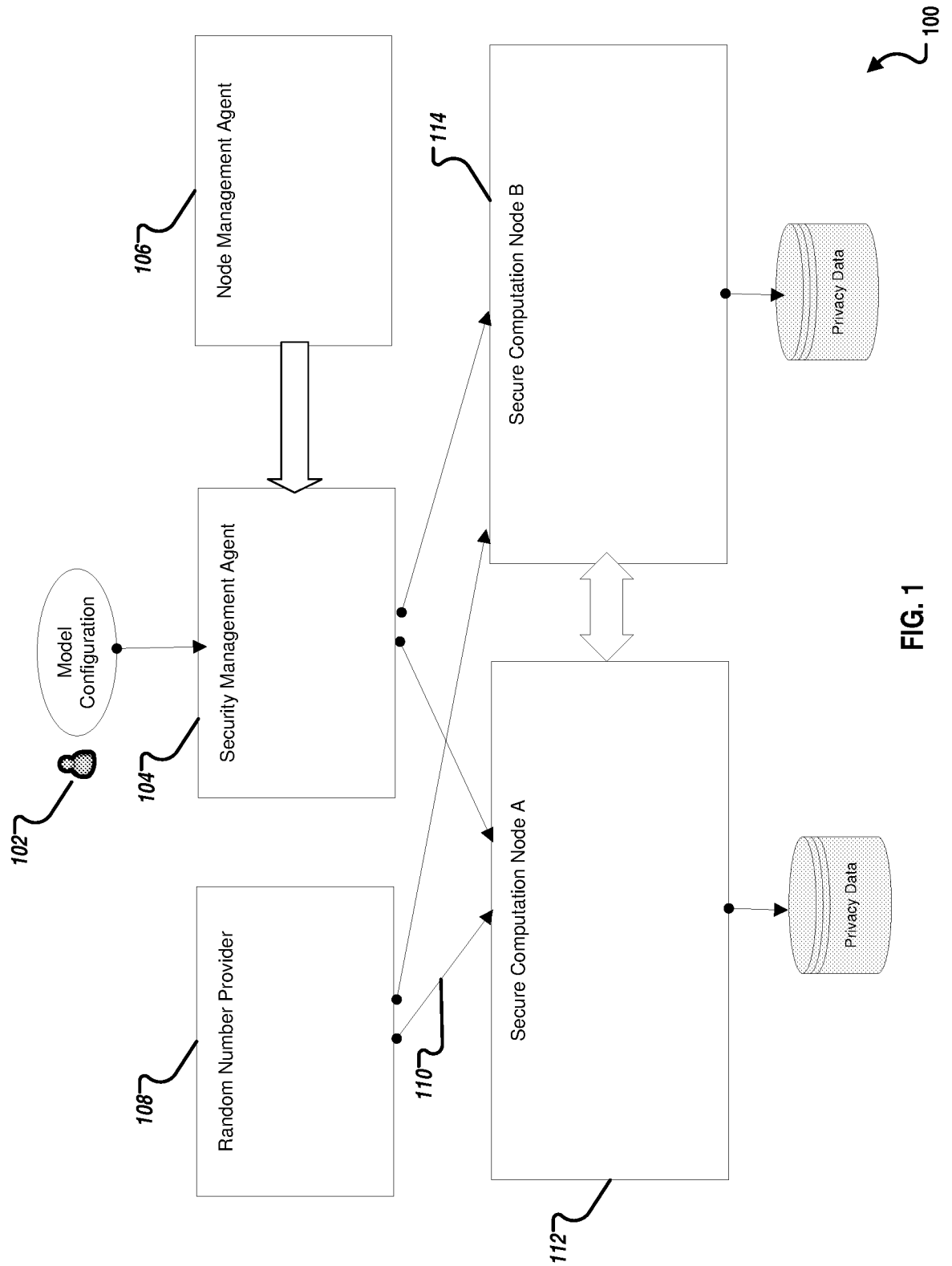
100
Node Management Agent
106
Secure Computation Node B
114
Privacy Data
Model Configuration
Security Management Agent
104
102
Random Number Provider
108
110
Secure Computation Node A
Privacy Data
112

FIG. 1

P0
P1
predetermine share of uv
$z_0 + z_1 = uv = u(v_0 + v_1)$
predetermined
$u, v_0, z_0$
predetermined
$v_1, z_1$
assume $a = X^L$, $b_0 = \langle \theta^L \rangle_0$
assume $b_1 = \langle \theta^L \rangle_1$
$e = a - u, b_0 - v_0$
$b_1 - v_1$
$f = b - v$
$f = b - v$
$c_0 = uf + eb_0 + z_0$
$c_0 + c_1 = uf + eb + uv$
$c_0 + c_1 = ub - uv + ab - ub + uv$
$c_0 + c_1 = ab$
$c_1 = eb_1 + z_1$
200a

FIG. 2A

P1
P0
predetermine share of uv
$z_0 + z_1 = uv = u(v_0 + v_1)$
predetermined
$u, v_0, z_0$
predetermined
$v_1, z_1$
assume $a = X^R$, $b_0 = \langle \theta^R \rangle_1$
assume $b_1 = \langle \theta^R \rangle_0$
$e = a - u, b_0 - v_0$
$b_1 - v_1$
$f = b - v$
$f = b - v$
$c_0 = uf + eb_0 + z_0$
$c_0 + c_1 = uf + eb + uv$
$c_0 + c_1 = ub - uv + ab - ub + uv$
$c_0 + c_1 = ab$
$c_1 = eb_1 + z_1$
200b

FIG. 2B

P0
P1
predetermine share of uv
$z_0 + z_1 = u \bullet v$
predetermined
$u, z_0$
predetermined
$v, z_1$
$a$
$b$
$e = a - u$
$f = b - v$
$c_0 = u \bullet f + z_0$
$c_0 + c_1 = u \bullet f + e \bullet b + u \bullet v$
$c_0 + c_1 = u \bullet b - u \bullet v + a \bullet b - u \bullet b + u \bullet v$
$c_0 + c_1 = a \bullet b$
$c_1 = e \bullet b + z_1$
300

FIG. 3

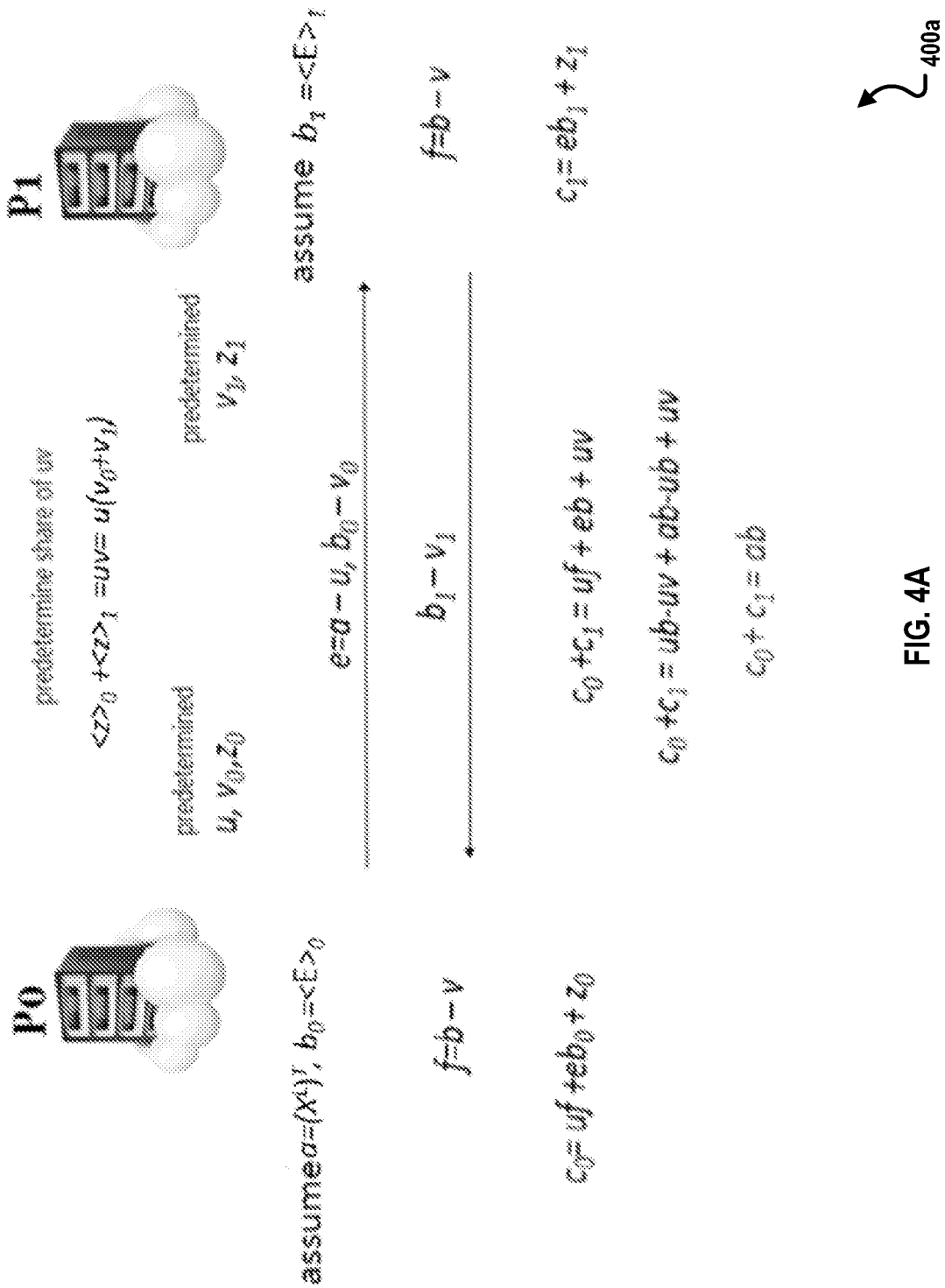
P0
P1
predetermine share of uv
$<z>_0 + <z>_1 = uv = u(v_0+v_1)$
predetermined
$u, v_0, z_0$
predetermined
$v_1, z_1$
assume $a=(X^i)^T$, $b_0=<E>_0$
$e=a-u, b_0-v_0$
assume $b_1=<E>_1$
$f=b-v$
$b_1-v_1$
$f=b-v$
$c_0=uf+eb_0+z_0$
$c_0+c_1=uf+eb+uv$
$c_0+c_1=ub-uv+ab-ub+uv$
$c_0+c_1=ab$
$c_1=eb_1+z_1$
400a

FIG. 4A

P1
P0
predetermine share of uv
<z>0 +<z>1 =uv= u(v0+v1)
predetermined
u, v0, z0
predetermined
v1, z1
assume a = (x^n)^T, b0 = < ε >1
assume b1 = < ε >0
e=a−u, b0−v0
b1−v1
f=b−v
f=b−v
c0= uf +eb0+ z0
c0+c1 = uf + eb + uv
c0+c1 = ub-uv + ab-ub + uv
c0 + c1 = ab
c1= eb1 + z1
400b

FIG. 4B

P0
P1
P0 generates random matrix u0 and v0
P1 generates random matrix u1 and v1
Both P0 and P1 predetermine Z0 and Z1
$(u_0+u_1)(v_0+v_1)=(z_0+z_1)$
$a_0$ $b_0$
$u_0, v_0, z_0$
$u_1, v_1, z_1$
$a_1$ $b_1$
$a_0-u_0, b_0-v_0$
$a_1-u_1, b_1-v_1$
$e = a - u$
$f = b - v$
$e = a - u$
$f = b - v$
computation result
$c_0 = -ef + a_0 f + eb_0 + z_0$
computation result
$c_0+c_1 = -ef + af + eb + uv$
$c_0+c_1 = uf + ab - ub + uv$
$c_0+c_1 = ub - uv + ab - ub + uv$
$c_0+c_1 = ab$
$c_1 = a_1 f + eb_1 + z_1$
500

FIG. 5

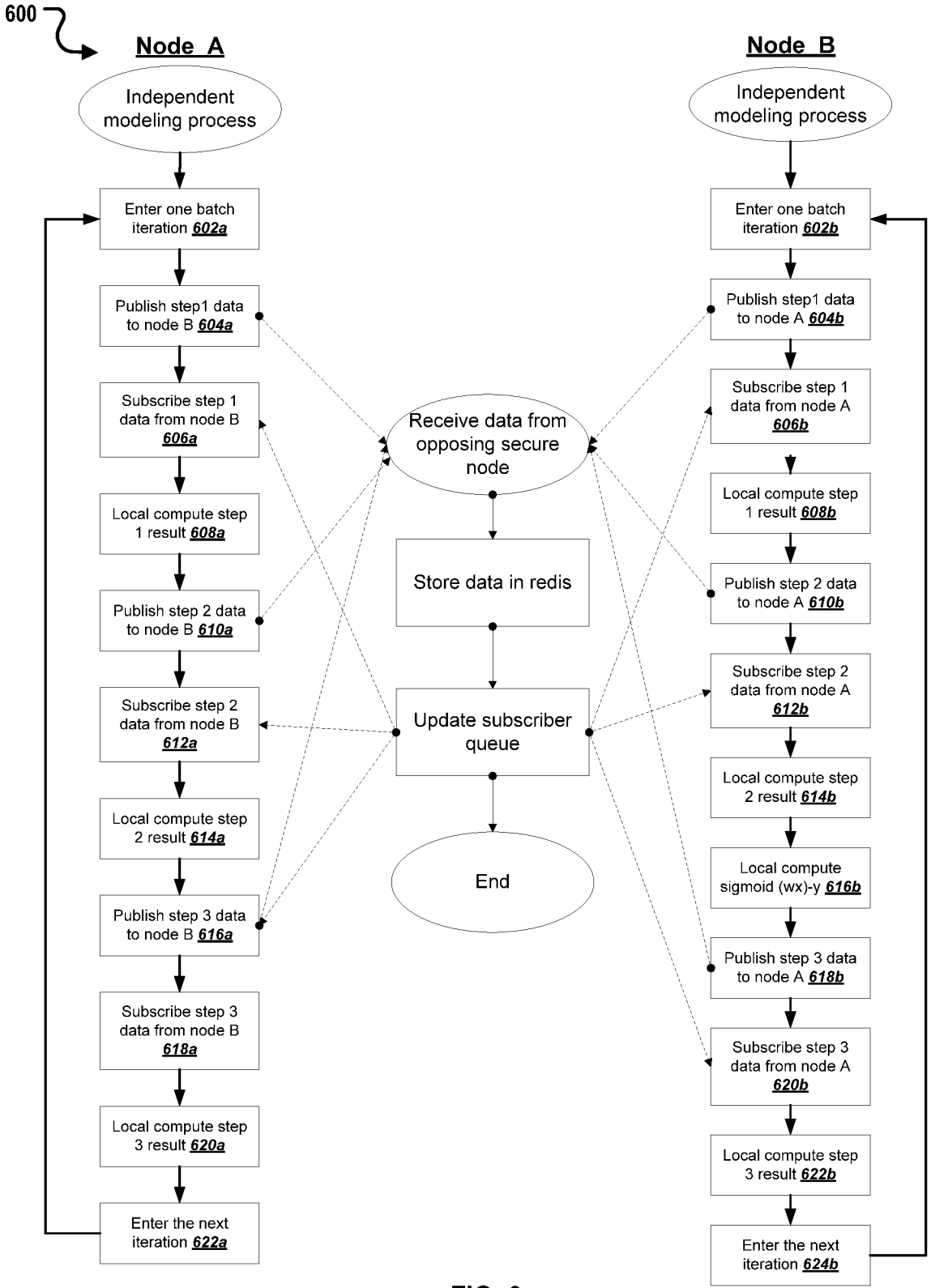
600
Node A
Independent modeling process
Enter one batch iteration 602a
Publish step1 data to node B 604a
Subscribe step 1 data from node B 606a
Local compute step 1 result 608a
Publish step 2 data to node B 610a
Subscribe step 2 data from node B 612a
Local compute step 2 result 614a
Publish step 3 data to node B 616a
Subscribe step 3 data from node B 618a
Local compute step 3 result 620a
Enter the next iteration 622a
Receive data from opposing secure node
Store data in redis
Update subscriber queue
End
Node B
Independent modeling process
Enter one batch iteration 602b
Publish step1 data to node A 604b
Subscribe step 1 data from node A 606b
Local compute step 1 result 608b
Publish step 2 data to node A 610b
Subscribe step 2 data from node A 612b
Local compute step 2 result 614b
Local compute sigmoid (wx)-y 616b
Publish step 3 data to node A 618b
Subscribe step 3 data from node A 620b
Local compute step 3 result 622b
Enter the next iteration 624b

FIG. 6

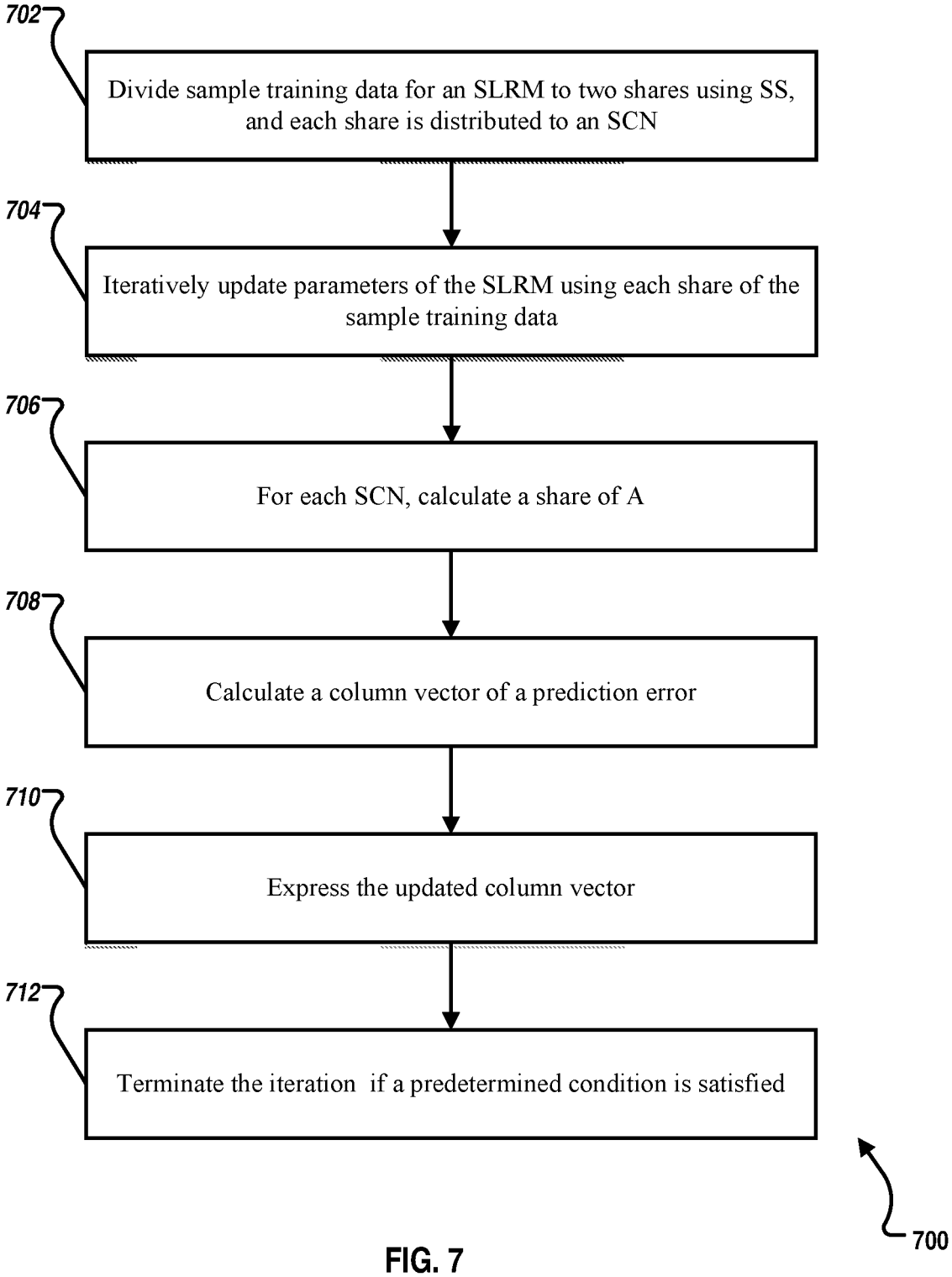
702
Divide sample training data for an SLRM to two shares using SS, and each share is distributed to an SCN
704
Iteratively update parameters of the SLRM using each share of the sample training data
706
For each SCN, calculate a share of A
708
Calculate a column vector of a prediction error
710
Express the updated column vector
712
Terminate the iteration if a predetermined condition is satisfied
700

FIG. 7

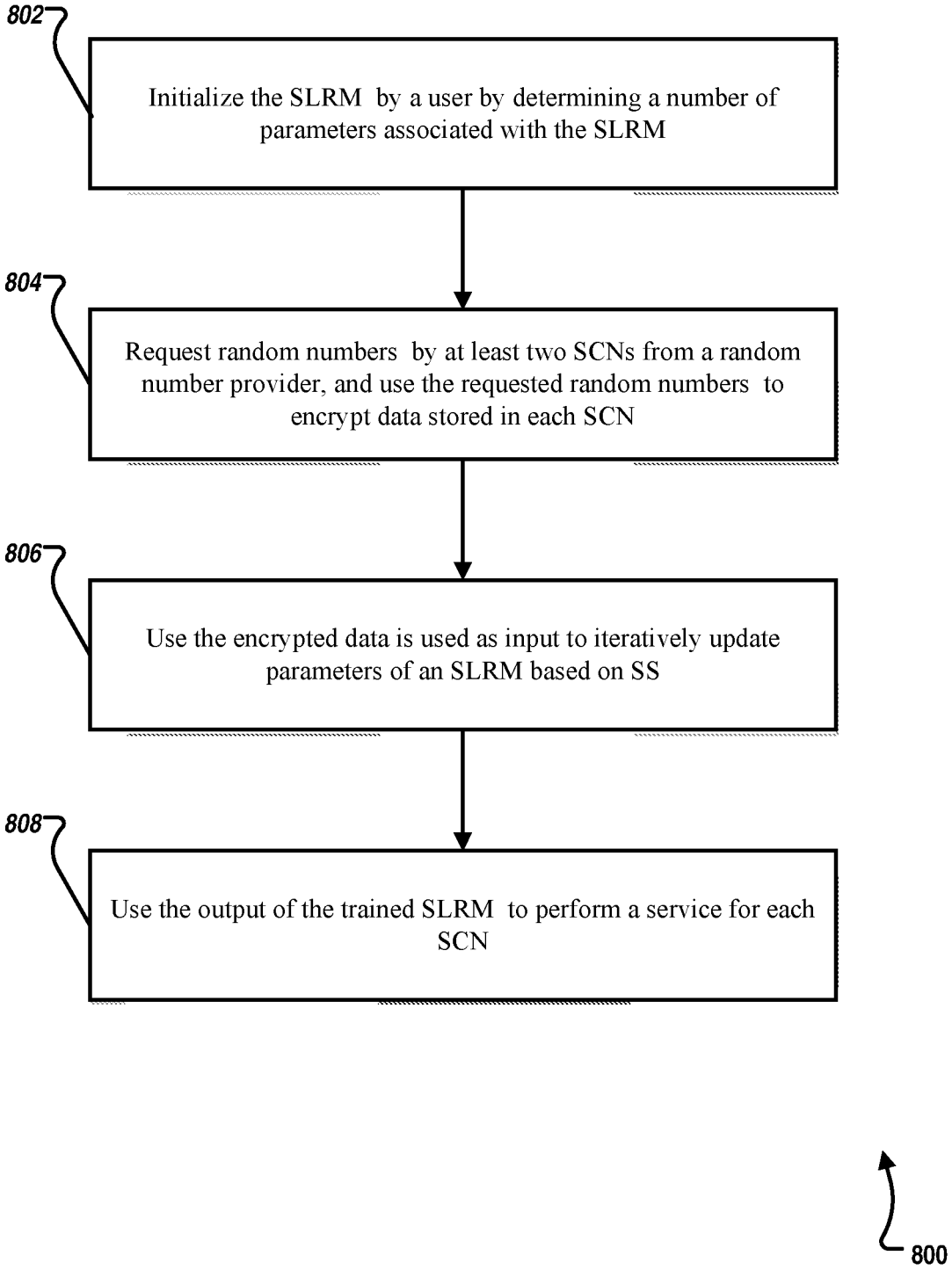
802
Initialize the SLRM by a user by determining a number of parameters associated with the SLRM
804
Request random numbers by at least two SCNs from a random number provider, and use the requested random numbers to encrypt data stored in each SCN
806
Use the encrypted data is used as input to iteratively update parameters of an SLRM based on SS
808
Use the output of the trained SLRM to perform a service for each SCN
800

FIG. 8

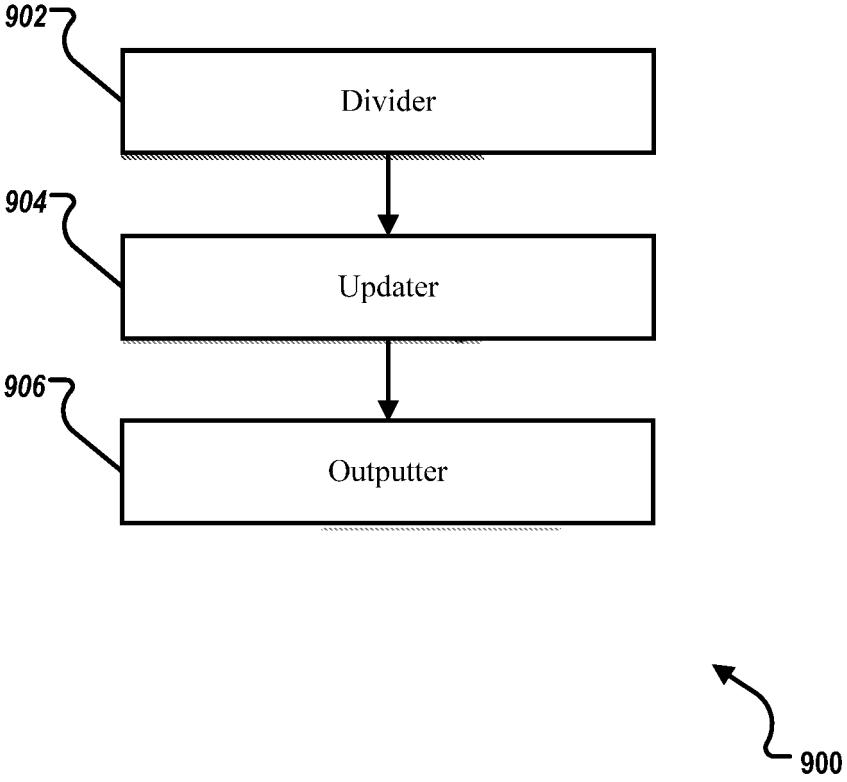


FIG. 9