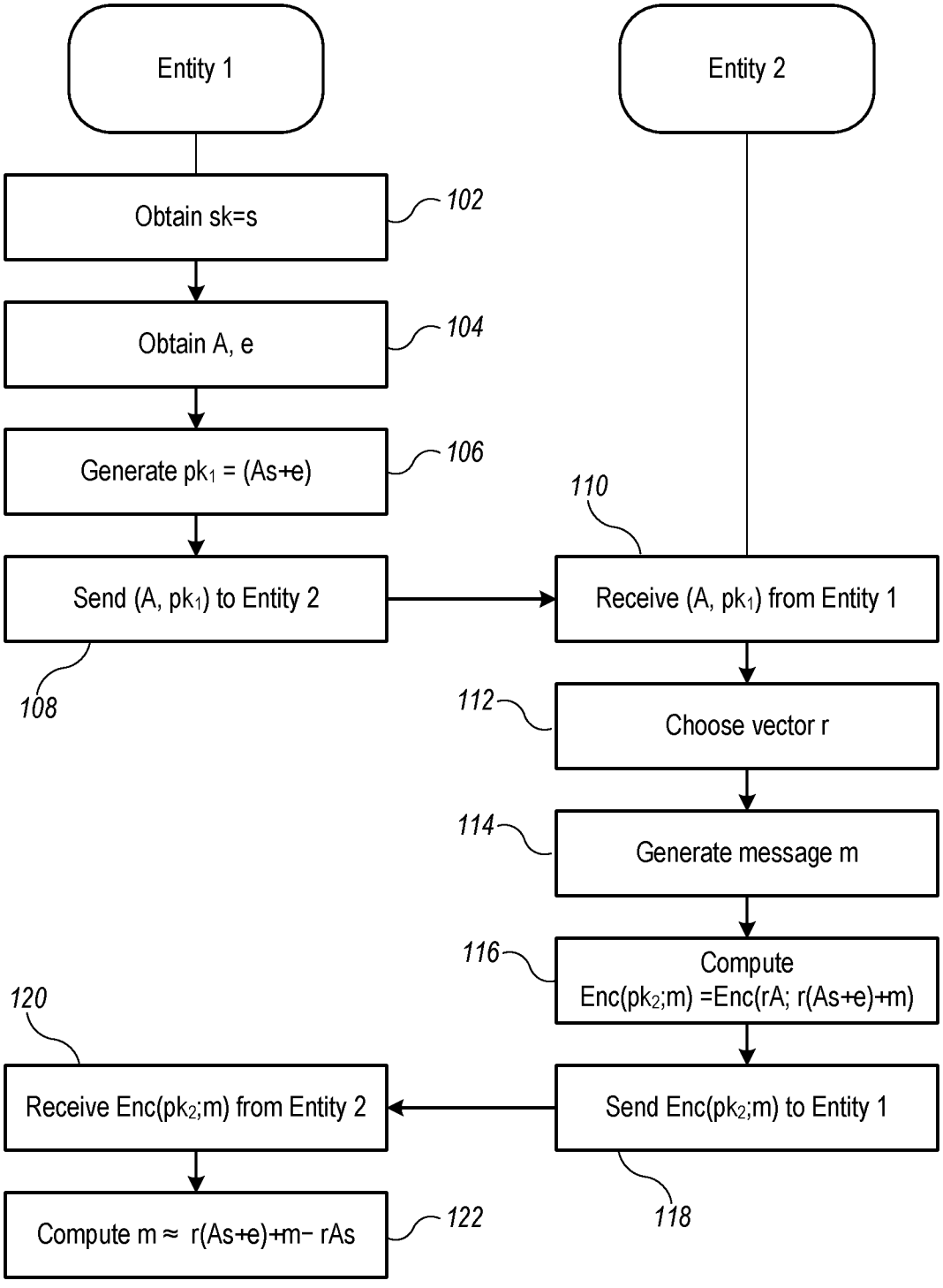

Entity 1
Entity 2
Obtain sk=s
102
Obtain A, e
104
Generate $pk_1 = (As+e)$
106
110
Send (A, pk_1) to Entity 2
Receive (A, pk_1) from Entity 1
108
112
Choose vector r
114
Generate message m
116
Compute
Enc(pk_2;m) =Enc(rA; r(As+e)+m)
120
Receive Enc(pk_2;m) from Entity 2
Send Enc(pk_2;m) to Entity 1
118
Compute m ≈ r(As+e)+m− rAs
122

FIG. 1

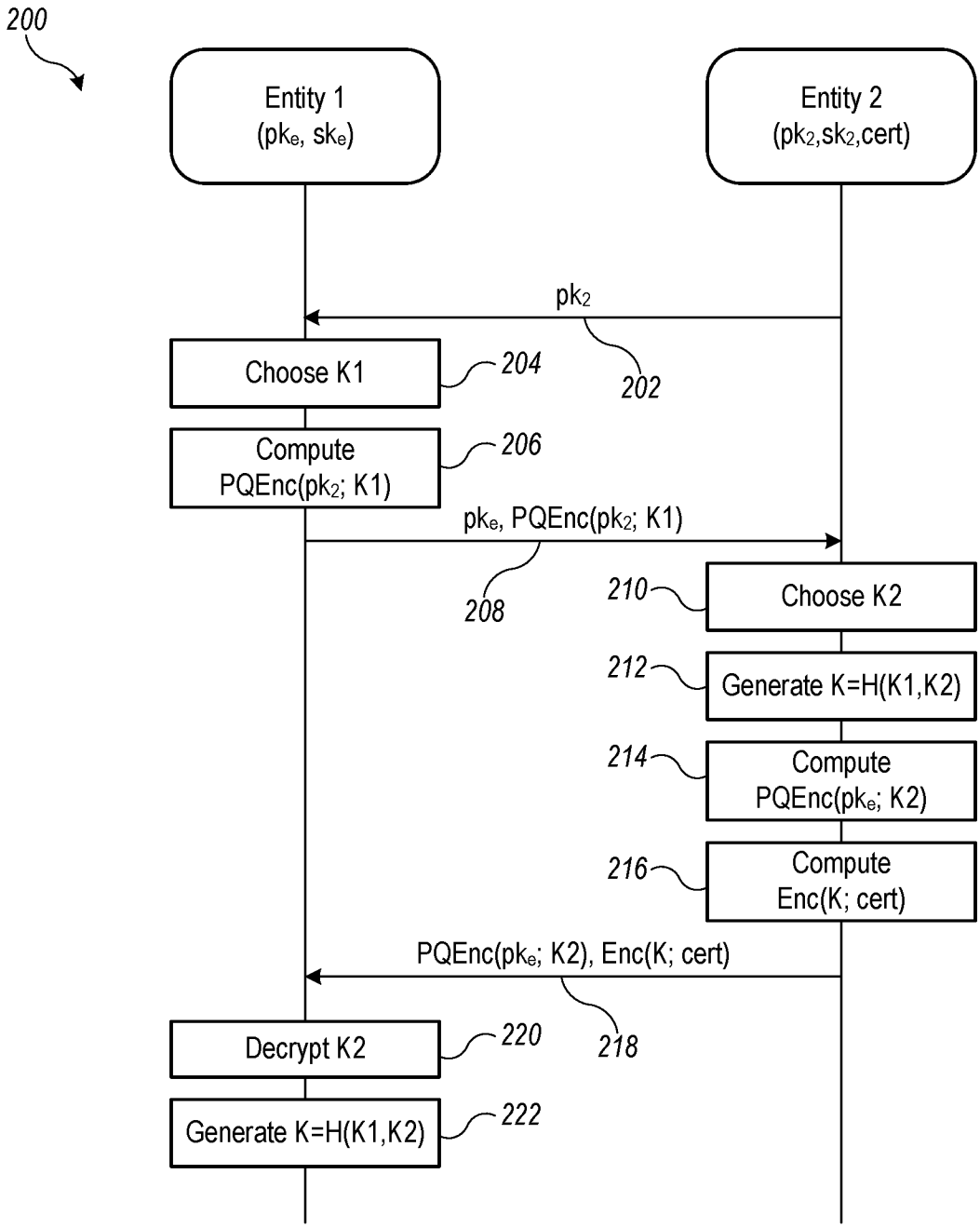

200
Entity 1
(pke, ske)
Entity 2
(pk2,sk2,cert)
pk2
202
Choose K1
204
Compute
PQEnc(pk2; K1)
206
pke, PQEnc(pk2; K1)
208
210
Choose K2
212
Generate K=H(K1,K2)
214
Compute
PQEnc(pke; K2)
216
Compute
Enc(K; cert)
PQEnc(pke; K2), Enc(K; cert)
218
Decrypt K2
220
Generate K=H(K1,K2)
222

FIG. 2

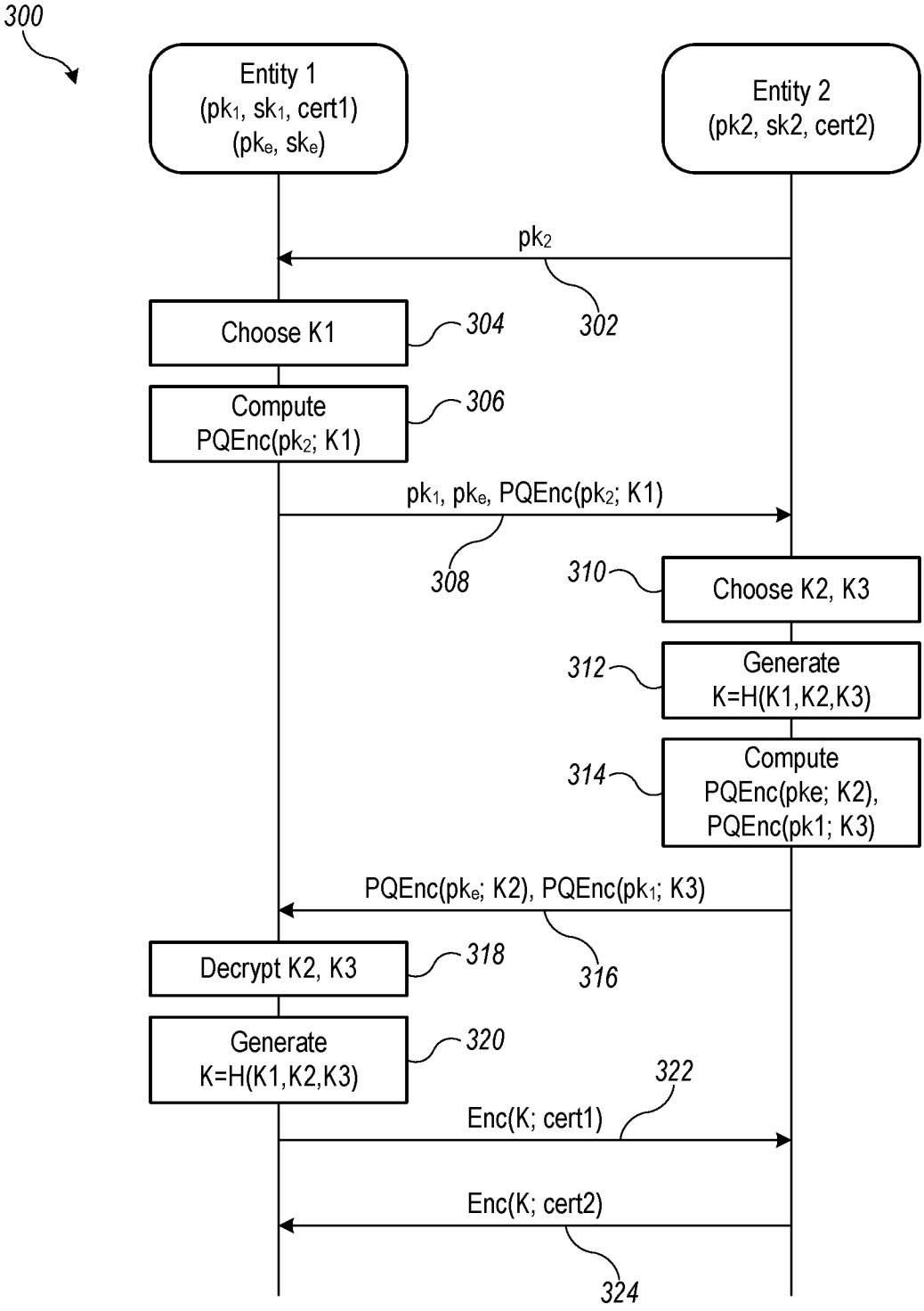

300
Entity 1
(pk1, sk1, cert1)
(pke, ske)
Entity 2
(pk2, sk2, cert2)
pk2
302
Choose K1
304
Compute
PQEnc(pk2; K1)
306
pk1, pke, PQEnc(pk2; K1)
308
310
Choose K2, K3
312
Generate
K=H(K1,K2,K3)
314
Compute
PQEnc(pke; K2),
PQEnc(pk1; K3)
PQEnc(pke; K2), PQEnc(pk1; K3)
316
Decrypt K2, K3
318
Generate
K=H(K1,K2,K3)
320
322
Enc(K; cert1)
Enc(K; cert2)
324

FIG. 3

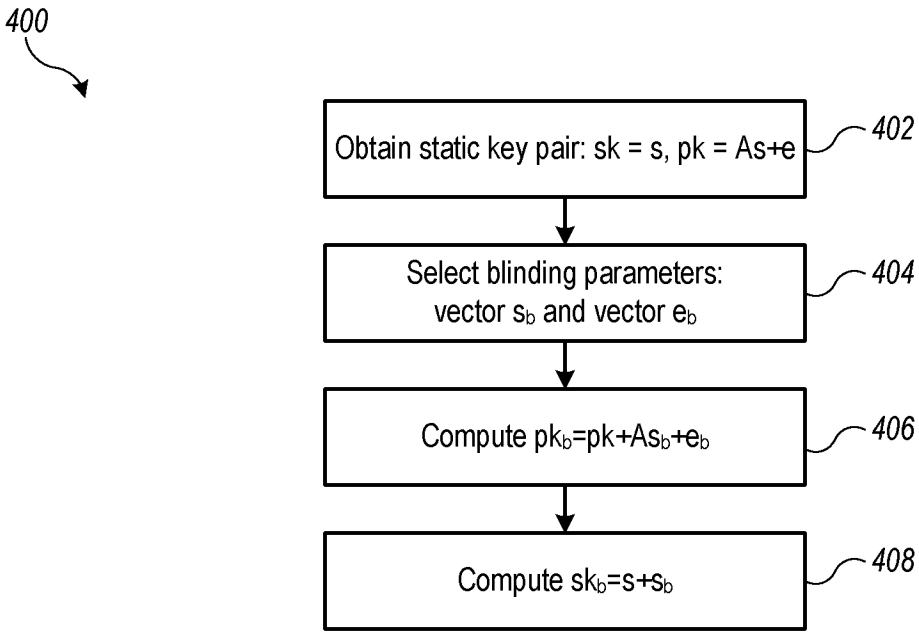

400
Obtain static key pair: sk = s, pk = As+e
402
Select blinding parameters: vector s_b and vector e_b
404
Compute $pk_b=pk+As_b+e_b$
406
Compute $sk_b=s+s_b$
408

FIG. 4

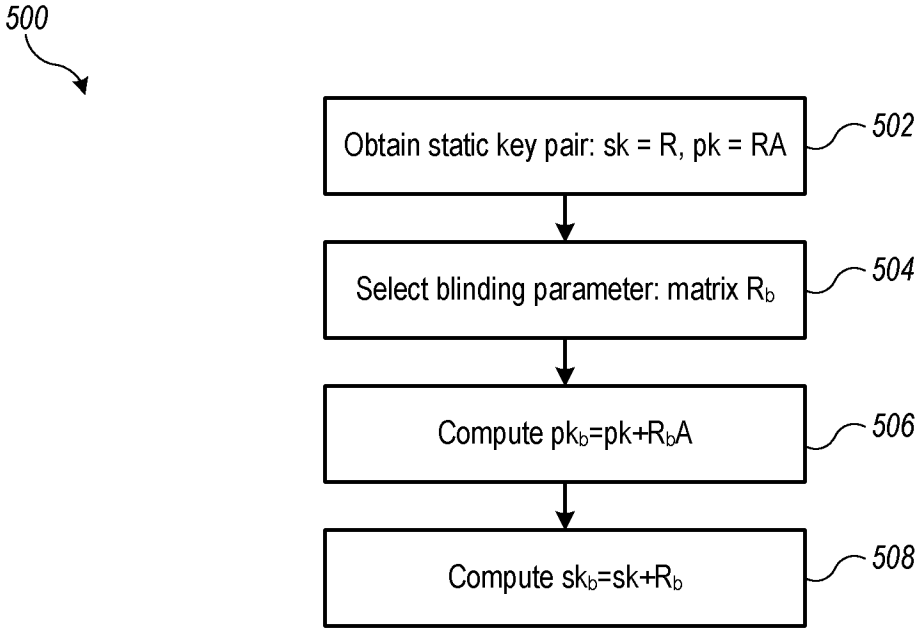

500
Obtain static key pair: sk = R, pk = RA
502
Select blinding parameter: matrix R_b
504
Compute $pk_b=pk+R_bA$
506
Compute $sk_b=sk+R_b$
508

FIG. 5

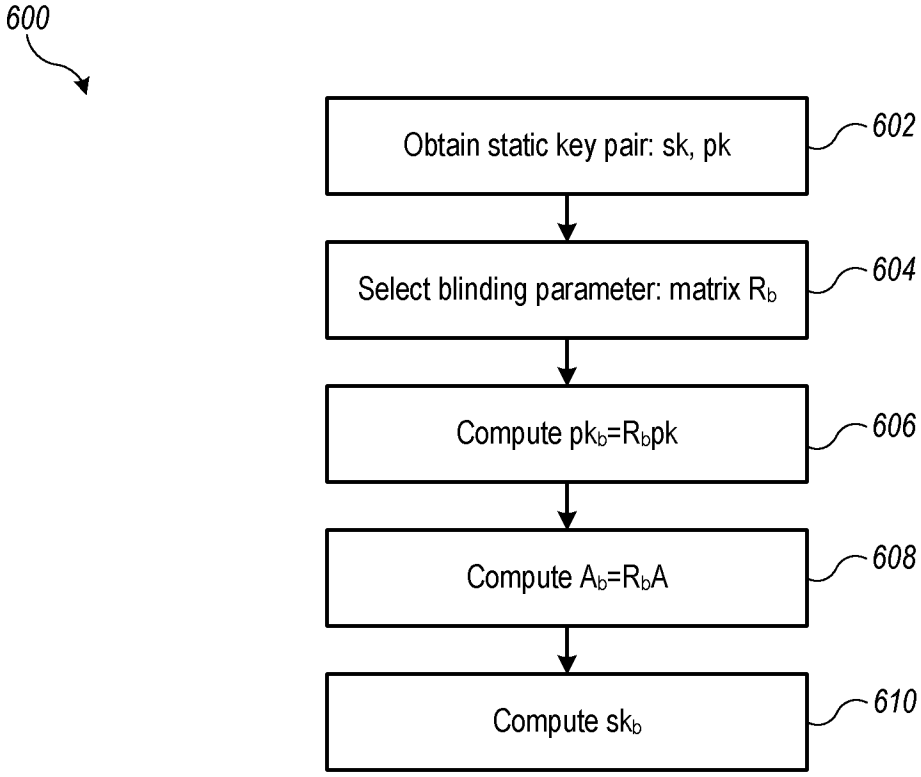

600
Obtain static key pair: sk, pk
602
Select blinding parameter: matrix R_b
604
Compute $pk_b=R_bpk$
606
Compute $A_b=R_bA$
608
Compute sk_b
610

FIG. 6

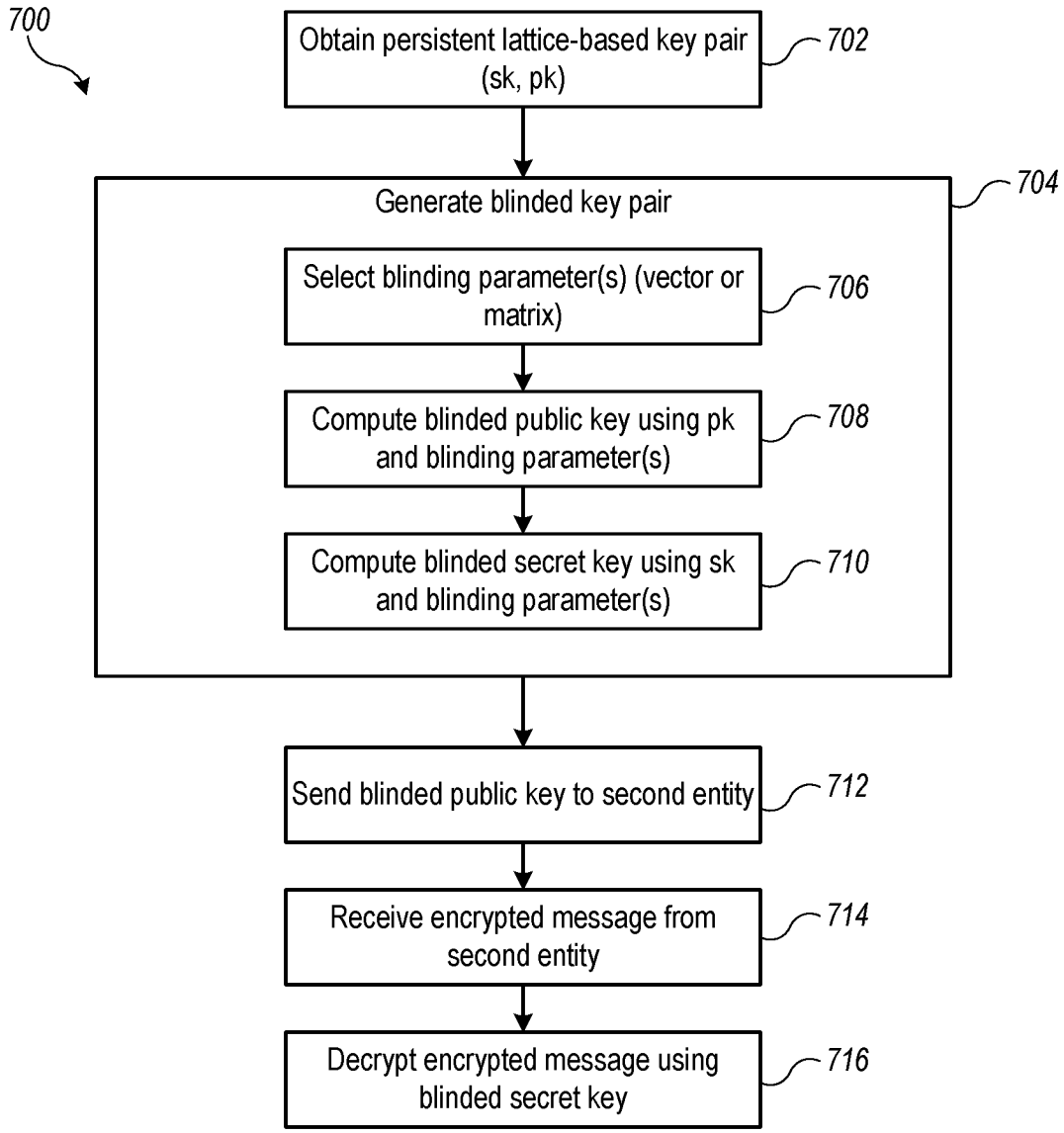

700
Obtain persistent lattice-based key pair (sk, pk)
702
Generate blinded key pair
704
Select blinding parameter(s) (vector or matrix)
706
Compute blinded public key using pk and blinding parameter(s)
708
Compute blinded secret key using sk and blinding parameter(s)
710
Send blinded public key to second entity
712
Receive encrypted message from second entity
714
Decrypt encrypted message using blinded secret key
716

FIG. 7

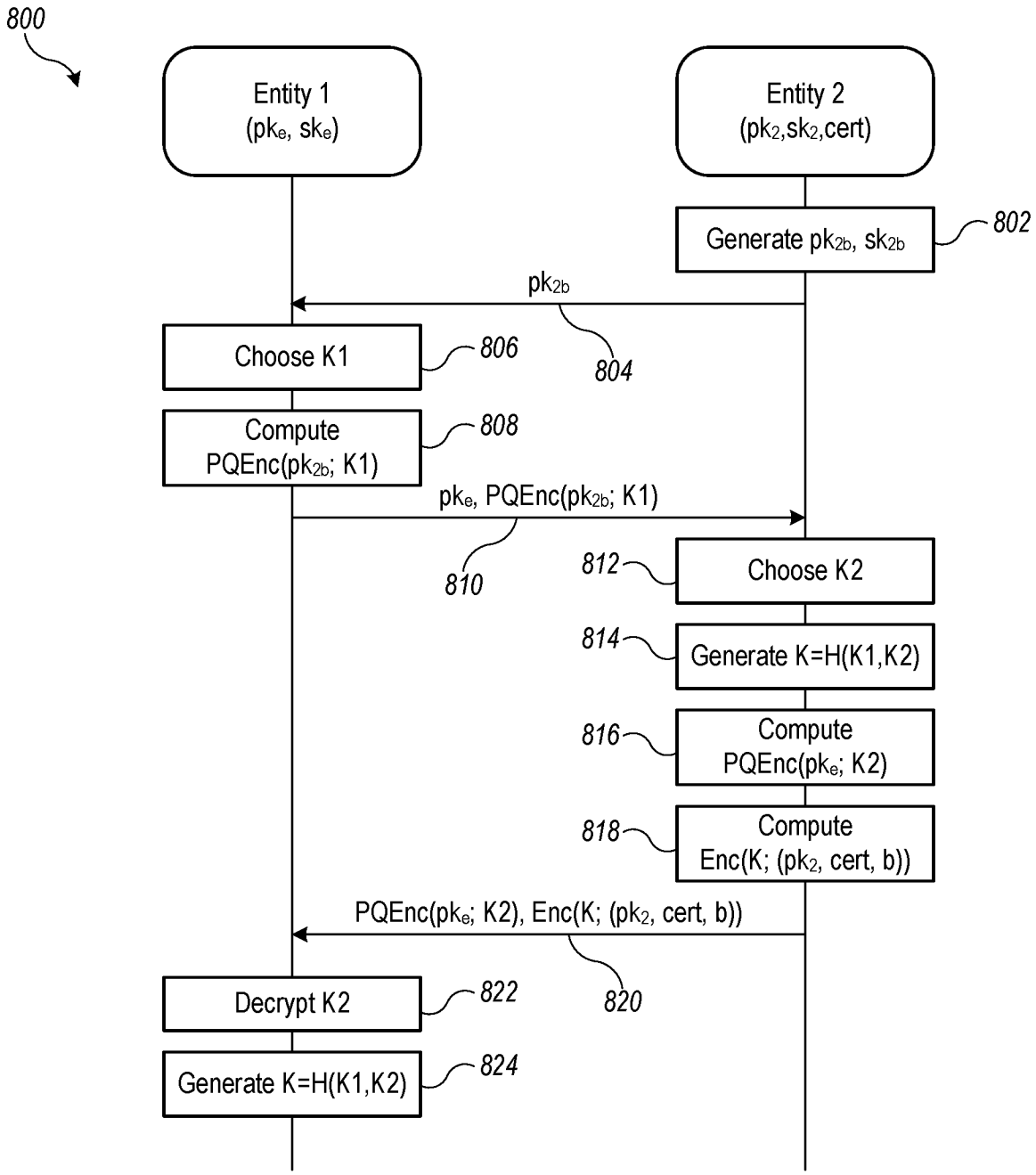

800
Entity 1
(pke, ske)
Entity 2
(pk2,sk2,cert)
Generate pk2b, sk2b
802
pk2b
804
Choose K1
806
Compute
PQEnc(pk2b; K1)
808
pke, PQEnc(pk2b; K1)
810
812
Choose K2
814
Generate K=H(K1,K2)
816
Compute
PQEnc(pke; K2)
818
Compute
Enc(K; (pk2, cert, b))
PQEnc(pke; K2), Enc(K; (pk2, cert, b))
820
Decrypt K2
822
Generate K=H(K1,K2)
824

FIG. 8

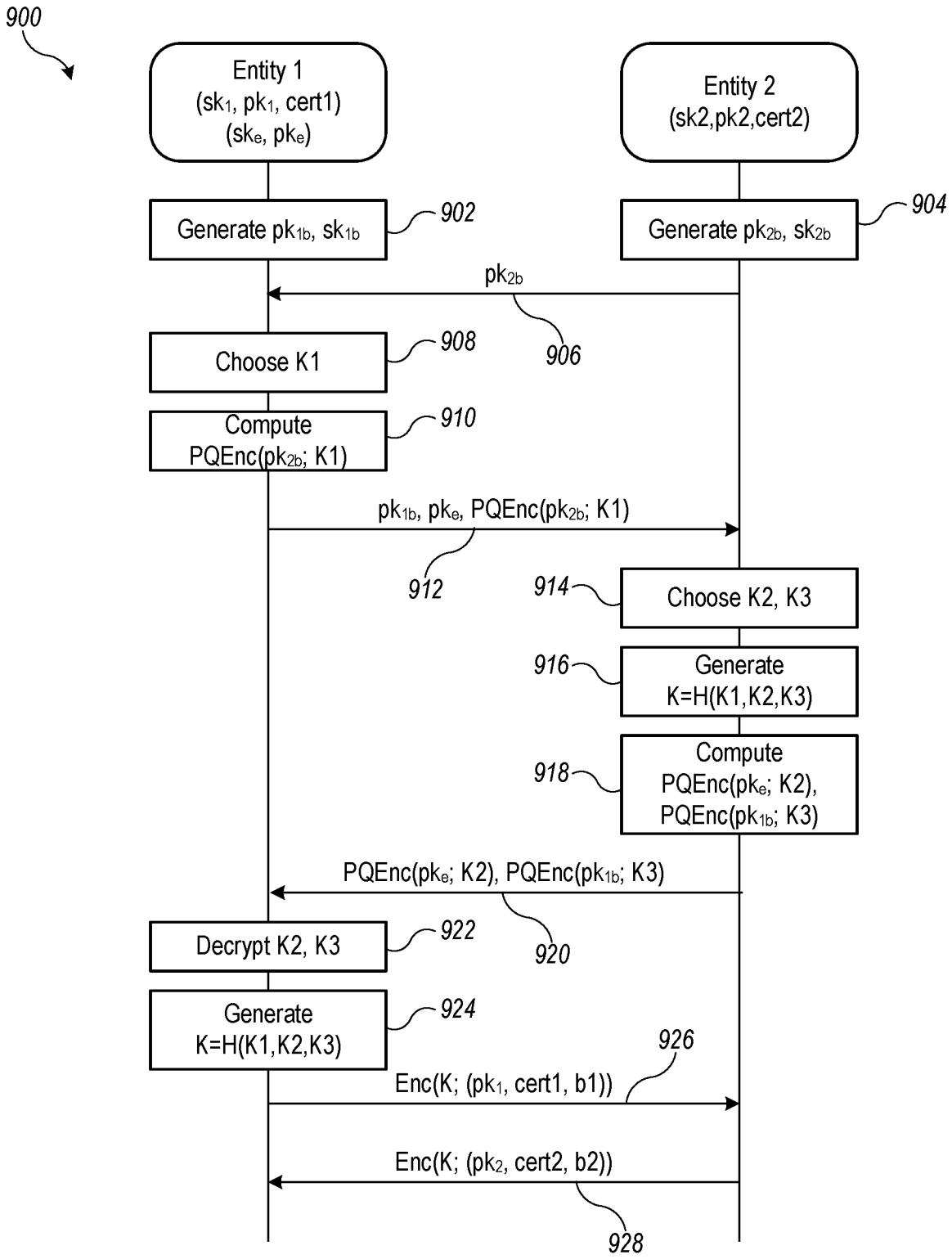

900
Entity 1
(sk1, pk1, cert1)
(ske, pke)
Entity 2
(sk2,pk2,cert2)
Generate pk1b, sk1b
902
Generate pk2b, sk2b
904
pk2b
906
Choose K1
908
Compute
PQEnc(pk2b; K1)
910
pk1b, pke, PQEnc(pk2b; K1)
912
914
Choose K2, K3
916
Generate
K=H(K1,K2,K3)
918
Compute
PQEnc(pke; K2),
PQEnc(pk1b; K3)
PQEnc(pke; K2), PQEnc(pk1b; K3)
920
Decrypt K2, K3
922
Generate
K=H(K1,K2,K3)
924
926
Enc(K; (pk1, cert1, b1))
Enc(K; (pk2, cert2, b2))
928

FIG. 9

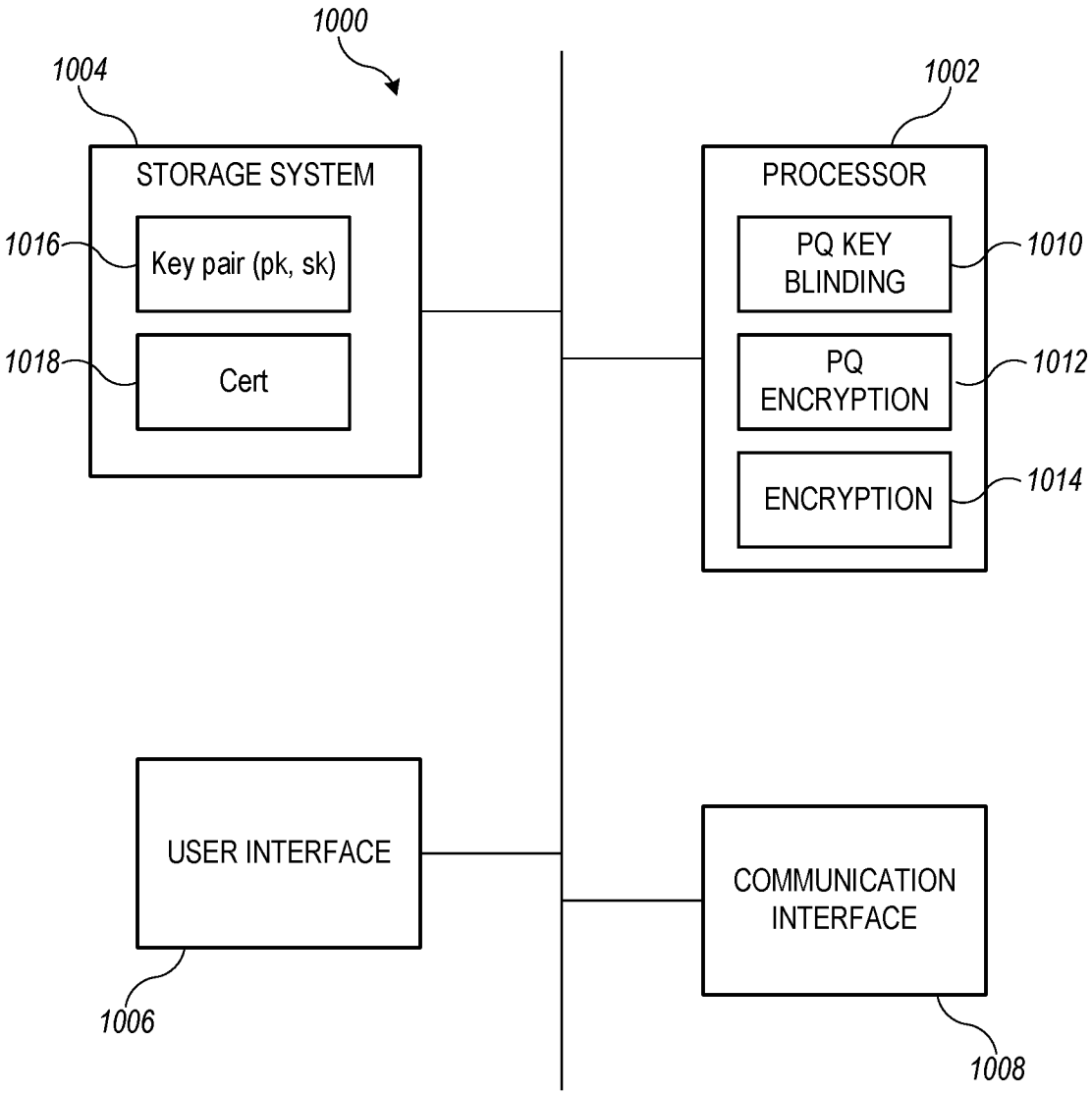

1000
1004
STORAGE SYSTEM
1016
Key pair (pk, sk)
1018
Cert
1002
PROCESSOR
PQ KEY BLINDING
1010
PQ ENCRYPTION
1012
ENCRYPTION
1014
USER INTERFACE
1006
COMMUNICATION INTERFACE
1008

FIG. 10