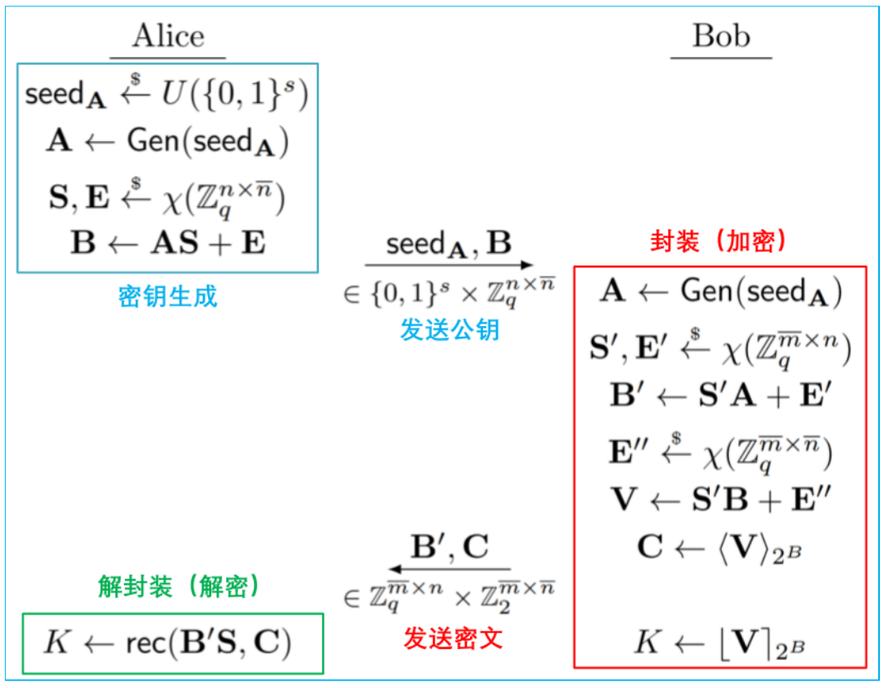
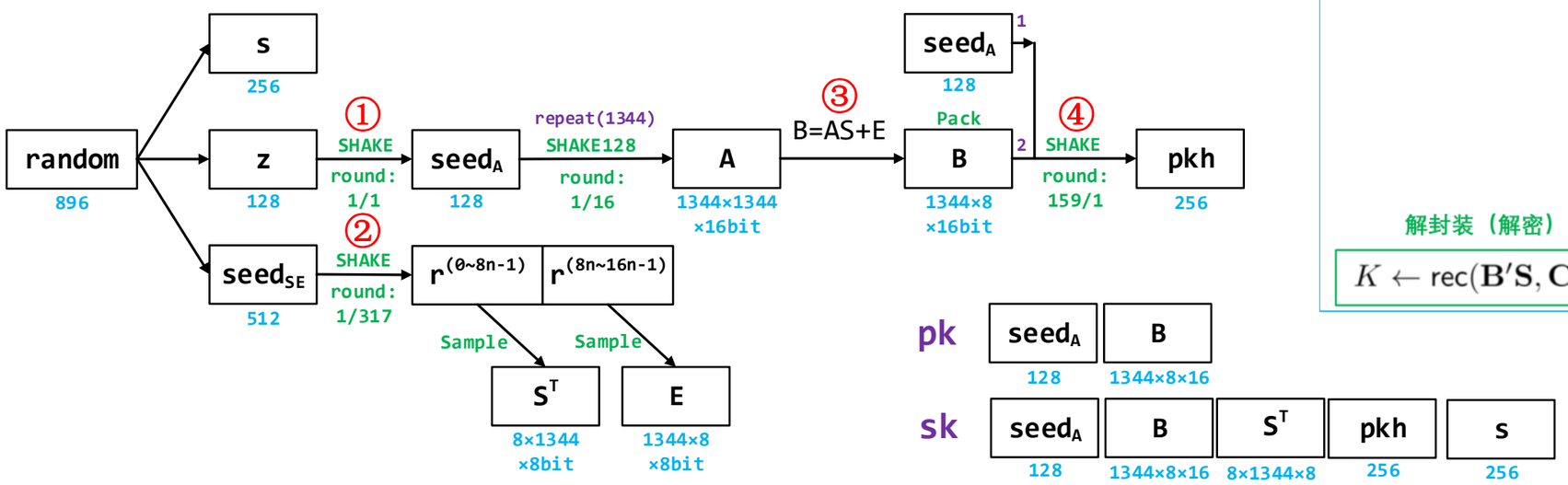
FrodoKEM-1344

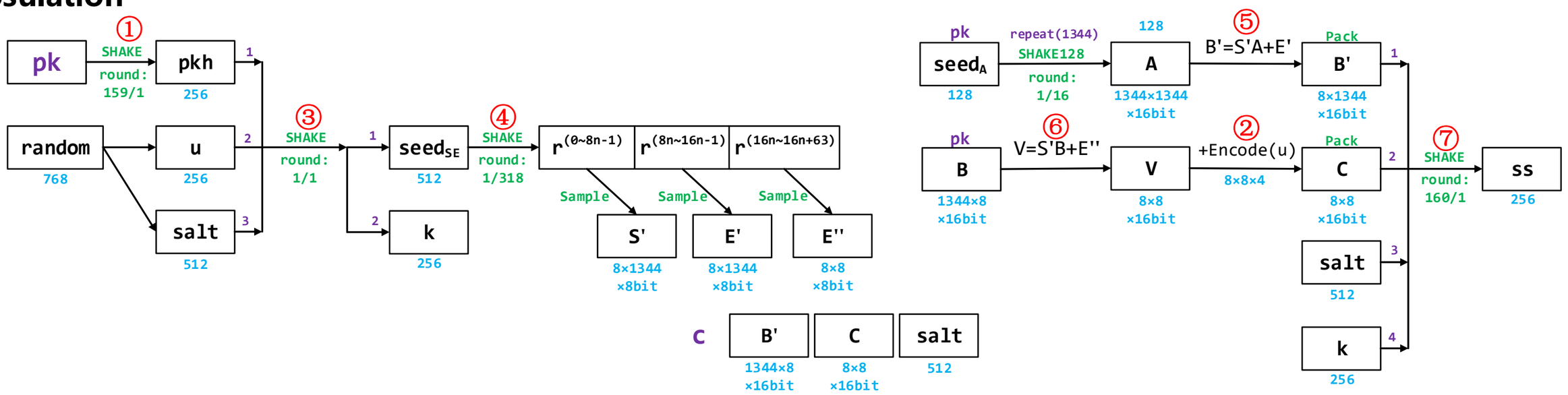
SHAKE
round:
absorb rounds/squeeze rounds



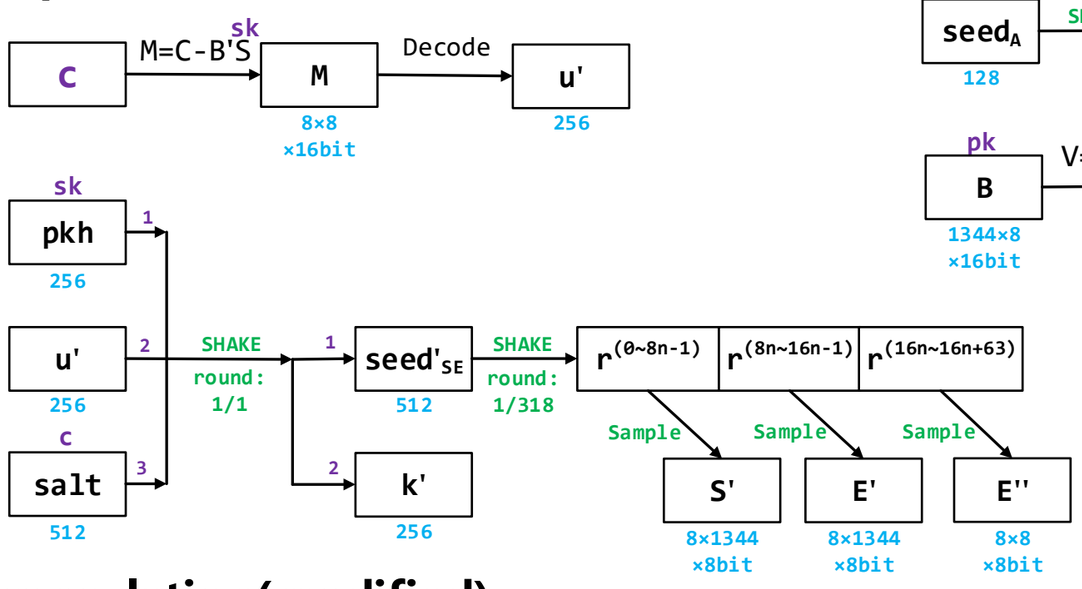
Key generation

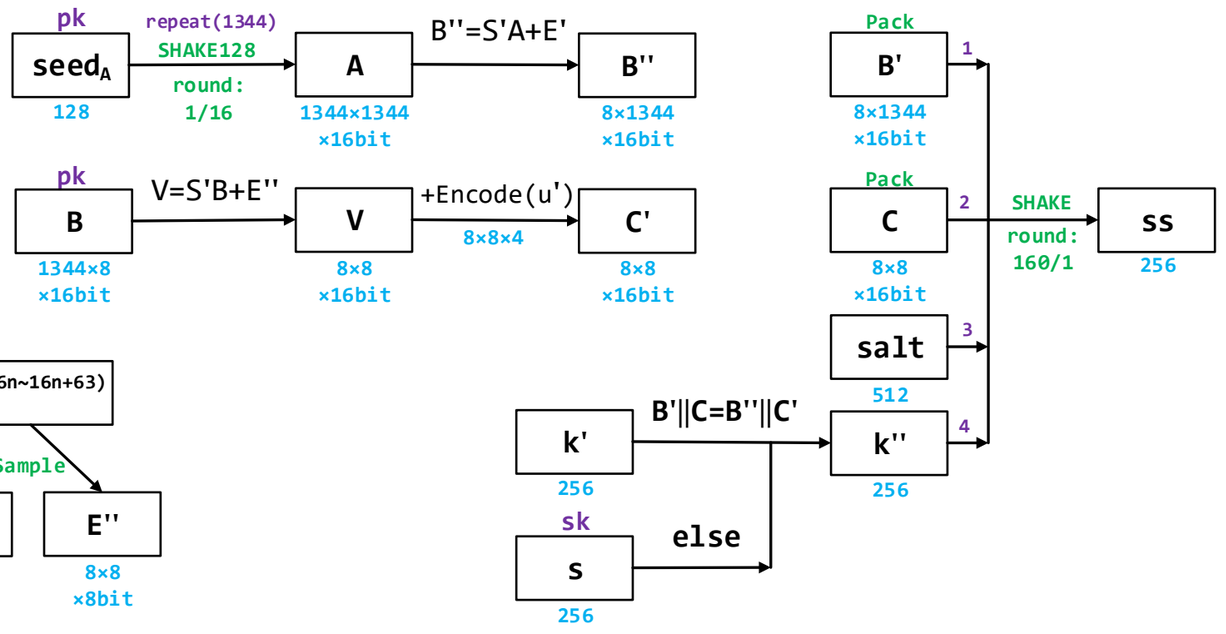


Encapsulation

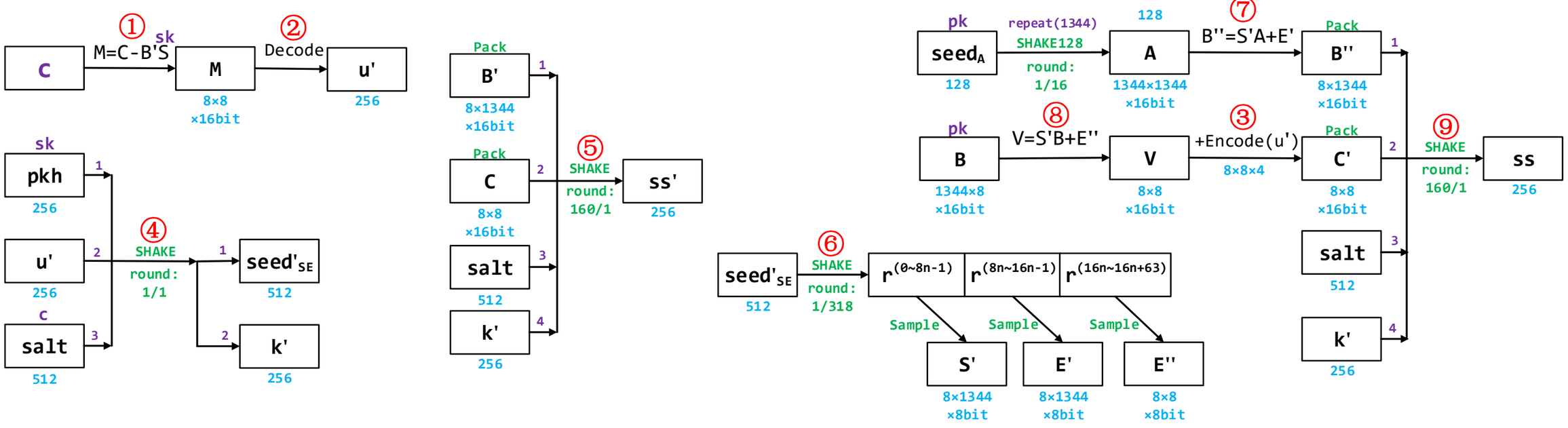


Decapsulation





Decapsulation(modified)



循环方向(从内到外)
第1层
第2层
第3层

版本v4用到了两个4×4的乘法器阵列

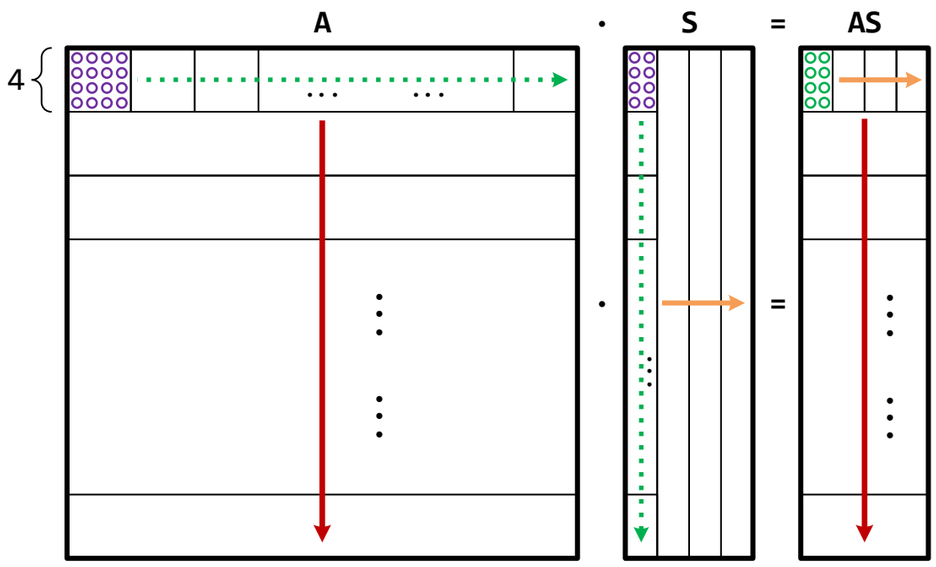
地址安排

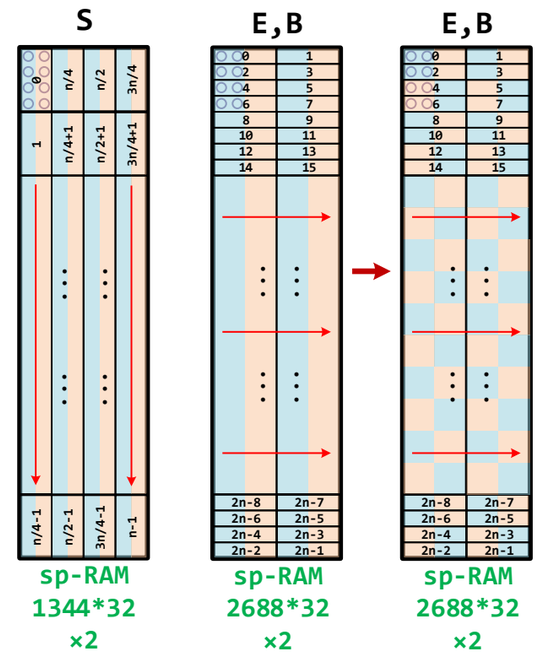
S,B sp-ram 4032*32 ×2
S',B' dp-ram 2016*32 ×4

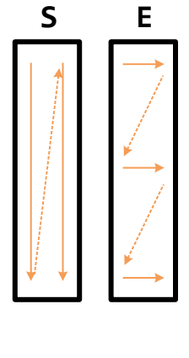
RAM0
RAM1

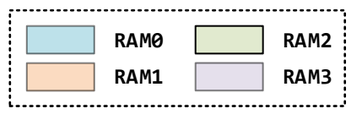
矩阵生成

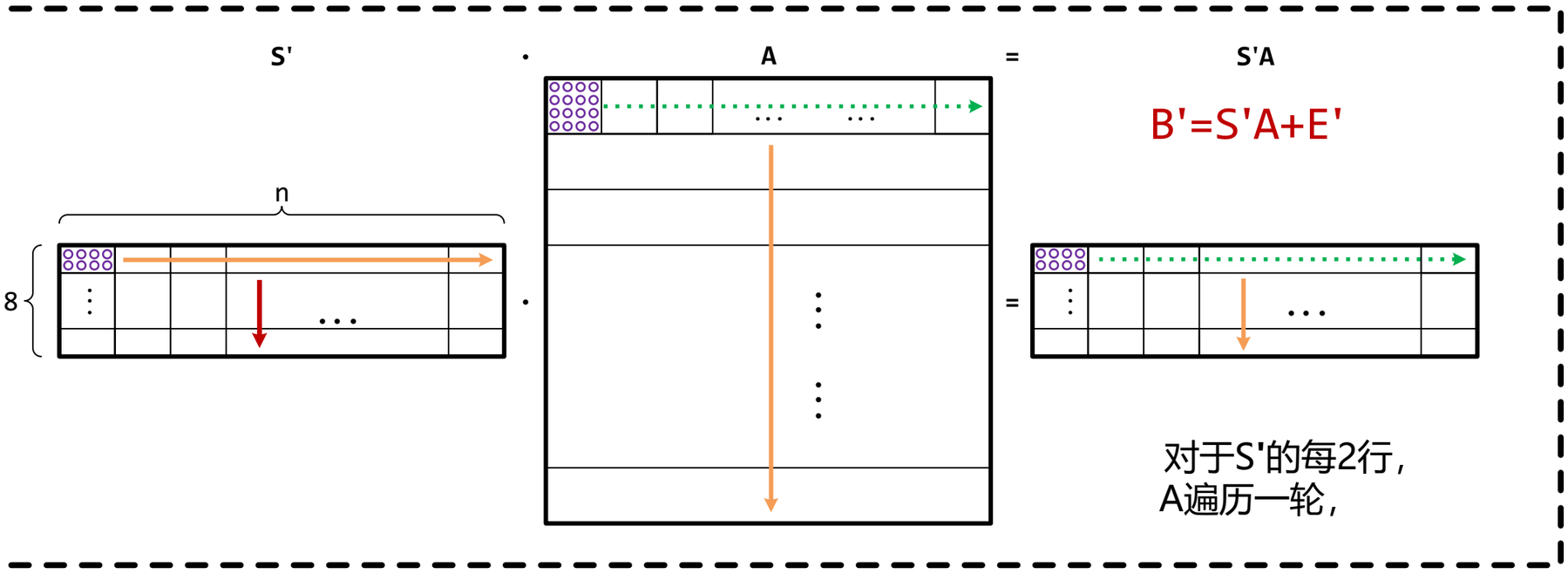
矩阵生成

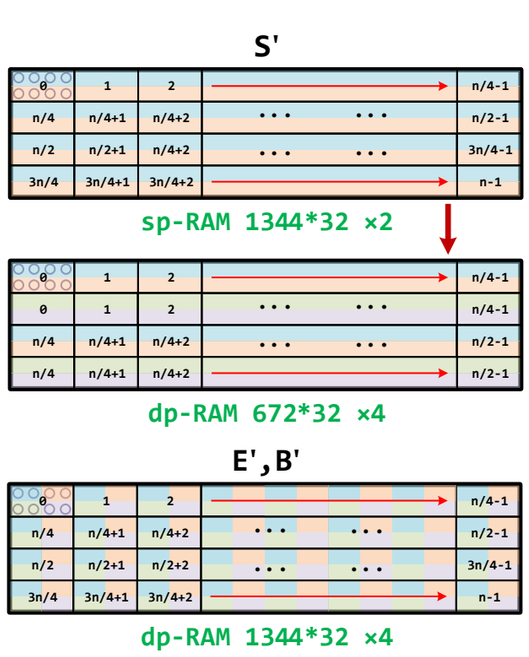


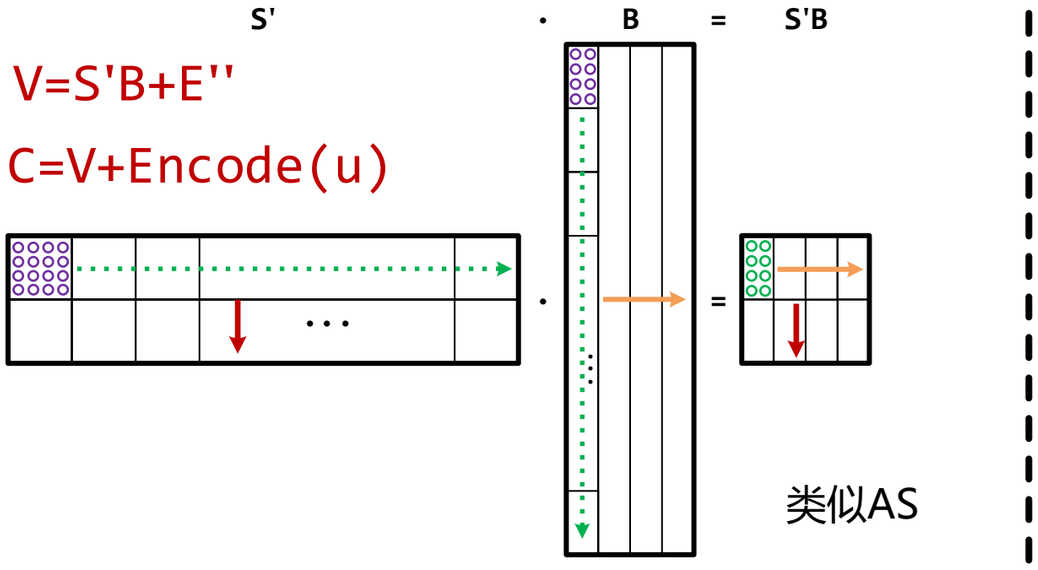


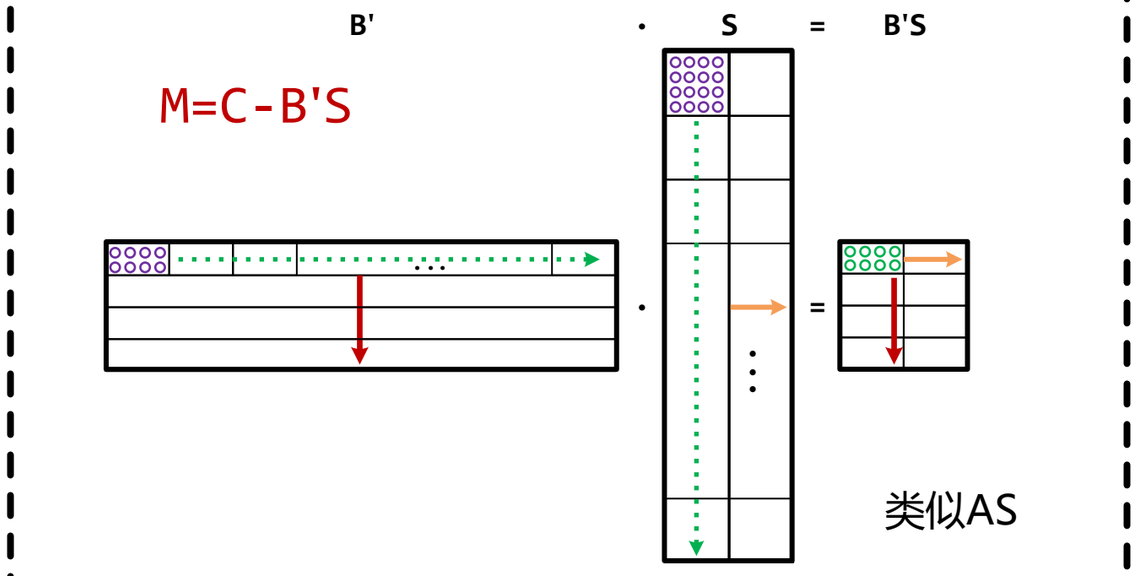






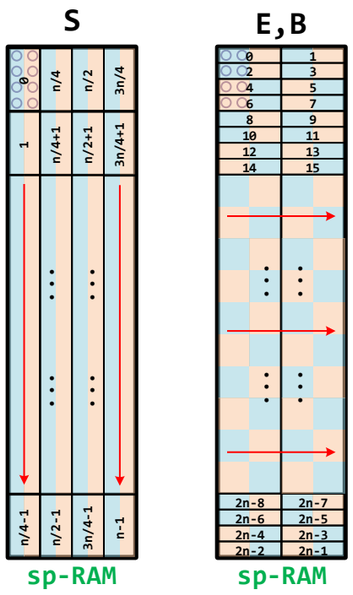


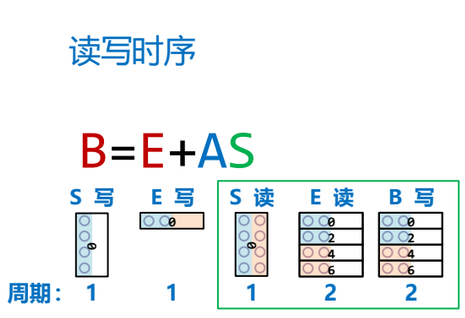


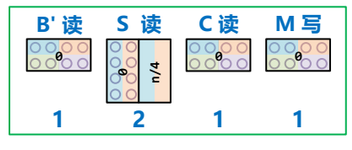


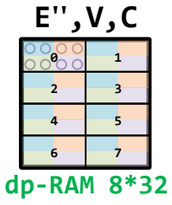
sp-ram 4032*32×2

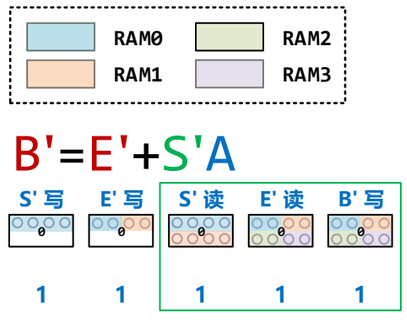
读写时序

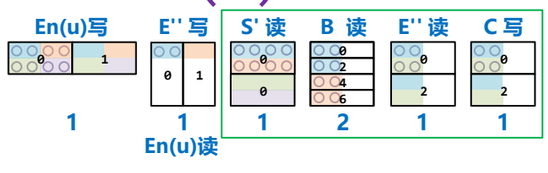




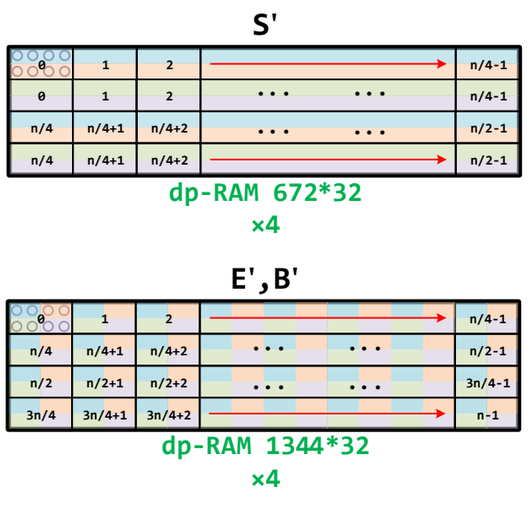




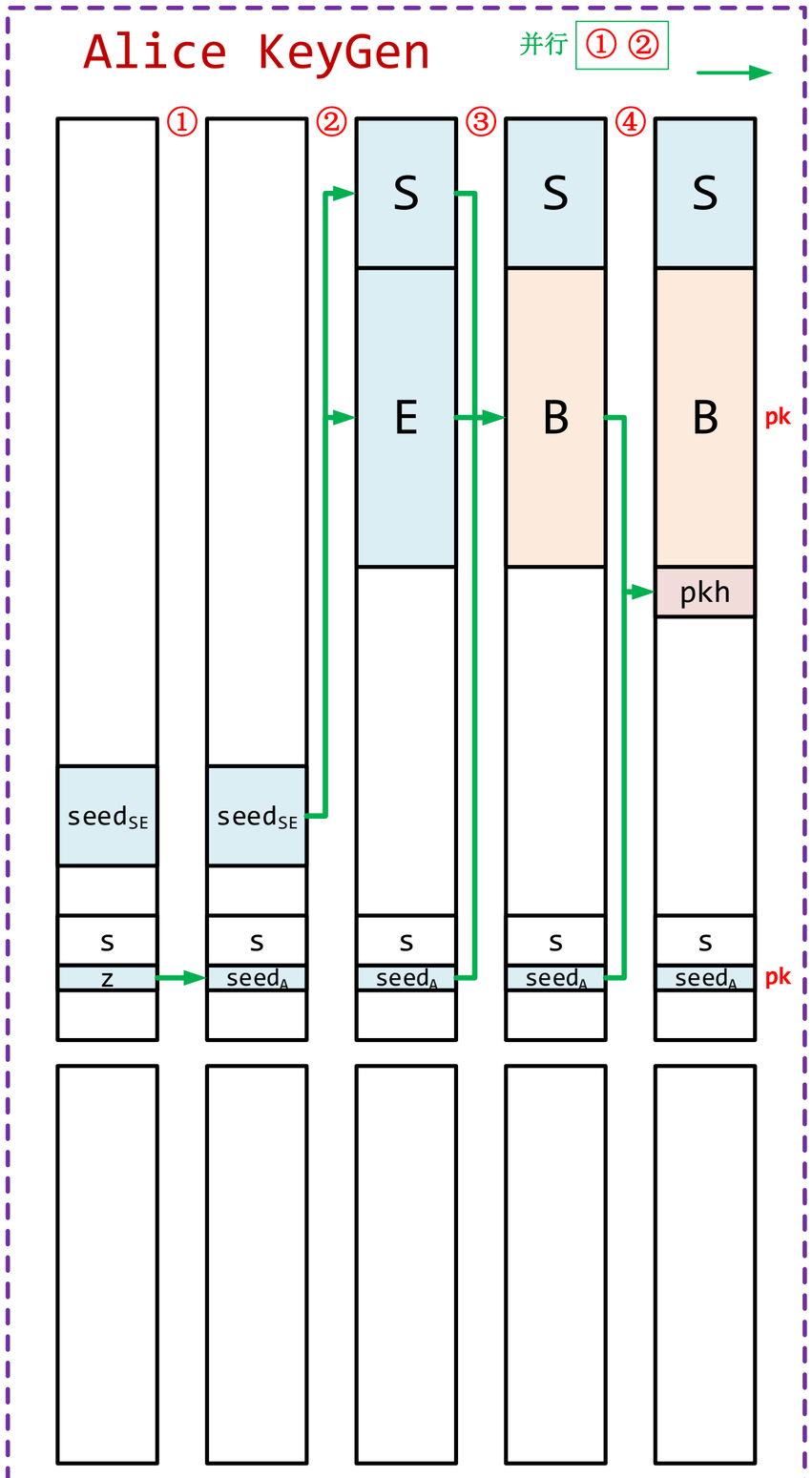




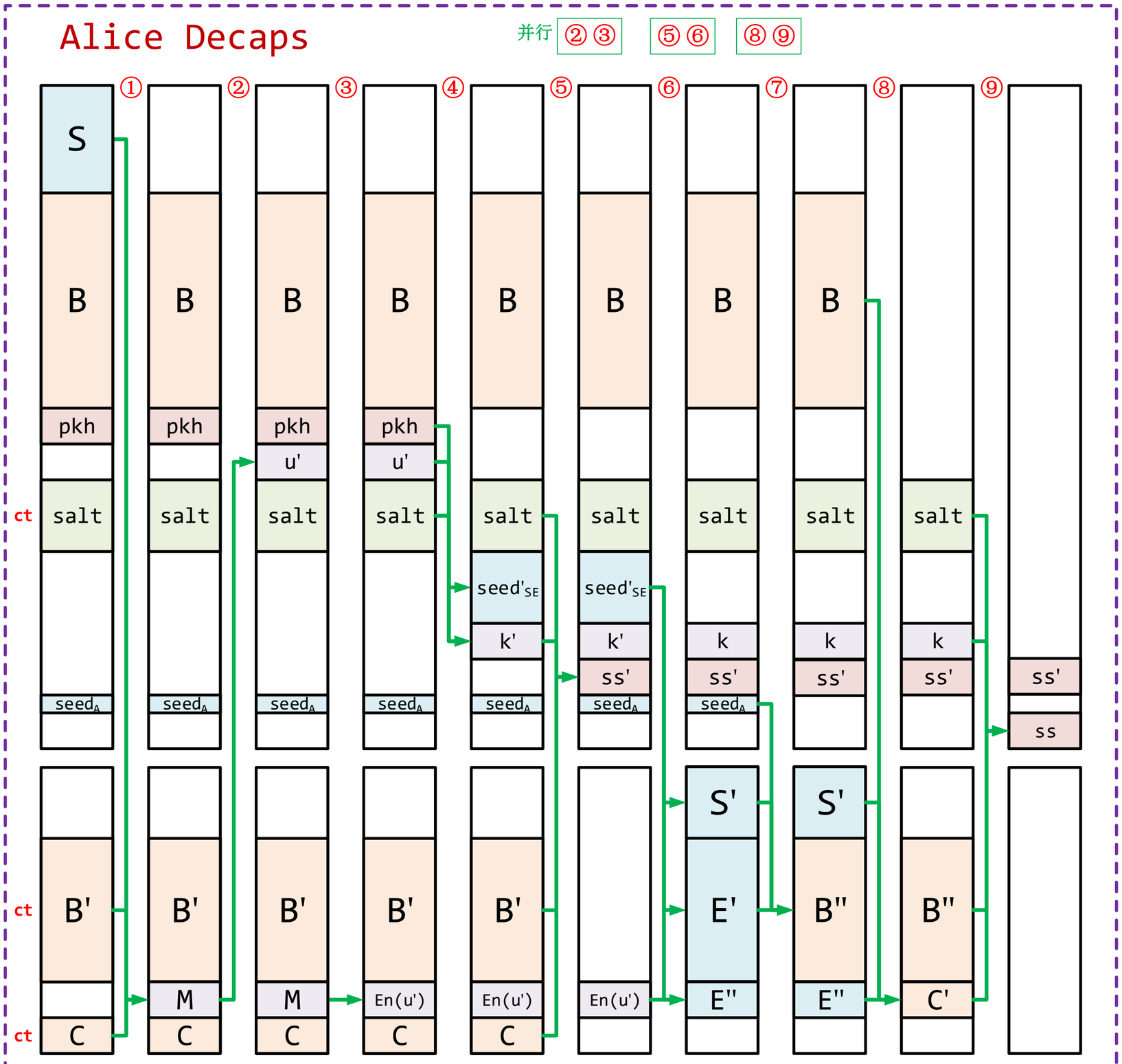
dp-ram 2024*32×4



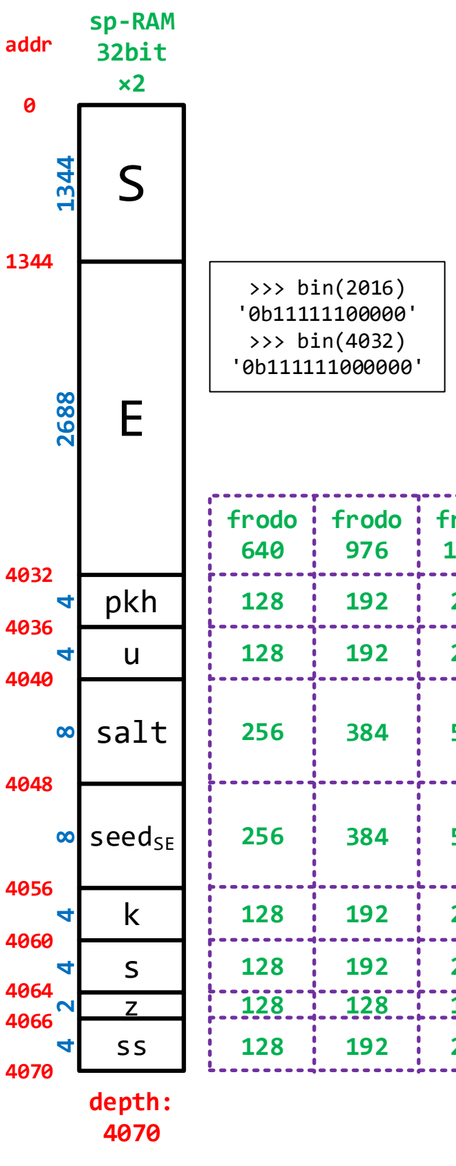
Alice KeyGen



Alice Decaps



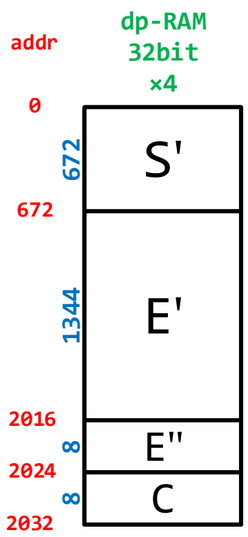
sp-RAM 32bit ×2



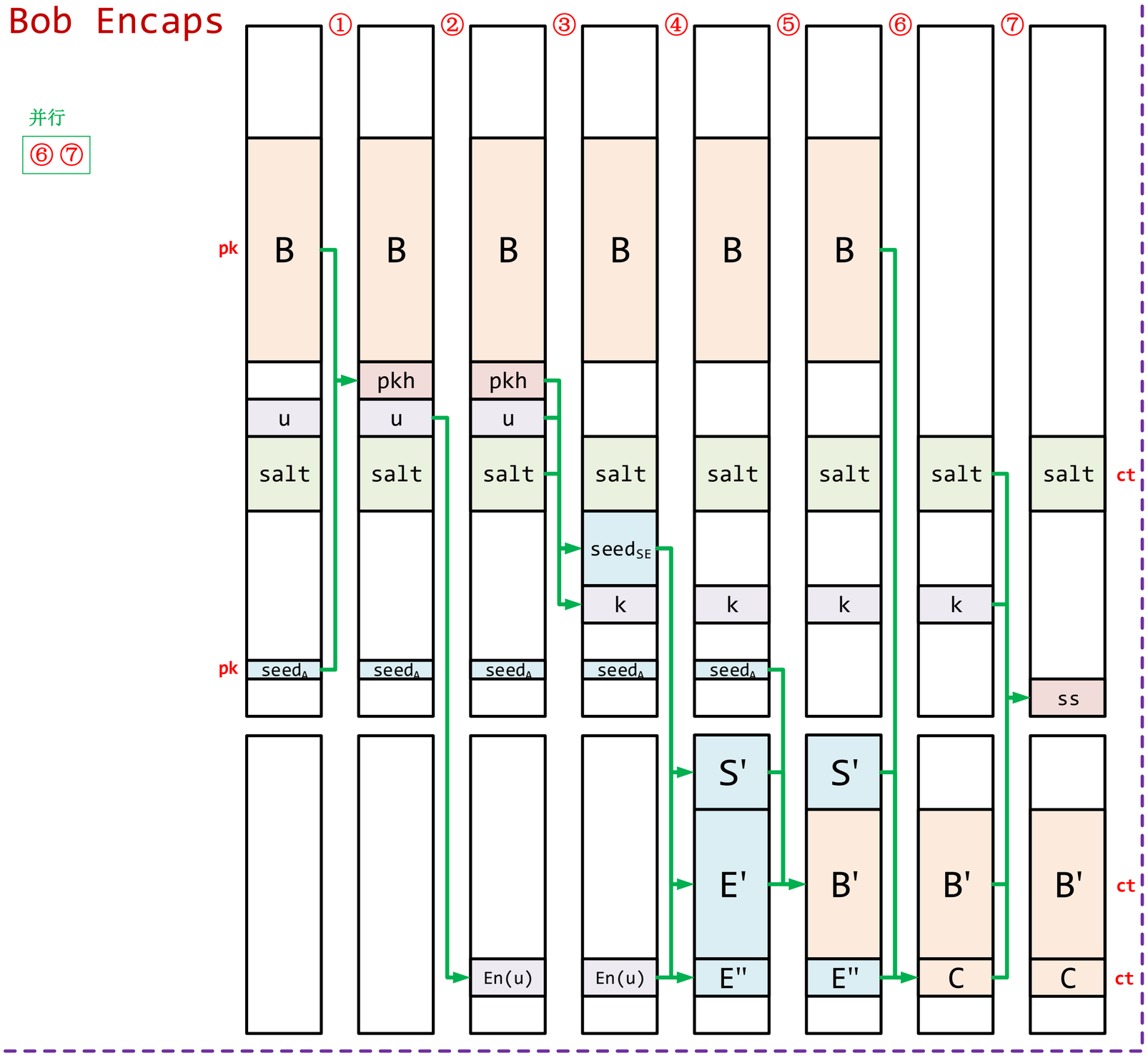

```
>>> bin(2016)
'0b11111100000'
>>> bin(4032)
'0b111111000000'
```

frodo 640	frodo 976	frodo 1344
128	192	256
128	192	256
256	384	512
256	384	512
128	192	256
128	192	256
128	128	128
128	192	256

dp-RAM 32bit ×4



Bob Encaps



RAM width : 64
round: 吸收/挤压轮数
cycle: 读写RAM的周期

2024/8/26

t →

Key generation

SHAKE
reg 128bit
z → $seed_A$

round: 1/1

SHAKE
RAM 1344*64
生成S存入RAM

round: 1/159
cycle: 1344/4*8

(159*24=3816)

SHAKE
RAM 1344*128
生成E存入RAM

round: 1/159
cycle: 1344/4*8

SHAKE × 4
$seed_A$ → A
32个16*5乘法器
生成A，计算B

生成A的4行
1 | 2 | ... | 1344/4

按列遍历S
1 | 2 | ... | 1344/4

cycle: 1344/4*8

存B前4行
1 | 2 | ... | 336

存B，
存到E的位置

最后4行

SHAKE
生成pkh

round: 159/1
cycle: 1344/4*8

Encapsulation

SHAKE
生成pkh

round: 159/1
cycle: 1344/4*8

SHAKE
reg 512bit
生成$seed_{SE}$

round: 1/1

SHAKE
RAM 1344*64
生成S'存入RAM

round: 1/159
cycle: 1344/4*8

SHAKE
RAM 1344*128
生成E'存入RAM

round: 1/159
cycle: 1344/4*8

SHAKE
reg 8*8*5
生成E''存入reg
提前+Encode(u)

round: 1/1

多次遍历S'的前4列
1 | 2 | ... | 1344/4

cycle: 1344/4*8

边取边存B'
1 | 2 | ... | 336

取、累加、存，
存到E'的位置

SHAKE × 4
$seed_A$ → A
32个16*5乘法器
生成A，计算B'

生成A的4行
1 | 2 | ... | 1344/4

32个16*5乘法器
计算C

B前4行
1 | ... | 1344/4

S'前4列 S'前4列
1 | 1

cycle:6

最后1轮
336

存C
1

B前4行
1344/4 | 1344/4

SHAKE
生成ss

round: 160/1
cycle: 1344/4*8

Decapsulation

重复Decap部分

16个16*5乘法器
仿照计算B'S，计算M

S前4行
1 | ... | 1344/4

B'前4列 B'前4列
1 | 1 | 1344/4 | 1344/4

cycle:6

存M
1 | 336

计算u'

SHAKE
reg 512bit
生成$seed_{SE}$

round: 1/1

SHAKE
RAM 1344*64
生成S'存入RAM

round: 1/159
cycle: 1344/4*8

SHAKE
RAM 1344*128
生成E'存入RAM

round: 1/159
cycle: 1344/4*8

SHAKE
reg 8*8*5
生成E''存入reg
提前+Encode(u)

round: 1/1

SHAKE × 4
$seed_A$ → A
32个16*5乘法器
生成A，计算B'

生成A的4行
1 | 2 | ... | 1344/4

多次遍历S'的前4列
1 | 2 | ... | 1344/4

cycle: 1344/4*8

边取边存B'
1 | 2 | ... | 336

取、累加、存，
存到E'的位置

32个16*5乘法器
计算C

B前4行
1 | ... | 1344/4

S'前4列 S'前4列
1 | 1

cycle:6

最后1轮
336

存C
1

B前4行
1344/4 | 1344/4

SHAKE
生成ss

round: 160/1
cycle: 1344/4*8