پروژه‌ی دوم

همان طور که در پروژه‌ی قبلی مشاهده کردید، پیاده سازی یک ارتباط امن میان server و client با استفاده از کلید فیزیکی، مشکلاتی را دربرداشت. از جمله اینکه اگر تعداد client ها برای server زیاد شود، استفاده از کلید فیزیکی غیرممکن می‌شود. در این پروژه قصد داریم با استفاده از رمزنگاری نامتقارن، کلید جلسه را مبادله کنیم. علاوه بر آن، امکاناتی مانند احراز اصالت پیام نیز باید پیاده سازی شود.

نکات پیاده‌سازی

- یک روند مبادله‌ی امن کلید جلسه طراحی کنید و با استفاده از کلید جلسه، همه‌ی ارتباطات را رمزگذاری کنید. برای این کار اگر از جاوا استفاده می‌کنید، می‌توانید از Java Crypto Extension (JCE) استفاده کنید که از روش‌های مبادله‌ی کلید Deffie-Hellman و رمزگذاری نامتقارن پشتیبانی می‌کند. ابتدا باید با الگوریتم‌های رمزگذاری نامتقارن، کلید عمومی و خصوصی را برای هریک از طرفین تعیین کرده و سپس با استفاده از آن، کلید جلسه را مبادله کنید.
- برای هر کدام از سرور و کلاینت، یک جدول برای نگهداری کلید عمومی و خصوصی برای رمزنگاری نامتقارن نگهداری کنید. برای server های خارجی، هر client باید نام server و کلید عمومی آن را ذخیره کند. همچنین کلید خصوصی خود کاربر برای ارتباط با server های مختلف، با استفاده از نام server و کلید خصوصی کاربر برای ارتباط با آن server ذخیره شود. در واقع یک جدول سه ستونه داریم که ستون اول username برای server، ستون دوم public key برای server و ستون سوم، private key کاربر برای ارتباط با server است. دقت کنید که این جدول را هر دو سمت server و client باید داشته باشند.
- برای ارتباط امن، با استفاده از رمزنگاری متقارن، کلید جلسه را با استفاده از رمز نگاری نامتقارن مبادله کنید و سپس همه‌ی پیغام‌ها را با استفاده از کلید جلسه مبادله کنید. (در مبدا رمزنگاری و در مقصد رمزگشایی کنید.)

- برای این که تشخیص دهیم که آیا پیغام در میانه‌ی راه دستکاری شده‌است یا نه، روند احراز اصالت پیغام را هم پیاده سازی می‌کنیم، بدین صورت که قبل از رمزگذاری، با استفاده از یکی از توابع امن درهم سازی مانند SHA، یک MAC برای پیغام ایجاد کرده، آن را به همراه بسته رمز گذاری کرده و می‌فرستیم. در مقصد، ابتدا بسته را رمزگشایی کرده، MAC آن را چک می‌کنیم. اگر نادرست بود، آن را دور میریزیم و یک پیغام خطا به فرستنده می فرستیم. در صورتی که یکی از packet های فایل، دستکاری شده و اشتباه ارسال شده باشد، بعد از ارسال خطا به فرستنده، کل فایل را مجددا از ابتدا ارسال کند.
- هر کلید، یک برچسب زمانی و مدت انقضا دارد که باید قبل از گذشتن مدت انقضای آن، کلید جدید مبادله شود. (مثلا مدت انقضا را یک دقیقه می‌توانید در نظر بگیرید)
- توجه کنید که از Secure Socket های آماده نمی‌توانید استفاده کنید و باید یک ارتباط TCP برقرار کرده و امکانات امنیتی آن را با استفاده از موارد ذکر شده پیاده سازی کنید.
- توجه کنید که برای ارسال داده‌ها، بصورت Byte Stream عمل کنید تا اینکه بتوان انواع فایل‌ها را هم رمزنگاری کرد.
- برای تست برنامه، باید روی سیستم‌های جداگانه صورت گیرد و یا اینکه از Virtual Machine استفاده شود.

امتیازی

- در صورتی که یک فایل را ارسال می‌کنید و در میانه‌ی راه، یکی از packet ها دستکاری شود، بگونه‌ای که در مقصد، MAC آن با پیغام تطابق نداشته باشد، بعد از دور ریختن پیغام و ارسال error به فرستنده، از همان packet ای پیغام را شروع کند که اشتباه فرستاده شده است، نه از ابتدا.

نکات دیگر

- پروژه را باید هر نفر به تنهایی پیاده سازی کند و تحویل حضوری دارد که زمان آن متعاقبا اعلام خواهد شد.
- کدها بررسی و در صورت وجود تطابق تقلب گرفته می‌شود.
- مهلت ارسال پروژه تا تاریخ جمعه ۲۹ آبان ماه ۱۳۹۸ می‌باشد.