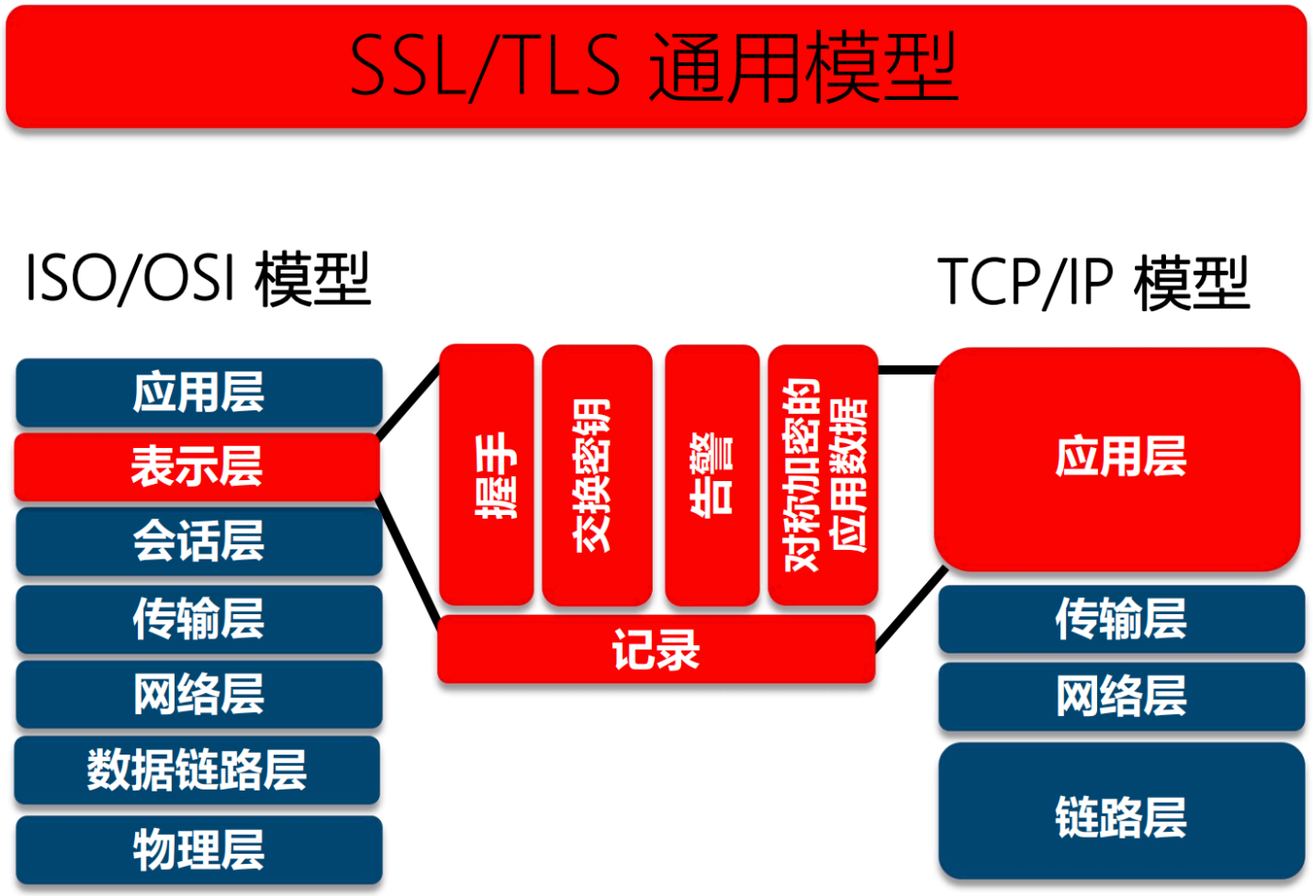
深入理解 HTTPS

TLS 协议的工作原理

TLS 发展

- 1995 SSL3.0
- 1999 TLS1.0
- 2006 TLS1.1
- 2008 TLS1.2
- 2018 TLS1.3



SSL(Secure Sockets Layer)
TLS(Transport Layer Security)

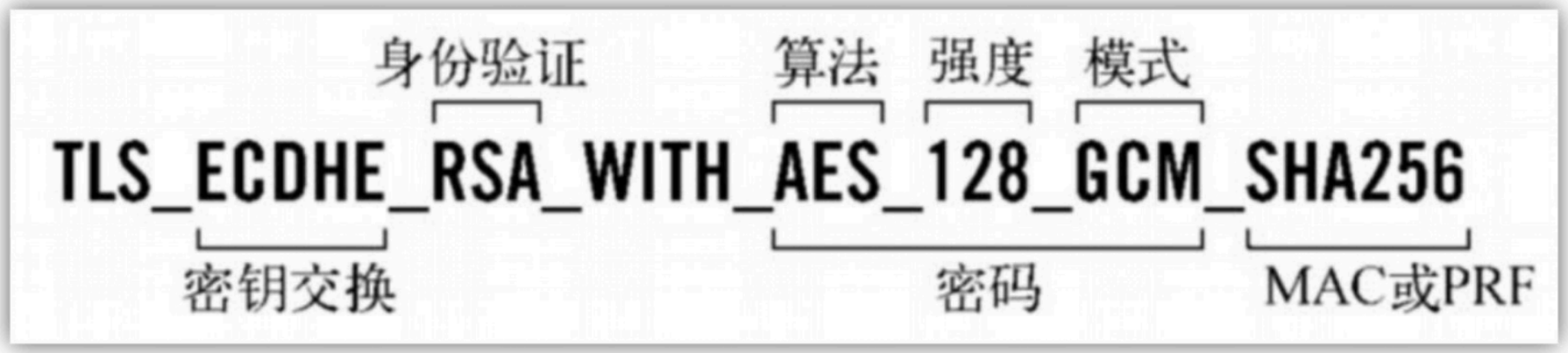
TLS 的设计目的

- 身份验证
- 保密性
- 完整性

TLS 协议

- **Record 记录协议**
  - 对称加密
- **HandShark 握手协议**
  - 验证通讯双方的身份
  - 交换加解密的安全套件
  - 协商加密参数

TLS 加密套件解读

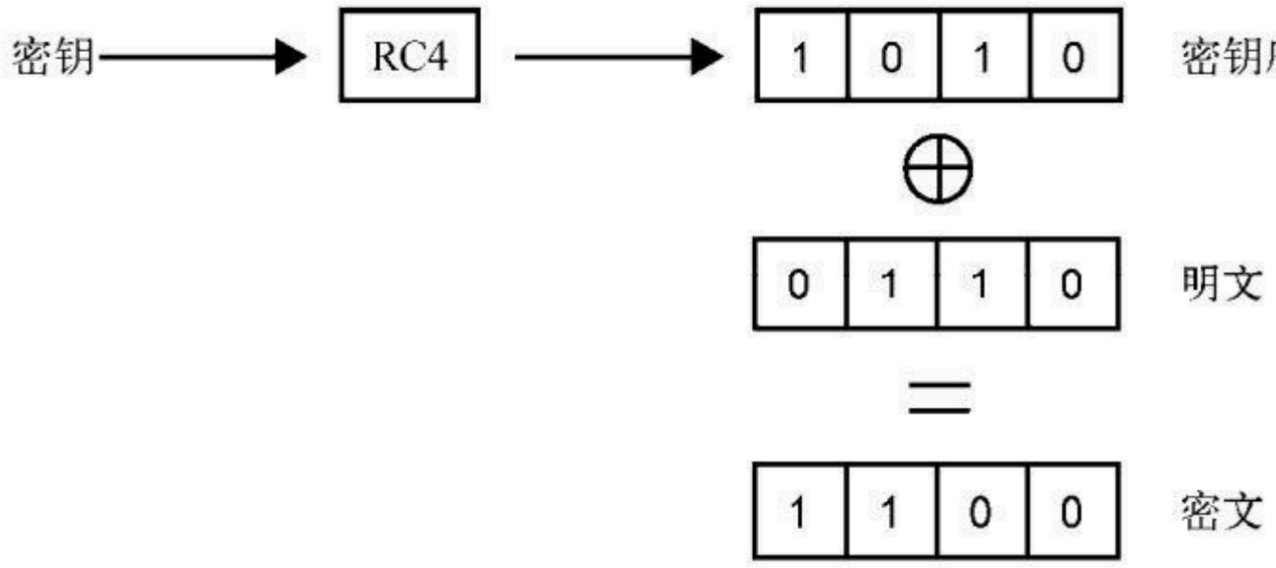

身份验证
算法
强度
模式
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
密钥交换
密码
MAC或PRF

对称加密的工作原理

对称加密

- 加密、解密使用同一个密钥

对称加密之异或运算



XOR Truth Table		
Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

对称加密之填充

- Block cipher 分组加密：将明文分为多个等长的 Block 块，对每个块分别加密
- 目的：当最后一个块不足时，需要填充
- 填充方法
  - 字节填充
    - PKCS#7：… | DD DD DD DD DD DD DD DD | DD DD DD DD 04 04 04 04

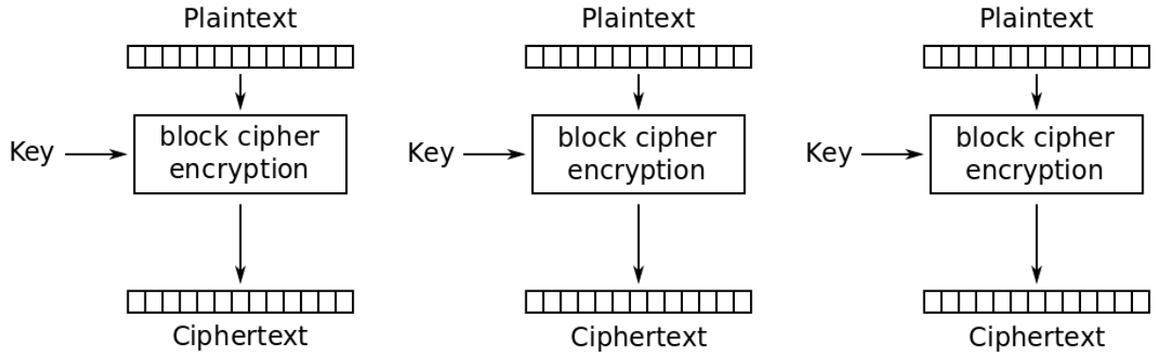
对称加密之分组工作模式

允许使用同一个分组密钥对多于一块的数据进行加密，并保障其安全性

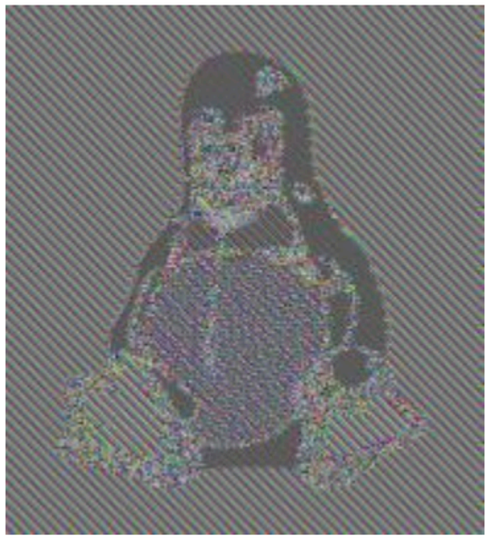
- ECB
- CBC
- CTR

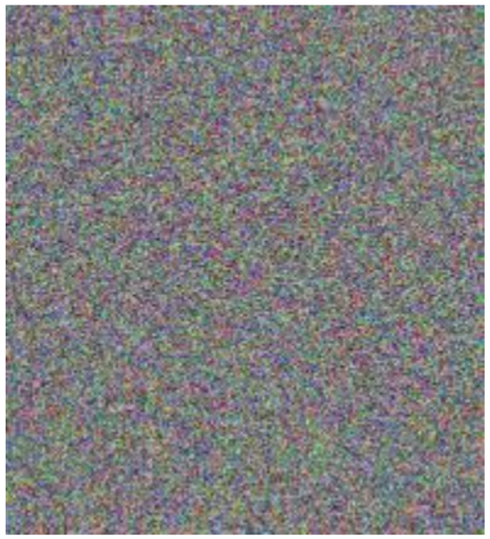
ECB 工作模式

- 直接将明文分解为多个块，对每个块进行独立加密
- 问题：无法隐藏数据特征



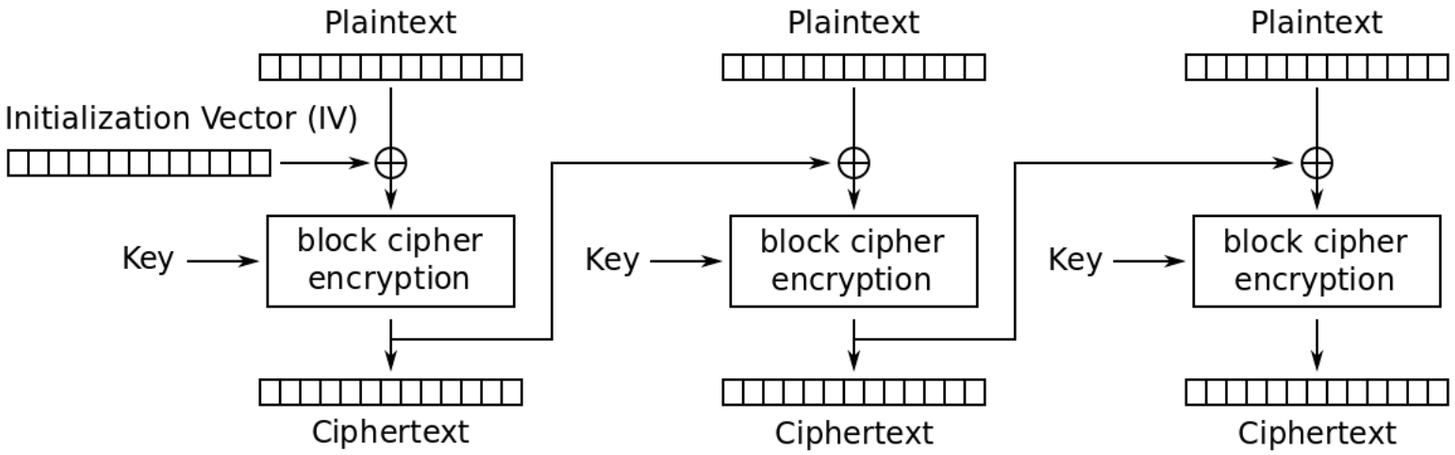
Electronic Codebook (ECB) mode encryption





CBC 工作模式

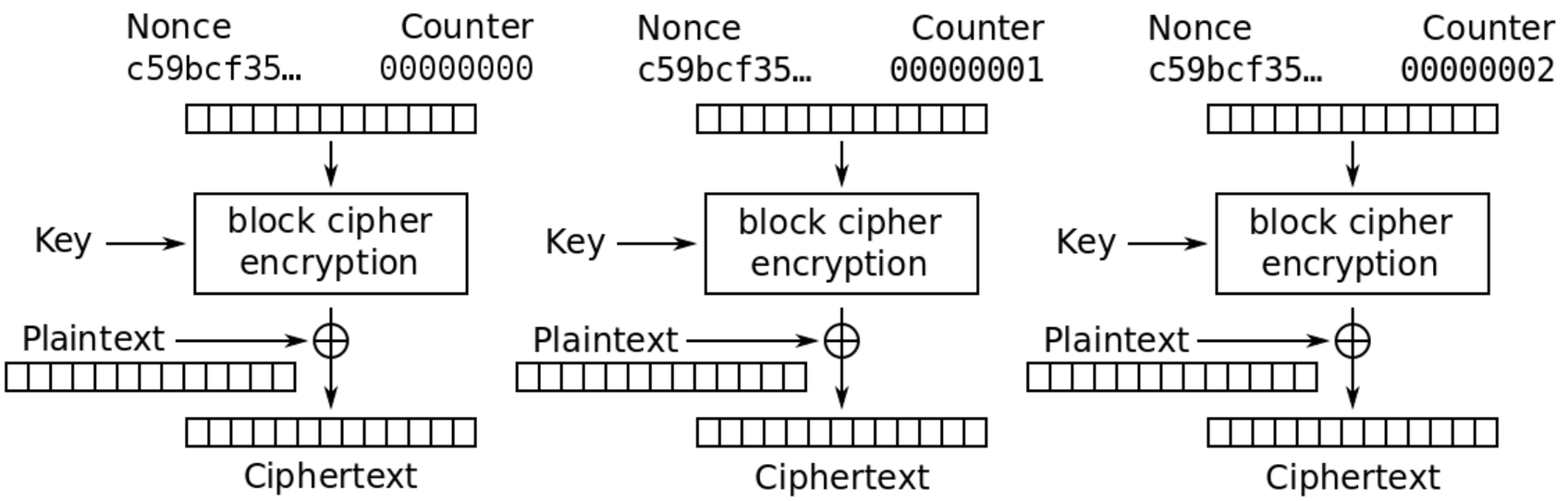
- 每个明文块先于前一个密文块进行异或后，在进行加密
- 问题：加密过程串行化



Cipher Block Chaining (CBC) mode encryption

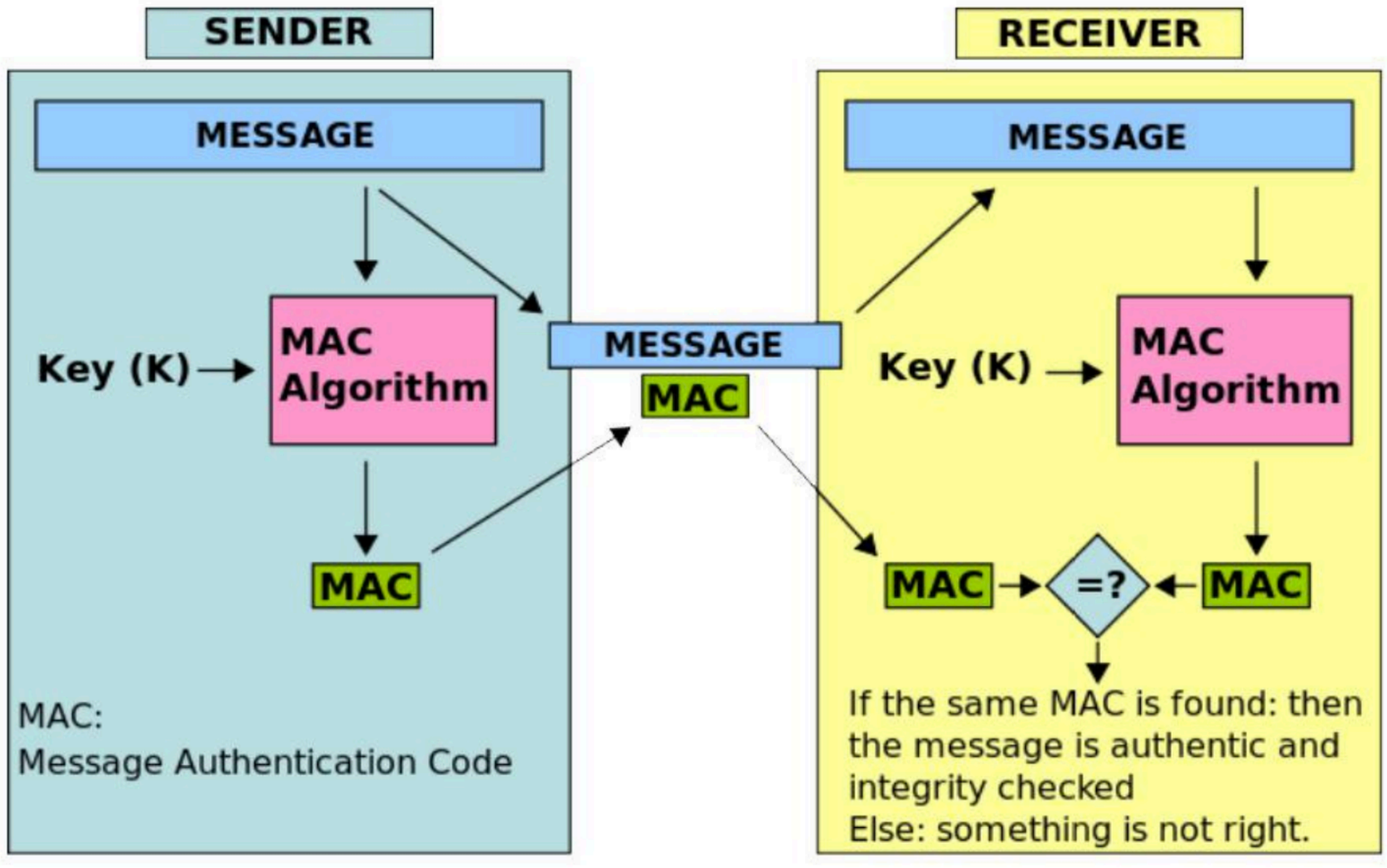
CTR 工作模式

- 通过递增一个加密计数器以产生连续的密钥流
- 问题：不能提供密文的完整性校验



Counter (CTR) mode encryption

完整性校验：MAC（Message Authentication Code）

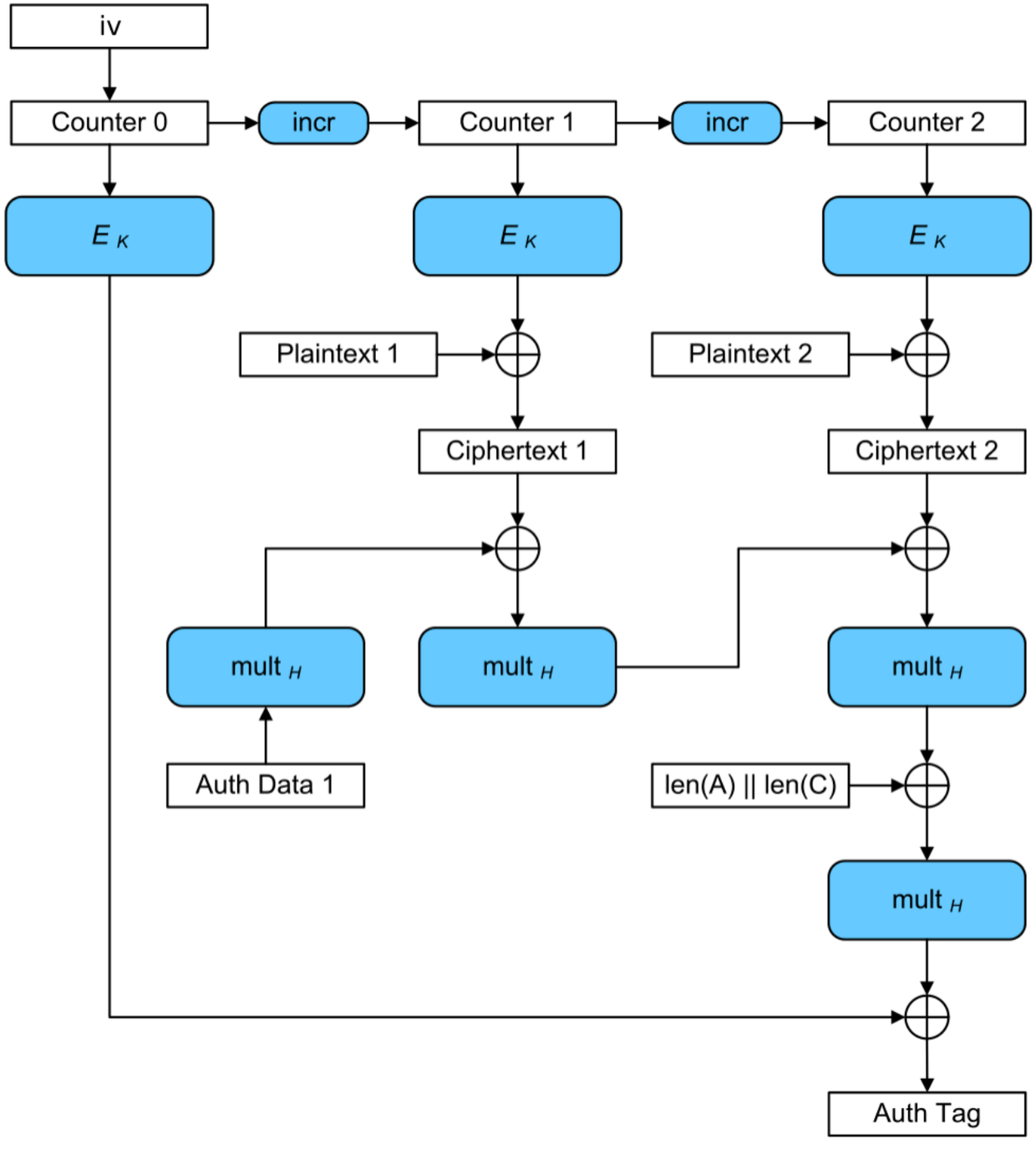


认证加密

- AE（认证加密）
  - EtM（Encrypt-then-MAC）
  - MtE
- AEAD（带有关联数据的认证加密）

GCM

- CTR + GHASH



AES 对称加密算法

AES 对称加密算法

- 常用填充算法：PKCS7
- 常用分组模式：GCM
- 三种密钥长度

AES	密钥长度（32 位比特)	分组长度(32 位比特)	加密轮数
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

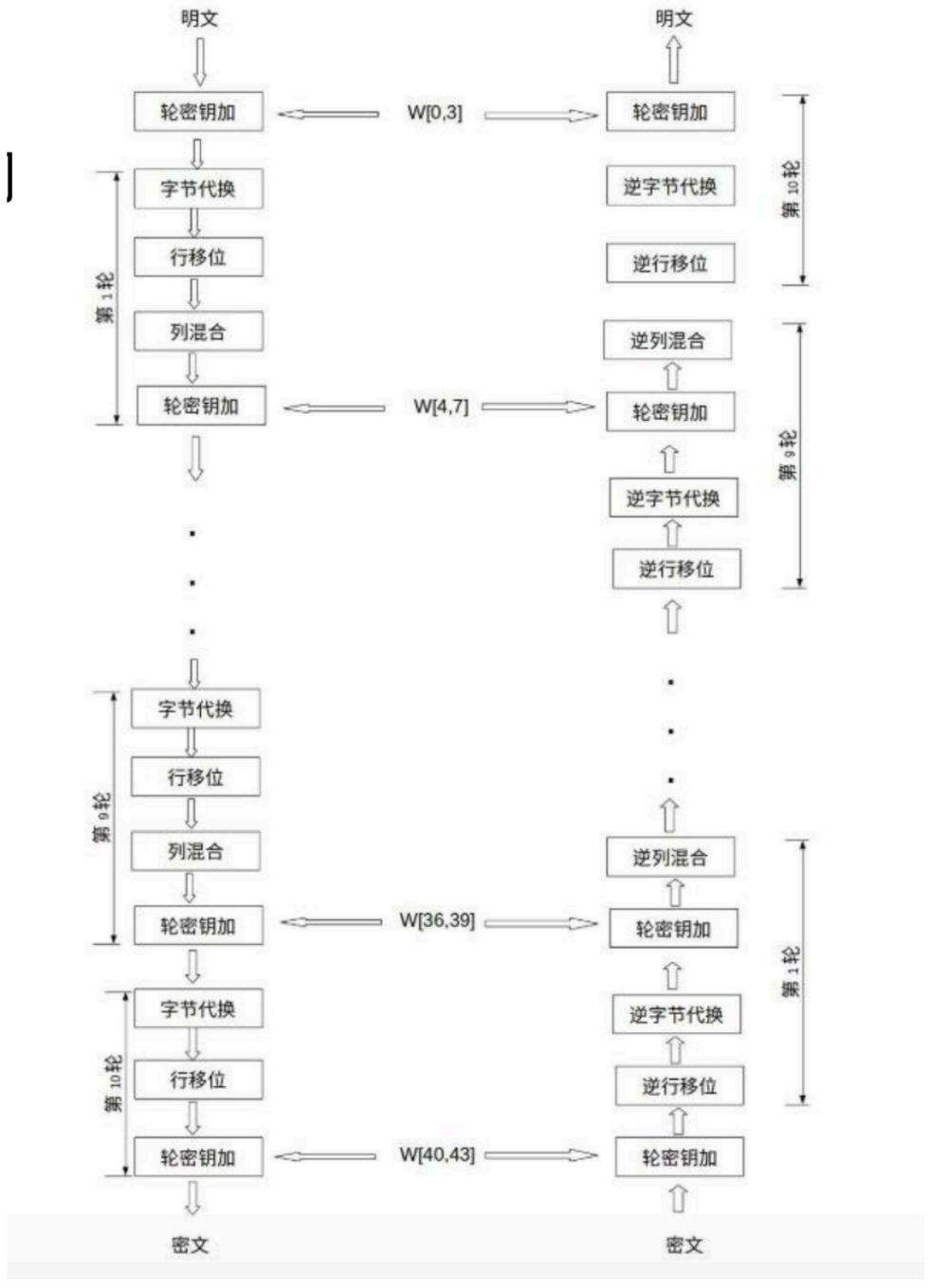
AES 加密步骤

- 把明文按照 128bit（16 字节）拆分为多个明文块，每个明文块是一个 4*4 的矩阵
- 按照选择的填充方式来填充最后一个明文块
- 每一个明文块利用 AES 加密器和密钥，加密成密文块
- 拼接所有的密文块，成为最终的密文结果

AES 加密流程

- C = E(K, P)，E 为每一轮算法，每轮密钥皆不同

- 初始轮
  - AddRoundKey 轮密钥加
- 普通轮
  - AddRoundKey 轮密钥加
    - SubBytes 字节替代
    - ShiftRows 行移位
    - MixColumns 列混合
- 最终轮
  - SubBytes 字节替换
  - ShiftRows 行移位
  - AddRoundKey 轮密钥加



非对称加密

非对称加密

- 公钥加密，只有私钥能解密
- 私钥加密，公钥解密

RSA 算法中公私钥的产生

1.随机选择两个不相等的质数 p 和 q

2.计算 p 和 q 的乘积 n

3.计算 n 的欧拉函数 v=(p-1)*(q-1)

4.随机选择一个整数 k

- 1 < k < v，且 k 与 v 互质

5.计算 k 对于 v 的模反元素 d

6.公钥： (k, n)

7.私钥： (d, n)

RSA algorithm

- Select two large prime numbers p, q
- Compute
  $n = p \times q$
  $v = (p-1) \times (q-1)$
- Select small odd integer k relatively prime to v
  $\gcd(k, v) = 1$
- Compute d such that
  $(d \times k)\%v = (k \times d)\%v = 1$
- Public key is (k, n)
- Private key is (d, n)

- example
  - p = 11
  - q = 29
  - n = 319
  - v = 280
  - k = 3
  - d = 187
- public key
  (3, 319)
- private key
  (187, 319)

RSA 算法加解密流程

- 加密：$c = m^k \pmod n$
  - m 是明文，c 是密文
- 解密：$m = c^d \pmod n$
- 举例：对明文数字 123 加解密
  - 公钥（3，319）加密
    - 123^3 mod 319 = 140
  - 对 140 密文用私钥（187，319）解密
    - 140^187 mod 319 = 123
- 私钥（187，319）加密
  - 123^187 mod 319 = 161
  - 公钥（3，319）解密
    - 161^3 mod 319 = 123

RSA algorithm

- Select two large prime numbers p, q
- Compute
  n = p × q
  v = (p-1) × (q-1)
- Select small odd integer k relatively prime to v
  gcd(k, v) = 1
- Compute d such that
  (d × k)%v = (k × d)%v = 1
- Public key is (k, n)
- Private key is (d, n)

- example
  - p = 11
  - q = 29
  - n = 319
  - v = 280
  - k = 3
  - d = 187
- public key
  (3, 319)
- private key
  (187, 319)

基于 openssl 验证 RSA

1.生成私钥：openssl genrsa -out private.pem

2.从私钥里提取公钥：openssl rsa -in private.pem -pubout -out public.pem

3.生成加密文件：openssl rsautl -encrypt -in hello.txt -inkey public.pem -pubin -out hello.en

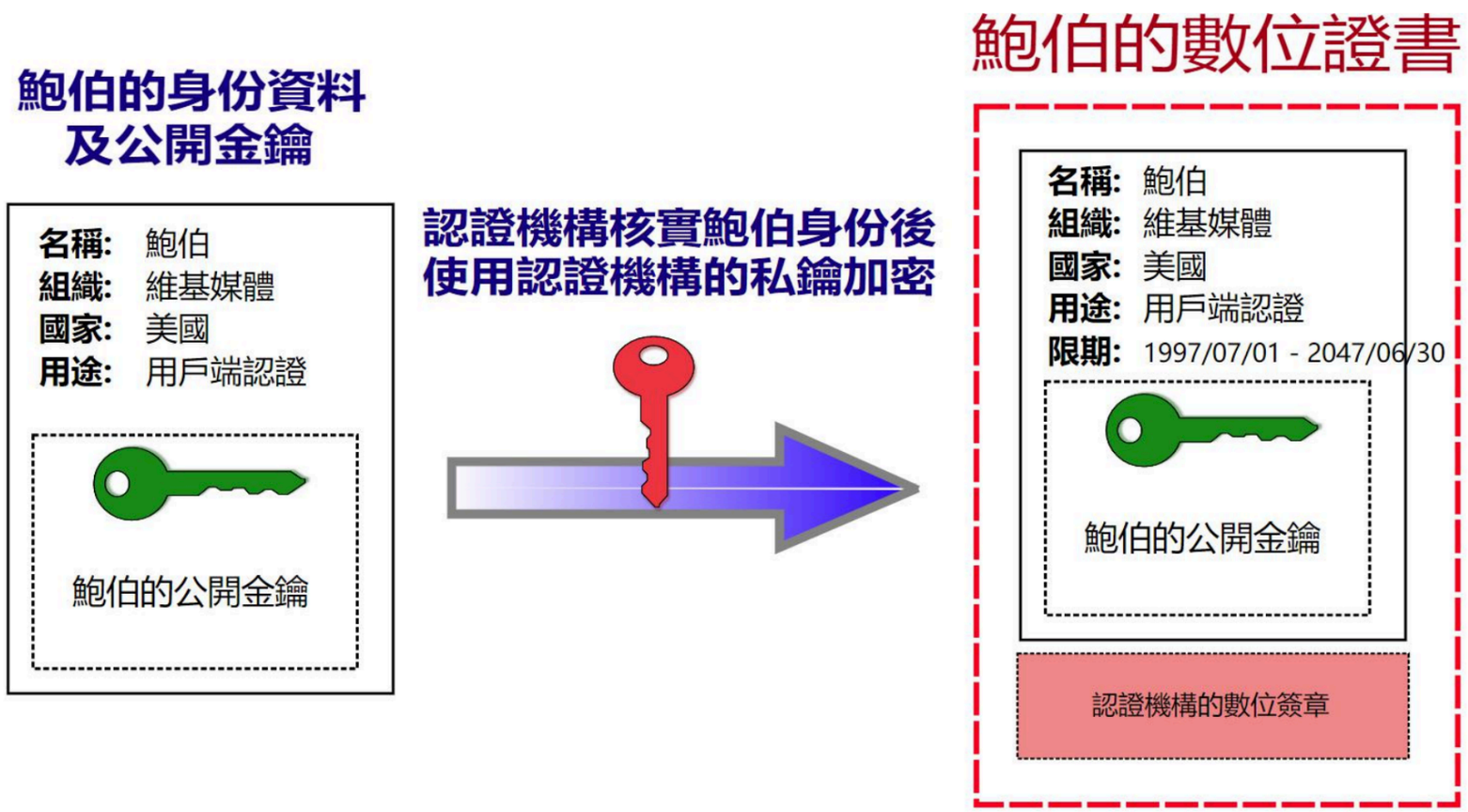
4.生成解密后的文件：openssl rsautl -decrypt -in hello.en -inkey private.pem -out hello.de

非对称密码应用

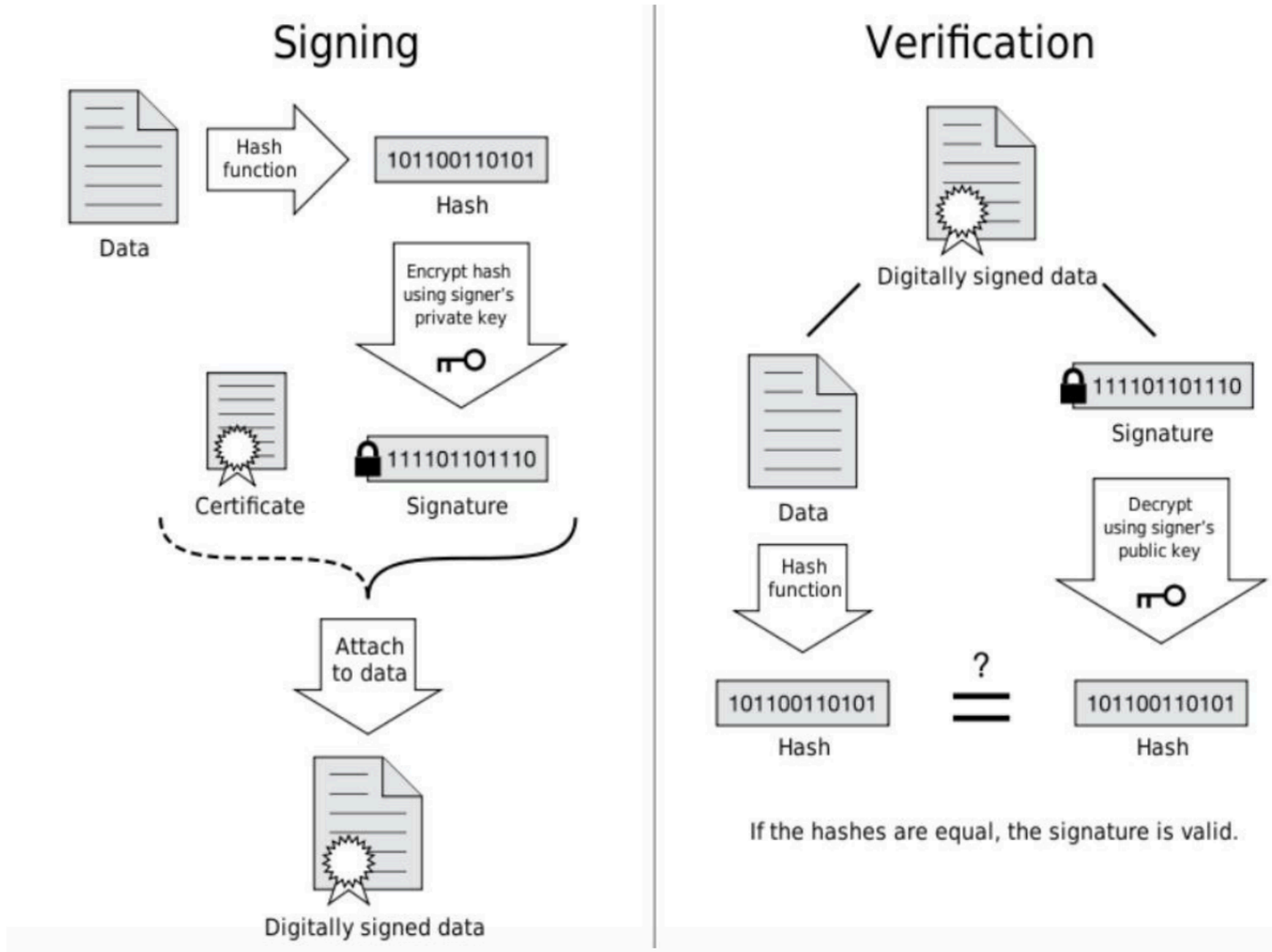
非对称密码应用：数字签名

- 基于私钥加密，只能使用公钥解密：起到身份认证的使用
- 公钥管理：Public Key Infrastructure（PKI）公钥基础设施
  - 由 Certificate Authority（CA）数字证书认证结构将用户个人身份与公开密钥关联在一起
  - 公钥数字证书组成
    - CA 信息、公钥用户信息、公钥、权威结构的签名、有效期
  - PKI 用户
    - 向 CA 注册公钥的用户
    - 希望使用已注册公钥的用户

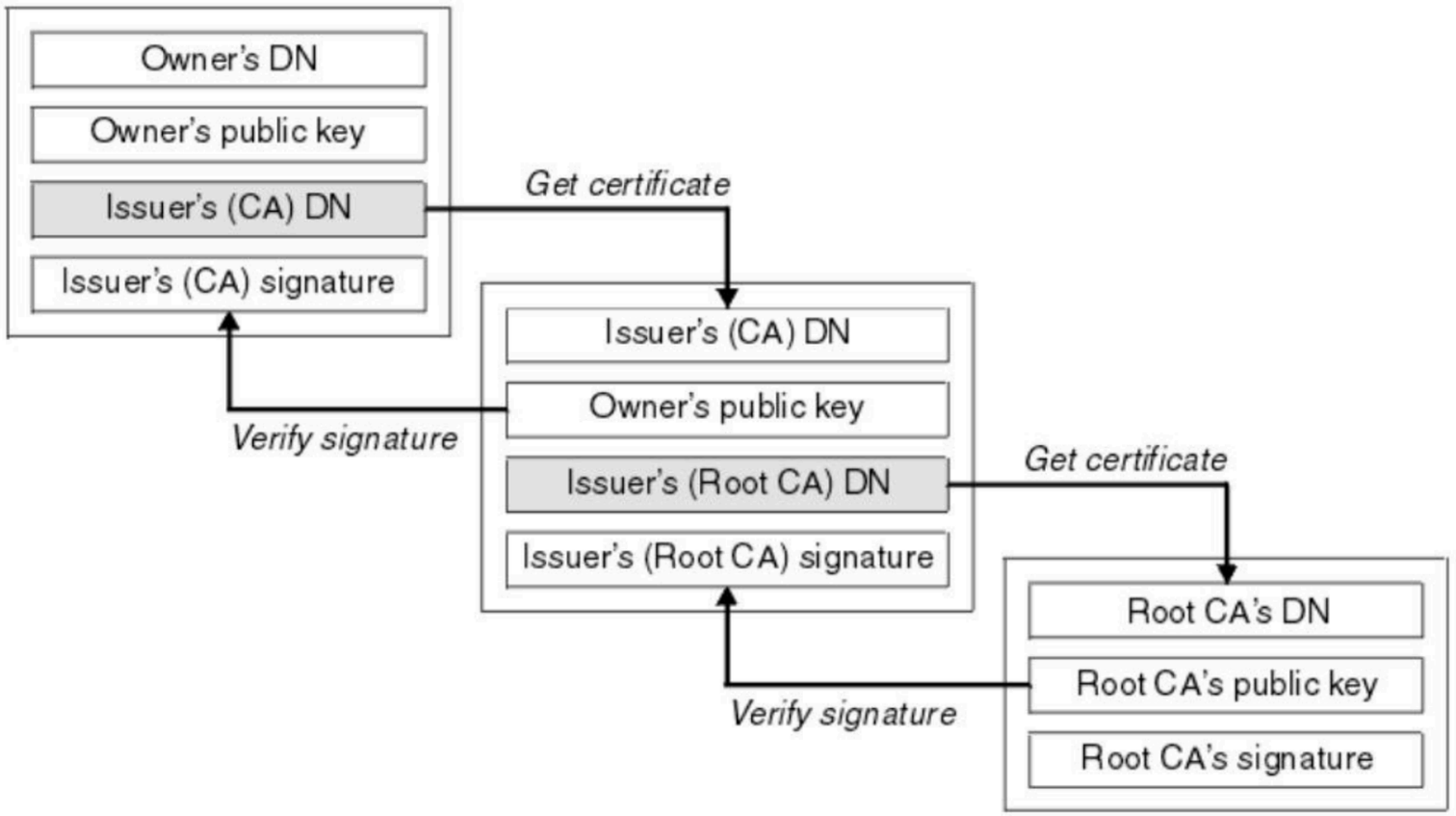
签发证书流程


鮑伯的身份資料
及公開金鑰
名稱: 鮑伯
組織: 維基媒體
國家: 美國
用途: 用戶端認證
鮑伯的公開金鑰
認證機構核實鮑伯身份後
使用認證機構的私鑰加密
鮑伯的數位證書
名稱: 鮑伯
組織: 維基媒體
國家: 美國
用途: 用戶端認證
限期: 1997/07/01 - 2047/06/30
鮑伯的公開金鑰
認證機構的數位簽章

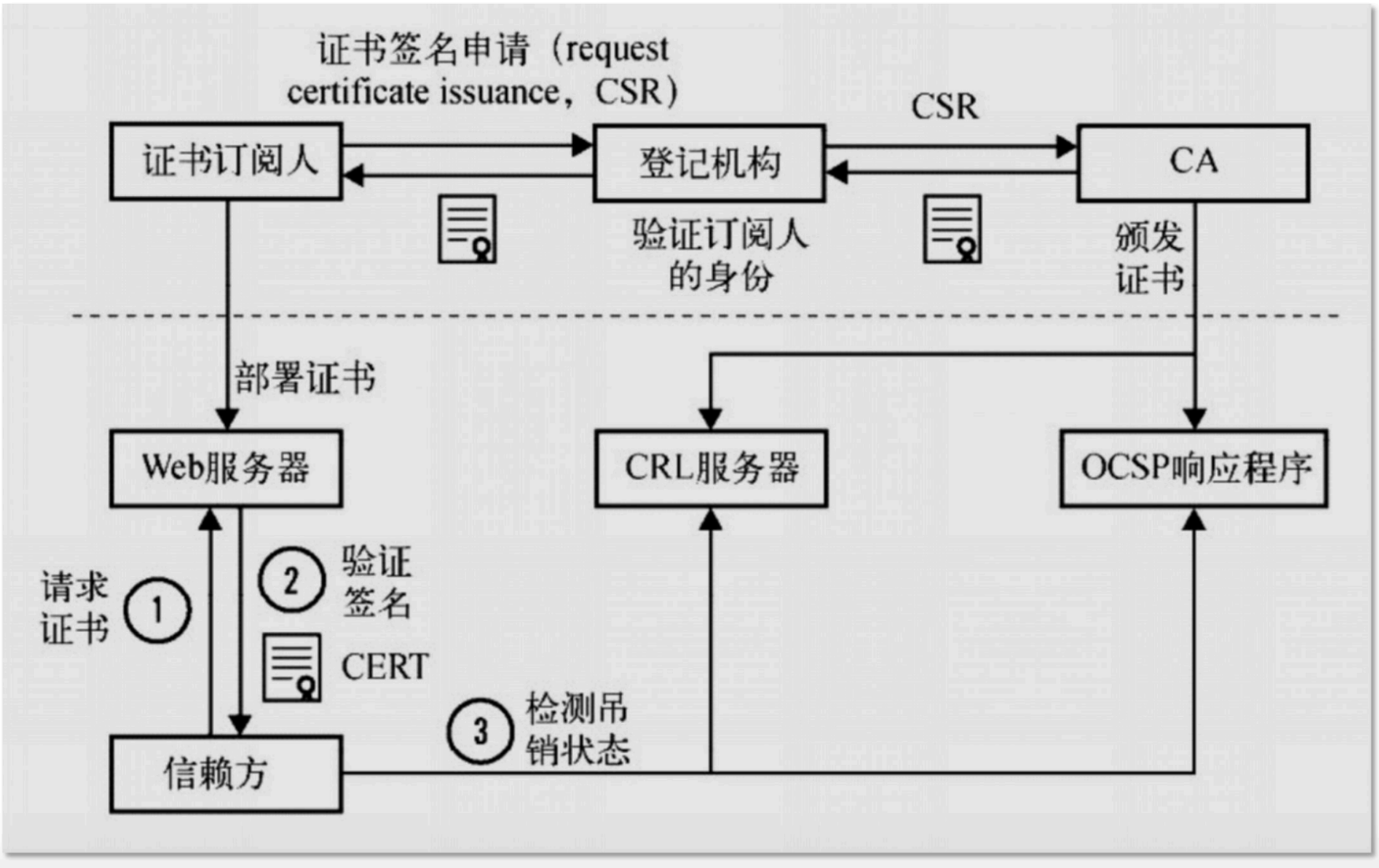
签名与验签流程


Signing
Data
Hash function
101100110101
Hash
Encrypt hash using signer's private key
Certificate
111101101110
Signature
Attach to data
Digitally signed data
Verification
Digitally signed data
Data
111101101110
Signature
Hash function
Decrypt using signer's public key
101100110101
Hash
?
101100110101
Hash
If the hashes are equal, the signature is valid.

证书信任链


Owner's DN
Owner's public key
Issuer's (CA) DN
Issuer's (CA) signature
Get certificate
Issuer's (CA) DN
Owner's public key
Issuer's (Root CA) DN
Issuer's (Root CA) signature
Verify signature
Get certificate
Root CA's DN
Root CA's public key
Root CA's signature
Verify signature

PKI 公钥基础设施

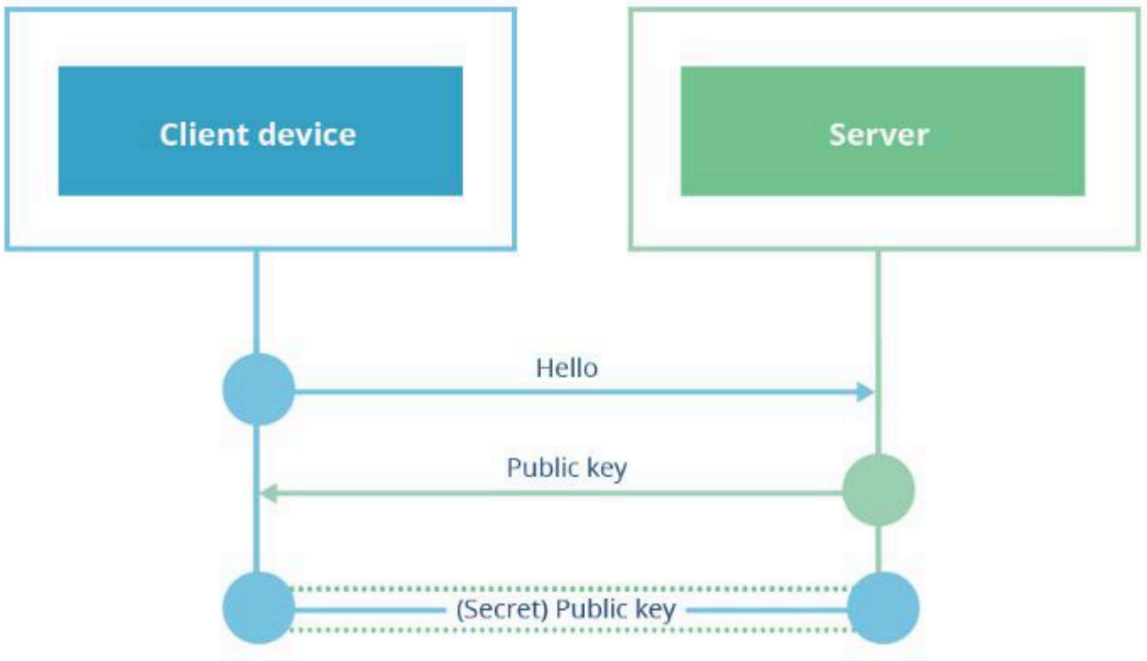

证书签名申请（request certificate issuance，CSR）
CSR
证书订阅人
登记机构
CA
验证订阅人的身份
颁发证书
部署证书
Web服务器
CRL服务器
OCSP响应程序
请求证书
1
2
验证签名
CERT
3
检测吊销状态
信赖方

验证证书链

非对称密码的应用：DH 密钥协商协议

RSA 密钥交换

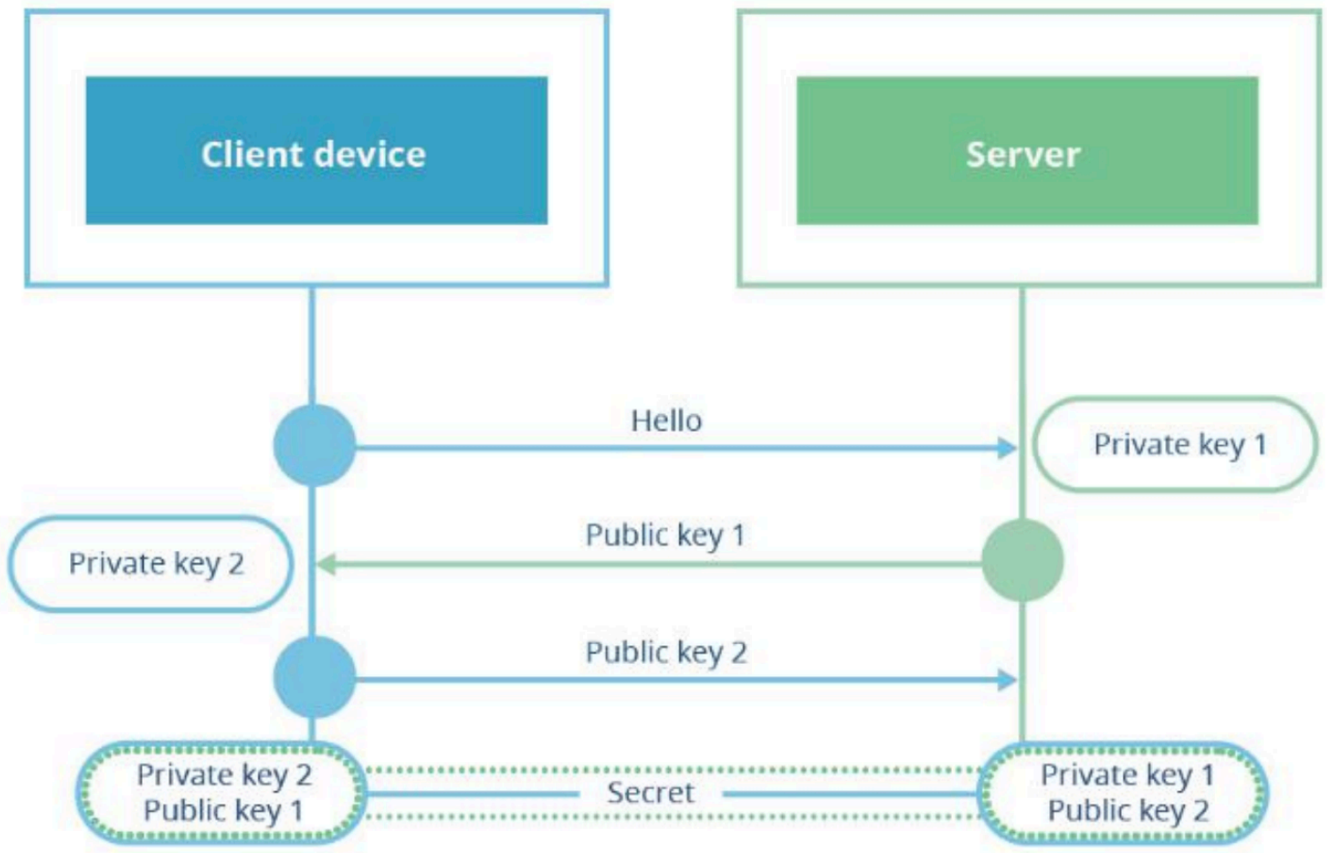
- 由客户端生成对称加密的密钥



- 问题：不具备前向安全性

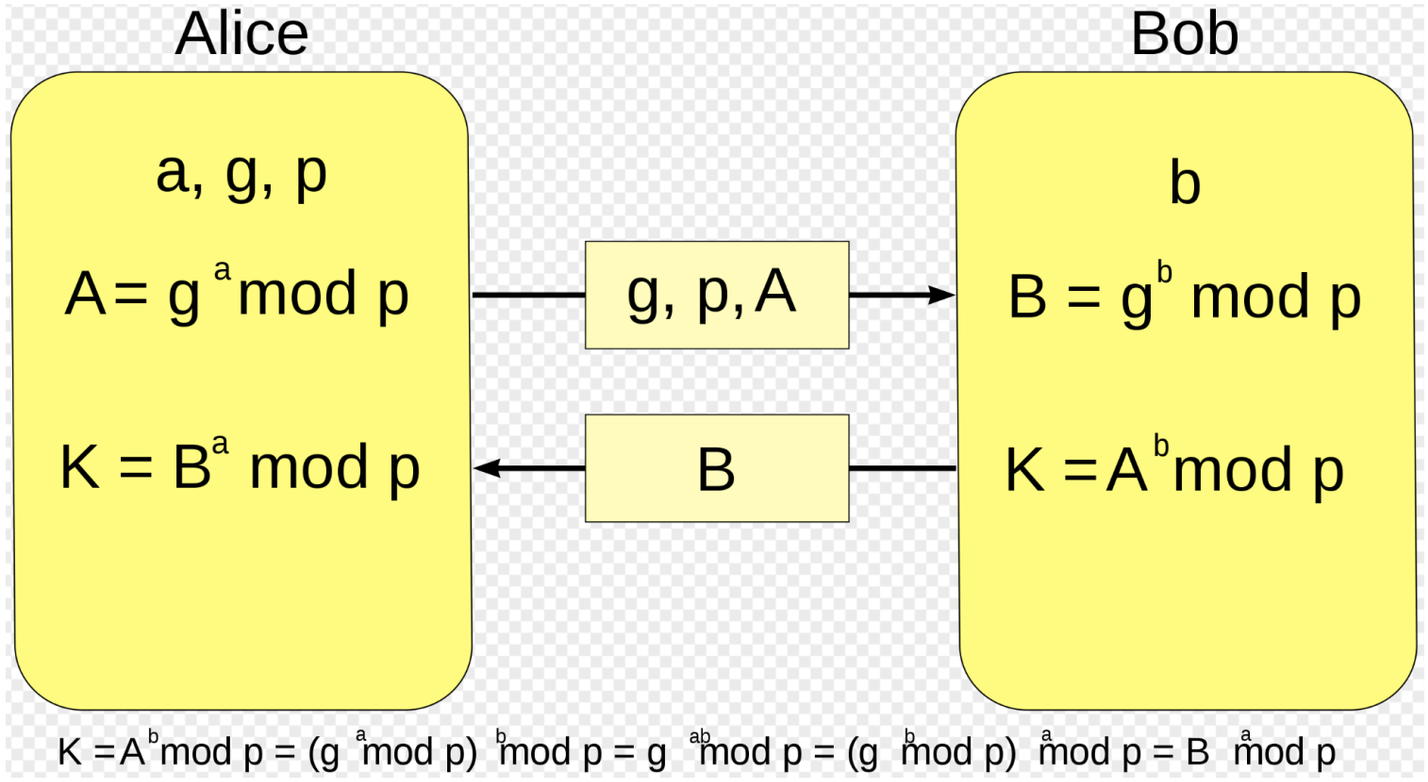
DH 密钥协商

- 它可以让双方在完全没有对方任何预先信息的条件下通过不安全信道创建起一个密钥



DH 密钥协商协议

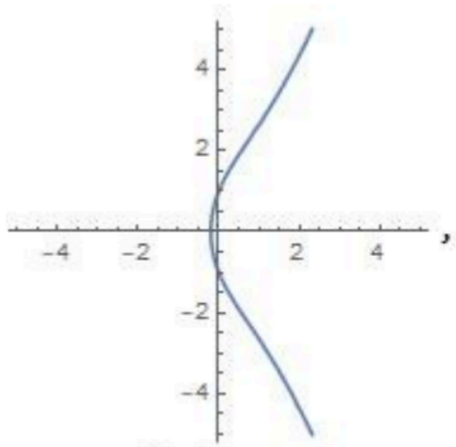
- a, b 保密
- g p a b 公开
- 生成共同密钥 K



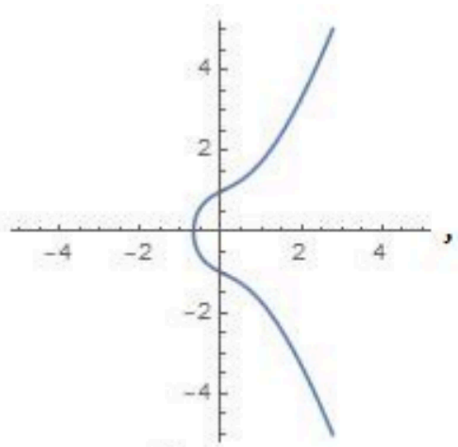
ECC 椭圆曲线

ECC 椭圆曲线的定义（secp256、secp384）

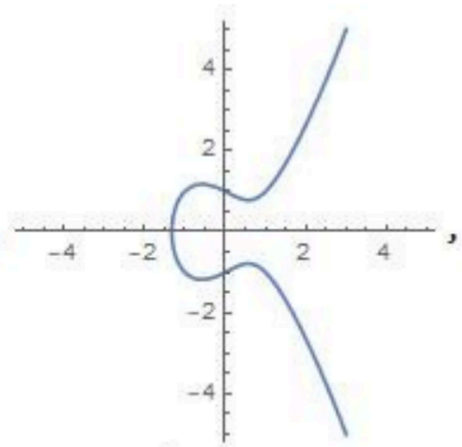
- 表达式：$y^2 = x^3 + ax + b,\ 4a^3 + 27b^2 \neq 0$
- 例如：



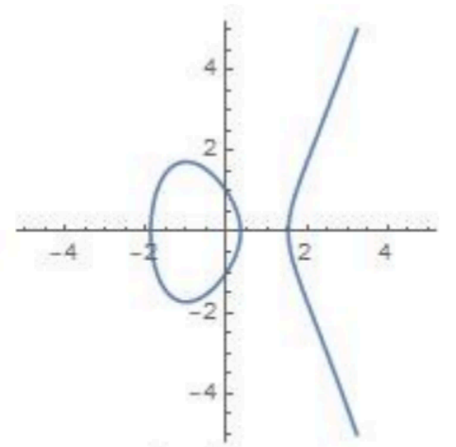
$y^2=x^3+5x+1$,



$y^2=x^3+x+1$,



$y^2=x^3-x+1$,



$y^2=x^3-3x+1$

TLS 协议分析

TLS 协议概述

1.握手协议（Handshaking Protocols）

1.握手协议（Handshaking Protocol）

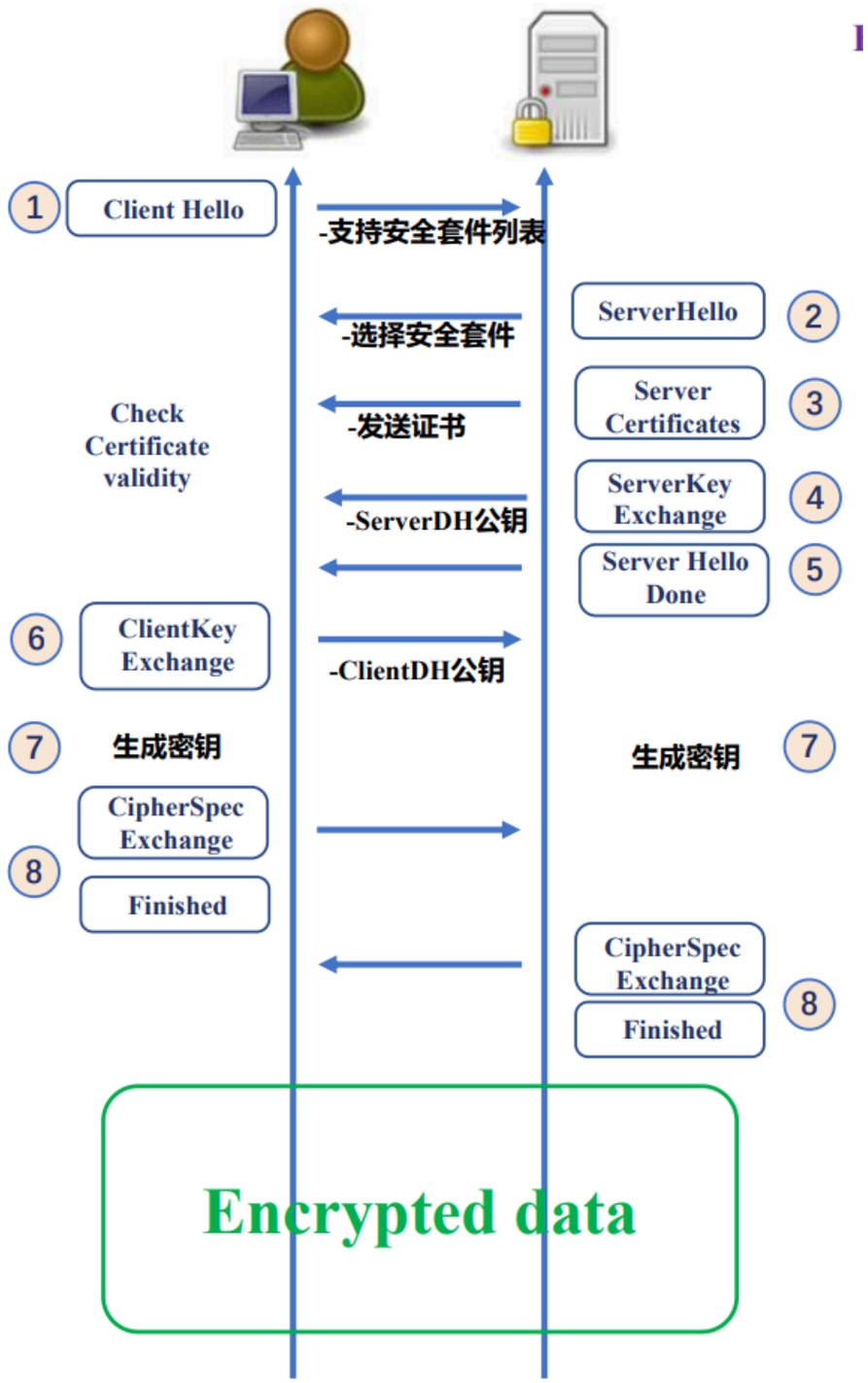
2.警告协议（Alert Protocol）

3.应用层协议（Application Data Protocol）

4.密码切换协议（Change Cipher Spec Protocol）

2.记录层协议（Record Protocol）

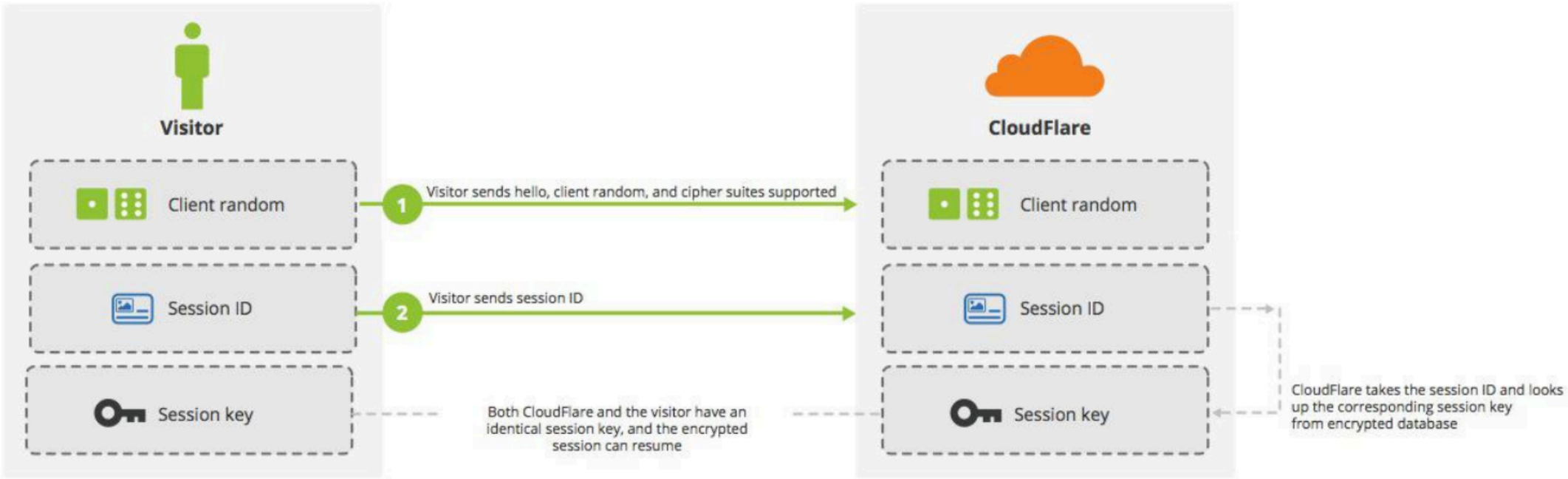
TLS 握手过程


1
Client Hello
-支持安全套件列表
ServerHello
2
-选择安全套件
Server Certificates
3
Check Certificate validity
-发送证书
ServerKey Exchange
4
-ServerDH公钥
Server Hello Done
5
6
ClientKey Exchange
-ClientDH公钥
7
生成密钥
生成密钥
7
CipherSpec Exchange
8
Finished
CipherSpec Exchange
Finished
8
Encrypted data

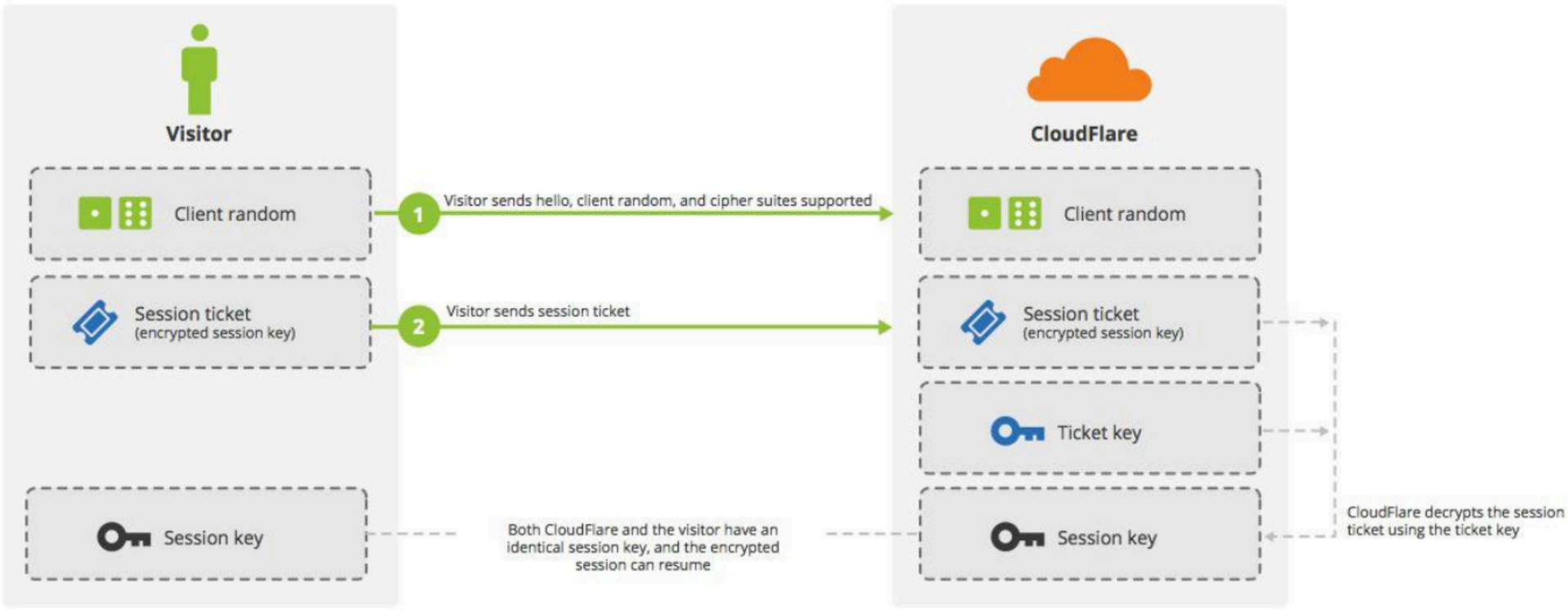
会话恢复

- 基于 Session ID
  - 2 RTT -> 1 RTT
  - 是 TLS 协议的一部分，兼容性和安全性更好
  - 服务器存储 + 不支持分布式 Session Cache
- 基于 SessionTicket
  - 客户端存储、服务端解密
  - TLS 协议的一个扩展

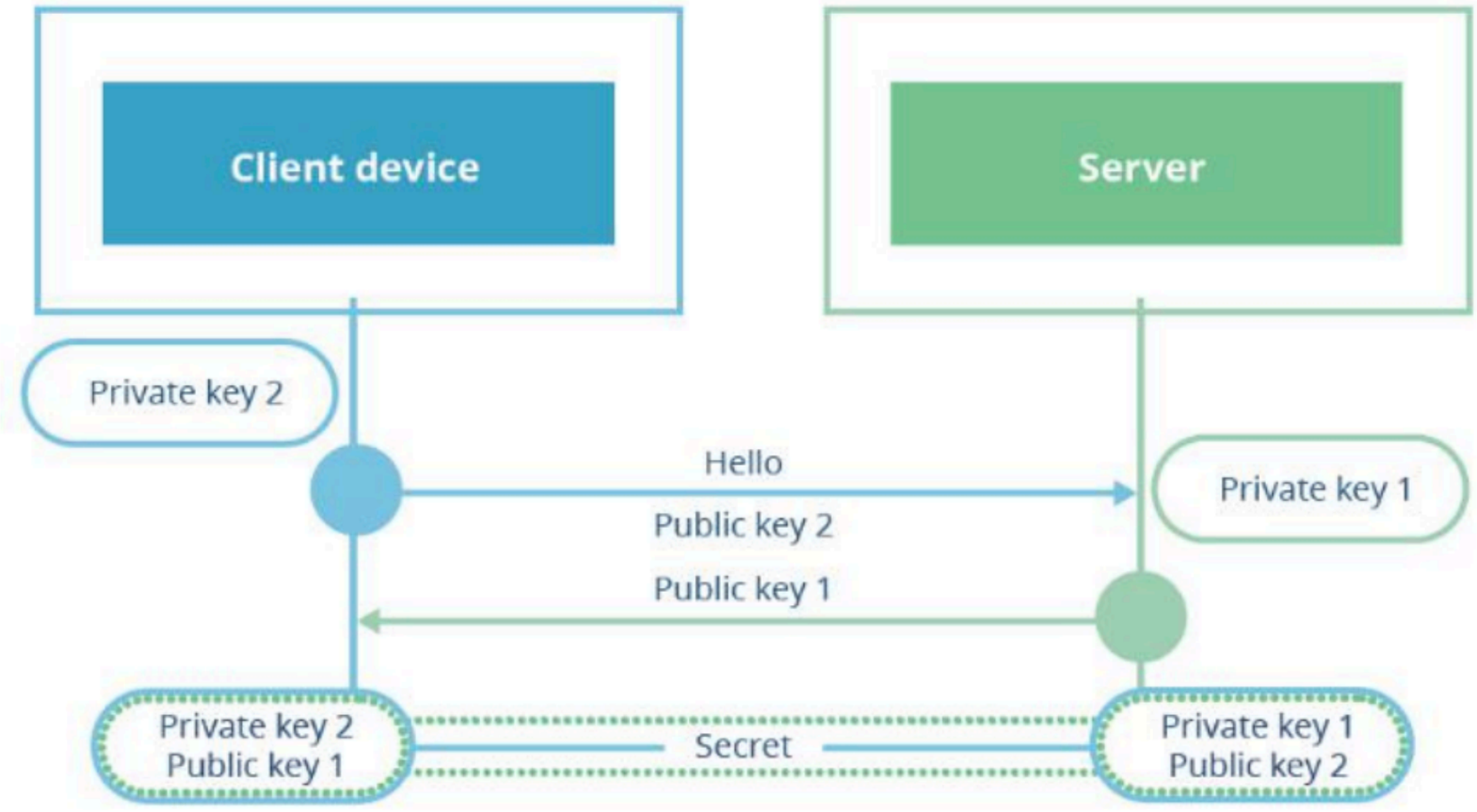
会话恢复之 Session Id



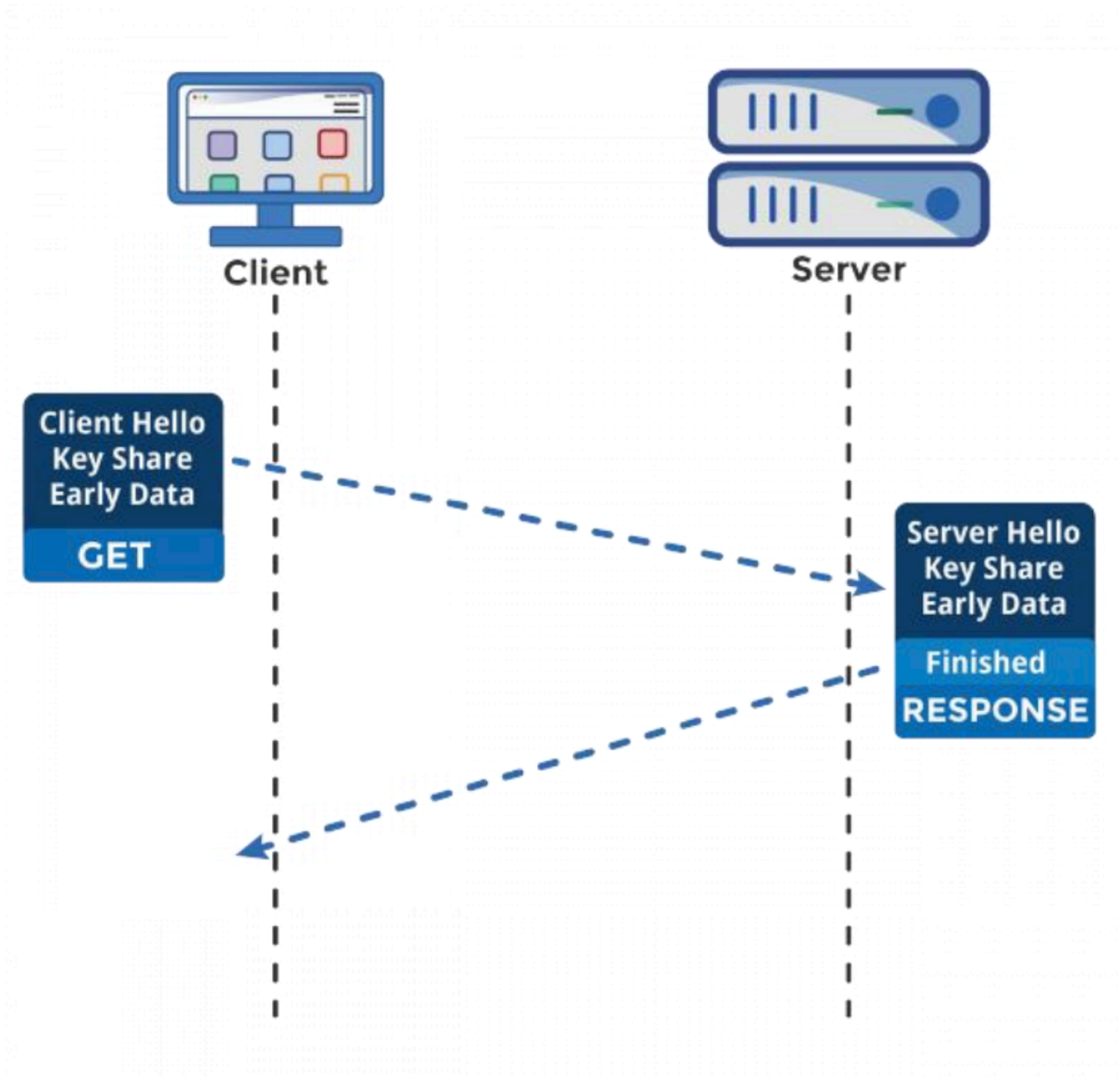
会话恢复之 Session Ticket



TLS 1.3 密钥交换


Client device
Server
Private key 2
Hello
Public key 2
Private key 1
Public key 1
Private key 2
Public key 1
Secret
Private key 1
Public key 2

TLS1.3 0 RTT 握手



参考资料

书籍：

1.《深入浅出 HTTPS 从原理到实战》

专栏：

1. Web协议详解与抓包实战

链接：

1. TLS 协议分析与现代加密通信设计
2. TLS 1.3 文档