



یاد‌الامن والامان

امنیت داده و شبکه

طراحی پروتکل‌های رمزنگاری

مرتضی امینی - سیدمهدی خرازی

نیمسال اول ۱۴۰۴-۱۴۰۳



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 پروتکل های رمزنگاری



فهرست

□ جایگاه رمز متقارن

□ مدیریت کلید

■ مفاهیم اساسی مدیریت کلید

■ سلسه مراتب کلید

■ تولید کلید و طول عمر کلید

■ اشتراک کلید مبتنی بر رمز متقارن

■ اشتراک کلید مبتنی بر رمز کلید عمومی

□ طراحی پروتکل های رمزنگاری



مقدمه

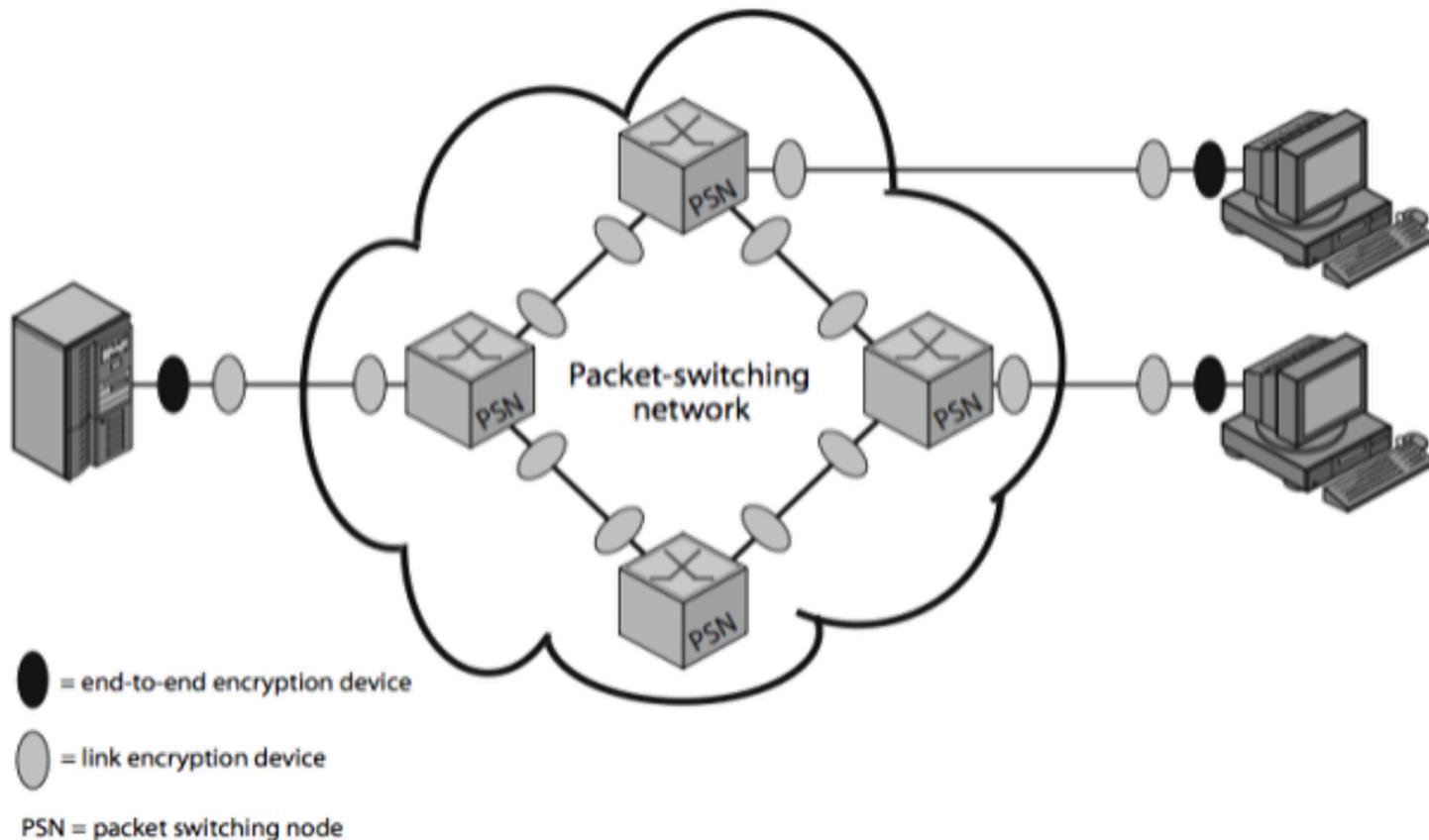
- پروتکل‌های رمزنگاری: پروتکل‌هایی که در آنها از الگوریتم‌های رمز استفاده می‌شود.
- پروتکل‌های احراز اصالت همراه با توزیع کلید
- برای طراحی پروتکل‌های رمزنگاری نیاز است به:
 - تبیین جایگاه رمزنگاری متقارن و نامتقارن
 - تبیین نحوه تولید کلید و توزیع آن



جایگاه رمز متقارن در شبکه

- 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 انتهای-به-انتهای (End-to-End)
 - رمزگذاری صرفاً بین مبدأ و مقصد پیام انجام می‌شود.
 - نیازمند یک کلید مشترک بین دو انتهای است.

جایگاه رمز متقارن در شبکه





جایگاه رمز متقارن در شبکه

- در رمزگذاری انتهای-به-انتهای، 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 انتهای-به-انتهای 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 در شبکه

- تابع رمزگذاری را می‌توان در هر یک از لایه‌های شبکه در مدل مرجع OSI قرار داد.
- رمزگذاری ارتباط (نقطه به نقطه) معمولاً در لایه‌های ۱ و ۲ انجام می‌پذیرد.
- رمزگذاری انتهای به انتهای 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

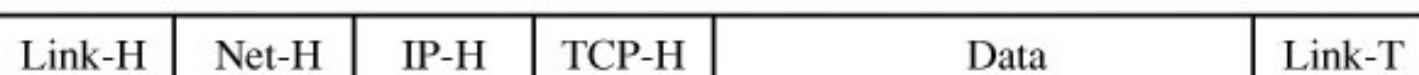


In gateways

(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

برنگذاری در الگوهای مختلف شبکه



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 پروتکل های رمزنگاری



مدیریت کلید چیست؟

- مدیریت کلید عبارتست از مجموعه‌ای از پروتکلهای 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 غیرمحفی...



مدیریت کلید شامل چه رویه هایی است؟

- تولید، توزیع و نصب داده های ارتباط کلیدی
- کنترل نحوه استفاده از این کلیدها
- به روز آوری، ابطال و نابود کردن (امحاء) داده های ارتباط کلیدی
- نگهداری، پشتیبان گیری و بازیابی داده های 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

■ سلسه مراتب کلید

■ تولید کلید و طول عمر کلید

■ اشتراک کلید مبتنی بر رمز متقارن

■ اشتراک کلید مبتنی بر رمز کلید عمومی

□ طراحی پروتکل های رمزنگاری

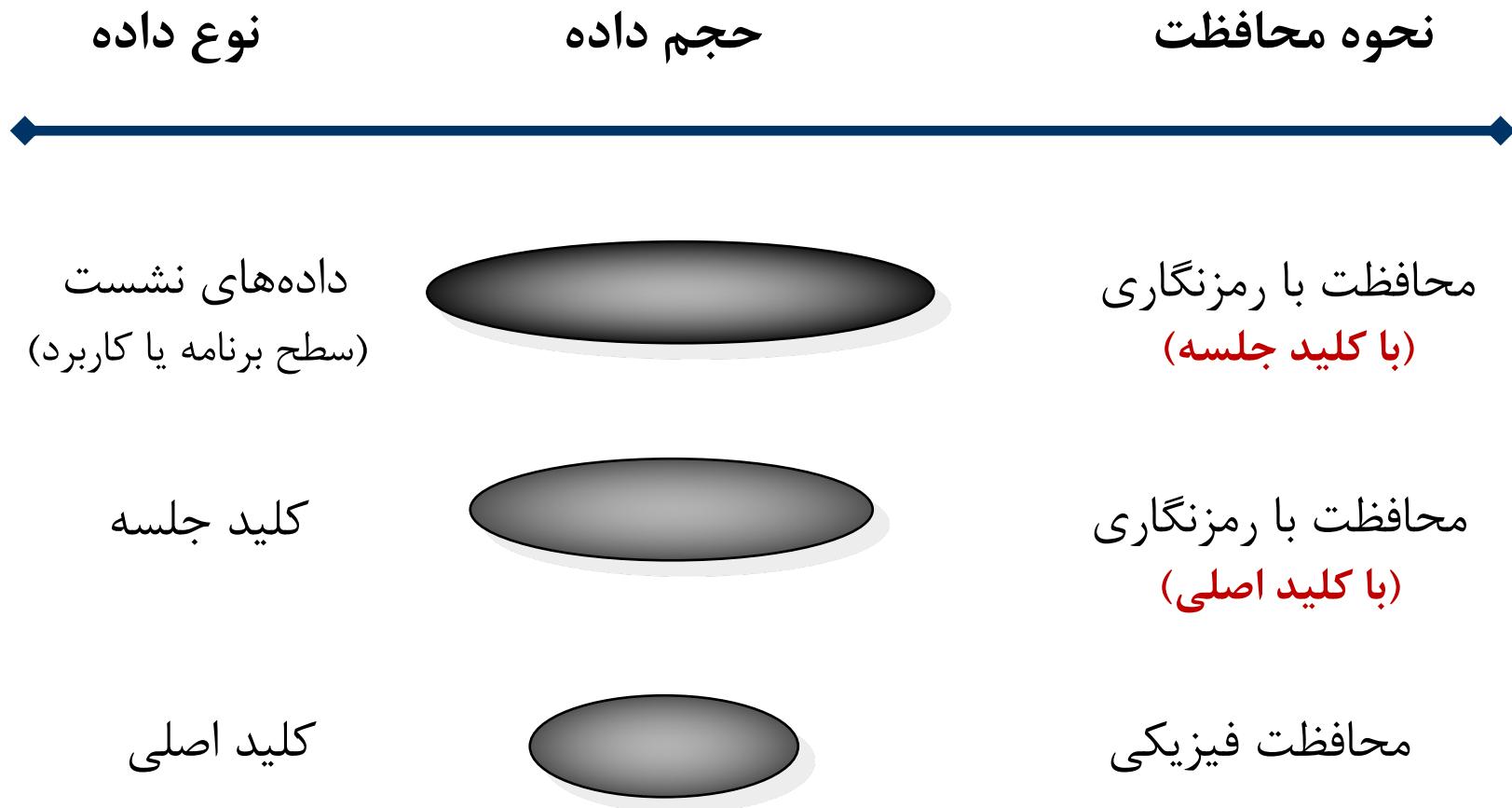


کلید جلسه و کلید اصلی: توصیف

- کلید اصلی عبارت است از یک کلید رمزگذاری سایر کلیدها. به این معنا که از این کلید برای توزیع کلید خصوصی موقتی به نام **کلید جلسه** استفاده می‌نماییم.
- از کلید جلسه برای رمزگاری و احراز صحت یا اصالت استفاده می‌کنیم.
 - رمزگاری متقارن



سلسله مراتب کلیدها





کلید جلسه و کلید اصلی: مقایسه

□ کلید اصلی:

- طول عمر نسبتاً زیاد،
- میزان استفاده محدود (فقط رمزنگاری کلیدهای جلسه)،
- خسارت گسترده در صورت 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 پروتکل های رمزنگاری



تولید کلید

- کلیدهای تولیدی باید کاملاً 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 می‌شوند.



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 افشاری 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

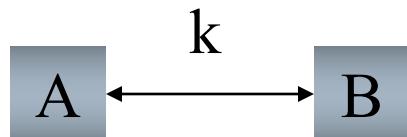
■ اشتراک کلید مبتنی بر رمز کلید عمومی

□ طراحی پروتکل های رمزنگاری



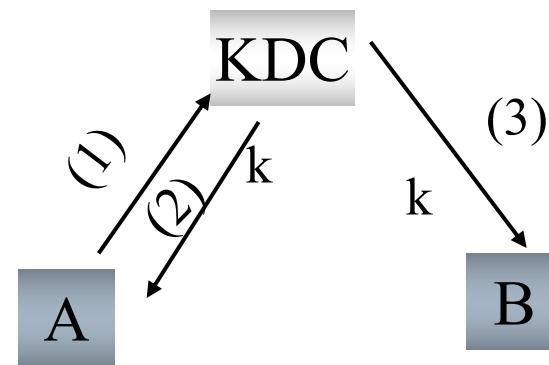
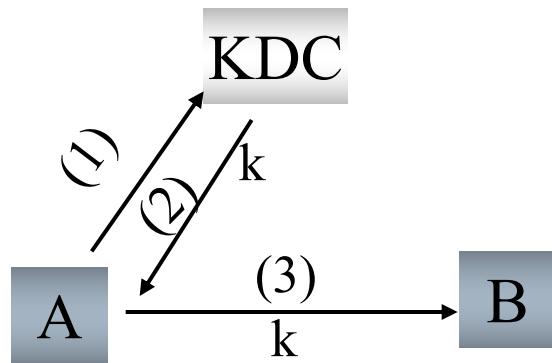
اشتراک کلید مبتنی بر رمز متقارن

□ دو رویکرد در اشتراک کلید جلسه



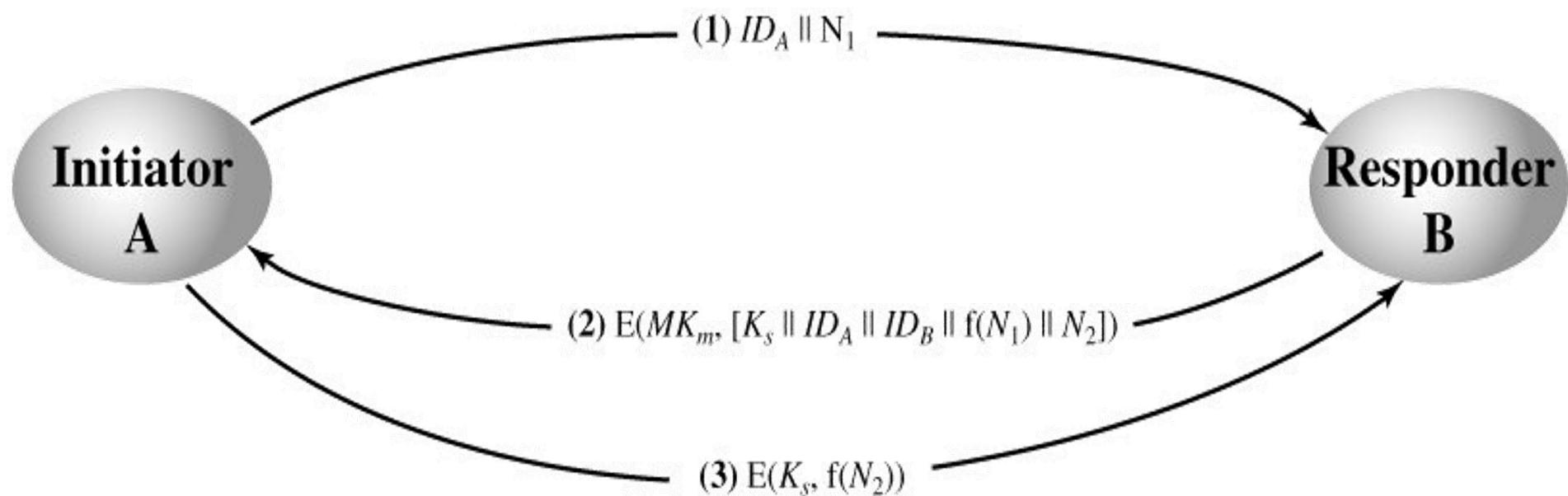
■ نقطه به نقطه

■ 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.

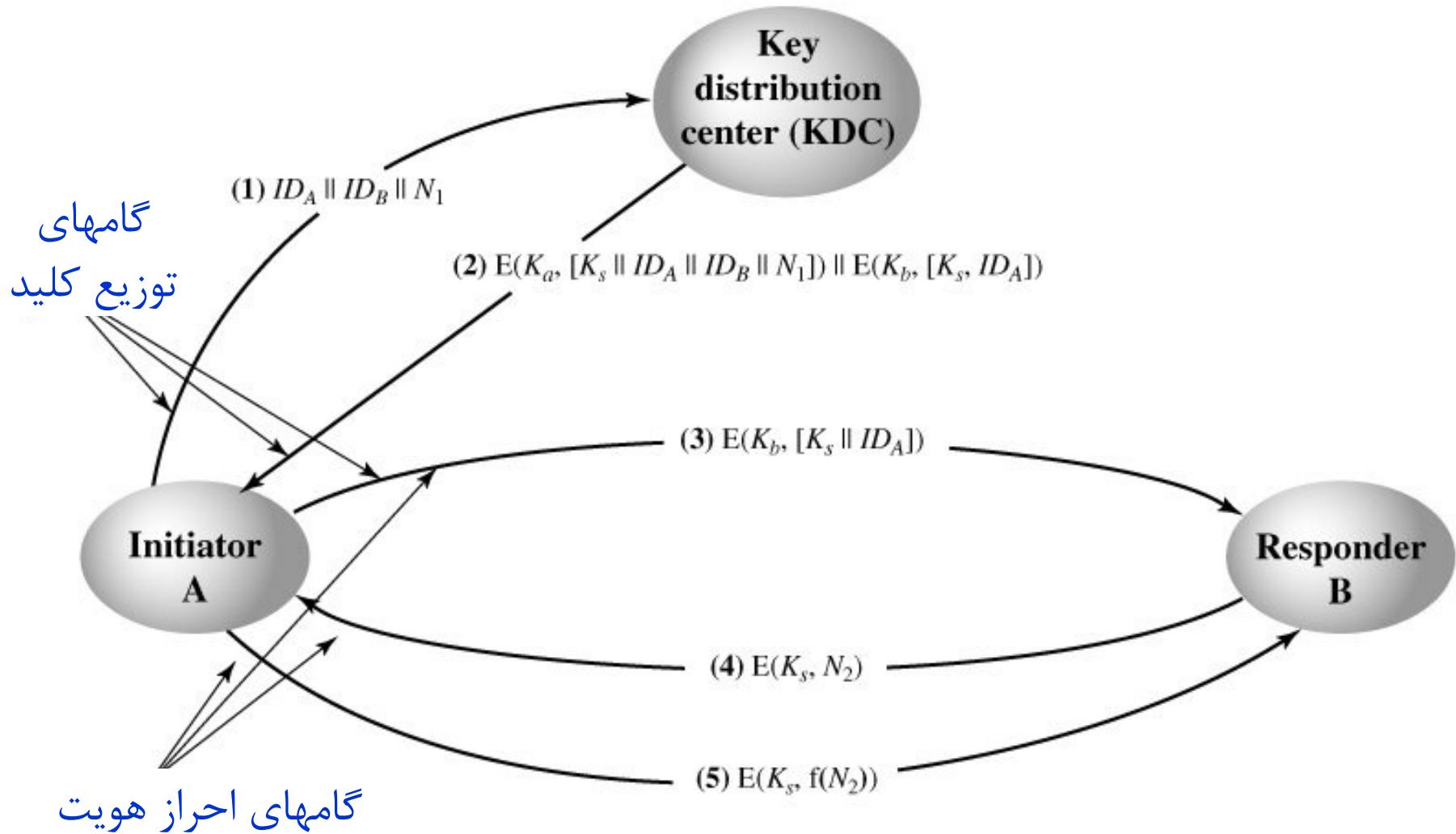




روش مت مرکز توزیع کلید

- 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 مراجعت فیزیکی) توزیع شده‌اند.
- آیده:
- هر بار که کاربری قصد ارتباط با دیگران را داشته باشد از KDC یک کلید جلسه درخواست می‌کند.

روش متمن کز توزیع کلید – مثال





روش متتمرکز توزیع کلید

□ نکات مثبت:

- تعداد کلید کمتر و قابلیت مقیاس پذیری

□ نکات منفی:

- کارگزار توزیع کلید گلوگاه امنیتی سیستم است.
- ترافیک بالا در کارگزار توزیع کلید باعث تبدیل آن به گلوگاه کارایی سیستم می شود.
- 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 پروتکل های 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 جانبی 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 جانبی 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 موردنی از رمزنگاری کلید عمومی برای به روز در آوردن 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 از 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 پروتکل های رمزنگاری



علائم و نمادها

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

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



علائم و نمادها

□ مُهر زمانی

- مُهر زمانی تولید شده توسط A_T
- در اغلب موارد فرض می کنیم که طرفهای ارتباط ساعت‌های خود را با استفاده از پروتکلی هماهنگ نگه می‌دارند.

□ پیام‌های مورد تبادل

- مثال: $([M \parallel ID_A \parallel ID_B], E(K_{as}, [M \parallel ID_A \parallel ID_B]))$

□ 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 می‌کند؟
 - پروتکل‌های مختلف، خصوصیات مختلفی دارند و اهداف مختلفی را دنبال می‌کنند.
 - بنابراین در انتخاب پروتکل باید دقیق کافی را به عمل آورد.



اهداف و خصوصیات پروتکل‌ها

□ تازگی (Freshness)

- کلید جلسه قبل استفاده نشده باشد (قدیمی نباشد) 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 صريح 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.



مهم‌ترین 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.



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 (یعنی کلید بین هر طرف با S)، برای انتقال امن کلید جلسه به کار می‌روند.
- احراز اصالت دو طرفه



پروتکل A

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

□ خصوصیات:

- لازم است که S یک کلید بلند مدت (همان کلید اصلی) را با A و B به اشتراک بگذارد.
- S کلیدی را تولید می‌کند و می‌گوید که برای استفاده بین A و B است.
- می‌داند که کلید را برای تعامل با A باید استفاده نماید.



پروتکل A

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

□ عیب اول: مهاجم می‌تواند خود را به جای طرفهای مختلف وانمود سازد. چرا که در این پروتکل عملاً احراز اصالت صورت نمی‌پذیرد.

- مهاجم D پیام سوم از A به B را دریافت می‌نماید.
- آن را با $E(K_{bs}, [K_{ab}]) \parallel ID_D$ جایگزین می‌کند.
- فکر می‌کند که کلید K_{ab} را برای تعامل با D باید استفاده کند.
- یا مهاجم E 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 A

□ عیب دوم: مشکل اصلی این پروتکل در امکان اجرای حمله مرد میانی است که در آن مهاجم (در میان دو طرف) به جای هر یک از طرفین خود را جا می‌زند و کلید تبادل می‌کند.

- $A \rightarrow E: ID_A \parallel ID_B$
- $E \rightarrow S: ID_A \parallel ID_E$
- $S \rightarrow E: E(K_{as}, [K_{ae}]) \parallel E(K_{es}, [K_{ae}])$
- $E \rightarrow A: E(K_{as}, [K_{ae}]) \parallel E(K_{es}, [K_{ae}])$
- $A \rightarrow E: E(K_{es}, [K_{ae}]) \parallel ID_A$
- $E \rightarrow S: ID_E \parallel ID_B$
- $S \rightarrow E: E(K_{es}, [K_{eb}]) \parallel E(K_{bs}, [K_{eb}])$
- $E \rightarrow B: E(K_{bs}, [K_{eb}]) \parallel ID_A$



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



پروتکل ۲

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

□ خصوصیات:

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



پروتکل ۲

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

- **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 می‌کند.
 - 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 **حدوده زمانی** قابل قبولی باشد. ضرورت همگامی ساعتها!

استفاده از Challenge/Response

- 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 خطأ، 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 ۳

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

□ 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 (عدم تایید کلید).

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

□ معايب:

- اين پروتکل نسبت به حمله تكرار آسيب پذير است.
- ممکن است کلید جلسه قبلی لو رفته باشد و بتوان جلسه جديدي را با تكرار از مرحله ۳ تشکيل داد و B در عمل نمي تواند از زنده بودن A مطمئن شود.
- همچنان A نمي تواند از زنده بودن B و دريافت کلید توسط آن مطمئن باشد.
- پيام چهارم عددی تصادفي است (رمز شده يك نانس تصادفي) و به A اطلاع خاصی نمي دهد.
- راه حل مقابله با حمله تكرار:
- تضمين تازگي پيام برای B (علاوه بر A)
- به طور مثال با اضافه کردن مهر زمانی به صورتی که در پروتکل بعد آمده است.



پروتکل Denning

- **A→S:** $ID_A \parallel ID_B$
- **S→A:** $E(K_{as}, [K_{ab} \parallel ID_B \parallel T_s \parallel E(K_{bs}, [K_{ab} \parallel ID_A \parallel T_s])])$
- **A→B:** $E(K_{bs}, [K_{ab} \parallel ID_A \parallel T_s])$
- **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 (و تایید کلید B) نمی‌تواند مطمئن شود.



پروتکل Denning

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

$$|\text{clock} - T_s| < \Delta t_1 + \Delta t_2$$

■ $\Delta t_2, \Delta t_1$ 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 Supress-replay و مقابله با آن

■ پروتکل فوق نسبت به حمله Supress-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 نанс



پروتکل Neuman

پروتکل بهبودیافته (جهت مقابله با حمله **(Suppress-Attack)**) □

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

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 (عدم تایید کلید). چرا که 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 مهر زمانی یا نانس استفاده کرد.

■ احراز اصالت دوطرفه



پروتکل A

□ کلید عمومی و مهر زمانی

□ **A→S:** $ID_A \parallel ID_B$

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



گواهی کلید عمومی A و B

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

□ **معایب:** نیاز به همگام بودن زمان سیستم‌های طرفین
ضمناً A از زنده بودن B و در اختیار داشتن کلید توسط B اطلاعی ندارد
(تایید کلید نداریم، هر چند احراز کلید را برای B داریم).



پروتکل ۲

کلید عمومی و نانس □

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



طراحی پروتکل

□ مبنای طراحی پروتکل‌های سری سوم

■ احراز اصالت یکطرفه

E-mail : □ نمونه‌ای از مورد کاربرد :

□ نیازمندی‌ها :

■ احراز اصالت فرستنده

■ محرمانگی

■ راه حل

□ رمزنگاری متقارن

□ رمزنگاری با کلید عمومی



پروتکل A

- استفاده از رمز متقارن (برای احراز اصالت یکطرفه در ارسال ایمیل)
- **A→S:** $ID_A \parallel ID_B \parallel N_a$
- **S→A:** $E(K_{as}, [K_{ab} \parallel ID_B \parallel N_a \parallel E(K_{bs}, [K_{ab} \parallel ID_A])])$
- **A→B:** $E(K_{bs}, [K_{ab} \parallel ID_A]) \parallel E(K_{ab}, M)$
- **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 ۲

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



پایان

پست الکترونیکی

amini@sharif.edu

kharrazi@sharif.edu