$N=3$

$$2^3 = 8 \left\{ \begin{array}{lll} 000 & \rightarrow & 110 \\ 001 & \rightarrow & 000 \\ 010 & \rightarrow & \\ 011 & & \\ 100 & & \\ 101 & & \\ 110 & & \\ 111 & & \end{array} \right.$$

possible permutations $= 2^3! = 8!$

Key Space = 64-bit, 2^{64} possible keys.

Block size = ~~64 bits~~ 128 bits, 2^{128} possible inputs

→ $2^{128}!$ possible permutations,

→ Block cipher defines 2^{64} permutations over 128-bit inputs.

$\underbrace{\text{Block cipher}}_{\text{AES, DES}}$

Key space = ~~8~~ ~~[illegible]~~ 2-bits long = {00, 01, 10, 11}

BC

Block size = 3 bits

Key: 00

input	output
000	111
001	000
010	001
011	011
100	010
101	110
110	101
111	100

Key: 01

input	output
000	101
001	110
010	111
011	100
100	011
101	001
110	010
111	000

BC(01, 001) = 110

decrypt w/ key 01

100 011 000 $\Rightarrow$ 011 100 111

attacker sees ^

Brute force attack

For k = 00, 01, 10, 11

try to decrypt w/ key K

Kerchoff Principle: Attacker knows algorithm.

keys = 2^{128}.

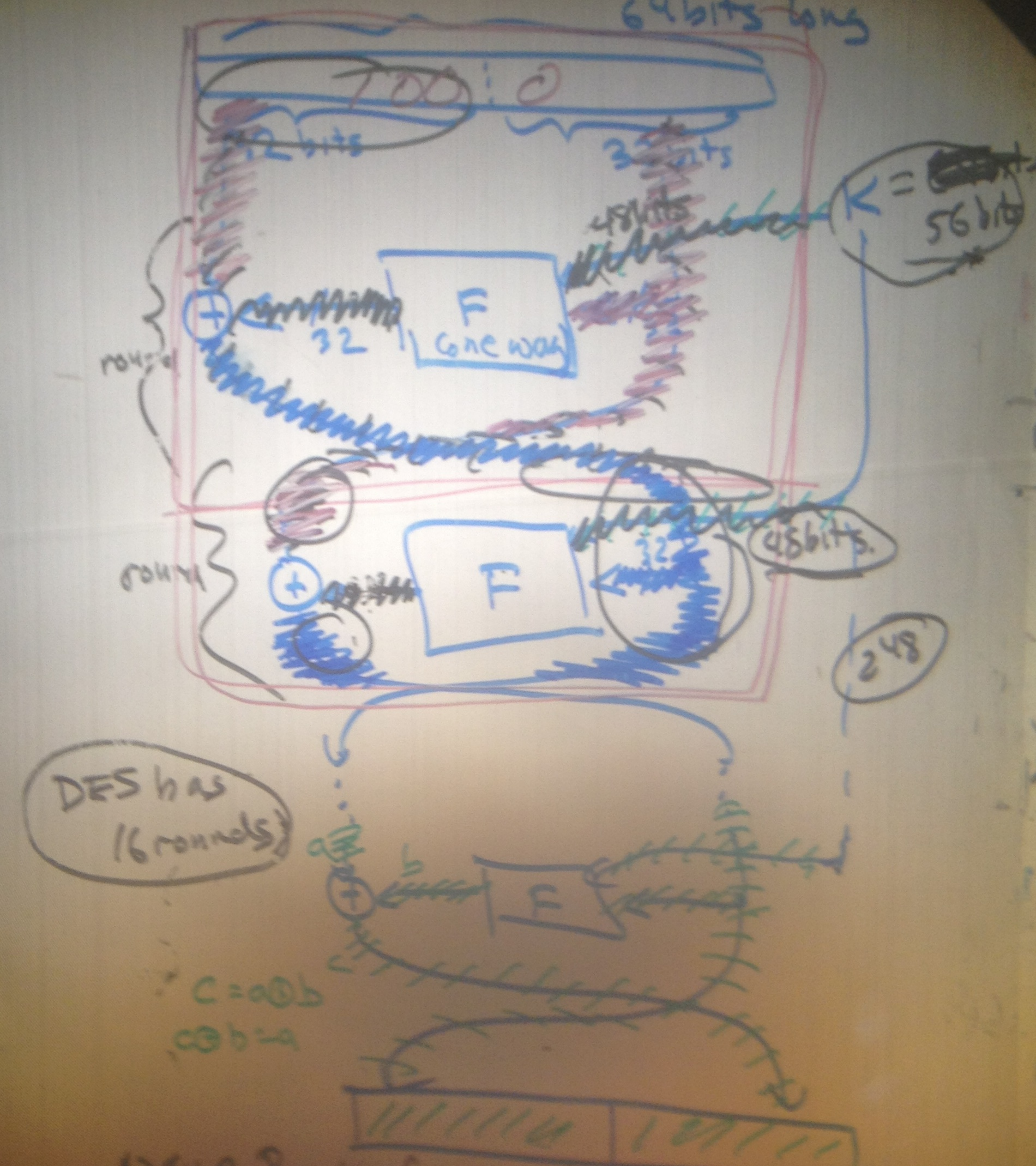
DES = keys = 56-bits $\rightsquigarrow 2^{56}$ keys

Message = 64 bits long 2^{64} messages

total # of perms possible $2^{64}!$

DES defines 2^{56} permutations

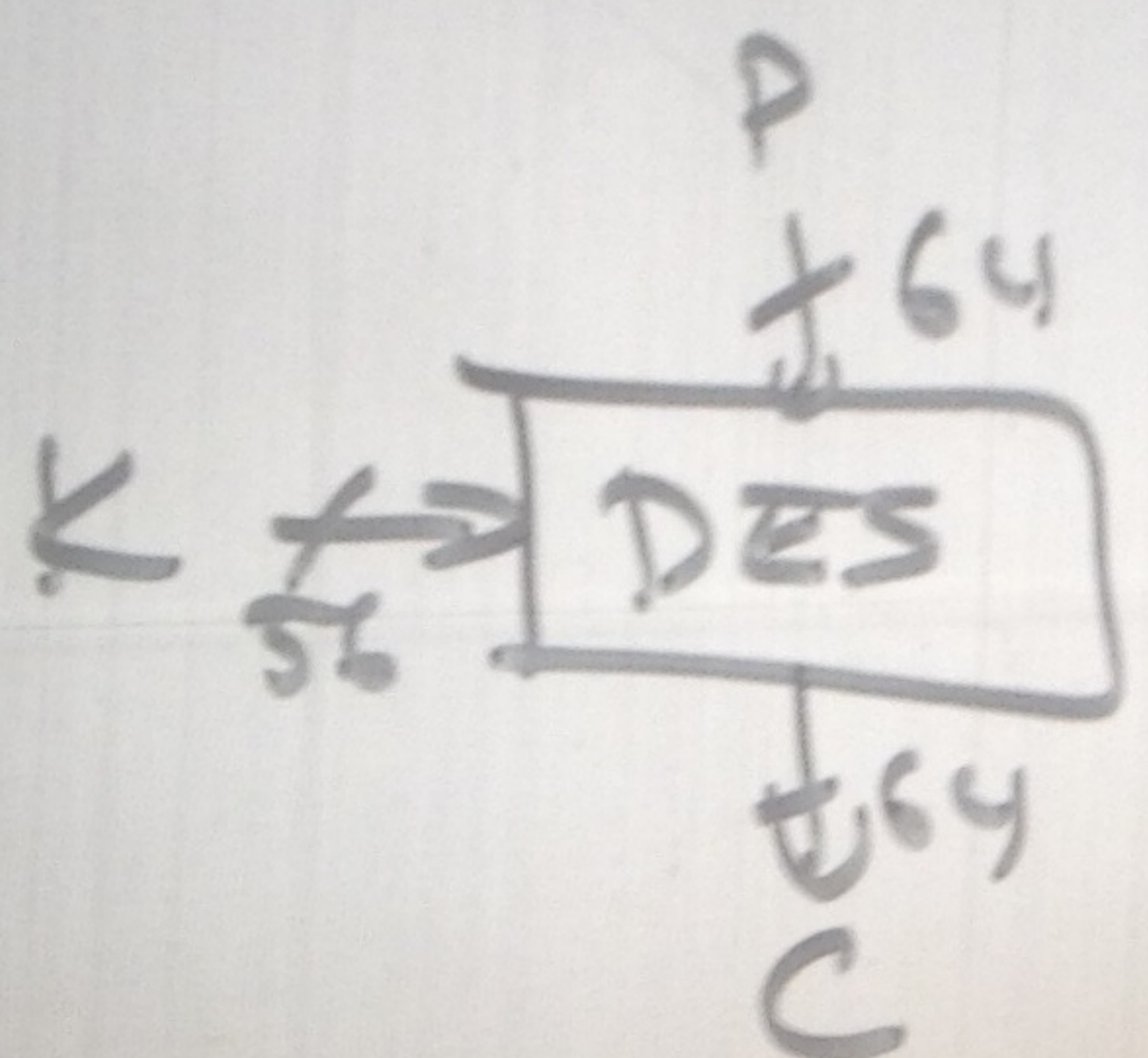
64 bits long



64 bits long
TDO
0
32 bits
32 bits
48 bits
K = 56 bits
F
one way
32
round
round
48 bits
F
32
2 48
DES has 16 rounds
a
b
F
c = a ⊕ b
c ⊕ b = a

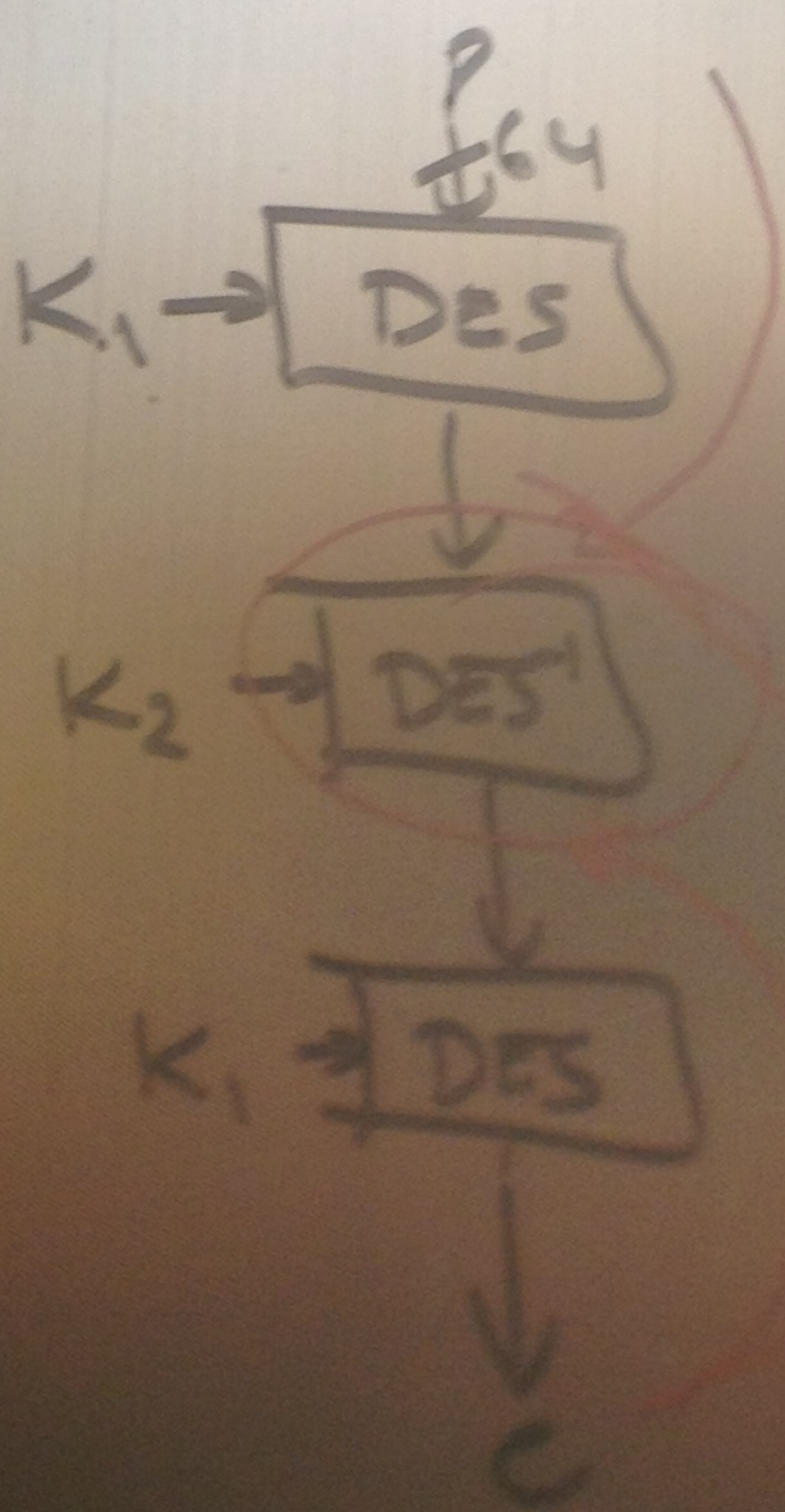
Naive Brute force attack: try 2^{56} keys

If just 2 rounds long: try 2^{98} key to learn the round keys.



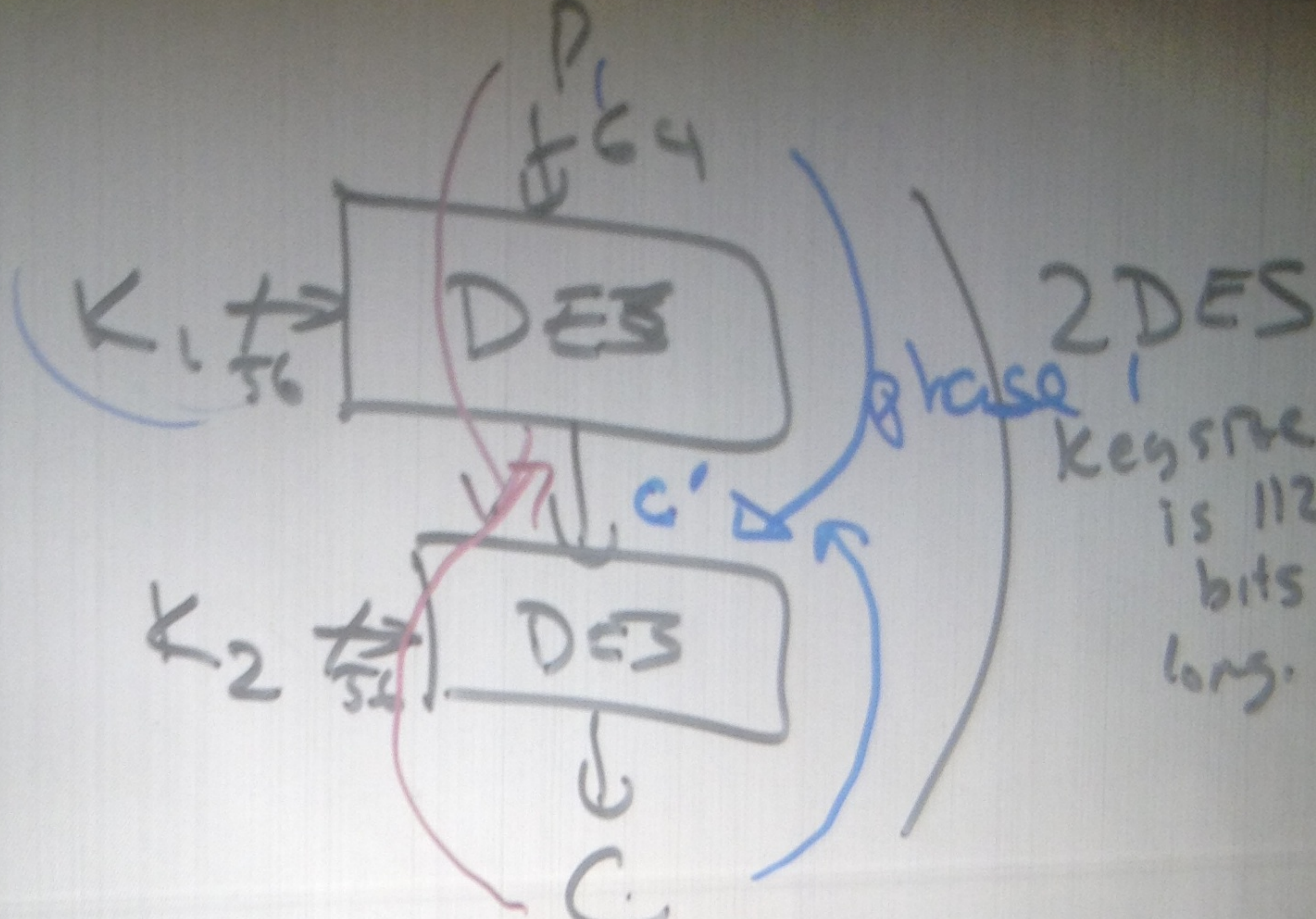
DES

Ksize = 56 bits



3DES

keysize = 112 bits



c'	K
—	—
—	—

Goal: 112-bits of security.

Want brute force attack to take 2^{112} max tries, 2^{111} avg tries.

Claim: Break 2DES w/ ~2^{56} DES operations
2^{57}

Input ~~Attack~~ knows (P_1, C_1), (P_2, C_2)

Alg: phase 1
- For all keys K in set of 56-bit DES keys
  - compute $C' = DES(K, P_1)$
  - store in HashTable K w/ index C'
- For all keys K' in set of 56-bit DES keys

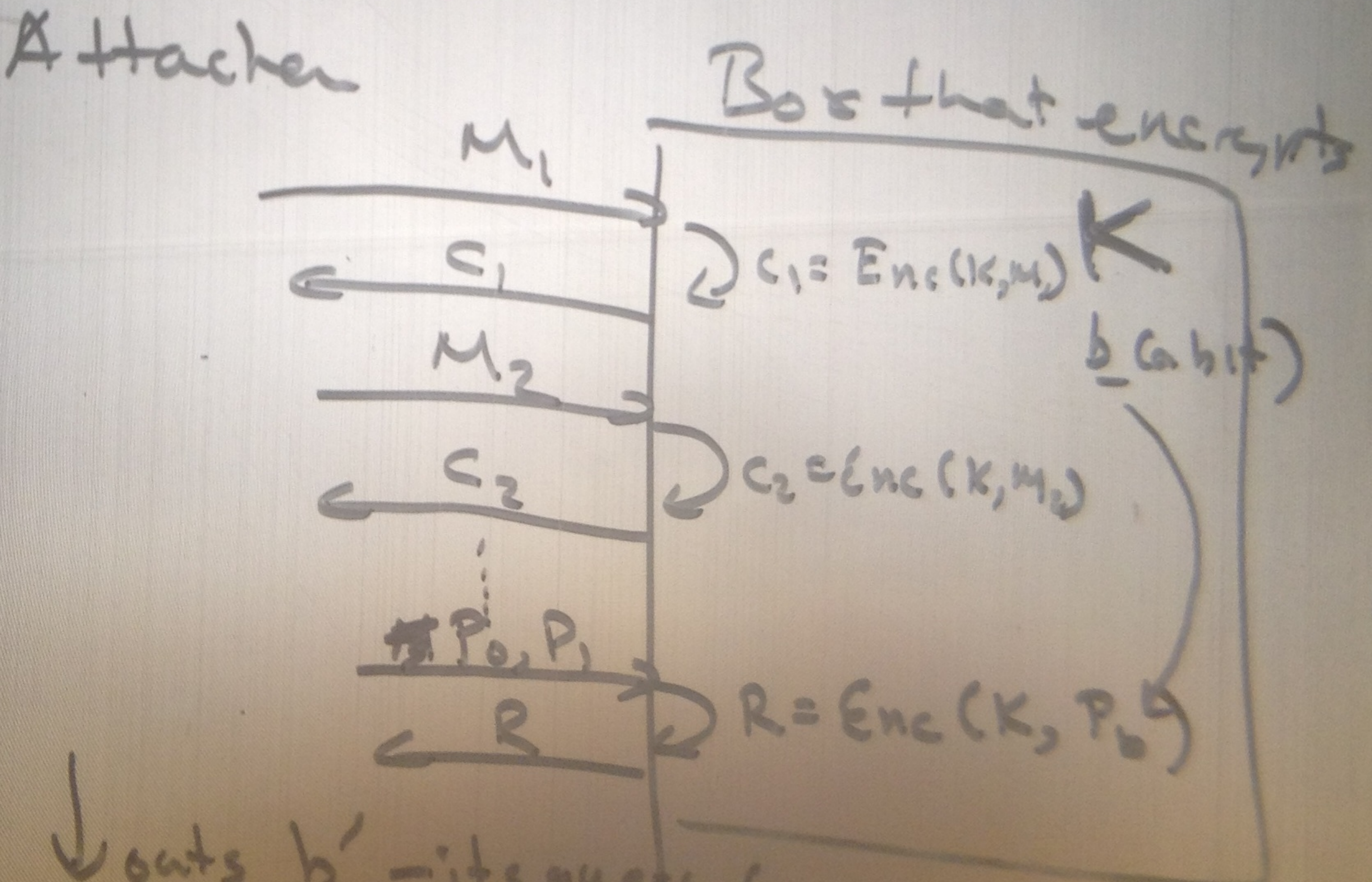
For all keys K' in set of 56-bit DES keys

compute $C'' = DES^{-1}(K', C_2)$

look up C'' in Hash Table.

if $\exists$ in Hash table, $K_2 = K'$

$K_1 = \text{HashTable}(C'')$.



outs b' – its guess for the value b.

Enc scheme secure against this attacker if $b' == b$ w/ prob close to $\frac{1}{2}$ (random guessing).

that's why we need random #s.