



آزمایشگاه امنیت داده و شبکه
<http://dnsl.ce.sharif.edu>



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

درس ۸: طراحی پروتکل‌های رمزنگاری

محمد صادق دوستی

□ پروتکل‌های رمزنگاری: پروتکل‌هایی که در آنها از الگوریتم‌های رمز استفاده می‌شود.

👉 مثال: تصدیق هویت، توزیع کلید، رأی‌گیری الکترونیکی، پرداخت الکترونیکی، امضای قرارداد، ...

□ در این درس به پروتکل‌های توزیع کلید می‌پردازیم.

□ 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 تبادل کلید

- ❑ مدیریت کلید عبارت است از مجموعه فرآیندهای تولید، تبادل، نگهداری، استفاده، امحا و جایگزینی کلیدهای موجود در سیستم.
- ❑ کلیدها می توانند عمومی یا خصوصی باشند.

□ اکثر حملات به رمزنگاری یک سیستم امنیتی، در بخش مدیریت کلید است.

👉 چرا که طرفهای ارتباط، امکان ارتباط فیزیکی برای تبادل کلید امن را با یکدیگر ندارند.

□ در حقیقت برخی این مسأله را دشوارترین جزء یک سیستم امن می دانند.

□ 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 تبادل کلید

□ کلید اصلی (Master Key)

☞ یا کلید طولانی مدت (Long-Term Key یا LTK)

☞ کلیدی که برای رمزگذاری و/یا تصدیق هویت کلیدهای دیگر مورد استفاده قرار می‌گیرد.

□ از کلید جلسه (نشست) برای رمزنگاری و تصدیق هویت پیامها استفاده می‌کنیم.

☞ رمزنگاری متقارن

نوع کلید	حجم اطلاعات	عمر	خسارت در صورت لو رفتن	نحوه محافظت
کلید اصلی	خیلی کم	طولانی	خیلی زیاد	محافظت فیزیکی
کلید جلسه (نشست)	کم	کوتاه	زیاد	با رمزنگاری
داده	زیاد	–	بسته به کاربرد	با رمزنگاری

□ 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 تبادل کلید

❑ کلیدهای تولیدی باید کاملاً تصادفی باشند.

☞ کامپیوتر نمی‌تواند اعداد تصادفی واقعی تولید نماید.

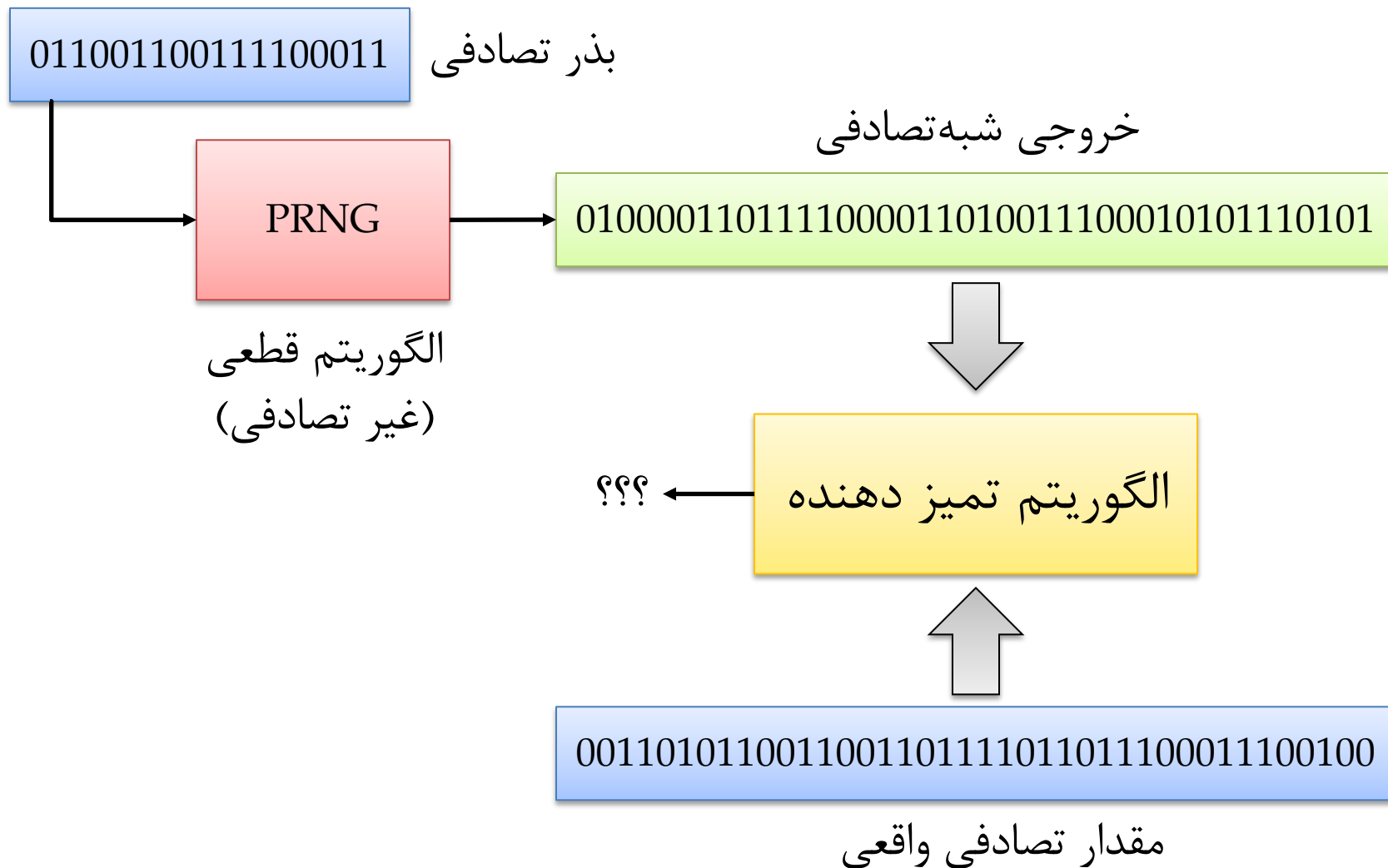
❑ استفاده از منابع تصادفی بیرونی (تعامل با کاربر و ...)

☞ تولید یک بذر (seed) تصادفی

❑ گسترش بذر تصادفی به یک مقدار شبه تصادفی
(pseudorandom)

☞ غیر قابل تمیز از مقدار تصادفی واقعی

❑ نیاز به الگوریتم‌های مولد اعداد شبه تصادفی (PRNG)



□ 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.

روشهای تبادل کلید جلسه (Key Distribution)

□ توافق کلید (Key Agreement)

➡ هر دو طرف در انتخاب کلید تاثیرگذار هستند.

➡ مثال: روش Diffie-Hellman

□ انتقال کلید (Key Transport)

➡ یکی از دو طرف کلید را معین کرده و به دیگری ارسال می نماید.

□ مدیریت کلید

➡ مفاهیم اساسی مدیریت کلید

➡ سلسله مراتب کلید

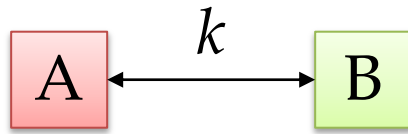
➡ تولید کلید و طول عمر کلید

➡ **اشتراک کلید مبتنی بر رمز متقارن**

➡ اشتراک کلید مبتنی بر رمز کلید عمومی

□ طراحی پروتکل‌های تبادل کلید

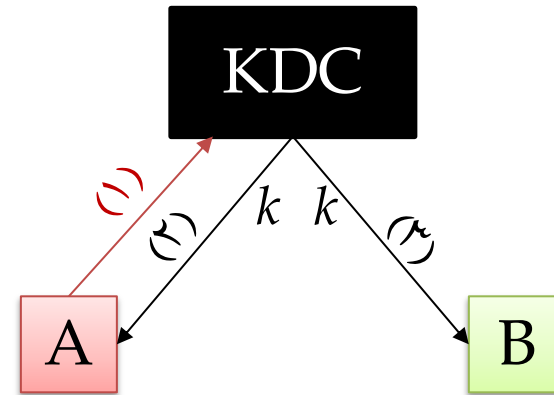
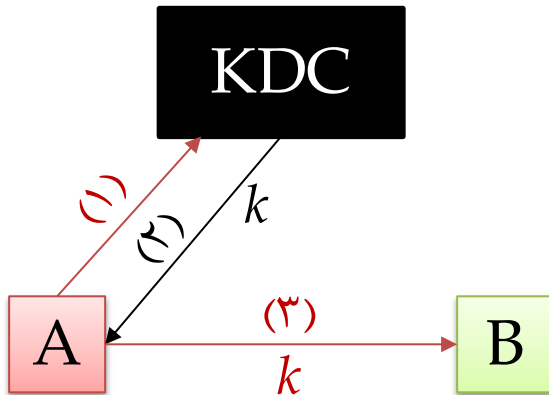
دو رویکرد در اشتراک کلید جلسه



□ بدون واسطه (همتا به همتا)

□ با واسطه (مرکز توزیع کلید یا KDC)

□ KDC: Key Distribution Center



روش همتا به همتا در توزیع کلید

□ مزیت: بدون نیاز به اعتماد به یک شخص ثالث

□ عیب: مشکل مقیاس پذیری؛ نیاز به کلید اصلی بین هر دو

موجودیت

➞ برای ارتباط n نفر باهم به $n(n-1)/2$ کلید اصلی احتیاج داریم.

روش با واسطه توزیع کلید

□ هر کاربر یک کلید اصلی با KDC به اشتراک گذاشته است.

☞ KDC یک شخص ثالث مورد اعتماد است.

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

□ هر بار که کاربری قصد ارتباط با دیگران را داشته باشد از KDC یک کلید جلسه درخواست می‌کند.

□ کلید جلسه به صورت تصدیق هویت شده در اختیار متقاضی (و بعضاً مخاطب) قرار می‌گیرد.

□ نکات مثبت:

➡ تعداد کلید کمتر و قابلیت مقیاس پذیری

□ نکات منفی:

➡ نیاز به اعتماد به شخص ثالث

➡ KDC تک نقطه خرابی (SPOF) است.

➡ ترافیک بالا در KDC **گلوگاه کارایی** سیستم است.

➡ نیاز به یک کارگزار برخاسته داریم. دخالت کارگزار در برقراری هر ارتباط ضروری است.

□ 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 تبادل کلید

جایگاه رمزنگاری کلید عمومی

□ الگوریتمهای نامتقارن بسیار کندتر از الگوریتمهای متقارن هستند
👉 از الگوریتمهای نامتقارن جهت توزیع کلید جلسه (و نه رمزگذاری) استفاده می‌شود.

□ با استفاده از رمزنگاری کلید عمومی:

👉 نیازی به تبادل کلیدهای اصلی و حفظ محرمانگی آنها نیست.

👉 اما کلید عمومی باید به روشی امن منتقل شود (مثلاً PKI).

👉 نیازی به کارگزار بر خط نیست.

□ 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 تبادل کلید

□ عامل‌ها/طرفهای ارتباط

☞ A و B با شناسه‌های ID_A و ID_B

☞ T شخص ثالث مورد اعتماد

□ K_{AT} و K_{BT} کلید طولانی مدت بین A و T و بین B و T

□ k_s کلید جلسه (Session)

□ مُهر زمانی (Timestamp)

☞ t_A مُهر زمانی تولید شده توسط A

☞ طرفین ساعتهای خود را به کمک پروتکلی هماهنگ نگه می‌دارند.

□ نانس (Nonce)

☞ مقداری تصادفی که تنها یک بار مورد استفاده قرار می‌گیرند.

☞ n_A نانس تولید شده توسط A

$$A \rightarrow B: \quad \{M \parallel ID_A \parallel ID_B\}K_{AT}$$

□ A فرستنده و B گیرنده

□ ترکیب M (پیام)، شناسه A و شناسه B با کلید K_{AT} رمز و تصدیق هویت شده است (مثال: روش ترکیب EtA).

□ اگر فقط قصد تصدیق هویت پیامی را داشته باشیم:

$$A \rightarrow B: \quad \langle\langle M \parallel ID_A \parallel ID_B \rangle\rangle K_{AT}$$

$$\langle\langle M \rangle\rangle K \stackrel{\text{def}}{=} M \parallel \text{MAC}(K, M)$$

اهداف و خصوصیات پروتکل‌های تبادل کلید – ۱

□ تازگی کلید (Freshness)

☞ کلید جلسه توسط اجرای جاری پروتکل تولید شده باشد (و نه اجرای قبلی).

□ محرمانگی پیشرو (Forward Secrecy)

☞ با لو رفتن کلید بلند مدت (اصلی)، کلیدهای جلسه قبلی امن بمانند.

□ استحکام در برابر کلید فاش شده (Known-Key Resilience)

☞ مهاجمی که به کلید یک جلسه دست یافته، نتواند در مورد کلید اصلی یا کلید جلسات دیگر اطلاعی به دست آورد.

اهداف و خصوصیات پروتکل‌های تبادل کلید – ۲

□ تصدیق هویت کلید (Key Authentication)

☞ یک طرف مطمئن است که هیچ کس جز طرف دوم (و احتمالاً سایر معتمدین) نمی‌تواند به کلید جلسه دسترسی داشته باشد.

□ تأیید کلید (Key Confirmation)

☞ یک طرف مطمئن است که طرف دوم واقعاً کلید جلسه را در اختیار دارد.

□ تصدیق هویت صریح کلید (Explicit Key Authentication)

☞ تصدیق هویت کلید + تأیید کلید

اهداف و خصوصیات پروتکل‌های تبادل کلید – ۳

□ تصدیق هویت یک طرفه (Unilateral)

☞ تنها یک طرف ارتباط، هویت خود را اثبات می‌کند.

☞ مثال: یک شخص یک پیام را در یک گروه عمومی منتشر می‌کند.

□ تصدیق هویت دو طرفه (Mutual)

☞ هر دو طرف ارتباط هویت خود را اثبات می‌کنند.

انواع حملات به پروتکل‌های تبادل کلید

❑ شنود (Eavesdropping)

➡ مهاجم اطلاعات و پیامهای تبادل شده در پروتکل را دریافت می‌نماید.

❑ تغییر (Modification)

➡ مهاجم اطلاعات ارسالی را تغییر می‌دهد.

❑ منع سرویس (Denial of Service)

➡ مهاجم مانع از کامل شدن پروتکل توسط طرف‌های مجاز می‌شود.

برخی گونه‌های مهم از «حمله تغییر»

□ تکرار (Replay)

➡ مهاجم پیامهای ارسالی در طی پروتکل را ثبت نموده، سپس به اجرای پروتکل با ارسال مجدد آنها می‌پردازد.

□ مرد میانی (Man in the Middle)

➡ مهاجم نقش A را برای B و نقش B را برای A بازی می‌کند.

□ دستکاری گواهی (Certificate Manipulation)

➡ مهاجم اطلاعات گواهی را تغییر می‌دهد.

□ استفاده از مُهر زمانی (Timestamp)

👉 گیرنده به پیام اعتماد می کند اگر در محدوده زمانی قابل قبولی باشد.

👉 ضرورت همگامی ساعتها.

روشهای مقابله با تکرار - ۲

□ استفاده از چالش-پاسخ (Challenge-Response)

👉 A انتظار یک پیام نو از B دارد.

👉 A یک چالش یا نانس به B ارسال می کند.

👉 A انتظار دارد که پیامی که دریافت می کند حاوی تغییر یافته (رمز شده) چالش یا نانس موردنظر باشد.

□ استفاده از توالی شمار (Sequence Number)

➡ پیامهای اجرای N ام پروتکل باید حاوی عدد N باشند.

➡ طرفین باید همواره مقدار N را نگه دارند.

➡ مشکلات در همگام نگه داشتن طرفین

□ 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.

مبنای طراحی پروتکل‌های سری اول

□ مبتنی بر رمز متقارن: استفاده از KDC (با نام T)

👉 T کلید جلسه را تولید می‌کند.

👉 کلیدهای اصلی (بین هر طرف با T) برای انتقال کلید جلسه بکار می‌رود.

□ تصدیق هویت دو طرفه

1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: k_s$
3. $A \rightarrow B: k_s \parallel ID_A$

□ عیب:

☞ مهاجم می‌تواند با شنود کلید جلسه k_s را به دست آورد.

□ راه حل: نیاز به رمزگذاری و تصدیق هویت کلید داریم.

1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: \{k_s\}K_{AT} \parallel \{k_s\}K_{BT}$
3. $A \rightarrow B: \{k_s\}K_{BT} \parallel ID_A$

□ عیب ۱: پروتکل تصدیق هویت طرفین ندارد.

➡ مهاجم X می تواند مانع رسیدن پیام سوم به B شود.

➡ به جای ID_A در آن، ID_X خودش را قرار دهد و پیام را بفرستد.

➡ B فکر می کند که باید با X صحبت کند.

1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: \{k_s\}K_{AT} \parallel \{k_s\}K_{BT}$
3. $A \rightarrow B: \{k_s\}K_{BT} \parallel ID_A$

□ عیب ۲: پروتکل تازگی پیام را بررسی نمی کند.

➡ مهاجم X مانع رسیدن پیام ۲ و ۳ پروتکل می شود.

➡ مهاجم X پیام ۲ و ۳ را از اجراهای قبلی پروتکل ارسال می کند.

1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: \{k_s\}K_{AT} \parallel \{k_s\}K_{BT}$
3. $A \rightarrow B: \{k_s\}K_{BT} \parallel ID_A$

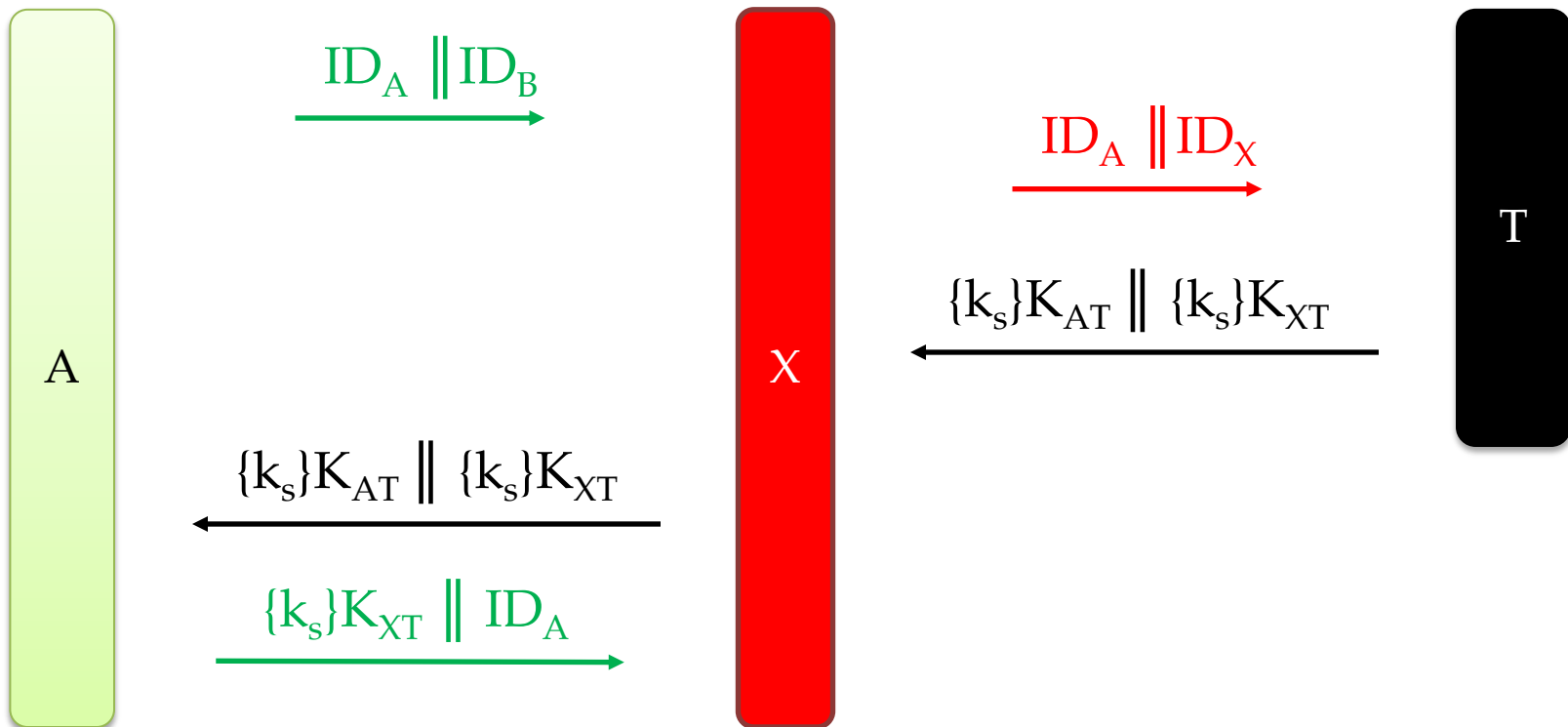
□ عیب ۳: پروتکل در برابر حمله MITM آسیب پذیر است.

➡ مهاجم X پیام اول را عوض کرده و به جای ID_B ، ID_X می گذارد.

➡ توصیف حمله در اسلاید بعد.

حمله MITM به پروتکل ۲

$A \rightarrow T: \quad ID_A \parallel ID_B$
 $T \rightarrow A: \quad \{k_s\}K_{AT} \parallel \{k_s\}K_{BT}$
 $A \rightarrow B: \quad \{k_s\}K_{BT} \parallel ID_A$



1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: \{k_s \parallel ID_B\}K_{AT} \parallel \{k_s \parallel ID_A\}K_{BT}$
3. $A \rightarrow B: \{k_s \parallel ID_A\}K_{BT}$

□ **خصوصیات:** شناسه مخاطب ارتباط و کلید جلسه با کلید اصلی رمز و تصدیق هویت می‌شوند.

□ **عیب:** کماکان تازگی پیام واریسی نمی‌شود.

1. $A \rightarrow T: ID_A \parallel ID_B \parallel n_A$
2. $T \rightarrow A: \{n_A \parallel k_s \parallel ID_B \parallel \{k_s \parallel ID_A\}K_{BT}\}K_{AT}$
3. $A \rightarrow B: \{k_s \parallel ID_A\}K_{BT}$

□ خصوصیات:

👉 تازگی کلید برای A (و نه B) با استفاده از نانس تصدیق می گردد.

1. $A \rightarrow T: ID_A \parallel ID_B \parallel n_A$
2. $T \rightarrow A: \{n_A \parallel k_s \parallel ID_B \parallel \{k_s \parallel ID_A\}K_{BT}\}K_{AT}$
3. $A \rightarrow B: \{k_s \parallel ID_A\}K_{BT}$

□ 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).

پروتکل Needham-Schroeder

□ یکی از نخستین پروتکل‌های تبادل کلید (۱۹۷۸)

☞ شناسایی حمله تکرار توسط Denning و Sacco (۱۹۸۱)

☞ ابداع منطقی توسط Burrows, Abadi و Needham برای

وارسی خودکار پروتکل‌ها و جلوگیری از حملات مشابه (منطق BAN) در سال ۱۹۹۰.



Roger Needham
(1935 – 2003)



Michael Schroeder
(1945 –)

□ کاربرد نسخه اصلاح شده:

Kerberos ☞

Active Directory ☞

پروتکل Needham-Schroeder

1. $A \rightarrow T: ID_A \parallel ID_B \parallel n_A$
2. $T \rightarrow A: \{n_A \parallel k_s \parallel ID_B \parallel \{k_s \parallel ID_A\}K_{BT}\}K_{AT}$
3. $A \rightarrow B: \{k_s \parallel ID_A\}K_{BT}$
4. $B \rightarrow A: \{n_B\}k_s$

در نسخه اصلی پروتکل، نانس فقط رمز می‌شد و تصدیق هویت نمی‌شد.
5. $A \rightarrow B: \{n_B - 1\}k_s$

□ خصوصیات:

👉 دو گام آخر برای تأیید کلید (از سوی B) است.

معایب پروتکل Needham-Schroeder

□ این پروتکل نسبت به حمله تکرار آسیب پذیر است.

☞ مهاجم می تواند پیام ۳ پروتکل را تکرار کند.

□ Denning و Sacco علاوه بر یافتن عیب فوق، پروتکل

جدیدی را بر مبنای مهر زمانی پیشنهاد کردند.

□ **ایراد دیگر:** A نمی تواند از زنده بودن B و دریافت کلید توسط

وی مطمئن باشد.

☞ پیام ۴ مقداری تصادفی است (رمز شده یک نانس تصادفی) و به

A اطلاع خاصی نمی دهد.

1. $A \rightarrow T: ID_A \parallel ID_B$
2. $T \rightarrow A: \{ k_s \parallel ID_B \parallel t_T \parallel \{k_s \parallel ID_A \parallel t_T\}K_{BT} \}K_{AT}$
3. $A \rightarrow B: \{k_s \parallel ID_A \parallel t_T\}K_{BT}$
4. $B \rightarrow A: \{n_B\}k_s$
5. $A \rightarrow B: \{n_B - 1\}k_s$

□ استفاده از مُهر زمانی برای جلوگیری از حمله تکرار؛ ولی:

👉 همچنان A از زنده بودن B نمی‌تواند مطمئن شود.

👉 پیام ۳ را می‌توان بلافاصله فرستاد $B \leftarrow$ دو بار k_s را می‌پذیرد.

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

$$\square |NOW - t_T| < \Delta t_1 + \Delta t_2$$

☞ Δt_1 : اختلاف ساعت محلی با T

☞ Δt_2 : میزان تأخیر مورد انتظار در شبکه.

□ **حمله Gong**: وقتی ساعت T جلوتر از ساعت A یا B باشد.

☞ مهاجم می‌تواند پیام را نگه داشته و وقتی زمان A یا B با زمان

ارسال پیام یکی شد ارسال نماید.

☞ معروف به حمله منع - تکرار (Suppress-Replay).

حمله منع - تکرار و مقابله با آن

□ پروتکل Denning-Sacco نسبت به حمله منع - تکرار آسیب پذیر است.

□ روشهای مقابله

☞ همگام سازی زمان در ابتدای پروتکل با زمان T

☞ توافق از طریق نانس به جای توافق از طریق زمان

☞ ترکیب نانس و زمان ← پروتکل Neuman (نیاز به همگامی ساعتها ندارد؛ فقط از ساعت B استفاده می شود)

1. $A \rightarrow B: ID_A \parallel n_A$
2. $B \rightarrow T: ID_B \parallel n_B \parallel \{ID_A \parallel n_A \parallel \theta_B\}K_{BT}$
3. $T \rightarrow A: \{ID_B \parallel n_A \parallel k_s \parallel \theta_B\}K_{AT} \parallel$
 $\{ID_A \parallel k_s \parallel \theta_B\}K_{BT} \parallel n_B$
4. $A \rightarrow B: \{ID_A \parallel k_s \parallel \theta_B\}K_{BT} \parallel \{n_B\}k_s$

θ_B مدت زمان مجاز استفاده از k_s (بر حسب ساعت B) است. \square

در این مدت، نیاز به تماس مجدد با T برای دریافت k_s جدید نیست.

مفهوم «بلیت» در پروتکل Neuman

□ عبارت $\{ID_A \parallel k_s \parallel \theta_B\}K_{BT}$ برای A (شروع کننده پروتکل) مثل بلیت عمل می کند.

□ مادام که برحسب ساعت B به زمان θ_B نرسیده ایم، A می تواند با استفاده از بلیت و k_s نشست جدیدی را آغاز کند:

1. $A \rightarrow B: \{ID_A \parallel k_s \parallel \theta_B\}K_{BT} \parallel n'_A$
2. $B \rightarrow A: n'_B \parallel \{n'_A\}k_s$
3. $A \rightarrow B: \{n'_B\}k_s$

مبنای طراحی پروتکل‌های سری دوم

□ مبتنی بر رمز متقارن

☞ بدون شخص ثالث معتمد

☞ هر دو موجودیت، از قبل کلید طولانی مدتی را به اشتراک گذارده‌اند.

□ تصدیق هویت دو طرفه

پروتکل انتقال کلید (Key Transfer)

1. $A \rightarrow B: ID_A \parallel n_A$
2. $B \rightarrow A: \{ k_s \parallel ID_A \parallel ID_B \parallel n_A \parallel n_B \} K_{AB}$
3. $A \rightarrow B: \{n_B\}k_s$

□ پیام ۲ به A اطمینان می‌دهد که B زنده است و کلید K_{AB} را در اختیار دارد (چرا که B نانس A را با این کلید تصدیق هویت می‌کند).

☞ همچنین B کلید k_s را دارد، چون آن را تصدیق هویت نموده است.

□ پیام ۳ به B اطمینان می‌دهد که A زنده است و کلیدهای K_{AB} و k_s را در اختیار دارد (چرا که A نانس B را با k_s تصدیق هویت می‌کند، و k_s فقط توسط کسی که K_{AB} را دارد قابل خواندن است).

پروتکل توافق کلید (Key Agreement)

1. $A \rightarrow B: ID_A \parallel n_A$

2. $B \rightarrow A: \langle \langle ID_B \parallel n_A \rangle \rangle_{k_s} \parallel n_B$

3. $A \rightarrow B: \langle \langle ID_A \parallel n_B \rangle \rangle_{k_s}$

فقط تصدیق هویت پیام
(بدون محرمانگی)

□ در گام ۲ و ۳، مقدار k_s با اعمال تابع ویژه f به مقادیر n_A ، n_B و K_{AB} بدست می‌آید.

□ مثلاً f می‌تواند HMAC باشد:

$$k_s = \text{HMAC}_{K_{AB}}(n_A \parallel n_B)$$

دیفی – هلمن تصدیق هویت شده (ADH) – تلاش ۱

□ $g \in \mathbb{Z}_p^*$ عنصری از مرتبه اول فرد q ؛ کلید محاسبات به پیمانه p .

1. $A \rightarrow B: ID_A \parallel g^\alpha$

2. $B \rightarrow A: \langle \langle ID_B \parallel g^\alpha \parallel g^\beta \rangle \rangle K_{AB}$

3. $A \rightarrow B: \langle \langle ID_A \parallel g^\alpha \parallel g^\beta \rangle \rangle K_{AB}$

□ مقدار کلید نشست (k_s) با استفاده از روش دیفی – هلمن بدست می‌آید:

$$k_s = g^{\alpha\beta}$$

1. $A \rightarrow B: ID_A \parallel g^\alpha$

2. $B \rightarrow A: \langle\langle ID_B \parallel g^\alpha \parallel g^\beta \rangle\rangle K_{AB}$

3. $A \rightarrow B: \langle\langle ID_A \parallel g^\alpha \parallel g^\beta \rangle\rangle K_{AB}$

□ نه A و نه B کلید k_s را تأیید (confirm) نمی کنند.


دیفی - هلمن تصدیق هویت شده (ADH) - تلاش ۲


1. $A \rightarrow B: ID_A \parallel g^\alpha$

2. $B \rightarrow A: \ll \langle \langle ID_B \parallel g^\alpha \parallel g^\beta \rangle \rangle K_{AB} \gg k_s$

3. $A \rightarrow B: \ll \langle \langle ID_A \parallel g^\alpha \parallel g^\beta \rangle \rangle K_{AB} \gg k_s$

□ خاصیت مهم ADH:

برقراری امنیت پیشرو (Forward Secrecy) 

هزینه: محاسبات زمان بر دیفی - هلمن 

پروتکل انتقال کلید (Key Transfer)

1. $A \rightarrow B: ID_A \parallel n_A$
2. $B \rightarrow A: \text{Msg} \parallel \langle\langle \text{Sign}(\text{PR}_B, \text{Msg}) \rangle\rangle_{k_s}$
3. $A \rightarrow B: \langle\langle ID_A \parallel n_B \rangle\rangle_{k_s}$

$$\text{Msg} = ID_B \parallel n_A \parallel n_B \parallel E(\text{PU}_A, k_s)$$

مبنای طراحی پروتکل‌های سری سوم

□ مبتنی بر رمز نامتقارن

➡ بدون شخص ثالث معتمد

➡ هر موجودیت، یک زوج کلید عمومی و خصوصی طولانی-مدت دارد.

➡ کلید خصوصی هر موجودیت فقط در اختیار خود او است.

➡ کلید عمومی به نحوی امن در اختیار دیگران قرار گرفته است.

□ تصدیق هویت دو طرفه

ADH با تصدیق هویت توسط کلید عمومی

1. $A \rightarrow B: ID_A \parallel g^\alpha$

2. $B \rightarrow A: ID_B \parallel g^\beta \parallel$

$$\langle\langle \text{Sign}(\text{PR}_B, ID_B \parallel g^\alpha \parallel g^\beta) \rangle\rangle k_s$$

3. $A \rightarrow B: ID_A \parallel \langle\langle \text{Sign}(\text{PR}_A, ID_A \parallel g^\alpha \parallel g^\beta) \rangle\rangle k_s$

□ اهداف:

✎ اجرای دیفی – هلمن بدون هیچ گونه سر بار ارتباطی

✎ مقاومت در برابر حمله MITM

✎ بدون تأیید کلید (Key Confirmation) و تصدیق هویت دو
جانبه.

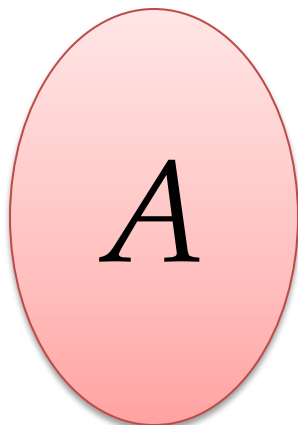
□ x و y به ترتیب کلید خصوصی A و B

□ کلیدهای عمومی به صورت $X = g^x$ و $Y = g^y$

□ H یک تابع درهم ساز (تصادفی) که برد آن \mathbb{Z}_q است.

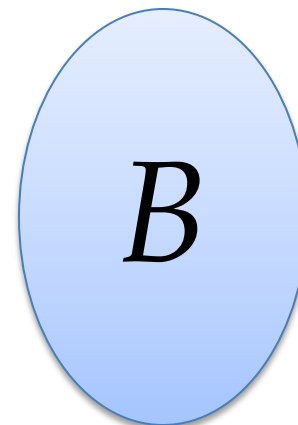
پروتکل HMQV – ۲

کلید خصوصی: x



کلیدهای عمومی:
 $X = g^x, \quad Y = g^y$

کلید خصوصی: y



$$d = H(D \parallel ID_B)$$
$$e = H(E \parallel ID_A)$$

$$k_s = (EY^e)\alpha + dx$$

=

$$k_s = (DX^d)\beta + ey$$