

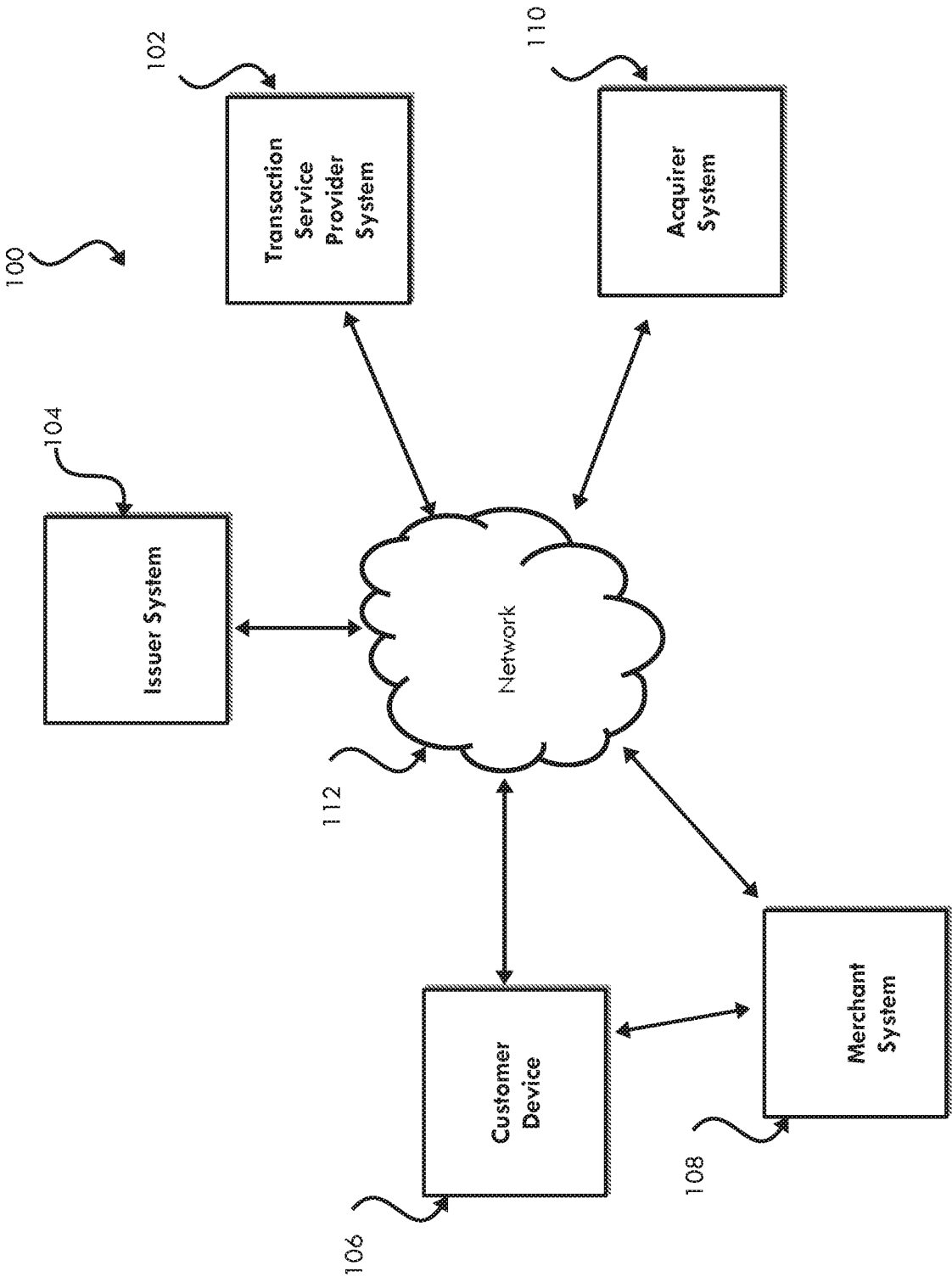


FIG. 1

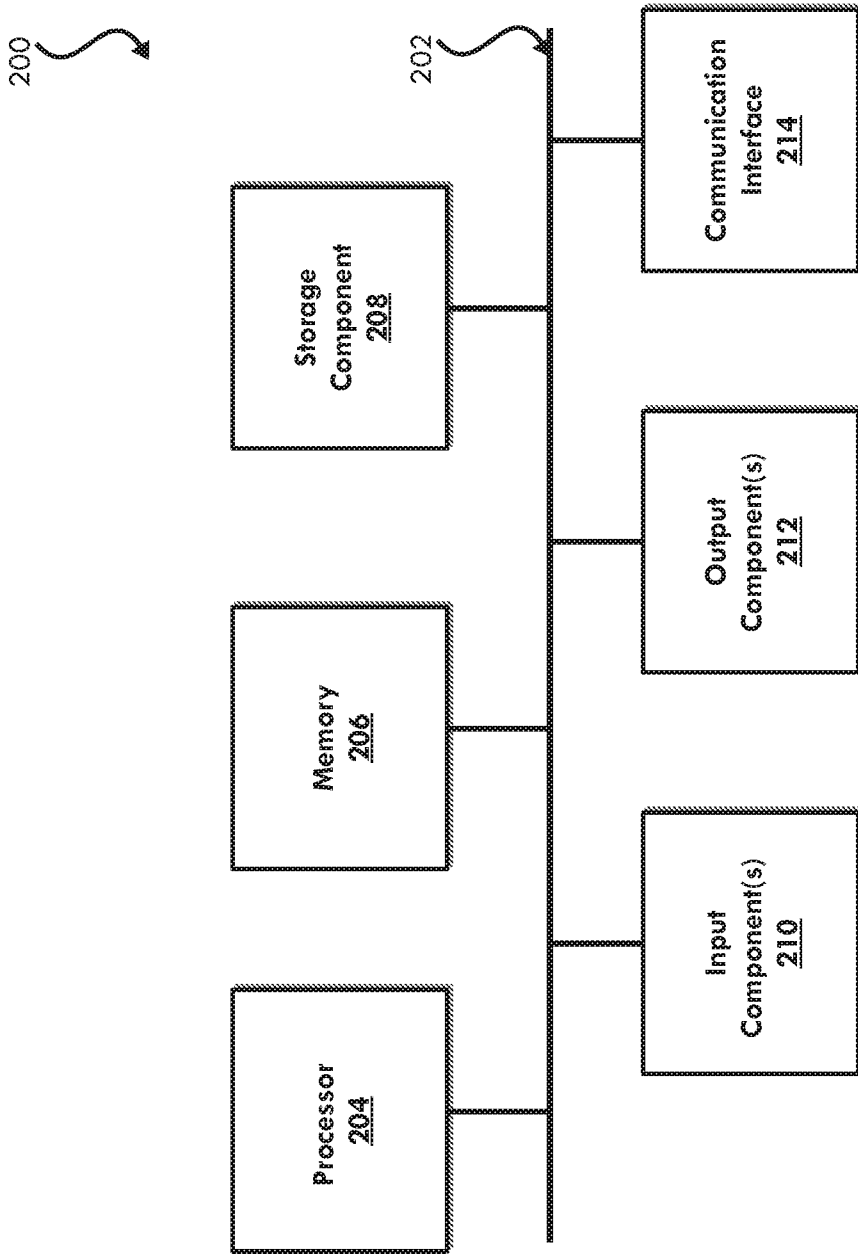


FIG. 2

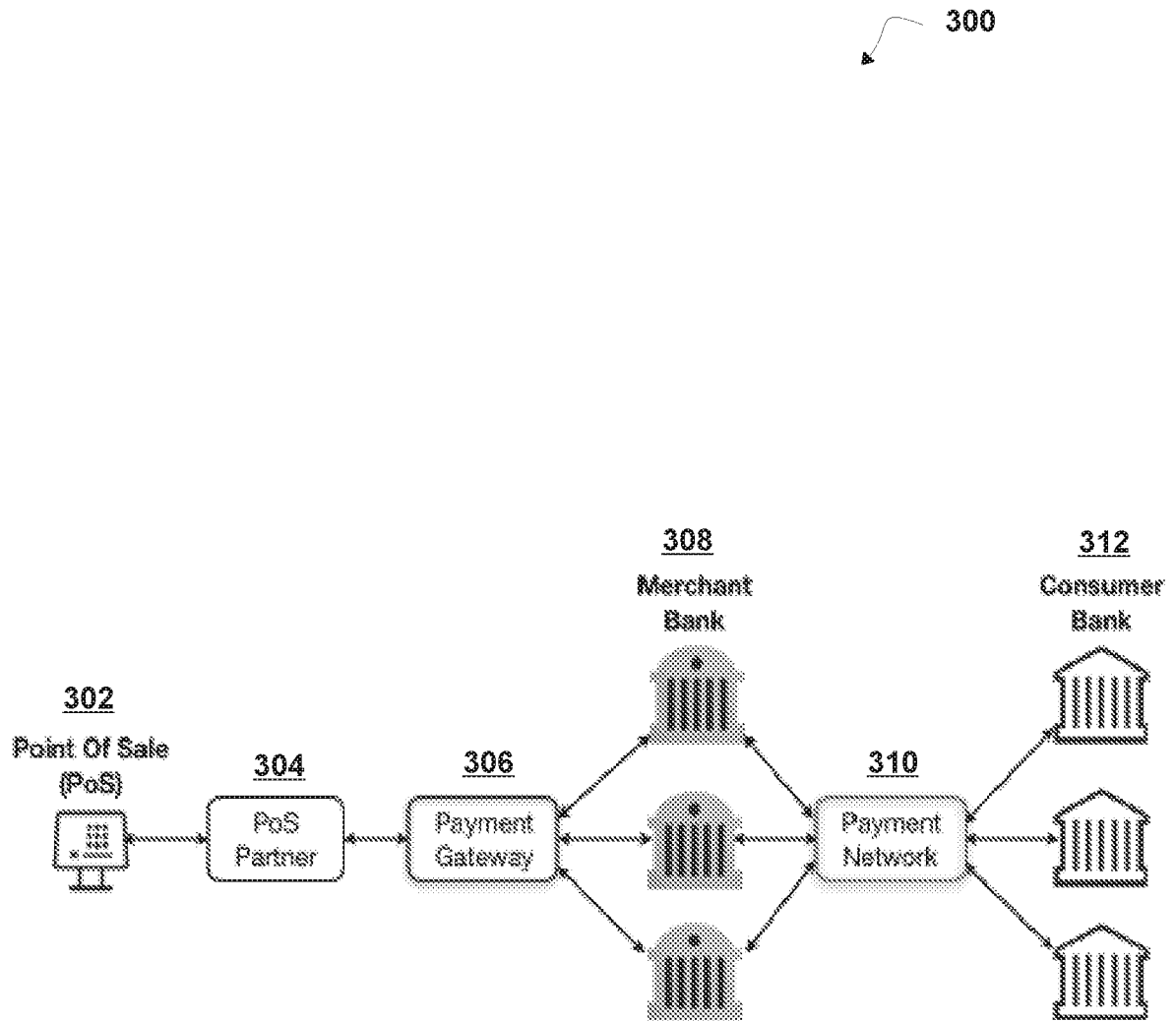


FIG. 3

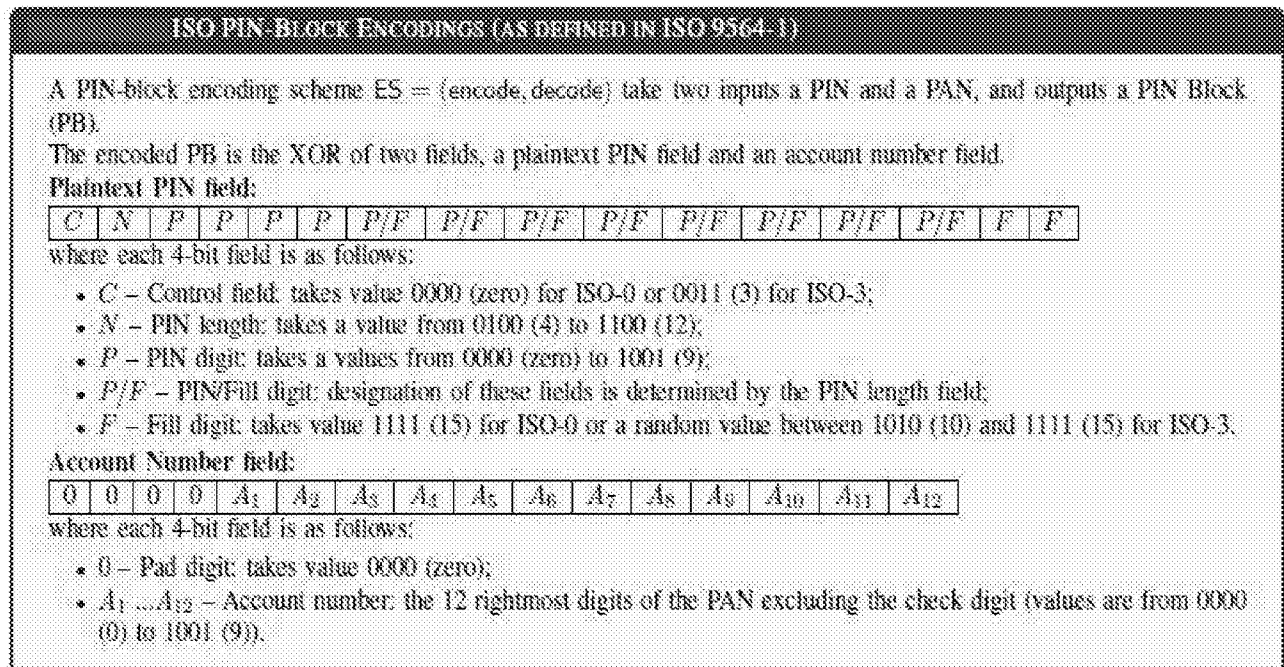


FIG. 4

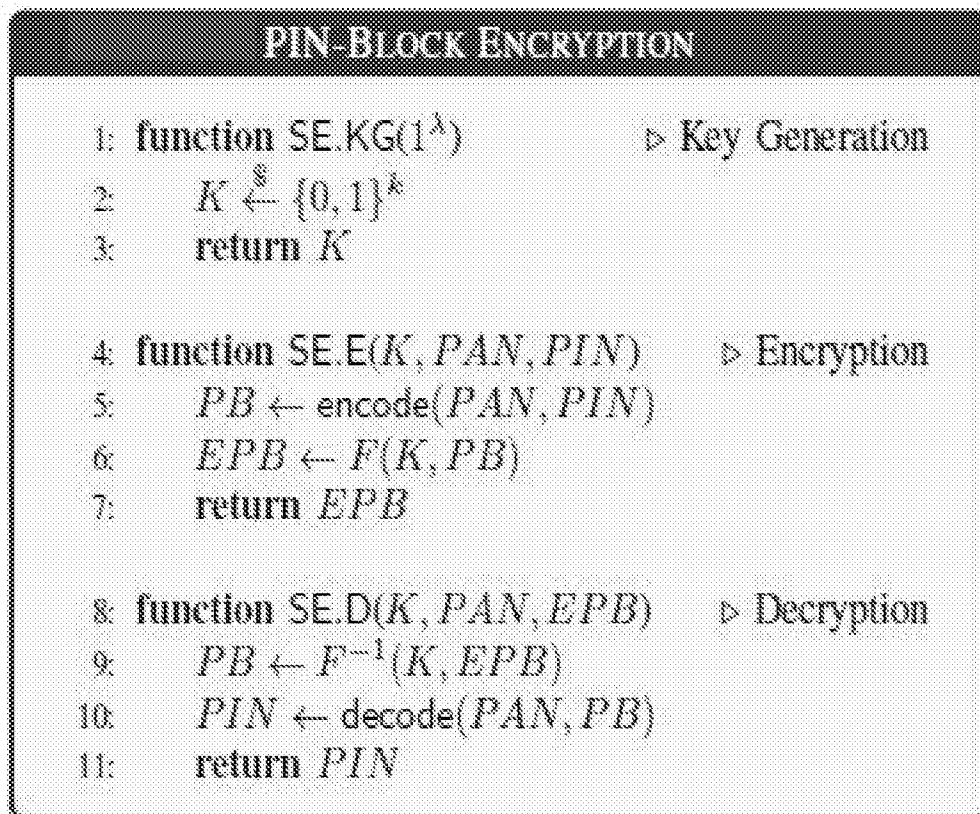


FIG. 5

PRE HRA-CCA GAME ORACLES

- 1: function $\mathcal{O}_{\text{RKG}}^{\text{UNI}}(i, j)$ ▷ Unidirectional Re-key
- 2: if $i \in H, j \in C$ then return $\perp$
- 3: return $\text{PRE.RKG}(sk_i, pk_i, pk_j)$

- 4: function $\mathcal{O}_{\text{RKG}}^{\text{BI}}(i, j)$ ▷ Bidirectional Re-key
- 5: if $\{i, j\} \not\subseteq H$ or $\{i, j\} \not\subseteq C$ then return $\perp$
- 6: return $\text{PRE.RKG}(sk_i, pk_i, sk_j, pk_j)$

- 7: function $\mathcal{O}_{\text{Chall}}(i, m_0, m_1)$ ▷ Challenge
- 8: if $i \notin H$ and $|m_0| \neq |m_1|$ then return $\perp$
- 9: return $\text{PRE.E}(pk_i, m_b)$

- 10: function $\mathcal{O}_{\text{E}}(i, m)$ ▷ Encryption
- 11: return $\text{PRE.E}(pk_i, m)$

- 12: function $\mathcal{O}_{\text{RE}}(i, j, c)$ ▷ Re-encryption
- 13: if $c \notin D^* \cup D$, or $c \in D^*$ and $j \in C$ then
- 14: return $\perp$
- 15: return $\text{PRE.RE}(rk_{i \rightarrow j}, c)$

- 16: function $\mathcal{O}_{\text{D}}(j, c)$ ▷ Decryption
- 17: if $c \in D^*$ then return $\perp$
- 18: return $\text{PRE.D}(sk_j, c)$

FIG. 6

PPA-HRCCA GAME ORACLES

- 1: function $\mathcal{O}_{\text{Enroll}}^{\text{UNI}}(i, j)$ ▷ Unidirectional Enroll
- 2: if $i \in H, j \in C$ then return $\perp$
- 3: return $\text{PPA.Enroll}(sk_i, pk_i, pk_j)$

- 4: function $\mathcal{O}_{\text{Enroll}}^{\text{BI}}(i, j)$ ▷ Bidirectional Enroll
- 5: if $\{i, j\} \not\subseteq H$ or $\{i, j\} \not\subseteq C$ then return $\perp$
- 6: return $\text{PPA.Enroll}(sk_i, pk_i, sk_j, pk_j)$

- 7: function $\mathcal{O}_{\text{Chall}}(i, a, m_0, m_1)$ ▷ Challenge
- 8: if $i \notin H$ and $|m_0| \neq |m_1|$ then return $\perp$
- 9: return $\text{PPA.Send}(pk_i, m_b)$

- 10: function $\mathcal{O}_{\text{Send}}(i, m)$ ▷ Send
- 11: return $\text{PPA.Send}(pk_i, m)$

- 12: function $\mathcal{O}_{\text{Relay}}(i, j, c)$ ▷ Relay
- 13: if $c \notin D^* \cup D$, or $c \in D^*$ and $j \in C$ then
- 14: return $\perp$
- 15: return $\text{PPA.Relay}(rk_{i \rightarrow j}, c)$

- 16: function $\mathcal{O}_{\text{Verify}}(j, c)$ ▷ Verify
- 17: if $c \in D^*$ then return $\perp$
- 18: return $\text{PPA.Verify}(sk_j, c)$

FIG.7

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

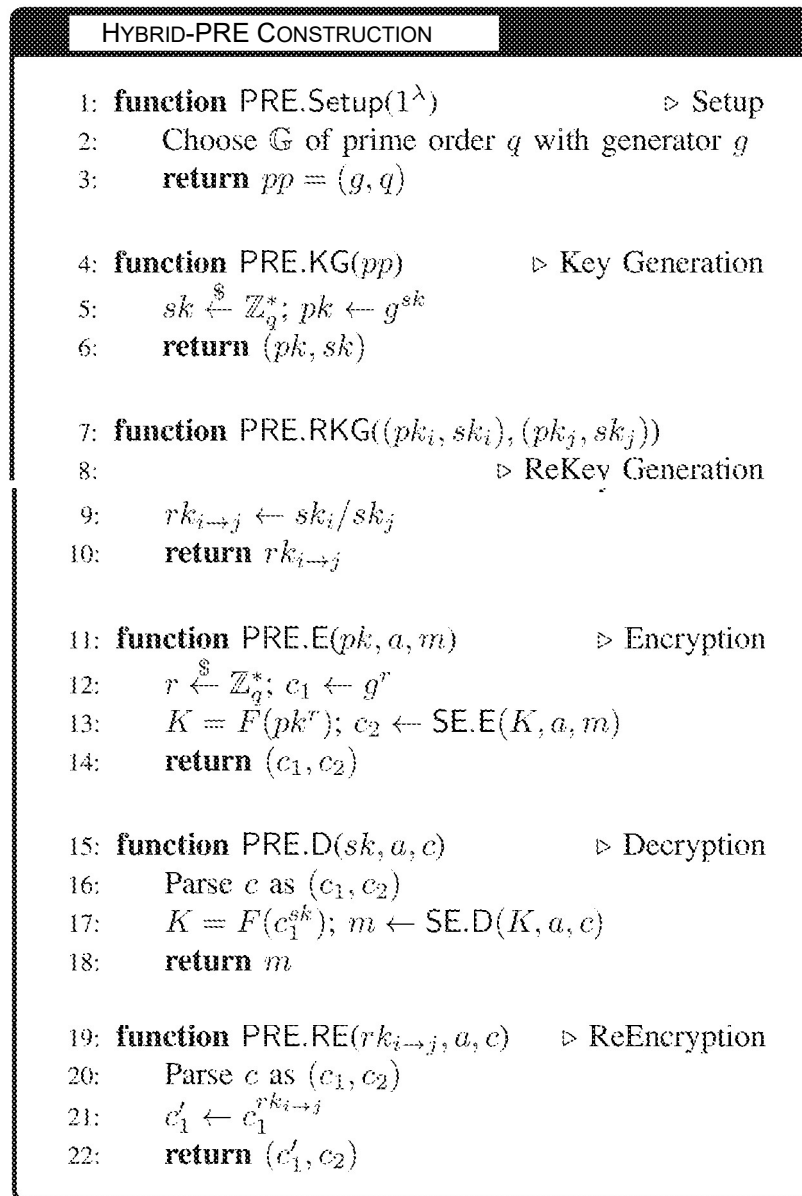


FIG. 9

10/21

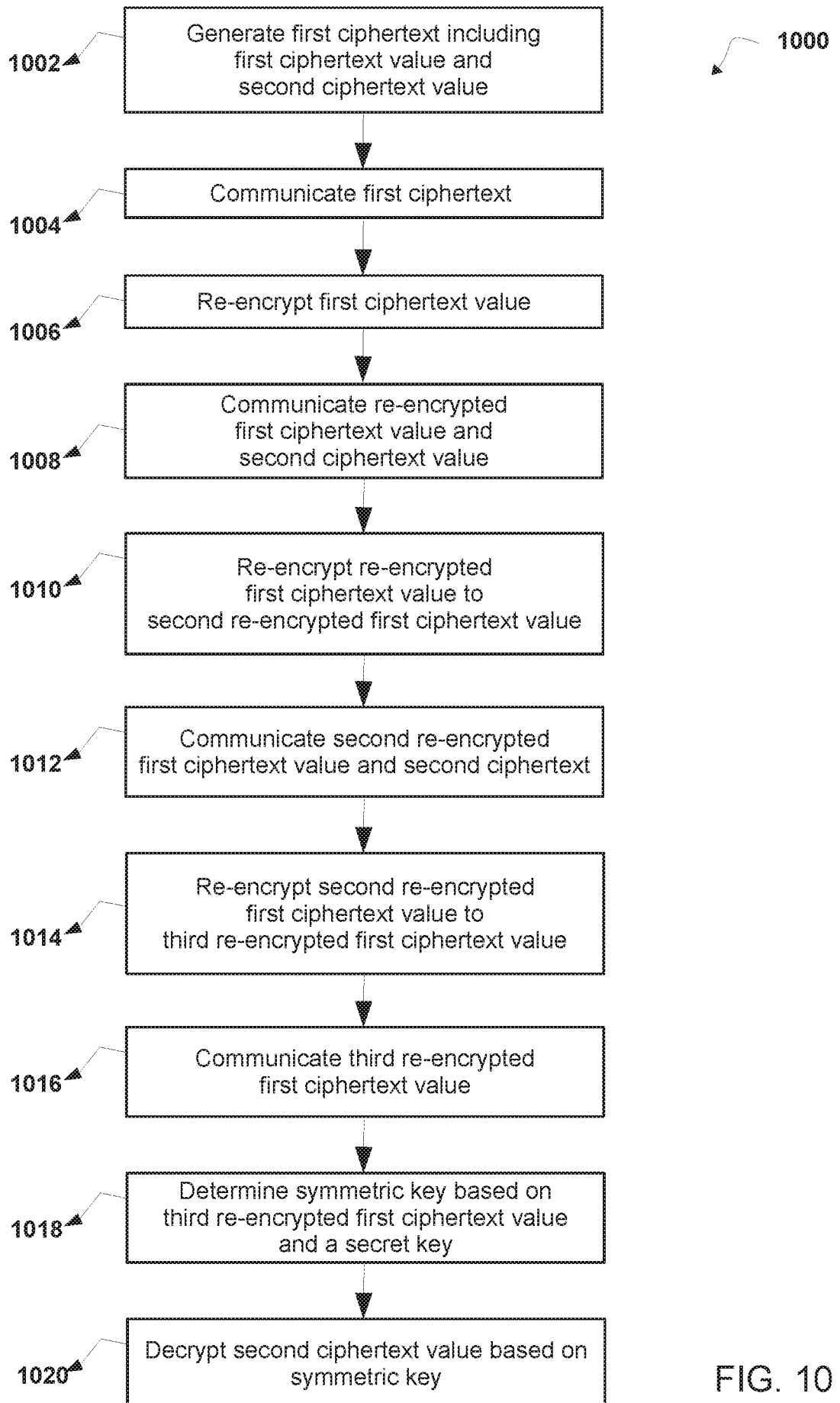


FIG. 10

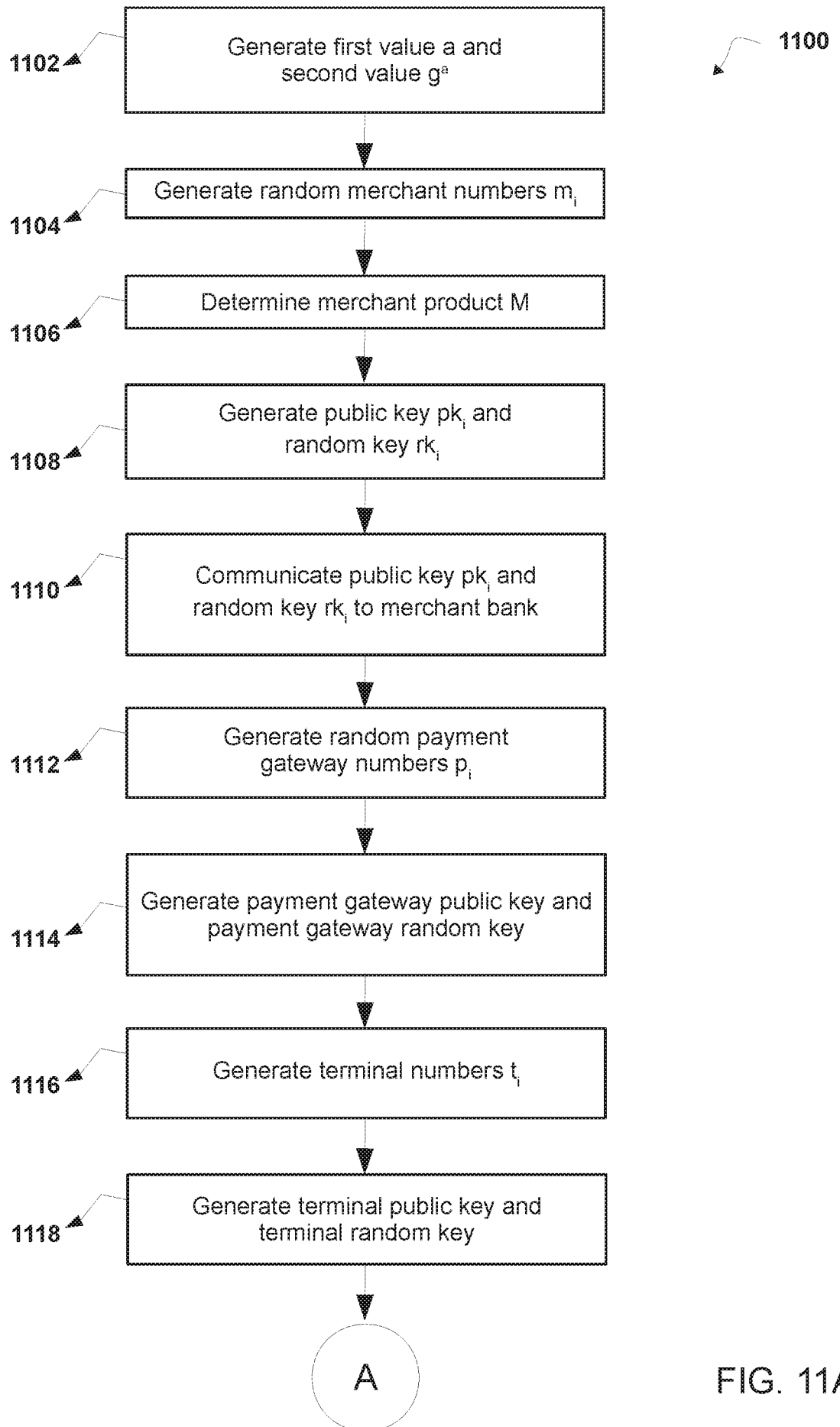


FIG. 11A

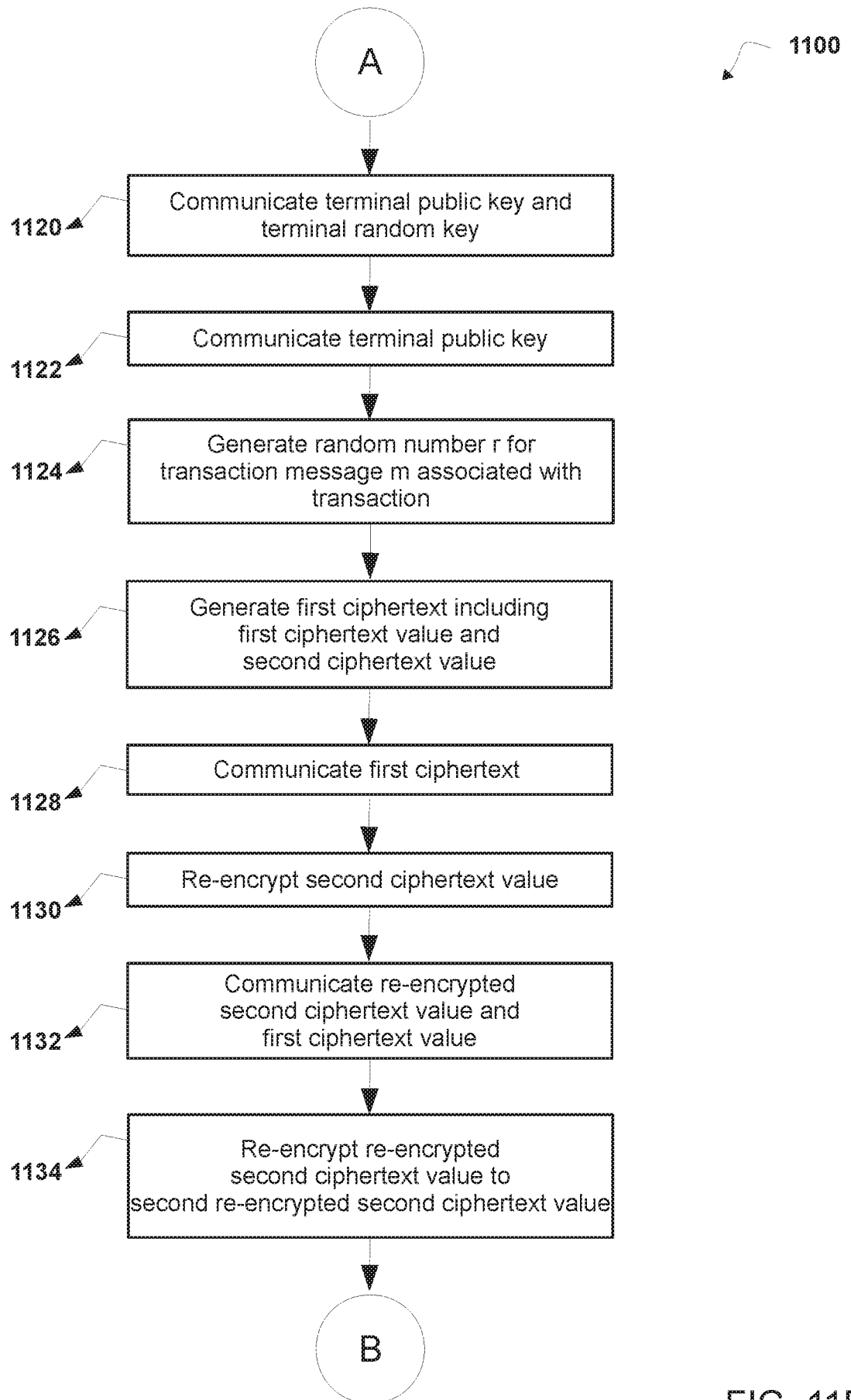


FIG. 11B

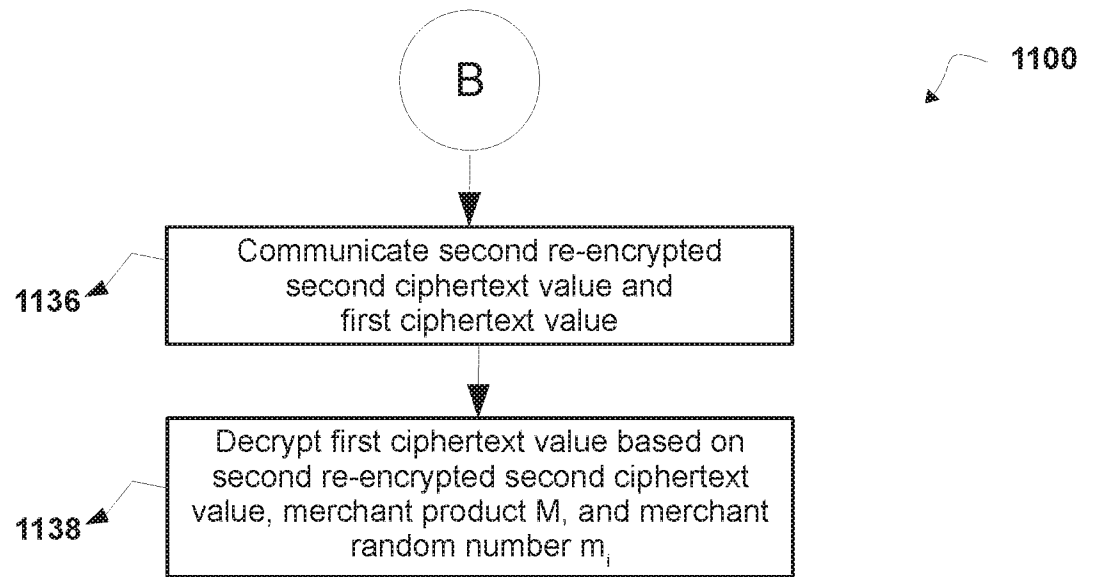


FIG. 11C

KREM CCA GAME ORACLES

- 1: function $\mathcal{O}_{\text{RKG}}(i, j)$ ▷ ReKey Generation
- 2: if $\{i, j\} \not\subseteq H$ or $\{i, j\} \not\subseteq C$ then return $\perp$
- 3: return $rk_{i \rightarrow j}$

- 4: function $\mathcal{O}_{\text{Chall}}(i)$ ▷ Challenge
- 5: if $i \notin H$ then return $\perp$
- 6: $(c, k_0) \leftarrow \text{KREM.Encaps}(pk_i)$
- 7: $k_1 \xleftarrow{\$} \{0, 1\}^{|k_0|}$
- 8: return (c, k_b)

- 9: function $\mathcal{O}_{\text{Encaps}}(i)$ ▷ Encapsulation
- 10: return $\text{KREM.Encaps}(pk_i)$

- 11: function $\mathcal{O}_{\text{ReEncaps}}(i, j, c)$ ▷ Re-encapsulation
- 12: if $c \notin D^* \cup D$, or $c \in D^*$ and $j \in C$ then
- 13: return $\perp$
- 14: return $\text{KREM.ReEncaps}(rk_{i \rightarrow j}, c)$

- 15: function $\mathcal{O}_{\text{Decaps}}(i, c)$ ▷ Decapsulation
- 16: if $c \in D^*$ then return $\perp$
- 17: return $\text{KREM.Decaps}(sk_i, c)$

FIG. 12A

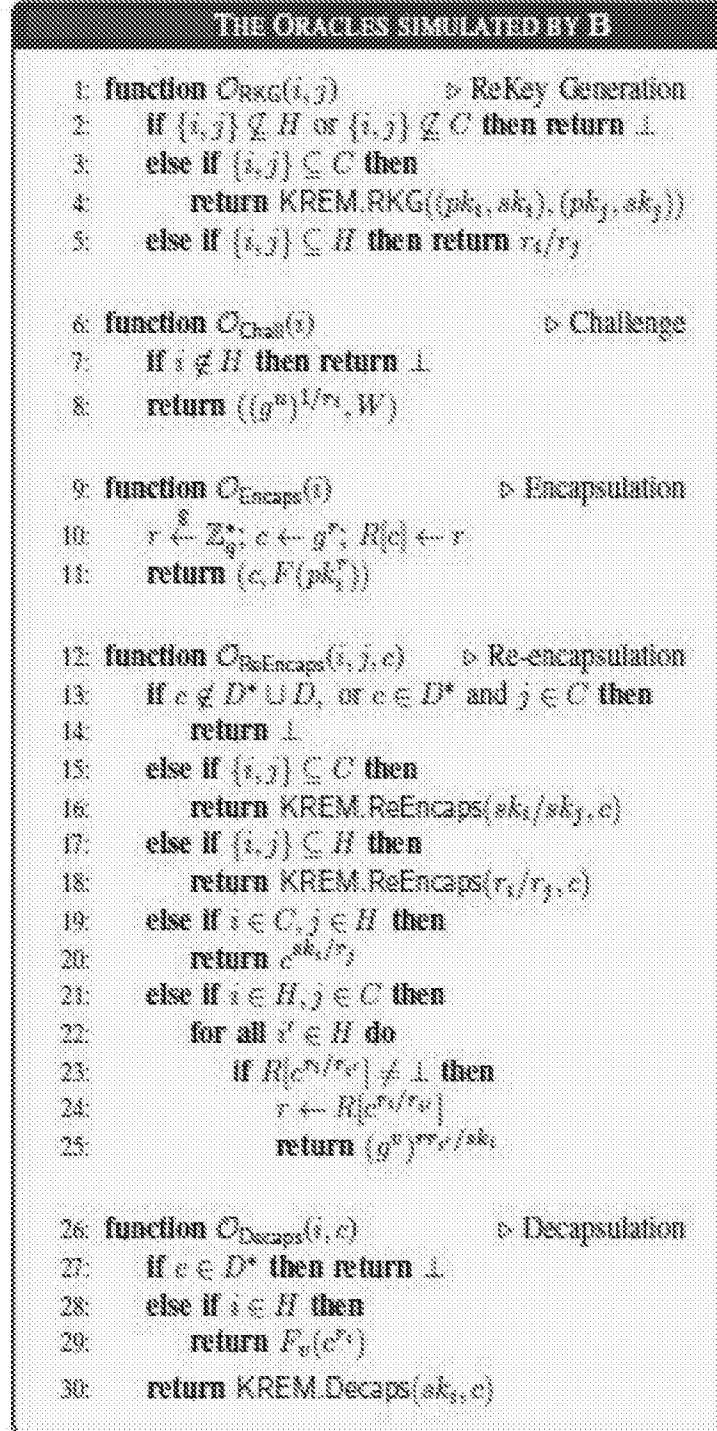


FIG. 12B

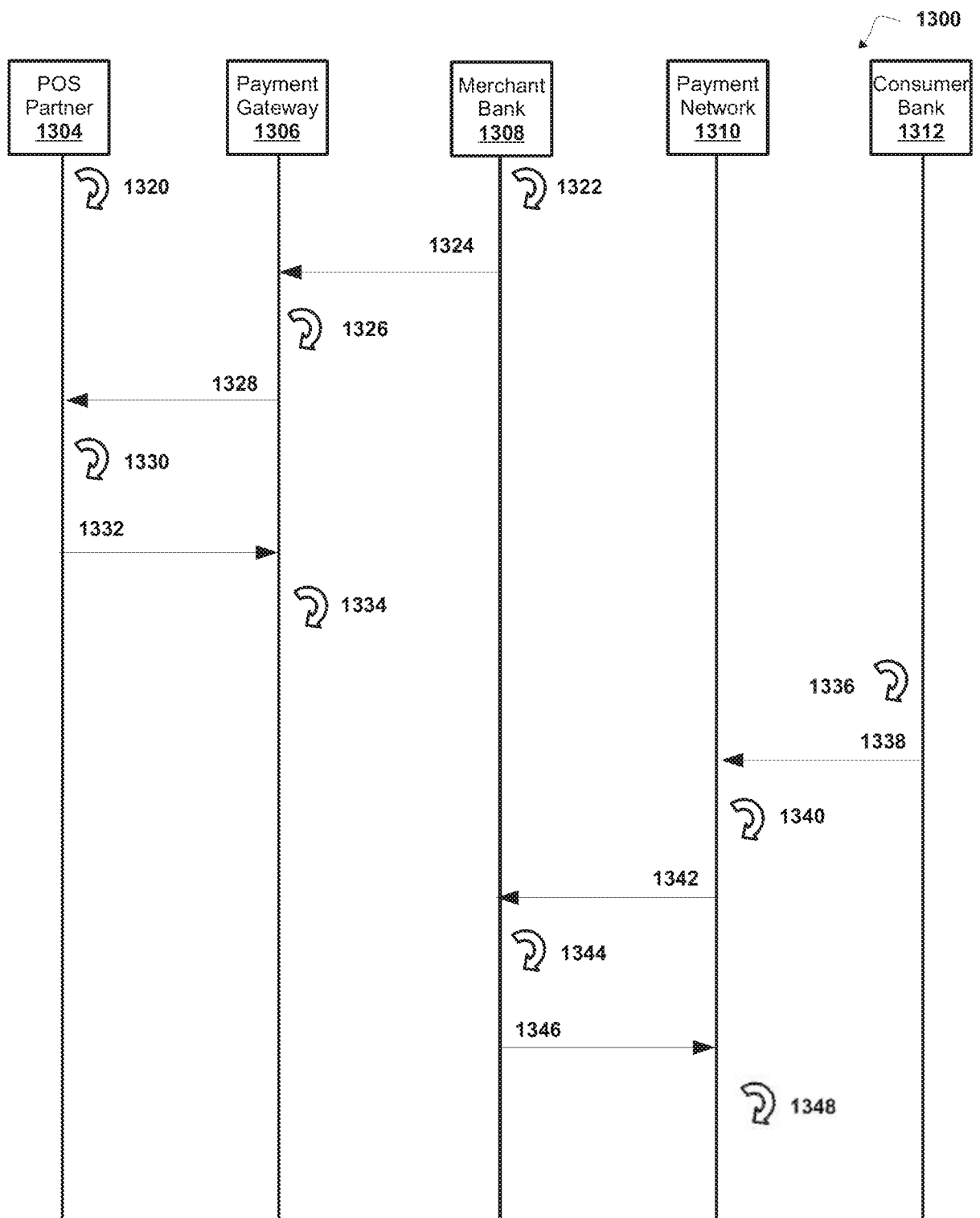


FIG. 13

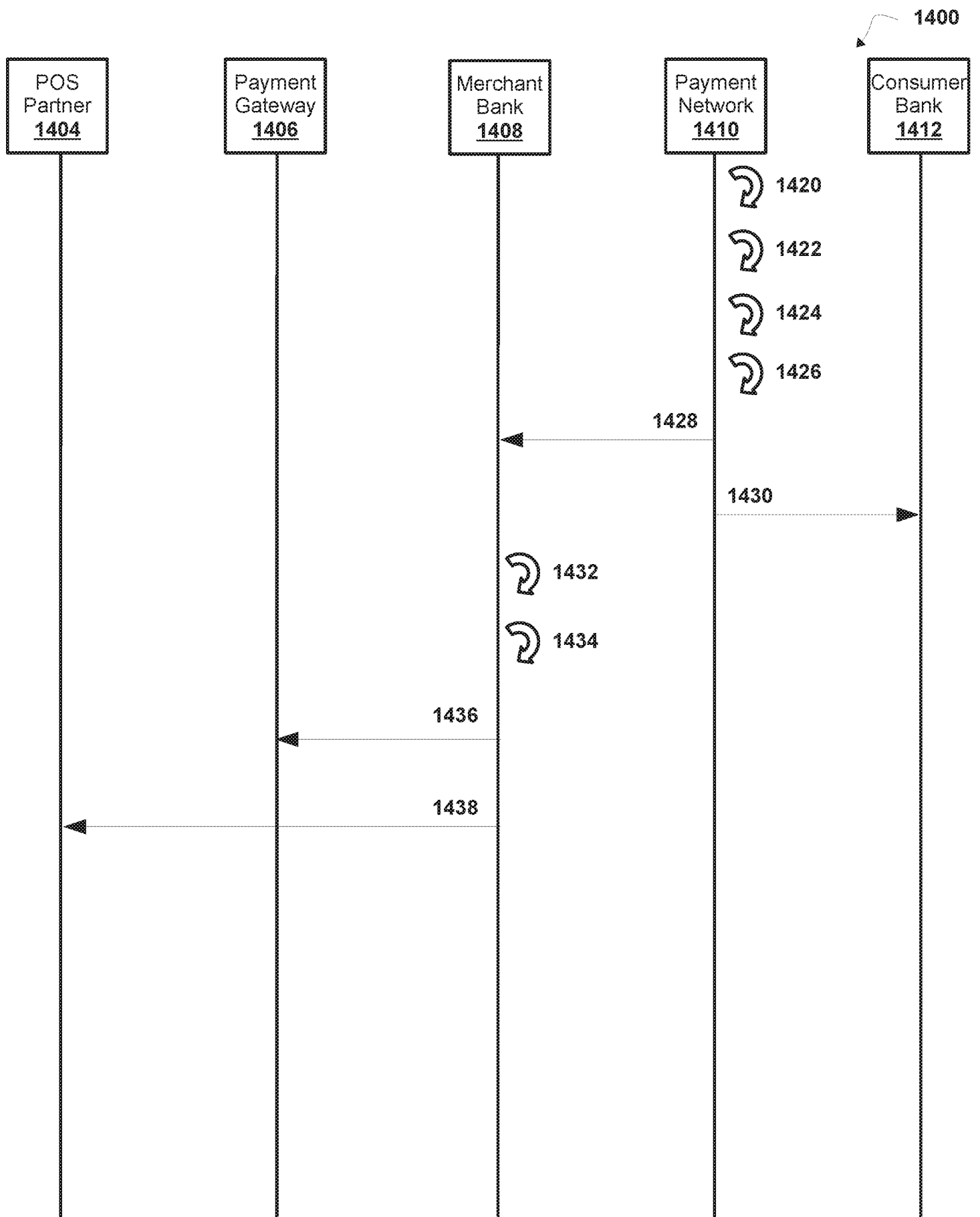


FIG. 14

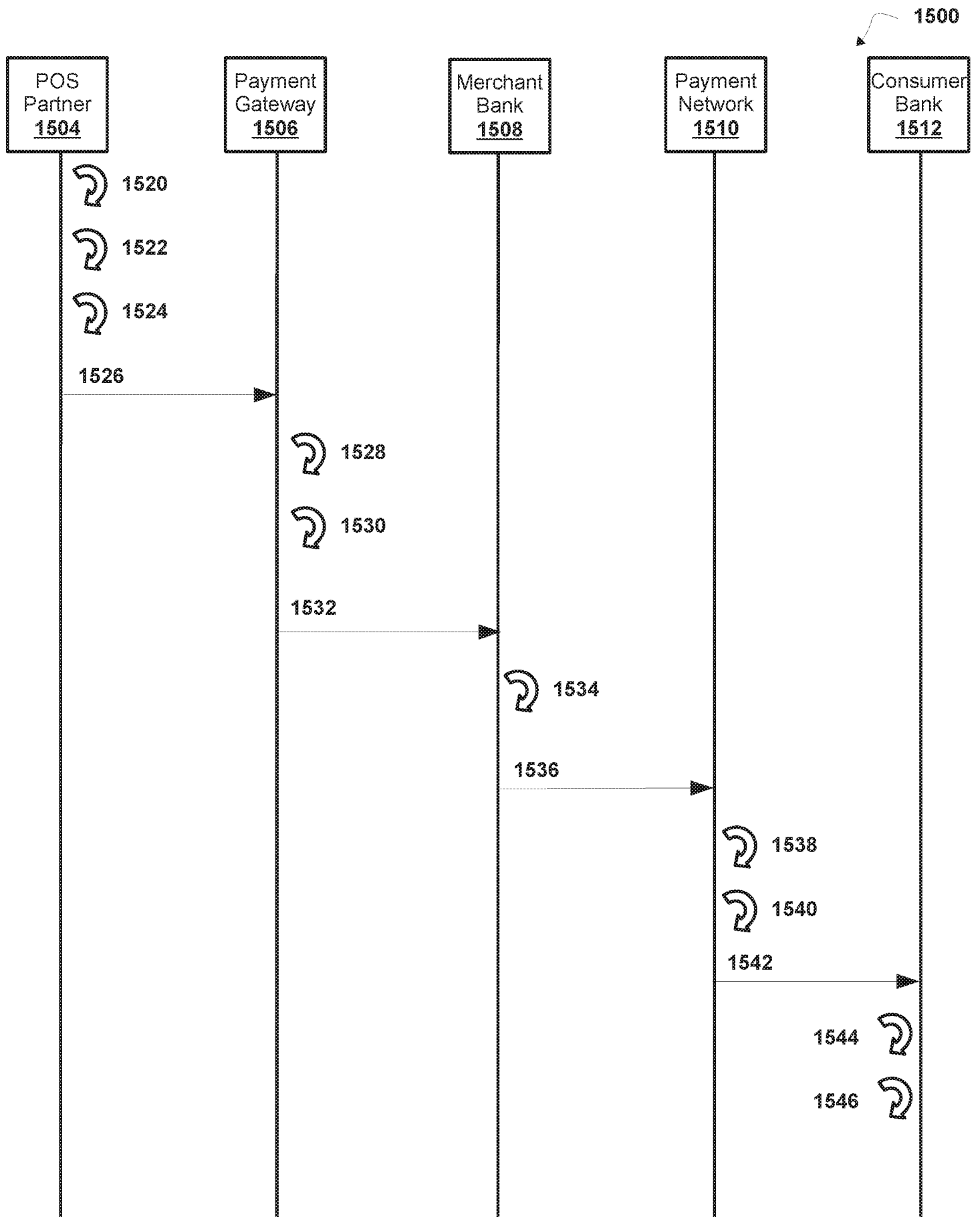


FIG. 15

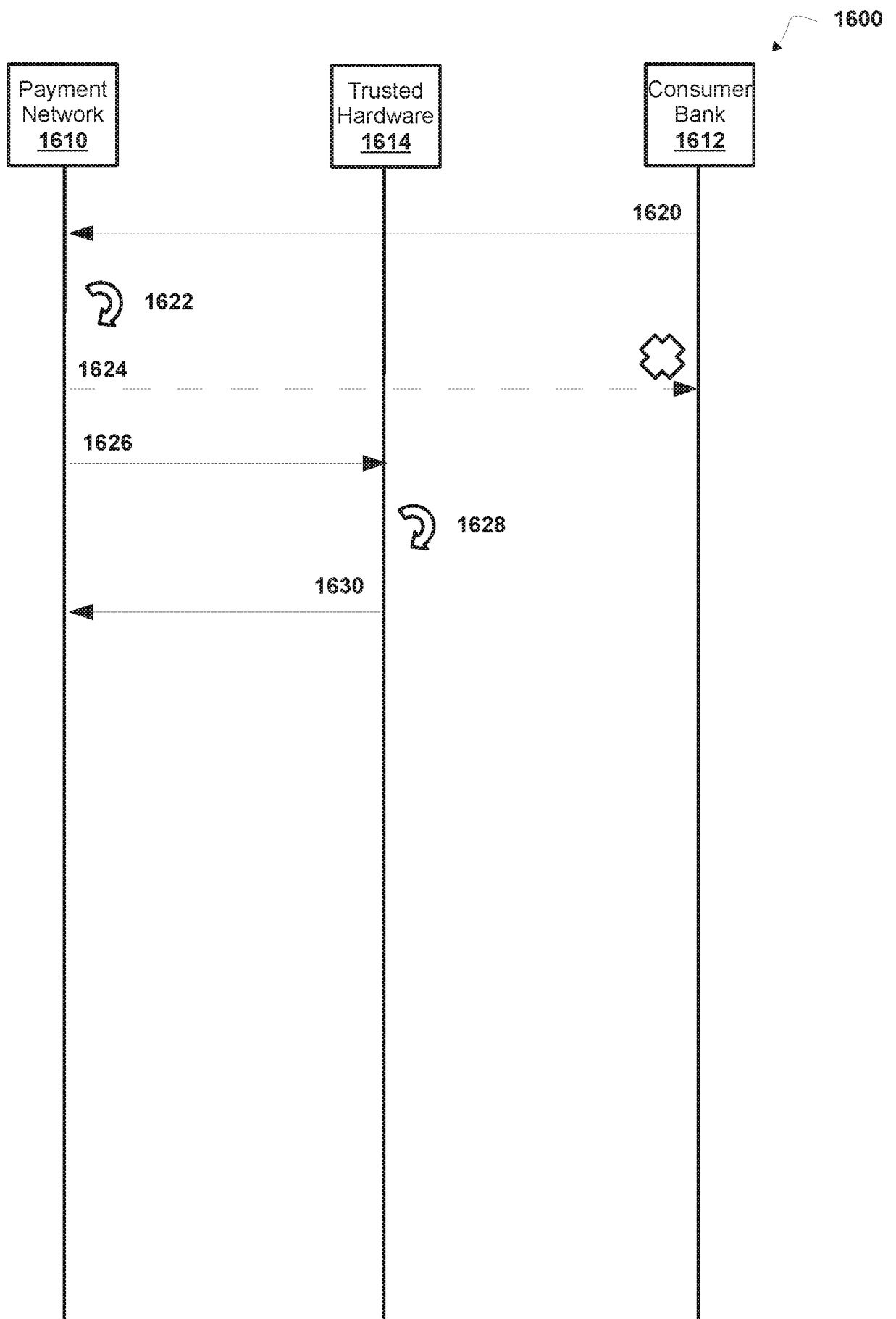


FIG. 16

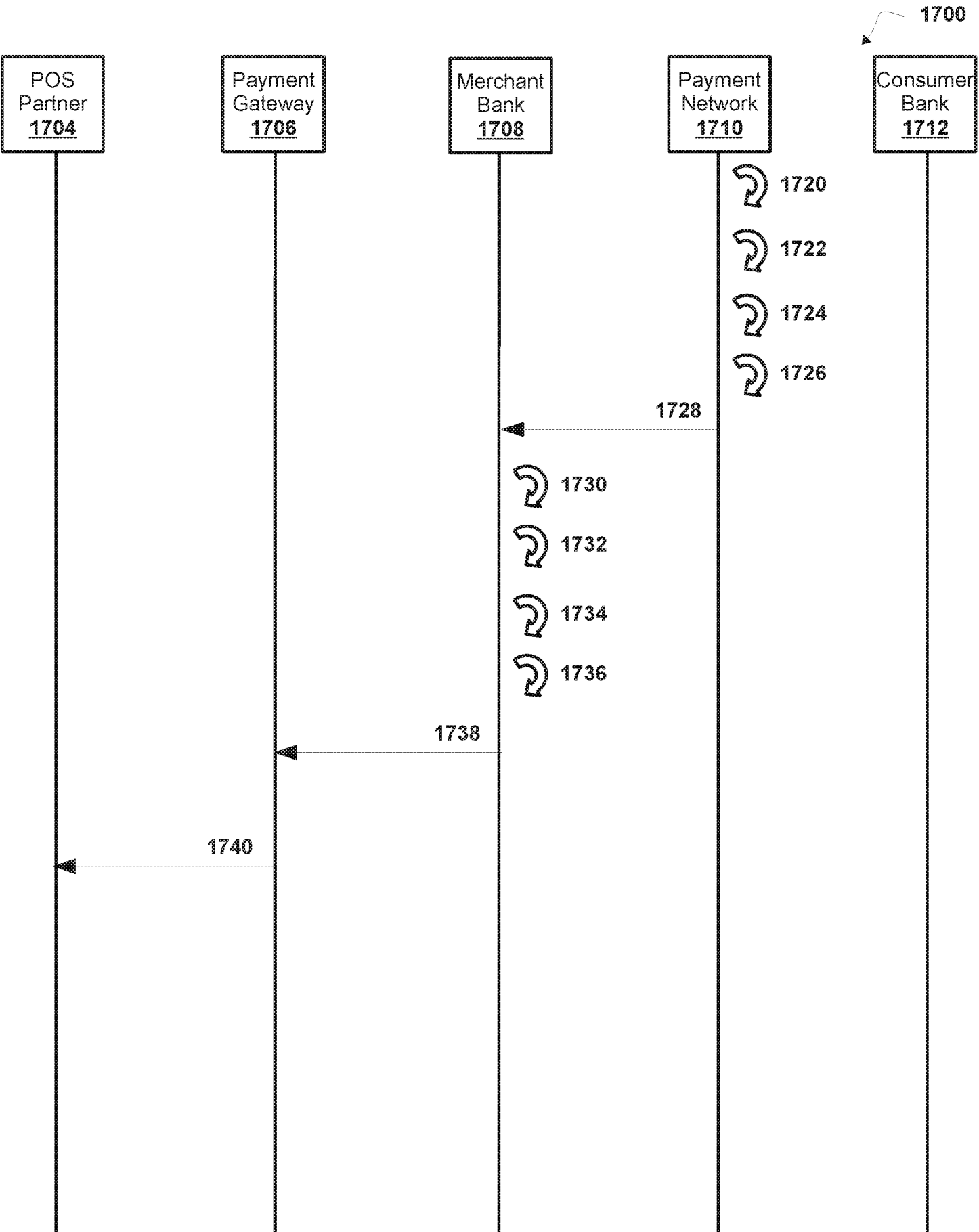


FIG. 17

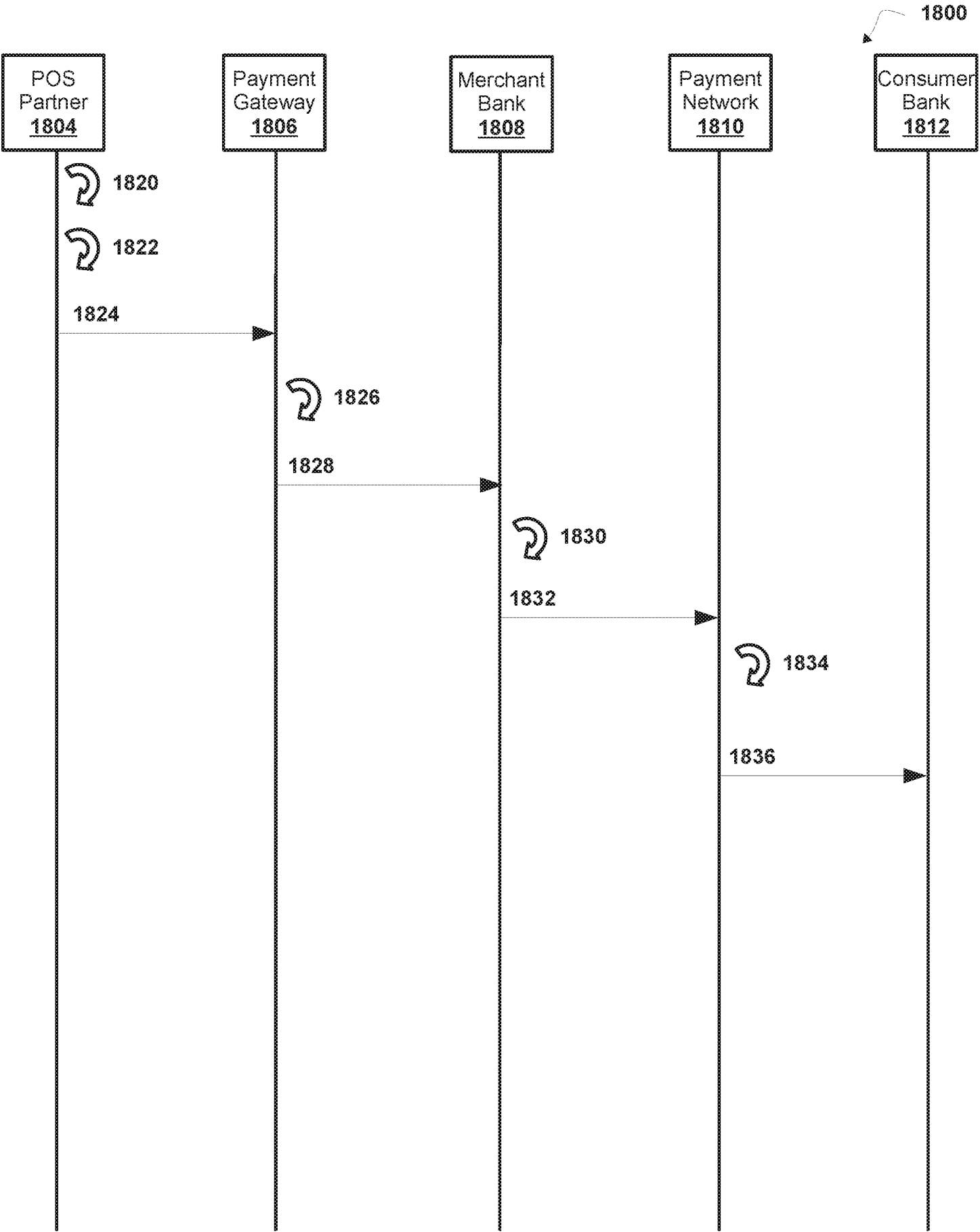


FIG. 18