4 JWS 的签名方式

复习一下前文中 JWS 的知识：

JWS 由三部分组成：Header + Payload + Signature，各部分以 . 分隔，结构如下：

Header.Payload.Signature

重点关注 Signature 部分，它是 JWS 保证安全的关键，

Signature，即签名，它由编码后的header和payload，使用用户指定的密钥(JWK)，采用header中指定的签名算法（JWA）生成。

Signature 可用于验证数据是否篡改（保证数据完整性）。

这其中，签名算法起到了关键作用。目前，JWS 签名算法有三种：

HMAC【哈希消息验证码(对称)】（HS256/HS384/HS512）

RSASSA【RSA签名算法(非对称)】（RS256/RS384/RS512）

ECDSA【椭圆曲线数据签名算法(非对称)】（ES256/ES384/ES512）

HMAC

Hmac算法就是一种基于密钥的消息认证码算法，它的全称是Hash-based Message Authentication Code，是一种更安全的消息摘要算法。

传统的Hash算法只能验证数据的完整性，但是无法保证数据防篡改。也就是说，攻击者可以将你的原文改了，并把hash值也改了，你没法确定数据是否被篡改。

Hmac算法在Hash算法基础增加了密钥，通过密钥+原文生成mac值，攻击者就算改了你的原文，但是他没有你的密钥就生成不了对应的mac值，这样，你在得到报文时就可以验证出是否被篡改。

SHA系列算法：HmacSHA256、HmacSHA384、HmacSHA512

RSASSA

RSA本身是一种非对称加密算法，在加解密上，使用公钥加密，私钥解密。

RSASSA 是签名算法，计算哈希是签名生成操作的一部分。使用私钥加签，公钥验签。

SHA系列算法：SHA256withRSA、SHA384withRSA、SHA512withRSA

在入门案例中，我们正是使用此签名算法，具体方法为generateRsaKey()。授权服务器使用私钥对 jws 加签，通过 /oauth2/jwks 接口暴露公钥信息给资源服务器，资源服务器可以对收到的 jws 验签，验证jws的合法性。

其他

签名可以使用对称签名（Symmetric Signatures）、不对称签名（Asymmetric Signatures），对应 对称加密算法、非对称加密算法

实现方式可通过消息认证码（Message Authentication Code, MAC） 或 数字签名（Digital Signature）

end