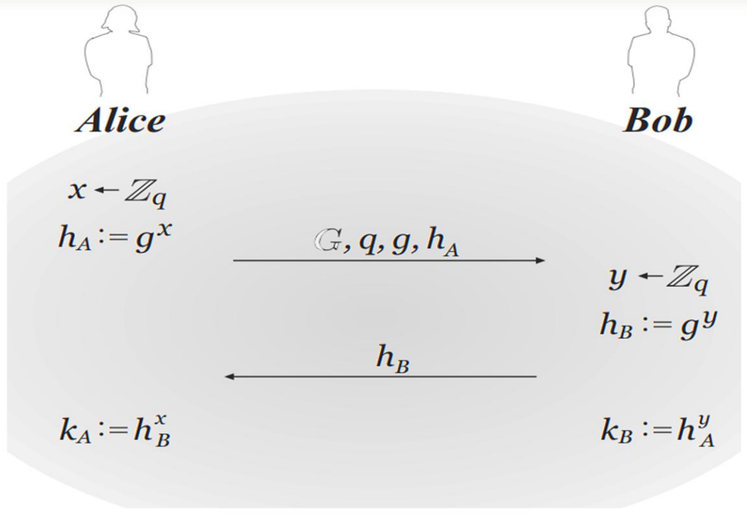

FIGURE 11.2: The Diffie–Hellman key-exchange protocol.

- **ورودی مشترک:** پارامتر امنیتی 1^n
- **پروتکل:**

1) آلیس الگوریتم $G(1^n)$را اجرا می‌کند تا (G, q, g)را به‌دست آورد.
2) آلیس یک عدد تصادفی $x \in \mathbb{Z}_q$انتخاب می‌کند و $h_A := g^x$را محاسبه می‌کند.
3) آلیس (G, q, g, h_A)را برای باب ارسال می‌کند.
4) باب (G, q, g, h_A)را دریافت می‌کند. او یک عدد تصادفی $y \in \mathbb{Z}_q$انتخاب کرده و $h_B := g^y$را محاسبه می‌کند. سپس h_Bرا برای آلیس ارسال می‌کند و کلید $k_B := h_A^y$را خروجی می‌دهد.
5) آلیس h_Bرا دریافت می‌کند و کلید $k_A := h_B^x$را خروجی می‌دهد.

- این پروتکل، همان **پروتکل تبادل کلید دیفی–هلمن** است.(شکل ۱۲: پروتکل تبادل کلید دیفی-هلمن)

- در توضیح ما، فرض کرده‌ایم که آلیس (G, q, g)را تولید کرده و این پارامترها را به‌عنوان بخشی از پیام اول خود برای باب ارسال می‌کند.
- در عمل، این پارامترها استاندارد شده و پیش از شروع پروتکل برای هر دو طرف شناخته‌شده هستند.
- در این حالت، آلیس تنها نیاز دارد h_Aرا ارسال کند و باب نیز نیازی ندارد منتظر دریافت پیام آلیس بماند تا h_Bرا محاسبه و ارسال کند.
- **مشاهده درستی پروتکل دشوار نیست:**

باب کلید $\boldsymbol{k_B = h_A^y = \left(g^x\right)^y = g^{xy}}$را محاسبه می‌کند و **آلیس کلید** $\boldsymbol{k_A = h_B^x = \left(g^y\right)^x = g^{xy}}$را محاسبه می‌کند، بنابراین $\boldsymbol{k_A = k_B}$.

(خواننده دقیق متوجه خواهد شد که کلید مشترک یک **عنصر گروهی** است، نه یک رشته بیت، به این نکته بعداً بازخواهیم گشت.)

❖ دیفی و هلمن امنیت پروتکل خود را اثبات نکردند. در واقع، مفاهیم مناسب (هم چارچوب تعریفی و هم ایده تدوین فرضیات دقیق) هنوز وجود نداشتند.

❖ **بیایید ببینیم چه نوع فرضی برای ایمن بودن پروتکل مورد نیاز است:**

- اولین مشاهده‌ای که توسط دیفی و هلمن انجام شده است، این است که حداقل الزام برای امنیت در اینجا این است که مسئله لگاریتم گسسته نسبت به G سخت باشد.
- در غیر این صورت، مهاجمی که رونوشت را در اختیار دارد (که به‌طور خاص شامل h_Aاست) می‌تواند مقدار مخفی یکی از طرفین (یعنی x) را محاسبه کرده و سپس به‌سادگی کلید مشترک را با استفاده از آن مقدار محاسبه کند.
- بنابراین، **دشواری(سختی) مسئله لگاریتم گسسته** برای امن‌بودن پروتکل **ضروری است**.
- با این حال، این شرط کافی نیست، زیرا ممکن است روش‌های دیگری برای محاسبه کلید $k_A = k_B$بدون محاسبه مستقیم xیا yوجود داشته باشد. **فرضیه محاسباتی دیفی–هلمن (Computational Diffie–Hellman)**، که تنها تضمین می‌کند کلید g^{xy}از روی رونوشت به‌طور کامل محاسبه‌کردنی نیست، نیز به‌تنهایی کافی نیست.

- آنچه در **تعریف ۱۱.۱** لازم است این است که کلید مشترک g^{xy}برای هر مهاجمی که g، g^x و g^y را در اختیار دارد، از یک توزیع یکنواخت **غیرقابل تشخیص** باشد. این دقیقاً همان **فرضیه دیفی–هلمن تشخیصی** (**Decisional Diffie–Hellman**)است که در بخش ۹،۳،۲ معرفی شد.
- همان‌طور که خواهیم دید، اثبات امنیت این پروتکل تقریباً بلافاصله از فرضیه دیفی–هلمن تشخیصی نتیجه می‌شود. این موضوع نباید تعجب‌آور باشد، زیرا فرضیات دیفی–هلمن — که مدت‌ها پس از انتشار مقالهٔ دیفی و هلمن معرفی شدند — دقیقاً با هدف انتزاع ویژگی‌هایی که زیربنای امنیت (حدسی) پروتکل دیفی–هلمن هستند، ارائه شدند.

- با توجه به این موضوع، این پرسش منصفانه است که آیا تعریف و اثبات امنیت در اینجا چیزی به دانسته‌های ما اضافه می‌کند یا نه؟ تا این مرحله از کتاب، امیدواریم که شما متقاعد شده باشید که پاسخ مثبت است.
- تعریف دقیق امنیت برای پروتکل‌های تبادل کلید، ما را مجبور می‌کند تا در مورد اینکه دقیقاً چه ویژگی‌های امنیتی می‌خواهیم، فکر کنیم.
- مشخص کردن یک فرض دقیق (**یعنی فرضیهٔ دیفی–هلمن تشخیصی**) به این معنی است که می‌توانیم آن فرض را مستقل از هر کاربرد خاصی مطالعه کنیم و (هنگامی که از معقول بودن آن مطمئن شدیم پروتکل‌های دیگری را بر اساس آن بسازیم، در نهایت، اثبات امنیت نشان می‌دهد که این فرض، در واقع، برای برآورده کردن مفهوم مورد نظر ما از امنیت توسط پروتکل کافی است.

- در اثبات امنیت خود، از نسخه‌ای اصلاح‌شده از **تعریف ۱۱.۱** استفاده می‌کنیم که در آن لازم است کلید مشترک به‌جای آنکه از یک رشته بیتی یکنواخت با طول nغیرقابل تشخیص باشد، از یک **عنصر یکنواخت از گروه** Gغیرقابل تشخیص باشد.
- این ناهماهنگی پیش از آنکه پروتکل در عمل مورد استفاده قرار گیرد باید برطرف شود (چرا که عناصر گروه معمولاً به‌عنوان کلیدهای رمزنگاری کاربردی نیستند و نمایش یک عنصر یکنواخت از گروه نیز، به‌طور کلی، یک رشته بیتی یکنواخت نخواهد بود.
- پس از ارائه اثبات، به‌طور خلاصه یکی از روش‌های استاندارد برای رفع این مسئله را بررسی می‌کنیم.
- فعلاً، با $\widehat{KE}^{eav}_{A,\Pi}(n)$ یک آزمایش اصلاح‌شده را مشخص می‌کنیم که در آن اگر $b = 1$باشد، آنگاه $\hat{k}$به‌طور یکنواخت از گروه Gانتخاب می‌شود، نه به‌طور یکنواخت از مجموعه$\{0,1\}^n$.

❖قضیه ۱۱.۳

اگر مسئله دیفی–هلمن تشخیصی نسبت به گروه Gسخت باشد، آنگاه پروتکل تبادل کلید دیفی–هلمن Πدر حضور یک شنودگر امن است (نسبت به آزمایش اصلاح‌شده $\widehat{KE}^{eav}_{A,\Pi}$).

❖عناصر گروه یکنواخت در مقابل رشته‌های بیتی یکنواخت:

قضیه قبلی نشان می‌دهد که خروجی کلید توسط آلیس و باب در پروتکل دیفی‌هلمن (برای یک شنودگر زمان چندجمله‌ای) از یک عنصر گروه یکنواخت قابل تشخیص نیست.

❖ برای استفاده از کلید برای کاربردهای رمزنگاری بعدی و همچنین برآورده کردن تعریف ۱۱.۱، خروجی کلید توسط طرفین باید از یک رشته بیتی یکنواخت با طول مناسب قابل تشخیص نباشد.

❖ **پروتکل دیفی‌هلمن را می‌توان برای دستیابی به این هدف اصلاح کرد**، به این صورت که طرفین یک تابع استخراج کلید مناسب (به بخش ۶.۶.۴ مراجعه کنید) را برای عنصر گروه مشترک g^{xy} که هر کدام محاسبه می‌کنند، اعمال کنند.

❖ دشمنان فعال:

تاکنون فقط یک دشمن شنود را در نظر گرفته‌ایم. اگرچه حملات شنود تاکنون رایج‌ترین هستند (زیرا انجام آنها آسان‌ترین است)، اما به هیچ وجه تنها حمله ممکن نیستند.

- **حملات فعال**، که در آن دشمن پیام‌های خود را به یکی از طرفین یا هر دو طرف ارسال می‌کند، نیز نگران‌کننده هستند و هر پروتکلی که در عمل استفاده می‌شود، باید در برابر چنین حملاتی نیز مقاوم باشد.
- **هنگام بررسی حملات فعال**، مفید است که به طور غیررسمی بین **حملات جعل هویت** که در آن دشمن در حین تعامل با طرف دیگر، خود را به جای یک طرف جا می‌زند، و **حملات مرد میانی** که در آن هر دو طرف صادق در حال اجرای پروتکل هستند و دشمن پیام‌های ارسالی از یک طرف به طرف دیگر را رهگیری و تغییر می‌دهد، تمایز قائل شویم.
- ما به **طور رسمی امنیت** در برابر هیچ یک از این دو دسته از حملات را **تعریف نخواهیم کرد**، زیرا چنین تعاریفی نسبتاً پیچیده هستند و بدون به اشتراک گذاشتن برخی اطلاعات از قبل توسط طرفین، قابل دستیابی نیستند.
- با این وجود، شایان ذکر است که پروتکل دیفی-هلمن در برابر حمله مرد میانی **کاملاً ناامن است**.
- در واقع، یک مهاجم مرد میانی می‌تواند به گونه‌ای عمل کند که آلیس و باب پروتکل را با کلیدهای مختلف k_A و k_B که هر دو برای مهاجم شناخته شده هستند، خاتمه دهند، اما نه آلیس و نه باب **نمی‌توانند تشخیص دهند** که حمله‌ای انجام شده **است**. جزئیات این حمله را به عنوان یک تمرین می‌گذاریم.

❖ تبادل کلید دیفی_هلمن در عمل:

پروتکل دیفی_هلمن در شکل اولیه خود معمولاً به دلیل **ناامنی در برابر حملات مرد میانی**، همانطور که در بالا بحث شد، **در عمل استفاده نمی‌شود**.

- ❖ این به هیچ وجه از اهمیت آن نمی‌کاهد. **پروتکل دیفی_هلمن به عنوان اولین اثبات این بود** که تکنیک‌های نامتقارن (و مسائل نظریه اعداد) می‌توانند برای کاهش مشکلات توزیع کلید در رمزنگاری استفاده شوند.
- ❖ علاوه بر این، پروتکل دیفی_هلمن در **هسته پروتکل‌های استاندارد تبادل کلید** قرار دارد که در برابر حملات مرد میانی مقاوم هستند و امروزه به طور گسترده مورد استفاده قرار می‌گیرند. یک نمونه قابل توجه TLS است. (به بخش ۱۳.۷ مراجعه کنید.)

۱۱.۴ انقلاب کلید عمومی

- علاوه بر تبادل کلید، دیفی و هلمن در کار پیشگامانهٔ خود مفهوم **رمزنگاری کلید عمومی** (یا **نامتقارن**) را نیز معرفی کردند.
- در **محیط کلید عمومی** (در مقابل محیط کلید خصوصی که تاکنون بررسی کرده‌ایم)، طرفی که می‌خواهد به‌صورت امن ارتباط برقرار کند یک **جفت کلید** تولید می‌کند: یک **کلید عمومی** که به‌طور گسترده منتشر می‌شود، و یک **کلید خصوصی** که آن را محرمانه نگه می‌دارد. (وجود دو کلید متفاوت است که این طرح را نامتقارن می‌کند.)
- پس از تولید این کلیدها، یک طرف می‌تواند با استفاده از آن‌ها **محرمانگی** پیام‌هایی را که دریافت می‌کند از طریق یک **طرح رمزنگاری کلید عمومی** تضمین کند، یا **یکپارچگی** پیام‌هایی را که ارسال می‌کند از طریق یک **طرح امضای دیجیتال** تضمین نماید (نگاه کنید به شکل ۱۱.۳).

تنظیم کلید عمومی	تنظیم کلید خصوصی	
رمزگذاری کلید عمومی	رمزگذاری کلید خصوصی	محرمانگی
طرح‌های امضای دیجیتال	کدهای احراز هویت پیام	یکپارچگی

شکل ۱۱,۳: اصول اولیه رمزنگاری در تنظیمات کلید خصوصی و کلید عمومی

- (ما در اینجا مختصری از این اصول اولیه را ارائه می‌دهیم و آنها را به ترتیب در فصل‌های ۱۲ و ۱۳ با جزئیات کامل مورد بحث قرار می‌دهیم.)

- **در یک طرح رمزگذاری کلید عمومی**، کلید عمومی تولید شده توسط یکی از طرفین به عنوان کلید رمزگذاری عمل می‌کند، هر کسی که آن کلید عمومی را بداند، می‌تواند از آن برای رمزگذاری پیام‌ها و تولید متن‌های رمزی مربوطه استفاده کند.
- کلید خصوصی به عنوان کلید رمزگشایی عمل می‌کند و توسط طرفی که آن را می‌داند برای بازیابی پیام اصلی از هر متن رمزی تولید شده با استفاده از کلید عمومی منطبق استفاده می‌شود.
- علاوه بر این، و شگفت‌انگیز است که چنین چیزی وجود دارد! محرمانه بودن پیام‌های رمزگذاری شده حتی در برابر دشمنی که **کلید عمومی را می‌داند (اما کلید خصوصی را نمی‌داند)** حفظ می‌شود.
- به عبارت دیگر، کلید رمزگذاری (**عمومی**) برای مهاجمی که سعی در رمزگشایی متن‌های رمزی رمزگذاری شده با استفاده از آن کلید دارد، فایده‌ای ندارد.

- بنابراین، برای امکان برقراری **ارتباط مخفی، گیرنده می‌تواند به سادگی کلید عمومی** خود را برای فرستنده بالقوه ارسال کند (بدون اینکه نگران استراق سمعی باشد که آن را مشاهده می‌کند)، یا کلید عمومی خود را در صفحه وب خود یا در یک پایگاه داده مرکزی منتشر کند. بنابراین، یک طرح رمزگذاری کلید عمومی، ارتباط خصوصی را بدون تکیه بر یک کانال خصوصی برای توزیع کلید، امکان‌پذیر می‌کند.
- یک **طرح امضای دیجیتال** همتای کلید عمومیِ **کُد احراز اصالت پیام** (MAC)است.
- در اینجا، **کلید خصوصی** به‌عنوان یک «کلید احراز اصالت» (که **کلید امضا** نامیده می‌شود) عمل می‌کند و به طرفی که این کلید را می‌داند امکان می‌دهد برای پیام‌هایی که ارسال می‌کند «برچسب‌های احراز اصالت» (یا **امضاها**) تولید کند.
- **کلید عمومی** نقش **کلید راستی‌آزمایی** را ایفا می‌کند و به هر کسی که آن را بداند اجازه می‌دهد امضاهای صادرشده توسط فرستنده را بررسی کند.

- همانند MACها، یک طرح امضای دیجیتال می‌تواند از دست‌کاری غیرقابل تشخیصِ پیام جلوگیری کند، با این تفاوت که در اینجا **امنیت حتی در برابر مهاجمی که کلید عمومی را می‌داند نیز برقرار است**.
- **عمومی‌بودن راستی‌آزمایی** (یعنی اینکه هر کسی که کلید عمومیِ فرستنده را بداند می‌تواند آن را انجام دهد) پیامدهای گسترده‌ای دارد، زیرا این امکان را فراهم می‌کند که سندی را که آلیس امضا کرده است نزد یک شخص ثالث (مثلاً یک قاضی) برای راستی‌آزمایی ارائه داد.
- این ویژگی **عدم‌انکار** (non-repudiation) نامیده می‌شود و کاربردهای فراوانی در تجارت الکترونیک دارد (برای مثال، در امضای اسناد حقوقی).
- امضاهای دیجیتال همچنین برای **توزیع امن کلیدهای عمومی** به‌عنوان بخشی از یک **زیرساخت کلید عمومی** به کار می‌روند، موضوعی که در بخش ۱۳.۶ با جزئیات بیشتری بررسی می‌شود.

- دیفی و هلمن در مقاله خود مفهوم رمزنگاری کلید عمومی را مطرح کردند، اما هیچ ساختار کاندیدی ارائه ندادند.
- یک سال بعد، ران ریوست، آدی شامیر و لن آدلمن مسئله RSA را پیشنهاد کردند و اولین طرح‌های رمزگذاری کلید عمومی و امضای دیجیتال را، بر اساس سختی آن مسئله ارائه دادند.
- انواع طرح‌های آنها اکنون جزو پرکاربردترین سیستم‌های رمزنگاری امروزی هستند.
- در سال ۱۹۸۵، طاهر الجمل یک طرح رمزگذاری ارائه داد که اساساً یک تغییر جزئی در پروتکل تبادل کلید دیفی_هلمن است، که انواع آن اکنون نیز به طور گسترده استفاده می‌شوند.
- بنابراین، اگرچه دیفی و هلمن در ساخت یک طرح رمزگذاری کلید عمومی (غیر تعاملی) موفق نشدند، اما بسیار به آن نزدیک شدند.

- در پایان، **خلاصه‌ای از چگونگی پرداختن رمزنگاری کلید عمومی به محدودیت‌های تنظیمات کلید خصوصی** که در بخش ۱۱.۱ مورد بحث قرار گرفت، ارائه می‌دهیم:

1. رمزنگاری کلید عمومی امکان توزیع کلید را از طریق کانال‌های عمومی (اما احراز هویت شده) فراهم می‌کند. این امر می‌تواند توزیع و به‌روزرسانی کلیدهای مشترک و مخفی را ساده کند
2. رمزنگاری کلید عمومی نیاز کاربران به ذخیره کلیدهای مخفی متعدد را کاهش می‌دهد. دوباره محیط یک شرکت بزرگ را در نظر بگیرید که در آن هر جفت از کارمندان به توانایی برقراری ارتباط ایمن نیاز دارند. با استفاده از رمزنگاری کلید عمومی، کافی است هر کارمند فقط یک کلید خصوصی (کلید خود) و کلیدهای عمومی سایر کارمندان را ذخیره کند. نکته مهم این است که این کلیدهای اخیر نیازی به مخفی نگه داشتن ندارند. آنها حتی می‌توانند در یک مخزن مرکزی (عمومی) ذخیره شوند.
3. در نهایت، رمزنگاری کلید عمومی (بیشتر) برای محیط‌های باز مناسب است که در آن طرفینی که قبلاً هرگز با هم تعامل نداشته‌اند، می‌خواهند توانایی برقراری ارتباط ایمن را داشته باشند. به عنوان یک مثال رایج، یک شرکت می‌تواند کلید عمومی خود را به صورت آنلاین ارسال کند.

- کاربری که خریدی انجام می‌دهد می‌تواند در صورت نیاز، کلید عمومی شرکت را در صورت نیاز به رمزگذاری اطلاعات کارت اعتباری خود برای ارسال به آن شرکت، دریافت کند.
- اختراع رمزگذاری کلید عمومی انقلابی در رمزنگاری بود.
- تصادفی نیست که تا اواخر دهه ۱۹۷۰ و اوایل دهه ۱۹۸۰، رمزگذاری و رمزنگاری به طور کلی متعلق به حوزه سازمان‌های اطلاعاتی و نظامی بود و تنها با ظهور تکنیک‌های کلید عمومی، استفاده از رمزنگاری گسترده شد.
- **چرا باید رمزنگاری کلید خصوصی را مطالعه کرد؟** باید واضح باشد که رمزنگاری کلید عمومی کاملاً قوی‌تر از رمزنگاری کلید خصوصی است؛ به طور خاص، هر طرح رمزگذاری کلید عمومی می‌تواند به عنوان یک طرح رمزگذاری کلید خصوصی استفاده شود. (کاربران در حال ارتباط می‌توانند به سادگی هم کلید عمومی و هم کلید خصوصی را به اشتراک بگذارند. اگر محرمانگی پیام‌های رمزگذاری شده حتی زمانی که شنودکننده کلید عمومی را می‌داند، برقرار باشد، پس وقتی کلید عمومی مخفی نگه داشته می‌شود، به وضوح برقرار است!)
- **پس چرا اصلاً زحمت مطالعه رمزنگاری کلید خصوصی را به خود دادیم؟** (پاسخ به سوال در اسلاید بعدی)

❖ پاسخ ساده است:

رمزنگاری کلید خصوصی بسیار کارآمدتر از رمزنگاری کلید عمومی است و باید در محیط‌هایی که مناسب است استفاده شود.

یعنی **در مواردی که امکان به اشتراک گذاشتن یک کلید برای طرفین ارتباط وجود دارد**، باید از **رمزنگاری کلید خصوصی استفاده شود.**

این شامل سیستم‌های بسته و کوچک کاربران و همچنین برنامه‌هایی مانند رمزگذاری دیسک می‌شود.

علاوه بر این، همانطور که در بخش‌های ۱۲,۳ و ۱۳,۷ خواهیم دید، رمزگذاری کلید خصوصی در تنظیمات کلید عمومی برای دستیابی به کارایی بهتر استفاده می‌شود.

❖ منابع و مطالعه بیشتر

❖ ما تنها به‌صورت مختصر به مسائل توزیع کلید و مدیریت کلید پرداخته‌ایم. برای اطلاعات بیشتر، توصیه می‌کنیم به کتاب‌های درسی امنیت شبکه مراجعه کنید، از جمله کتاب کافمن و همکاران [۱۱۳].

❖ ما تلاشی برای ارائه تاریخچه کامل توسعه رمزنگاری کلید عمومی نکرده‌ایم.

افزون بر دیفی و هلمن، افراد دیگری نیز در دهه ۱۹۷۰ روی ایده‌های مشابه کار می‌کردند. به‌ویژه می‌توان از **رالف مرکل** نام برد که به‌طور مستقل پژوهش‌های مشابهی انجام داد و از نظر بسیاری یکی از هم‌مخترعان رمزنگاری کلید عمومی به‌شمار می‌رود (هرچند انتشار کارهای او پس از دیفی و هلمن صورت گرفت).

❖ همچنین از **مایکل رابین** یاد می‌کنیم که حدود یک سال پس از کار **ریوست، شمیر و ادلمن** [۱۷۱]، ساختارهایی برای طرح‌های امضای دیجیتال و رمزنگاری کلید عمومی بر پایه سختی مسئله فاکتورگیری ارائه داد.

❖ مطالعه مقاله اصلی دیفی و هلمن [۶۵] را به‌شدت توصیه می‌کنیم و برای آشنایی بیشتر با جنبه‌های سیاسی و تاریخی انقلاب کلید عمومی، خواننده را به کتاب **لِوی** [۱۲۹] ارجاع می‌دهیم.

❖ نکته جالب این است که برخی جنبه‌های رمزنگاری کلید عمومی پیش از انتشار در ادبیات علمی باز، در جامعه اطلاعاتی کشف شده بودند.

❖ در اوایل دهه ۱۹۷۰، **جیمز الیس**، **کلیفورد کاکس** و **مالکوم ویلیامسون** از سازمان اطلاعاتی بریتانیا(GCHQ) مفهوم رمزنگاری کلید عمومی، گونه‌ای از رمزنگاری RSA، و گونه‌ای از پروتکل تبادل کلید دیفی–هلمن را ابداع کردند.

❖ این کارها تا سال ۱۹۹۷ از حالت محرمانه خارج نشد. اگرچه ممکن است ریاضیات زیربنایی رمزنگاری کلید عمومی پیش از سال ۱۹۷۶ کشف شده باشد، اما منصفانه است بگوییم که پیامدهای گسترده این فناوری نوین تا زمانی که دیفی و هلمن مطرح شدند، به‌درستی درک نشده بود.

با تشکر از توجه شما