





# 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:

## ■ رمزنگاری خط ارتباطی (نقطه-به-نقطه)

□ رمزگذاری روی هر خط ارتباطی به صورت مستقل صورت می‌پذیرد.

□ باید در هر یک از تجهیزات ارتباطی رمزگشایی شود.

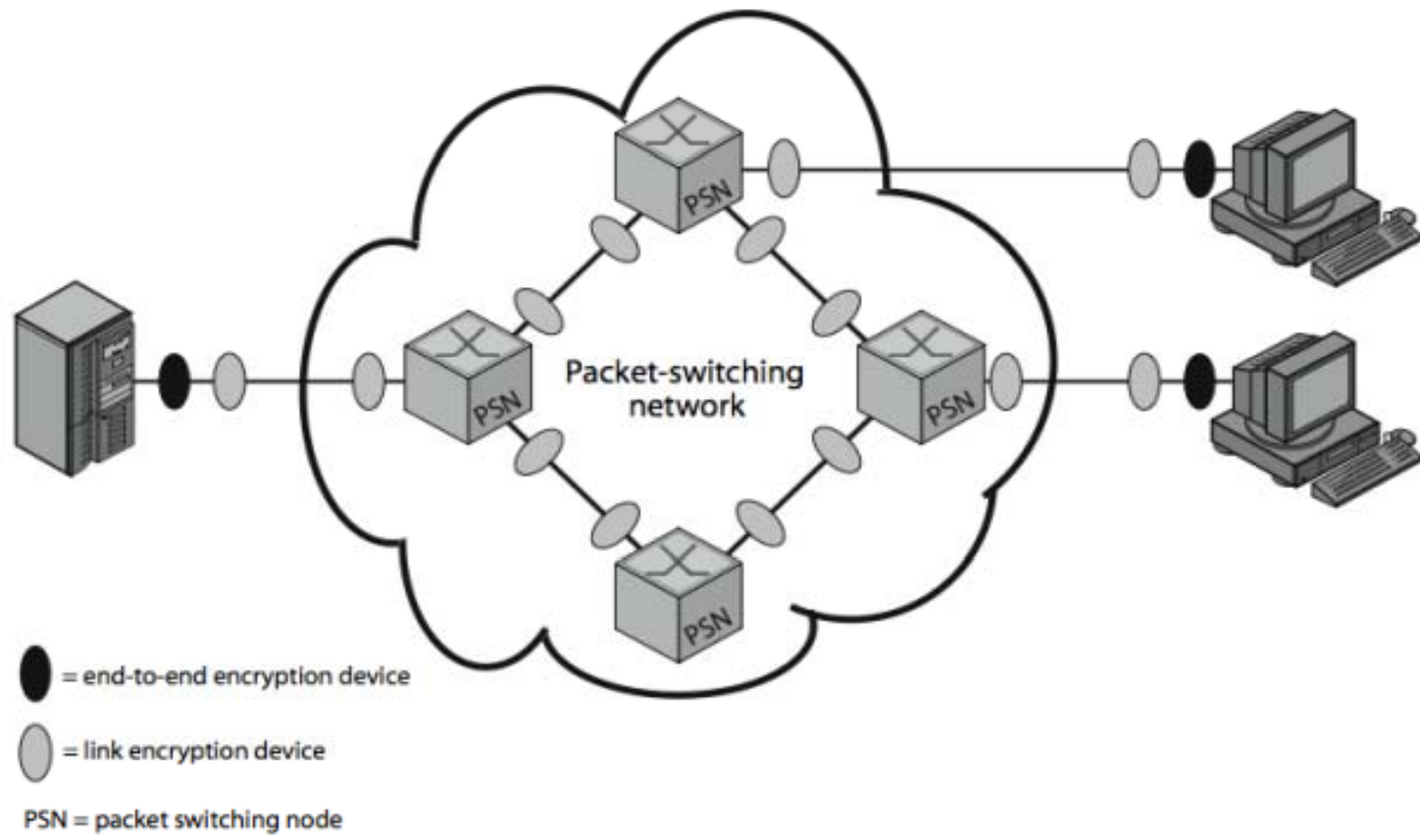
□ نیازمند تجهیزات متعددی است، هر کدام با کلیدهای مجزا.

## ■ رمزنگاری انتها-به-انتها

□ رمزگذاری صرفاً بین مبدا و مقصد پیام انجام می‌شود.

□ نیازمند یک کلید مشترک بین دو انتها است.

# جایگاه رمز متقارن





# جایگاه رمز متقارن

□ در رمز گذاری 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:

■ رمز گذاری انتها-به-انتها محتوای داده‌ها را بر روی کل مسیر حفاظت نماید و امکان احراز اصالت داده را نیز فراهم آورد.

■ رمز گذاری خط ارتباطی (نقطه-به-نقطه) جریان داده را از مانیتورینگ حفاظت نماید.



# جایگاه رمز متقارن

□ تابع رمزگذاری را می توان در هر یک از لایه های شبکه در مدل مرجع OSI قرار داد.

■ رمزگذاری ارتباط در لایه های ۱ و ۲ انجام می پذیرد.

■ رمزگذاری انتها به انتها در لایه های بالاتر.

□ هر چه قدر به لایه های بالاتر برویم،

■ اطلاعات کمتری رمز می شود، ولی امنیت بیشتری فراهم می گردد.

■ پیچیدگی، بیشتر و همچنین موجودیتها و کلیدهای درگیر، بیشتر می شود.





(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



# 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.
- می‌تواند برای ایجاد کانالهای پنهان مورد استفاده قرار گیرد.
- رمزگذاری خط ارتباطی می‌تواند جزئیات سرآیند را مخفی کند.
- اما حجم ترافیک شبکه و داده‌ها در دو انتهای ارتباط همچنان آشکار است.
- لایه‌گذاری (Padding) در ترافیک نیز می‌تواند جریان داده‌ها را ناشفاف نماید، ولی هزینه سر بار 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 شگردها و رویه‌ها برای دایر نمودن 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؟

- مقداردهی اولیه سیستم‌های کاربران
- 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 امکان ارتباط فیزیکی برای تبادل کلید امن را با یکدیگر ندارند.
- 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

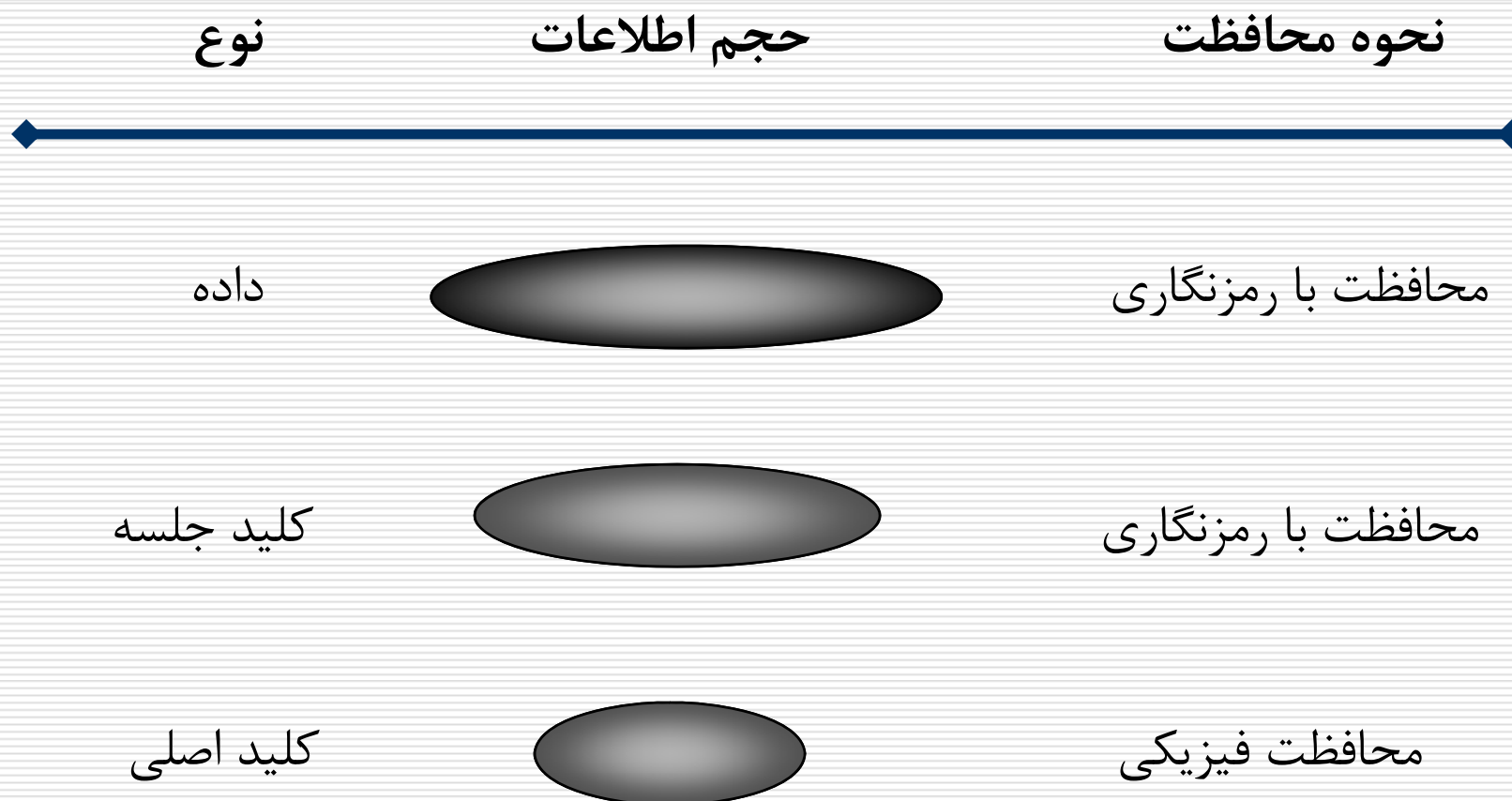
□ کلید اصلی عبارتست از یک کلید رمزکننده سایر کلیدها. به این معنا که از این کلید برای توزیع کلید خصوصی موقتی به نام کلید جلسه استفاده می‌نماییم.

□ از کلید جلسه برای رمزنگاری و احراز هویت استفاده می‌کنیم.  
■ رمزنگاری 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 پروتکل‌های رمزنگاری



# تولید کلید

- کلیدهای تولیدی باید کاملاً تصادفی باشند و از آنترופی (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 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

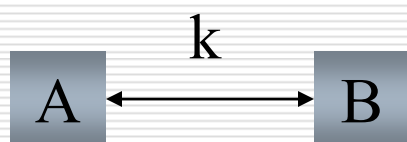
■ اشتراک کلید مبتنی بر رمز متقارن

■ اشتراک کلید مبتنی بر رمز کلید عمومی

□ طراحی پروتکل‌های رمزنگاری



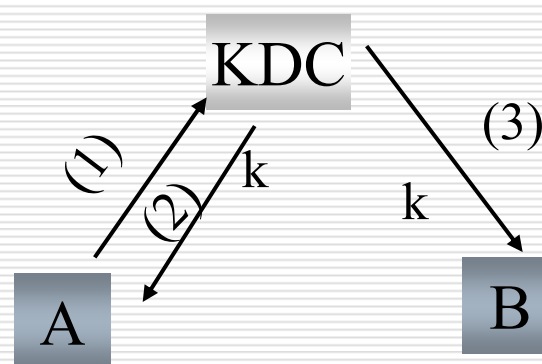
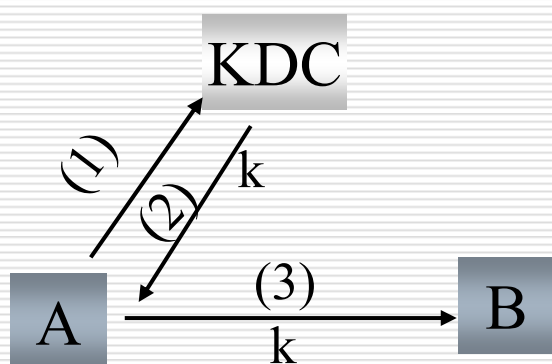
# اشتراک کلید مبتنی بر رمز متقارن



□ دو رویکرد در اشتراک کلید جلسه

■ نقطه به نقطه

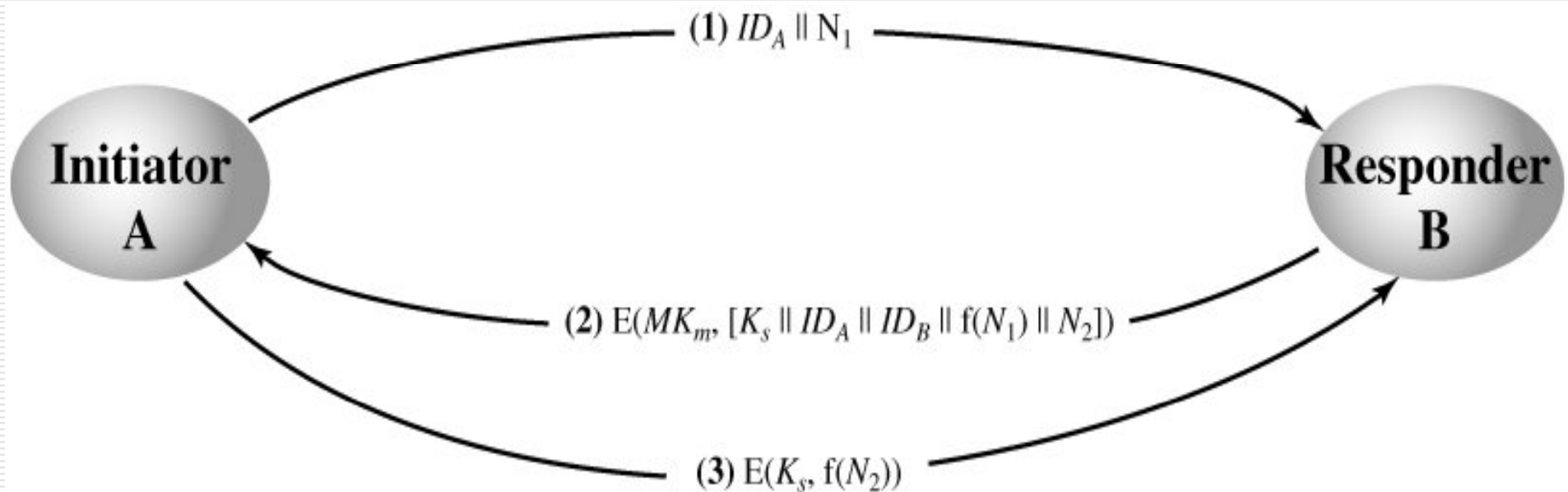
■ مرکز توزیع کلید





# روش نقطه به نقطه توزیع کلید

- نیاز به توافق بر روی کلید پیش از برقراری ارتباط بین هر دو نفر
- مقیاس پذیری: مشکل اصلی
- 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.

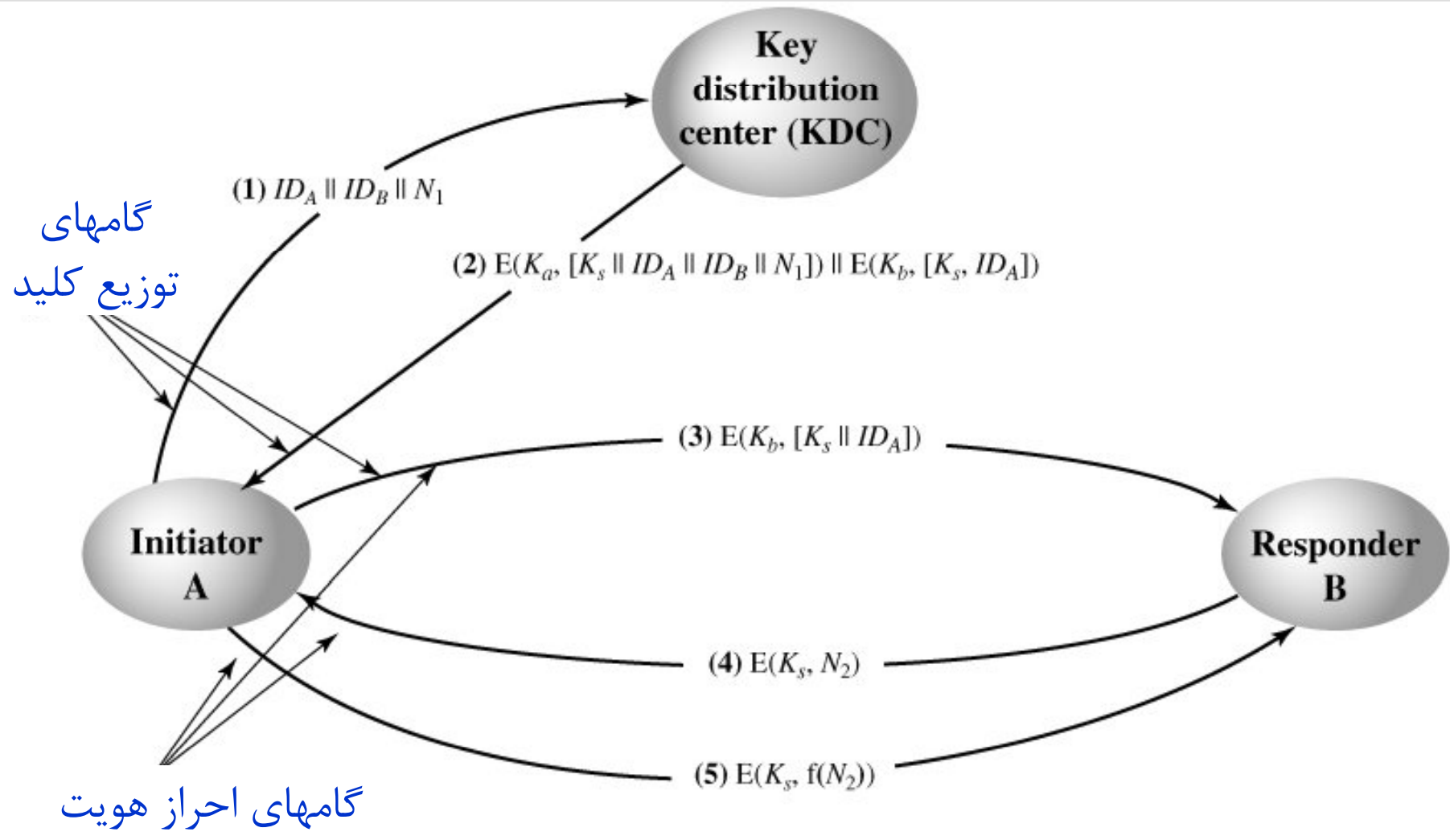
■ کلیدهای اصلی با یک روش امن (مثلاً مراجعه فیزیکی) توزیع شده‌اند.

□ ایده:

■ هر بار 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

□ نکات منفی:

■ کارگزار توزیع کلید گلوگاه امنیتی سیستم است.

■ ترافیک بالا در کارگزار توزیع کلید گلوگاه کارایی سیستم است.

■ 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 رمزنگاری کلید عمومی
  - نیازی به تبادل 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 می نماید.

□ مثال : الگوریتم توزیع کلید در SSL (در درس های بعدی معرفی می شود).



# روش ترکیبی

- کلید عمومی+رمزنگاری متقارن
- توزیع مداوم کلید با رمزنگاری کلید عمومی کارآیی سیستم را کاهش می دهد.
- با کمک روش ترکیبی به طور موردی از رمزنگاری کلید عمومی برای به روز درآوردن کلید اصلی بهره می جوئیم.
- شامل سه سطح
- توافق KDC با هر یک از کاربران روی یک کلید اصلی
- استفاده از کلید عمومی برای توزیع کلیدهای اصلی
- 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 شناسه‌های  $ID_A$  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

## □ مَهر زمانی

■  $T_a$  مَهر زمانی تولید شده توسط A

■ فرض می کنیم که طرفهای ارتباط ساعت‌های خود را با استفاده از پروتکلی هماهنگ نگه می‌دارند.

## □ پیام‌های مورد تبادل

■  $A \rightarrow B: M, E(K_{as}, [M \parallel ID_A \parallel ID_B])$

■ 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 دارند.
- طرف‌های ارتباط می‌خواهند برای ارتباط با یکدیگر کلید جلسه به اشتراک بگذارند.
- اما چه خصوصیتی 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 کلید فاش شده (Known Key Resilience)

- مهاجمی که به کلید یک جلسه دست یافته، در مورد کلید اصلی و کلید جلسات دیگری 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.

## □ بازتاب (Reflection)

■ مهاجم از پیامهای دریافتی از گیرنده (در طی یک اجرای دیگر از پروتکل) برای احراز خود به گیرنده استفاده می‌کند.



# 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.





# روشهای مقابله با تکرار

## □ 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.

## □ استفاده از مهر زمانی (Timestamp)

- گیرنده به پیام اعتماد می کند اگر در محدوده زمانی قابل قبولی باشد. ضرورت همگامی ساعتها!

## □ استفاده از Challenge/Response

- $Y$  که انتظار یک پیام نو از  $X$  دارد، یک نانس به  $X$  ارسال می کند و انتظار دارد که پیامی که دریافت می کند حاوی نانس موردنظر باشد.



# 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 کلید جلسه را تولید می‌کند.

□ کلیدهای اصلی (بین هر طرف با S) برای انتقال کلید جلسه بکار می‌رود.

■ **احراز اصالت دوطرفه**



# پروتکل ۱

□  $A \rightarrow S: ID_A \parallel ID_B$

□  $S \rightarrow A: K_{ab}$

□  $A \rightarrow B: K_{ab} \parallel ID_A$

□ خصوصیات:

■ S کلیدی را تولید می کند و می گوید که برای استفاده بین A و B است.

■ B می داند که کلید را برای تعامل با A باید استفاده نماید.

□ معایب:

■ مهاجم می تواند با شنود کلید مخفی  $K_{ab}$  را به دست آورد.

□ راه حل: نیاز به رمزگذاری کلید داریم.



## پروتکل ۲

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

□ خصوصیات:

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



## پروتکل ۲

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

□ عیب اول:

- مهاجم D پیام سوم از A به B را دریافت می نماید.
- آن را با  $ID_D \parallel E(K_{bs}, [K_{ab}])$  جایگزین می کند.
- B فکر می کند که کلید  $K_{ab}$  را برای تعامل با D باید استفاده کند.
- بنابراین مهاجم می تواند خود را به جای طرفهای مختلف وانمود سازد.



## پروتکل ۲

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

□ **عیب دوم:** مشکل اصلی این پروتکل در امکان اجرای **حمله مرد میانی** است.

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

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



## پروتکل ۳

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

□ خصوصیات:

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





## پروتکل ۳

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

□ **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  $K_{bs}$  و  $K_{as}$

□ **راه حل:** لازم است به گونه‌ای از تازگی کلید اطمینان حاصل نماییم.



## پروتکل ۴

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

□ خصوصیات:

■ تازگی برای A با استفاده از نانس فراهم 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.

□ 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 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.

## □ راه حل:

- تضمین تازگی پیام برای B (علاوه بر A)
- به طور مثال با اضافه کردن مُهر زمانی به صورتی که در پروتکل بعد آمده



# پروتکل Denning

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

□ خصوصیات:

■ استفاده از مُهر زمانی برای جلوگیری از حمله تکرار



# پروتکل Denning

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

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

■  $\Delta t_1, \Delta t_2$  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 آسیب پذیر است.
- این حمله از سنکرون نبودن ساعت‌های فرستنده و گیرنده ناشی می‌شود.
- وقتی ساعت فرستنده جلوتر از ساعت گیرنده باشد، مهاجم می‌تواند پیام‌ها را در زمان مقرر بازرسان نماید.
- روشهای مقابله :
  - چک کردن متناوب با زمان S
  - توافق از طریق نانس



# پروتکل Neuman

□ پروتکل بهبود یافته (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





# 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 ۱

□ کلید عمومی و مهر زمانی

□  $A \rightarrow S: ID_A \parallel ID_B$

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

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

□ **معایب:** نیاز به همگام بودن زمان سیستم‌های طرفین



## پروتکل ۲

□ کلید عمومی و نانس

- $A \rightarrow S: ID_A \parallel ID_B$
- $S \rightarrow A: E(PR_s, [ID_B \parallel PU_b])$
- $A \rightarrow B: E(PU_b, [N_a \parallel ID_A])$
- $B \rightarrow S: ID_A \parallel ID_B \parallel E(PU_s, N_a)$
- $S \rightarrow B: E(PR_s, [ID_A \parallel PU_a]) \parallel E(PU_b, E(PR_s, [N_a \parallel K_{ab} \parallel ID_A \parallel ID_B]))$
- $B \rightarrow A: E(PU_a, E(PR_s, [(N_a \parallel K_{ab} \parallel ID_A \parallel ID_B) \parallel N_b]))$
- $A \rightarrow B: E(K_{ab}, N_b)$



# 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 ۱

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

□  $A \rightarrow S: ID_A \parallel ID_B \parallel N_a$

□  $S \rightarrow A: E(K_{as}, [K_{ab} \parallel ID_B \parallel N_a \parallel E(K_{bs}, [K_{ab} \parallel ID_A])])$

□  $A \rightarrow B: E(K_{bs}, [K_{ab} \parallel ID_A]) \parallel E(K_{ab}, M)$

□ **خصوصیات:** گیرنده یکبار پیام را دریافت می کند و می تواند فرستنده را احراز نماید.

□ **معایب:** امکان حمله تکرار در آن وجود دارد.



## پروتکل ۲

□ استفاده از رمز کلید عمومی (برای احراز اصالت یکطرفه در ارسال ایمیل)

■ هدف: محرمانگی

■  $A \rightarrow B: E(PU_b, K_s) \parallel E(K_s, M)$

■ هدف: احراز اصالت فرستنده

■  $A \rightarrow B: M \parallel E(PR_a, H(M))$

■ محرمانگی و احراز اصالت فرستنده

■  $A \rightarrow B: E(PU_b, [M \parallel E(PR_a, H(M))])$

■ احراز اصالت، بدون اطلاع طرفین از کلید عمومی یکدیگر

■  $A \rightarrow B: M \parallel E(PR_a, H(M)) \parallel E(PR_s, [T \parallel ID_A \parallel PU_a])$



# پایان

مرکز امنیت شبکه شریف  
<http://dnsl.ce.sharif.edu>

پست الکترونیکی  
[m\\_amani@ce.sharif.edu](mailto:m_amani@ce.sharif.edu)