به نام خدا

دانشکده مهندسی کامپیوتر

تمرین چهارم درس مبانی امنیت اطلاعات

فصل چهارم: مدیریت کلید

استاد درس: دکتر شهریاری

تمرین تشریحی

سوال 1 (5 نمره)

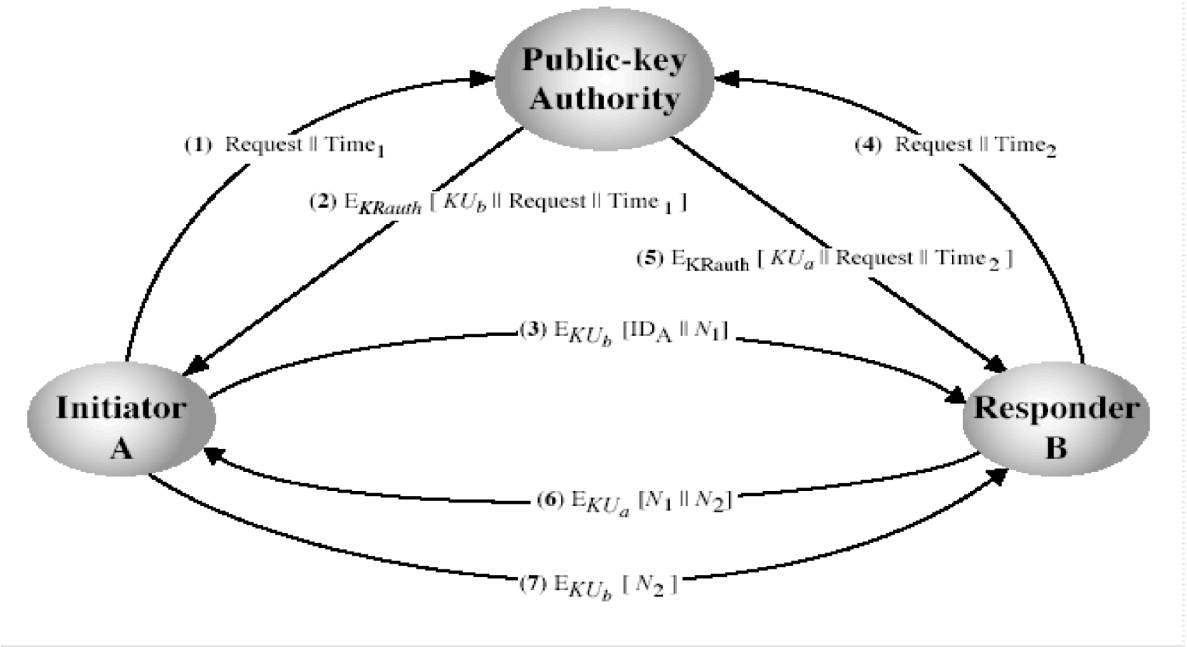
به هر یک از موارد زیر پاسخ کوتاه بدهید.

الف) مدیریت کلید چیست؟

ب) تهدیدهای مدیریت کلید را نام ببرید و هر یک را به زبان ساده تعریف کنید.

ج) کلید جلسه و کلید اصلی را تعریف کنید و بیان کنید که هر یک چه کاربردی در برقراری ارتباط امن دارند.

د) مرحله به مرحله‌ی شکل زیر را توضیح دهید[1]:



[1] در این سوال لازم به توضیح اضافه نیست؛ صرفا با چند جمله هر کدام از مراحل را تعریف کنید.

سوال 2 (5 نمره)

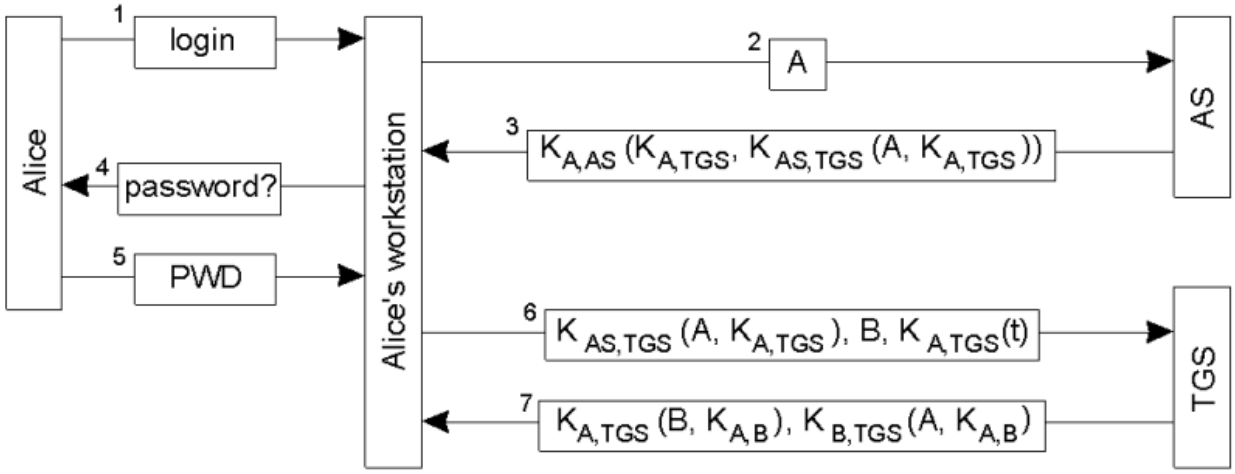
به سوالات زیر در مورد کربروس پاسخ دهید:

الف) چهار نیاز عمومی کربروس را توضیح دهید.

ب) تفاوت بین نسخه 4 و 5 کربروس را شرح دهید.

ج) با توجه به شکل زیر، کاربرد nonce یا همان t در شکل زیر و در پیام 6 را بنویسید.

د) دلیل رمزنگاری nonce را توضیح دهید.



سوال 3 (5 نمره)

در رابطه با مفاهیم کلید عمومی به سوالات زیر پاسخ دهید:

الف) گواهی کلید عمومی چیست و چه اطلاعاتی را شامل می‌شود؟

ب) مرجع گواهی (CA) چه نقشی در صدور گواهی‌ها و تأمین امنیت آن‌ها دارد؟

ج) مراحل صدور یک گواهی کلید عمومی چیست و چگونه می‌توان اعتبار آن را بررسی کرد؟

د) گواهی‌های کلید عمومی چه مزایایی دارند و چگونه امنیت تبادل کلیدها را تضمین می‌کنند؟

ه) چالش‌های استفاده از گواهی‌های کلید عمومی چیست و در کدام پروتکل‌های امنیتی کاربرد دارند؟

سوال 4 (5 نمره)

دو طرف، آلیس و باب، می‌خواهند با استفاده از الگوریتم دیفی-هلمن یک کلید جلسه مشترک ایجاد کنند.

شرایط اولیه:

- عدد اول: q=599
- مولد: α=5
- عدد خصوصی آلیس: xA = 117
- عدد خصوصی باب: xB= 219

سؤال:

الف) مقادیر عمومی y_a و y_b را که توسط آلیس و باب محاسبه و تبادل می‌شوند، به دست آورید.

ب) کلید جلسه k_{ab} را از دید هر دو طرف محاسبه کنید.

ج) اگر مهاجم به مقادیر q، α، y_a و y_b دسترسی داشته باشد، آیا می‌تواند کلید جلسه K_{ab} را محاسبه کند؟ دلیل بیاورید.

تمرین عملی:

در این بخش قرار است تا پروتوکل کربروس[2] که در اسلاید های درس با آن آشنا شده اید را در یک سامانه احراز هویت با معماری کلاینت - سرور[3]، پیاده سازی کنیم. در این تمرین باید موارد زیر را پیاده سازی کنید:

- مرکز توزیع کلید[4] ساده
- سرویس اعطای بلیط[5]
- سرویسی برای شبیه سازی ارتباط امن میان کاربر و سرور با استفاده از کتابخانه requests-kerberos پایتون

هدف از انجام این تمرین:

- آشنایی با نحوه کارکرد کربروس برای احراز هویت امن و مبتنی بر بلیط میان کاربران و سرویس های مختلف
- استفاده از کتابخانه requests-kerberos برای احراز درخواست های api یک اپلیکشن flask.

کربروس

کربروس یک پروتکل احراز هویت بر روی شبکه است که از بلیط برای اعتبارسنجی هویت کاربران بر روی یک شبکه غیر امن استفاده می‌کند. کربروس از مرکز توزیع کلید استفاده می‌کند تا با تولید بلیط، دسترسی به خدمت‌ها را بدون نیاز به ارسال رمز عبور بر روی شبکه ممکن سازد. مولفه های کلیدی کربروس عبارت اند از:

- مرکز توزیع کلید: وظیفه احراز کاربران و تولید بلیط را بر عهده دارد.
- سرویس اعطای بلیط: بلیط های خدمت را بر اساس بلیط های های اعطای بلیط[6] که از مرکز توزیع کلید گرفته است، تولید می‌کند.

---

[2] kerberos
[3] Client - Server
[4] Key distribution center (KDC)
[5] Ticket granting service
[6] Ticket-granting ticket

- خدمت: برنامه یا منبعی که کاربر با داشتن بلیط کربروس می‌خواهد به آن دسترسی داشته باشد.

نحوه کار کربروس

- احراز کاربر: کاربر با استفاده از نام کاربری و رمز عبورش توسط مرکز توزیع کلید، احراز می‌شود.
- بلیط اعطای بلیط: اگر کاربر احراز شده باشد، مرکز توزیع کلید بلیط اعطای بلیط را فراهم می‌کند تا برای دسترسی به سایر سرویس ها امکان درخواست دادن فراهم شود.
- بلیط خدمت: کاربر با ارائه بلیط اعطای بلیط به سرویس اعطای بلیط، می‌تواند یک بلیط برای درخواست منبع یا یک برنامه دریافت کند.
- دسترسی به خدمت: برنامه هدف بلیط خدمت را اعتبار سنجی می‌کند و دسترسی را به کاربر می‌دهد.

کتابخانه requests-kerberos

این کتابخانه احراز هویت کربروس را با کتابخانه محبوب پایتون یعنی requests تلفیق می‌کند. و به ما این امکان را می‌دهد تا با استفاده از ویژگی احراز کربروس، درخواست های http بر روی شبکه ارسال کنیم. فرایند مدیریت بلیط های کربروس در این کتابخانه اتومات شده است. از ویژگی ها کلیدی این کتابخانه می‌توان موارد زیر را نام برد:

- مطمئن می‌شود هم کاربر هم سرور یکدیگر را احراز کرده باشند.
- اتصال مبتنی بر بلیط به همراه اعتبارسنجی منبع

ساختار پروژه

به همراه دستورکار این تمرین، فایل های مورد نظر برای بخش عملی قرار داده شده اند. فایل‌های این تمرین به شکل زیر می‌باشند:

- Kerberos_service.py: یک برنامه نوشته شده با flask که مولفه های اصلی را پیاده سازی می‌کند.
- kerberos-request.py: در این فایل نحوه ارسال درخواست‌های احراز شده کربروسی پیاده سازی شده است. شما باید تغییرات خود را در این فایل اعمال کنید.
- Requirements.txt: کتابخانه های پایتونی که باید برای اجرای برنامه نصب باشند، در این فایل ذکر شده است.

مولفه های اصلی

- مرکز توزیع کلید (KDC): ثبت نام و احراز هویت کاربران را مدیریت می‌کند.
- سرور اعطای بلیط (TGS): بلیط‌ها را اعتبارسنجی می‌کند و بلیط خدمت تولید می‌کند.
- خدمت (Service): یک خدمت هدف که بلیط های خدمت را اعتبارسنجی می‌کند و بر اساس معتبر بودن بلیط، اجازه دسترسی به کاربر می‌دهد.
- کاربر (Client): درخواست‌ها را به برنامه flask با استفاده از کتابخانه requests-kerberos می‌زند.

نحوه اجرای پروژه

1. ابتدا پروژه را از این آدرس کلون کنید و وارد پوشه HW2 شوید. برای این کار می‌توانید از دستور زیر استفاده کنید:

```
git clone https://github.com/AUT-basics-of-security-fall-2024/HW4.git
cd HW4
```

2. کتابخانه های نیازمند به نصب پایتونی را با اجرای دستور زیر نصب کنید:

```
pip install -r requirements.txt
```

3. برنامه flask را با اجرای دستورات زیر، اجرا کنید:

```
cd kerberos_request
python kerberos_service.py
```

4. در یک پنجره ترمینال دیگر، با اجرای دستور زیر به برنامه flask درخواست ارسال کنید:

```
python kerberos-request.py
```

پیاده سازی

مقدار دهی kerberos_auth

در ابتدا باید متغیر kerberos_auth را با استفاده از تابع HTTPKerberosAuth که از کتابخانه requests_kerberos ایمپورت شده است، مقدار دهی کنید.

تابع make_kerberos_request

این تابع درخواست های مورد نیاز را به سرویس کربروس می‌زند. در این تابع شما باید دو نوع درخواست POST و GET را پیاده سازی کنید. برای انجام این بخش، باید از کتابخانه requests و متد های post و get استفاده کنید. آدرس اندپوینت به عنوان url، داده ورودی تابع به عنوان داده json، و متغیر kerberos_auth به عنوان auth به متد های نام برده داده می‌شوند.

روند اجرای برنامه

گام اول (ثبت نام کاربر)

ابتدا با ارسال درخواست POST به سرویس کربروس، کاربر را ثبت نام می‌کنیم.

گام دوم (تایید بلیط)

در این مرحله، با ارسال داده های احراز به سرویس کربروس با استفاده از متد GET، متغیر auth_response را مقدار دهی می‌کنیم. اگر عملیات احراز موفق آمیز بود، بلیط و کلید جلسه را دریافت خواهیم کرد.

در ادامه باید دو متغیر ticket و session_key را از دیکشنری auth_response با همین نام های کلید، استخراج و مقدار دهی کنید.

در ادامه با ارسال درخواستی به سرویس کربروس به همراه متغیر verify_ticket_data که شامل داده مربوط به بلیط خود است، تاییدیه بلیط خود را دریافت می‌کنیم.

متغیر verify_ticket_data یک دیکشنری است که بلیط مورد نظر را کلیدی به نام ticket در خود ذخیره کرده است.

گام سوم (تولید بلیط خدمت)

در این مرحله، برای تولید بلیط خدمت، درخواستی با متد GET به سرویس کربروس می‌زنیم تا نتیجه تولید بلیط خود را دریافت کنیم. برای ارسال این درخواست، متغیری به نام issue_ticket_data به سرویس کربروس ارسال می‌شود. این متغیر یک دیکشنری است با دو کلید:

- service_name: این کلید نام خدمت مورد نظر را شامل می‌شود (برای مثال می‌توانید از مقدار "example_service" استفاده کنید).
- session_key: این کلید، مقدار کلید جلسه را که در مراحل قبلی ذخیره کردیم، شامل می‌شود.

گام چهارم (تایید بلیط خدمت)

در این مرحله، برای تایید بلیط خدمت، باید درخواستی به سرویس کربروس با متد GET بزنیم و به همراه درخواست خود، متغیر verify_service_ticket_data را ارسال کنیم. این متغیر یک دیکشنری است با یک کلید به نام service_ticket که مقدار آن برابر با بلیط خدمت است. برای به دست آوردن مقدار بلیط خدمت، باید آن را از متغیر issue_ticket_response که کلیدی به نام service_ticket دارد، استخراج کنیم.

مثال استفاده

ایجاد کاربر

برای ایجاد یک کاربر، می توان با ارسال یک درخواست با متد POST به مسیر /register به همراه دو فیلد username و password در قالب JSON، کاربر جدید را ساخت.

احراز کاربر

با ارسال یک درخواست با متد GET به مسیر /authenticate به همراه نام کاربری و رمزعبور، یک بلیط کلید جلسه دریافت خواهید کرد.

اعتبارسنجی بلیط

با ارسال بلیط به مسیر /verify_ticket می‌توانید معتبر بودن آن را بسنجید

تولید بلیط خدمت

با استفاده از کلید جلسه و نام خدمت و ارسال آن به مسیر /issue_service_ticket یک بلیط خدمت دریافت کنید.

اعتبارسنجی بلیط خدمت

در نهایت، با ارسال بلیط خدمت به مسیر /verify_service_ticket با خدمت مورد نظر دسترسی پیدا کنید.

بخش امتیازی

تولید سرتیفیکیت با استفاده از X.509

در این بخش قرار است با استفاده از X.509 به تولید یک سرتیفیکیت بپردازیم.

X.509 استانداردی برای تعریف قالب گواهی کلید عمومی است. از X.509 در خیلی از پروتکل‌های اینترنتی، شامل TLS/SSL که مبنای HTTPS (پروتکل امن برای مرور وب) است، استفاده می‌شود. از این استاندارد در کاربردهای آفلاین، مثل امضای الکترونیکی، نیز استفاده می‌شود.

برای پیاده سازی در این بخش، از ماژول X. 509 موجود در کتابخانه cryptography استفاده می‌کنیم. برای آشنایی با متدها و توابع این ماژول می‌توانید به این لینک مراجعه کنید.

برای این تمرین، نیاز است تا تغییرات خود را در فایل نوتبوک x509_certificate.ipynb اعمال کنید.

گام اول: تولید لیست alternative name

لیست نام‌های جایگزین برای افزودن شناسه‌های اضافی (مثل نام‌های DNS، آدرس‌های IP، ایمیل‌ها) به گواهینامه استفاده می‌شود تا انعطاف‌پذیری و سازگاری در کاربردهای مختلف فراهم شود.

در این گام ابتدا باید متغیر certificate_name را مقدار دهی کنید. برای این کار باید از متد Name استفاده کنید. آرگومان این متد، یک لیست است که آن را با استفاده از آرگومان هایی که در نوتبوک مشخص شده اند و به متد NameAttribute می‌دهید، پر می‌کنید.

لیست اولیه از نام های جایگزین را با به کارگیری نام dns تشکیل می دهیم. متغیر subject_alternative_names، مقدار نهایی نام های جایگزین ما خواهند بود.

برای مقدار دهی به این متغیر باید نام dns و آدرس IP را برای IP عمومی به لیست alternative_names اضافه کنید. برای این امر از متد های DNSName و IPAddress می توانید استفاده کنید. همچنین در صورت موجود بودن IP خصوصی، مقدار های dns و آدرس IP این IP را نیز به لیست اضافه کنید.

در نهایت با استفاده از متد SubjectAlternativeName و استفاده از لیست تولید شده به عنوان آرگومان متد، متغیر را مقدار دهی کنید.

گام دوم: تعیین مدت انقضای سرتیفیکیت

در این گام نیاز است تا مهلت استفاده و معتبر بودن سرتیفیک را مشخص کنیم.

برای پیاده سازی این گام نیاز است تا متغیر deadline را مقداردهی کنیم. برای این امر می توانید از تابع timedelta استفاده کنید و زمان دلخواه خودتان را وارد کنید.

گام سوم: تولید کلید خصوصی

تولید کلید خصوصی با RSA برای ایجاد یک کلید امن جهت رمزنگاری و امضای دیجیتال استفاده می‌شود.

برای مقدار دهی به متغیر key می توانید از متد generate_private_key ماژول rsa استفاده کنید. آرگومان های این متد به شکل زیر است:

- Public exponent: این مقدار باید برابر با 65537 باشد.
- key_size: اندازه کلید را مقدار دهی کنید
- backend: از این مقدار برای برقراری ارتباط میان واسط های رمزنگاری و API های openssl استفاده می‌شود. مقدار این آرگومان را برابر با backend قرار دهید.

متغیر encoding مشخص کننده نوع کدگذاری است. در این تمرین از نوع PEM استفاده می‌کنیم. برای مقدار دهی به این متغیر از ماژول serialization می‌توانید استفاده کنید و نوع PEM را برای کدگذاری انتخاب کنید.

متغیر private_form فرمت کلید را مشخص می کند. رایج ترین فرمت کلید، Traditional OpenSSL می‌باشد که در این تمرین نیز باید همین مقدار به کار برده شود. برای این امر نیاز است تا دوباره از ماژول serialization استفاده کنید.

متغیر encryption_algorithm الگوریتم رمزنگاری برای تولید کلید را مشخص می‌کند. در این تمرین نمیخواهیم کلید خصوصی مان نیازمند گذرواژه باشد. برای مقدار دهی این متغیر می‌توانید از متد NoEncryption ماژول serialization استفاده کنید.

گام چهارم: تعیین محدودیت ها

در این گام، محدودیت های اولیه گواهی خود را تعیین می‌کنیم. برای پیاده سازی این بخش باید متغیر basic_constraints را مقدار دهی کنیم. برای این امر می‌توان از متد BasicConstraints استفاده کرد.

آرگومان‌های مورد نیاز در نوتبوک مشخص شده اند.

گام پنجم: تولید سرتیفیکیت

در این گام که گام نهایی می‌باشد با اجرای متد CertificateBuilder گواهی خود را تولید می‌کنیم.

در این گام نیاز است تا با اجرای کد داده شده، سرتیفیکیت خود را تولید کرده و خروجی آن را مشخص کنید.

آنچه خواهید آموخت

هدف انجام این پروژه آشنایی و یادگیری شما با موارد زیر می‌باشد:

- فهمیدن نقش کربروس در امنیت شبکه و احراز کاربران.
- پیاده سازی یک سامانه ساده احراز توسط سرتیفیکیت X.509.
- استفاده از کتابخانه requests-kerberos برای مدیریت درخواست های امن و پاسخ ها.

موارد تحویلی

لطفا یک فایل zip حاوی کل فایل های پروژه (چه فایل هایی که توسط شما تغییر داده شده اند؛ چه فایل هایی که تغییری نداشته اند) اپلود کنید.

موفق باشید