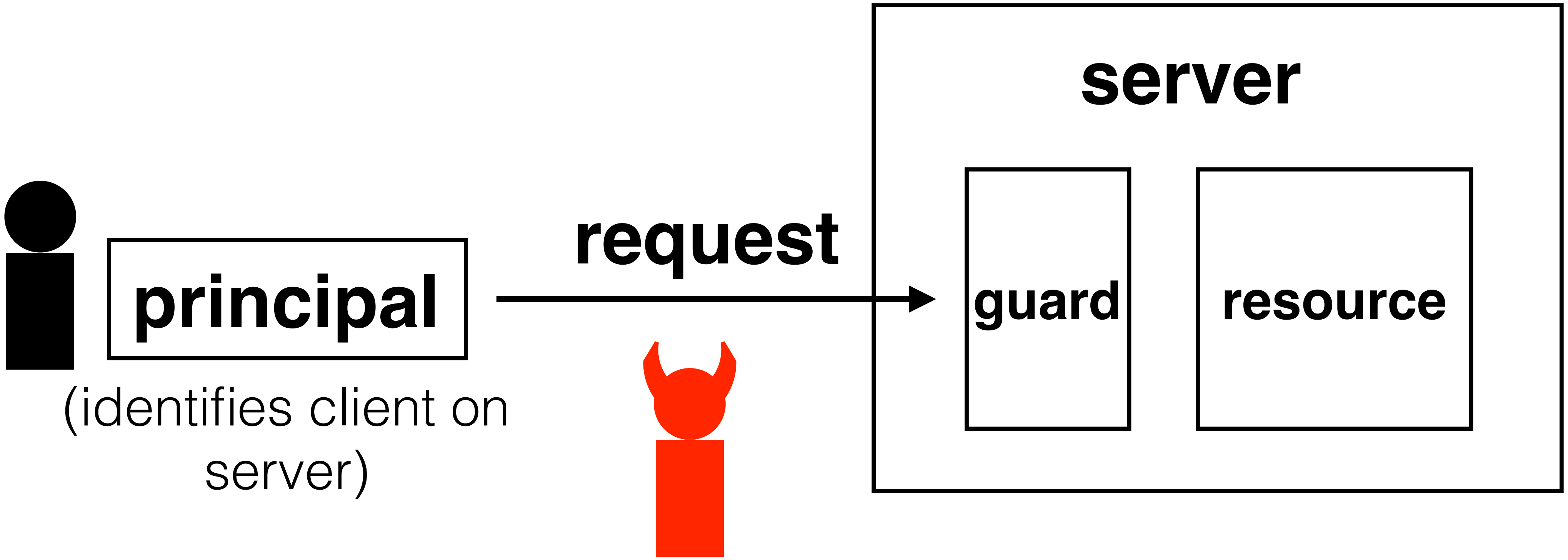
6.033 Spring 2017

Lecture #22

- **Combating network adversaries**
  - **Secure Channels**
  - **Signatures**


principal
(identifies client on server)
request
server
guard
resource

```
14:36:30.270515 1964945311us tsft bad-fcs -95dB noise antenna 0 2412 MHz 11g ht/20 72.2 Mb/s MCS 7 20 MHz
short GI greenfield BCC FEC [bit 20] CF +QoS IP 18.111.89.99.57297 > 51.254.196.92.80: Flags [P.], seq
1410:1776, ack 58893, win 4096, options [nop,nop,TS val 751598929 ecr 695202096], length 366: HTTP: GET /
stats/impression.php?adid=163D9B15-D06C-4A30-8B3E-77F246241FEF&source=com.ketchapp.ballz&name=com.ketchap
p.pixall&image=http://173.244.217.213/videos/Pixall_1/index.m3u8&type=1 HTTP/1.1

     0x0000:  aaaa 0300 0000 0800 4500 01a2 0485 4000  ........E.....@.
     0x0010:  4006 d0a4 126f 5963 33fe c45c dfd1 0050  @....oYc3..\...P
     0x0020:  800c 3f6b 6466 db1a 8018 1000 c9cc 0000  ..?kdf..........
     0x0030:  0101 080a 2ccc 7d51 296f f130 4745 5420  ....,.}Q)o.0GET.
     0x0040:  2f73 7461 7473 2f69 6d70 7265 7373 696f  /stats/impressio
     0x0050:  6e2e 7068 703f 6164 6964 3d31 3633 4439  n.php?adid=163D9
     0x0060:  4231 352d 4430 3643 2d34 4133 302d 3842  B15-D06C-4A30-8B
     0x0070:  3345 2d37 3746 3234 3632 3431 4645 4626  3E-77F246241FEF&
     0x0080:  736f 7572 6365 3d63 6f6d 2e6b 6574 6368  source=com.ketch
     0x0090:  6170 702e 6261 6c6c 7a26 6e61 6d65 3d63  app.ballz&name=c
     0x00a0:  6f6d 2e6b 6574 6368 6170 702e 7069 7861  om.ketchapp.pixa
     0x00b0:  6c6c 2669 6d61 6765 3d68 7474 703a 2f2f  ll&image=http://
     0x00c0:  3137 332e 3234 342e 3231 372e 3231 332f  173.244.217.213/
     0x00d0:  7669 6465 6f73 2f50 6978 616c 6c5f 312f  videos/Pixall_1/
     0x00e0:  696e 6465 782e 6d33 7538 2674 7970 653d  index.m3u8&type=
     0x00f0:  3120 4854 5450 2f31 2e31 0d0a 486f 7374  1.HTTP/1.1..Host
     0x0100:  3a20 7777 772e 6b65 7463 6861 7070 2e6f  :.www.ketchapp.o
     0x0110:  7267 0d0a 4163 6365 7074 3a20 2a2f 2a0d  rg..Accept:.*/*.
     0x0120:  0a41 6363 6570 742d 4c61 6e67 7561 6765  .Accept-Language
     0x0130:  3a20 656e 2d75 730d 0a43 6f6e 6e65 6374  :.en-us..Connect
     0x0140:  696f 6e3a 206b 6565 702d 616c 6976 650d  ion:.keep-alive.
     0x0150:  0a41 6363 6570 742d 456e 636f 6469 6e67  .Accept-Encoding
     0x0160:  3a20 677a 6970 2c20 6465 666c 6174 650d  :.gzip,.deflate.
     0x0170:  0a55 7365 722d 4167 656e 743a 2062 616c  .User-Agent:.bal
     0x0180:  6c7a 2f31 2e31 2e33 2043 464e 6574 776f  lz/1.1.3.CFNetwo
     0x0190:  726b 2f38 3038 2e33 2044 6172 7769 6e2f  rk/808.3.Darwin/
     0x01a0:  3136 2e33 2e30 0d0a 0d0a                16.3.0....
```

```
14:36:35.210953 1969884900us tsft -73dB signal -95dB noise antenna 0 2412 MHz 11g ht/20 65.0 Mb/s MCS 7 20
MHz lon GI mixed BCC FEC [bit 20] CF +QoS IP 18.111.89.99.57297 > 51.254.196.92.80: Flags [P.], seq
1776:2104, ack 59076, win 4096, options [nop,nop,TS val 751603333 ecr 695202151], length 328: HTTP: GET /
crosspromo/interstitiel/interstitiel_gameover.xml?app=com.ketchapp.ballz&orientation=&lang=en-
US&adid=163D9B15-D06C-4A30-8B3E-77F246241FEF HTTP/1.1
```

```
        0x0000:  aaaa 0300 0000 0800 4500 017c b7b9 4000  ........E..|..@.
        0x0010:  4006 1d96 126f 5963 33fe c45c dfd1 0050  @....oYc3..\...P
        0x0020:  800c 40d9 6466 dbd1 8018 1000 1fab 0000  ..@.df..........
        0x0030:  0101 080a 2ccc 8e85 296f f167 4745 5420  ....,...)o.gGET.
        0x0040:  2f63 726f 7373 7072 6f6d 6f2f 696e 7465  /crosspromo/inte
        0x0050:  7273 7469 7469 656c 2f69 6e74 6572 7374  rstitiel/interst
        0x0060:  6974 6965 6c5f 6761 6d65 6f76 6572 2e78  itiel_gameover.x
        0x0070:  6d6c 3f61 7070 3d63 6f6d 2e6b 6574 6368  ml?app=com.ketch
        0x0080:  6170 702e 6261 6c6c 7a26 6f72 6965 6e74  app.ballz&orient
        0x0090:  6174 696f 6e3d 266c 616e 673d 656e 2d55  ation=&lang=en-U
        0x00a0:  5326 6164 6964 3d31 3633 4439 4231 352d  S&adid=163D9B15-
        0x00b0:  4430 3643 2d34 4133 302d 3842 3345 2d37  D06C-4A30-8B3E-7
        0x00c0:  3746 3234 3632 3431 4645 4620 4854 5450  7F246241FEF.HTTP
        0x00d0:  2f31 2e31 0d0a 486f 7374 3a20 7777 772e  /1.1..Host:.www.
        0x00e0:  6b65 7463 6861 7070 2e6f 7267 0d0a 4163  ketchapp.org..Ac
        0x00f0:  6365 7074 3a20 2a2f 2a0d 0a41 6363 6570  cept:.*/*..Accep
        0x0100:  742d 4c61 6e67 7561 6765 3a20 656e 2d75  t-Language:.en-u
        0x0110:  730d 0a43 6f6e 6e65 6374 696f 6e3a 206b  s..Connection:.k
        0x0120:  6565 702d 616c 6976 650d 0a41 6363 6570  eep-alive..Accep
        0x0130:  742d 456e 636f 6469 6e67 3a20 677a 6970  t-Encoding:.gzip
        0x0140:  2c20 6465 666c 6174 650d 0a55 7365 722d  ,.deflate..User-
        0x0150:  4167 656e 743a 2062 616c 6c7a 2f31 2e31  Agent:.ballz/1.1
        0x0160:  2e33 2043 464e 6574 776f 726b 2f38 3038  .3.CFNetwork/808
        0x0170:  2e33 2044 6172 7769 6e2f 3136 2e33 2e30  .3.Darwin/16.3.0
        0x0180:  0d0a 0d0a
```

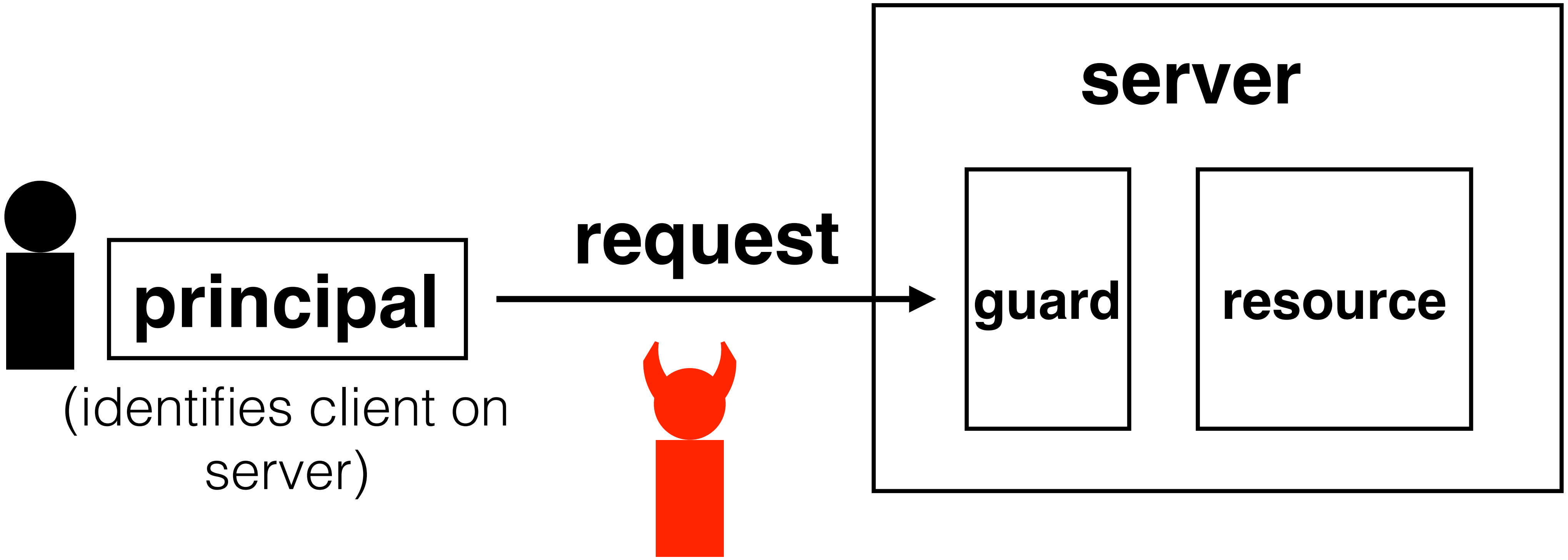
```
14:15:57.156383 731851825us tsft -95dB noise antenna 0 2412 MHz 11g ht/20 26.0 Mb/s MCS 3 20 MHz lon GI
greenfield BCC FEC [bit 20] CF +QoS IP dhcp-18-111-89-99
.dyn.mit.edu.57061 > 17.154.66.156.https: Flags [P.], seq 0:517, ack 1, win 8192, length 517

        0x0000:  aaaa 0300 0000 0800 4500 022d 9fd8 4000  ........E..-..@.
        0x0010:  4006 d8ea 126f 5963 119a 429c dee5 01bb  @....oYc..B.....
        0x0020:  f7f4 9d92 e59a 1614 5018 2000 ae38 0000  ........P....8..
        0x0030:  1603 0102 0001 0001 fc03 0359 077b 5d64  ...........Y.{]d
        0x0040:  6a53 0208 0cde 5c0a 26e8 5732 151d c778  jS....\.&.W2...x
        0x0050:  16c3 d1cc d5e6 c8a1 b940 3220 3ce6 c3c9  .........@2.<...
        0x0060:  ccb5 f523 3ae1 bf92 cd1f 1ac9 efc4 b155  ...#:..........U
        0x0070:  576a 4af8 4bc9 5b38 38dd 5d0e 0026 00ff  WjJ.K.[88.]..&..
        0x0080:  c02c c02b c024 c023 c00a c009 c030 c02f  .,.+.$.#.....0./
        0x0090:  c028 c027 c014 c013 009d 009c 003d 003c  .(.'.........=.<
        0x00a0:  0035 002f 0100 018d 0000 001d 001b 0000  .5./............
        0x00b0:  1870 3331 2d62 7579 2e69 7475 6e65 732e  .p31-buy.itunes.
        0x00c0:  6170 706c 652e 636f 6d00 0a00 0800 0600  apple.com.......
        0x00d0:  1700 1800 1900 0b00 0201 0000 0d00 1200  ................
        0x00e0:  1004 0102 0105 0106 0104 0302 0305 0306  ................
        0x00f0:  0333 7400 0000 1000 3000 2e02 6832 0568  .3t.....0...h2.h
        0x0100:  322d 3136 0568 322d 3135 0568 322d 3134  2-16.h2-15.h2-14
        0x0110:  0873 7064 792f 332e 3106 7370 6479 2f33  .spdy/3.1.spdy/3
        0x0120:  0868 7474 702f 312e 3100 0500 0501 0000  .http/1.1.......
        0x0130:  0000 0012 0000 0017 0000 0015 00f7 0000  ................
        0x0140:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0150:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0160:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0170:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0180:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0190:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01d0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x01f0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
```

```
14:15:58.090994 732786159us tsft bad-fcs -68dB signal -93dB noise antenna 0 2412 MHz 11g ht/20 19.5 Mb/s MC
2 20 MHz lon GI mixed BCC FEC [bit 20] CF +QoS IP 18.111.89.99.57063 > 216.157.12.18.80: Flags [P.], seq
0:1091, ack 1, win 8192, length 1091: HTTP

        0x0000:  aaaa 0300 0000 0800 4500 046b 1832 4000  ........E..k.2@.
        0x0010:  4006 cdd9 126f 5963 d89d 0c12 dee7 0050  @....oYc.......P
        0x0020:  5797 e83d 727f 615f 5018 2000 9898 0000  W..=r.a_P.......
        0x0030:  4745 5420 2f67 6574 4164 3f61 7069 643d  GET./getAd?apid=
        0x0040:  3231 3434 3733 2661 743d 6226 6174 653d  214473&at=b&ate=
        0x0050:  7472 7565 2662 6c3d 3832 2663 6163 6865  true&bl=82&cache
        0x0060:  6476 6964 656f 3d74 7275 6526 636e 3d53  dvideo=true&cn=S
        0x0070:  7072 696e 7426 636f 6e6e 3d77 6966 6926  print&conn=wifi&
        0x0080:  636f 756e 7472 793d 5553 2664 656e 7369  country=US&densi
        0x0090:  7479 3d32 2664 6d3d 6950 686f 6e65 3725  ty=2&dm=iPhone7%
        0x00a0:  3243 3226 646f 3d70 6f72 7472 6169 7426  2C2&do=portrait&
        0x00b0:  6476 3d31 302e 322e 3126 6861 6964 3d6d  dv=10.2.1&haid=m
        0x00c0:  6d68 5f35 6638 3139 6332 3933 3164 3364  mh_5f819c2931d3d
        0x00d0:  3938 3534 3039 3636 3432 6362 3236 3636  9854096642cb2666
        0x00e0:  3664 335f 3134 6566 3465 3763 3739 3462  6d3_14ef4e7c794b
        0x00f0:  3563 3130 3063 6263 3530 3139 6433 6364  5c100cbc5019d3cd
        0x0100:  3064 3038 fa62 3835 3762 3537 2668 6561  0d08.b857b57&hea
        0x0110:  6470 686f 6e65 733d 6661 6c73 6526 6870  dphones=false&hp
        0x0120:  783d 3636 3726 6873 6874 3d35 3026 6873  x=667&hsht=50&hs
        0x0130:  7764 3d33 3230 266c 616e 6775 6167 653d  wd=320&language=
        0x0140:  656e 266c 6f63 3d66 616c 7365 266d 6363  en&loc=false&mcc
        0x0150:  3d33 3130 266d 6963 3d75 6e6b 6e6f 776e  =310&mic=unknown
        0x0160:  266d 6e63 3d31 3230 2670 6970 3d66 6538  &mnc=120&pip=fe8
        0x0170:  3025 3341 2533 4134 3661 2533 4136 6262  0%3A%3A46a%3A6bb
        0x0180:  3725 3341 3630 3525 3341 3730 3132 2670  7%3A605%3A7012&p
        0x0190:  6b69 643d 636f 6d2e 6f65 636f 7761 792e  kid=com.oecoway.
        0x01a0:  6672 6965 6e64 6c79 4c69 7465 2670 6b6e  friendlyLite&pkn
        0x01b0:  6d3d 4672 6965 6e64 6c79 2670 6c75 6767  m=Friendly&plugg
        0x01c0:  6564 3d66 616c 7365 2672 6571 7479 7065  ed=false&reqtype
        0x01d0:  3d67 6574 6164 2673 646b 7665 7273 696f  =getad&sdkversio
        0x01e0:  6e3d 362e 332e 312d 6434 6430 6334 652e  n=6.3.1-d4d0c4e.
        0x01f0:  6926 7365 6375 7265 636f 6e74 656e 743d  i&securecontent=
14:05:29.947459 104653458us tsft -70dB signal -92dB noise antenna 0 2412 MHz 11g ht/20 39.0 Mb/s MCS 10 20
MHz lon GI mixed BCC FEC [bit 20] CF +QoS IP 10.189.6.135.5353 > 224.0.0.251.5353: 0*- [0q] 2/0/3 (Cache
flush) PTR Bobs-iPhone.local., (Cache flush) PTR Bobs-iPhone.local. (217)

        0x0000:  aaaa 0300 0000 0800 4500 00f5 2053 0000  ........E....S..
        0x0010:  ff11 a865 0abd 0687 e000 00fb 14e9 14e9  ...e............
        0x0020:  00e1 5867 0000 8400 0000 0002 0000 0003  ..Xg............
        0x0030:  0137 0135 0144 0133 0139 0130 0138 0133  .7.5.D.3.9.0.8.3
        0x0040:  0135 0135 0139 0144 0144 0141 0143 0130  .5.5.9.D.D.A.C.0
        0x0050:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
        0x0060:  0130 0130 0130 0130 0130 0138 0145 0146  .0.0.0.0.0.8.E.F
        0x0070:  0369 7036 0461 7270 6100 000c 8001 0000  .ip6.arpa.......
        0x0080:  0078 0015 0d45 6c69 7a61 732d 6950 686f  .x.....Bobs-iPho
        0x0090:  6e65 056c 6f63 616c 0003 3133 3501 3603  ne.local..135.6.
        0x00a0:  3138 3902 3130 0769 6e2d 6164 6472 c050  189.10.in-addr.P
        0x00b0:  000c 8001 0000 0078 0002 c060 c00c 002f  .......x...`.../
        0x00c0:  8001 0000 0078 0006 c00c 0002 0008 c075  .....x.........u
        0x00d0:  002f 8001 0000 0078 0006 c075 0002 0008  ./.....x...u....
        0x00e0:  0000 2905 a000 0011 9400 1200 0400 0e00  ..).............
        0x00f0:  256e 8dc1 7d01 b16c 8dc1 7d01 b1         %n..}..l..}..
```



confidentiality: adversary cannot learn message contents

integrity: adversary cannot tamper with message contents
(if they do, client and/or server will detect it)

```
encrypt(key, message) → ciphertext
decrypt(key, ciphertext) → message
```

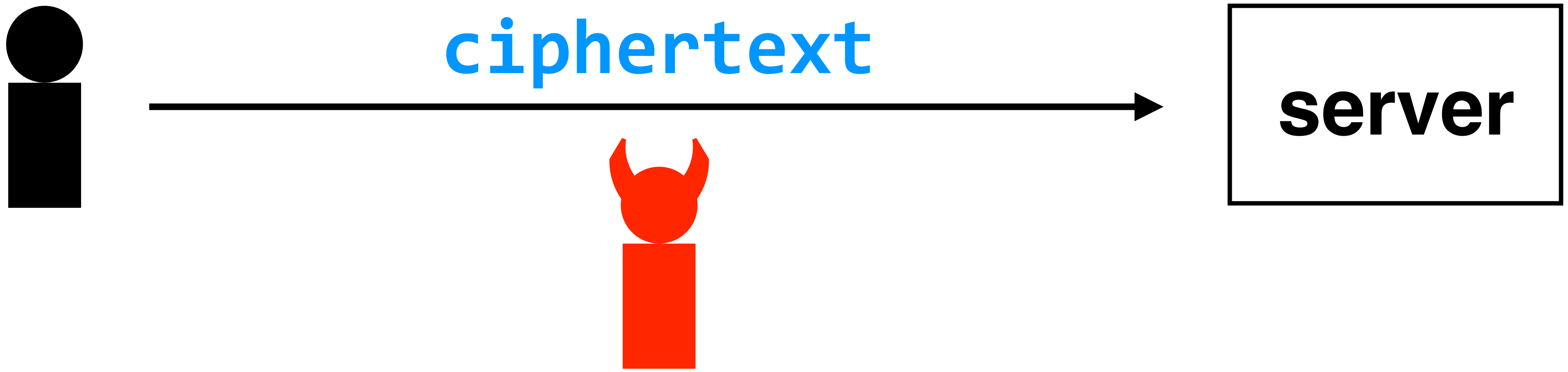
```
encrypt(34fbcbd1, “hello, world”) = 0x47348f63a67926cd393d4b93c58f78c
decrypt(34fbcbd1, “0x47348f63a67926cd393d4b93c58f78c”) = hello, world
```

property: given the `ciphertext`, it is (virtually) impossible to obtain the `message` without knowing the `key`

```
encrypt(key, message) → ciphertext
decrypt(key, ciphertext) → message
```

```
encrypt(34fbcbd1, “hello, world”) = 0x47348f63a67926cd393d4b93c58f78c
decrypt(34fbcbd1, “0x47348f63a67926cd393d4b93c58f78c”) = hello, world
```

property: given the `ciphertext`, it is (virtually) impossible to obtain the `message` without knowing the `key`



ciphertext

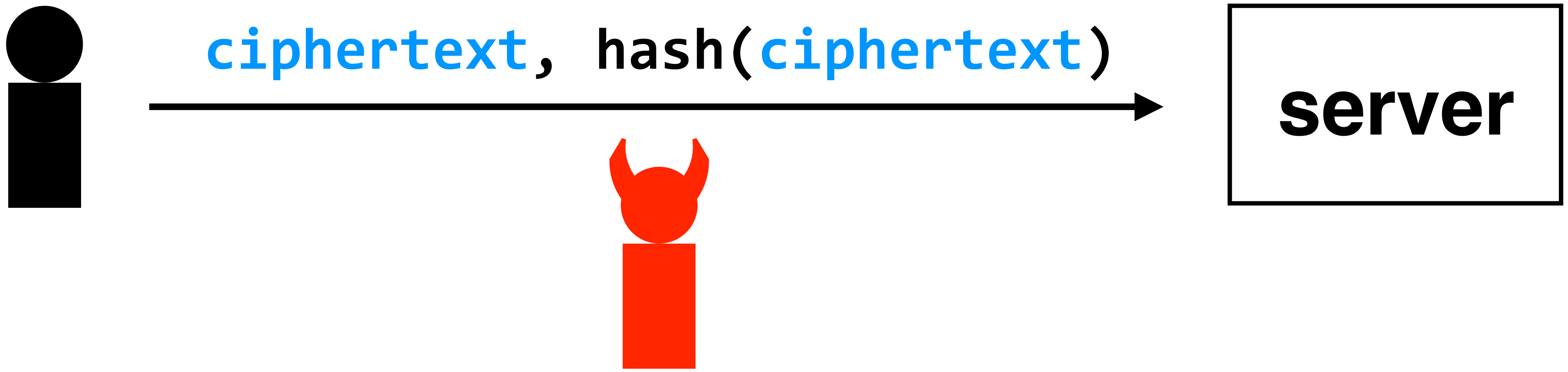
server

adversary can't determine `message`, *but* might be able to cleverly alter `ciphertext` so that it decrypts to a different message

```
encrypt(key, message) → ciphertext
decrypt(key, ciphertext) → message
```

```
encrypt(34fbcbd1, “hello, world”) = 0x47348f63a67926cd393d4b93c58f78c
decrypt(34fbcbd1, “0x47348f63a67926cd393d4b93c58f78c”) = hello, world
```

property: given the `ciphertext`, it is (virtually) impossible to obtain the `message` without knowing the `key`



`ciphertext, hash(ciphertext)`

server

no good — if the adversary changes `ciphertext`, it can also (correctly) update the hash

```
encrypt(key, message) → ciphertext
decrypt(key, ciphertext) → message
```

```
encrypt(34fbcbd1, “hello, world”) = 0x47348f63a67926cd393d4b93c58f78c
decrypt(34fbcbd1, “0x47348f63a67926cd393d4b93c58f78c”) = hello, world
```

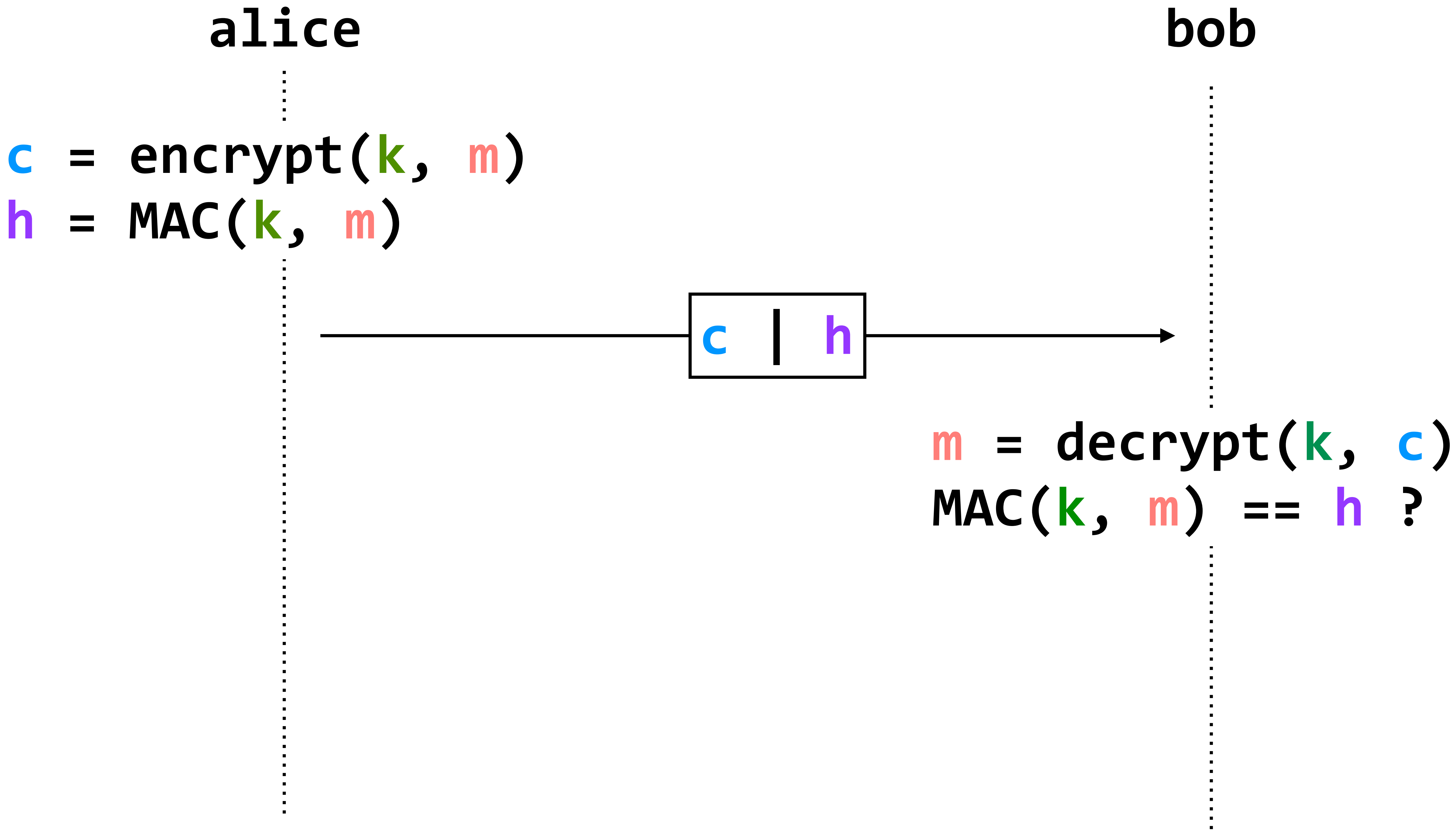
property: given the `ciphertext`, it is (virtually) impossible to obtain the `message` without knowing the `key`

```
MAC(key, message) → token
```

```
MAC(34fbcbd1, “hello, world”) = 0x59cccc95723737f777e62bc756c8da5c
```

property: given the `message`, it is (virtually) impossible to obtain the `token` without knowing the `key`

(it is also impossible to go in the reverse direction)



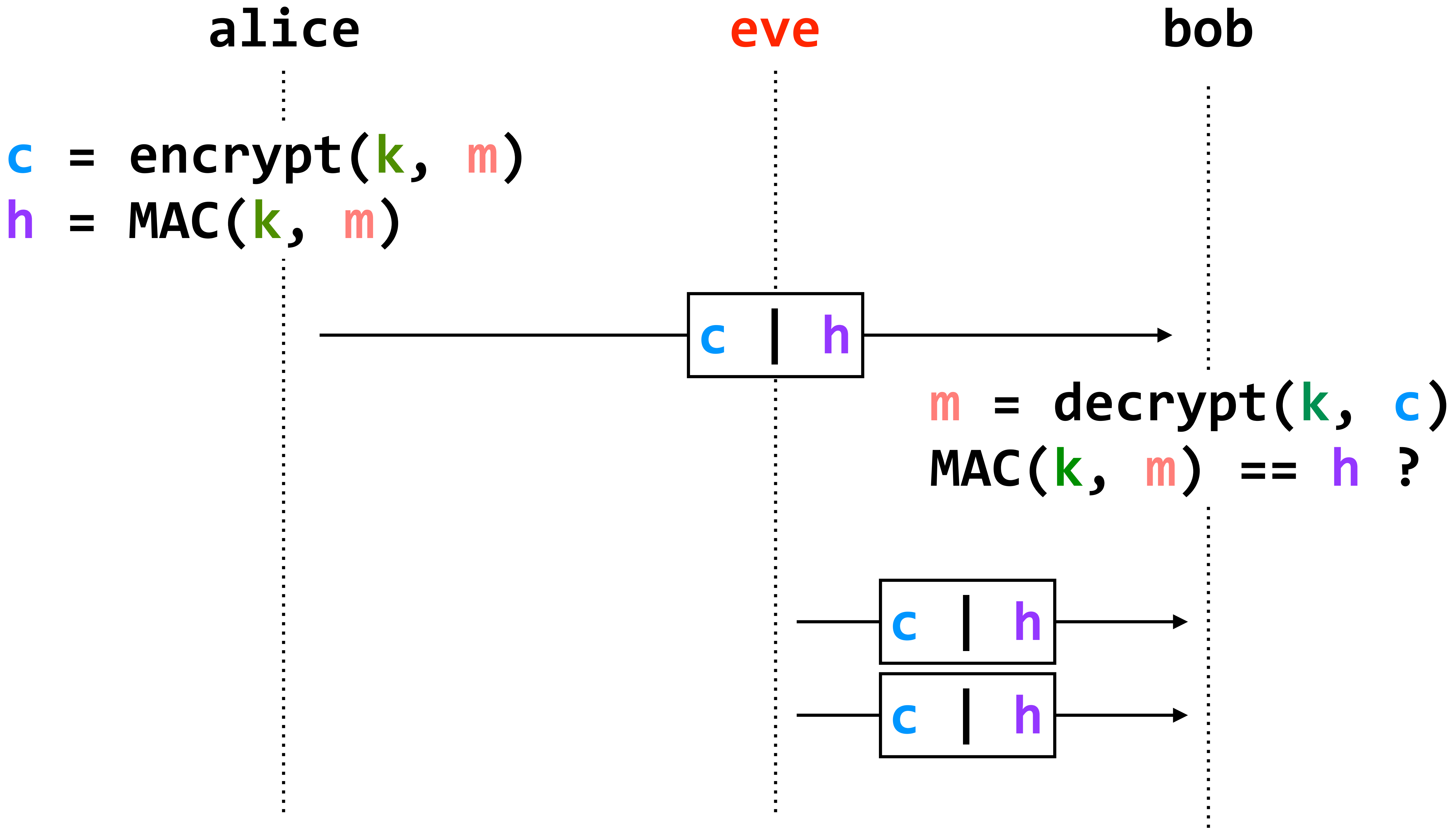
alice

bob

```
c = encrypt(k, m)
h = MAC(k, m)
```

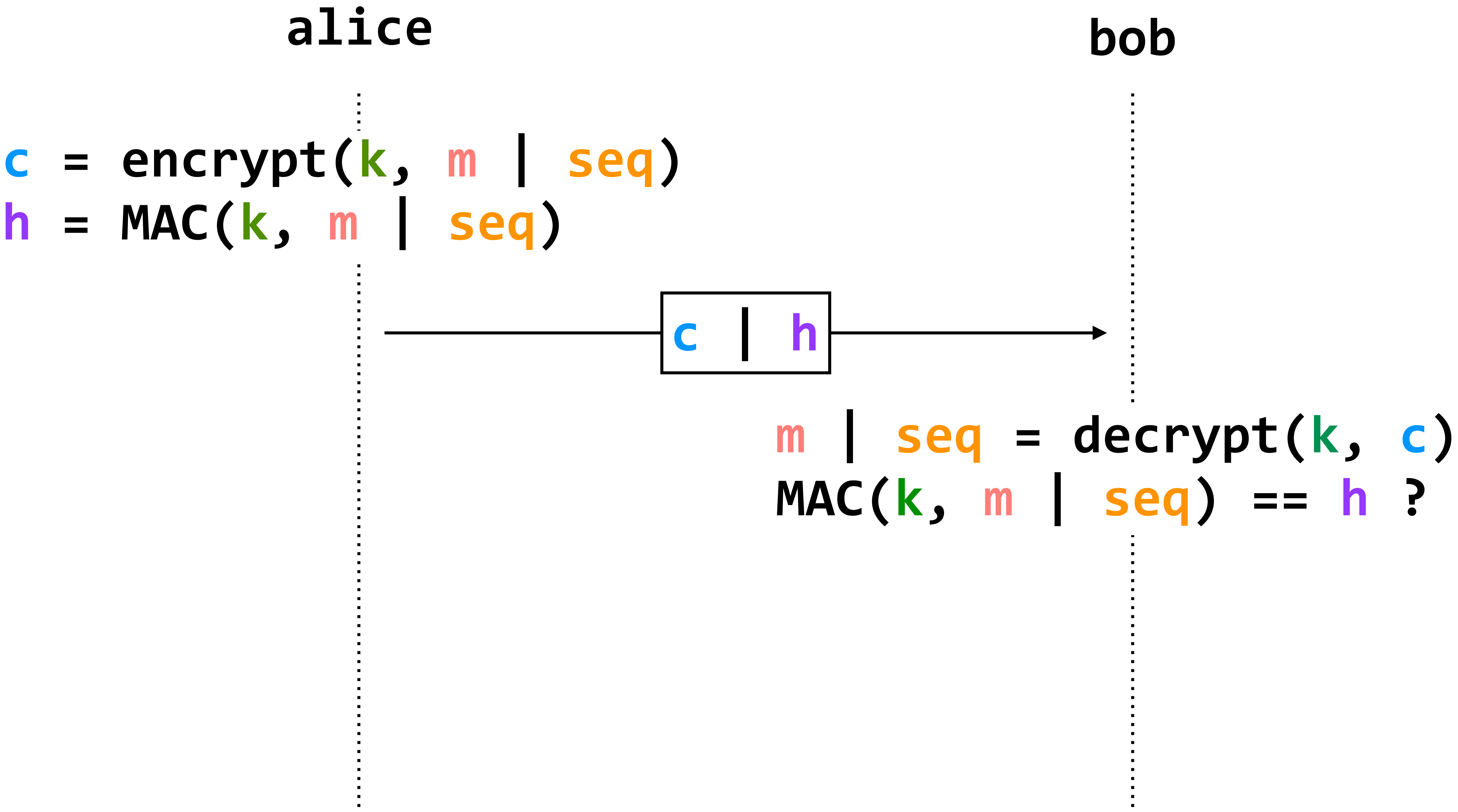
c | h

```
m = decrypt(k, c)
MAC(k, m) == h ?
```



problem: replay attacks

(adversary could intercept a message, re-send it at a later time)



alice

bob

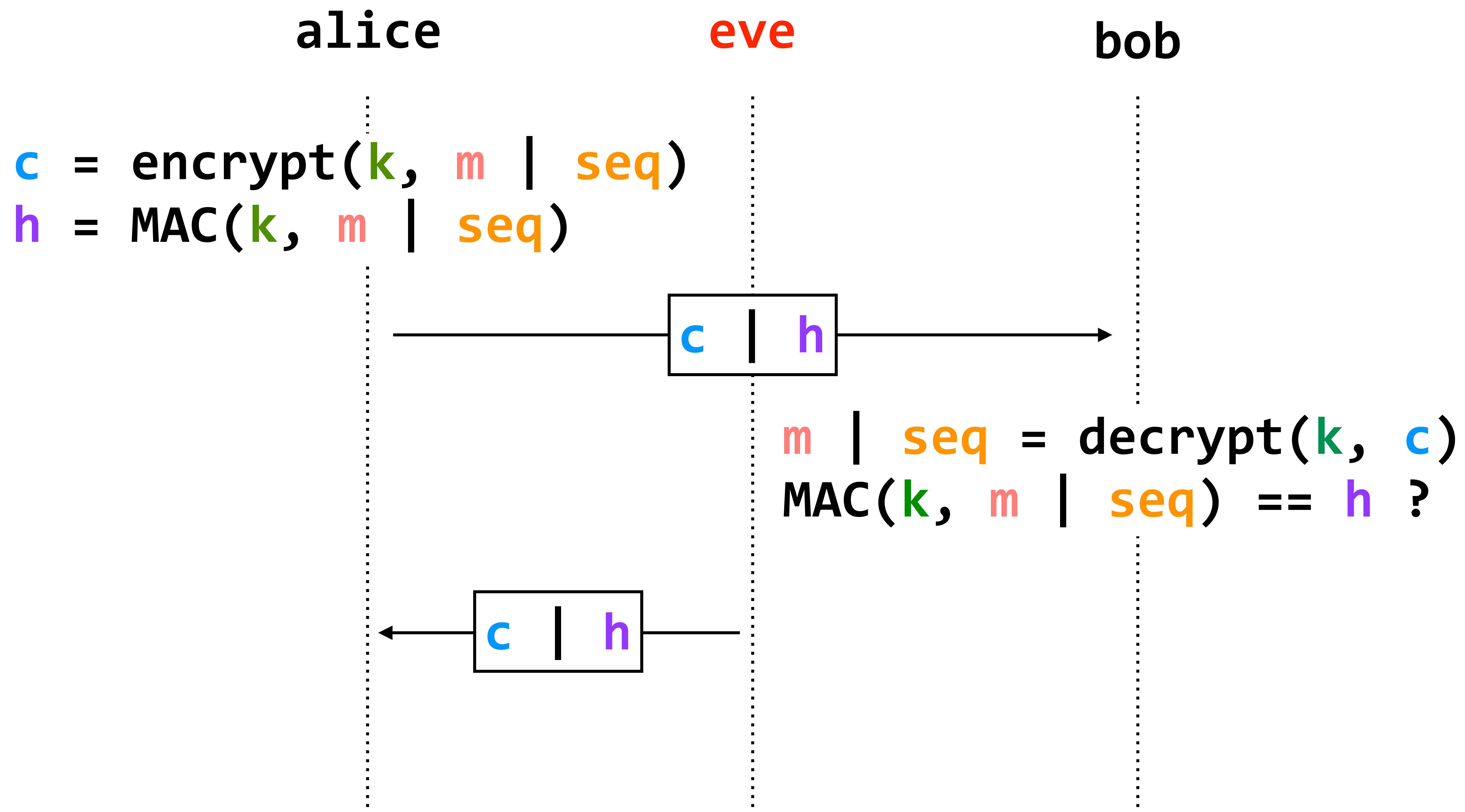
c = encrypt(k, m | seq)

h = MAC(k, m | seq)

c | h

m | seq = decrypt(k, c)

MAC(k, m | seq) == h ?



alice

eve

bob

```
c = encrypt(k, m | seq)
h = MAC(k, m | seq)
```

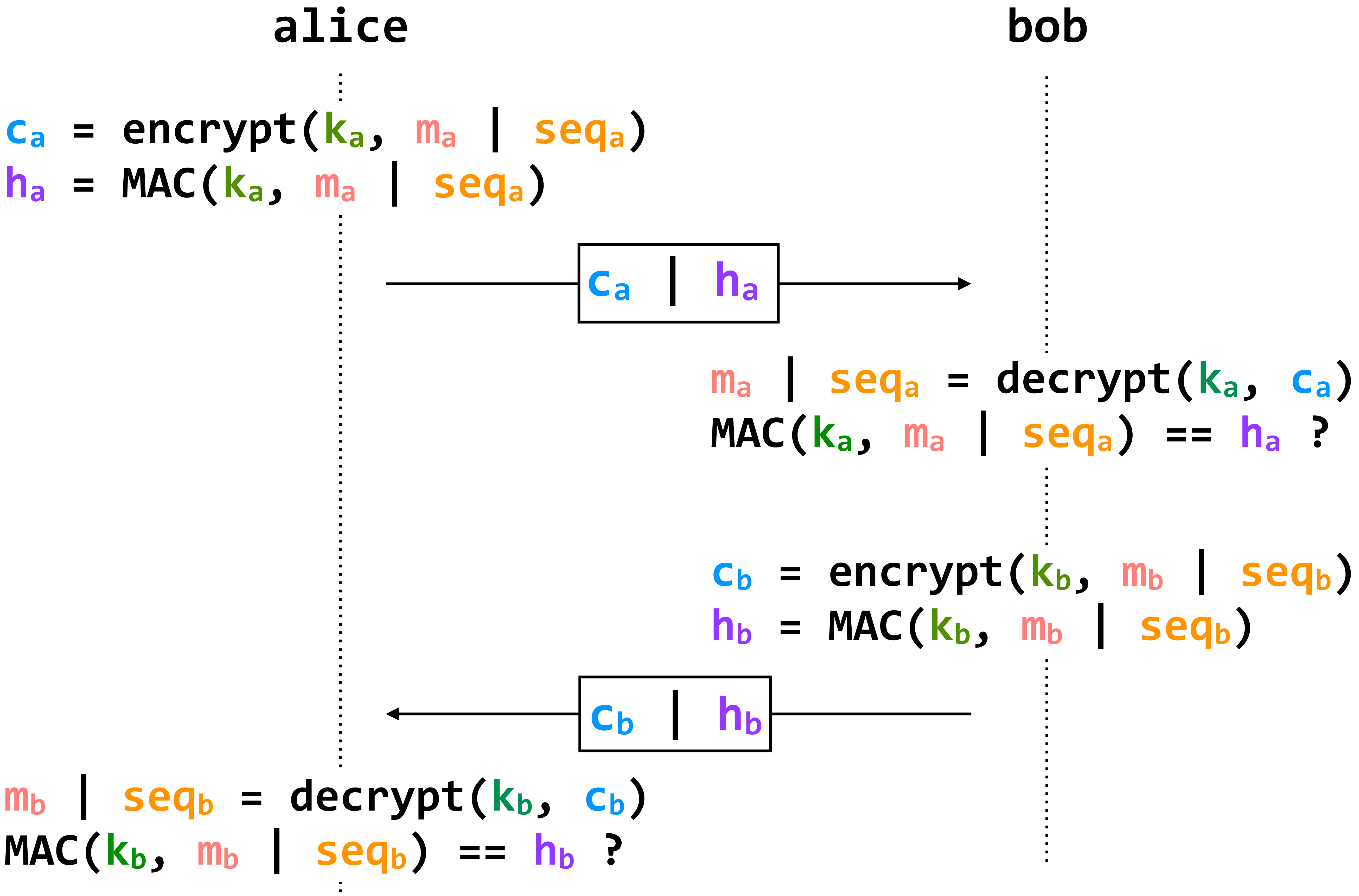
c | h

```
m | seq = decrypt(k, c)
MAC(k, m | seq) == h ?
```

c | h

problem: reflection attacks

(adversary could intercept a message, re-send it at a later time in the opposite direction)



alice

bob

$c_a = \text{encrypt}(k_a, m_a \mid seq_a)$

$h_a = \text{MAC}(k_a, m_a \mid seq_a)$

$c_a \mid h_a$

$m_a \mid seq_a = \text{decrypt}(k_a, c_a)$

$\text{MAC}(k_a, m_a \mid seq_a) == h_a$?

$c_b = \text{encrypt}(k_b, m_b \mid seq_b)$

$h_b = \text{MAC}(k_b, m_b \mid seq_b)$

$c_b \mid h_b$

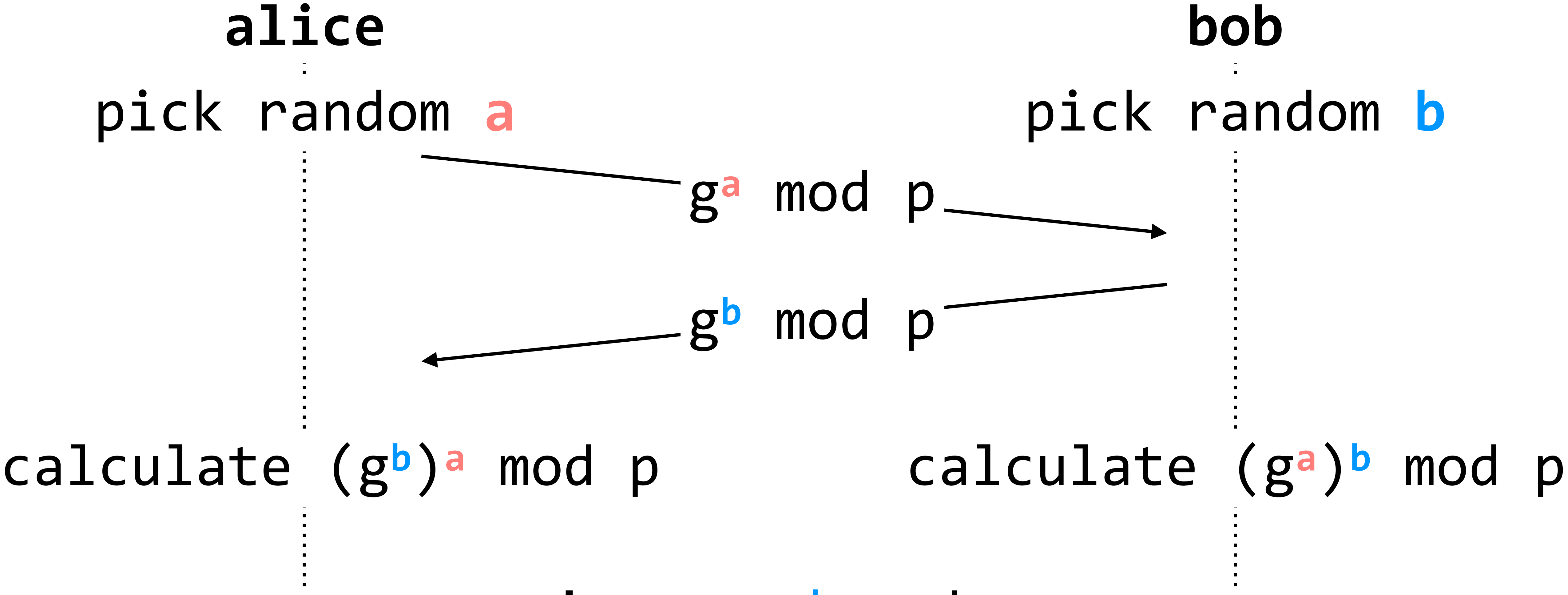
$m_b \mid seq_b = \text{decrypt}(k_b, c_b)$

$\text{MAC}(k_b, m_b \mid seq_b) == h_b$?

problem: how do the parties know the keys?

known: p (prime), g

property: given $\mathbf{g^r \bmod p}$, it is (virtually) impossible to determine **r** *even if* you know **g** and **p**



alice

pick random a

g^a mod p

calculate $(g^b)^a$ mod p

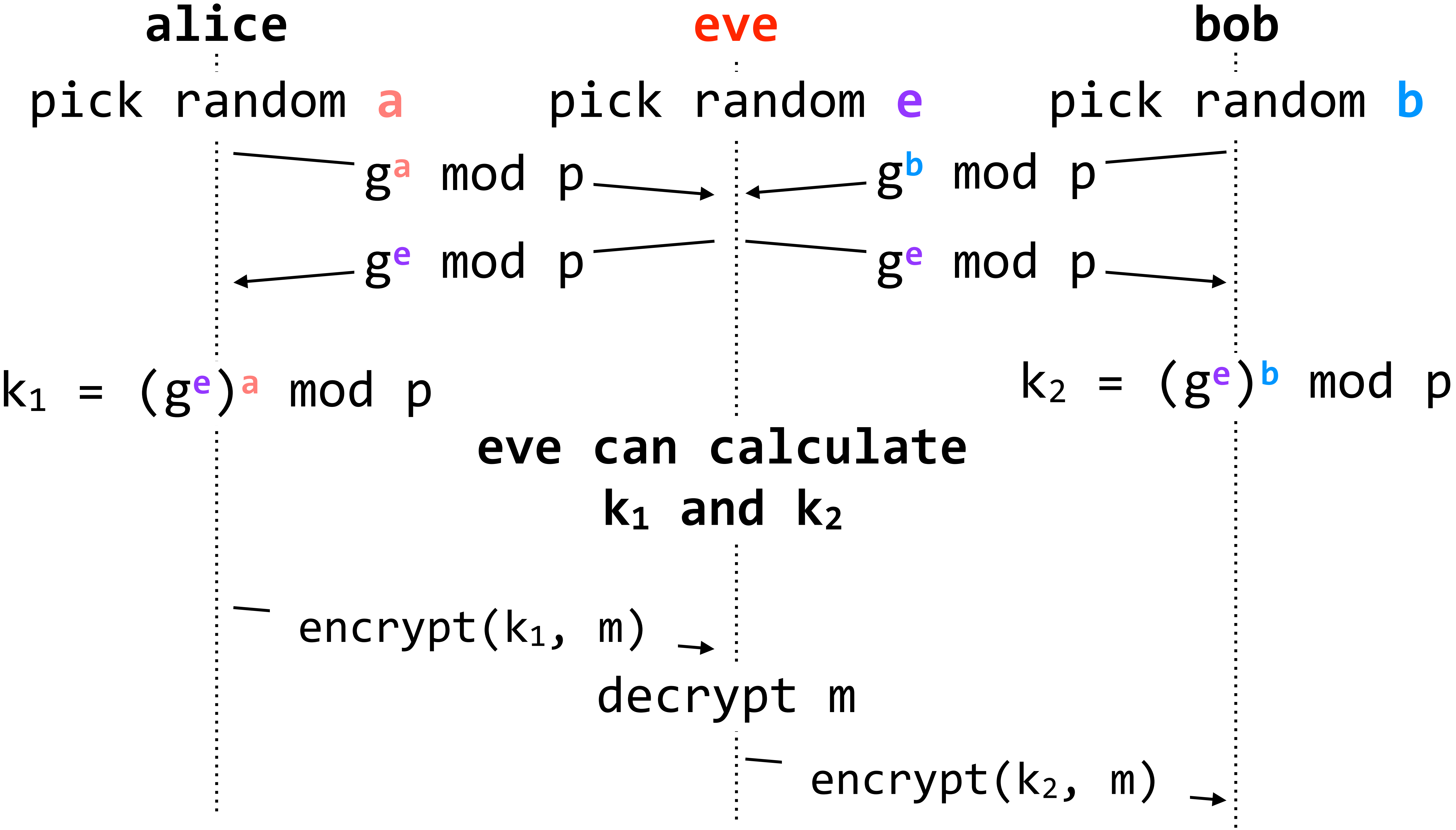
bob

pick random b

g^b mod p

calculate $(g^a)^b$ mod p

key = g^{ab} mod p



alice **eve** **bob**

pick random a

pick random e

pick random b

$g^a \bmod p$

$g^b \bmod p$

$g^e \bmod p$

$g^e \bmod p$

$k_1 = (g^e)^a \bmod p$

$k_2 = (g^e)^b \bmod p$

eve can calculate k_1 and k_2

encrypt(k_1, m)

decrypt m

encrypt(k_2, m)

problem: alice and bob don’t know they’re not communicating directly

cryptographic signatures

allow users to verify identities using public-key cryptography

users generate key pairs

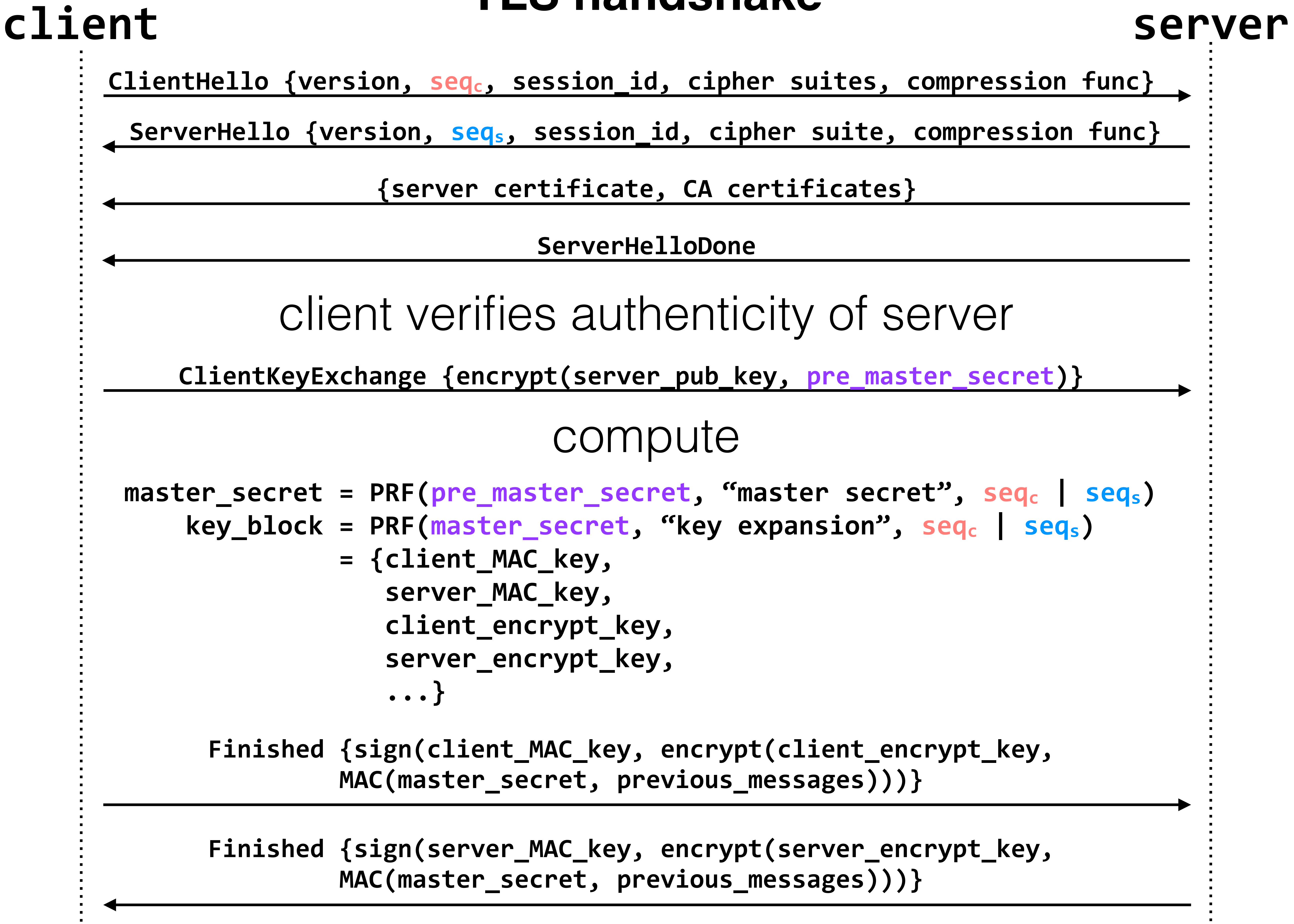
the two keys in the pair are related mathematically

```
{public_key, secret_key}
```

```
sign(secret_key, message) → sig
verify(public_key, message, sig) → yes/no
```

property: it is (virtually) impossible to compute `sig` without `secret_key`

TLS handshake



client

server

ClientHello {version, seq$_c$, session_id, cipher suites, compression func}

ServerHello {version, seq$_s$, session_id, cipher suite, compression func}

{server certificate, CA certificates}

ServerHelloDone

client verifies authenticity of server

ClientKeyExchange {encrypt(server_pub_key, pre_master_secret)}

compute

master_secret = PRF(pre_master_secret, “master secret”, seq$_c$ | seq$_s$)
key_block = PRF(master_secret, “key expansion”, seq$_c$ | seq$_s$)
= {client_MAC_key,
server_MAC_key,
client_encrypt_key,
server_encrypt_key,
...}

Finished {sign(client_MAC_key, encrypt(client_encrypt_key,
MAC(master_secret, previous_messages)))}

Finished {sign(server_MAC_key, encrypt(server_encrypt_key,
MAC(master_secret, previous_messages)))}

- **Secure channels** protect us from adversaries that can observer and tamper with packets in the network.

- Encrypting with **symmetric keys** provides secrecy, and using **MACs** provides integrity. **Diffie-Hellman key exchange** lets us exchange the symmetric key securely.

- To verify identities, we use **public-key cryptography** and cryptographic **signatures**. We often distribute public keys with **certificate authorities**, though this method is not perfect.