

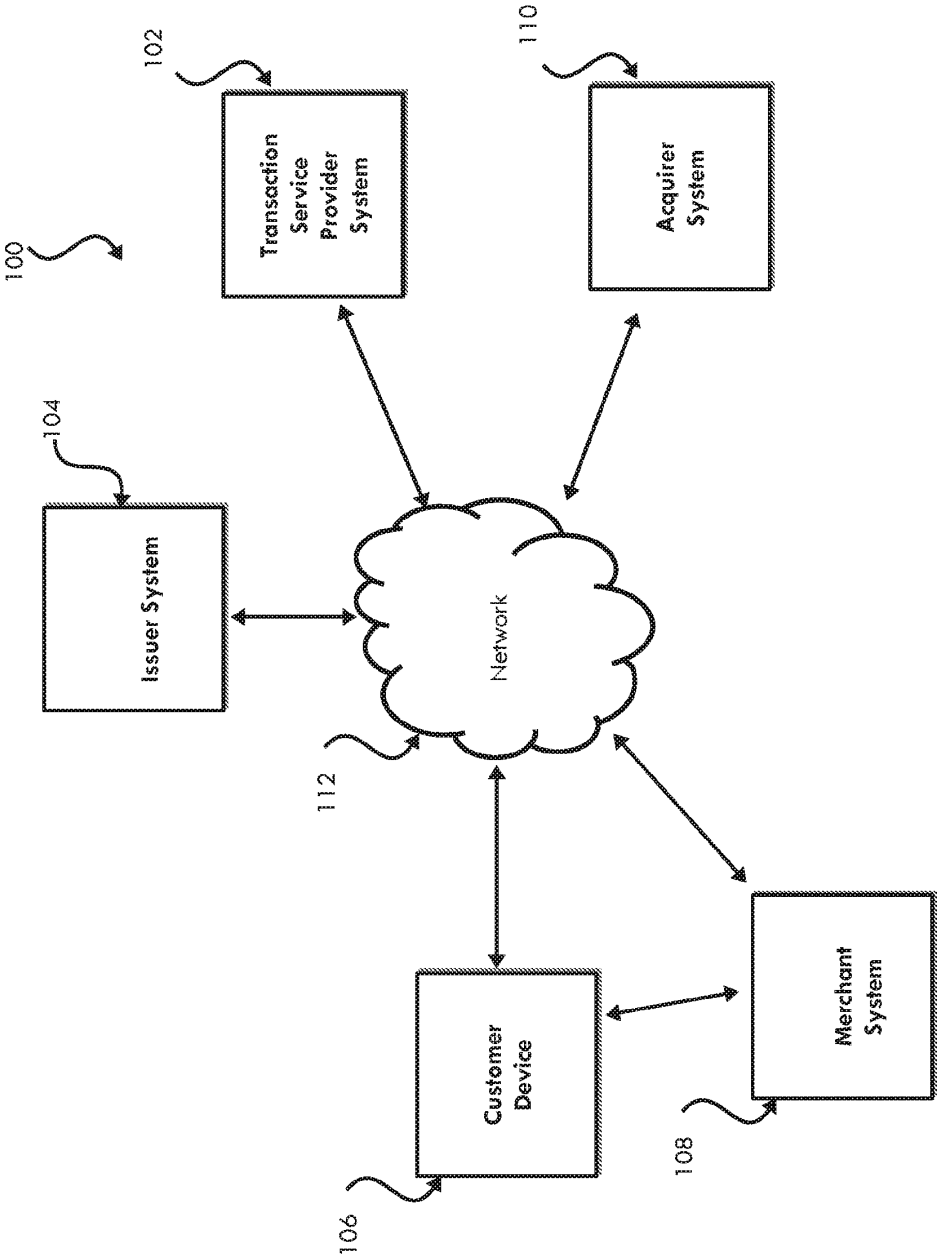


FIG. 1

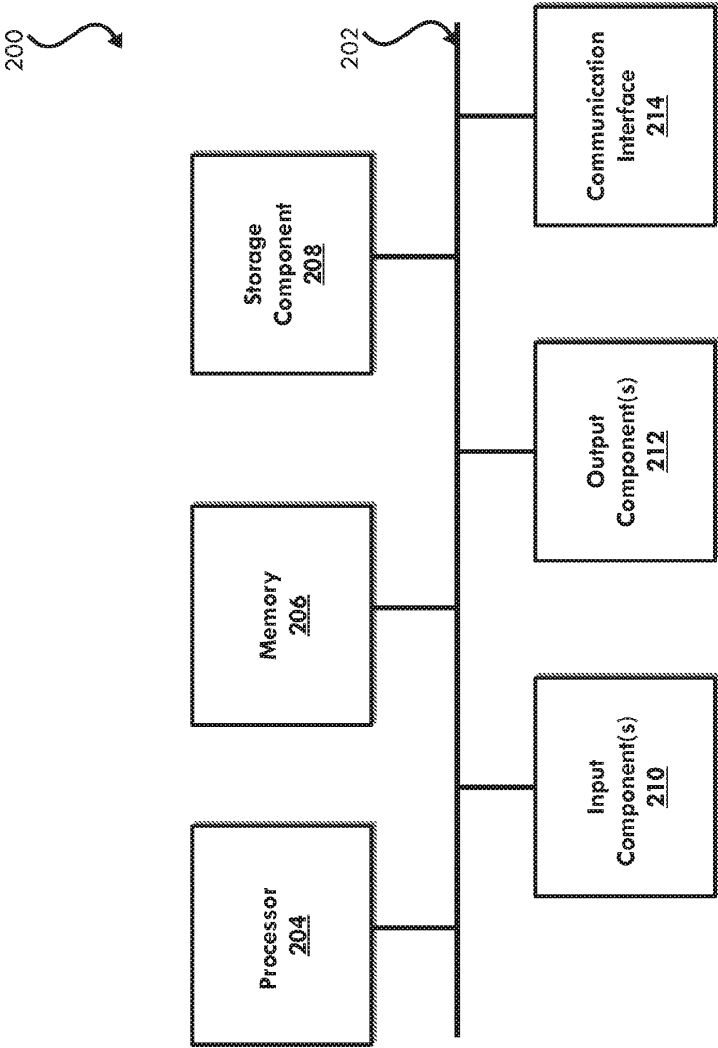


FIG. 2

3/21

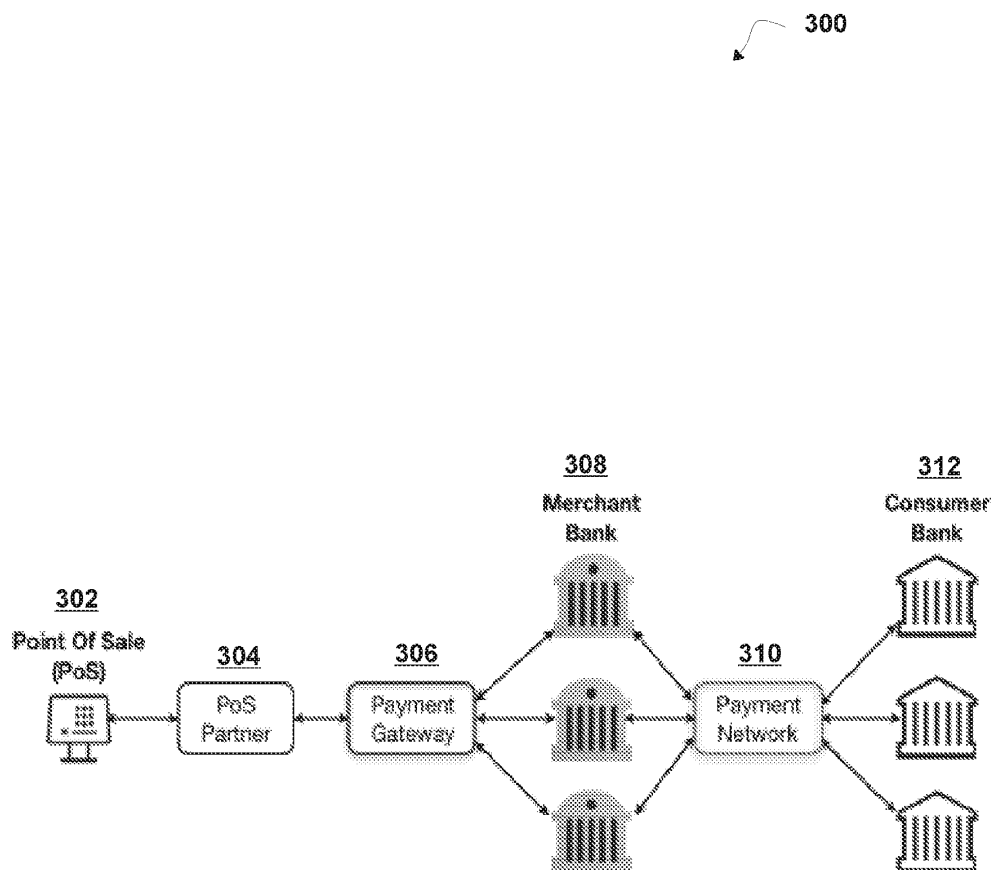


FIG. 3

400

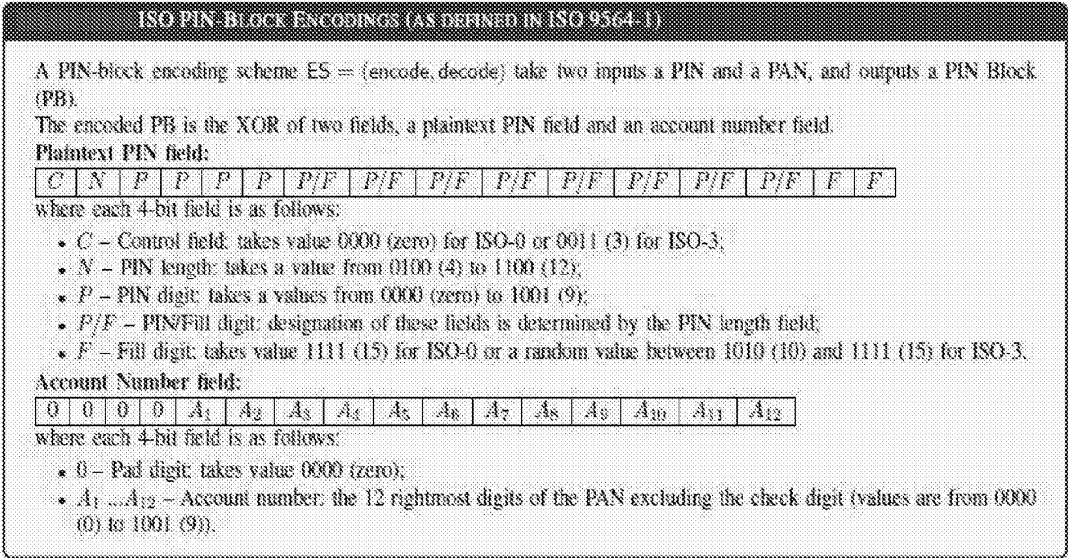


FIG. 4

5/21

500

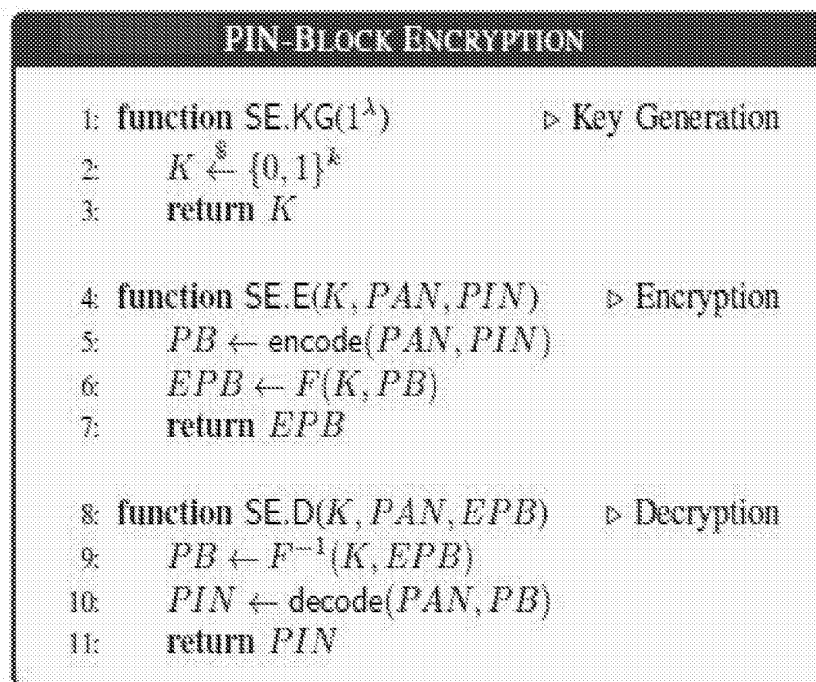


FIG. 5

6/21

600

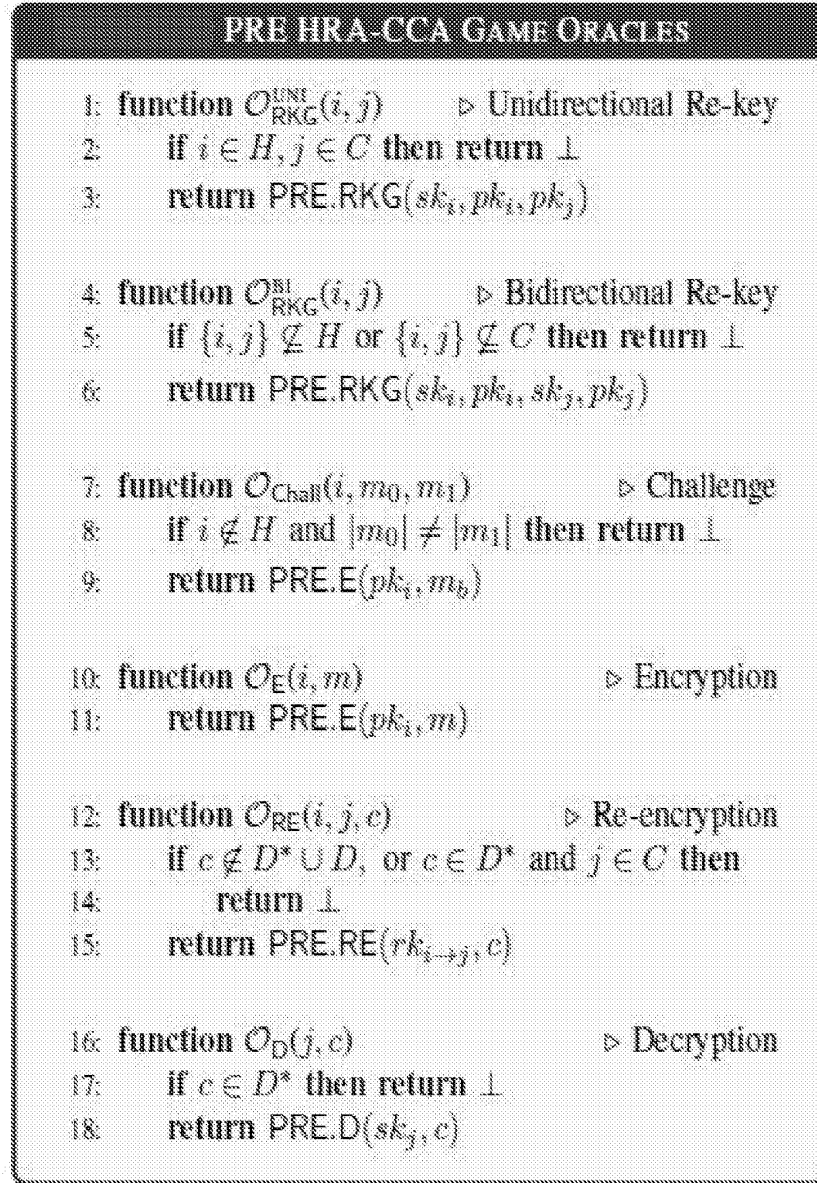


FIG. 6

7/21

700

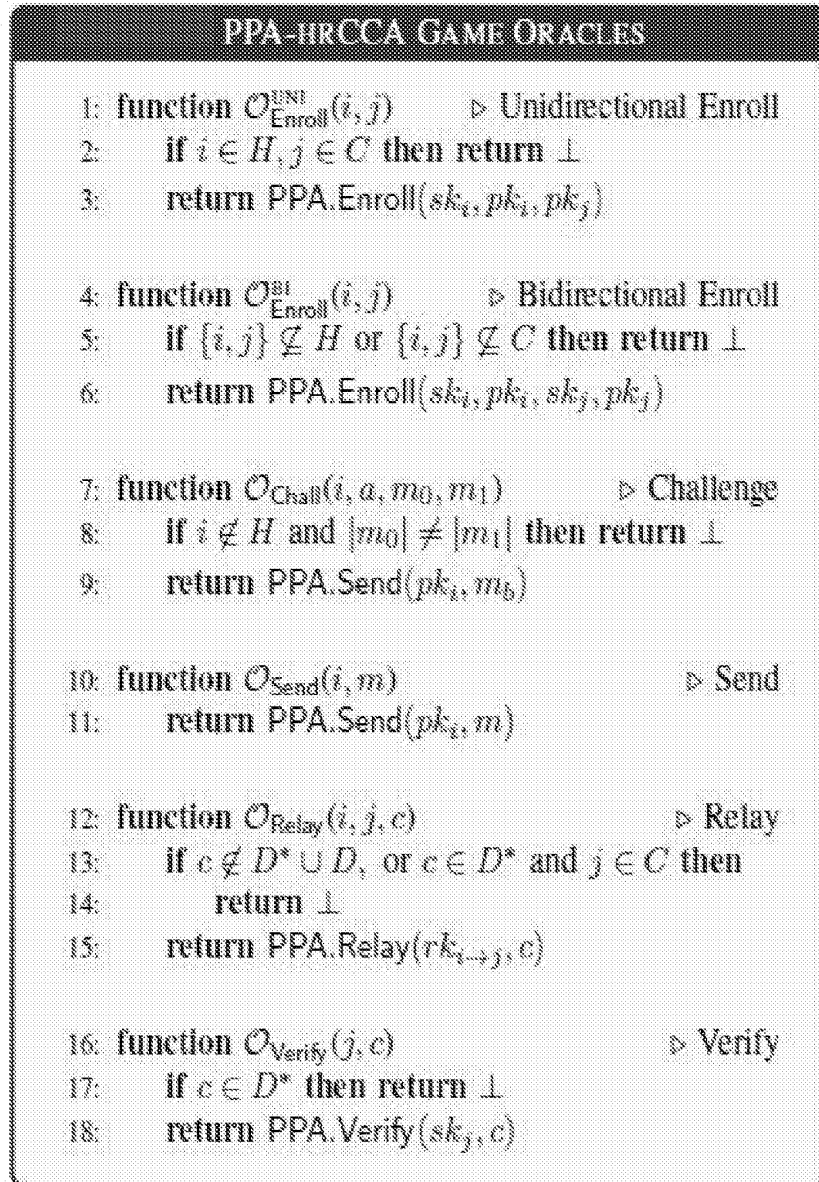


FIG.7

8/21

800

PPA CONSTRUCTION

```

1: function PPA.Setup( $1^\lambda$ )                                ▷ Setup
2:    $pp \leftarrow \text{PRE.Setup}(1^\lambda)$ 
3:    $ak \leftarrow \text{Init}(A)$ 
4:   return  $pp$ 

5: function PPA.KG( $pp$ )                                    ▷ Key Generation
6:    $(pk, sk) \leftarrow \text{PRE.KG}(pp)$ 
7:   return  $(pk, sk)$ 

8: function PPA.Enroll( $((pk_i, sk_i), (pk_j, sk_j^*))$ )
9:                                     ▷ Enroll User/Proxy
10:   $rk_{i \rightarrow j} \leftarrow \text{PRE.RKG}((pk_i, sk_i), (pk_j, sk_j^*))$ 
11:  return  $rk_{i \rightarrow j}$ 

12: function PPA.Send( $pk, a, m$ )                            ▷ Send Message
13:   $c \leftarrow \text{PRE.E}(pk, a, m)$ 
14:  return  $c$ 

15: function PPA.Verify[ $A; ak$ ]( $sk, a, c$ )
16:                                     ▷ Receive and Verify Message
17:   $m \leftarrow \text{PRE.D}(pk, a, c)$ 
18:   $b \leftarrow A(ak, a, m)$ 
19:  return  $b$ 

20: function PPA.Relay( $a, c, \mathcal{R}_i$ )                            ▷ Relay Message
21:   $rk_{i \rightarrow j} \leftarrow L(a, \mathcal{R}_i)$ 
22:   $c' \leftarrow \text{PRE.RE}(rk_{i \rightarrow j}, a, c)$ 
23:  return  $c'$ 

```

FIG. 8

9/21

900

HYBRID-PRE CONSTRUCTION

```

1: function PRE.Setup( $1^\lambda$ )                                ▷ Setup
2:   Choose  $G$  of prime order  $q$  with generator  $g$ 
3:   return  $pp = (g, q)$ 

4: function PRE.KG( $pp$ )                                    ▷ Key Generation
5:    $sk \xleftarrow{\$} \mathbb{Z}_q^*$ ;  $pk \leftarrow g^{sk}$ 
6:   return  $(pk, sk)$ 

7: function PRE.RKG( $(pk_i, sk_i), (pk_j, sk_j)$ )
8:                                     ▷ ReKey Generation
9:    $rk_{i \rightarrow j} \leftarrow sk_i / sk_j$ 
10:  return  $rk_{i \rightarrow j}$ 

11: function PRE.E( $pk, a, m$ )                                ▷ Encryption
12:    $r \xleftarrow{\$} \mathbb{Z}_q^*$ ;  $c_1 \leftarrow g^r$ 
13:    $K = F(pk^r)$ ;  $c_2 \leftarrow \text{SE.E}(K, a, m)$ 
14:   return  $(c_1, c_2)$ 

15: function PRE.D( $sk, a, c$ )                                ▷ Decryption
16:   Parse  $c$  as  $(c_1, c_2)$ 
17:    $K = F(c_1^{sk})$ ;  $m \leftarrow \text{SE.D}(K, a, c)$ 
18:   return  $m$ 

19: function PRE.RE( $rk_{i \rightarrow j}, a, c$ )                    ▷ ReEncryption
20:   Parse  $c$  as  $(c_1, c_2)$ 
21:    $c'_1 \leftarrow c_1^{rk_{i \rightarrow j}}$ 
22:   return  $(c'_1, c_2)$ 

```

FIG. 9

10/21

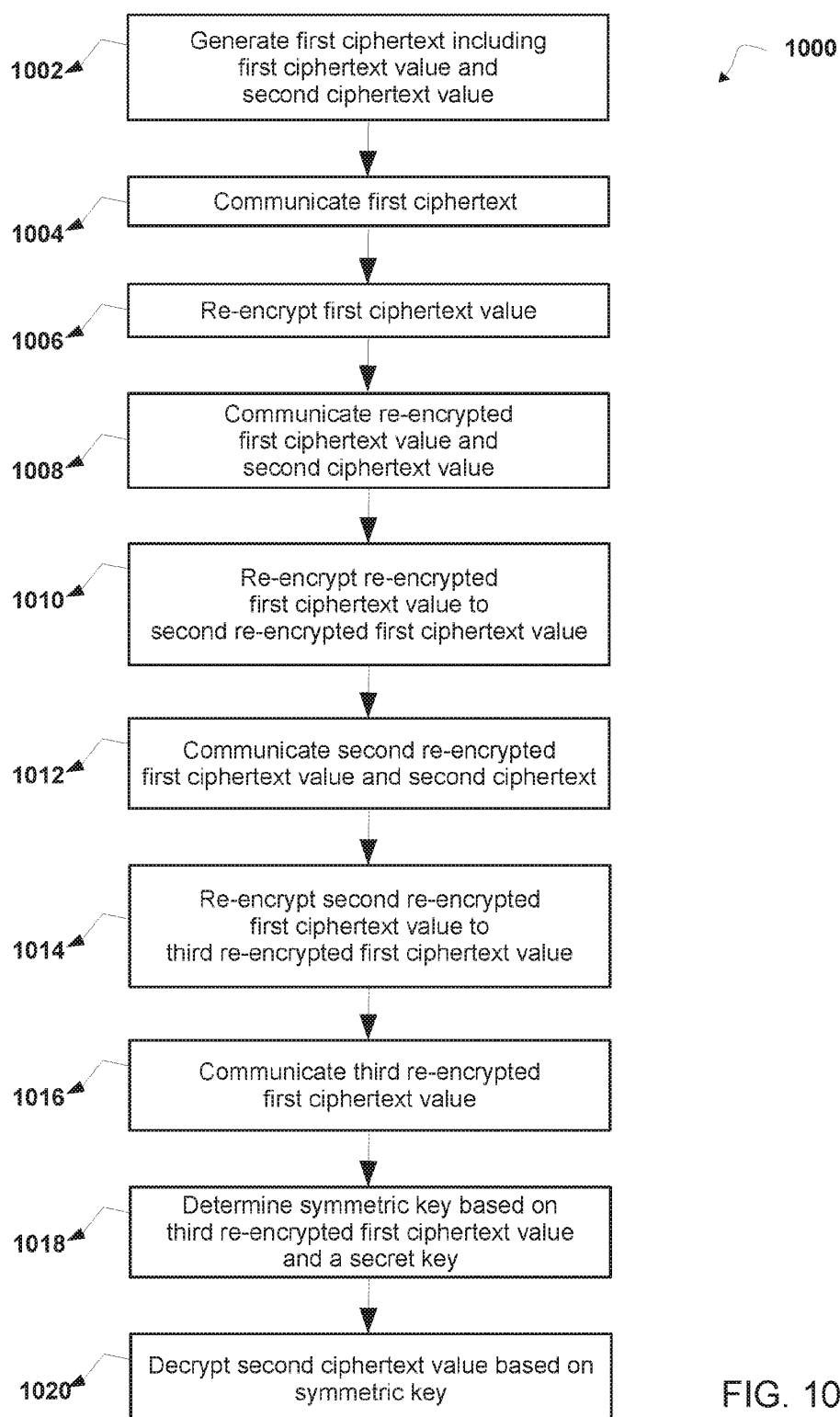


FIG. 10

11/21

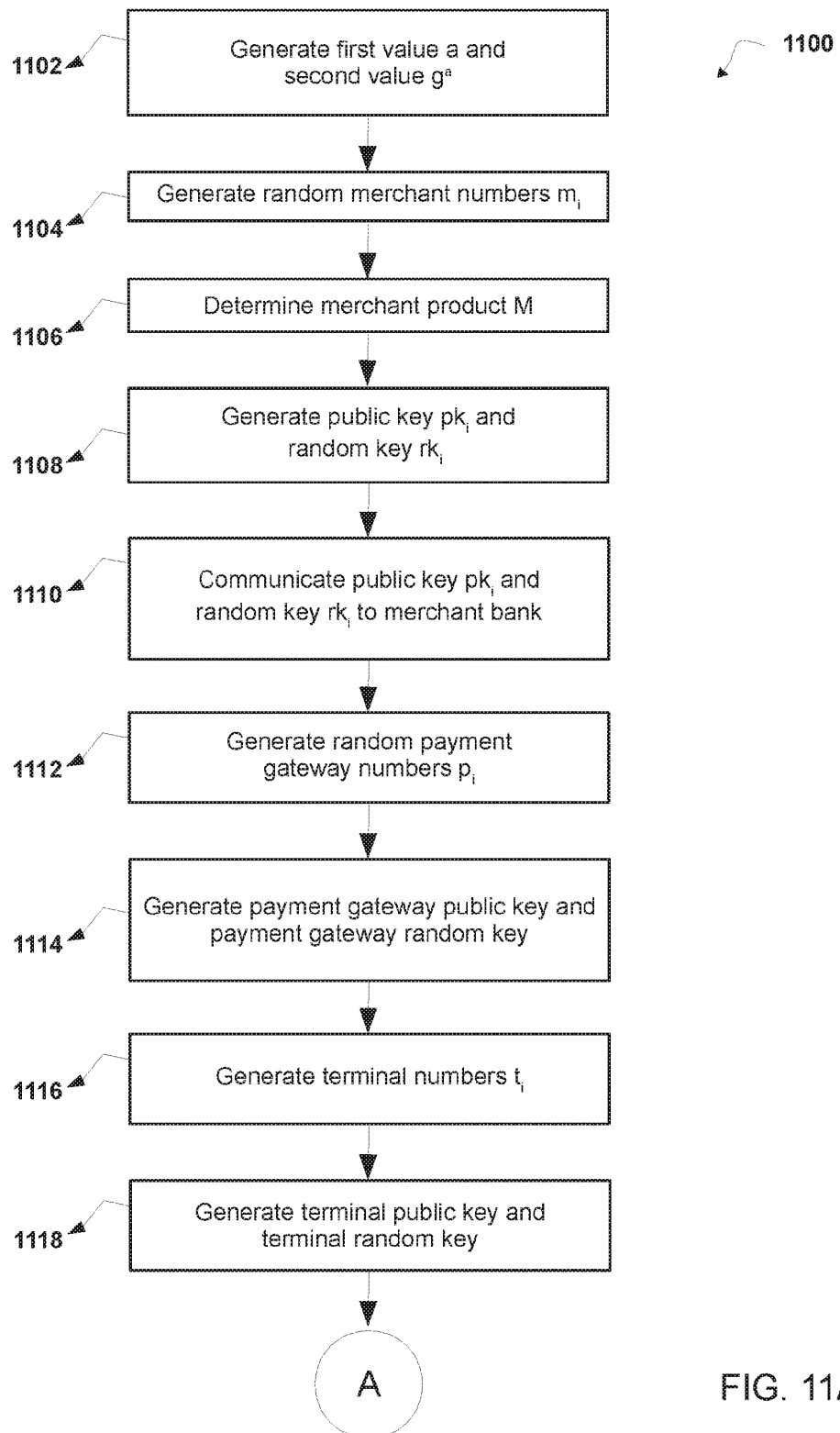


FIG. 11A

12/21

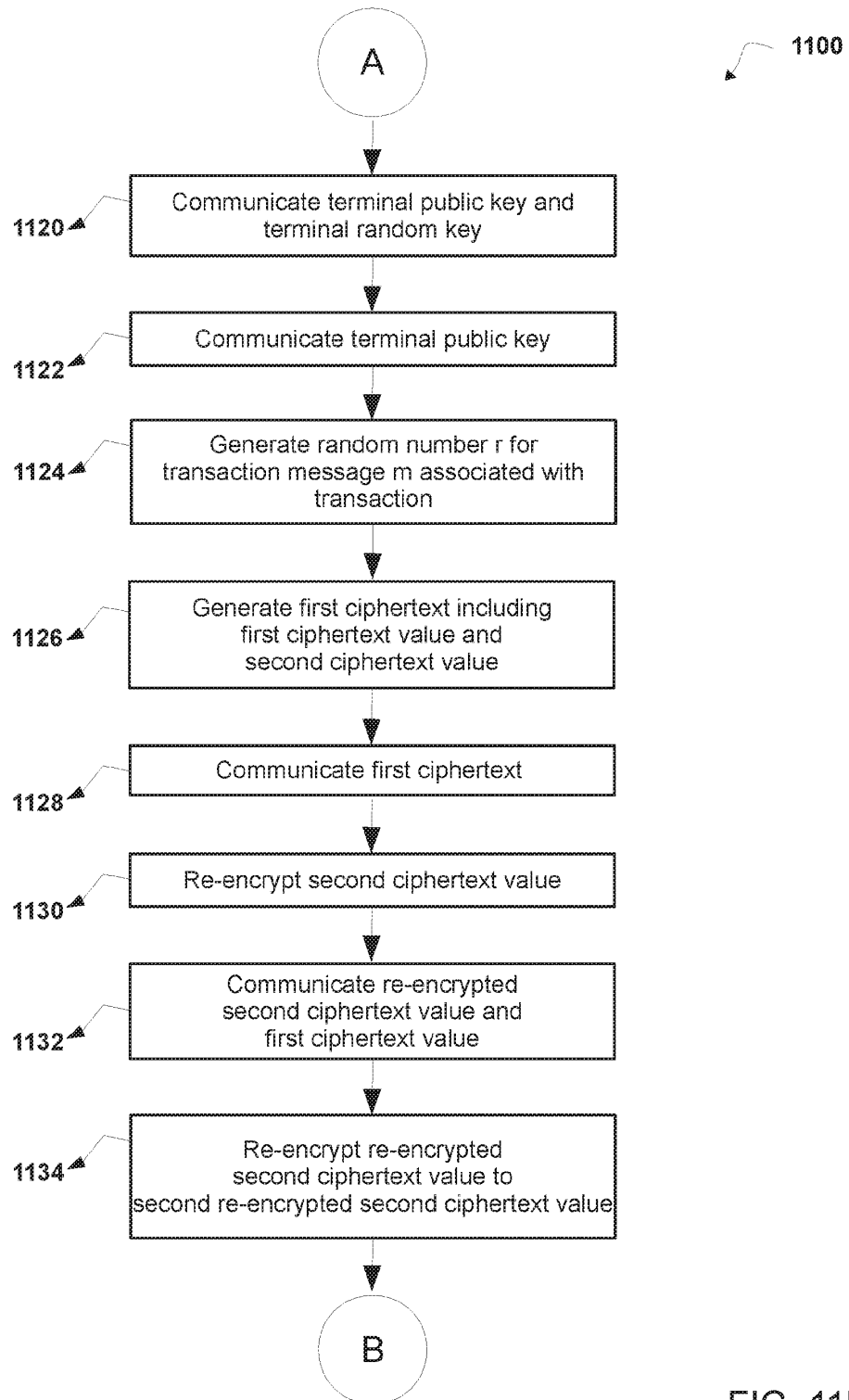


FIG. 11B

13/21

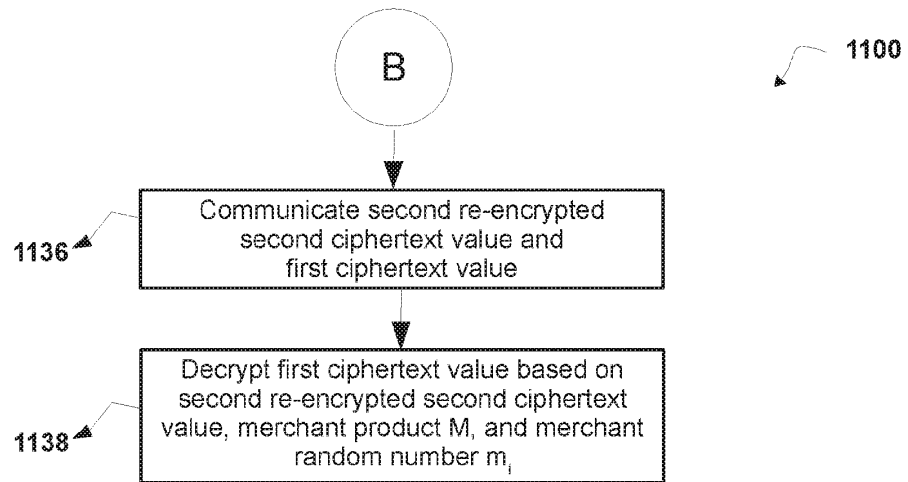


FIG. 11C

14/21

1200

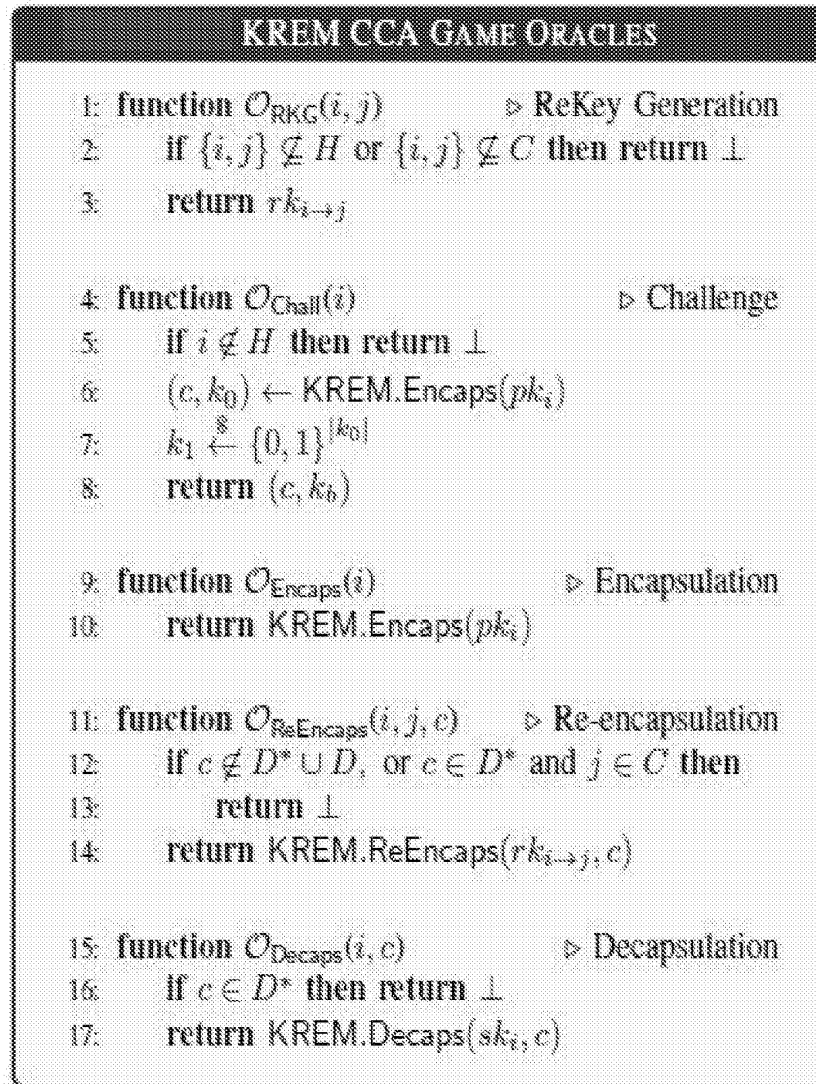


FIG. 12A

15/21

1250

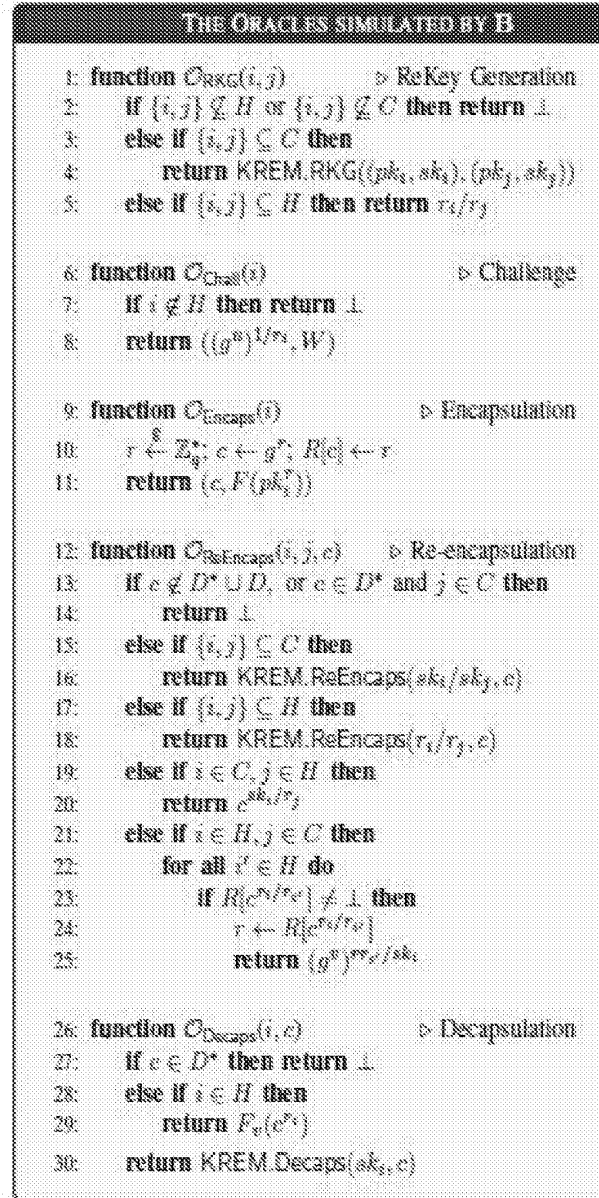


FIG. 12B

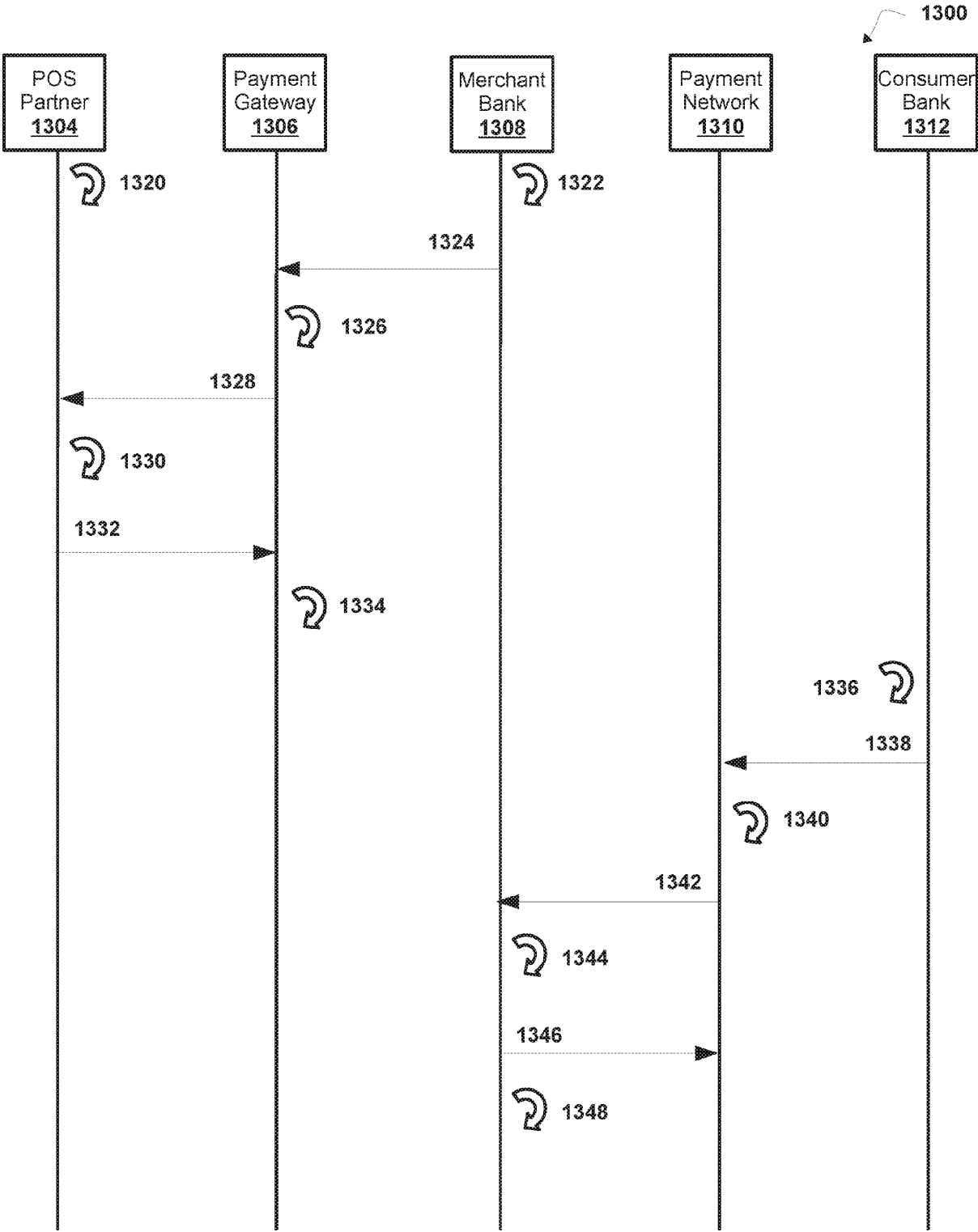


FIG. 13

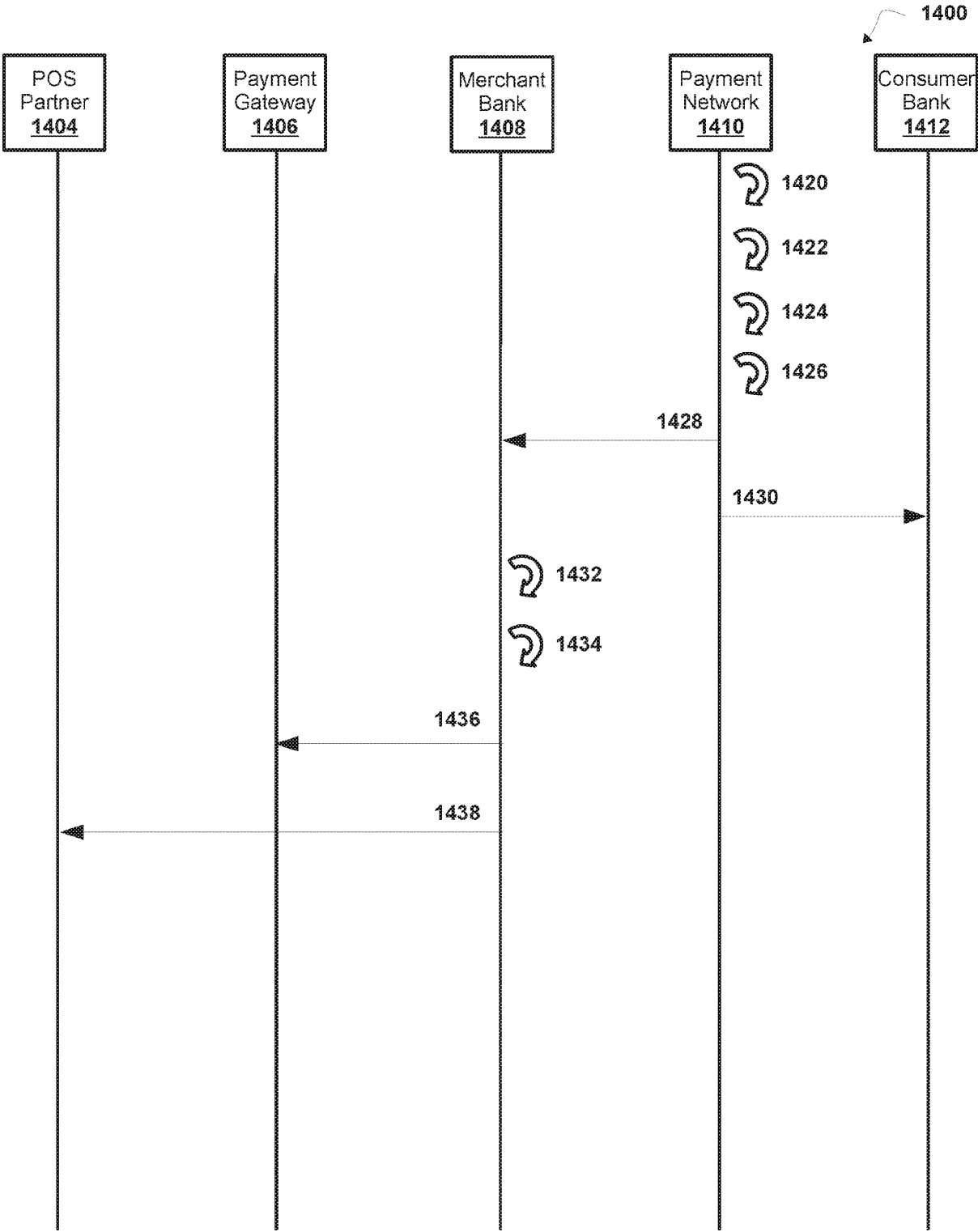


FIG. 14

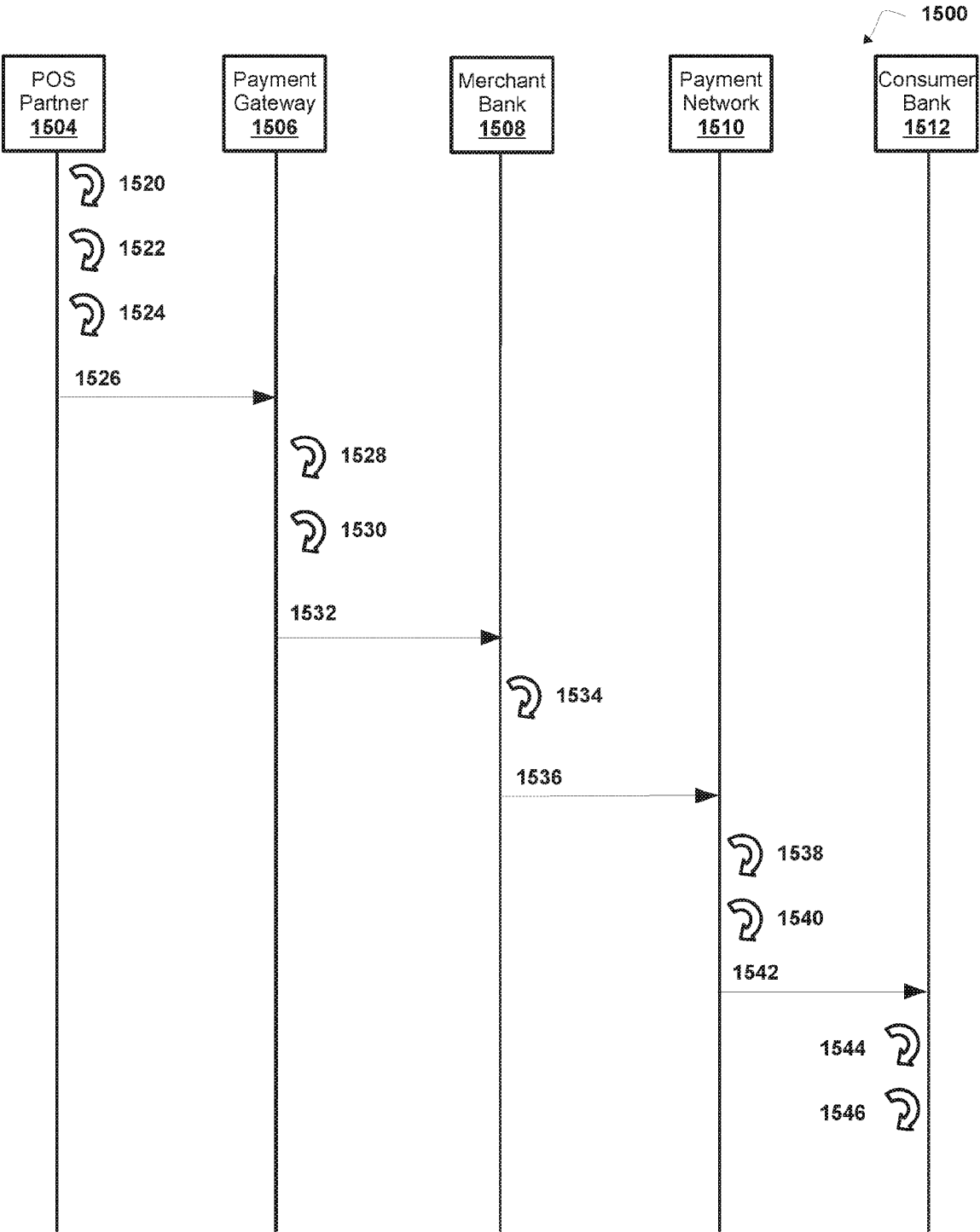


FIG. 15

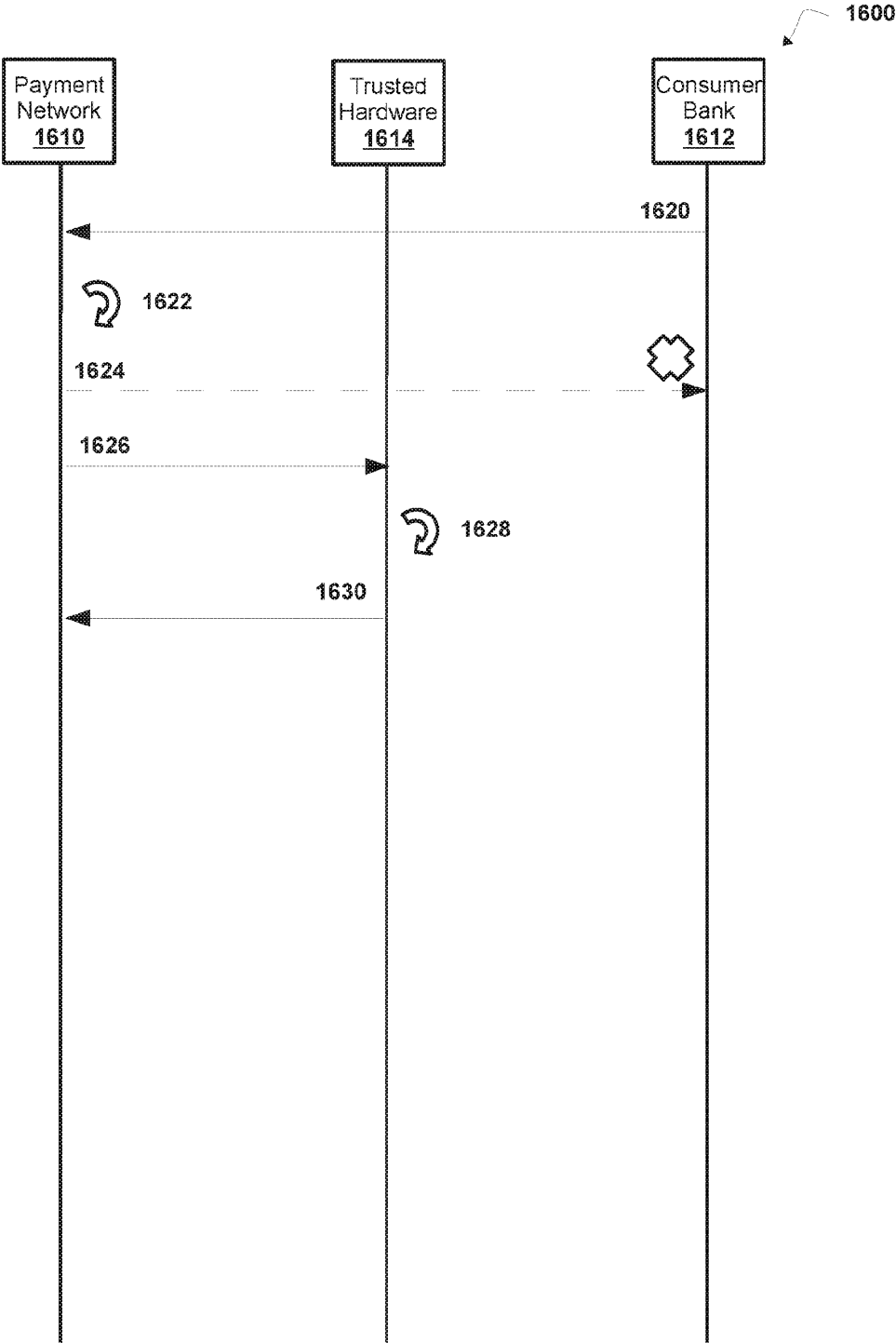


FIG. 16

20/21

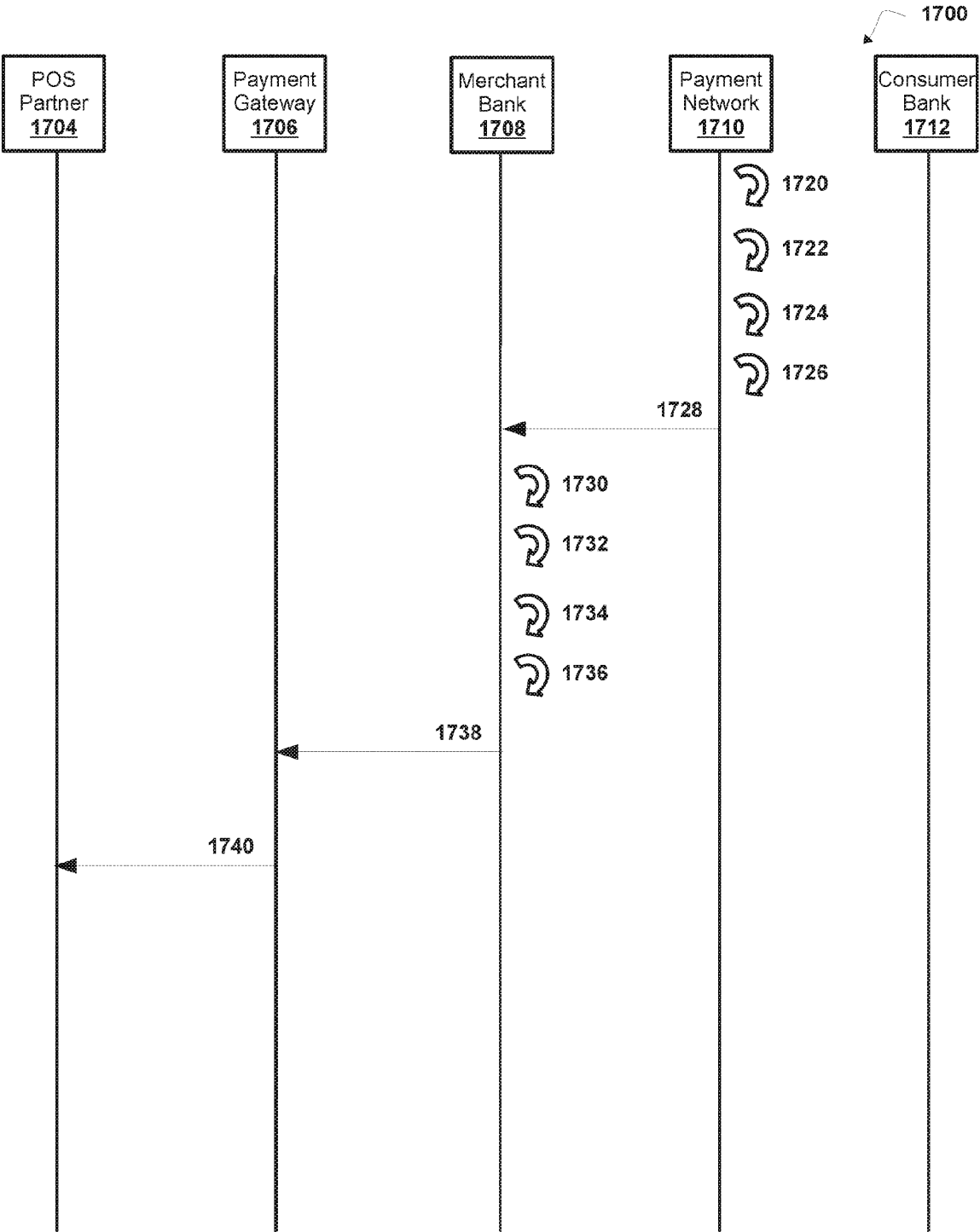


FIG. 17

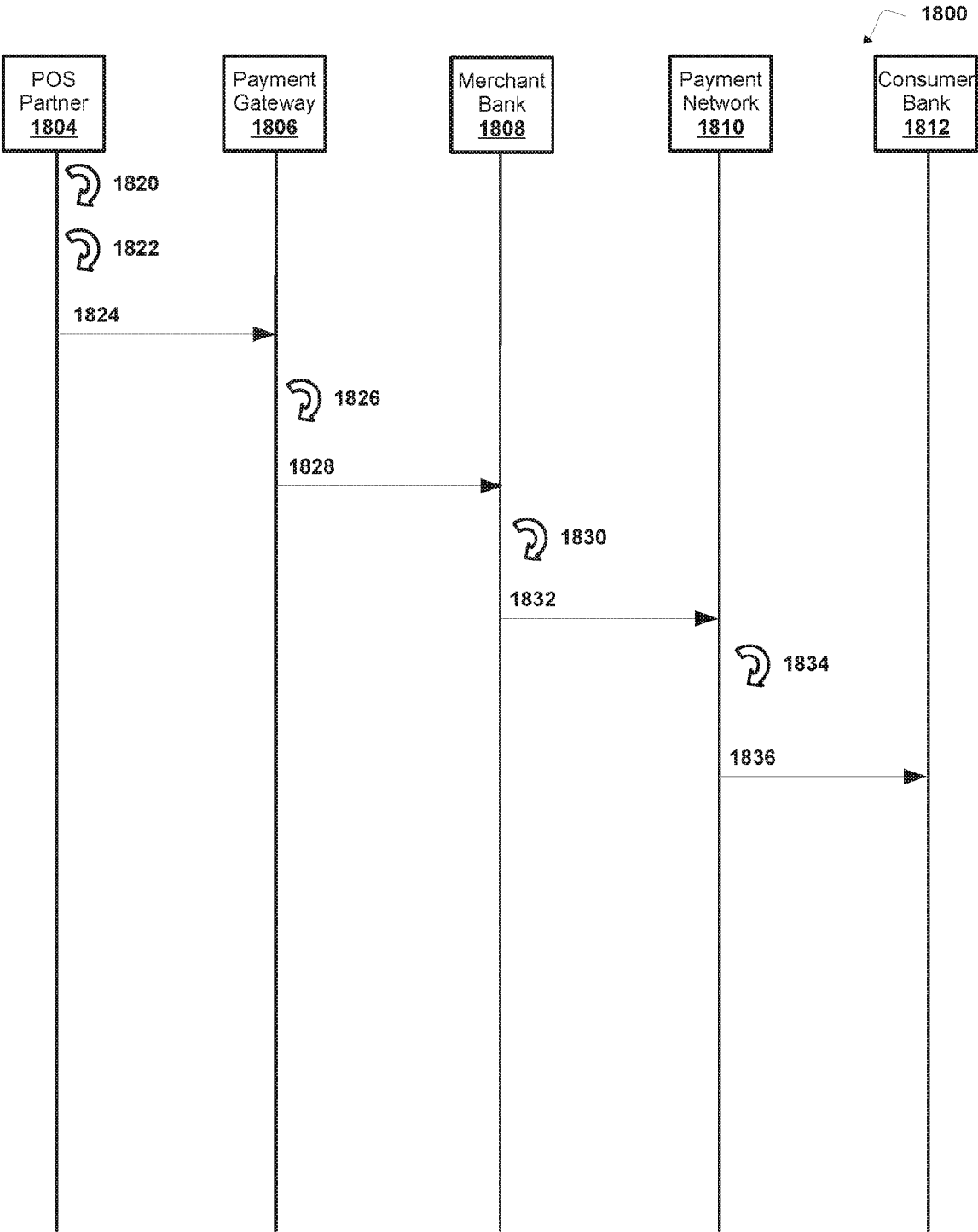


FIG. 18