امنیت داده و شبکه

طراحی پروتکل‌های رمزنگاری

فهرست

- جایگاه رمز متقارن
- مدیریت کلید
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

مقدمه

- **پروتکل‌های رمزنگاری:** پروتکل‌هایی که در آنها از الگوریتم‌های رمز استفاده می‌شود.
  - پروتکل‌های احراز اصالت همراه با توزیع کلید
- برای طراحی پروتکل‌های رمزنگاری نیاز است به:
  - تبیین جایگاه رمزنگاری متقارن و نامتقارن
  - تبیین نحوه تولید کلید و توزیع آن

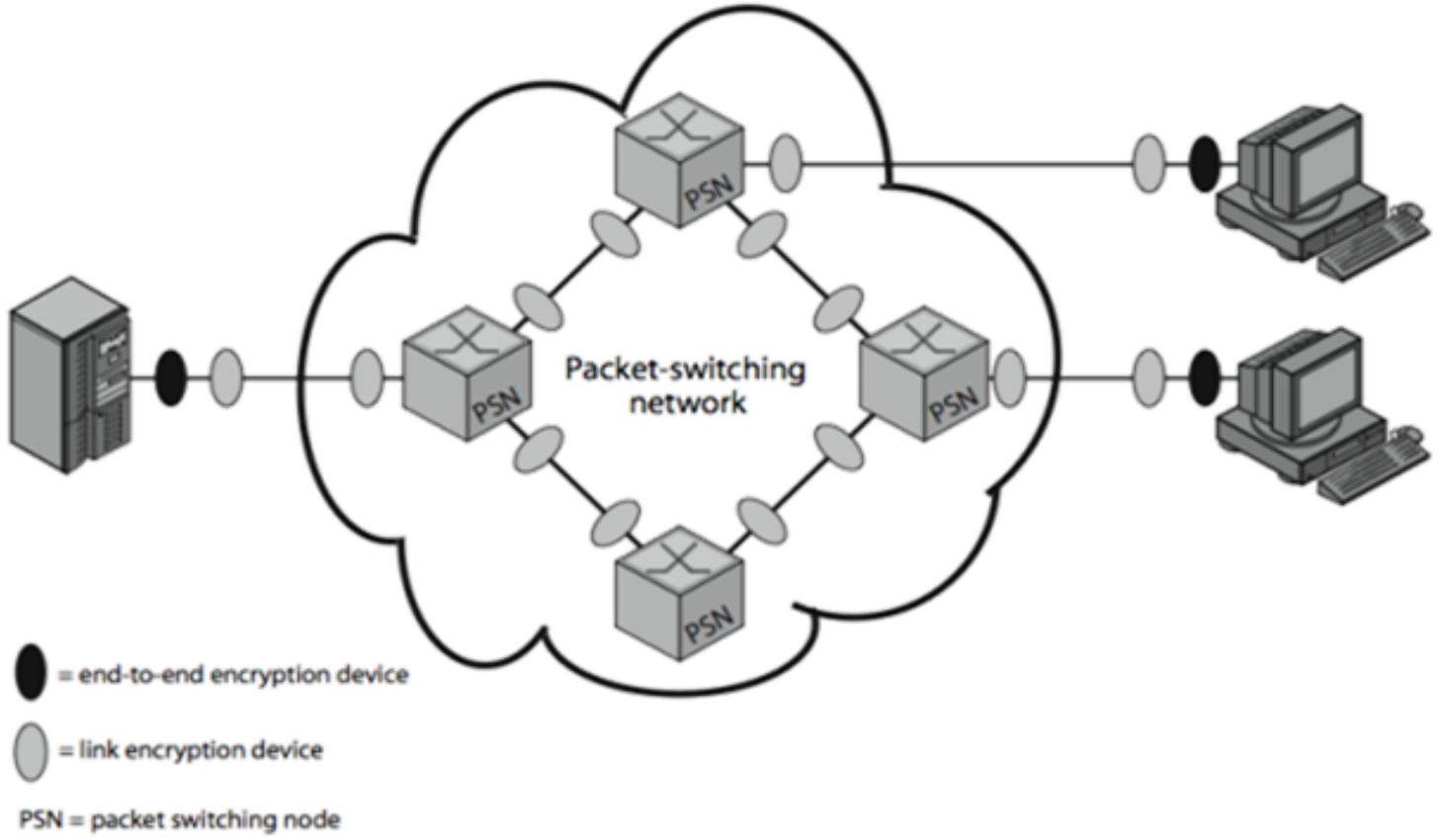
فهرست

- **جایگاه رمز متقارن**
- مدیریت کلید
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

جایگاه رمز متقارن

- دو رویکرد در استفاده از رمز متقارن در شبکه:
  - رمزنگاری خط ارتباطی به صورت نقطه-به-نقطه (Point-to-Point)
    - رمزگذاری روی هر خط ارتباطی به صورت مستقل صورت می‌پذیرد.
    - باید در هر یک از تجهیزات ارتباطی رمزگشایی شود.
    - نیازمند تجهیزات متعددی است، هر کدام با کلیدهای مجزا.
  - رمزنگاری انتها-به-انتها (End-to-End)
    - رمزگذاری صرفاً بین مبدا و مقصد پیام انجام می‌شود.
    - نیازمند یک کلید مشترک بین دو انتها است.

جایگاه رمز متقارن



جایگاه رمز متقارن

- در رمزگذاری انتها-به-انتها، سرآیند بسته‌ها باید آشکار باقی بمانند.
  - لذا شبکه به راحتی می‌تواند بسته‌ها را مسیریابی کند.
- بنابراین اگر چه محتوا حفاظت می‌شود، ولی الگوی ترافیک و جریان داده‌ها آشکار است.
- به طور ایده‌آل می‌خواهیم:
  - رمزگذاری انتها-به-انتها محتوای داده‌ها را بر روی کل مسیر حفاظت نماید و امکان احراز اصالت داده را نیز فراهم آورد.
  - رمزگذاری خط ارتباطی (نقطه-به-نقطه) جریان داده را از مانیتورینگ حفاظت نماید.

جایگاه رمز متقارن

- تابع رمزگذاری را می‌توان در هر یک از لایه‌های شبکه در مدل مرجع OSI قرار داد.
  - رمزگذاری ارتباط (نقطه به نقطه) معمولا در لایه‌های ۱ و ۲ انجام می‌پذیرد.
  - رمزگذاری انتها به انتها در لایه‌های بالاتر.
- هر چه قدر به لایه‌های بالاتر شبکه برویم،
  - اطلاعات کمتری رمز می‌شود، ولی امنیت بیشتری فراهم می‌گردد.
  - پیچیدگی، بیشتر و همچنین موجودیت‌ها و کلیدهای درگیر، بیشتر می‌شود.

(a) Application-level encryption (on links and at routers and gateways)

On links and at routers

In gateways

(b) TCP-level encryption

On links

In routers and gateways

(c) Link-level encryption

Shading indicates encryption.

TCP-H	=	TCP header
IP-H	=	IP header
Net-H	=	Network-level header (e.g., X.25 packet header, LLC header)
Link-H	=	Data link control protocol header
Link-T	=	Data link control protocol trailer

رمزگذاری در مقابل سطح پروتکل

تحلیل ترافیک

- تحلیل ترافیک به معنای مانیتورینگ جریان داده‌ها در ارتباطات بین بخش‌های مختلف است.
  - هم در بخش نظامی و هم در بخش تجاری می‌تواند مفید باشد.
- رمزگذاری خط ارتباطی می‌تواند جزئیات سرآیند را مخفی کند.
  - اما حجم ترافیک شبکه و داده‌ها در دو انتهای ارتباط همچنان آشکار است.
- لایی‌گذاری (Padding) در ترافیک نیز می‌تواند جریان داده‌ها را ناشفاف نماید، ولی هزینه سربار بالایی دارد.

فهرست

- جایگاه رمز متقارن
- **مدیریت کلید**
  - **مفاهیم اساسی مدیریت کلید**
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

مدیریت کلید چیست؟

- **مدیریت کلید** عبارتست از مجموعه‌ای از پروتکلها و رویه‌ها برای ایجاد و نگهداری «ارتباط کلیدی» بین طرفین مجاز.
- **ارتباط کلیدی** وضعیتی است که در آن طرفین برقرار کننده ارتباط داده معینی را به اشتراک می‌گذارند که مورد نیاز الگوریتم‌های رمز است.
  - کلیدهای عمومی یا خصوصی،
  - مقداردهی‌های اولیه،
  - سایر پارامترهای غیرمخفی...

مدیریت کلید شامل چه رویه هایی است؟

- تولید، توزیع و نصب داده‌های ارتباط کلیدی
- کنترل نحوه استفاده از این کلیدها
- به روزآوری، ابطال و نابود کردن داده‌های ارتباط کلیدی
- نگهداری، نسخه برداری و بازیابی داده‌های ارتباط کلیدی

اهمیت مدیریت کلید

- اکثر حملات به رمزنگاری یک سیستم امنیتی در لایه مدیریت کلید است و کمتر به الگوریتم‌هایی است که از کلیدها (داده‌های مشترک) بهره می‌برند.
- طرفهای ارتباط به طور دائم، امکان ارتباط فیزیکی برای تبادل کلید امن را با یکدیگر ندارند و از پروتکلهای توزیع کلید استفاده می‌کنند.
- در حقیقت برخی این مساله را دشوارترین جزء یک سیستم امن می‌دانند.

فهرست

- جایگاه رمز متقارن
- **مدیریت کلید**
  - مفاهیم اساسی مدیریت کلید
  - **سلسله مراتب کلید**
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

کلید جلسه و کلید اصلی: توصیف

- **کلید اصلی** عبارت است از یک کلید رمزکننده سایر کلیدها. به این معنا که از این کلید برای توزیع کلید خصوصی موقتی به نام **کلید جلسه** استفاده می‌نماییم.

- از کلید جلسه برای رمزنگاری و احراز صحت یا اصالت استفاده می‌کنیم.
  - رمزنگاری متقارن

سلسله مراتب کلیدها

نحوه محافظت	حجم اطلاعات	نوع
محافظت با رمزنگاری **(با کلید جلسه)**		داده
محافظت با رمزنگاری **(با کلید اصلی)**		کلید جلسه
محافظت فیزیکی		کلید اصلی

کلید جلسه و کلید اصلی: مقایسه

- **کلید اصلی:**
  - طول عمر نسبتاً زیاد،
  - میزان استفاده محدود (فقط رمزنگاری کلیدهای جلسه)،
  - خسارت گسترده درصورت افشاء.
- **کلید جلسه:**
  - طول عمر نسبتاً کوتاه،
  - استفاده نامحدود در طول جلسه،
  - خسارت محدود به داده‌های جلسه.

فهرست

- جایگاه رمز متقارن
- **مدیریت کلید**
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - **تولید کلید و طول عمر کلید**
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

تولید کلید

- کلیدهای تولیدی باید کاملا تصادفی باشند و از آنتروپی (بی‌نظمی) کافی برخوردار باشند.
- نیاز به تولید اعداد تصادفی داریم، به گونه‌ای که:
  - به طور آماری، تصادفی باشند، با توزیع یکنواخت و مستقل از یکدیگر،
  - امکان پیش‌بینی مقادیر آتی بر اساس مقادیر فعلی وجود نداشته باشد.
- عموماً از روشهای الگوریتمی برای تولید اعداد تصادفی استفاده می‌شود.
  - به طور واقعی تصادفی نیستند.
  - به عنوان اعداد **شبه تصادفی** شناخته می شوند.

طول عمر کلید جلسه

- اگر طول عمر کوتاه باشد:
  - امنیت بالا
    - حجم داده برای تحلیل رمز ناچیز است.
    - میزان استفاده کم است.
    - حتی پس از افشای کلید، زمان زیادی برای سوء استفاده موجود نیست.
  - کارایی کم
    - دائما باید کلید را بروز کنیم.
- اگر طول عمر زیاد باشد:
  - کارایی بالا، امنیت کم

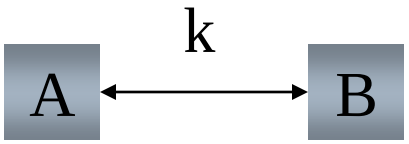
یک مصالحه میان امنیت و کارایی بر سر تعیین طول عمر کلید جلسه برقرار است.

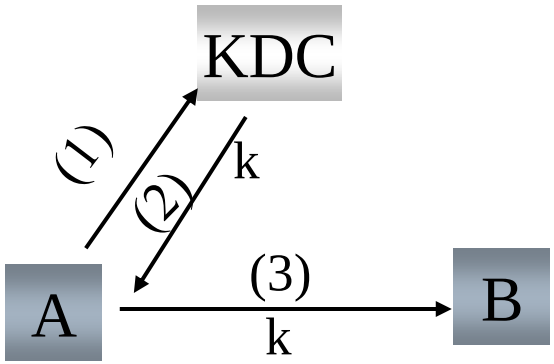
فهرست

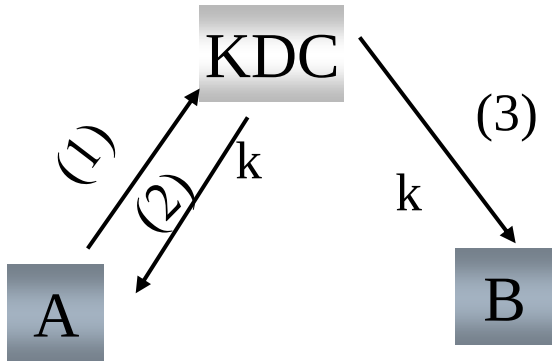
- جایگاه رمز متقارن
- **مدیریت کلید**
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - **اشتراک کلید مبتنی بر رمز متقارن**
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- طراحی پروتکل‌های رمزنگاری

اشتراک کلید مبتنی بر رمز متقارن

- دو رویکرد در اشتراک کلید جلسه
  - نقطه به نقطه
  - مرکز توزیع کلید

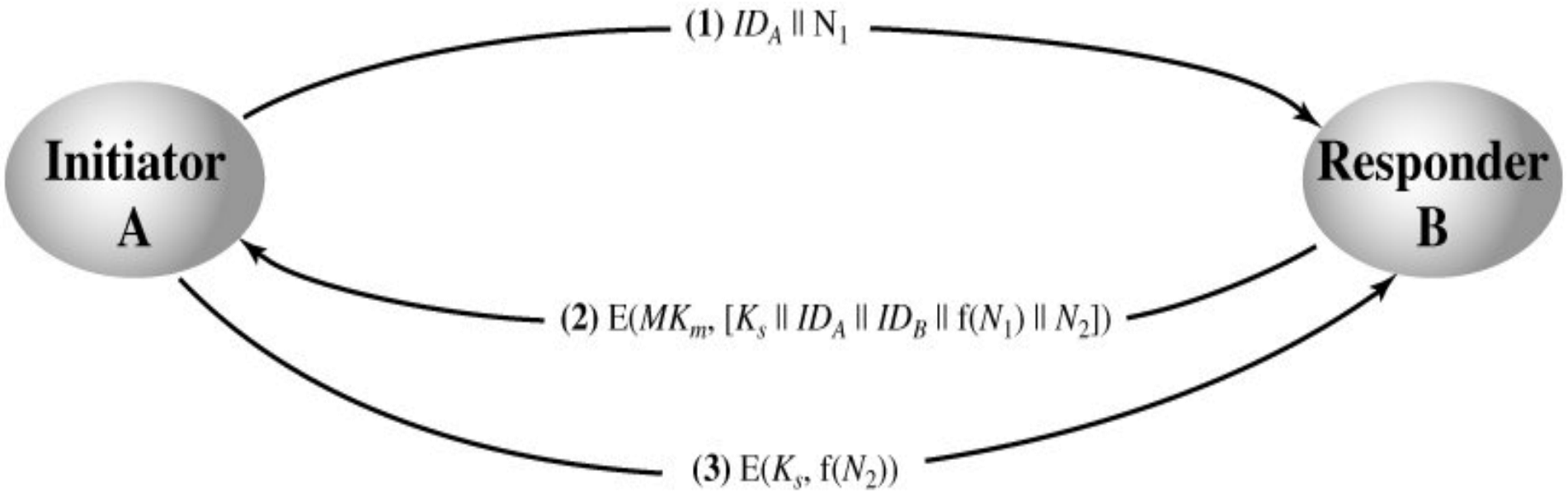






روش نقطه به نقطه توزیع کلید

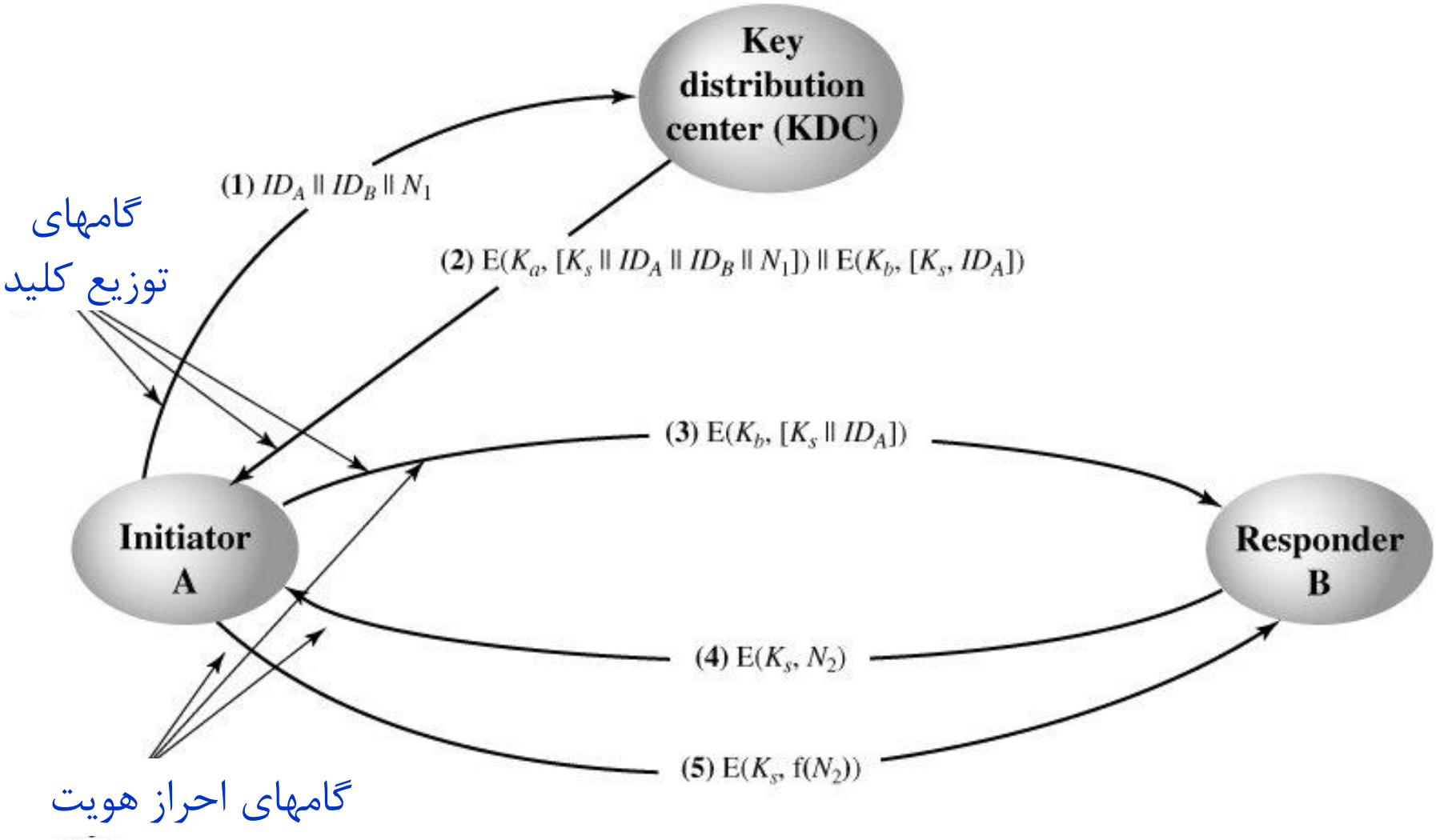
- نیاز به توافق بر روی کلید اصلی پیش از برقراری ارتباط بین هر دو نفر
- مشکل اصلی: مقیاس‌پذیری
  - برای ارتباط n نفر باهم به n(n-1)/2 کلید اصلی احتیاج داریم.



روش متمرکز توزیع کلید

- هر کاربر یک کلید اصلی با کارگزار توزیع کلید KDC به اشتراک گذاشته است.
  - KDC یک شخص ثالث مورد اعتماد است.
  - کلیدهای اصلی با یک روش امن (مثلاً مراجعه فیزیکی) توزیع شده‌اند.
- ایده:
  - هربار که کاربری قصد ارتباط با دیگران را داشته باشد از KDC یک کلید جلسه درخواست می‌کند.

روش متمرکز توزیع کلید - مثال



روش متمرکز توزیع کلید

- نکات مثبت:
  - تعداد کلید کمتر و قابلیت مقیاس‌پذیری
- نکات منفی:
  - کارگزار توزیع کلید گلوگاه امنیتی سیستم است.
  - ترافیک بالا در کارگزار توزیع کلید باعث تبدیل آن به گلوگاه کارایی سیستم می‌شود.
  - نیاز به یک کارگزار برخط داریم.
    - دخالت کارگزار در برقراری هر ارتباط ضروری است.

فهرست

- جایگاه رمز متقارن
- **مدیریت کلید**
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - **اشتراک کلید مبتنی بر رمز کلید عمومی**
- طراحی پروتکل‌های رمزنگاری

جایگاه رمزنگاری کلید عمومی

- از آنجا که الگوریتم‌های کلید عمومی بسیار کندتر از الگوریتم‌های مرسوم (کلید خصوصی) هستند، از این الگوریتم‌ها جهت توزیع کلید جلسه (و نه رمزگذاری) استفاده می‌شود.
- با استفاده از رمزنگاری کلید عمومی
  - نیازی به تبادل فیزیکی کلیدهای اصلی و حفظ محرمانگی آنها نیست.
  - نیازی به کارگزار بر خط نیست.

اشتراک کلید مبتنی بر رمز نامتقارن

- **توافق کلید (Key Agreement)**: بنا نهادن دو جانبه کلید جلسه
  - طرفین به طور مستقل در انتخاب کلید تاثیرگذار هستند.
    - مثال: روش Diffie-Hellman (قبلاً در رمز نامتقارن معرفی شد).

- **توزیع کلید (Key Distribution)**: توزیع یک جانبه کلید جلسه
  - یکی از دو طرف کلید را معین کرده و به دیگری ارسال می‌نماید.
    - مثال: الگوریتم توزیع کلید در SSL (در درس‌های بعدی معرفی می‌شود).

روش ترکیبی

- کلید عمومی+رمزنگاری متقارن
- توزیع مداوم کلید با رمزنگاری کلید عمومی کارآیی سیستم را کاهش می‌دهد.
  - با کمک روش ترکیبی به طور موردی از رمزنگاری کلید عمومی برای به‌روز درآوردن کلید اصلی بهره می‌جوییم.
- کلیدها در سه سطح:
  - استفاده از **کلید عمومی** برای توزیع کلیدهای اصلی
  - توافق KDC با هر یک از کاربران روی یک **کلید اصلی** (با استفاده کلید عمومی)
  - استفاده از کلید اصلی (رمزنگاری متقارن) برای توزیع **کلیدهای جلسه**

فهرست

- جایگاه رمز متقارن
- مدیریت کلید
  - مفاهیم اساسی مدیریت کلید
  - سلسله مراتب کلید
  - تولید کلید و طول عمر کلید
  - اشتراک کلید مبتنی بر رمز متقارن
  - اشتراک کلید مبتنی بر رمز کلید عمومی
- **طراحی پروتکل‌های رمزنگاری**

علائم و نمادها

علائم و نمادهای به کار رفته در پروتکل‌ها به صورت زیر هستند.

- عامل‌ها/طرفهای ارتباط
  - A و B با شناسه های ID_A و ID_B
  - S شخص ثالث مورد اعتماد
- کلیدهای مخفی مشترک (کلید جلسه)
  - K_{ab} کلید مشترک بین A و B
- نانس‌ها
  - اعداد تصادفی هستند که تنها یک بار مورد استفاده قرار می‌گیرند.
  - N_a نانس تولید شده توسط A

علائم و نمادها

- مُهر زمانی
  - T_a مُهر زمانی تولید شده توسط A
  - در اغلب موارد فرض می کنیم که طرفهای ارتباط ساعتهای خود را با استفاده از پروتکلی هماهنگ نگه می‌دارند.
- پیام‌های مورد تبادل
  - **مثال:** $A \rightarrow B: M, E(K_{as}, [M \parallel ID_A \parallel ID_B])$
  - A فرستنده و B گیرنده
  - ترکیب M (پیام)، شناسه A و شناسه B با کلید K_{as} رمز شده است.

اهداف و خصوصیات پروتکل‌ها

- فرض می‌کنیم که طرفهای A و B با شخص ثالث S کلید مخفی مشترک (کلید اصلی) و یا گواهی کلید عمومی دارند.
- طرف‌های ارتباط می‌خواهند برای ارتباط با یکدیگر کلید جلسه به اشتراک بگذارند.
- اما چه خصوصیاتی را در اشتراک کلید دنبال می‌کنند؟
  - پروتکل‌های مختلف، خصوصیات مختلفی دارند و اهداف مختلفی را دنبال می‌کنند.
  - بنابراین در انتخاب پروتکل باید دقت کافی را به عمل آورد.

اهداف و خصوصیات پروتکل‌ها

- تازگی (Freshness)
  - کلید جلسه توسط طرف دیگری استفاده نشده باشد و اخیراً تولید شده باشد.
- محرمانگی پیشرو (Forward Secrecy)
  - با شکستن کلید بلند مدت (اصلی)، اطلاعی در مورد کلیدهای جلسه توافق شده قبلی مستخرج نشود.
- استحکام در مقابل کلید فاش شده (Known Key Resilience)
  - مهاجمی که به کلید یک جلسه دست یافته، در مورد کلید اصلی و کلید جلسات دیگر نتواند اطلاعی به دست آورد.

اهداف و خصوصیات پروتکل‌ها

- احراز اصالت کلید (Key Authentication)
  - یک طرف مطمئن است که هیچ کس جزء طرف دوم به کلید دسترسی ندارد. این دانش ممکن است صرفاً ضمنی باشد.
- تایید کلید (Key Confirmation)
  - یک طرف مطمئن است که طرف دوم واقعاً کلید مشترک را در اختیار دارد.
- احراز اصالت صریح کلید (Explicit Key Authentication)
  - احراز اصالت ضمنی کلید و تایید کلید

اهداف و خصوصیات پروتکل‌ها

- احراز اصالت دو طرفه
  - هر دو طرف ارتباط باید صحت هویت همدیگر را احراز نمایند و به تبادل کلید بپردازند.

- احراز اصالت یک طرفه
  - لازم است تنها یک طرف ارتباط هویت خود را اثبات کند.
  - مورد استفاده: یک شخص یک پیام را در یک گروه عمومی منتشر می‌کند.

انواع حملات به پروتکل‌ها

- شنود (Eavesdropping)
  - مهاجم اطلاعات و پیامهای تبادل شده در پروتکل را دریافت می‌نماید.
- تغییر (Modification)
  - مهاجم اطلاعات ارسالی را تغییر می‌دهد.
- تکرار (Replay)
  - مهاجم پیامهای ارسالی در طی پروتکل را ثبت نموده، سپس به اجرای پروتکل با ارسال مجدد آنها می‌پردازد.

انواع حملات به پروتکل‌ها

- منع سرویس (Denial of Service)
  - مهاجم مانع از کامل شدن پروتکل توسط طرف‌های مجاز می‌شود.
- حملات نوع داده‌ای (Typing Attacks)
  - مهاجم داده یک فیلد پیام را با داده‌ای از نوع دیگر جایگزین می‌کند.
- دستکاری گواهی (Certificate Manipulation)
  - مهاجم اطلاعات گواهی را دستکاری کرده و یا عوض می‌کند.

طراحی پروتکل

- در اسلایدهای بعد چگونگی طراحی پروتکلی برای اشتراک کلید بین دو طرف A و B را بررسی می‌نماییم.
- با معرفی هر پروتکل، مشکلات موجود در آن را بررسی نموده، سعی می‌کنیم در طراحی پروتکل بعدی آنها را مرتفع نماییم.

طراحی پروتکل

- مبنای طراحی پروتکل‌های سری اول
  - **مبتنی بر رمز متقارن:** استفاده از مرکز توزیع کلید مطمئن (با نام S)
    - S با هر یک از طرف‌های ارتباط یک کلید اصلی (از قبل تبادل شده) دارد.
    - S کلید جلسه را تولید می‌کند.
    - کلیدهای اصلی (بین هر طرف با S) برای انتقال کلید جلسه به کار می‌رود.
  - **احراز اصالت دوطرفه**

پروتکل 1

- **A→S:** $ID_A \| ID_B$
- **S→A:** K_{ab}
- **A→B:** $K_{ab} \| ID_A$

- خصوصیات:
  - S کلیدی را تولید می‌کند و می‌گوید که برای استفاده بین A و B است.
  - B می‌داند که کلید را برای تعامل با A باید استفاده نماید.
- معایب:
  - مهاجم می‌تواند با شنود کلید مخفی K_{ab} را به دست آورد.
- راه حل: نیاز به رمزگذاری کلید داریم.

پروتکل 2

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab}]) \| E(K_{bs}, [K_{ab}])$
- **A→B:** $E(K_{bs}, [K_{ab}]) \| ID_A$

- خصوصیات:
  - لازم است که S یک کلید بلند مدت (همان کلید اصلی) را با A و B به اشتراک بگذارد.

پروتکل 2

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab}]) \| E(K_{bs}, [K_{ab}])$
- **A→B:** $E(K_{bs}, [K_{ab}]) \| ID_A$

- عیب اول: مهاجم می‌تواند خود را به جای طرفهای مختلف وانمود سازد. چرا که در این پروتکل عملا احراز اصالت صورت نمی‌پذیرد.
  - مهاجم D پیام سوم از A به B را دریافت می‌نماید.
  - آن را با $E(K_{bs}, [K_{ab}]) \| ID_D$ جایگزین می‌کند.
  - B فکر می‌کند که کلید K_{ab} را برای تعامل با D باید استفاده کند.
  - یا می‌تواند کلیدی که برای ارتباط بین خود و B از S گرفته به B با شناسه ID_A ارسال کند که B فکر کند با این کلید با A در تعامل است.

پروتکل 2

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab}]) \| E(K_{bs}, [K_{ab}])$
- **A→B:** $E(K_{bs}, [K_{ab}]) \| ID_A$

- عیب دوم: مشکل اصلی این پروتکل در امکان اجرای حمله مرد میانی است که در آن مهاجم (درمیان دو طرف) به جای هر یک از طرفین خود را جا می‌زند و کلید تبادل می‌کند. مثلا برای A، خود را به جای B به صورت زیر جا می‌زند.

- **A→E:** $ID_A \| ID_B$
- **E→S:** $ID_A \| ID_E$
- **S→E:** $E(K_{as}, [K_{ae}]) \| E(K_{es}, [K_{ae}])$
- **E→A:** $E(K_{as}, [K_{ae}]) \| E(K_{es}, [K_{ae}])$
- **A→E:** $E(K_{es}, [K_{ae}]) \| ID_A$

- راه حل: لازم است که شناسه طرفها را به کلیدها مقید نماییم.

پروتکل 3

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B]) \| E(K_{bs}, [K_{ab} \| ID_A])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A])$

- خصوصیات:
  - شناسه طرف ارتباط و کلید جلسه با کلید اصلی رمز می‌شوند.

پروتکل 3

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B]) \| E(K_{bs}, [K_{ab} \| ID_A])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A])$

- معایب: امکان اجرای **حمله تکرار** وجود دارد.
  - فرض کنید که K_{ab} کلید اجرای قبلی پروتکل باشد.
    - کلیدهای کوتاه مدت جلسه به اندازه کلیدهای اصلی بلند مدت امن نگهداری نمی‌شوند.
  - سپس مهاجم به جای S این کلید را به عنوان کلید جلسه جدید با ارسال مجدد پیام دوم از اجرای قبلی توزیع می کند.
    - بدون نیاز به دانستن کلیدهای اصلی K_{as} و K_{bs}
- راه حل: لازم است به گونه‌ای از تازگی کلید اطمینان حاصل نماییم.

روشهای مقابله با تکرار

- **استفاده از مُهر زمانی (Timestamp)**
  - گیرنده به پیام اعتماد می‌کند اگر در محدوده زمانی قابل قبولی باشد. ضرورت همگامی ساعتها!
- **استفاده از Challenge/Response**
  - Y که انتظار یک پیام نو از X دارد، یک چالش یا نانس به X ارسال می‌کند و انتظار دارد که پیامی که دریافت می‌کند حاوی تغییریافته (رمزشده) چالش یا نانس موردنظر باشد.
- **استفاده از اعداد متوالی (Sequence Number)**
  - مشکلات متعددی در خصوص نگهداری این اعداد و عوامل تاثیرگذار بر آن در صورت بروز خطا، تاخیر و غیره دارد.
  - نیازمند احراز اصالت اعداد متوالی ارسالی (برای اطمینان از ارسال آنها از سوی طرف مقابل)- مثلا با استفاده از MAC

پروتکل 4

- **A→S:** $ID_A \| ID_B \| N_a$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B \| N_a \| E(K_{bs}, [K_{ab} \| ID_A])])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A])$

- خصوصیات:
  - تازگی کلید برای A (و نه B) با استفاده از نانس احراز می‌گردد.
- معایب:
  - طرف A مطمئن نیست که طرف B کلید را دریافت کرده و زنده است.
  - طرف B نیز نمی‌داند که واقعاً طرف A کلید را می‌داند و زنده است (ممکن است پیغام سوم دریافتی، قدیمی و تکراری باشد).
- راه حل: نیاز به تایید کلید است.

پروتکل Needham-Schroeder

- **A→S:** $ID_A \| ID_B \| N_a$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B \| N_a \| E(K_{bs}, [K_{ab} \| ID_A])])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A])$
- **B→A:** $E(K_{ab}, N_b)$
- **A→B:** $E(K_{ab}, f(N_b))$

- خصوصیات:
  - دو گام آخر برای تایید کلید (از سوی B) است.

پروتکل Needham-Schroeder

- معایب:
  - این پروتکل نسبت به حمله تکرار آسیب پذیر است.
    - ممکن است کلید جلسه قبلی لو رفته باشد و بتوان جلسه جدیدی را با تکرار از مرحله ۳ تشکیل داد و B در عمل نمی‌تواند از زنده بودن A مطمئن شود.
  - همچنان A نمی‌تواند از زنده بودن B و دریافت کلید توسط آن مطمئن باشد.
    - پیام چهارم عددی تصادفی است (رمز شده یک نانس تصادفی) و به A اطلاع خاصی نمی‌دهد.
- راه حل مقابله با حمله تکرار:
  - تضمین تازگی پیام برای B (علاوه بر A)
    - به طور مثال با اضافه کردن مُهر زمانی به صورتی که در پروتکل بعد آمده است.

پروتکل Denning

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B \| T_s \| E(K_{bs}, [K_{ab} \| ID_A \| T_s])])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A \| T_s])$
- **B→A:** $E(K_{ab}, N_b)$
- **A→B:** $E(K_{ab}, f(N_b))$

- خصوصیات:
  - استفاده از مُهر زمانی برای جلوگیری از حمله تکرار
  - همچنان A از زنده بودن B نمی‌تواند مطمئن شود.

پروتکل Denning

- A و B از طریق زیر به تازه بودن پیام پی می‌برند:

$$|clock - T_s| < \Delta t_1 + \Delta t_2$$

  - $\Delta t_2, \Delta t_1$ به ترتیب اختلاف ساعت محلی با S و میزان تاخیر مورد انتظار در شبکه هستند.

- اگر ساعت فرستنده جلوتر از ساعت گیرنده باشد! مهاجم می‌تواند با ارسال در زمان مربوطه، حمله تکرار (Suppress-replay) داشته باشد!

پروتکل Denning

- حمله Suppress-replay و مقابله با آن
  - پروتکل فوق نسبت به حمله Suppress-replay آسیب‌پذیر است.
    - این حمله از سنکرون نبودن ساعت‌های فرستنده و گیرنده ناشی می‌شود.
    - وقتی ساعت فرستنده جلوتر از ساعت گیرنده باشد، مهاجم می‌تواند پیام‌ها را ذخیره و در زمان مقرر بازارسال نماید.
  - روشهای مقابله :
    - چک کردن متناوب با زمان S
    - توافق از طریق نانس

پروتکل Neuman

- پروتکل بهبودیافته (جهت مقابله با حمله Suppress-Attack)
- **A→B:** $ID_A \| N_a$
- **B→S:** $ID_B \| N_b \| E(K_{bs}, [ID_A \| N_a \| T_b])$
- **S→A:** $E(K_{as}, [ID_B \| N_a \| K_{ab} \| T_b]) \| E(K_{bs}, [ID_A \| K_{ab} \| T_b]) \| N_b$
- **A→B:** $E(K_{bs}, [ID_A \| K_{ab} \| T_b]) \| E(K_{ab}, N_b)$

T_b: time limit on ticket usage

مشکل: A از زنده بودن B اطمینان دارد ولی نمی‌تواند مطمئن شود که B کلید را در اختیار دارد. اگر پیغام آخر به B نرسد، A نمی‌تواند مطلع شود.

طراحی پروتکل

- مبنای طراحی پروتکل‌های سری دوم
  - **مبتنی بر رمز کلید عمومی**
    - کارگزار احراز اصالت (S) علاوه بر توزیع کلید جلسه، وظیفه ایجاد گواهی کلید عمومی را بر عهده دارد.
    - مانند رمزنگاری متقارن، می‌توان از مُهر زمانی یا نانس استفاده کرد.
  - **احراز اصالت دوطرفه**

پروتکل 1

- کلید عمومی و مُهر زمانی

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(PR_s, [ID_A \| PU_a \| T]) \| E(PR_s, [ID_B \| PU_b \| T])$

گواهی کلید عمومی **A** و **B**

- **A→B:** $E(PR_s, [ID_A \| PU_a \| T]) \| E(PR_s, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_{ab} \| T]))$

- معایب: نیاز به همگام بودن زمان سیستم‌های طرفین
  ضمنا A از زنده بودن B و در اختیار داشتن کلید توسط B اطلاعی ندارد (تایید کلید نداریم).

پروتکل 2

- کلید عمومی و نانس

- **A→S:** $ID_A \| ID_B$
- **S→A:** $E(PR_s, [ID_B \| PU_b])$ ⇐ گواهی کلید عمومی **B**
- **A→B:** $E(PU_b, [N_a \| ID_A])$
- **B→S:** $ID_A \| ID_B \| E(PU_s, N_a)$
- **S→B:** $E(PR_s, [ID_A \| PU_a]) \| E(PU_b, E(PR_s, [N_a \| K_{ab} \| ID_A \| ID_B]))$ ⇐ گواهی کلید عمومی **A**
- **B→A:** $E(PU_a, [E(PR_s, [N_a \| K_{ab} \| ID_A \| ID_B]) \| N_b])$
- **A→B:** $E(K_{ab}, N_b)$

طراحی پروتکل

- مبنای طراحی پروتکل‌های سری سوم
  - **احراز اصالت یکطرفه**
    - نمونه‌ای از مورد کاربرد : E-mail
    - نیازمندی‌ها :
      - احراز اصالت فرستنده
      - محرمانگی
  - **راه حل**
    - رمزنگاری متقارن
    - رمزنگاری با کلید عمومی

پروتکل 1

- استفاده از رمز متقارن (برای احراز اصالت یکطرفه در ارسال ایمیل)

- **A→S:** $ID_A \parallel ID_B \parallel N_a$
- **S→A:** $E(K_{as}, [K_{ab} \| ID_B \| N_a \| E(K_{bs}, [K_{ab} \parallel ID_A])])$
- **A→B:** $E(K_{bs}, [K_{ab} \| ID_A]) \parallel E(K_{ab}, M)$

- خصوصیات: گیرنده یکبار پیام را دریافت می‌کند و می‌تواند فرستنده را احراز نماید.
- معایب: امکان حمله تکرار (در مرحله سوم) وجود دارد. این باعث می‌شود که B از زنده بودن A نیز مطمئن نباشد.

پروتکل 2

- استفاده از رمز کلید عمومی (برای احراز اصالت یکطرفه در ارسال ایمیل)
  - هدف: محرمانگی
  - **A→B:** $E(PU_b, K_s) \| E(K_s, M)$
  - هدف: احراز اصالت فرستنده
  - **A→B:** $M \| E(PR_a, H(M))$
  - محرمانگی و احراز اصالت فرستنده
  - **A→B:** $E(PU_b, [M \| E(PR_a, H(M))])$
  - محرمانگی و احراز اصالت فرستنده به صورت کارا
  - **A→B:** $E(PU_b, K_s) \| E(K_s, [M \| E(PR_a, H(M))])$

پایان