توضیحات کلی

۱. کافی است یکی از پروژه‌ها را به دلخواه انتخاب کنید.
۲. پروژه‌ی دوم را می‌توانید در گروه‌های دو نفره نیز انجام دهید ولی پروژه اول را باید به صورت فردی انجام دهید.
۳. کدهای پروژه را به همراه گزارشی به فرمت pdf به صورت یک فایل ZIP با نام شماره پروژه و شماره دانشجویی خود بارگذاری نمایید. در گزارش خود در مورد ساختار کلی الگوریتمی که به کار برده‌اید و کارکرد تمام کدها توضیح دهید. تمام فایل‌ها را نیز با شماره دانشجویی خود نام‌گذاری نمایید. به عنوان مثال برای شخصی با شماره دانشجویی 12345678، کدی با نام TIGER را TIGER_12345678 نام‌گذاری نمایید و همچنین اگر پروژه انتخابی پروژه اول باشد، نام فایل ZIP، Project1_12345678 نام‌گذاری شود.
۴. گزارشی از الگوریتم به کار رفته و نتایج پروژه به همراه کدهای پروژه تحویل دهید.
۵. به هیچ عنوان از کدهای سایر دوستان استفاده ننمایید! تشابه کدها بررسی می‌شود و در صورت مشاهده تشابه بین پروژه‌ها نمره‌ای به هیچ یک از پروژه‌های مشابه تعلق نمی‌گیرد.
۶. مهلت تحویل پروژه ۴ مرداد است.
۷. روز یکشنبه مورخ ۵ مرداد، تحویل مجازی پروژه در سامانه آموزش مجازی درس خواهد بود. زمان‌بندی دقیق اعلام خواهد شد.

پروژه اول – کیف پول دیجیتال

برای این پروژه شما باید یک نرم افزار برای نگه داری اطلاعات محرمانه نظیر اطلاعات حساب کاربری پیاده سازی کنید. از آنجا که اصل محرمانگی در حفظ چنین اطلاعاتی اهمیت دارد قبل از ذخیره سازی، اطلاعات باید رمزگذاری شود. برای این کار شما باید با استفاده از نام کاربری (user name) و پسورد (و یا فقط پسورد) و استفاده از salt و توابع Hash کلید مورد نیاز برای رمزگذاری متقارن اطلاعات را تولید کنید.

salt داده‌ای تصادفی است که در کنار پسورد قرار می‌گیرد، و این دنباله‌ی جدید (پسورد+salt) به عنوان ورودی به تابع چکیده‌ساز داده می‌شود. این کار امنیت را در برابر حمله جستجوی جامع افزایش می‌دهد.[1]

توضیحات:

۱) نرم افزار باید امکان معرفی کاربر جدید را داشته باشد. دقت کنید که پسورد کاربر به هیچ طریقی (plain- hashed-encrypted) نباید جایی ذخیره شود. اگر نیاز دارید از صحت پسورد ورودی کاربر مطلع شوید باید از روشی غیر از ذخیره پسورد استفاده کنید.

۲) نرم افزار باید امکان تغییر پسورد را در اختیار کاربر قرار دهد. مشخص است که پس از تغییر پسورد اطلاعات باید در صورت نیاز رمزگشایی و با کلید وابسته به پسورد جدید رمز گذاری شود.

۳) برای رمزگذاری متقارن اطلاعات می‌توانید از AES استفاده کنید. هم چنین برای توابع چکیده‌ساز می‌توانید از توابع SHA-2، SHA-3، WHIRLPOOL و یا TIGER استفاده نمایید. برای این توابع می‌توانید از کتابخانه‌های موجود و یا کدهای آماده استفاده نمایید. در صورتی که کد این توابع را خودتان آماده کنید، امتیاز اضافی خواهد داشت (درگزارش خود مشخص نمایید کد را خود آماده کرده‌اید یا خیر).

۴) شما آزاد هستید اطلاعات **رمزگذاری** شده را با هر فرمتی روی دیسک یا پایگاه داده ذخیره کنید.

---

[1] معمولا کاربران پسوردهای مشخصی را انتخاب می‌کنند. مهاجم می‌تواند همه‌ی این پسوردهای ممکن را در جدولی در نظر بگیرد و خروجی تابع چکیده‌ساز معادل هر کدام را محاسبه نماید. از طرفی دیگر اگر مهاجم به خروجی تابع چکیده‌ساز کاربران (که معمولا در پایگاه داده‌ای ذخیره شده) دسترسی داشته باشد، با مقایسه این خروجی‌ها با جدولی که خود تشکیل داده است، می‌تواند پسورد کاربران را تشخیص دهد. استفاده از salt به دلیل تصادفی ساختن پسورد، احتمال چنین حمله‌ای را کاهش می‌دهد.

پروژه دوم – ارتباطات محلی امن

هدف از این پروژه امن سازی ارتباطات درون شبکه های محلی (LAN) از طریق رمزنگاری متقارن است. یک راه برای حل این مسئله آن است که برای هر دو گره یکتا در شبکه یک کلید متقارن وجود داشته باشد. واضح است که برای این روش نیاز به O(n2) کلید متقارن است و هر گره باید کلید های ارتباطی با گره‌های دیگر را ذخیره کند. از آنجا که چنین روشی علاوه بر دارا بودن مشکلات امنیتی حفظ کلید در هر گره، برای شبکه‌های بزرگ بهینه نیست.

انتخاب دیگر استفاده از مرکز تولید کلید مطمئن است. این مرکز که یکی از گره‌های شبکه است و وظیفه دارد برای ارتباطات بین گره‌ها کلید تولید کرده و در اختیار آن‌ها قرار دهد. در این روش گره‌های A و B در مرحله راه اندازی، به ترتیب کلیدهای مخفی متقارن Kas و Kbs را با مرکز تولید کلید به اشتراک می‌گذارند سپس در مراحل زیر گره‌های A و B به کلید موقت نشست Kab دست می‌یابند.

- A → S: IDA + IDB
- S → A: E(Kas, [Kab + IDB + E(Kbs, [Kab, IDA])])
- A → B: E(Kbs, [Kab, IDA])
- B → A: E(Kab, N)
- A → B: E(Kab, [f(N) + پیام])

الف) پروتکل توافق کلید گفته شده را پیاده سازی کنید.

ب)دو گام آخر پروتکل (استفاده از تک‌شمار) برای تایید کلید می‌باشد. همچنین این پروتکل در مقابل حمله تکرار آسیب‌پذیر است. با استفاده از مهرزمانی سعی کنید این پروتکل را به نحوی اصلاح کنید که در مقابل حمله تکرار مقاوم باشد همچنین در پروتکل پیشنهادی سعی کنید تعداد ارتباطات ممکن را به حداقل برسانید. سپس پروتکل پیشنهادی خود را پیاده سازی کنید.