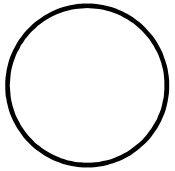
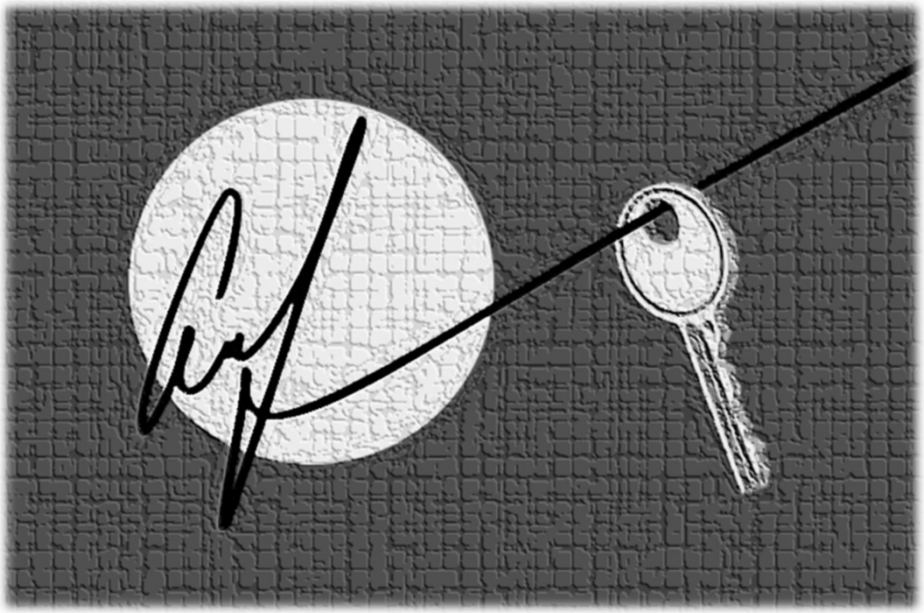
امضای دیجیتال



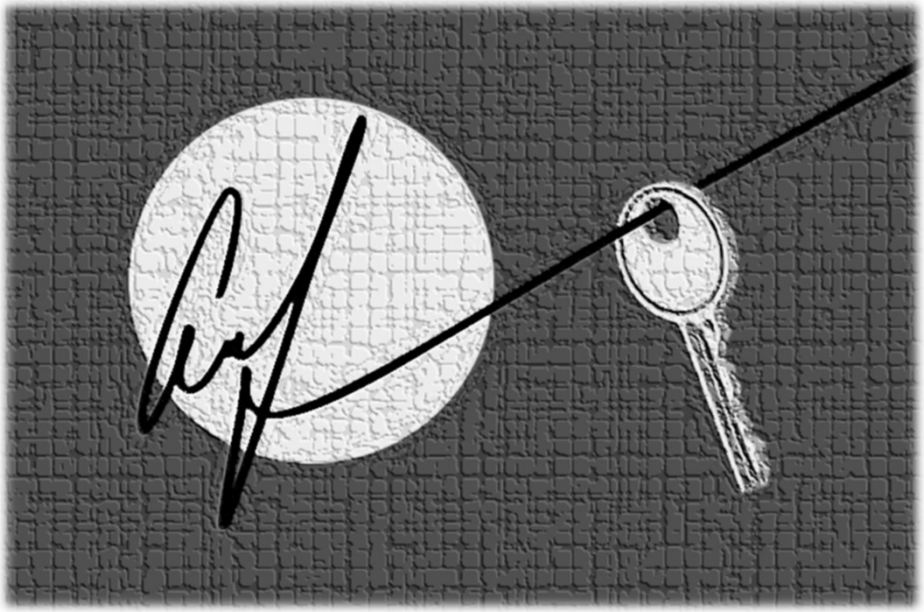


1. **امضای دیجیتال**
- امضای دیجیتال و هدف آن
- ویژگی های کلیدی: اصالت، یکپارچگی و عدم انکار.

۲. **تعاریف**
- تعاریف و نمادهای رسمی
- بحث در خصوص اهداف امنیتی امضای دیجیتال

۳. **پارادایم Hash-and-Sign**
- نحوه استفاده از توابع هش در امضای دیجیتال
- مثال ها و مزایای این رویکرد

۴. **امضاهای مبتنی بر RSA**
- در مورد امضاهای RSA ساده بحث میشود
- استانداردهای RSA-FDH و PKCS #1 را توضیح میدهد



۵. **امضاهای مسئله لگاریتم گسسته**

- مروری بر طرح های شناسایی و امضاها
- طرح های شناسایی/امضای Schnorr را معرفی میکند
- DSA و ECDSA را مورد بحث قرار میدهد

۶. **گواهی ها و زیرساخت های کلید عمومی (PKIs)**

- نقش گواهینامه ها را در تأیید کلیدهای عمومی توضیح میدهد
- در مورد مفهوم PKIصحبت میکند

۷. **TLS**

- نقش امضای دیجیتال در ایمن سازی TLS را برجسته کنید.
- گردش کار یک جلسه ارتباط امن را توضیح دهید.

۸. **Signcryption**

- به طور خلاصه مفهوم **Signcryption** را معرفی میکند

۹. **مراجع و تمرینات**

- برای تمرین مطالب و منابع خواندنی اضافی را ذکر میکند.

بخش ۱: امضاهای دیجیتال

امضای دیجیتال چیست؟

امضای دیجیتال مکانیزم رمزنگاری است که برای اطمینان از:

Authenticity: هویت فرستنده را تأیید می کند.

Integrity: تأیید می کند که پیام در حین انتقال تغییر نکرده است.

Non-repudiation: مانع از انکار فرستنده می شود که پیام را ارسال کرده است.

امضای دیجیتال معادل دیجیتالی امضاهای دست نویس یا مهرهای مهر شده است، اما به دلیل مبانی رمزنگاری، آنها بسیار ایمن تر هستند.

بخش ۱: امضاهای دیجیتال

اجزای کلیدی اسکیمای امضای دیجیتال

یک اسکیمای امضای دیجیتال معمولاً شامل سه الگوریتم است:

۱. تولید کلید Key Generation (KeyGen)

- یک جفت کلید تولید می کند: یک کلید خصوصی (که برای امضا استفاده می شود) و یک کلید عمومی (برای تأیید).

۲. الگوریتم امضاء Signing Algorithm (Sign)

- پیام و کلید خصوصی را به عنوان ورودی می گیرد.
- خروجی یک امضای دیجیتال

۳. الگوریتم تأیید Verification Algorithm (Verify)

- پیام، امضای دیجیتال و کلید عمومی را به عنوان ورودی می گیرد.
- یک مقدار بولی را خروجی می دهد که نشان می دهد آیا امضا معتبر است یا خیر.

بخش ۱: امضاهای دیجیتال

موارد استفاده در دنیای واقعی

- **ارتباط امن با ایمیل:** اطمینان حاصل می کند که ایمیل ها توسط فرستنده ادعا شده ارسال شده و دستکاری نشده اند.
- **توزیع نرم افزار:** صحت به روز رسانی یا بسته های نرم افزار را تأیید می کند.
- **تراکنش های مالی:** مجوز ایمن را برای پرداخت های آنلاین فراهم می کند.
- **بلاک چین:** تراکنش ها را تایید می کند و یکپارچگی داده ها را در سیستم های غیرمتمرکز تضمین می کند.

بخش ۱: امضاهای دیجیتال

فرآیند سطح بالا

سمت فرستنده:

- فرستنده از کلید خصوصی خود برای امضای پیام استفاده می کند.
- امضا به همراه پیام ارسال می شود.

سمت گیرنده:

- گیرنده از کلید عمومی فرستنده برای تأیید امضا استفاده می کند.
- در صورت موفقیت آمیز بودن تأیید، پیام معتبر و بدون دستکاری در نظر گرفته می شود.

بخش ۱: امضاهای دیجیتال

چرا امضای دیجیتال مهم است؟

- آنها ارتباطات ایمن را در محیط های غیر قابل اعتماد مانند اینترنت امکان پذیر می کنند.
- آنها سنگ بنای پروتکل های رمزنگاری مدرن، مانند TLS، بلاک چین و سیستم های رای گیری الکترونیکی هستند.

بخش دوم: تعاریف

در این بخش، به تعاریف رسمی و ویژگی‌های امنیتی طرح‌های امضای دیجیتال می‌پردازیم. این امر درک محکمی از مبانی نظری را تضمین می کند.

تعریف رسمی طرح امضای دیجیتال

طرح امضای دیجیتال شامل سه الگوریتم است:

- KeyGen
- Sign
- Verify

بخش دوم: تعاریف

:KeyGen

یک الگوریتم احتمالی که یک جفت کلید تولید می کند

$$(\mathrm{sk}, \mathrm{pk}) \leftarrow \mathrm{KeyGen}(1^{\lambda})$$

- sk: Private (signing) key
- pk: Public (verification) key
- λ: Security parameter (controls the strength of the scheme)

بخش دوم: تعاریف

Sign:

یک الگوریتم (احتمالاً احتمالی) که از کلید خصوصی sk برای تولید یک امضا برای یک پیام معین m استفاده می کند.

$$\sigma \leftarrow \text{Sign}(\text{sk}, m)$$

Verify:

یک الگوریتم قطعی که از کلید عمومی pk، پیام m و امضا σ برای تأیید اعتبار امضا استفاده می کند.

$$\text{Verify}(\text{pk}, m, \sigma) \rightarrow \{\text{True}, \text{False}\}$$

بخش دوم: تعاریف

خاصیت درستی

برای اینکه طرح امضای دیجیتال درست باشد، موارد زیر باید برای همه جفت‌های کلید معتبر (pk ،sk) و همه پیام‌های m وجود داشته باشد:

$$\mathrm{Verify}(\mathrm{pk}, m, \mathrm{Sign}(\mathrm{sk}, m)) = \mathrm{True}$$

این تضمین می کند که امضای تولید شده توسط کلید خصوصی همیشه می تواند توسط کلید عمومی مربوطه تایید شود.

بخش دوم: تعاریف

اهداف امنیتی

یک طرح امضای دیجیتال امن باید موارد زیر را رعایت کند:

جعل ناپذیری: (Unforgeability)

- مهاجم نباید بتواند بدون دسترسی به کلید خصوصی یک امضای معتبر برای پیام ایجاد کند.
- این قابلیت و هدف به دلیل حمله Existential Unforgeability under Chosen Message Attack (EUF-CMA) رسمیت یافت

عدم انکار: (Non-repudiation)

امضا کننده نمی تواند امضای یک پیام را انکار کند زیرا امضا به طور منحصر به فردی به کلید خصوصی او گره خورده است.

صداقت: (Integrity)

هر گونه تغییر در پیام امضا شده، امضا را باطل می کند.

بخش دوم: تعاریف

مدل خصمانه

مهاجم در مدل EUF-CMA:

- به کلید عمومی pk دسترسی دارد.
- می تواند از اوراکل امضا کننده پرس و جو کند تا برای پیام های مورد نظر خود امضا بگیرد.
- هدف آنها تولید یک امضای معتبر σ برای یک پیام جدید *m است که از اوراکل درخواست نشده است.

بخش دوم: تعاریف

علامت گذاری ریاضی

M: فضای پیام (مجموعه همه پیام های معتبر).

S: فضای امضا (مجموعه همه امضاهای ممکن).

t: پیچیدگی زمانی (منابع محاسباتی دشمن).

ϵ: احتمال جعل موفقیت آمیز امضا توسط دشمن.

تعاریف کلیدی

امنیت یک امضای دیجیتال به پیچیدگی الگوریتم های رمزنگاری (به عنوان مثال، فاکتورگیری اعداد صحیح بزرگ، لگاریتم های گسسته) متکی است.

تعاریف دقیق تضمین می کند که امضاهای دیجیتال حتی در برابر حملات پیچیده ایمن باقی می مانند.

بخش سوم: پارادایم Hash-and-Sign

پارادایم Hash-and-Signیک رویکرد پرکاربرد در ساخت طرح های امضای دیجیتال کارآمد و ایمن است. استفاده از توابع هش رمزنگاری را با یک الگوریتم امضای اولیه ترکیب می‌کند تا عملکرد و امنیت را تضمین کند.

چرا از پارادایم Hash-and-Sign استفاده کنیم؟

- امضای مستقیم پیام های بزرگ با استفاده از الگوریتم های رمزنگاری (مانند RSAیا DSA) می تواند از نظر محاسباتی گران باشد. در عوض، پارادایم Hash-and-Sign این فرآیند را با موارد زیر ساده می کند:
- کاهش اندازه پیام به یک خلاصه با طول ثابت با استفاده از یک تابع هش.
- امضای خلاصه به جای کل پیام.

بخش سوم: پارادایم Hash-and-Sign

چگونه کار میکند:

- یک تابع هش رمزنگاری H به پیام m اعمال می شود
- h صرف نظر از طول پیام اصلی، یک خلاصه هش با اندازه ثابت است
- کلید خصوصی sk برای امضای هش h استفاده می شود
- برای تأیید، گیرنده هش پیام دریافتی m را محاسبه می کند
- سپس، گیرنده با استفاده از کلید عمومی pk بررسی می کند که آیا امضای σ برای h′ معتبر است

How It Works

1. **Hashing the Message**:
   - A cryptographic hash function H is applied to the message m:
     $$h = H(m)$$
   - h is a fixed-size hash digest, regardless of the original message's length.
2. **Signing the Digest**:
   - The private key sk is used to sign the hash digest h:
     $$\sigma = \mathrm{Sign}(\mathrm{sk}, h)$$
3. **Verification**:
   - To verify, the receiver computes the hash of the received message m:
     $$h' = H(m)$$
   - Then, the receiver checks whether the signature σ is valid for h' using the public key pk:
     $$\mathrm{Verify}(\mathrm{pk}, h', \sigma)$$

بخش سوم: پارادایم Hash-and-Sign

مزایای پارادایم Hash-and-Sign

کارایی: (Efficiency)

- هش کردن بسیار سریعتر از امضا است، به خصوص برای پیام های بزرگ.
- بار محاسباتی روی الگوریتم امضا را کاهش می دهد.

امنیت: (Security)

امنیت طرح به ویژگی های تابع هش و الگوریتم امضای زیرین بستگی دارد.

انعطاف پذیری: (Flexibility)

الگوریتم امضای یکسانی را می توان با توابع هش مختلف استفاده کرد و در صورت ناامن شدن یک تابع هش، امکان ارتقاء آسان را فراهم می کند.

بخش سوم: پارادایم Hash-and-Sign

توابع هش رمزنگاری

برای اینکه پارادایم Hash-and-Sign ایمن باشد، تابع هش H باید ویژگی های زیر را داشته باشد:

مقاومت قبل از تصویر: (Pre-image Resistance)

با در نظر گرفتن هش h، یافتن پیام m باید از نظر محاسباتی غیرممکن باشد به طوری که $H(m)=h$

مقاومت پیش تصویر دوم: (Second Pre-image Resistance)

با توجه به یک پیام $m1$، یافتن یک پیام متفاوت m2 از نظر محاسباتی غیر ممکن است به طوری که $H(m1)=H(m2)$

مقاومت در برابر برخورد: (Collision Resistance)

یافتن دو پیام مجزا m1 و m2 باید از نظر محاسباتی غیرممکن باشد به طوری که $H(m1)=H(m2)$

توابع هش محبوب عبارتند از SHA-256، SHA-3 و BLAKE2.

بخش سوم: پارادایم Hash-and-Sign

برنامه های کاربردی

گواهی های دیجیتال:

Hash-and-Sign در گواهی های X.509 برای امضای محتوای گواهی استفاده می شود.

پروتکل های SSL/TLS:

با امضای داده های session، ارتباط ایمن را تضمین می کند.

بلاک چین:

تراکنش ها برای یکپارچگی و صحت، هش و امضا می شوند.

بخش سوم: پارادایم Hash-and-Sign

نقاط ضعف بالقوه

- اگر تابع هش H ضعیف باشد (به عنوان مثال، در برابر برخورد آسیب پذیر است)، کل طرح می تواند به خطر بیفتد.
- امنیت الگوریتم امضا نیز باید قوی باشد.

مثال گویا

1. **Message**: "Hello, Digital Signatures!"
2. **Hash Function**: SHA-256
   - Hash digest: $H(m) = 2\text{cf24dba5fb0a}...$ (a fixed-length string)
3. **Sign**:
   - Use RSA with a private key to sign $H(m)$: $\sigma = \text{Sign}(\text{sk}, \text{H}(\text{m}))$
4. **Verify**:
   - Receiver computes $H(m')$ and checks $\text{Verify}(\text{pk}, \text{H}(\text{m'}), \backslash\text{sigma})$.

قسمت چهارم: امضاهای مبتنی بر RSA

این بخش، طرح‌های امضای دیجیتال مبتنی بر RSA را معرفی می‌کند که از اولین و پرکاربردترین روش‌های امضای دیجیتال هستند. امضاهای مبتنی بر RSA از ویژگی‌های ریاضی الگوریتم RSA، یک سیستم رمزنگاری کلید عمومی استفاده می‌کنند.

مروری بر امضاهای دیجیتال مبتنی بر RSA

الگوریتم RSA را می توان با معکوس کردن نقش های رمزگذاری/رمزگشایی آن برای امضای دیجیتال تطبیق داد:

- برای امضای پیام از کلید خصوصی استفاده می شود.
- کلید عمومی برای تأیید امضا استفاده می شود.
- این فرآیند بر دشواری فاکتورگیری اعداد صحیح بزرگ است که امنیت RSA را پایه ریزی می کند.

قسمت چهارم: امضاهای مبتنی بر RSA

الگوریتم های کلیدی

یک طرح امضای مبتنی بر RSA شامل الگوریتم های زیر است:

1. **Key Generation**:
   - Choose two large prime numbers p and q, and compute $N = p \cdot q$.
   - Compute $\phi(N) = (p-1)(q-1)$.
   - Choose a public exponent e such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$.
   - Compute the private exponent d such that $e \cdot d \equiv 1 \pmod{\phi(N)}$.
   - The public key is (N, e), and the private key is (N, d).
2. **Signing**:
   - Given a message m, compute its hash $h = H(m)$ using a cryptographic hash function.
   - Use the private key d to compute the signature:
     $$\sigma = h^d \pmod{N}$$
3. **Verification**:
   - Given the signature σ, the public key (N, e), and the message m:
     - Compute $h' = H(m)$.
     - Verify the signature by checking:
       $$\sigma^e \pmod{N} = h'$$
       If true, the signature is valid.

قسمت چهارم: امضاهای مبتنی بر RSA

انواع طرح های امضای مبتنی بر RSA

امضاهای RSA ساده:

- مستقیماً الگوریتم RSA را برای امضای هش پیام اعمال کنید.
- ساده اما در برابر حملات خاص (مانند جعل وجودی) ایمن نیست.

RSA-FDH (هش دامنه کامل):

- هش پیام به کل محدوده ورودی های RSA ممکن نگاشت می شود.
- تضمین های امنیتی قوی تر از RSA ساده.

استانداردهای PKCS #1:

- پیاده سازی های عملی امضاهای RSA اغلب از استاندارد PKCS #1 پیروی می کنند که شامل طرح های padding برای جلوگیری از حملات است.
- RSASSA-PKCS1-v1_5: از padding قطعی استفاده می کند.
- RSASSA-PSS: یک طرح لایه‌بندی احتمالی که تضمین‌های امنیتی قوی‌تری ارائه می‌دهد.

قسمت چهارم: امضاهای مبتنی بر RSA

امنیت امضاهای مبتنی بر RSA

امنیت امضاهای مبتنی بر RSA به موارد زیر بستگی دارد:

سختی فاکتورینگ:

شکستن RSA مستلزم فاکتورگیری $N=p\cdot q$ است، که از نظر محاسباتی برای N بزرگ غیر ممکن است.

امنیت عملکرد هش:

یک تابع هش ایمن (به عنوان مثال، SHA-256) برای جلوگیری از جعل یا تصادم حیاتی است.

طرح های Padding:

Padding مناسب (مانند PKCS #1) برای خنثی کردن حملات عملی مانند حمله Bleichenbacher ضروری است.

قسمت چهارم: امضاهای مبتنی بر RSA

مزایا

پشتیبانی گسترده:

RSAیک الگوریتم بالغ و به خوبی مطالعه شده با پذیرش گسترده است.

تایید کارآمد:

راستی‌آزمایی امضا سریع‌تر از تولید است، و آن را برای برنامه‌هایی مناسب می‌سازد که تأیید مکررتر است.

معایب

امضای آهسته:

تولید امضا در مقایسه با طرح‌های دیگر مانند ECDSA از نظر محاسباتی گران است.

اندازه کلید:

RSA به اندازه‌های کلید بزرگ‌تری (مثلاً ۲۰۴۸ یا ۳۰۷۲ بیت) نیاز دارد تا به همان سطح امنیتی طرح‌های مبتنی بر منحنی بیضی دست یابد که منجر به هزینه‌های ذخیره‌سازی و انتقال بالاتر می‌شود.

قسمت چهارم: امضاهای مبتنی بر RSA

برنامه های کاربردی

گواهینامه های TLS/SSL:

امضاهای مبتنی بر RSA معمولاً برای تأیید اعتبار گواهی‌های دیجیتال استفاده می‌شوند.

امضای کد:

یکپارچگی و اصالت بسته های نرم افزاری را تضمین می کند.

رمزگذاری ایمیل:

در پروتکل هایی مانند S/MIME برای برقراری ارتباط امن ایمیل استفاده می شود.

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

این بخش مفهوم طرح های شناسایی(**identification schemes**) و ارتباط آنها با امضای دیجیتال را معرفی می کند.

طرح های شناسایی:

این‌ها پروتکل‌های تعاملی هستند که در آن یک اثبات‌کننده تأییدکننده هویت خود را بدون فاش کردن کلید خصوصی خود متقاعد می‌کند.

بسیاری از طرح‌های امضای دیجیتال را می‌توان با اعمال تبدیل Fiat-Shamir از طرح‌های شناسایی استخراج کرد، که پروتکل‌های تعاملی را به پروتکل‌های غیر تعاملی تبدیل می‌کند.

ارتباط با امضای دیجیتال:

یک طرح امضا را می توان به عنوان یک نسخه "غیر تعاملی" از یک طرح شناسایی دید، که در آن امضا به عنوان مدرک هویت عمل می کند.

قسمت پنجم: طرح های شناسایی و امضاها
(Identification Schemes and Signatures)

طرح های شناسایی/امضای Schnorr

Schnorr Signature Schemeیکی از کارآمدترین و پرکاربردترین طرح های امضا بر اساس DLP است.

1. **Setup**:
   - Choose a large prime p and a generator g of a subgroup $\mathbb{Z}_p^*$ of order q.
   - The private key x is a random number in $[1, q-1]$.
   - The public key is $y = g^x \pmod{p}$.
2. **Signing**:
   - Generate a random $k \in [1, q-1]$ and compute $r = g^k \pmod{p}$.
   - Compute the hash $e = H(m \| r)$, where m is the message and r is the commitment.
   - Compute the signature $s = k - x \cdot e \pmod{q}$.
   - The signature is (r, s).
3. **Verification**:
   - Compute $v_1 = g^s \cdot y^e \pmod{p}$, where $e = H(m \| r)$.
   - Verify if $v_1 = r$. If true, the signature is valid.

قسمت پنجم: طرح های شناسایی و امضاها
(Identification Schemes and Signatures)

طرح های شناسایی/امضای Schnorr

Schnorr Signature Schemeیکی از کارآمدترین و پرکاربردترین طرح های امضا بر اساس DLP است.

1. **Setup**:
   - Choose a large prime p and a generator g of a subgroup $\mathbb{Z}_p^*$ of order q.
   - The private key x is a random number in $[1, q-1]$.
   - The public key is $y = g^x \pmod{p}$.
2. **Signing**:
   - Generate a random $k \in [1, q-1]$ and compute $r = g^k \pmod{p}$.
   - Compute the hash $e = H(m \| r)$, where m is the message and r is the commitment.
   - Compute the signature $s = k - x \cdot e \pmod{q}$.
   - The signature is (r, s).
3. **Verification**:
   - Compute $v_1 = g^s \cdot y^e \pmod{p}$, where $e = H(m \| r)$.
   - Verify if $v_1 = r$. If true, the signature is valid.

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

مزایا:

- کارآمد از نظر محاسبات و ارتباطات.
- امنیت قابل اثبات تحت فرض لگاریتم گسسته.

برنامه های کاربردی:

- ارزهای رمزنگاری شده (به عنوان مثال، امضاهای Schnorr برای بهبود بیت کوین پیشنهاد شده است).
- پروتکل های رمزنگاری سبک

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

DSA (الگوریتم امضای دیجیتال):

یک استاندارد دولتی ایالات متحده برای امضای دیجیتال بر اساس DLP.

مشابه امضاهای Schnorr است اما از رویکرد کمی متفاوت برای امضا و تأیید استفاده می کند.

- Steps:
  - Key Generation: Similar to Schnorr.
  - Signing: Compute $r = (g^k \pmod p) \pmod q$, $s = k^{-1}(H(m) + x \cdot r) \pmod q$.
  - Verification: Check if $r = ((g^s \cdot y^r) \pmod p) \pmod q$.

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

ECDSA (الگوریتم امضای دیجیتال منحنی بیضوی):

گونه ای از DSA که از گروه های منحنی بیضوی به جای میدان های محدود استفاده می کند.

مزایا نسبت به DSA:

اندازه های کلید کوچکتر برای امنیت معادل.

محاسبات سریع تر، به ویژه برای تأیید.

برنامه های کاربردی:

ECDSA به طور گسترده در سیستم های مدرن مانند TLS، SSL و پلتفرم های بلاک چین (به عنوان مثال، بیت کوین، اتریوم) استفاده می شود.

قسمت پنجم: طرح های شناسایی و امضاها
(Identification Schemes and Signatures)

Comparison of Signature Schemes

Scheme	Key Size	Efficiency	Security Assumption
RSA Signatures	Large	Slower (large keys)	Factoring large integers
Schnorr Signatures	Moderate	Efficient	Discrete logarithm problem
DSA	Moderate	Moderate	Discrete logarithm problem
ECDSA	Small	Highly Efficient	Elliptic curve discrete log prob

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

مزایای امضاهای مبتنی بر DLP

کارایی:

اندازه های کلید کوچکتر در مقایسه با RSA.

محاسبات سریعتر برای سطوح امنیتی معادل.

امنیت:

بر اساس مسائل به خوبی مطالعه شده مانند DLP و ECDLP (مسئله لگاریتم گسسته منحنی بیضوی).

انعطاف پذیری:

می تواند با تنظیمات رمزنگاری مختلف (به عنوان مثال، منحنی های بیضوی) سازگار شود.

قسمت پنجم: طرح های شناسایی و امضاها (Identification Schemes and Signatures)

معایب

انتخاب پارامتر:

به انتخاب دقیق پارامترها (به عنوان مثال p,q,g) برای جلوگیری از آسیب‌پذیری نیاز دارد.

حملات تخصصی:

در صورت استفاده از پارامترهای نامناسب (مثلاً حملات زیرگروهی کوچک) در برابر حملات خاص آسیب پذیر است.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

این بخش نقش گواهی‌ها و زیرساخت‌های کلید عمومی (PKI) را در تضمین اعتماد و اصالت در ارتباطات دیجیتال مورد بحث قرار می‌دهد. این مؤلفه‌ها برای تأیید کلیدهای عمومی و جلوگیری از جعل هویت بازیگران مخرب بسیار مهم هستند.

گواهی (Certificate) چیست؟

گواهی دیجیتال یک سند الکترونیکی است که برای اثبات مالکیت کلید عمومی استفاده می شود. این توسط یک شخص ثالث قابل اعتماد به نام مرجع صدور گواهی (CA) صادر می شود و یک کلید عمومی را به یک فرد، سازمان یا نهاد متصل می کند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

ساختار یک گواهی:

موضوع (Subject): نهاد (شخص، سازمان یا سرور) که گواهی برای آن صادر شده است.

کلید عمومی(Public Key): کلید عمومی موضوع.

صادرکننده(Issuer): مرجع صدور گواهی که گواهی را صادر کرده است.

دوره اعتبار(Validity Period): تاریخ شروع و پایانی است که طی آن گواهی معتبر است.

امضا(Signature): امضای دیجیتال CA، که از صحت گواهی اطمینان می دهد.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

زیرساخت کلید عمومی (PKI) چیست؟

زیرساخت کلید عمومی (PKI) چارچوبی است که ایجاد، توزیع و اعتبارسنجی گواهی‌های دیجیتال را مدیریت می‌کند. این یک محیط قابل اعتماد برای ارتباط امن ایجاد می کند.

اجزای PKI:

۱. مرجع صدور گواهی (CA-Certificate Authority):

یک نهاد مورد اعتماد که گواهینامه ها را صادر و باطل می کند.

مثال: DigiCert، Let's Encrypt.

۲. سازمان ثبت (RA–Registration Authority):

هویت نهادهای درخواست کننده گواهی از طرف CA را تأیید می کند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

اجزای PKI:

۳. مخزن گواهی (Certificate Repository) :

یک پایگاه داده در دسترس عموم که در آن گواهی های صادر شده و لیست های ابطال ذخیره می شوند.

٤. فهرست ابطال گواهی (CRL-Certificate Revocation List):

فهرستی از گواهینامه هایی که قبل از تاریخ انقضا به دلیل مصالحه یا مسائل دیگر باطل شده اند.

۵. موجودیت های پایانی(End Entities):

کاربران، دستگاه‌ها یا سرورهایی که از گواهی‌ها برای احراز هویت و رمزگذاری ارتباطات استفاده می کنند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

نحوه کار گواهینامه ها در امضای دیجیتال

گواهینامه ها در تأیید صحت کلیدهای عمومی در حین تأیید امضای دیجیتال بسیار مهم هستند:

سمت فرستنده:

- فرستنده با استفاده از کلید خصوصی خود پیامی را امضا می کند.
- گواهی آنها، حاوی کلید عمومی آنها، به پیام پیوست شده است.

سمت گیرنده:

- گیرنده از گواهی فرستنده برای دریافت کلید عمومی استفاده می کند.
- آنها امضا را با استفاده از کلید عمومی تأیید می کنند.
- آنها همچنین صحت گواهی را با بررسی امضای CA تأیید می کنند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

انواع گواهینامه ها

اعتبار سنجی دامنه (Domain Validation):

- مالکیت دامنه را تأیید می کند.
- برای ایمن سازی وب سایت ها با HTTPS استفاده می شود.

اعتبار سنجی سازمان (Organization Validation):

- مشروعیت یک سازمان را تأیید می کند.
- اعتماد بالاتری نسبت به اعتبارسنجی دامنه ارائه می دهد.

اعتبار سنجی توسعه یافته (Extended Validation):

- شامل بررسی های دقیق برای تأیید هویت سازمان است.
- یک نوار آدرس سبز رنگ در مرورگرهای وب برای اعتماد بالا نمایش می دهد.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

چالش های PKI

مدیریت گواهی (Certificate Management)

اطمینان از تمدید و ابطال به موقع گواهی ها می تواند چالش برانگیز باشد.

مدل اعتماد (Trust Model)

کل سیستم به اعتماد مقامات صدور گواهی بستگی دارد. اگر یک CA به خطر بیفتد، می تواند منجر به مشکلات گسترده شود.

مقیاس پذیری (Scalability)

مدیریت تعداد زیادی گواهینامه در یک شبکه جهانی می تواند پیچیده باشد.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

مزایای گواهینامه ها و PKI

اعتماد (Trust):

یک کانال ارتباطی امن و قابل اعتماد ایجاد می کند.

احراز هویت (Authentication):

هویت نهادها و اشخاص را تأیید می کند.

یکپارچگی داده ها (Data Integrity):

تضمین می کند که پیام ها در حین انتقال دستکاری نمی شوند.

محرمانه بودن(Confidentiality):

رمزگذاری ایمن داده ها را فعال می کند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

کاربردهای PKI

TLS/SSL :

ترافیک وب را با رمزگذاری ارتباط بین مشتریان و سرورها ایمن می کند.

مثال: وب سایت های .HTTPS

امنیت ایمیل:

ارتباط امن ایمیل را با استفاده از پروتکل هایی مانند S/MIME تضمین می کند.

امضای کد:

اصالت و یکپارچگی نرم افزار را تایید می کند.

امنیت اینترنت اشیا:

از ارتباطات بین دستگاه های اینترنت اشیا محافظت می کند.

قسمت ششم: گواهینامه ها و زیرساخت های کلید عمومی (Certificates and Public-Key Infrastructures)

نتیجه گیری

گواهی ها و PKI اجزای اساسی در رمزنگاری مدرن هستند. آنها یک رویکرد ساختاریافته برای مدیریت کلیدهای عمومی و اطمینان از اعتماد در ارتباطات دیجیتال ارائه می دهند. بدون PKI، تراکنش‌های آنلاین امن، ایمیل‌های رمزگذاری‌شده، و سایر کاربردهای امضای دیجیتال امکان‌پذیر نخواهد بود.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

این بخش مفاهیم امضای دیجیتال و رمزنگاری کلید عمومی را با TLS (امنیت لایه حمل و نقل)، پروتکلی که ارتباط را از طریق اینترنت ایمن می‌کند، پیوند می‌دهد. توضیح می دهد که چگونه TLS از امضای دیجیتال برای ایجاد اعتماد و اطمینان از محرمانه بودن، یکپارچگی و احراز هویت استفاده می کند.

TLS چیست؟

TLS (Transport Layer Security)یک پروتکل رمزنگاری است که ارتباط ایمن را از طریق یک شبکه، معمولاً بین یک کلاینت (به عنوان مثال، یک مرورگر وب) و یک سرور (به عنوان مثال، یک وب سایت) فراهم می کند. این جانشین SSL (لایه سوکت های امن) است و به طور گسترده برای ایمن سازی ترافیک وب، ایمیل و سایر خدمات آنلاین استفاده می شود.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

این بخش مفاهیم امضای دیجیتال و رمزنگاری کلید عمومی را با TLS (امنیت لایه حمل و نقل)، پروتکلی که ارتباط را از طریق اینترنت ایمن می‌کند، پیوند می‌دهد. توضیح می دهد که چگونه TLS از امضای دیجیتال برای ایجاد اعتماد و اطمینان از محرمانه بودن، یکپارچگی و احراز هویت استفاده می کند.

TLS چیست؟

TLS (Transport Layer Security)یک پروتکل رمزنگاری است که ارتباط ایمن را از طریق یک شبکه، معمولاً بین یک کلاینت (به عنوان مثال، یک مرورگر وب) و یک سرور (به عنوان مثال، یک وب سایت) فراهم می کند. این جانشین SSL (لایه سوکت های امن) است و به طور گسترده برای ایمن سازی ترافیک وب، ایمیل و سایر خدمات آنلاین استفاده می شود.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

اهداف TLS

محرمانه بودن (Confidentiality)

اطمینان حاصل می کند که داده های مبادله شده بین طرفین رمزگذاری شده است و نمی تواند توسط نهادهای غیرمجاز رهگیری یا خوانده شود.

صداقت (Integrity)

اطمینان حاصل می کند که داده ها در طول انتقال تغییر نمی کنند.

احراز هویت (Authentication)

هویت طرف‌های ارتباطی را تأیید می‌کند، معمولاً با استفاده از گواهی‌ها و امضاهای دیجیتال.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

فرآیند TLS Handshake

TLS Handshakeفرآیندی است که از طریق آن مشتری و سرور یک کانال ارتباطی امن ایجاد می کنند. امضای دیجیتال نقش مهمی در این فرآیند ایفا می کند.

۱. Client Hello:

مشتری پیامی به سرور ارسال می کند که نشان می دهد می خواهد یک اتصال امن برقرار کند.

این پیام شامل الگوریتم های رمزگذاری پشتیبانی شده و یک مقدار تولید شده به صورت تصادفی است.

۲. Server Hello:

سرور با الگوریتم های رمزگذاری پشتیبانی شده و مقدار تصادفی خود پاسخ می دهد.

سرور همچنین گواهی دیجیتال خود را ارسال می کند که حاوی کلید عمومی آن است و توسط یک مرجع گواهی معتبر (CA) امضا شده است.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

۳. تایید گواهی (Certificate Verification)

کلاینت گواهی سرور را با استفاده از کلید عمومی CA تأیید می کند.

اگر گواهی معتبر باشد، کلاینت به کلید عمومی سرور اعتماد دارد.

٤. تبادل کلید (Key Exchange)

کلاینت و سرور با استفاده از یک الگوریتم تبادل کلید (به عنوان مثال، Diffie-Hellmanیا RSA) روی یک کلید مخفی مشترک توافق می کنند.

سرور ممکن است به صورت دیجیتالی قسمت هایی از Handshake را برای اثبات هویت خود امضا کند.

٥. تولید کلید جلسه (Session Key Generation)

هر دو طرف یک کلید session را از shared secret و مقادیر تصادفی که قبلاً رد و بدل شده بودند استخراج می کنند.

٦. ارتباط امن (Secure Communication)

کلید جلسه برای رمزگذاری و رمزگشایی تمام ارتباطات بعدی استفاده می شود.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

نقش امضای دیجیتال در TLS

احراز هویت سرور(Server Authentication)

گواهی سرور حاوی کلید عمومی آن است که توسط یک CA امضا شده است.

مشتری برای اطمینان از صحت سرور، امضای CA را تأیید می کند.

صداقت (Integrity)

امضای دیجیتالی برای اطمینان از عدم دستکاری پیام های دست دادن در حین انتقال استفاده می شود.

احراز هویت مشتری (اختیاری) – Client Authentication (Optional)

در برخی موارد، کلاینت گواهینامه ای برای احراز هویت خود به سرور ارائه می دهد.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

TLS Cipher Suites

مجموعه رمزگذاری ترکیبی از الگوریتم‌های رمزنگاری است که در هنگام Handshake و ارتباط TLS استفاده می‌شود. معمولاً شامل:

الگوریتم تبادل کلید(Key Exchange Algorithm):

نحوه ایجاد shared secret را تعیین می کند (به عنوان مثال، RSA، ECDHE).

الگوریتم امضای دیجیتال(Digital Signature Algorithm):

صحت پیام های Handshake (مانند RSA، ECDSA) را تأیید می کند.

الگوریتم رمزگذاری(Encryption Algorithm):

ارتباطات را رمزگذاری می کند (به عنوان مثال، AES).

عملکرد هش(Hash Function):

یکپارچگی داده ها را تضمین می کند (به عنوان مثال، SHA-256).

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

TLS در عمل

وبگردی (Web Browsing)

هنگامی که از یک وب سایت HTTPS بازدید می کنید، TLS تضمین می کند که اتصال شما ایمن است.

مرورگر یک نماد قفل را نمایش می دهد تا اتصال ایمن را نشان دهد.

امنیت ایمیل (Email Security)

TLS ایمیل های ارسال شده بین سرورهای ایمیل را رمزگذاری می کند و از محرمانه بودن اطمینان حاصل می کند.

بانکداری آنلاین (Online Banking)

TLS از اطلاعات حساس مانند رمز عبور و جزئیات کارت اعتباری در طول تراکنش های آنلاین محافظت می کند.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

مزایای استفاده از TLS

امنیت سرتاسری (End-to-End Security)

ارتباط بین کلاینت و سرور را رمزگذاری می کند و در برابر استراق سمع محافظت می کند.

اعتماد(Trust)

هویت سرور را با استفاده از گواهی های دیجیتال تأیید می کند.

یکپارچگی داده ها(Data Integrity)

اطمینان حاصل می کند که داده ها در طول انتقال تغییر نمی کنند.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

چالش ها

مدیریت گواهی(Certificate Management)

سوء مدیریت گواهی ها می تواند منجر به آسیب پذیری های امنیتی شود.

حملات Man-in-the-Middle

اگر مهاجم بتواند جعل هویت یک CA قابل اعتماد داشته باشد، می تواند ارتباطات ایمن را رهگیری کند.

سازگاری به عقب (Backward Compatibility)

نسخه های قدیمی تر TLS (به عنوان مثال، TLS 1.0، ۱.۱) آسیب پذیری های شناخته شده ای دارند و نباید از آنها استفاده شود.

قسمت هفتم: قرار دادن همه چیز در کنار هم – TLS

نتیجه گیری

TLS نشان می‌دهد که چگونه امضای دیجیتال، رمزنگاری کلید عمومی و گواهی‌ها برای ایمن کردن ارتباطات از طریق اینترنت با هم کار می‌کنند. این یک کاربرد واقعی از اصول رمزنگاری است که تضمین می کند میلیاردها تراکنش آنلاین ایمن و قابل اعتماد هستند.

قسمت هشتم: Signcryption

این بخش مفهوم رمزگذاری را معرفی می کند، یک تکنیک رمزنگاری که عملکردهای امضای دیجیتال و رمزگذاری را در یک عملیات واحد ترکیب می کند. این برای دستیابی به محرمانه بودن و اصالت به طور همزمان طراحی شده است و مزایایی را از نظر کارایی و امنیت ارائه می دهد.

Signcryption چیست؟

Signcryptionیک رمزنگاری اولیه است که فعالیت های زیر را انجام می دهد:

رمزگذاری(Encryption): محرمانه بودن پیام را تضمین می کند.

امضای دیجیتال(Digital Signature): احراز هویت و صحت پیام را ارائه می دهد.

بر خلاف رویکردهای سنتی که در آن رمزگذاری و امضا به صورت متوالی انجام می شود (رمزگذاری-سپس-امضا یا امضا-سپس-رمزگذاری)، رمزگذاری Signcryption هر دو عملیات را در یک مرحله واحد ترکیب می کند.

قسمت هشتم: Signcryption

چرا از Signcryption استفاده کنیم؟

کارایی (Efficiency)

رمزگذاری از نظر محاسباتی کارآمدتر از انجام رمزگذاری و امضای جداگانه است. هزینه محاسباتی کلی و اندازه متن رمز را کاهش می دهد.

امنیت(Security)

با ترکیب رمزگذاری و امضا، رمزگذاری از آسیب‌پذیری‌های خاصی که در فرآیندهای رمزگذاری و امضای جداگانه ایجاد می‌شوند، جلوگیری می‌کند.

به عنوان مثال، در یک رویکرد سنتی رمزگذاری و سپس علامت، متن رمزی (ciphertext) امضا شده ممکن است اطلاعاتی را درباره متن ساده (plaintext) نشان دهد.

کاربردهای عملی (Practical Applications)

این به ویژه در محیط های محدود به منابع (به عنوان مثال، دستگاه های IoT) که در آن کارایی محاسباتی حیاتی است، مفید است.

قسمت هشتم: Signcryption

Signcryption چگونه کار می کند؟

طرح های رمزگذاری علامت معمولاً شامل مراحل زیر است:

سمت فرستنده:

- فرستنده رمزگذاری و امضا را در یک عملیات واحد ترکیب می کند.
- پیام با استفاده از کلید عمومی گیرنده رمزگذاری می شود.
- به طور همزمان، کلید خصوصی فرستنده برای امضای پیام استفاده می شود.

سمت گیرنده:

- گیرنده پیام را با استفاده از کلید خصوصی خود رمزگشایی می کند.
- گیرنده امضا را با استفاده از کلید عمومی فرستنده تأیید می کند.

قسمت هشتم: Signcryption

Comparison: Signcryption vs. Traditional Methods

Feature	Traditional Methods (Encrypt-then-Sign or Sign-then-Encrypt)	Signcryption
Efficiency	Higher computational and bandwidth costs	Lower costs
Message Size	Larger due to separate ciphertext and signature	Smaller
Security	Potential vulnerabilities in combining encryption and signing	Stronger due to unified operation

قسمت هشتم: Signcryption

طرح های Signcryption

رمزگذاری با کلید عمومی:

بر اساس رمزنگاری کلید عمومی.

مثال: رمزگذاری منحنی بیضی (ECS).

رمزگذاری با کلید متقارن:

از یک کلید مخفی مشترک بین فرستنده و گیرنده استفاده می کند.

کمتر از رمزگذاری کلید عمومی رایج است.

قسمت هشتم: Signcryption

کاربردهای Signcryption

پیام رسانی امن:

محرمانه بودن و اعتبار را در سیستم های ایمیل و پیام فوری تضمین می کند.

تجارت الکترونیک:

از اطلاعات حساس مانند جزئیات کارت اعتباری محافظت می کند و در عین حال از صحت تراکنش اطمینان می دهد.

دستگاه های اینترنت اشیا:

دستگاه‌های محدود به منابع می‌توانند از کارایی رمزگذاری علامت برای ارتباطات ایمن بهره ببرند.

بلاک چین:

برای اطمینان از یکپارچگی و محرمانه بودن داده ها در معاملات ارزهای دیجیتال استفاده می شود.

قسمت هشتم: Signcryption

مزایای Signcryption

عملکرد:

سربار محاسباتی و استفاده از پهنای باند را کاهش می دهد.

امنیت پیشرفته:

آسیب پذیری های مرتبط با رمزگذاری و امضای جداگانه را از بین می برد.

خروجی فشرده:

در مقایسه با روش‌های سنتی، متن رمزی کوچک‌تری تولید می‌کند.

قسمت هشتم: Signcryption

چالش های Signcryption

استاندارد سازی:

پروتکل های استاندارد کمتر در مقایسه با رمزگذاری و امضای سنتی.

مدیریت کلید:

نیاز به مدیریت دقیق کلیدهای عمومی و خصوصی، مشابه دیگر طرح های رمزنگاری، دارد.

پیچیدگی:

طراحی و اجرای طرح‌های Signcryption می‌تواند پیچیده‌تر باشد.

قسمت هشتم: Signcryption

نتیجه گیری

Signcryptionیک تکنیک رمزنگاری ابتکاری است که محدودیت‌های روش‌های رمزگذاری و امضای سنتی را برطرف می‌کند. با ترکیب این دو عملیات در یک عملیات، هم کارایی و هم امنیت افزایش یافته است و آن را به ابزاری قدرتمند برای سیستم های ارتباطی ایمن مدرن تبدیل می کند.