

第 4 部分 TLS/SSL 协议



扫码试看/订阅极客时间

《Web协议详解与抓包实战》视频课程

第 1 课 TLS/SSL 协议的工作原理

TLS 设计目的

- 身份验证
- 保密性
- 完整性

TLS/SSL 发展



SSL/TLS 通用模型

ISO/OSI 模型



TCP/IP 模型



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

TLS 协议

- Record 记录协议
 - 对称加密
- Handshake 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 安全密码套件解读



密钥交换算法

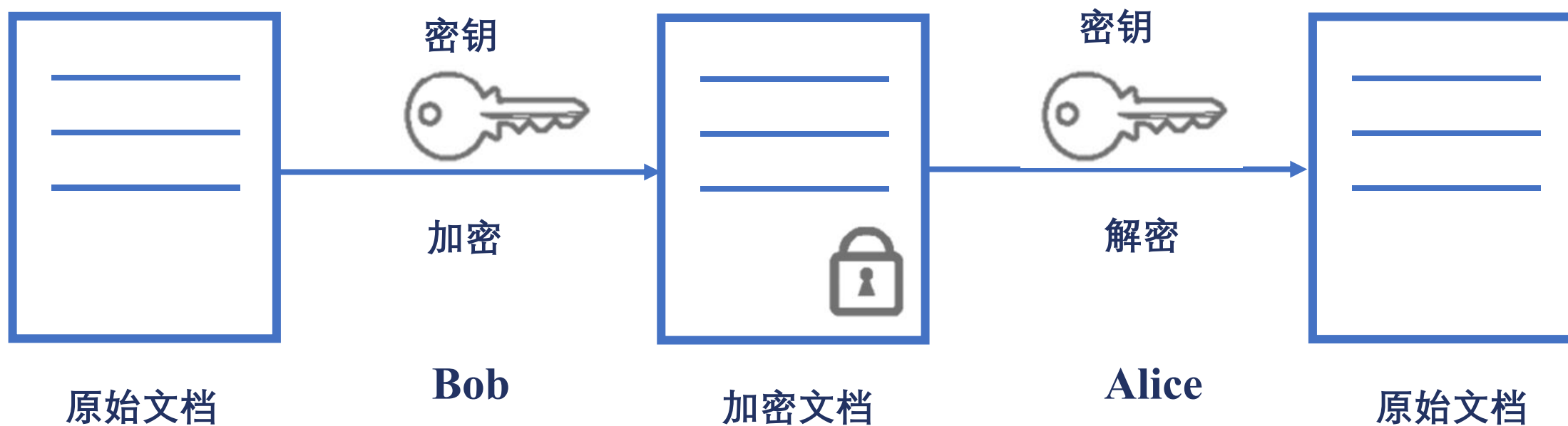
身份验证算法

对称加密算法、强度、工作模式

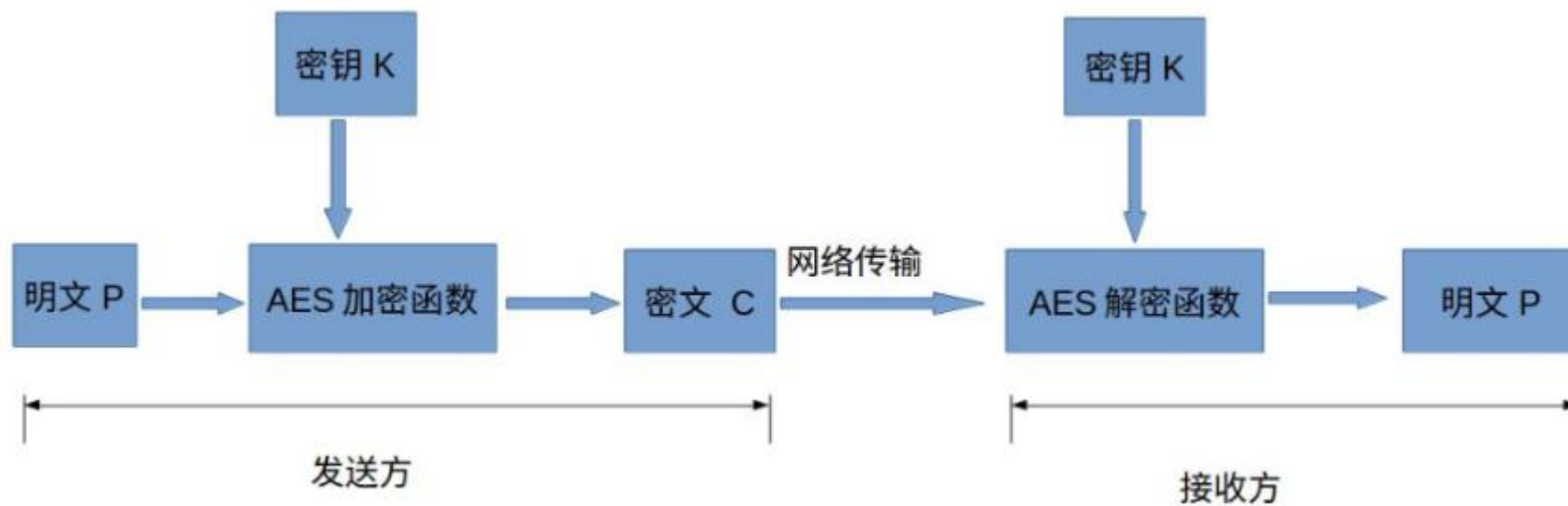
签名hash算法

第 2 课 对称加密的工作原理 (1) : XOR 与填充

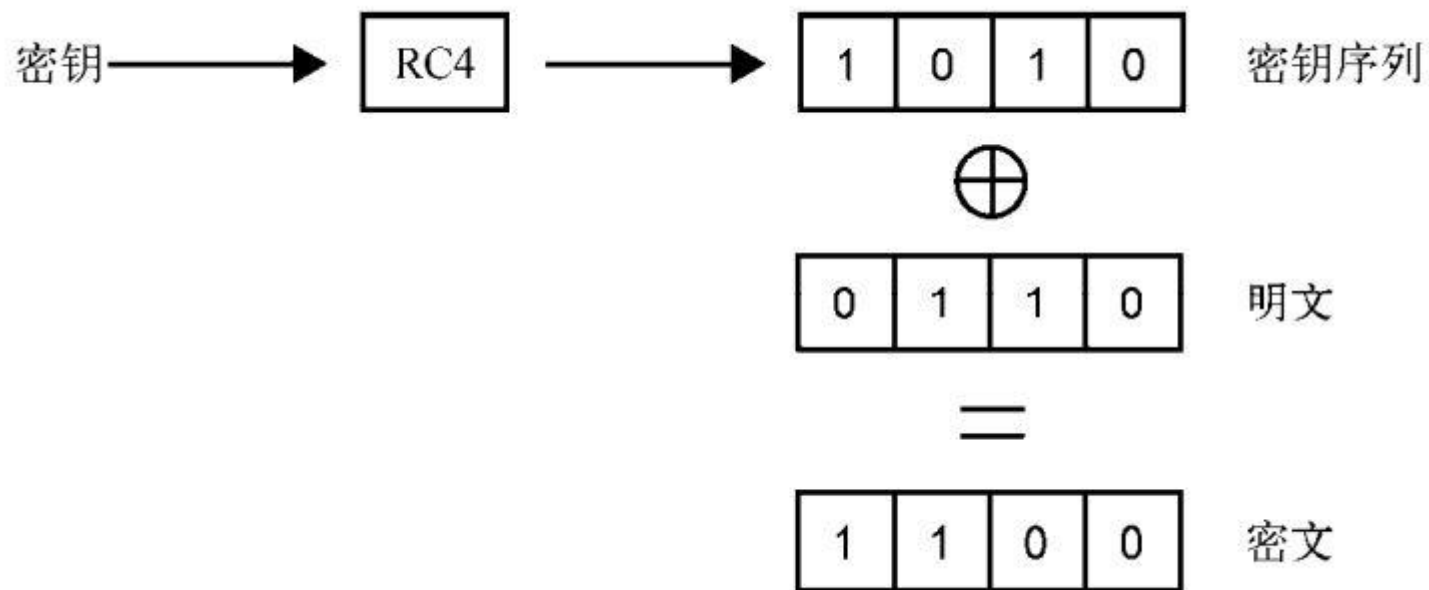
对称加密



AES 对称加密在网络中的应用



对称加密与 XOR 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

填充 padding

- Block cipher 分组加密：将明文分成多个等长的 Block 模块，对每个模块分别加解密
- 目的：当最后一个明文 Block 模块长度不足时，需要填充
- 填充方法
 - 位填充：以 bit 位为单位来填充
 - ... | 1011 1001 1101 0100 0010 0111 0000 0000 |
 - 字节填充：以字节为单位为填充
 - 补零：... | DD DD DD DD DD DD DD DD | DD DD DD DD 00 00 00 00 |
 - ANSI X9.23：... | DD DD DD DD DD DD DD DD | DD DD DD DD 00 00 00 04 |
 - ISO 10126：... | DD DD DD DD DD DD DD DD | DD DD DD DD 81 A6 23 04 |
 - PKCS7 (RFC5652) : ... | DD DD DD DD DD DD DD DD | DD DD DD DD 04 04 04 04 |

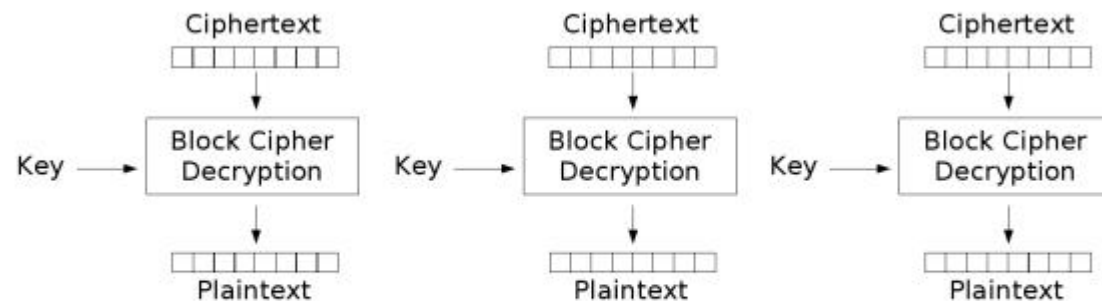
第 3 课 对称加密的工作原理（2）：工作模式

分组工作模式 block cipher mode of operation

- 允许使用同一个分组密码密钥对多于一块的数据进行加密，并保证其安全性

ECB (Electronic codebook) 模式

- 直接将明文分解为多个块，对每个块独立加密
- 问题：无法隐藏数据特征

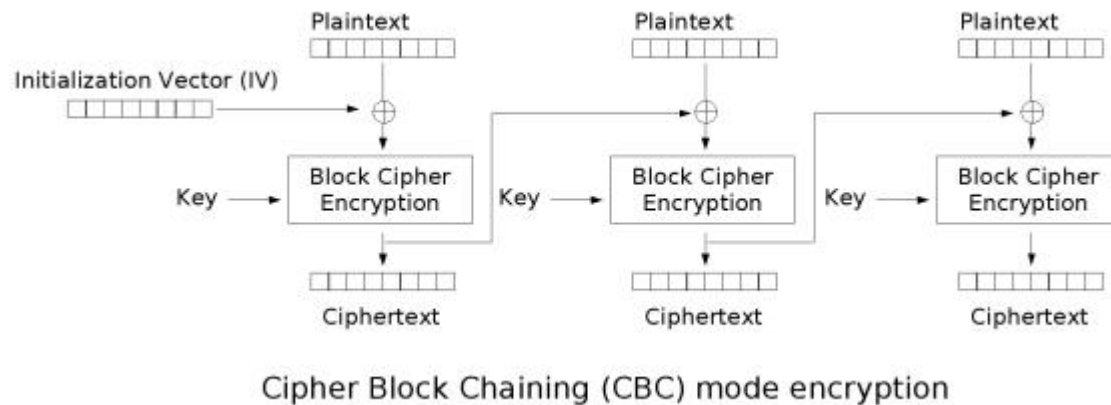


Electronic Codebook (ECB) mode decryption



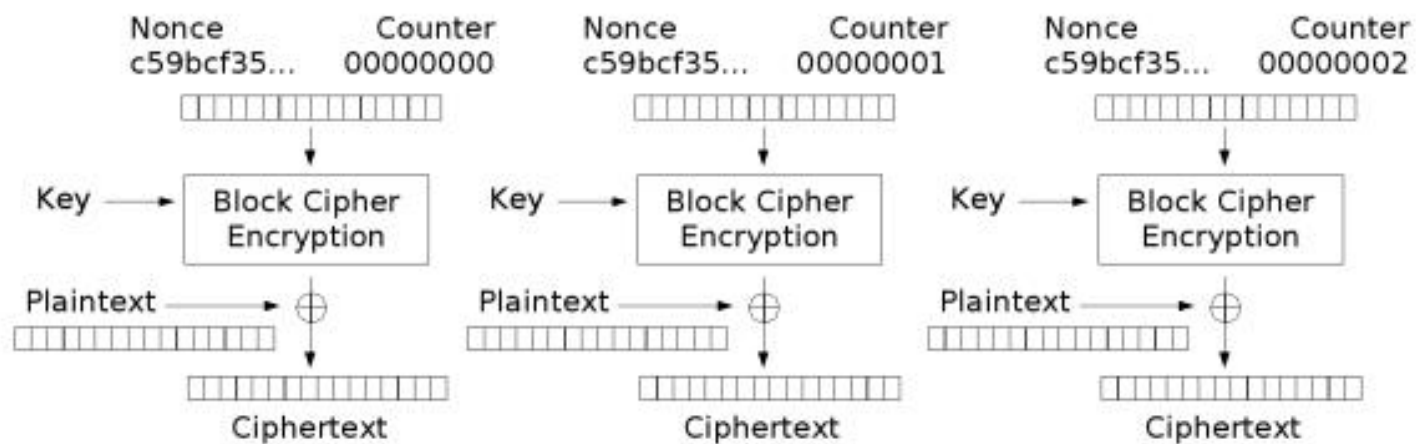
CBC (Cipher-block chaining) 模式

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



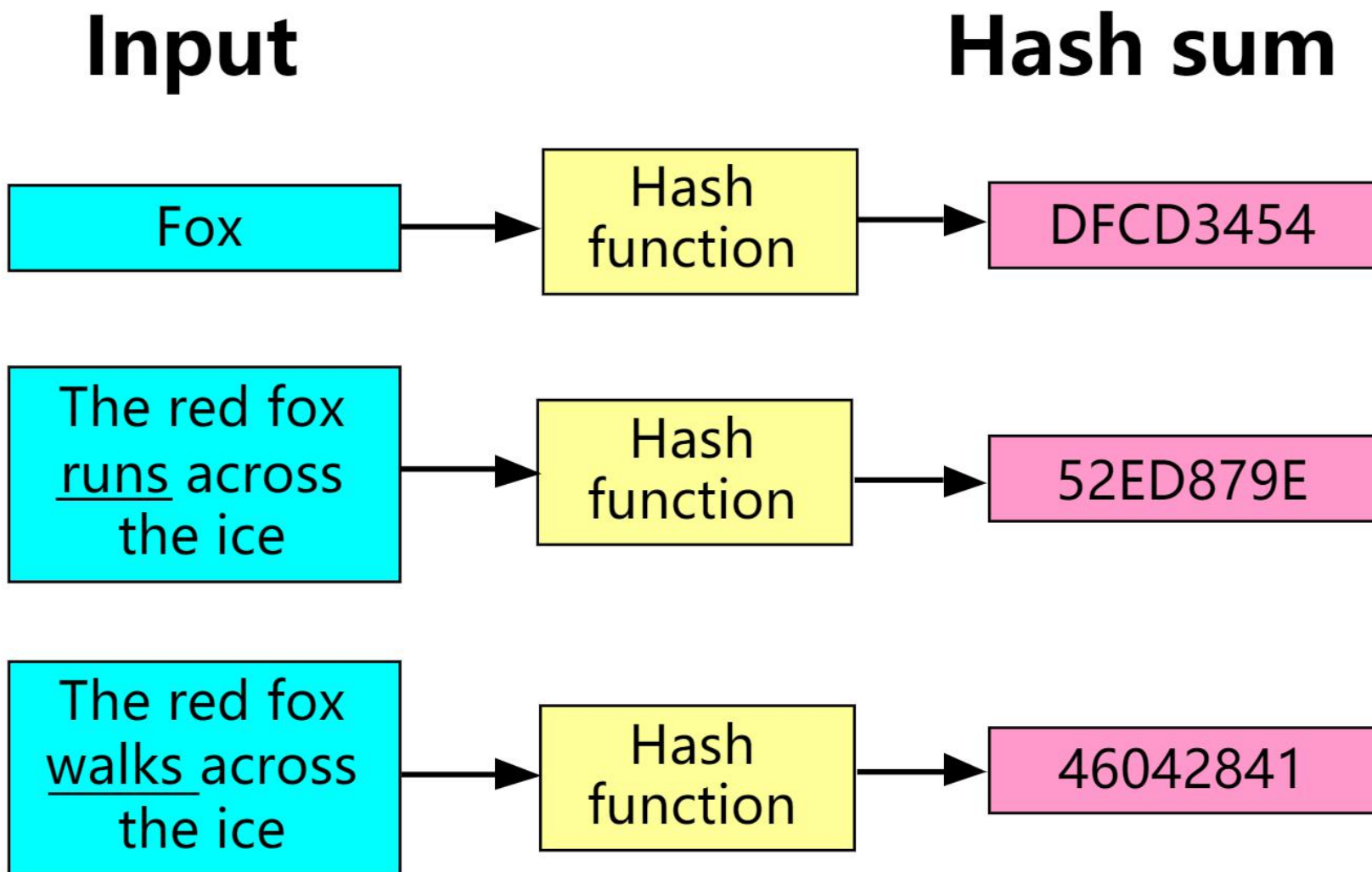
CTR (Counter) 模式

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

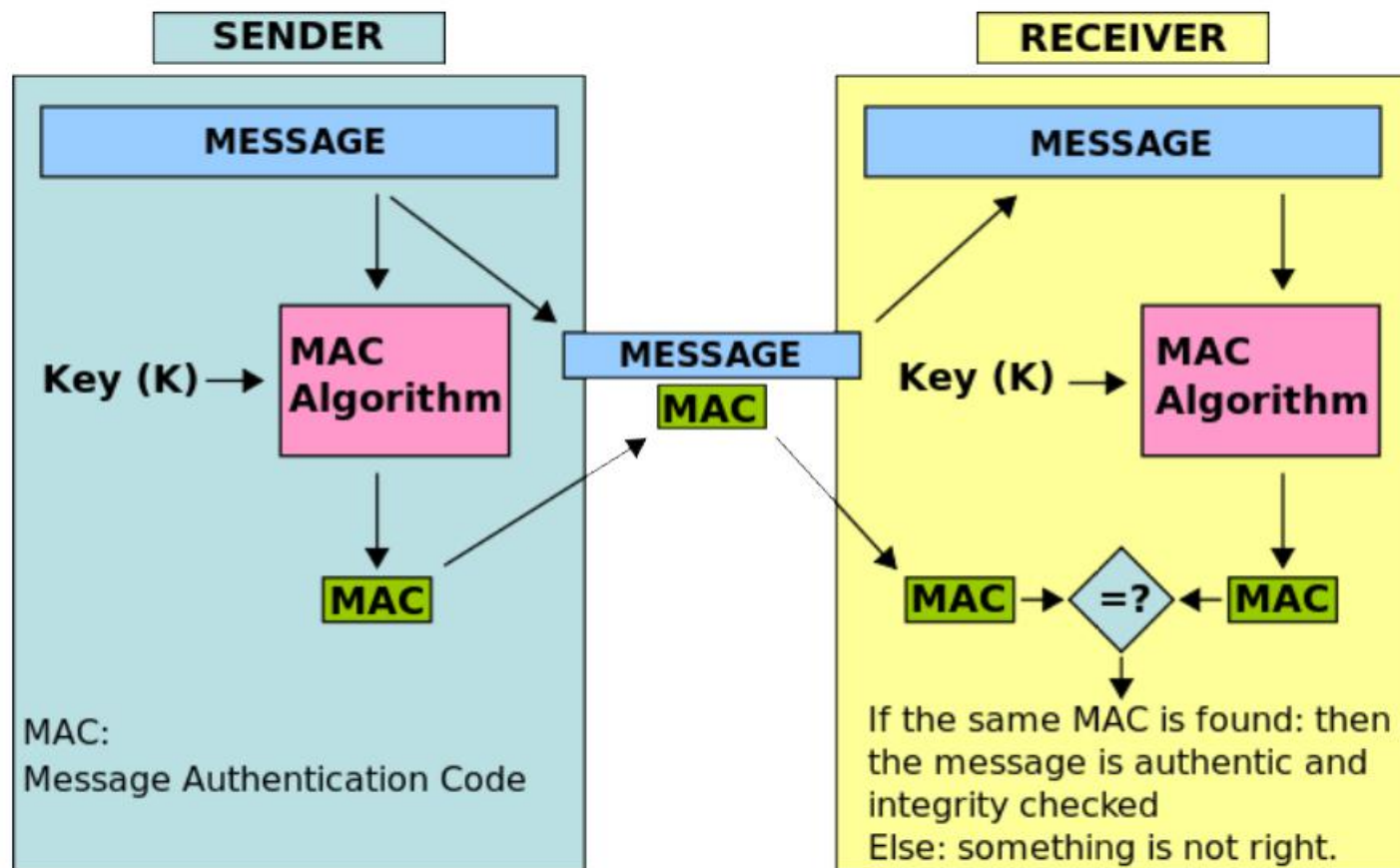


Counter (CTR) mode encryption

验证完整性: hash 函数

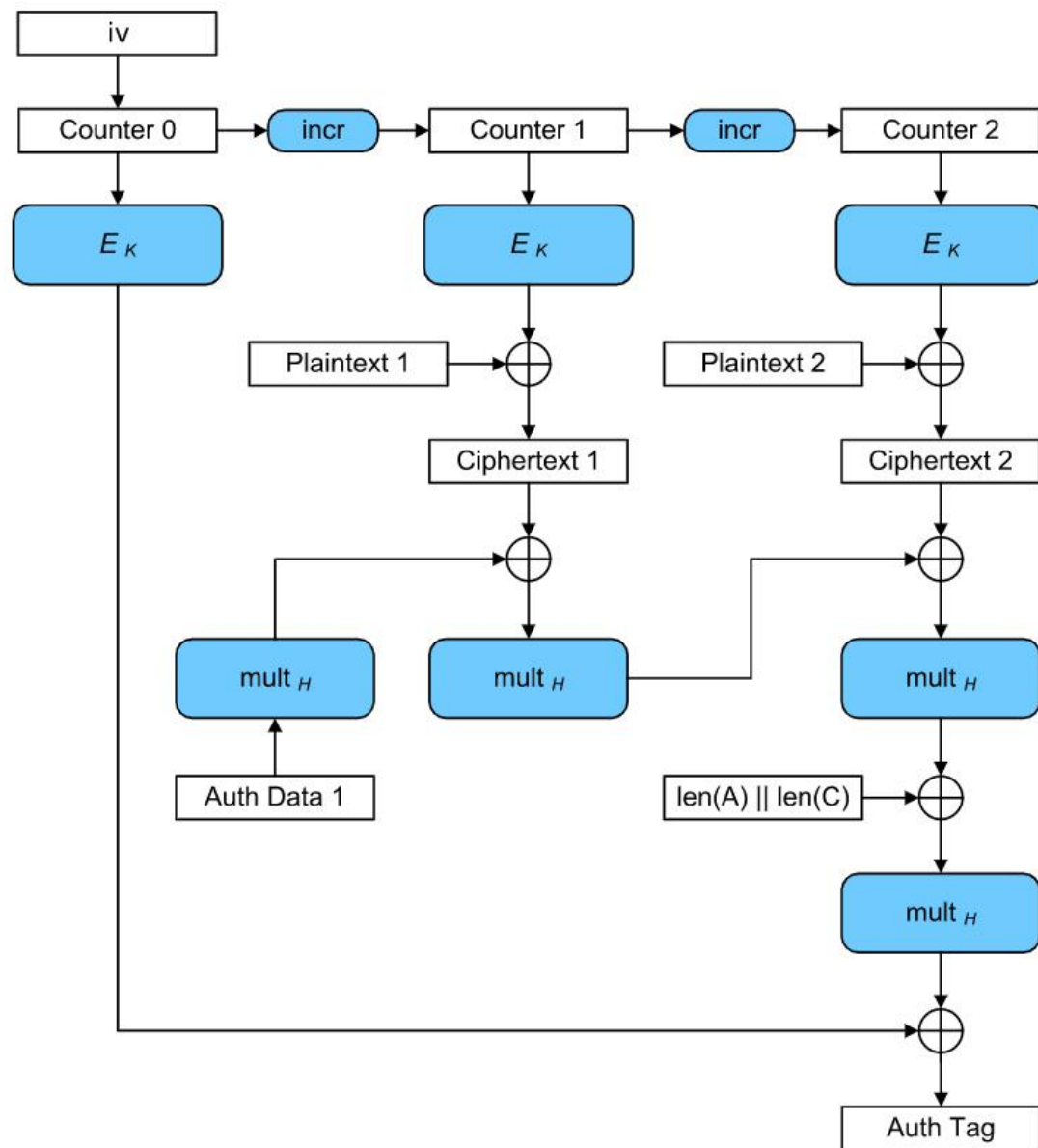


验证完整性：MAC (Message Authentication Code)



GCM

- **Galois/Counter Mode**
 - CTR+GMAC



第 4 课 详解 AES 对称加密算法

AES (Advanced Encryption Standard) 加密算法

- 为比利时密码学家 Joan Daemen 和 Vincent Rijmen 所设计, 又称 Rijndael 加密算法
- 常用填充算法: PKCS7
- 常用分组工作模式: GCM

AES 的三种密钥长度

- AES分组长度是 128 位 (16 字节)

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 的加密步骤

1. 把明文按照 128bit (16 字节) 拆分成若干个明文块, 每个明文块是 4×4 矩阵
2. 按照选择的填充方式来填充最后一个明文块
- 3. 每一个明文块利用 AES 加密器和密钥, 加密成密文块**
4. 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

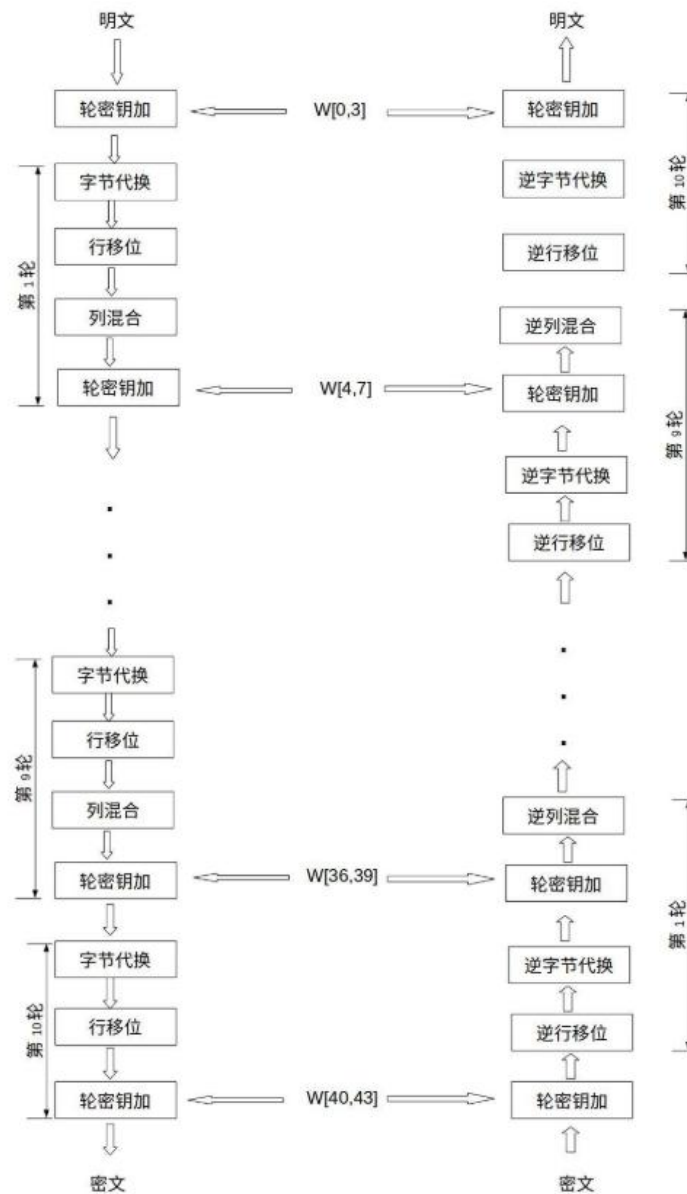
- AddRoundKey 轮密钥加

- 普通轮

- AddRoundKey 轮密钥加
 - SubBytes 字节替代
 - ShiftRows 行移位
 - MixColumns 列混合

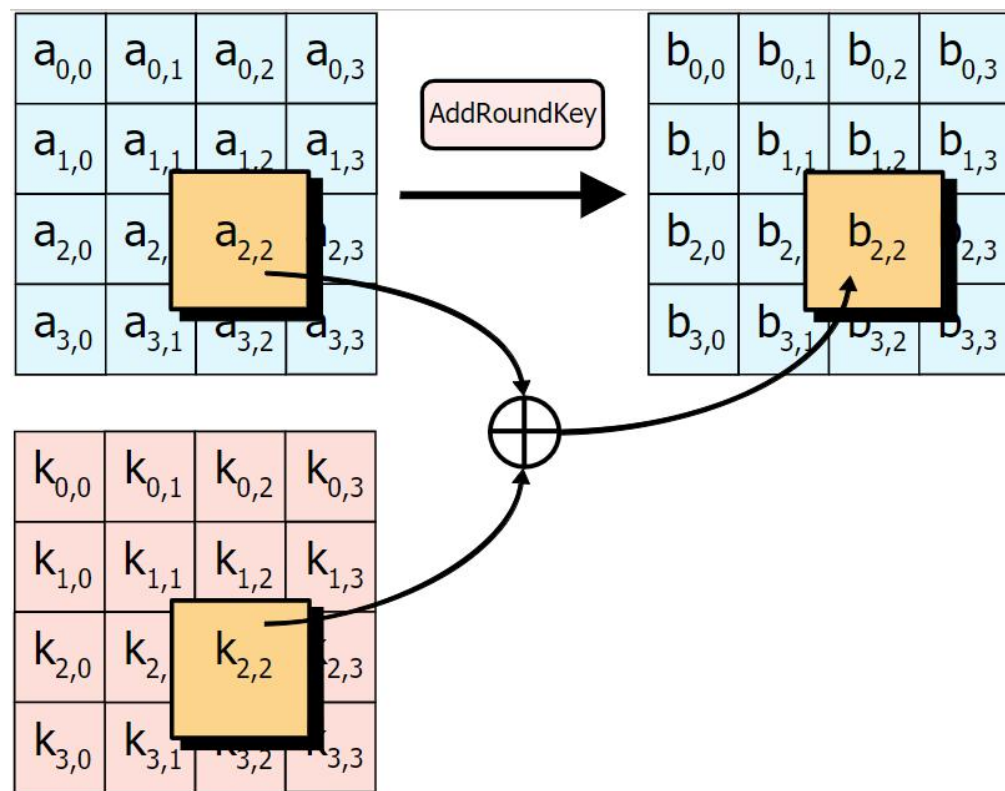
- 最终轮

- SubBytes 字节替代
 - ShiftRows 行移位
 - AddRoundKey 轮密钥加



(1) AddRoundKey 步骤

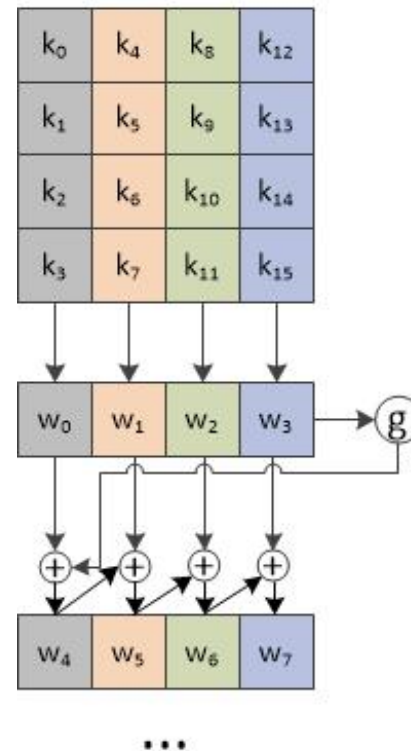
- 矩阵中的每一个字节都与该次回合密钥 (round key) 做 XOR 运算；每个子密钥由密钥生成方案产生。



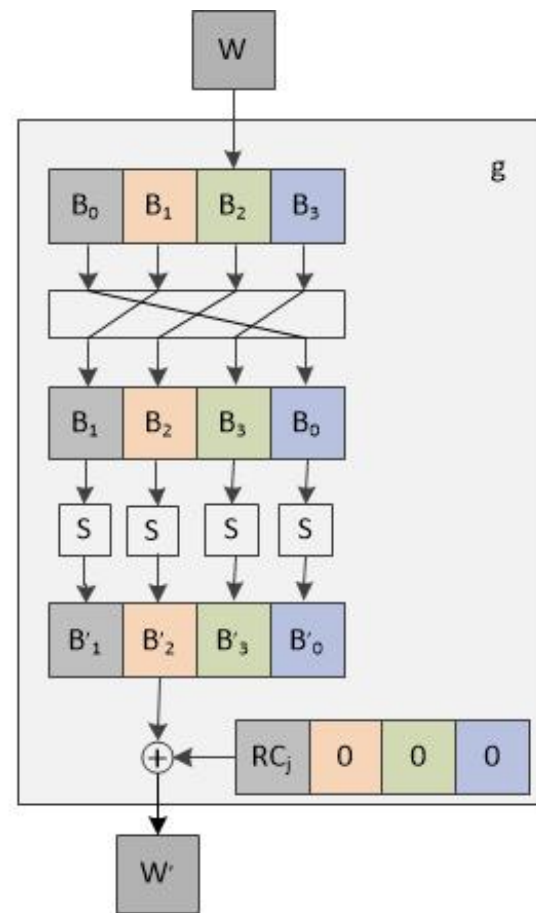
密钥扩展

- 函数 g 步骤

- a. 字循环：左移 1 个字节
- b. 使用 S 盒字节代换
- c. 同轮常量 $RC[j]$ 进行异或，其中 j 表示轮数



(a) 总体算法

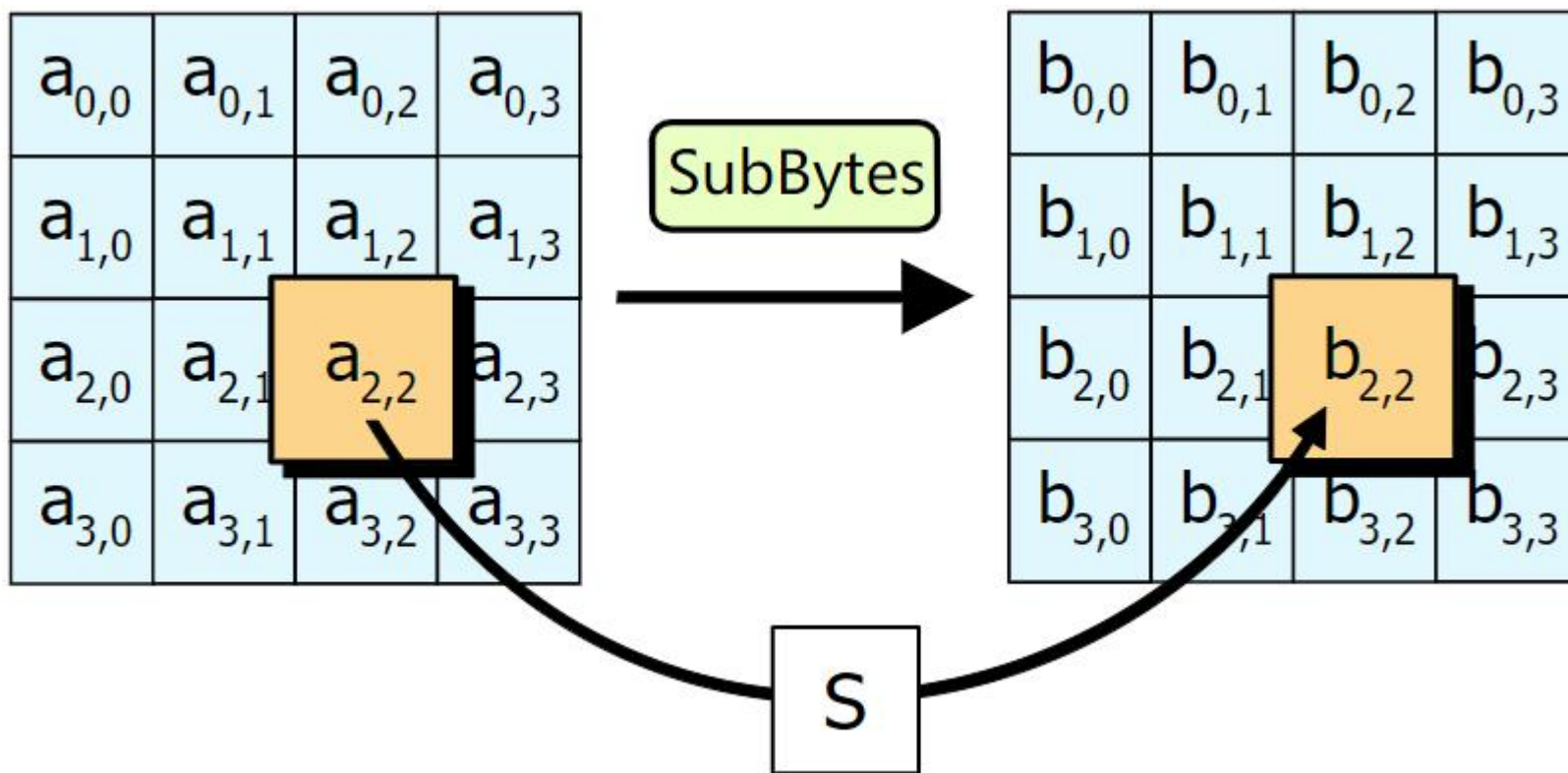


(b) 函数 g

- $RC = \{0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1B, 0x36\}$

(2) SubBytes 步骤

- 透过一个非线性的替换函数，用查找表的方式把每个字节替换成对应的字节。
 - 提供非线性变换能力，避免简单代数性质的攻击



S 盒

行/列	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0x63	0x7c	0x77	0x7b	0xf2	0x6b	0x6f	0xc5	0x30	0x01	0x67	0x2b	0xfe	0xd7	0xab	0x76
1	0xca	0x82	0xc9	0x7d	0xfa	0x59	0x47	0xf0	0xad	0xd4	0xa2	0xaf	0x9c	0xa4	0x72	0xc0
2	0xb7	0xfd	0x93	0x26	0x36	0x3f	0xf7	0xcc	0x34	0xa5	0xe5	0xf1	0x71	0xd8	0x31	0x15
3	0x04	0xc7	0x23	0xc3	0x18	0x96	0x05	0x9a	0x07	0x12	0x80	0xe2	0xeb	0x27	0xb2	0x75
4	0x09	0x83	0x2c	0x1a	0x1b	0x6e	0x5a	0xa0	0x52	0x3b	0xd6	0xb3	0x29	0xe3	0x2f	0x84
5	0x53	0xd1	0x00	0xed	0x20	0xfc	0xb1	0x5b	0x6a	0xcb	0xbe	0x39	0x4a	0x4c	0x58	0xcf
6	0xd0	0xef	0xaa	0xfb	0x43	0x4d	0x33	0x85	0x45	0xf9	0x02	0x7f	0x50	0x3c	0x9f	0xa8
7	0x51	0xa3	0x40	0x8f	0x92	0x9d	0x38	0xf5	0xbc	0xb6	0xda	0x21	0x10	0xff	0xf3	0xd2
8	0xcd	0x0c	0x13	0xec	0x5f	0x97	0x44	0x17	0xc4	0xa7	0x7e	0x3d	0x64	0x5d	0x19	0x73
9	0x60	0x81	0x4f	0xdc	0x22	0x2a	0x90	0x88	0x46	0xee	0xb8	0x14	0xde	0x5e	0x0b	0xdb
A	0xe0	0x32	0x3a	0x0a	0x49	0x06	0x24	0x5c	0xc2	0xd3	0xac	0x62	0x91	0x95	0xe4	0x79
B	0xe7	0xc8	0x37	0x6d	0x8d	0xd5	0x4e	0xa9	0x6c	0x56	0xf4	0xea	0x65	0x7a	0xae	0x08
C	0xba	0x78	0x25	0x2e	0x1c	0xa6	0xb4	0xc6	0xe8	0xdd	0x74	0x1f	0x4b	0xbd	0x8b	0x8a
D	0x70	0x3e	0xb5	0x66	0x48	0x03	0xf6	0x0e	0x61	0x35	0x57	0xb9	0x86	0xc1	0x1d	0x9e
E	0xe1	0xf8	0x98	0x11	0x69	0xd9	0x8e	0x94	0x9b	0x1e	0x87	0xe9	0xce	0x55	0x28	0xdf
F	0x8c	0xa1	0x89	0x0d	0xbf	0xe6	0x42	0x68	0x41	0x99	0x2d	0x0f	0xb0	0x54	0xbb	0x16

(3) ShiftRows 步骤

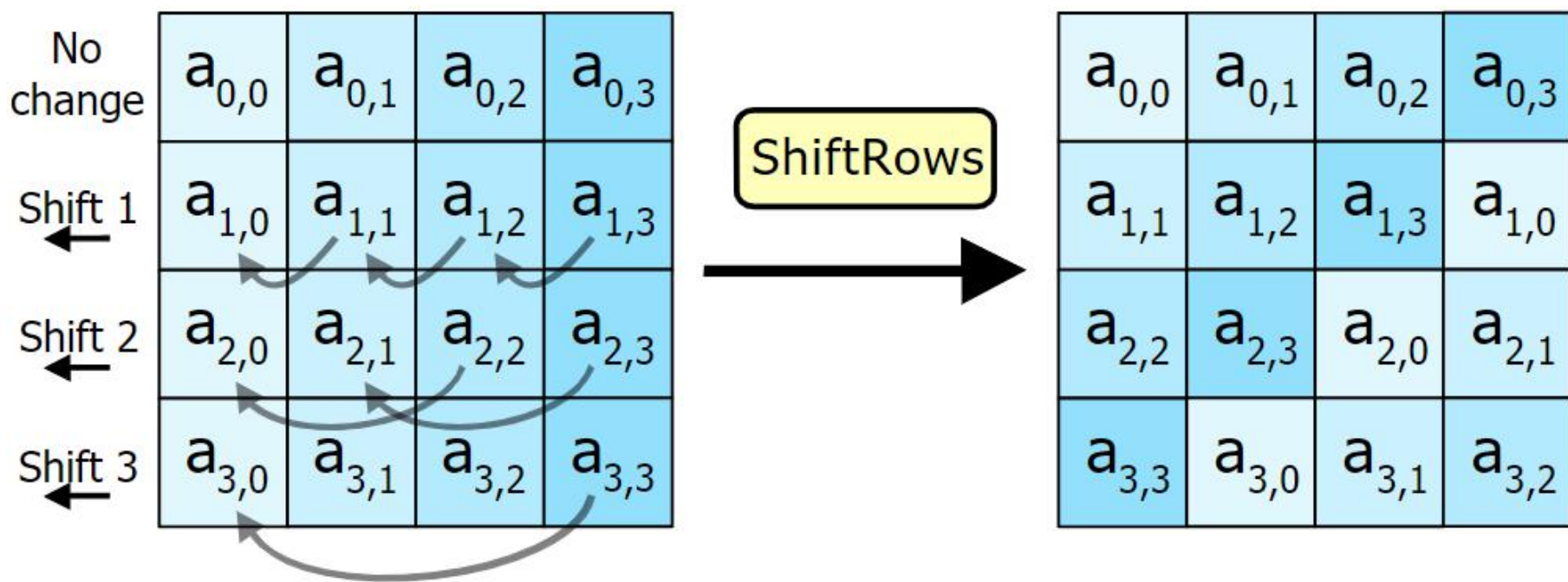
将矩阵中的每个横列进行循环式移位。

第一行不变

第二行循环左移 1 个字节

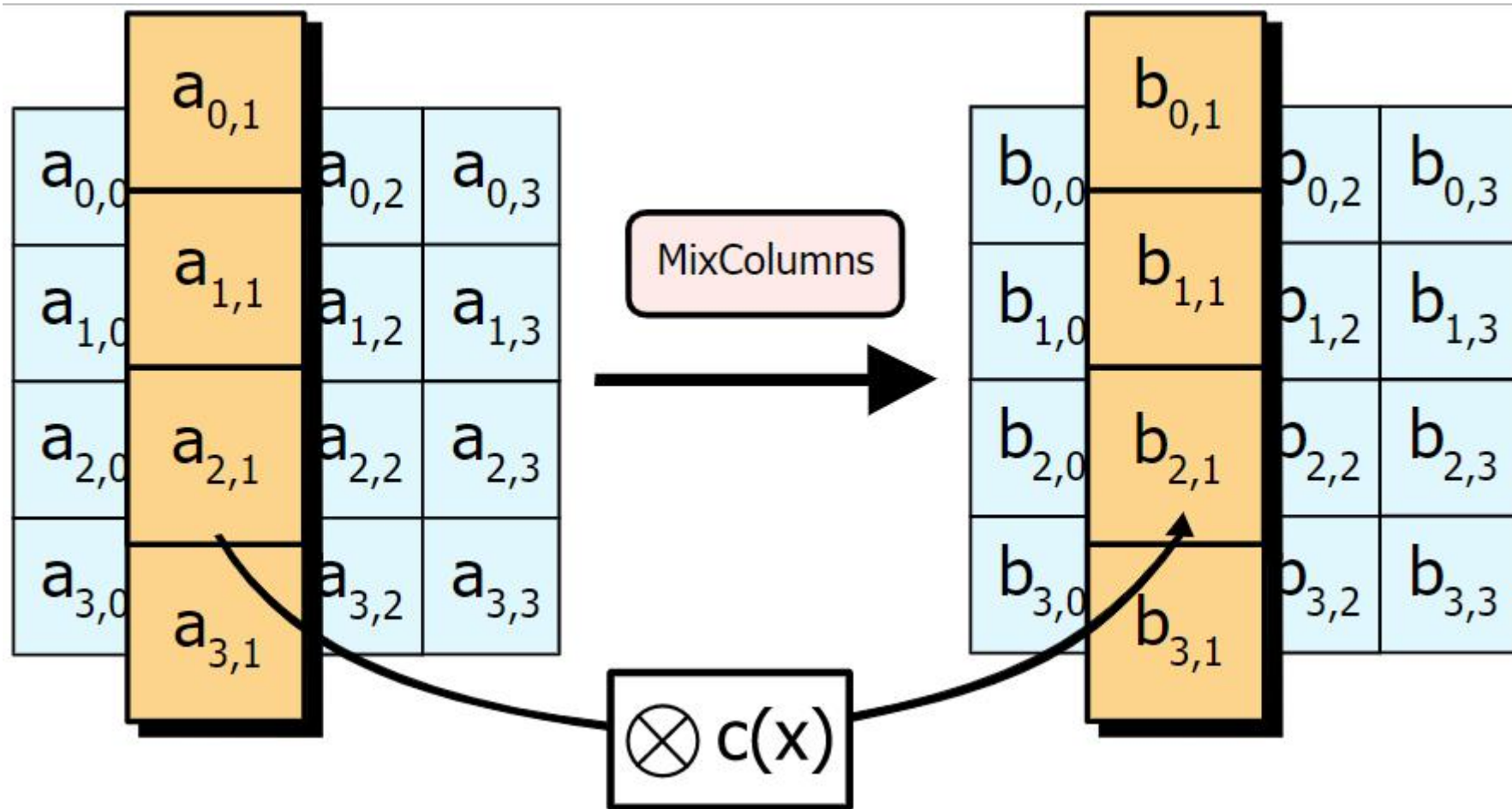
第三行循环左移 2 个字节

第四行循环左移 3 个字节



(4) MixColumns 步骤

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$



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

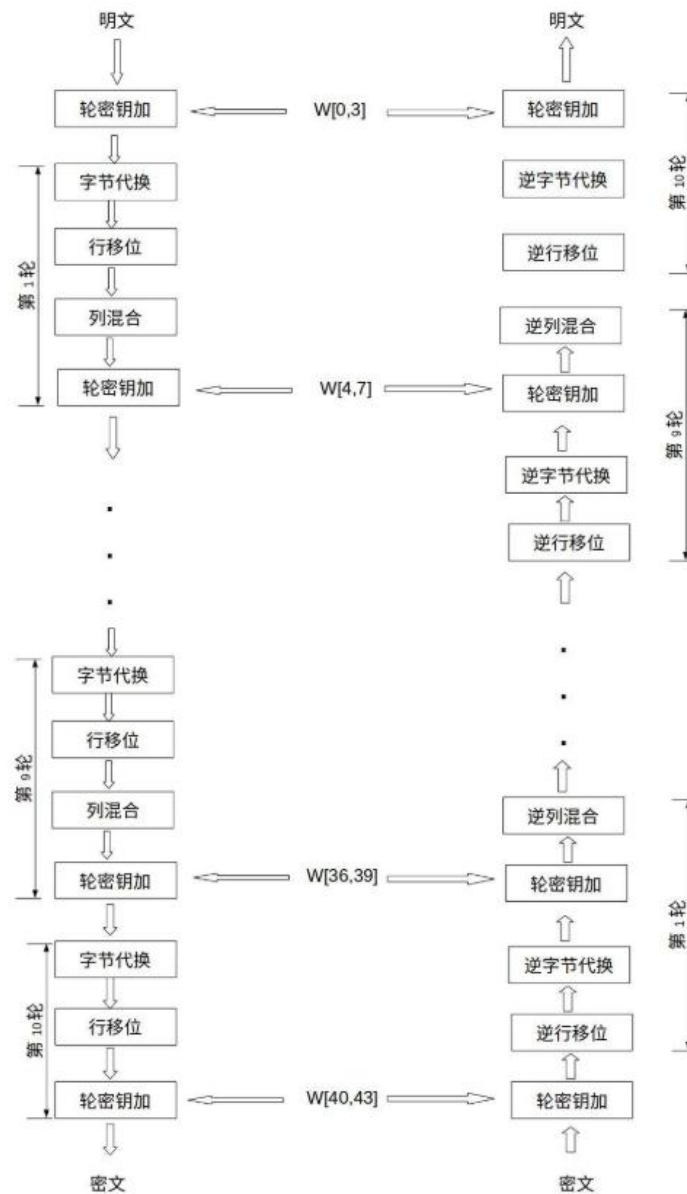
- AddRoundKey 轮密钥加

- 普通轮

- AddRoundKey 轮密钥加
 - SubBytes 字节替代
 - ShiftRows 行移位
 - MixColumns 列混合

- 最终轮

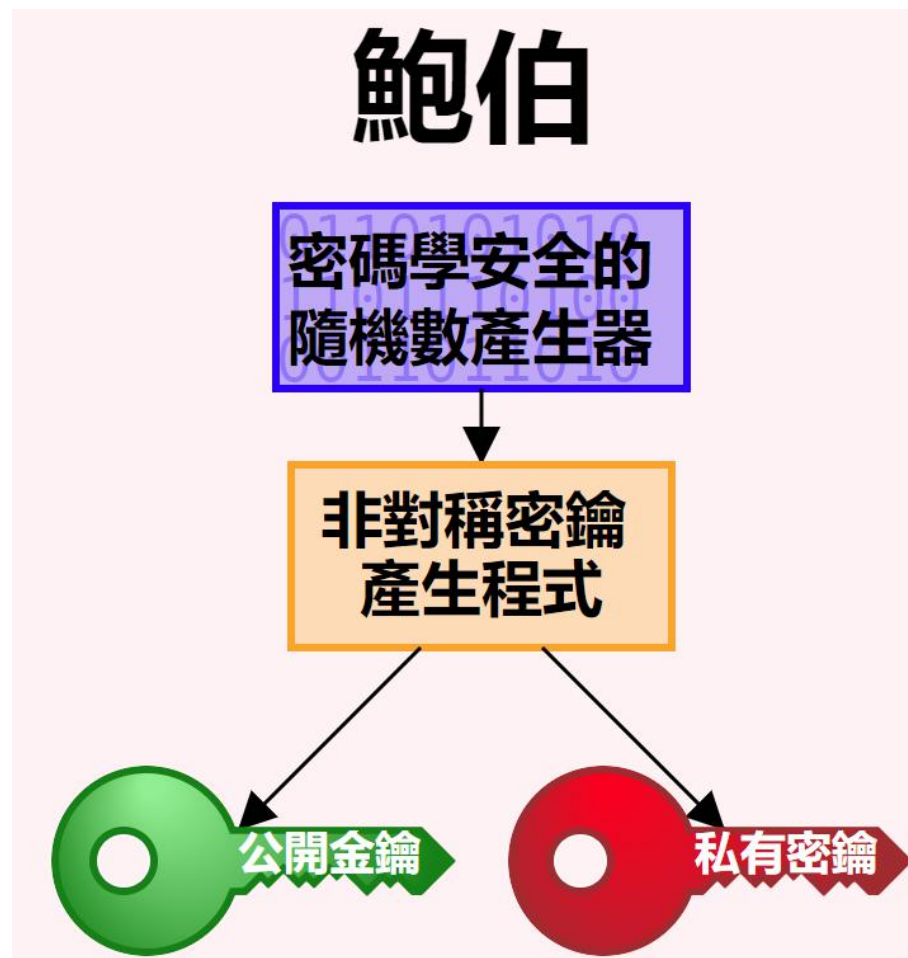
- SubBytes 字节替代
 - ShiftRows 行移位
 - AddRoundKey 轮密钥加



第 5 课 非对称密码与 RSA 算法

非对称密码

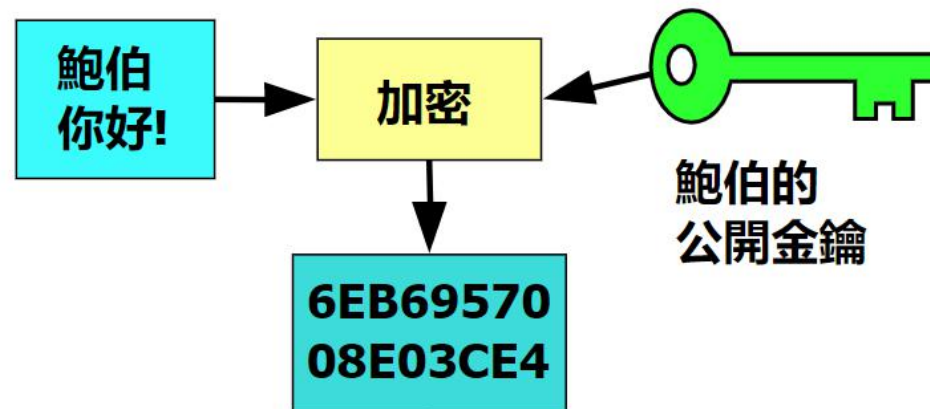
- 每个参与方都有一对密钥
 - 公钥
 - 向对方公开
 - 私钥
 - 仅自己使用



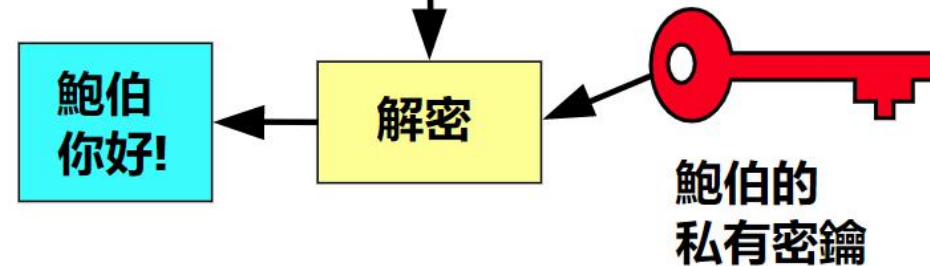
非对称加解密的过程

- 加密
 - 使用对方的公钥加密消息
- 解密
 - 使用自己的私钥解密消息

愛麗斯



鮑伯



RSA 算法

- 1977 年由罗纳德·李维斯特 (Ron Rivest)、阿迪·萨莫尔 (Adi Shamir) 和伦纳德·阿德曼 (Leonard Adleman) 一起提出, 因此名为 RSA 算法

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 (明文小于 n)
3. 计算 n 的欧拉函数 $v = \varphi(n)$
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: $c \equiv m^k \pmod{n}$
 - m 是明文, c 是密文
- 解密: $m \equiv c^d \pmod{n}$
- 举例: 对明文数字 123 加解密
 - 公钥 (3,319) 加密
 - $123^3 \pmod{319} = 140$
 - 对140密文用私钥 (187,319) 解密
 - $140^{187} \pmod{319} = 123$
 - 私钥 (187,319) 加密
 - $123^{187} \pmod{319} = 161$
 - 公钥 (3,319) 解密
 - $161^3 \pmod{319} = 123$

RSA algorithm

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

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

第6课 基于 openssl 实战验证 RSA

使用 openssl 基于 RSA 算法生成公私钥 (1)

- 生成私钥（公私钥格式参见 RFC3447）
 - openssl genrsa -out private.pem

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEApcUpkkr0JWd23UT3gJbCYaMBRmEm0VE8HWQxC3G3VvaF0FCgQ
WpMj8Z3ZctmPQgkgIbY/fj5708zDx0P8bJZ0UFAePo3qJ+/dNkJE/P1LTLEk+H07
3Z5nx0pgX2v+cxLlMqCpwaQlXwgyacFN3m9ek6eq4AWhANvSxfLBIW6yDbSr0dGM
WIcrFXYZlWIn3fTeIW2pkfDEBGyuGF0qqt2emIPyVa0z1LE9HzSURk47UR7b8Xlo
7x74zNx9bsSIjz+9z55M7n+d0erMP3eY+zVeuKlX8R7XrHVgctK3x0HQXCMBQ0bY
enmXM80wz2b7FxKQHk4mAXmGbKBsq1cnhfkmLQIDAQABAoIBAFaSdxN+0q2FnS5k
gnkiilC+jVJV6TMSH1j624SwtSt3mNu0qd1rg2iajSZC0ojMkVLWcTRfNdW7UIFW
kTqg06ZkKx9o79eybaSBXStlWEKRp2MbgeiH9Eoq0xtgt0nyTEngbMgBj+hIRXZ7
iltHt8T+iGPw9BKk3cyjFPNSE7yf9aAyYLt7D0FVDfUyAG40tvDoI3thmrZkIsR9
kc4Tc4jawwpmafMs07R4V9GUfeFbNV+0Y0arJY5tAIZEcPcu81YvVeARjDoEtaG4
hn7l6NbimmZ1bdf8P1k1L6jptAq0YkluCminQSYEGHhDkkY0NJTshuTNjvbwTrpQ
klgKK+ECgYEA2Nly83F0Pmr7Ds17wQYcdn/IA2sy9WT/bz1L58k2obV0aSdaAWN
ashjFRsd8uH0a+SegyYkECJZnVksptQA49vo8c3WU5wXzt1XXNIps9XVTfgnTU6B
aHElQnYzo9u96SSHnmrZuPDVtIpFtpRTVgayn4YvNG0dxZqa4TB05bkCgYEAwc2W
hPN1fD+p26d5sb3sDX/G/U4kev9zdVU5wcT5GtJJZoLHZxNhQRLK7CeHuJBXHLhNB
gzTSFV3Qgkf2ys7SMWAYXQkRp8x2XH/roj0iNe09ACBD64Q8YbqlmBoA+h74T95
IITTKJbj9CHYinNPAfk0JicAu5kEYms5Tbp7vhUCgYBaHjtaUZJ2CvL0MRW5oXed
DPkt6pVQhzzf1pnk6a2qQngnnnJ7VzXTLRx5fj7RX3a+jVQqW6G6sjA7BQZCo6g
JhvIvLvTHnKKyVZxPuyUUVEJyhVTvUhuB3DsEcLe3Qsu5r7zage6JWc9Vacp/rYw
cDI3uYbPKIQ/yEWNSR2dCQKBgCgqwwqVJwF0dHD5GgjTsyvtEsIHQxpW/YgQ/OKD
P5nmhPsAQYFB0Akc/hUDRYcnGFwdU50tC+mvwvptjeHk0b24C/SkX4tmnhV30c+R
YgpJPD6zqaATjSw4Mf//S7qK7U13CVTLERsZ6VT8+tNfKL3g94T8ynMX0MhpLrxE
bUctAoGARQSh9jrREtWF96wHV/4b4Wj9LNQVeUhmElYxXrfPBXuNSA6FQ2yHaV9r
g6wd3X6AMXFtFM01JTER9TvuDS34Dtau4QY4zNLH3gjYHEAg3From0D5WQRiuV4j
i2BjmUnm4crP4jD6DcpSdwgnKKs+RwxHLPUn8QrHHJJKhSNqoWU=
-----END RSA PRIVATE KEY-----
```


使用 openssl 基于 RSA 算法生成公私钥 (2)

- 从私钥中提取出公钥
 - `openssl rsa -in private.pem -pubout -out public.pem`

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApcUpkkr0JWd23UT3gJbC
YaMBrEm0VE8HWQxC3G3VvaF0FCgQWpMj8Z3ZctmPQgkgIbY/fj5708zDx0P8bJZ0
UFAePo3qJ+/dNkJE/P1LTLEk+H073Z5nx0pgX2v+cxLMqCpwaQlXwgyacFN3m9e
k6eq4AWhANvSxflBIW6yDbSr0dGMWicrFxZYlWIn3fTeIW2pkfDEBGyuGF0qqt2e
mIPyVa0z1LE9HzSURk47UR7b8Xlo7x74zNx9bsSIjz+9z55M7n+d0erMP3eY+zVe
uKlX8R7XrHVgctK3x0HQXCMBQ0bYenmXM80wz2b7FxKQHk4mAXmGbKBsq1cnhfkm
LQIDAQAB
-----END PUBLIC KEY-----
```

使用 openssl 基于 RSA 算法生成公私钥 (3)

- 查看 ASN.1 格式的私钥
 - openssl asn1parse -i -in private.pem

```

RSAPrivateKey ::= SEQUENCE {
    version Version,
    modulus INTEGER, -- n
    publicExponent INTEGER, -- k
    privateExponent INTEGER, -- d
    prime1 INTEGER, -- p
    prime2 INTEGER, -- q
    exponent1 INTEGER, -- d mod (p-1)
    exponent2 INTEGER, -- d mod (q-1)
    coefficient INTEGER, -- (inverse of q) mod p
    otherPrimeInfos OtherPrimeInfos OPTIONAL
}
    
```

```

0:d=0  hl=4  l=1186 cons: SEQUENCE
4:d=1  hl=2  l=  1 prim:  INTEGER           :00
7:d=1  hl=4  l= 257 prim:  INTEGER           :A425299
67450501E3E8DEA27EFDD364244FCFD4B4CB124F87D3BDD9E67C74A6
1F0C4046CAE185D2AAADD9E9883F255A3B394B13D1F3494464E3B511
0CF66FB1712901E4E260179866CA06CAB572785F9262D
268:d=1  hl=2  l=  3 prim:  INTEGER           :010001
273:d=1  hl=4  l= 256 prim:  INTEGER           :5692771
03BA6642B1F68EFD7B26DA4815D2B65584291A7631B81E887F44A2AD
B619AB66422C47D91CE137388DAC30A6669F32CD3B47857D1947DE15
078439246343494EC86E4CD8EF7704EBA5092580A2BE1
533:d=1  hl=3  l= 129 prim:  INTEGER           :D8D2F2F
400E3DBE8F1CDD6539C17CEDD575CD229B3D5D54DF8274D4E8168712
665:d=1  hl=3  l= 129 prim:  INTEGER           :C1CD968
ED23160185D0911A7CC765C7FEBA233A235ED3D01C043EB843C61BAA
797:d=1  hl=3  l= 128 prim:  INTEGER           :5A1E3B5
A8EA0261BC8BCBBD31E728AC956713EEC94515109CA1553BD486E077
928:d=1  hl=3  l= 128 prim:  INTEGER           :282AC30
4D1BDB80BF4A45F8B669E1577D1CF91620A493C3EB3A9A0138D2C383
1059:d=1 hl=3  l= 128 prim:  INTEGER           :4504A1F
DF80ED6AEE10638CCD2C7DE08D81C4020DC5AE898E0F9590462B95E2
    
```

使用 openssl 基于 RSA 算法生成公私钥 (4)

- 查看 ASN.1 格式的公钥

- openssl asn1parse -i -in public.pem (X.590)

```
0:d=0  hl=4 l= 290 cons: SEQUENCE
4:d=1  hl=2 l=  13 cons: SEQUENCE
6:d=2  hl=2 l=   9 prim: OBJECT           :rsaEncryption
17:d=2  hl=2 l=   0 prim: NULL
19:d=1  hl=4 l= 271 prim: BIT STRING
```

- openssl asn1parse -i -in public.pem -strparse 19

```
0:d=0  hl=4 l= 266 cons: SEQUENCE
4:d=1  hl=4 l= 257 prim: INTEGER           :A42529924ACF
67450501E3E8DEA27EFDD364244FCFD4B4CB124F87D3BDD9E67C74A605F6E
1F0C4046CAE185D2AAADD9E9883F255A3B394B13D1F3494464E3B511EDBF
0CF66FB1712901E4E260179866CA06CAB572785F9262D
265:d=1  hl=2 l=   3 prim: INTEGER           :010001
```

```
RSAPublicKey ::= SEQUENCE {
    modulus INTEGER, -- n
    publicExponent INTEGER -- k
}
```

使用 RSA 公钥加解密

- 加密文件

- openssl rsautl -encrypt -in hello.txt -inkey public.pem -pubin -out hello.en

- 解密文件

- openssl rsautl -decrypt -in hello.en -inkey private.pem -out hello.de

第 7 课 非对称密码应用：PKI 证书体系

非对称密码应用：数字签名

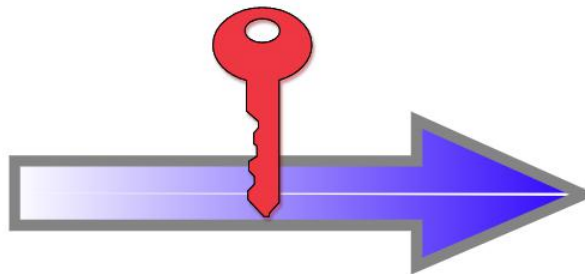
- 基于私钥加密，只能使用公钥解密：起到身份认证的使用
- 公钥的管理：Public Key Infrastructure (PKI) 公钥基础设施
 - 由 Certificate Authority (CA) 数字证书认证机构将用户个人身份与公开密钥关联在一起
 - 公钥数字证书组成
 - CA 信息、公钥用户信息、公钥、权威机构的签字、有效期
 - PKI 用户
 - 向 CA 注册公钥的用户
 - 希望使用已注册公钥的用户

签发证书流程

鮑伯的身份資料 及公開金鑰



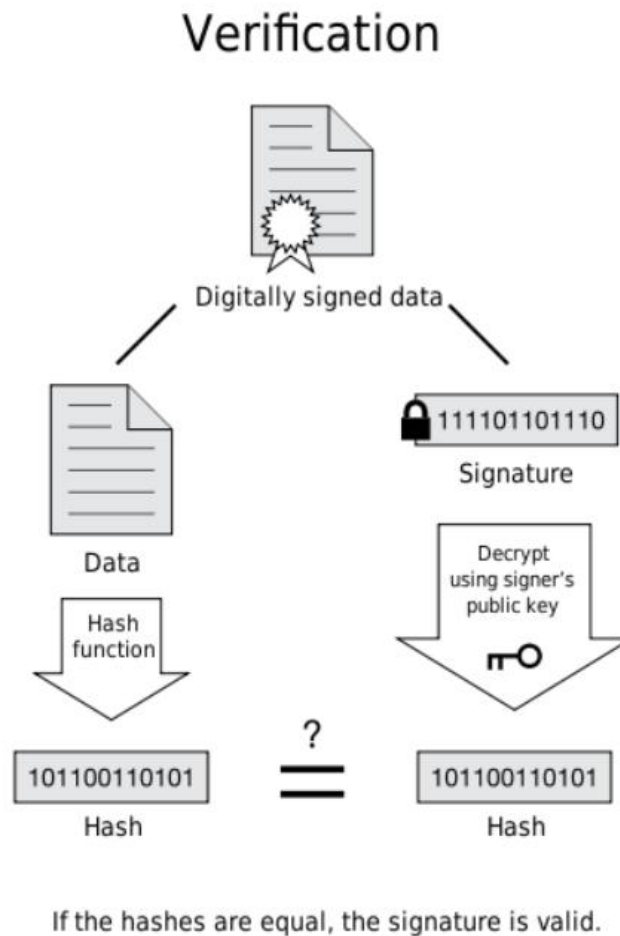
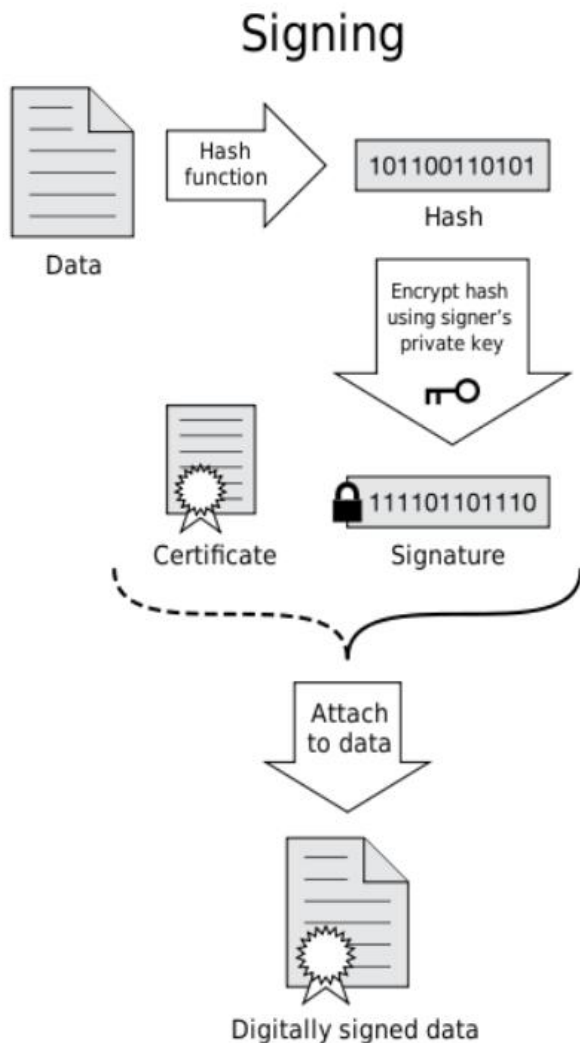
認證機構核實鮑伯身份後
使用認證機構的私鑰加密



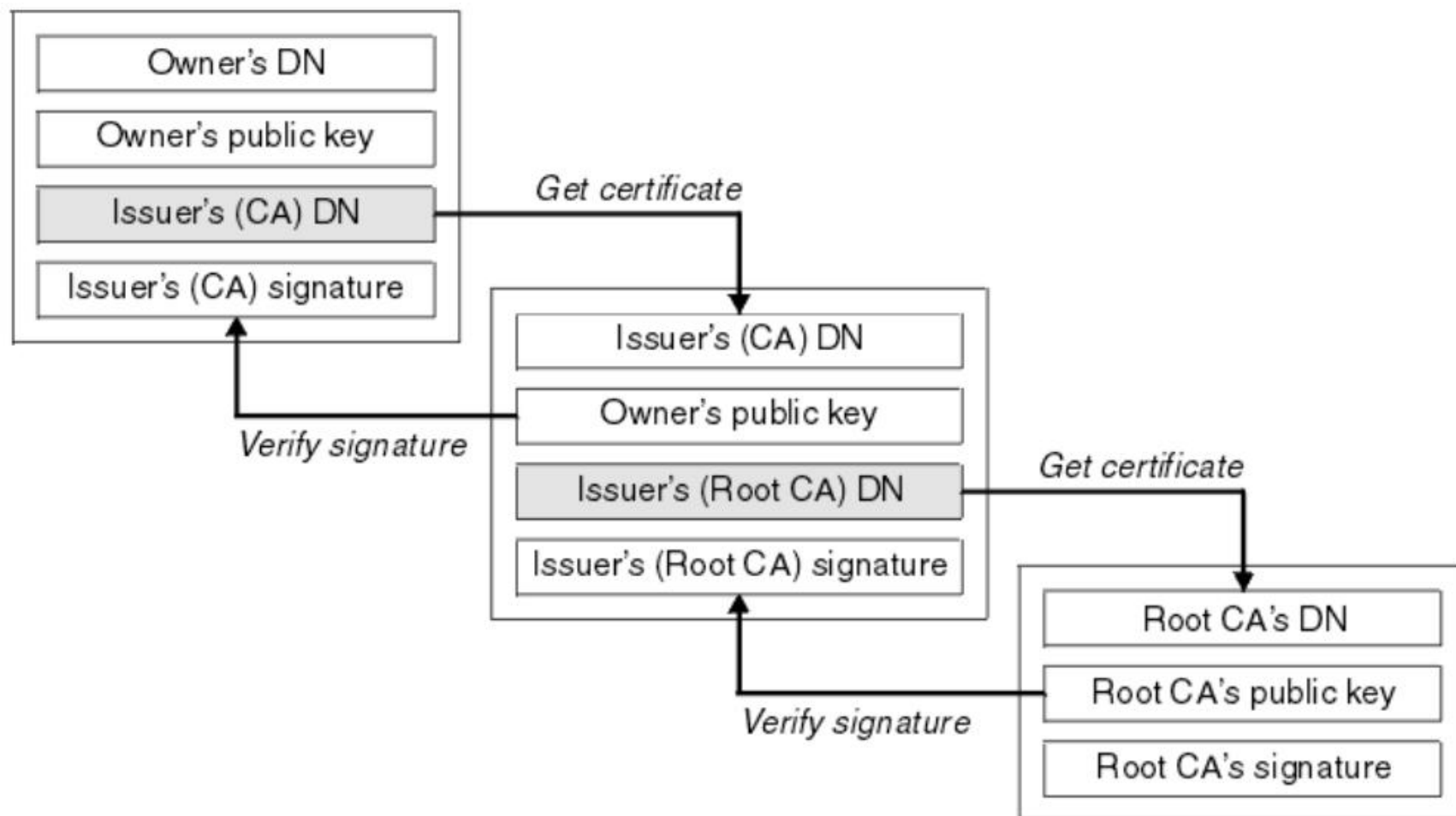
鮑伯的數位證書



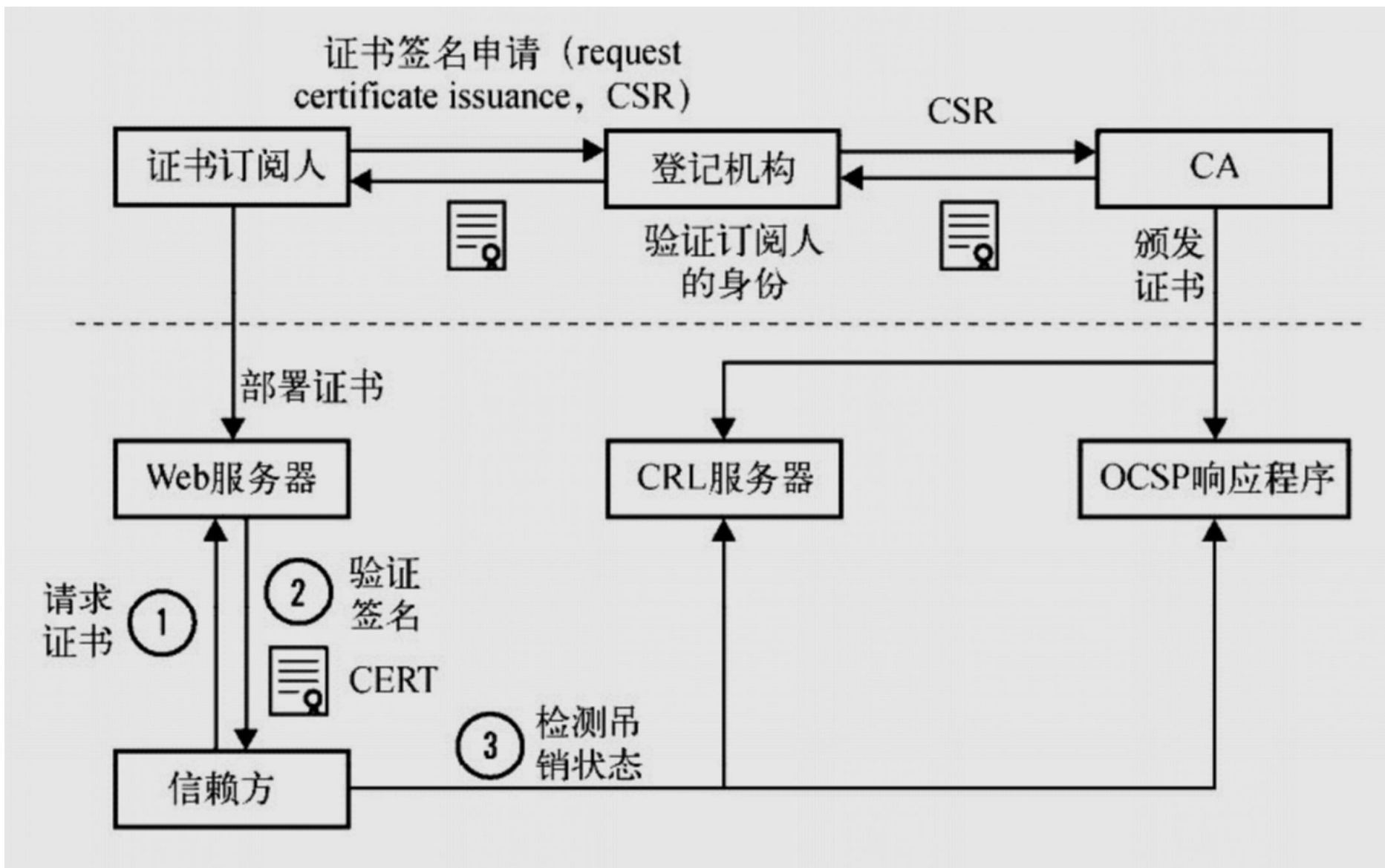
签名与验签流程



证书信任链



PKI 公钥基础设施



证书类型

域名验证 (domain validated, DV) 证书



①  https://www.taohui.pub

组织验证 (organization validated, OV) 证书



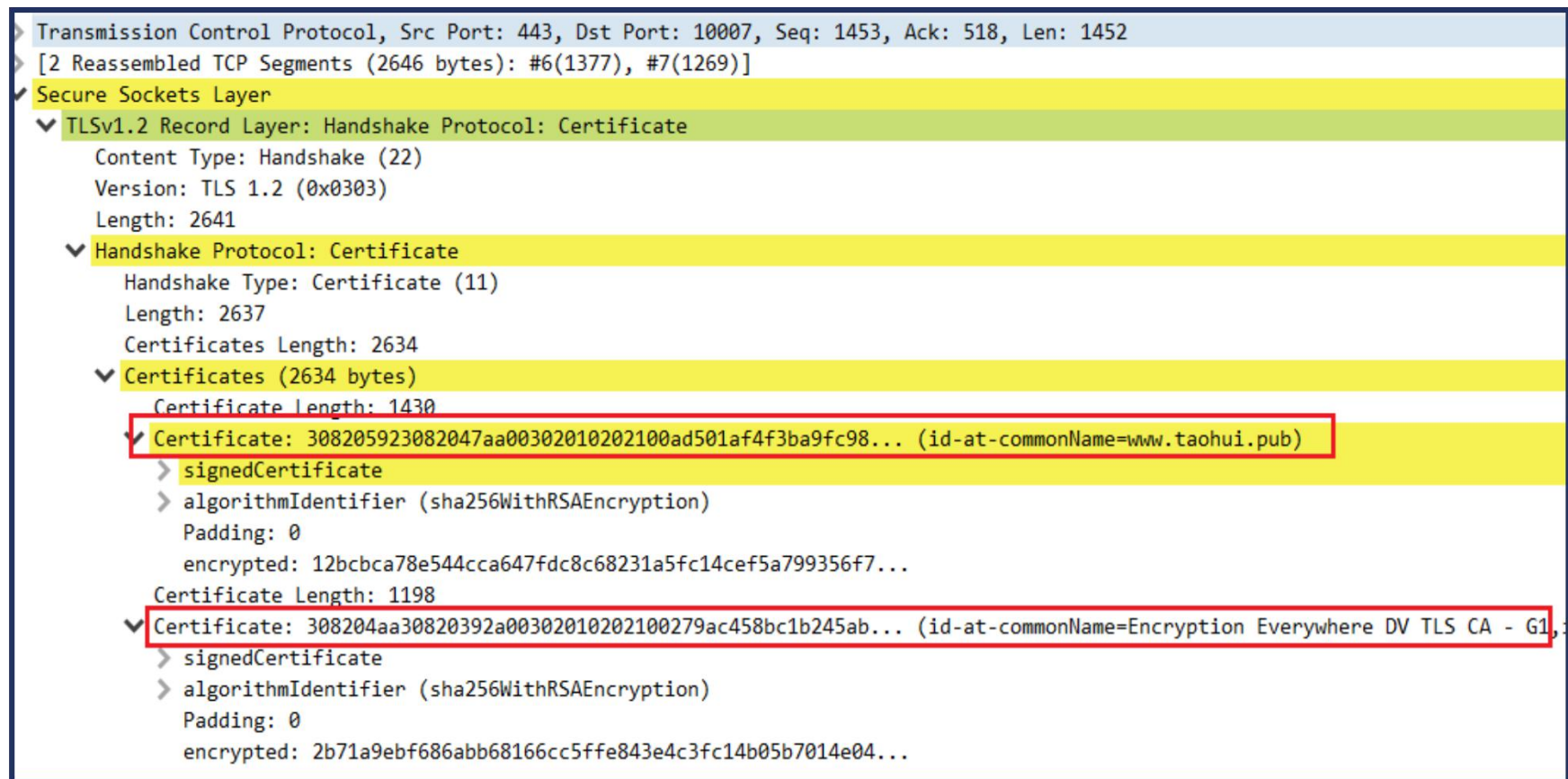
①  https://www.jd.com

扩展验证 (extended validation, EV) 证书



①  浙江名友金融信息服务有限公司 (CN) | https://www.mingyou.com

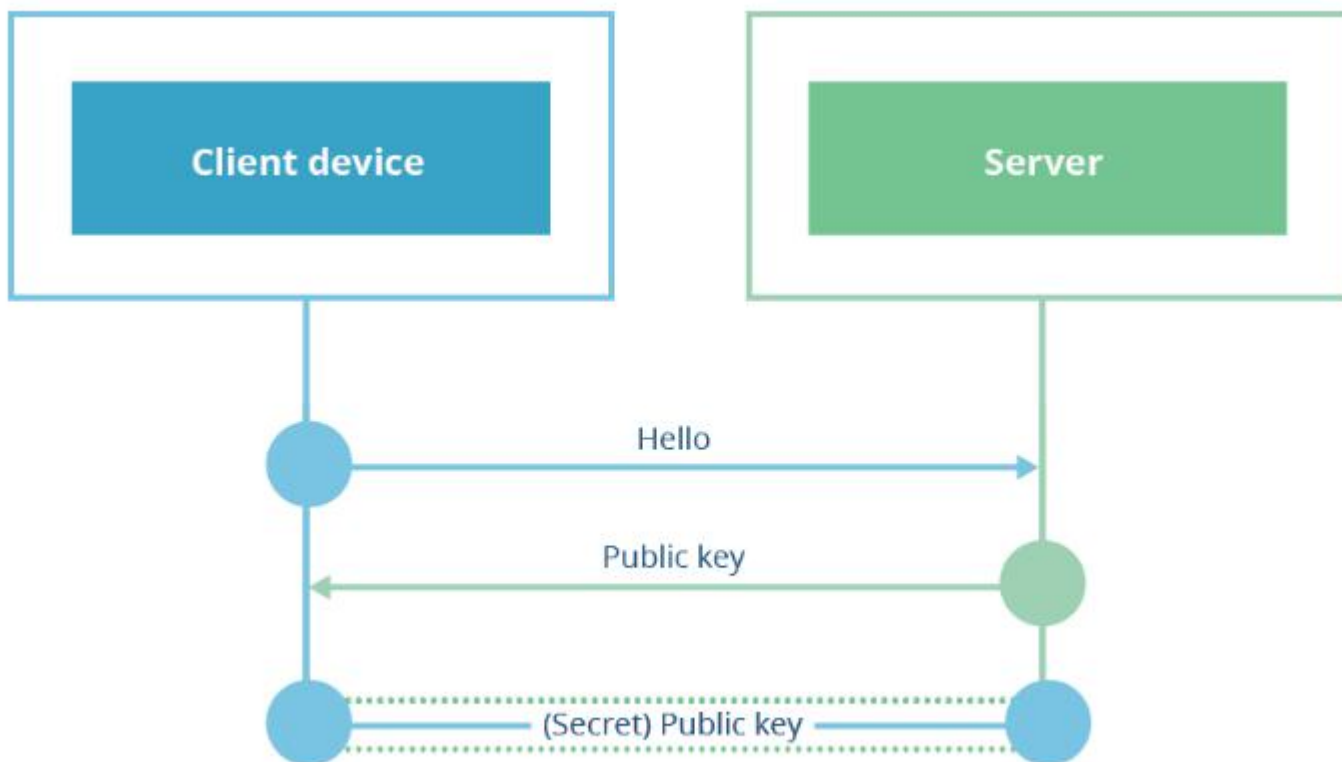
验证证书链



第 8 课 非对称密码应用：DH 密钥交换协议

RSA 密钥交换

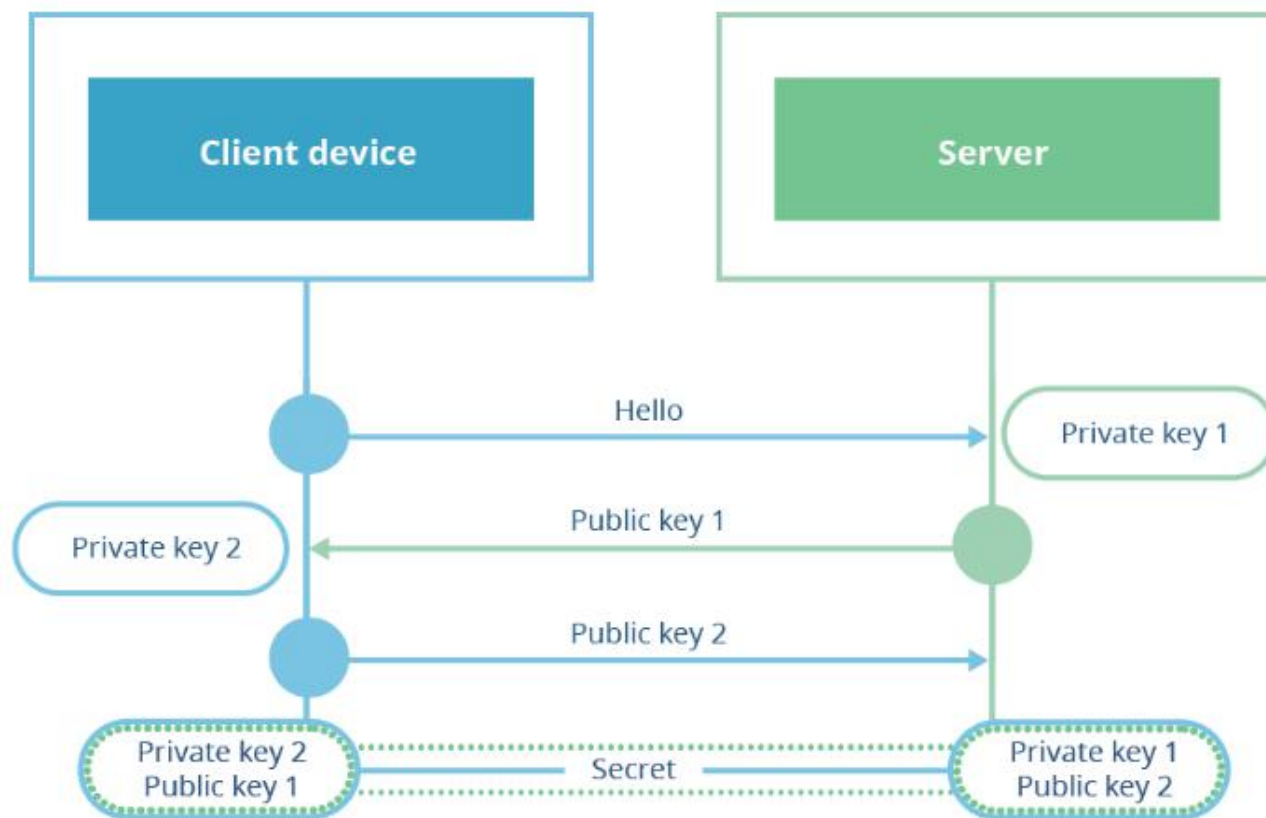
- 由客户端生成对称加密的密钥



- 问题：没有前向保密性

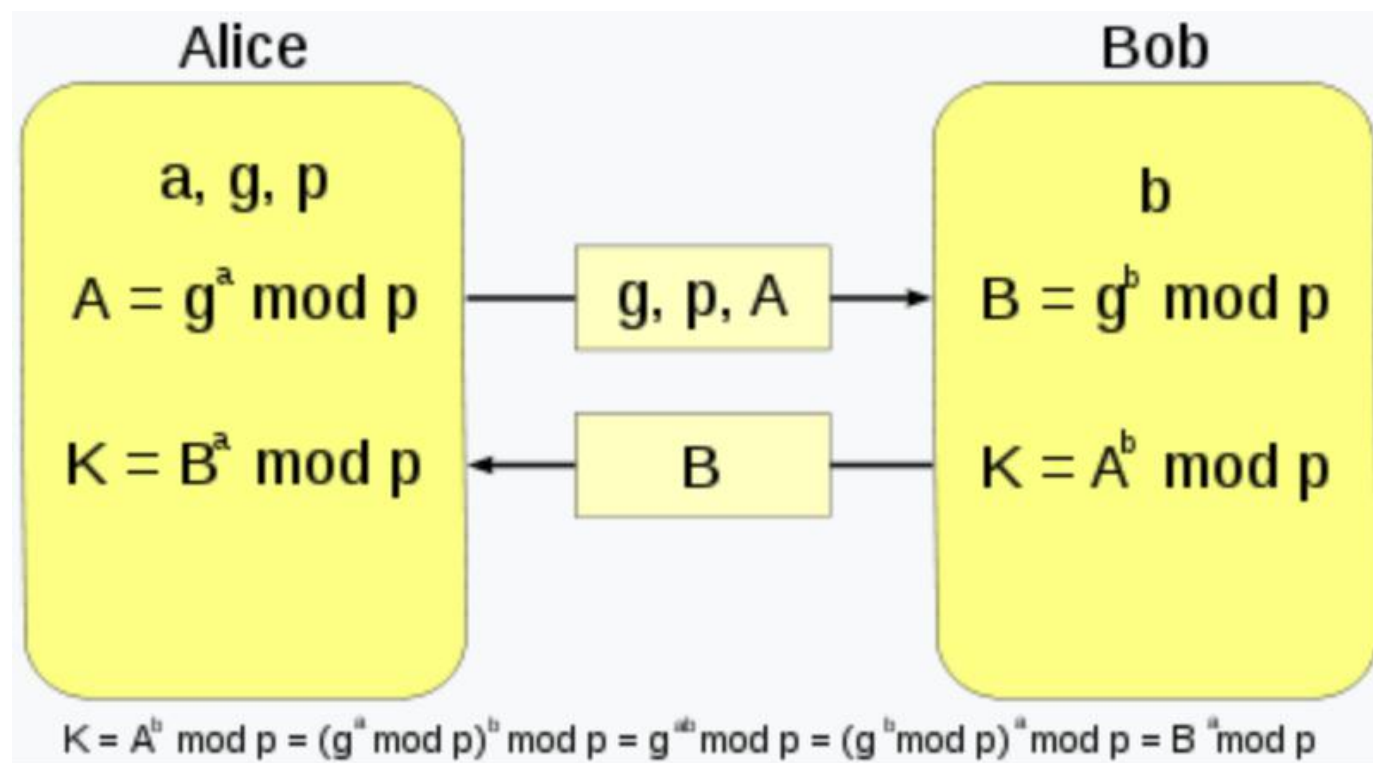
DH 密钥交换

- 1976 年由 Bailey Whitfield **Diffie** 和 Martin Edward **Hellman** 首次发表，故称为 **Diffie-Hellman** key exchange，简称 **DH**
- 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 密钥交换协议举例 (1)

- g 、 p 、 A 、 B 公开
- a, b 保密
- 生成共同密钥 K



DH 密钥交换协议举例 (2)

- 协定使用 $p=23$ 以及 base $g=5$.
- 爱丽丝选择一个秘密整数 $a=6$, 计算 $A = g^a \bmod p$ 并发送给鲍伯。
 - $A = 5^6 \bmod 23 = 8$.
- 鲍伯选择一个秘密整数 $b=15$, 计算 $B = g^b \bmod p$ 并发送给爱丽丝。
 - $B = 5^{15} \bmod 23 = 19$.
- 爱丽丝计算 $s = B^a \bmod p$
 - $19^6 \bmod 23 = 2$.
- 鲍伯计算 $s = A^b \bmod p$
 - $8^{15} \bmod 23 = 2$.

爱丽丝			鲍伯		
秘密	非秘密	计算	计算	非秘密	秘密
	p, g			p, g	
a					b
		$g^a \bmod p$...	
	...		$g^b \bmod p$		
	$(g^b \bmod p)^a \bmod p$			$(g^a \bmod p)^b \bmod p$	

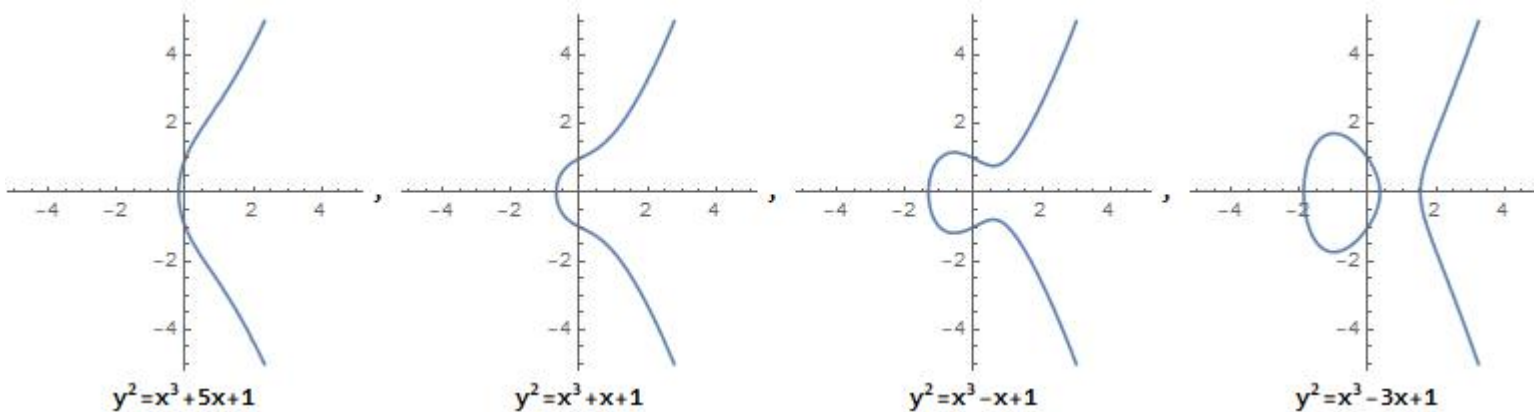
DH 密钥交换协议的问题

- 中间人伪造攻击
 - 向 Alice 假装自己是 Bob, 进行一次 DH 密钥交换
 - 向 Bob 假装自己是 Alice, 进行一次 DH 密钥交换
- 解决中间人伪造攻击
 - 身份验证

第 9 课 ECC 椭圆曲线的原理

ECC椭圆曲线的定义

- 椭圆曲线的表达式: $y^2 = x^3 + ax + b, 4a^3 + 27b^2 \neq 0$
- 例如:
 - 始终关于 X 轴对称 (y 平方的存在)



ECC 曲线的特性：+运算

- $P+Q=R$

- +运算的几何意义：R 为 P、Q 连续与曲线交点在 X 轴上的镜像

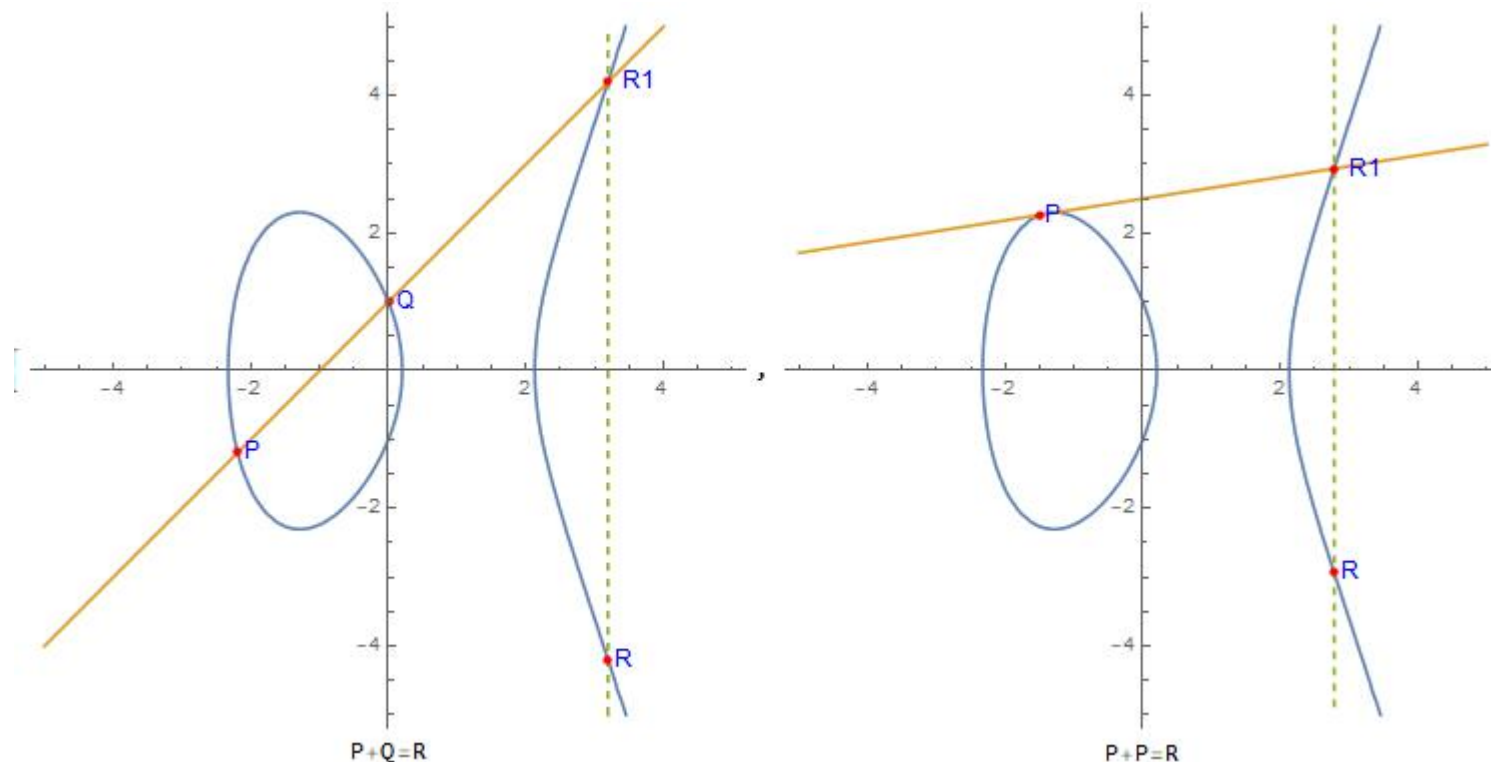
- $P+P=R$

- +运算满足交换律

$$a + b = b + a$$

- +运算满足结合律

$$(a + b) + c = a + (b + c)$$



+运算的代数计算方法

- 先计算出斜率 m ，再计算出 R 点的坐标

$$x_R = m^2 - x_P - y_P$$

$$y_R = y_P + m(x_R - x_P)$$

$$m = \begin{cases} \frac{y_P - y_Q}{x_P - x_Q} & x_P \neq x_Q \\ \frac{3x_P^2 + a}{2y_P} & P = Q \end{cases}$$

ECC+运算举例

- 设曲线: $y^2 = x^3 - 7x + 10$
- 设 $P=(1,2)$, $Q=(3,4)$, 计算出 $R(-3,-2)$
 - P 在曲线上, 因为 $2^2=4=1^3-7*1+10$
 - Q 在曲线上, 因为 $4^2=16=3^3-3*7+10=27-21+10$
 - R 在曲线上, 因为 $-2^2=4=-3^3-7*(-3)+10=-27+21+10$

$$m = \frac{y_P - y_Q}{x_P - x_Q} = \frac{2-4}{1-3} = 1$$

$$x_R = m^2 - x_P - x_Q = 1^2 - 1 - 3 = -3$$

$$y_R = y_P + m(x_R - x_P) = 2 + 1 \cdot (-3 - 1) = -2$$

$$= y_Q + m(x_R - x_Q) = 4 + 1 \cdot (-3 - 3) = -2$$

$$x_R = m^2 - x_P - y_P$$

$$y_R = y_P + m(x_R - x_P)$$

$$m = \begin{cases} \frac{y_P - y_Q}{x_P - x_Q} & x_P \neq x_Q \\ \frac{3x_P^2 + a}{2y_P} & P = Q \end{cases}$$

ECC 的关键原理

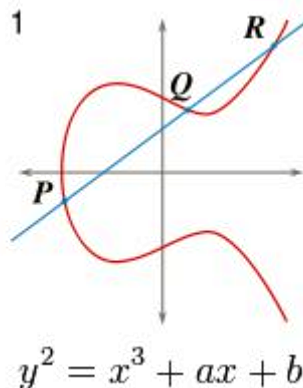
- $Q = K \cdot P$
 - 已知 K 与 P , 正向运算快速
 - 已知 Q 与 P , 计算 K 的逆向运算非常困难

$$Q = \underbrace{P + \overbrace{P + \dots + P}^K}$$

第 10 课 DH 协议升级：基于椭圆曲线的 ECDH 协议

ECDH 密钥交换协议

- DH 密钥交换协议使用椭圆曲线后的变种，称为 Elliptic Curve Diffie–Hellman key Exchange，缩写为 ECDH，优点是比 DH 计算速度快、同等安全条件下密钥更短
- ECC (Elliptic Curve Cryptography)：椭圆曲线密码学
- 魏尔斯特拉斯椭圆函数 (Weierstrass 's elliptic functions)： $y^2 = x^3 + ax + b$



ECC 的关键原理

- $Q = K \cdot P$
 - 已知 K 与 P , 正向运算快速
 - 已知 Q 与 P , 计算 K 的逆向运算非常困难

$$Q = \underbrace{P + \overbrace{P + \dots + P}^K}$$

ECDH 的步骤

- 步骤

1. Alice 选定大整数 K_a 作为私钥
2. 基于选定曲线及曲线上的共享 P 点, Alice 计算出 $Q_a = K_a \cdot P$
3. Alice 将 Q_a 、选定曲线、共享 P 点传递给 Bob
4. Bob 选定大整数 K_b 作为私钥, 将计算了 $Q_b = K_b \cdot P$, 并将 Q_b 传递给 Alice
5. Alice 生成密钥 $Q_b \cdot K_a = (X, Y)$, 其中 X 为对称加密的密钥
6. Bob 生成密钥 $Q_a \cdot K_b = (X, Y)$, 其中 X 为对称加密的密钥

- $Q_b \cdot K_a = K_a \cdot (K_b \cdot P) = K_a \cdot K_b \cdot P = K_b \cdot (K_a \cdot P) = Q_a \cdot K_b$

X25519 曲线

- 椭圆曲线变种：Montgomery curve 蒙哥马利曲线

- $By^2 = x^3 + Ax^2 + x$

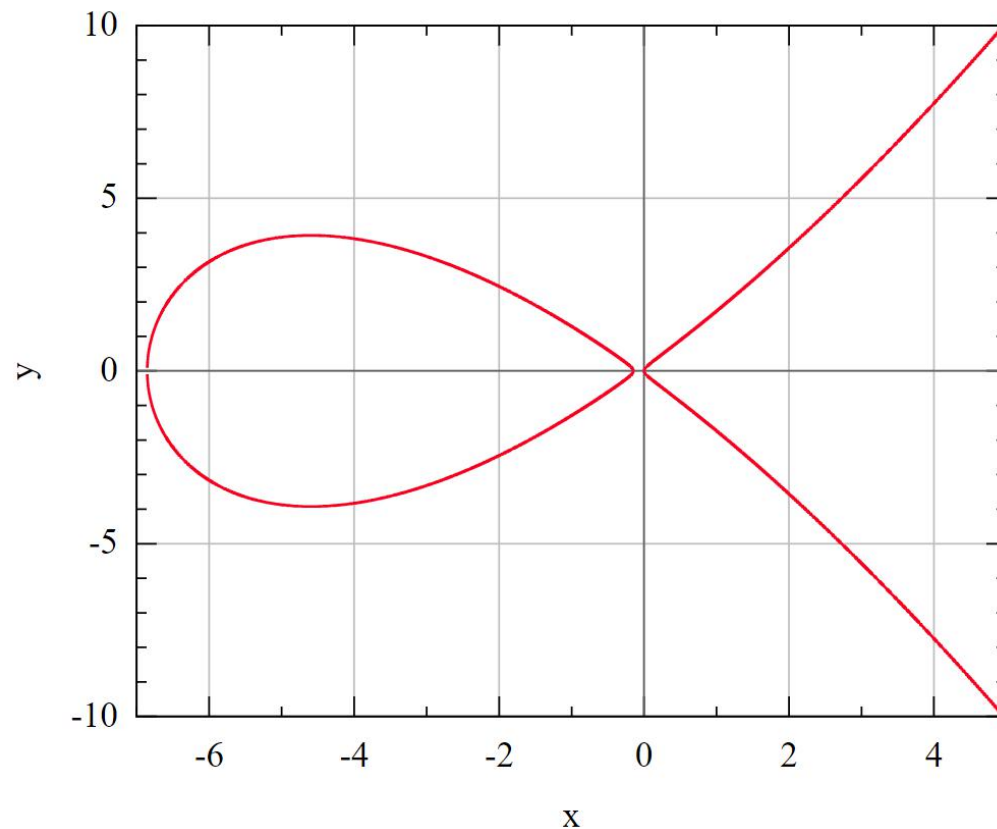
- 如右图：A=7, B=3

- X25519: $y^2 = x^3 + 486662x^2 + x$

- p 等于 $2^{255} - 19$, 基点 G=9

- order N

- $2^{252} + 0x14def9dea2f79cd65812631a5cf5d3ed$



第 11 课 TLS1.2 与 TLS1.3 中的 ECDH 协议

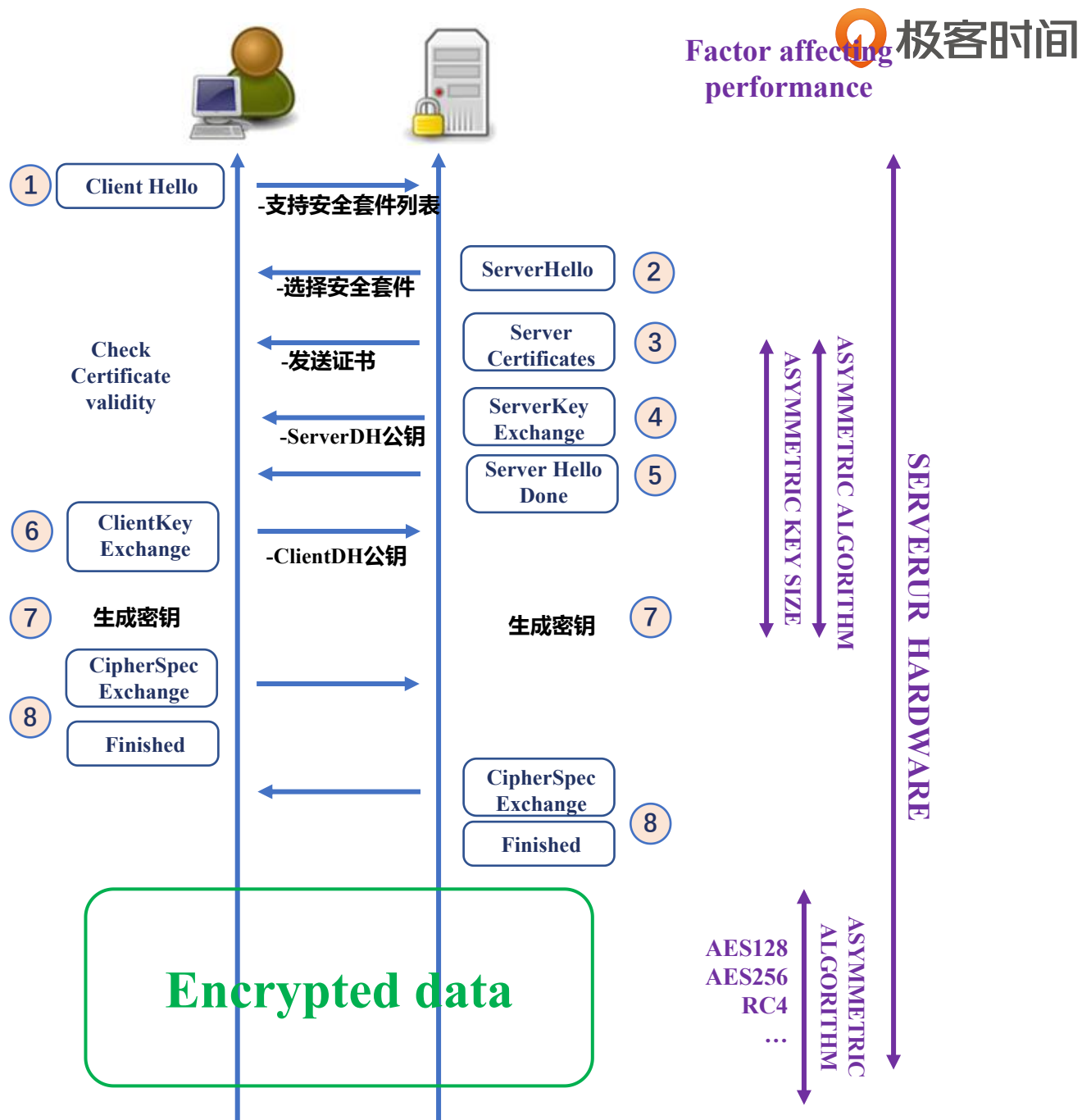
TLS1.2 通讯过程

验证身份

达成安全套件共识

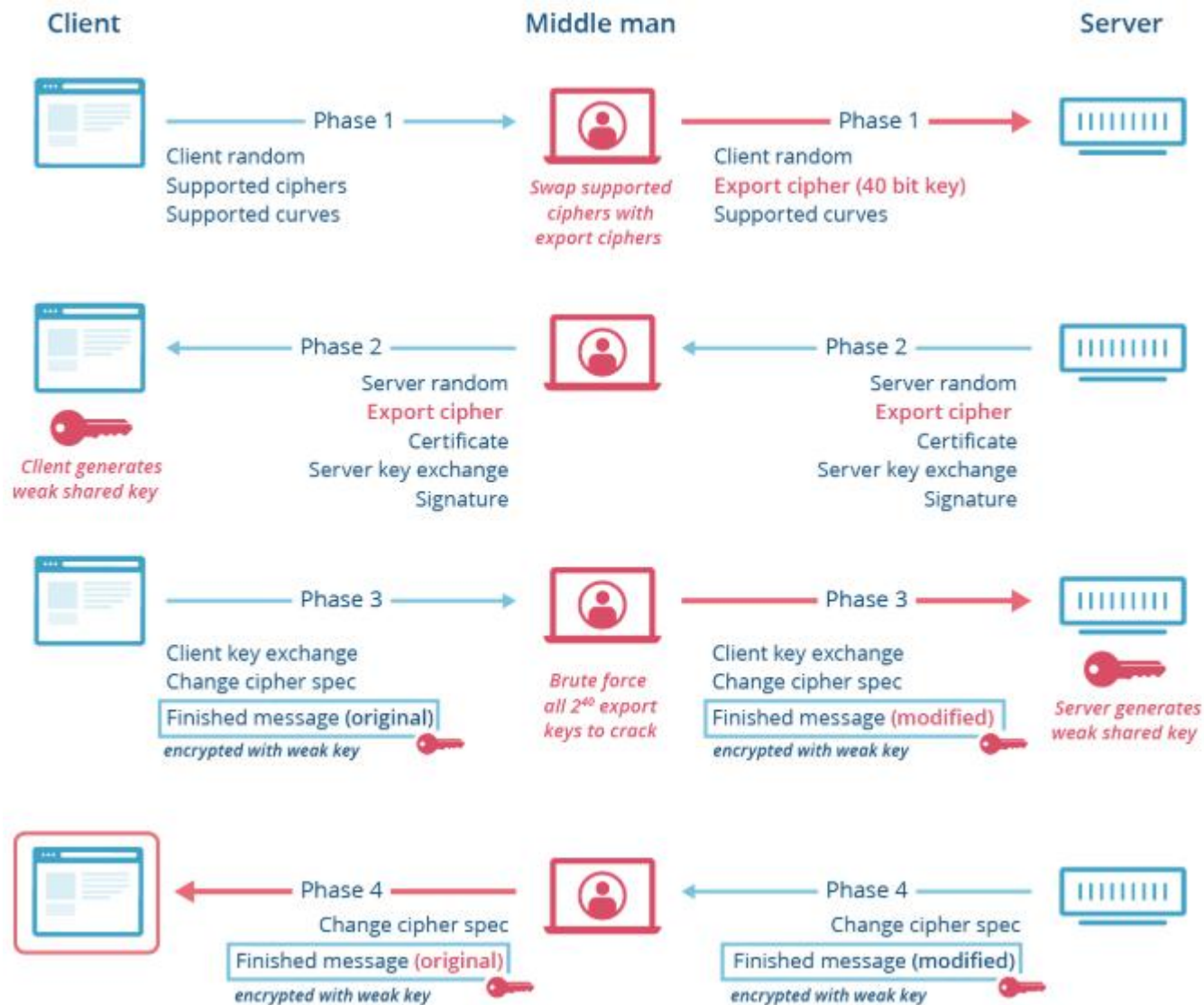
传递密钥

加密通讯



FREAK 攻击

- 2015 年发现漏洞
- 90 年代引入
 - 512 位以下 RSA 密钥可轻易破解




openssl 1.1.1 版本对 TLS1.3 的支持情况


- Ciphersuites 安全套件
 - TLS13-AES-256-GCM-SHA384
 - TLS13-CHACHA20-POLY1305-SHA256
 - TLS13-AES-128-GCM-SHA256
 - TLS13-AES-128-CCM-8-SHA256
 - TLS13-AES-128-CCM-SHA256

测试 TLS 站点支持情况

- <https://www.ssllabs.com/ssltest/index.html>

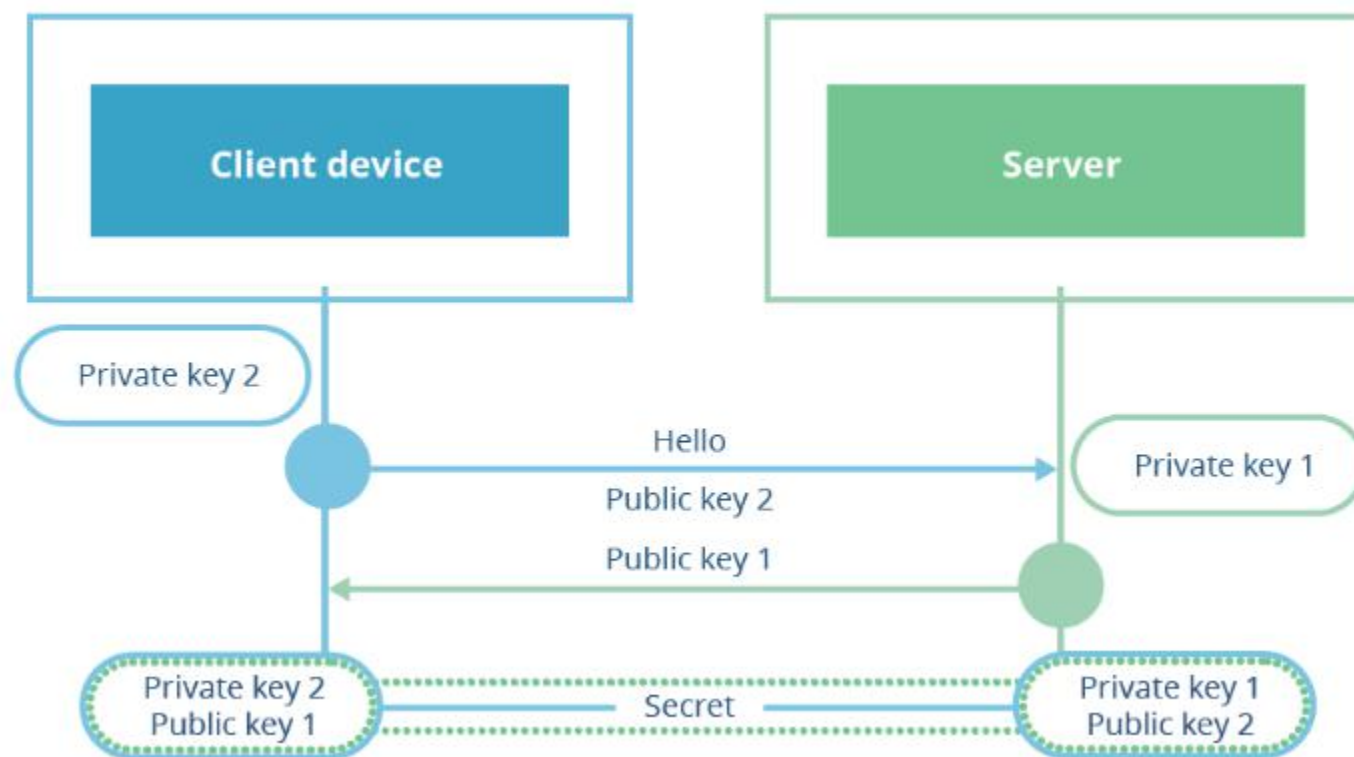


TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No
For TLS 1.3 tests, we only support RFC 8446.	



# TLS 1.3 (suites in server-preferred order)		<input type="checkbox"/>
TLS_AES_256_GCM_SHA384 (0x1302)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303)	ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_AES_128_GCM_SHA256 (0x1301)	ECDH x25519 (eq. 3072 bits RSA) FS	128

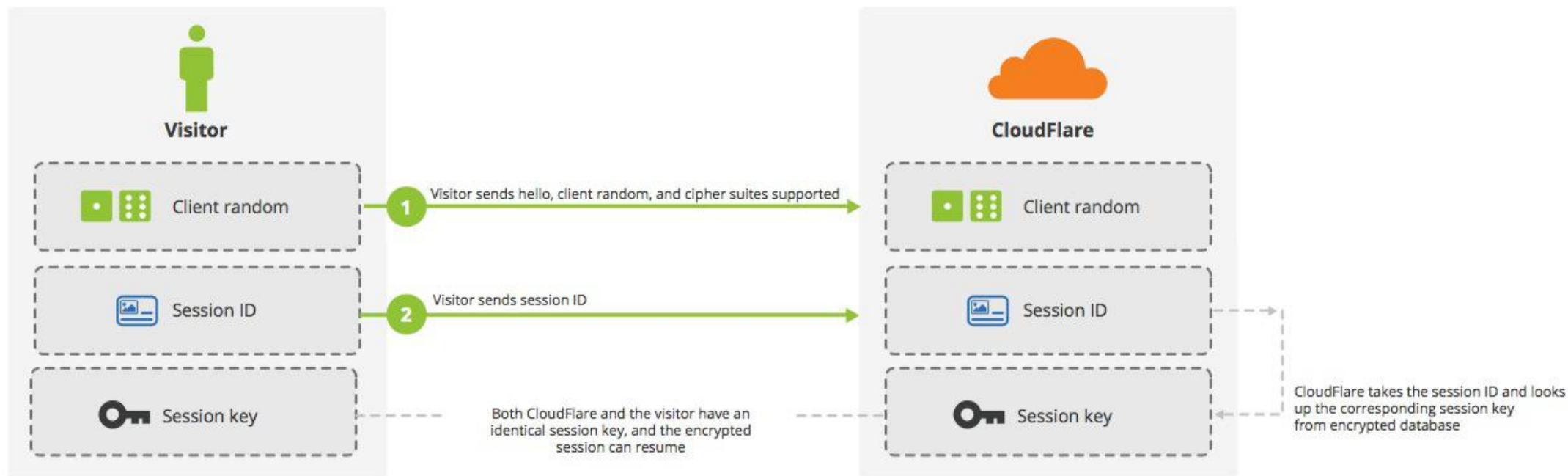
TLS1.3 中的密钥交换



第 12 课 握手的优化：session 缓存、ticket 票据及 TLS1.3 的 0-RTT

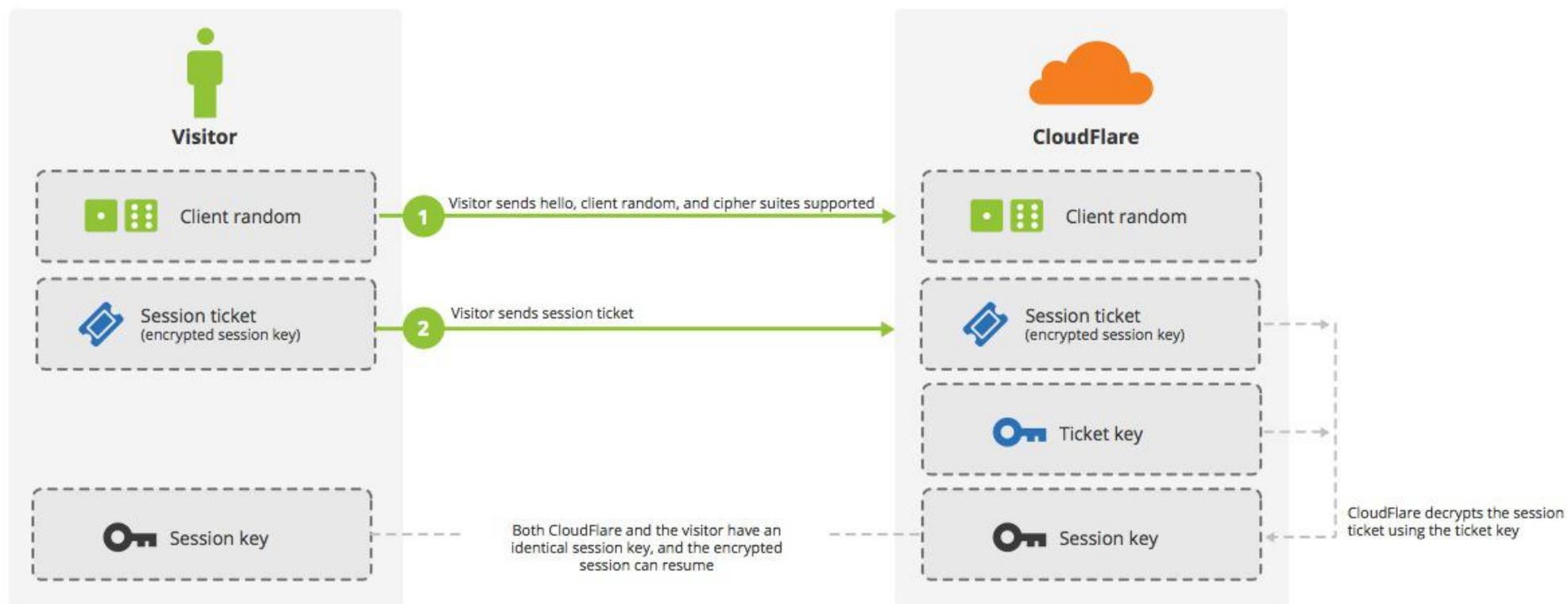
session 缓存：以服务器生成的 session ID 为依据

Session resume with session ID

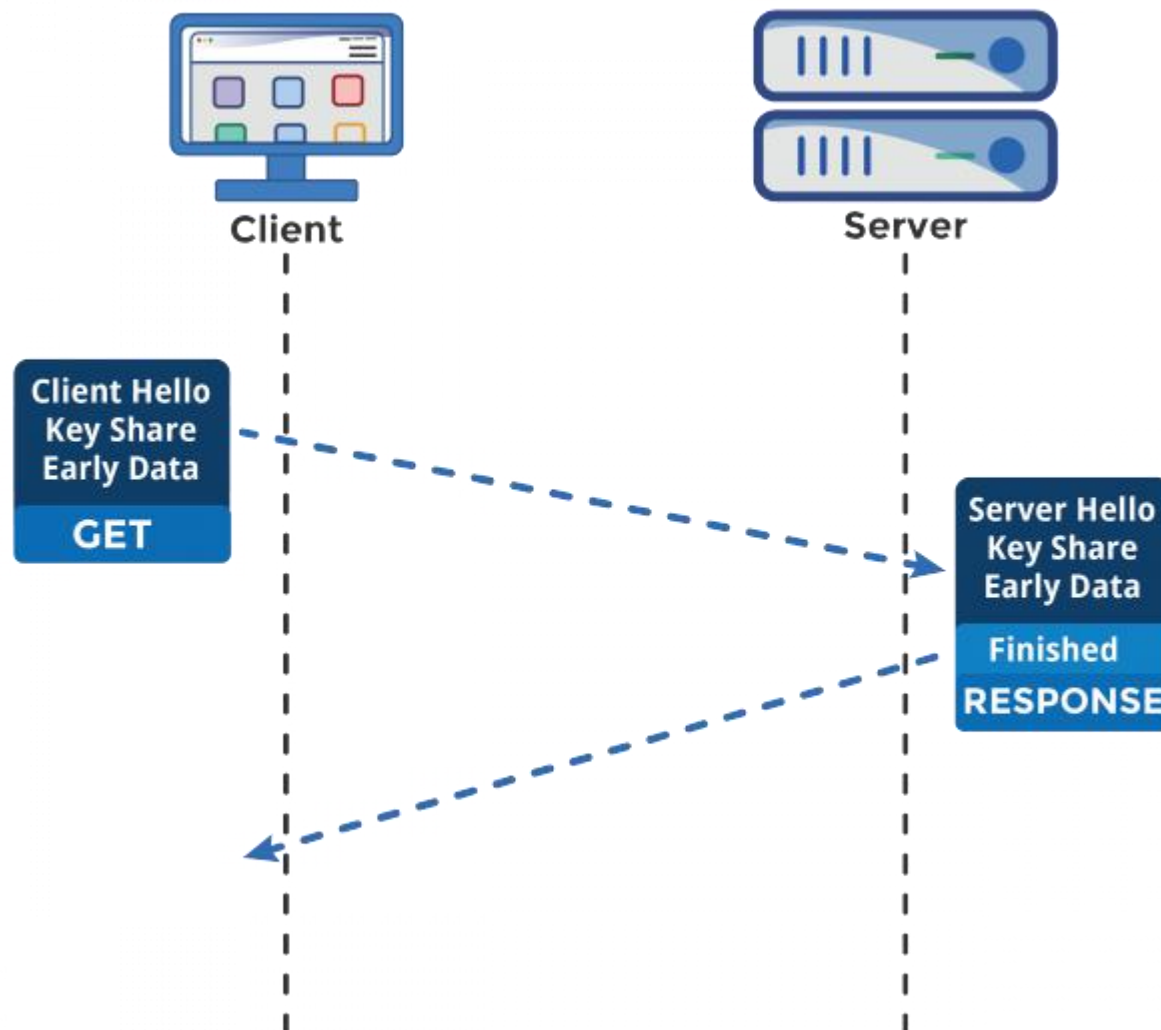


session ticket

Session resume with session ticket

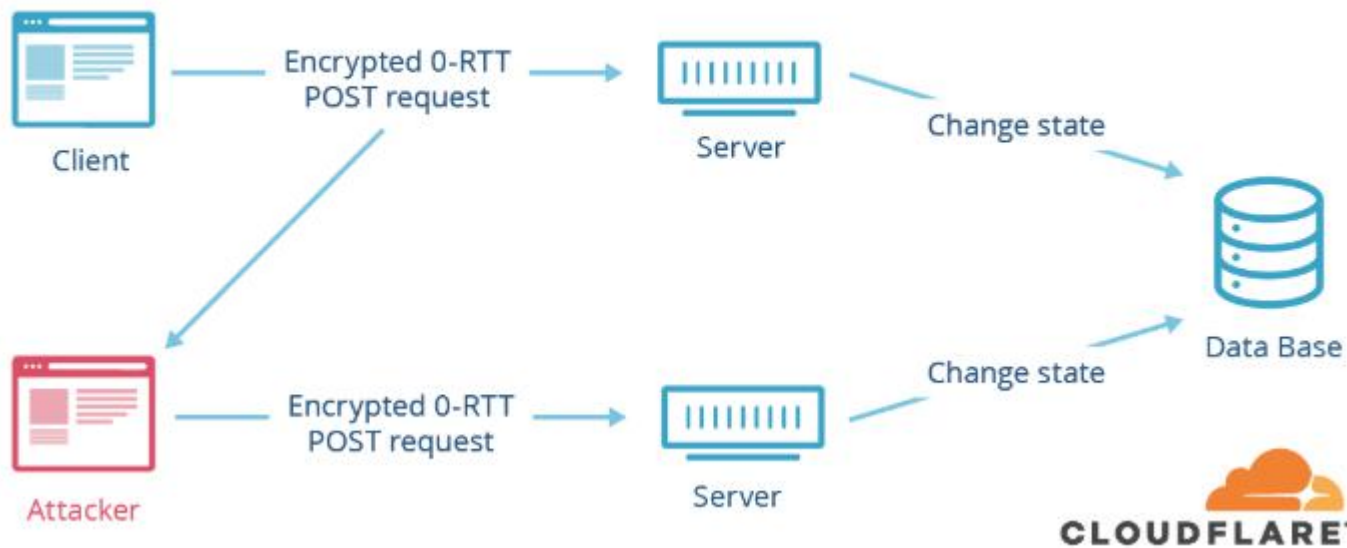


TLS1.3 的 0RTT 握手



0-RTT 面临的 replay 攻击

0-RTT Attack



第 13 课 TLS 与量子通讯的原理

TLS 密码学回顾

- 通讯双方在身份验证的基础上，协商出一次性的、随机的密钥
 - PKI 公钥基础设施
 - TLS 中间件生成一次性的、随机的密钥参数
 - DH 系列协议基于非对称加密技术协商出密钥
- 使用分组对称加密算法，基于有限长度的密钥将任意长度的明文加密传输
 - 密钥位数
 - 分组工作模式

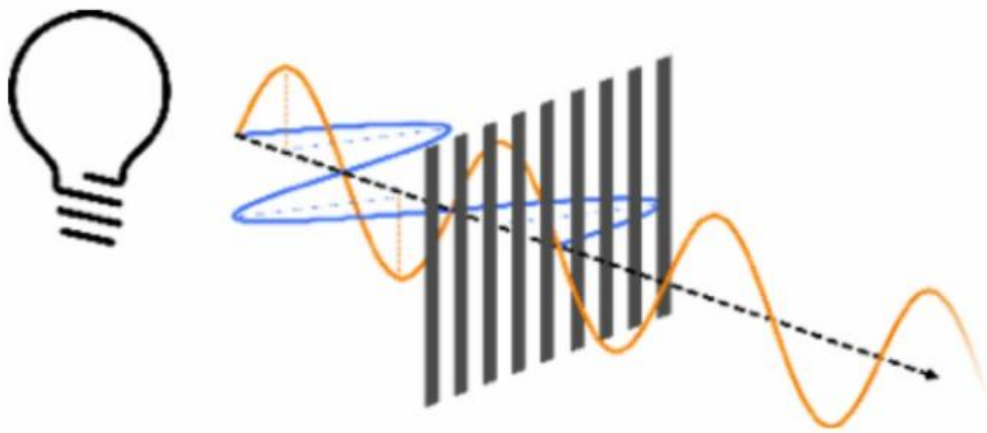
克劳德·艾尔伍德·香农：信息论

- 证明 one-time-pad (OTP) 的绝对安全性
 - 密钥是随机生成的
 - 密钥的长度大于等于明文长度
 - 相同的密钥只能使用一次
- 如何传递密钥？



QKD 与光偏振原理

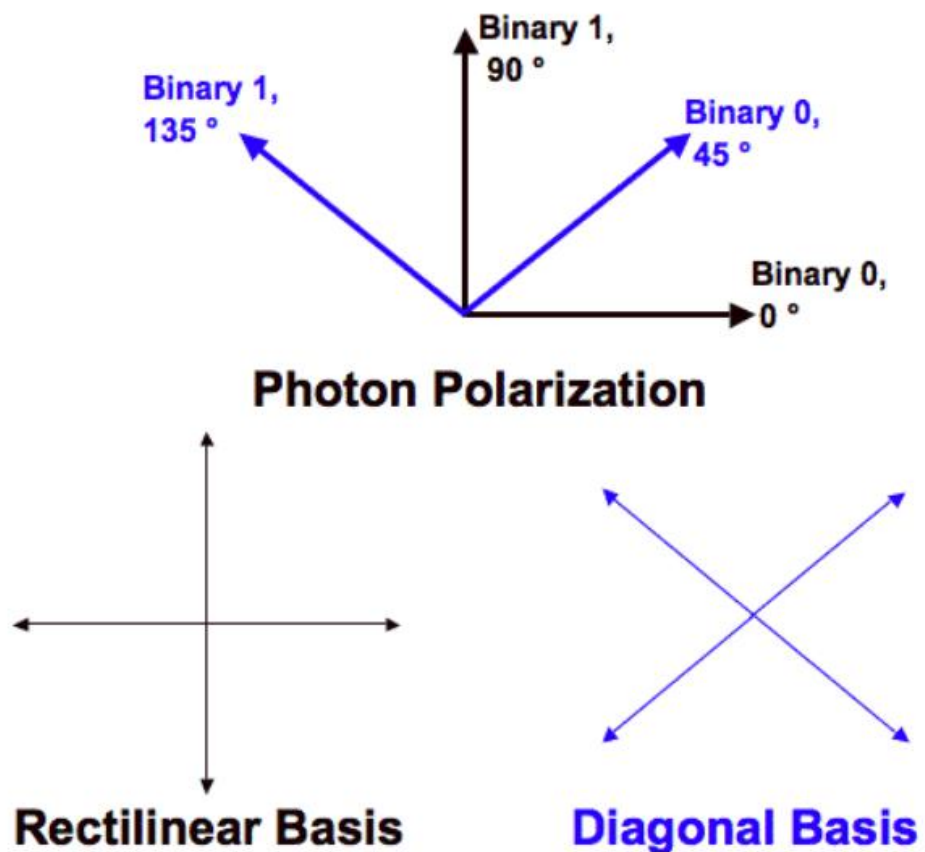
- 量子密钥分发 quantum key distribution, 简称 QKD
 - 量子力学：任何对量子系统的测量都会对系统产生干扰
 - QKD：如果有第三方试图窃听密码，则通信的双方会察觉



第 14 课 量子通讯BB84协议的执行流程

BB84 协议

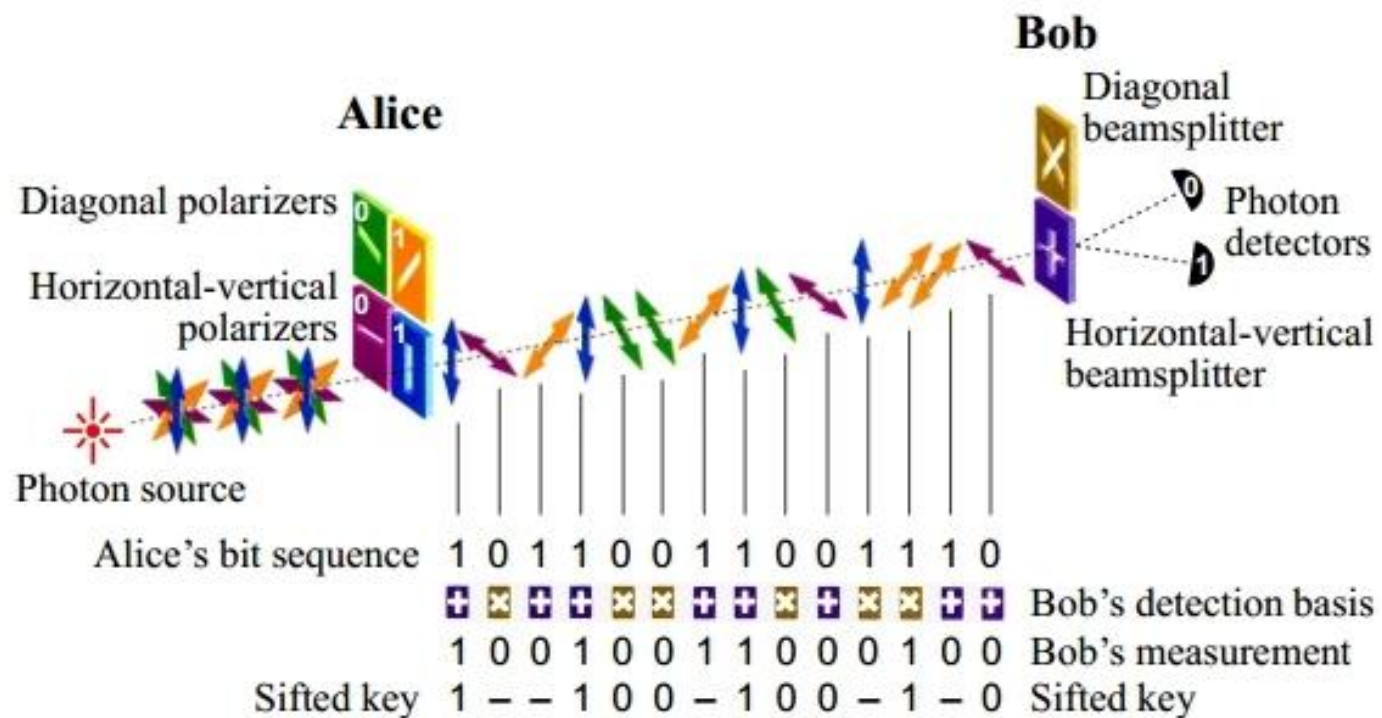
- 由 Charles Bennett 与 Gilles Brassard 在 1984 年发表



基	0	1
+	↑	→
×	↗	↘

BB84 协议示意图

- $50\% \times 50\% = 25\%$ 的错误率



QKD 密钥纠错与隐私增强

Alice's bit	0	1	1	0	1	0	0	1
Alice's basis	+	+	X	+	X	X	X	+
Alice's polarization	↑	→	↖	↑	↖	↗	↗	→
Bob's basis	+	X	X	X	+	X	+	+
Bob's measurement	↑	↗	↖	↗	→	↗	→	→
Public discussion								
Shared Secret key	0		1			0		1

基	0	1
+	↑	→
X	↗	↖