1. SSL 与 TLS

SSL：（Secure Socket Layer） 安全套接层，于 1994 年由网景公司设计，并于 1995 年发布了 3.0 版本
TLS：（Transport Layer Security）传输层安全性协议，是 IETF 在 SSL3.0 的基础上设计的协议
以下全部使用 TLS 来表示

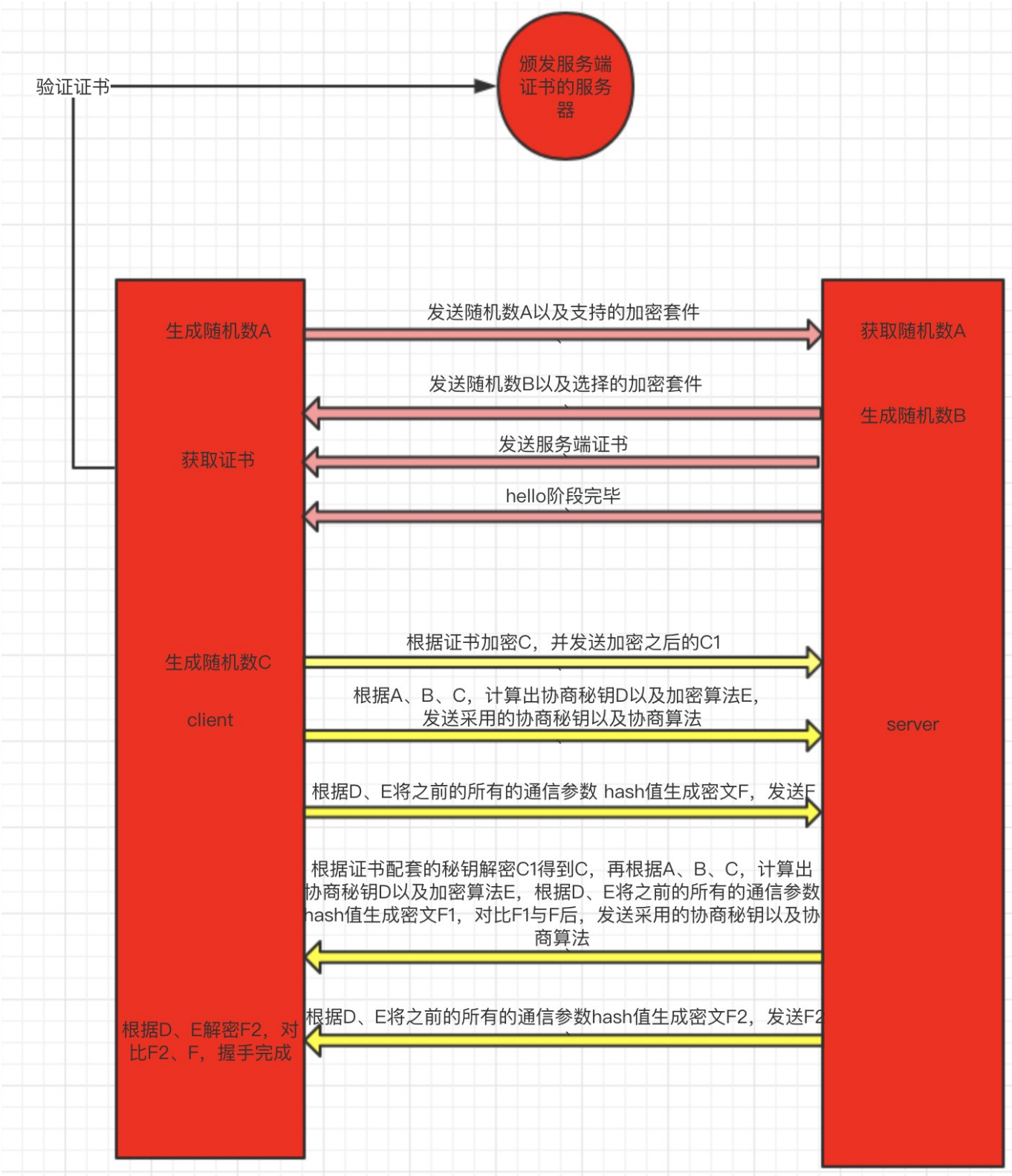
2. 从网络协议的角度理解 HTTPS

HTTP：HyperText Transfer Protocol 超文本传输协议
HTTPS：Hypertext Transfer Protocol Secure 超文本传输安全协议
TLS：位于 HTTP 和 TCP 之间的协议，其内部有 TLS握手协议、TLS记录协议
HTTPS 经由 HTTP 进行通信，但利用 TLS 来保证安全，即 HTTPS = HTTP + TLS

3. 从密码学的角度理解 HTTPS

HTTPS 使用 TLS 保证安全，这里的"安全"分两部分，一是传输内容加密、二是服务端的身份认证

3.1. TLS 工作流程



此为服务端单向认证，还有客户端/服务端双向认证，流程类似，只不过客户端也有自己的证书，并发送给服务器进行验证

3.2. 密码基础

3.2.1. 伪随机数生成器

为什么叫伪随机数，因为没有真正意义上的随机数，具体可以参考 Random/TheadLocalRandom
它的主要作用在于生成对称密码的秘钥、用于公钥密码生成秘钥对

3.2.2. 消息认证码

消息认证码主要用于验证消息的完整性与消息的认证，其中消息的认证指"消息来自正确的发送者"

> 消息认证码用于验证和认证，而不是加密

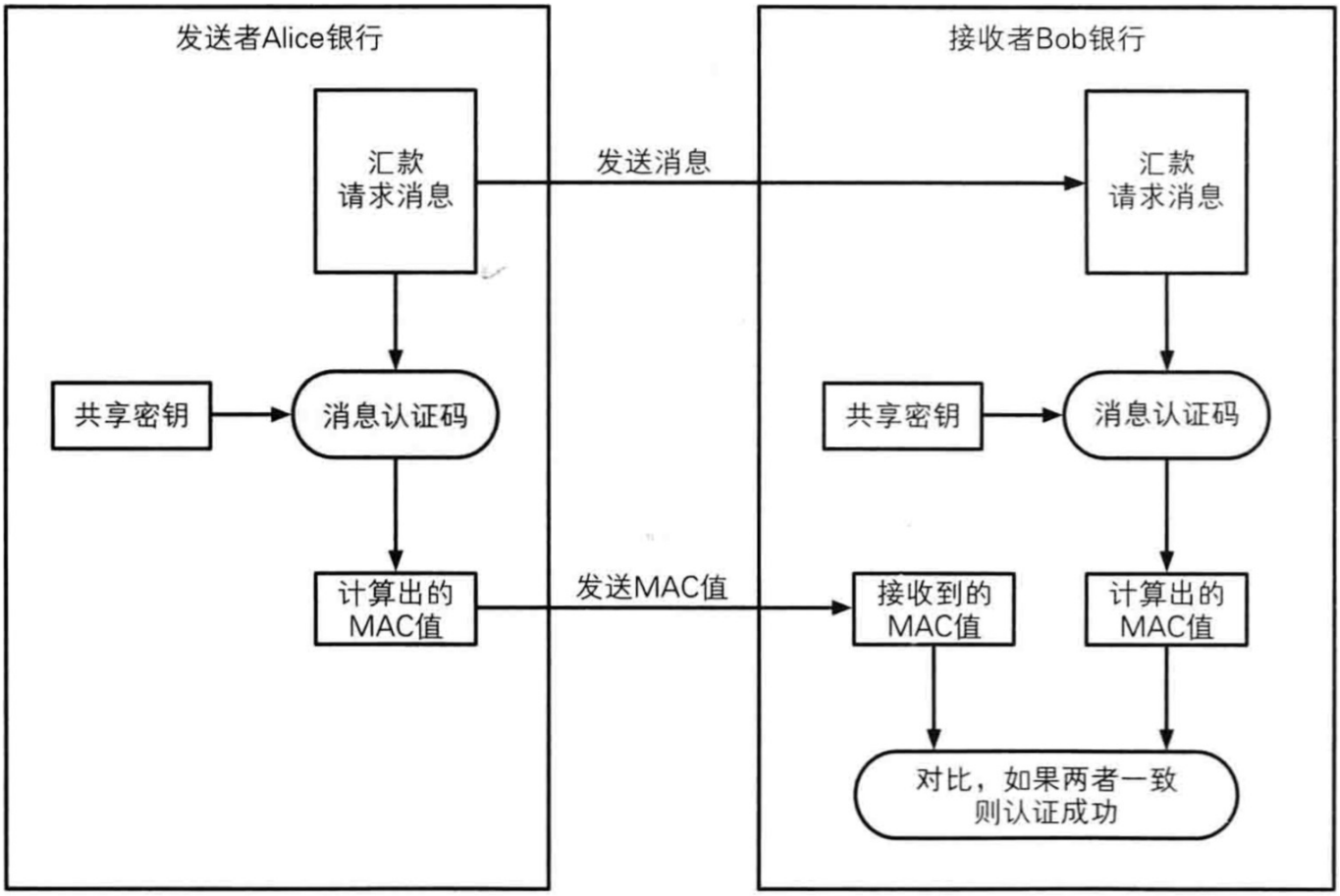


图 8-2 消息认证码的使用步骤

1. 发送者与接收者事先共享秘钥
2. 发送者根据发送消息计算 MAC 值
3. 发送者发送消息和 MAC 值
4. 接收者根据接收到的消息计算 MAC 值
5. 接收者根据自己计算的 MAC 值与收到的 MAC 对比
6. 如果对比成功，说明消息完整，并来自于正确的发送者

3.2.3. 数字签名

消息认证码的缺点在于**无法防止否认**，因为共享秘钥被 client、server 两端拥有，server 可以伪造 client 发送给自己的消息（自己给自己发送消息），为了解决这个问题，我们需要它们有各自的秘钥不被第二个知晓（这样也解决了共享秘钥的配送问题）

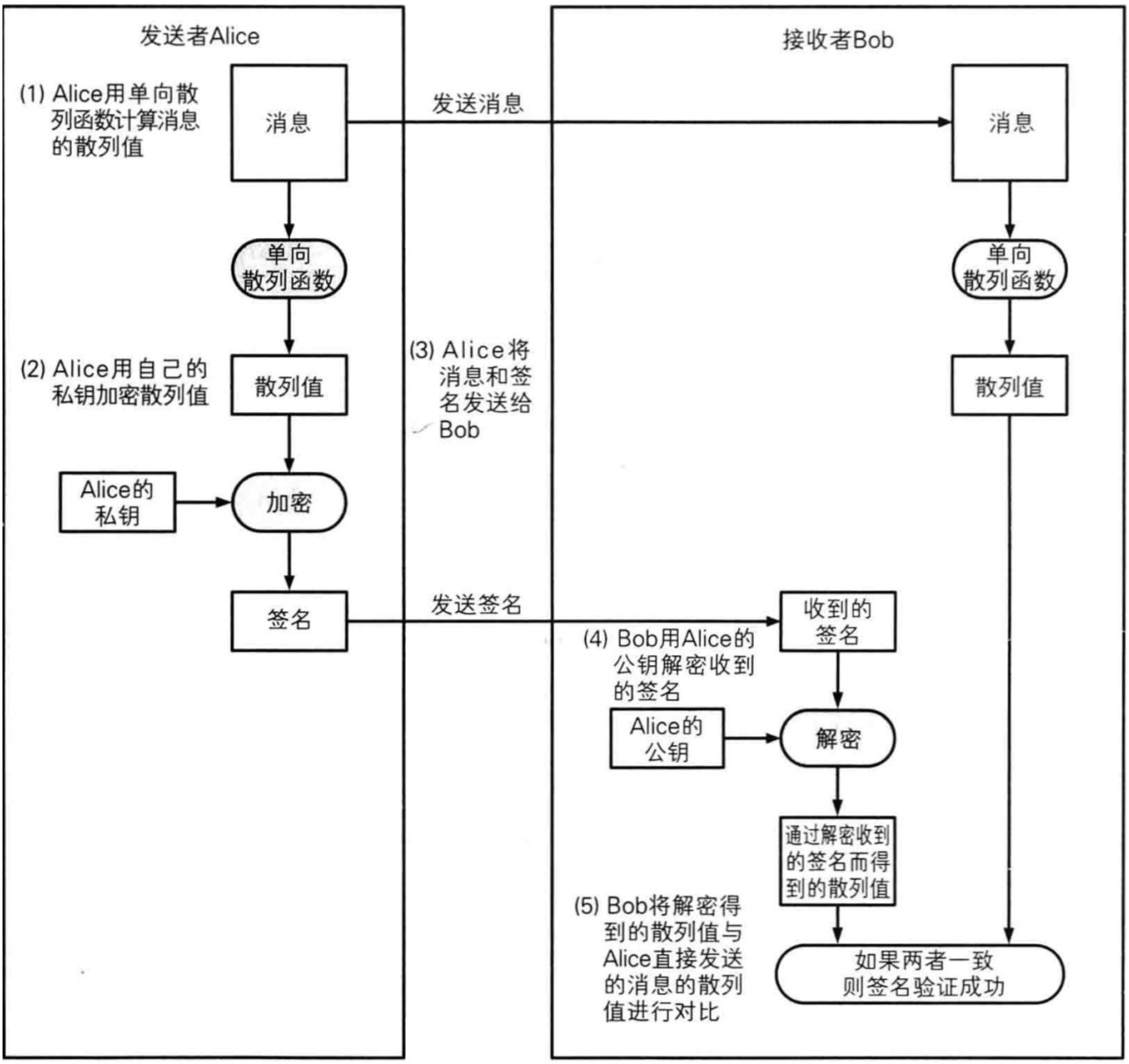


图 9-6 Alice 对消息的散列值签名，Bob 验证签名

> 数字签名和消息认证码都**不是为了加密**
> 可以将单向散列函数获取散列值的过程理解为使用 md5 摘要算法获取摘要的过程

使用自己的私钥对自己所认可的消息生成一个该消息专属的签名，这就是数字签名，表明我承认该消息来自自己

注意：**私钥用于加签，公钥用于解签，每个人都可以解签，查看消息的归属人**

3.2.4. 公钥密码

公钥密码也叫非对称密码，由公钥和私钥组成，它最开始是为了解决秘钥的配送传输安全问题，即，我们不配送私钥，只配送公钥，私钥由本人保管

它与数字签名相反，公钥密码的私钥用于解密、公钥用于加密，每个人都可以用别人的公钥加密，但只有对应的私钥才能解开密文

client：明文 + 公钥 = 密文

server：密文 + 私钥 = 明文

注意：**公钥用于加密，私钥用于解密，只有私钥的归属者，才能查看消息的真正内容**

3.2.5. 证书

证书：全称公钥证书（Public-Key Certificate, PKC）,里面保存着归属者的基本信息，以及证书过期时间、归属者的公钥，并由认证机构（Certification Authority, **CA**）施加数字签名，表明，某个认证机构认定该公钥的确属于此人

> 想象这个场景：你想在支付宝页面交易，你需要支付宝的公钥进行加密通信，于是你从百度上搜索关键字“支付宝公钥”，你获得了支什宝的公钥，这个时候，支什宝通过中间人攻击，让你访问到了他们支什宝的页面，最后你在这个支什宝页面完美的使用了支什宝的公钥完成了与支什宝的交易

在上面的场景中，你可以理解支付宝证书就是由支付宝的公钥、和给支付宝颁发证书的企业的数字签名组成
任何人都可以给自己或别人的公钥添加自己的数字签名，表明：我拿我的尊严担保，我的公钥/别人的公钥是真的，至于信不信那是另一回事了

3.2.6. 密码小结

密码	作用	组成
消息认证码	确认消息的完整、并对消息的来源认证	共享秘钥+消息的散列值
数字签名	对消息的散列值签名	公钥+私钥+消息的散列值
公钥密码	解决秘钥的配送问题	公钥+私钥+消息
证书	解决公钥的归属问题	公钥密码中的公钥+数字签名

3.3. TLS 使用的密码技术

1. 伪随机数生成器：秘钥生成随机性，更难被猜测
2. 对称密码：对称密码使用的秘钥就是由伪随机数生成，相较于非对称密码，效率更高
3. 消息认证码：保证消息信息的完整性、以及验证消息信息的来源
4. 公钥密码：证书技术使用的就是公钥密码
5. 数字签名：验证证书的签名，确定由真实的某个 CA 颁发

6. 证书：解决公钥的真实归属问题，降低中间人攻击概率

3.4. TLS 总结

TLS 是一系列密码工具的框架，作为框架，它也是非常的灵活，体现在每个工具套件它都可以替换，即：客户端与服务端之间协商密码套件，从而更难的被攻破，例如使用不同方式的对称密码，或者公钥密码、数字签名生成方式、单向散列函数技术的替换等

4. RSA 简单示例

RSA 是一种公钥密码算法，我们简单的走一遍它的加密解密过程
加密算法：密文 = (明文^E) mod N，其中公钥为{E,N}，即"求明文的E次方的对 N 的余数"
解密算法：明文 = (密文^D) mod N，其中秘钥为{D,N}，即"求密文的D次方的对 N 的余数"
例：我们已知公钥为{5,323}，私钥为{29,323}，明文为300，请写出加密和解密的过程：

> 加密：密文 = 123 ^ 5 mod 323 = 225
> 解密：明文 = 225 ^ 29 mod 323 = [[(225 ^ 5) mod 323] * [(225 ^ 5) mod 323] * [(225 ^ 5) mod 323] * [(225 ^ 5) mod 323] * [(225 ^ 5) mod 323] * [(225 ^ 4) mod 323]] mod 323 = (4 * 4 * 4 * 4 * 4 * 290) mod 323 = 123

5. 参考

1. SSL加密发生在哪里：https://security.stackexchange.com/questions/19681/where-does-ssl-encryption-take-place
2. TLS工作流程：https://blog.csdn.net/ustccw/article/details/76691248
3. 《图解密码技术》：https://book.douban.com/subject/26822106/ 豆瓣评分 9.5