به نام خدا

دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری ۱۷ اردیبهشت ۱۳۹۲

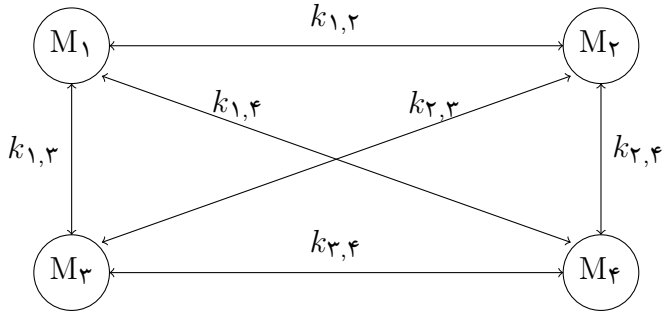
جلسه‌ی ۱۹: روش‌های تبادل کلید

مدرس: دکتر شهرام خزائی *نگارنده: نیلوفر صفی صمغ آبادی*

در جلسات قبل دیدیم که چگونه با استفاده از رمزنگاری توسط یک کلید خصوصی[۱]، می‌توانیم در یک کانال ناامن، یک ارتباط امن داشته باشیم که تحت آن از داده‌های ارسالی، در برابر حملات مهاجم[۲] محافظت شود. اکنون، هدف، معرفی سیستم‌های رمز نامتقارن[۳] یا مدل‌های رمزنگاری کلید عمومی[۴] است. در این راستا، ابتدا، به معرفی مفهوم تبادل کلید[۵] و طراحی پروتکل‌های تبادل کلید امن می‌پردازیم.

۱ مسأله تبادل کلید

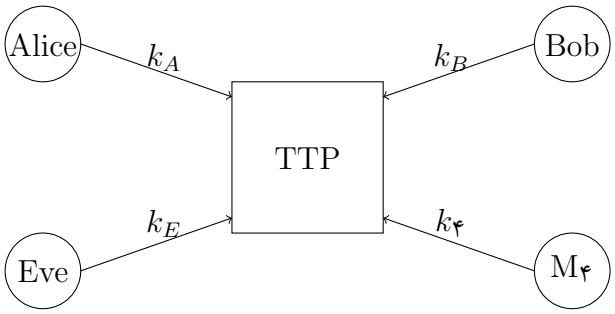
سؤال فرض کنید که n کاربر می‌خواهند دو به دو با هم ارتباط برقرار کنند. هر کاربر برای ارتباط با سایرین چگونه از کلید خصوصی‌اش استفاده می‌کند؟
روش اوّل هر کاربر، با هر یک از $n-1$ کاربر دیگر یک کلید خصوصی متمایز را به اشتراک گذارد. در این حالت، بطور مثال، اگر چهار کاربر داشته باشیم تخصیص کلید به شکل زیر صورت می‌گیرد:



مشکل این روش، مدیریت تعداد زیاد کلیدهای به اشتراک گذاشته شده است. در حقیقت، در یک شبکه‌ی n-کاربره هر کاربر باید برای برقراری ارتباط با سایر کاربران، $\mathcal{O}(n)$ کلید را ذخیره سازد که این خوب نیست؛ زیرا، هر چه تعداد کلیدها بیشتر باشد، محافظت از آنها دشوارتر بوده و احتمال دستیابی به تعدادی از آنها برای مهاجم بیشتر

[۱] secret key
[۲] adversary
[۳] asymmetric
[۴] public-key encryption schemes
[۵] key exchange

است. برای رفع این معضل، روش دیگری را معرفی می‌کنیم.
روش دوم یک شخص ثالث معتمد (TTP[۶]) وارد عمل می‌شود که نقش یک مرکز تولید کلید (KDC[۷]) را بازی می‌کند. بطور مثال اگر چهار کاربر داشته باشیم، هر یک از کاربران، بصورت زیر کلید خود را با او به اشتراک می‌گذارند:

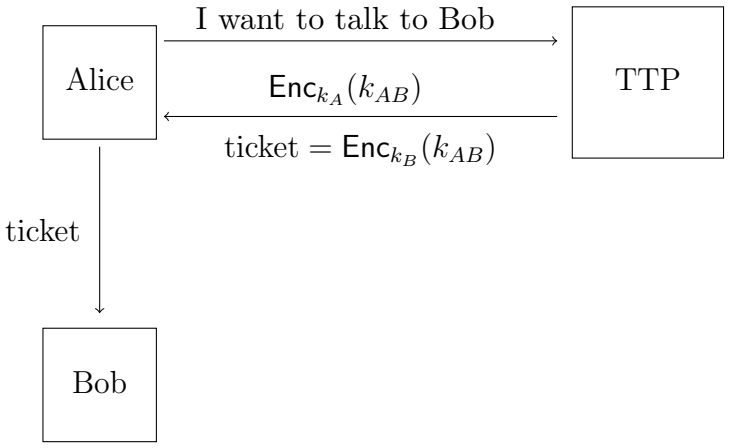


در این روش، هر کاربر تنها باید یک کلید را ذخیره سازد. فرض کنیم که Alice می‌خواهد با Bob ارتباط برقرار کند. بدین منظور، باید از یک پروتکل تبادل کلید مطمئن استفاده کند، تا کلید خصوصی K_{AB} را با Bob به اشتراک گذارد. این عمل، به شیوه‌ی زیر انجام می‌شود:
طبق شکل بالا، k_A و k_B به ترتیب کلیدهای خصوصی Alice و Bob هستند، که هر دو در اختیار TTP قرار دارند. ابتدا، Alice یک پیام، مبنی بر اینکه تمایل دارد با Bob ارتباط برقرار کند، برای TTP ارسال می‌کند. سپس TTP کلید تصادفی k_{AB} را تولید می‌کند و بعنوان پاسخ، پیامی را برای Alice می‌فرستد که دارای دو قسمت است: قسمت اول، $\mathsf{Enc}_{k_A}(k_{AB})$ است که با استفاده از کلید خصوصی Alice، بدست می‌آید. قسمت دوم پیامی که TTP برای Alice ارسال می‌کند، گذرانه[۸] نامیده می‌شود و با استفاده از کلید خصوصی Bob، بشکل زیر محاسبه می‌شود:

$$\text{ticket} = \mathsf{Enc}_{k_B}(k_{AB})$$

ارتباط Alice با TTP، پس از دریافت متن رمزی مخصوص به خودش (قسمت اول پیام دریافتی) و یک گذرانه برای Bob (قسمت دوم پیام دریافتی)، خاتمه می‌یابد. سپس، Alice ابتدا قسمت اول پیام را، با کلید خصوصی‌اش رمزگشایی کرده و k_{AB} را بدست می‌آورد و در ادامه گذرانه را برای Bob ارسال می‌کند. Bob نیز با استفاده از کلید خصوصی‌اش، گذرانه دریافتی را رمزگشایی کرده و کلید k_{AB} را بدست می‌آورد. حال، Alice و Bob، یک کلید اختصاصی مشترک و امن دارند، که توسط آن قادرند با یکدیگر ارتباط برقرار کنند. شمایی از کل عملیات مذکور در شکل زیر نشان داده شده است:

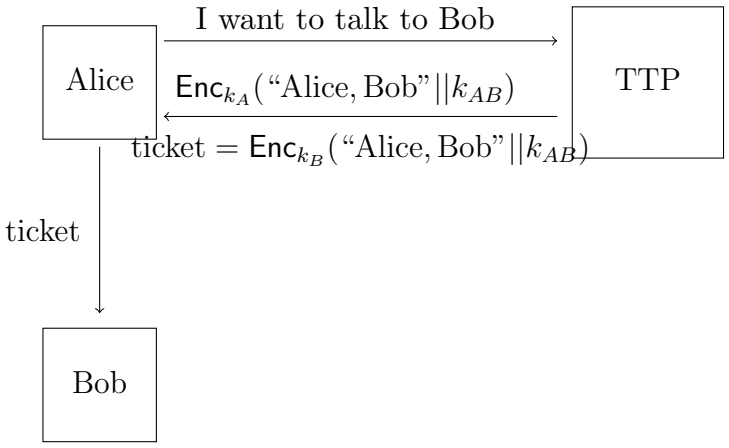
[۶] trusted third party
[۷] key distribution center
[۸] ticket



اگر سیستم رمزنگاری متقارن استفاده شده دارای امنیت متن اصلی انتخابی (CPA-امن) باشد، پروتکل تبادل کلید معرفی شده، در برابر مهاجم منفعل[۹] (یا شنودگر[۱۰])، امن است. چرا که شهوداً مهاجم منفعل، حتی با دیدن پیام ارسالی TTP به Alice هم نمی‌تواند متن رمزی کلید k_{AB} را از متن رمزی یک کلید کاملا تصادفی تمییز دهد. بعبارتی، نمی‌تواند هیچ اطلاعاتی در مورد k_{AB} بدست آورد.

حمله جعل هویت اما پروتکل فوق در برابر مهاجم فعال[۱۱] امن نیست. به عنوان مثال در پروتکل مذکور Eve به عنوان یک مهاجم می‌تواند پیامی مبنی بر اینکه تمایل دارد با Bob ارتباط برقرار کند، برای TTP ارسال کند. سپس TTP کلید تصادفی k_{BE} را تولید می‌کند و پیام‌های $\mathsf{Enc}_{k_E}(k_{BE})$ و $\text{ticket} = \mathsf{Enc}_{k_B}(k_{BE})$ را برای Eve ارسال می‌کند. Eve با استفاده از کلید خصوصی‌اش می‌تواند کلید k_{BE} را محاسبه کند. سپس، Eve با استفاده از گذرانه‌ای که از TTP دریافت کرده‌است خود را به‌جای Alice به Bob معرفی می‌کند و کلید k_{BE} را با Bob به اشتراک می‌گذارد. این یک حمله فعال محسوب می‌شود که به آن حمله جعل هویت[۱۲] گفته می‌شود. برای مقاوم کردن پروتکل فوق در برابر حمله جعل هویت می‌توان در پیام‌های رمزی ارسالی توسط TTP به فرد درخواست کننده ارتباط، هویت افرادی را که TTP برای آنها کلید مشترک تولید می‌کند وارد کرد. شکل زیر نحوه این محاسبه را نشان می‌دهد.

[۹] passive
[۱۰] eavesdropping
[۱۱] active attacks
[۱۲] impersonation attack



نکته ۱ *پروتکل تبادل کلید به کمک* TTP*، دارای دو خاصیت است:*

- *برای هر تبادل کلید در شبکه،* TTP *مورد نیاز است.*
- TTP *همه‌ی کلیدها (کلیدهای خصوصی کاربران و کلیدهای امن مشترک بین هر دو کاربر) را می‌داند.*

این خصوصیات باعث می‌شود که اگر TTP *تسخیر شود، مهاجم به راحتی بتواند به تمامی کلیدهای مشترک بین کاربران دست یابد.*

۲ تبادل کلید بدون اشتراک کلید

راهکارهایی که در بخش قبل ارائه شدند مسأله تبادل کلید را به طور کامل مرتفع نمی‌کنند زیرا کاربران باید از قبل کلید یا کلیدهایی را با یکدیگر و یا یک شخص ثالث معتمد به اشتراک بگذارند. برای تبادل کلید، پروتکل‌های دیگری هم وجود دارند که در آنها نیاز به اشتراک کلید از قبل و یا وجود شخص ثالث معتمد نیست که در ادامه به معرفی آنها می‌پردازیم. این راه حل‌ها برای مهاجم منفعل امن است. مهاجم فعال می‌تواند حمله‌ای موسوم به حمله‌ی مرد میانی[۱۳] را علیه هر پروتکل تبادل کلیدی اعمال نماید که در انتها توضیح داده خواهد شد.

تاریخچه شایان ذکر است که امکان وجود روشی برای تبادل کلید به هیچ‌وجه بدیهی نیست. طرح این سؤال و راه حل‌هایی که به دنبال آن ارائه شد، نقطه عطفی در تاریخ علم رمزشناسی محسوب می‌شود که منجر به شکل‌گیری رمزنگاری نوین شد. این مسأله اولین بار در جامعه علمی توسط مرکل در سال ۱۹۷۴ هنگامی که دانشجوی کارشناسی ارشد بود مطرح و راه حلی که اکنون به پازل مرکل معرف است برای آن ارائه شد. اما راه حل وی در سال ۱۹۷۸ منتشر شد؛ پازل مرکل دو طرفی را که هرگز یکدیگر را ملاقات نکرده‌اند قادر می‌سازد که با ارسال اطلاعات در زمان $O(n)$ کلیدی را به اشتراک گذارند. ولی مهاجمی که مکالمه آنها را شنود کرده است، در زمان $O(n^2)$ قادر به کشف کلید مشترک است. هرچند راه حل مرکل چندان عملی نیست اما سوال وی پروتکل دیفی-هلمن را در سال ۱۹۷۶ برای جامعه رمزنگاری به ارمغان آورد. پروتکل دیفی-هلمن، تبادل کلید را برای دو طرف در در زمان چندجمله‌ای امکان پذیر می‌کند، اما تحت فرض‌هایی کاملاً پذیرفته شده مهاجم به زمان نمایی یا زیرنمایی برای محاسبه کلید نیاز دارد. یکسال بعد، ایده سیستم رمز کلیدعمومی و امضای دیجیتال توسط ریوست، شامیر و آدلمن تحت سیستمی که

[۱۳] man-in-the-middle attac

اکنون به RSA معروف است، مطرح شد. جالب است بدانید مسأله تبادل کلید و سیستم رمز کلید عمومی از سال ۱۹۶۰ ذهن ریاضیدانان و رمزنگاران ستاد ارتباطات دولت انگلیس[۱۴] را به خود مشغول کرده است و قبل از سال ۱۹۷۵ به کشف داخلی سیستم‌هایی مشابه دیفی-هلمن و RSA منجر شده است. اسناد مربوط تنها در سال ۱۹۹۷ انتشار عمومی یافت.

۱.۲ پازل مرکل

هدف: Alice و Bob می‌خواهند مستقیماً و با استفاده از تکنیک‌های رمزنگاری متقارن، یک کلید خصوصی به اشتراک گذارند، که هیچ مهاجم منفعلی نتواند به آن دست پیدا کند.
برای دستیابی به این هدف، در ادامه، پروتکل پازل مرکل[۱۵] را معرفی می‌نماییم که ابزار اصلی مورد استفاده در آن، پازل‌ها هستند.

تعریف ۱ *پازل مسئله‌ای است که حل آن دشوار است، ولی توسط یک سری تلاش‌ها قابل حل است.*

مثال ۲ *فرض کنید $\mathsf{Enc}_k(\cdot)$، یک الگوریتم رمزنگاری متقارن دارای امنیت متن اصلی انتخابی (CPA-امن) با کلید ۱۲۸ بیتی k باشد. کلید انتخابی ما برای این رمز متقارن دارای این ویژگی است که ۹۶ بیت اول آن صفر است و ۳۲ بیت آخر آن (که جواب پازل ماست و با P نشان داده می‌شود) به شکل تصادفی انتخاب می‌شود؛ یعنی: $k = 0^{96}||P$. در اینجا $\mathsf{puzzle} = \mathsf{Enc}_k(m)$ یک پازل است و هدف یافتن جواب آن، P، با فرض داشتن اطلاعاتی در باره پیام m و با بررسی 2^{32} حالت ممکن است. بدین منظور کافی است متن رمزی (پازل) puzzle را توسط هر یک از 2^{32} کلید ممکن رمزگشایی کنیم و هرگاه نتیجه با متن اولیه سازگار شد، کلید را یافته‌ایم. برای اینکه تنها یکی از 2^{32} کلید ممکن با متن اصلی سازگار باشد، پیام m باید به اندازه‌ی کافی دارای افزونگی باشد. بعنوان مثال، اگر m پیامی باشد که با t تا صفر شروع شده باشد، بطور متوسط حدود $2^{32-t} + 1$ کاندیدا برای P پیدا خواهد شد. بنابراین، اگر $t = 128$ باشد، با احتمال بسیار زیاد تنها یک گزینه برای P پیدا خواهد شد که همان جواب درست پازل است.*

تعریف ۳ *(پازل مرکل) با توجه به مثال بالا، تبادل کلید با استفاده از پروتکل پازل مرکل به صورت زیر انجام می‌شود:*

- *ابتدا Alice، 2^{32} پازل را فراهم می‌کند. نحوه‌ی ساخت پازل‌ها به این صورت است که Alice، به ازای هر $i = 1, ..., 2^{32}$، رشته‌های تصادفی $P_i \in \{0,1\}^{32}$ و $x_i, k_i \in \{0,1\}^{128}$ را انتخاب می‌کند و پازل i-ام را به شکل زیر می‌سازد:*

$$\mathsf{puzzle}_i \leftarrow \mathsf{Enc}_{0^{96}||P_i}(0^{128}||x_i||k_i)$$

که x_i نشان‌دهنده‌ی شماره‌ی پازل و k_i کلید متناظر با پازل شماره i است. در نهایت، $\mathsf{puzzle}_1, ..., \mathsf{puzzle}_{2^{32}}$ برای Bob ارسال می‌شوند.

- *Bob، 2^{32} پازل را دریافت می‌کند. سپس، به صورت تصادفی پازل j-ام یعنی puzzle_j را انتخاب کرده و حل می‌کند. بنابراین، (x_j, k_j) را بدست می‌آورد و در انتها x_j را برای Alice می‌فرستد.*

[۱۴] Government Communications Headquarters (GCHQ)
[۱۵] merkle puzzle

- Alice پازل شماره‌ی x_j را پیدا می‌کند و از کلید متناظر با آن یعنی k_j به عنوان کلید خصوصی مشترک استفاده می‌کند.

نکته ۲ در حالت کلی اگر Alice، n پازل بسازد، کاری که انجام می‌دهد از $\mathcal{O}(n)$ است. Bob نیز باید یکی از n پازل دریافتی را انتخاب کرده و آن را حل کند. اگر هریک از پازل‌ها در زمان $\mathcal{O}(n)$ قابل حل باشند، کاری که Bob انجام می‌دهد هم از $\mathcal{O}(n)$ است. اما مهاجم شنودگری که قصد حمله به این پروتکل را داشته باشد، چون نمی‌داند که x_j شماره‌ی کدام پازل است، باید n پازلی را که Alice به Bob ارسال می‌کند حل کند. بنابراین کاری که مهاجم برای حمله صورت می‌دهد، از $\mathcal{O}(n^۲)$ خواهد بود. در تعریفی که در بالا ارائه دادیم $n = ۲^{۳۲}$ است؛ پس مهاجم با $۲^{۶۴}$ رمزگشایی، می‌تواند به پروتکل حمله کند که می‌توان فرض کرد امروزه قابل دسترسی نیست.

۲.۲ پروتکل دیفی-هلمن

تعریف ۴ (پروتکل تبادل کلید دیفی- هلمن[۱۶]) تبادل کلید با استفاده از پروتکل دیفی- هلمن بین Alice و Bob به صورت زیر انجام می‌شود:

- Alice با دریافت پارامتر امنیتی ۱^n، مولد $\mathsf{GroupGen}(۱^n)$ را برای بدست آوردن $(\mathbb{G}, q, g)$ اجرا می‌کند. سپس $x \leftarrow \mathbb{Z}_q$ را بطور یکنواخت و تصادفی انتخاب می‌کند و $h_۱ = g^x$ را محاسبه می‌نماید و $(\mathbb{G}, q, g, h_۱)$ را برای Bob ارسال می‌کند.
- Bob، $(\mathbb{G}, q, g, h_۱)$ را دریافت کرده و $y \leftarrow \mathbb{Z}_q$ را بطور یکنواخت و تصادفی انتخاب می‌کند. سپس، $h_۲ = g^y$ را محاسبه کرده و به Alice ارسال می‌کند و در نهایت، کلید $k_B = h_۱^y$ را به خروجی می‌دهد.
- Alice، $h_۲$ را دریافت کرده و کلید $k_A = h_۲^x$ را تولید می‌کند.

نحوه‌ی تبادل کلید در پروتکل دیفی- هلمن، در شکل زیر نشان داده شده است:

Alice		Bob
$x \leftarrow \mathbb{Z}_q$		
$h_۱ = g^x$	$\longrightarrow$	
		$y \leftarrow \mathbb{Z}_q$
	$\longleftarrow$	$h_۲ = g^y$
$k_A = h_۲^x$		$k_B = h_۱^y$

بررسی این مسئله که این پروتکل صحیح است کار دشواری نیست؛ چرا که Alice و Bob به ترتیب کلیدهای k_B و k_A را به شکل زیر محاسبه می‌کنند:

$$k_B = h_۱^y = (g^x)^y = g^{xy}$$

و

$$k_A = h_۲^x = (g^y)^x = g^{xy}$$

بوضوح می‌بینیم که $k_B = k_A$.

[۱۶] the Diffie-Hellman protocol

امنیت پروتکل دیفی-هلمن در نگاه اول ممکن است به نظر برسد که مهاجم منفعل برای بدست آوردن کلید مشترک در پروتکل دیفی-هلمن باید بتواند از روی g^x مقدار x را محاسبه کند. این مسأله تحت عنوان مسأله لگاریتم گسسته شناخته می‌شود. اگر مسأله لگاریتم گسسته برای گروه مورد استفاده در پروتکل دیفی-هلمن آسان باشد، پروتکل در برابر مهاجم منفعل امن نیست زیرا به راحتی می‌تواند کلید مشترک را محاسبه کند.

تعریف ۵ *(فرض لگاریتم گسسته[۱۷]) می‌گوییم فرض لگاریتم گسسته (یا DL) برای گروه* GroupGen *برقرار است (یا مسئله‌ی* DL *سخت است) اگر هیچ مهاجم کارایی نتواند از روی* $(\mathbb{G}, q, g, g^x)$ *مقدار* x *را وقتی که* x *به تصادف از* $\mathbb{Z}_q$ *تولید شده باشد، با احتمال قابل توجهی محاسبه کند.*

اگر با دقت بیشتری به پروتکل دیفی-هلمن نگاه کنیم متوجه خواهیم شد که مهاجم منفعل برای بدست آوردن کلید مشترک لزما نباید مسأله لگاریتم گسسته را حل کند. مسأله‌ای که مهاجم با آن مواجه است محاسبه g^{xy} از روی g^x و g^y است وقتی که x و y به تصادف از $\mathbb{Z}_q$ تولید شده باشد. این مسأله تحت عنوان مسأله دیفی-هلمن محاسباتی شناخته می‌شود.

تعریف ۶ *(فرض دیفی- هلمن محاسباتی[۱۸]) می‌گوییم فرض دیفی- هلمن محاسباتی (یا* CDH*) برای گروه* GroupGen *برقرار است (یا مسئله‌ی* CDH *سخت است) اگر هیچ مهاجم کارایی نتواند از روی* $(\mathbb{G}, q, g, g^x, g^y)$ *مقدار* g^{xy} *را وقتی که* x *و* y *به تصادف از* $\mathbb{Z}_q$ *تولید شده باشد، با احتمال قابل توجهی محاسبه کند.*

برای تعیین امنیت پروتکل دیفی-هلمن در برابر مهاجم منفعل باید مفهوم امنیت را به طور دقیق برای یک پروتکل تبادل کلید تعریف کرد که این کار را در بخش بعدی انجام می‌دهیم. اما به طور شهودی در یک پروتکل تبادل کلید امن (در برابر مهاجم منفعل)، حمله‌کننده نباید اطلاعاتی درباره‌ی کلید به اشتراک گذاشته شده کسب کند. برای اثبات امنیت پروتکل دیفی-هلمن به فرض زیر نیاز داریم.

تعریف ۷ *(فرض دیفی- هلمن تصمیمی[۱۹]) می‌گوییم فرض دیفی- هلمن تصمیمی (یا* DDH*) برای گروه* GroupGen *برقرار است (یا مسئله‌ی* DDH *سخت است) اگر دو توزیع* $(\mathbb{G}, q, g, g^x, g^y, g^{xy})$ *و* $(\mathbb{G}, q, g, g^x, g^y, g^z)$ *وقتی که* x, y, z *به تصادف از* $\mathbb{Z}_q$ *تولید شده باشد، تمایزناپذیرند.*

۳.۲ تبادل کلید با سیستم رمز کلید عمومی

یک راه حل دیگری برای حل مسأله تبادل کلید استفاده از یک سیستم رمز کلید عمومی است ... .

۳ حمله مرد میانی

همه پروتکل‌های مطرح شده تنها در برابر مهاجم منفعل امن هستند. اگر ارتباط دو طرف از طریق یک کانال قابل اعتماد[۲۰] انجام شود، مهاجم نمی‌تواند به صورت فعال عمل کند و پیام‌های ارسالی دو طرف را تغییر دهد. یک راه تبدیل یک کانال غیرقابل اعتماد به کانال قابل اعتماد، استفاده از کدهای اصالت‌سنجی پیام است. اما این خود

[۱۷] discrete logarithm assumption
[۱۸] Computational Diffie-Hellman
[۱۹] Decisional Diffie-Hellman
[۲۰] authentic

نیازمند این است که دو طرف یک کلید محرمانه به اشتراک بگذارند. راه حل دیگر استفاده از امضای دیجیتال است که در آن یکی از دو طرف باید کلید عمومی طرف مقابل را به طریقی به دست آورده باشد. اگر کانال قابل اعتماد نباشد، مهاجم فعال قادر به اعمال حمله مرد میانی خواهد بود. مرد میانی خود را به جای Alice برای Bob جا می‌زند و بالعکس. به عنوان مثال پروتکل دیفی-هلمن را در نظر بگیرید ...

۴ تعریف امنیت برای پروتکل تبادل کلید

امنیت در برابر مهاجم منفعل را برای پروتکل تبادل کلید می‌توان با استفاده از آزمایش زیر تعریف کرد.

تعریف ۸ (*آزمایش تبادل کلید*[۲۱]) $\mathsf{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$

۱. *طرفین مبادله با تعیین پارامتر امنیتی* 1^n *، پروتکل* Π *را اجرا می‌کنند. پس از اجرا، خروجی یک رونوشت*[۲۲] *مانند* trans *شامل همه‌ی پیام‌های ارسالی طرفین، و یک کلید* $k \in \{0,1\}^n$ *است:*

$$(\mathrm{trans}, k) \leftarrow \Pi(1^n)$$

۲. *بیت* $b \in \{0,1\}$ *و کلید* $k_1 \in \{0,1\}^n$ *بطور کاملاً تصادفی انتخاب می‌شود:*

$$b \leftarrow \{0,1\}$$

$$k_1 \leftarrow \{0,1\}^n$$

۳. *مهاجم* $\mathcal{A}$*، رونوشت* trans *و کلید* k_b *را دریافت کرده و بیت* $\hat{b}$ *را تولید می‌کند:*

$$\hat{b} \leftarrow \mathcal{A}(\mathrm{trans}, k_b)$$

خروجی این آزمایش، با متغیر تصادفی $\mathsf{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n)$ *نشان داده می‌شود و برابر ۱ تعریف می‌شود اگر* $\hat{b} = b$ *باشد.*

تعریف ۹ (*امنیت پروتکل تبادل کلید*) *پروتکل تبادل کلید* Π*، در برابر مهاجم منفعل امن است، اگر برای هر مهاجم احتمالاتی* $\mathcal{A}$ *که در زمان چندجمله‌ای عمل می‌کند، تابع ناچیز* $\varepsilon(n)$ *موجود باشد، بطوریکه:*

$$\Pr\{\mathsf{KE}^{\mathrm{eav}}_{\mathcal{A},\Pi}(n) = 1\} \leq \frac{1}{2} + \varepsilon(n)$$

امنیت پروتکل تبادل کلید در برابر مهاجم فعال به صورت پیچیده‌تری تعریف می‌شود.

قضیه ۱ *اگر مسئله‌ی* DDH *نسبت به* GroupGen *سخت باشد، آنگاه پروتکل تبادل کلید دیفی- هلمن* Π*، در حضور مهاجم شنودگر امن است.*

[۲۱] key-exchange experiment
[۲۲] transcript