

Figure 1

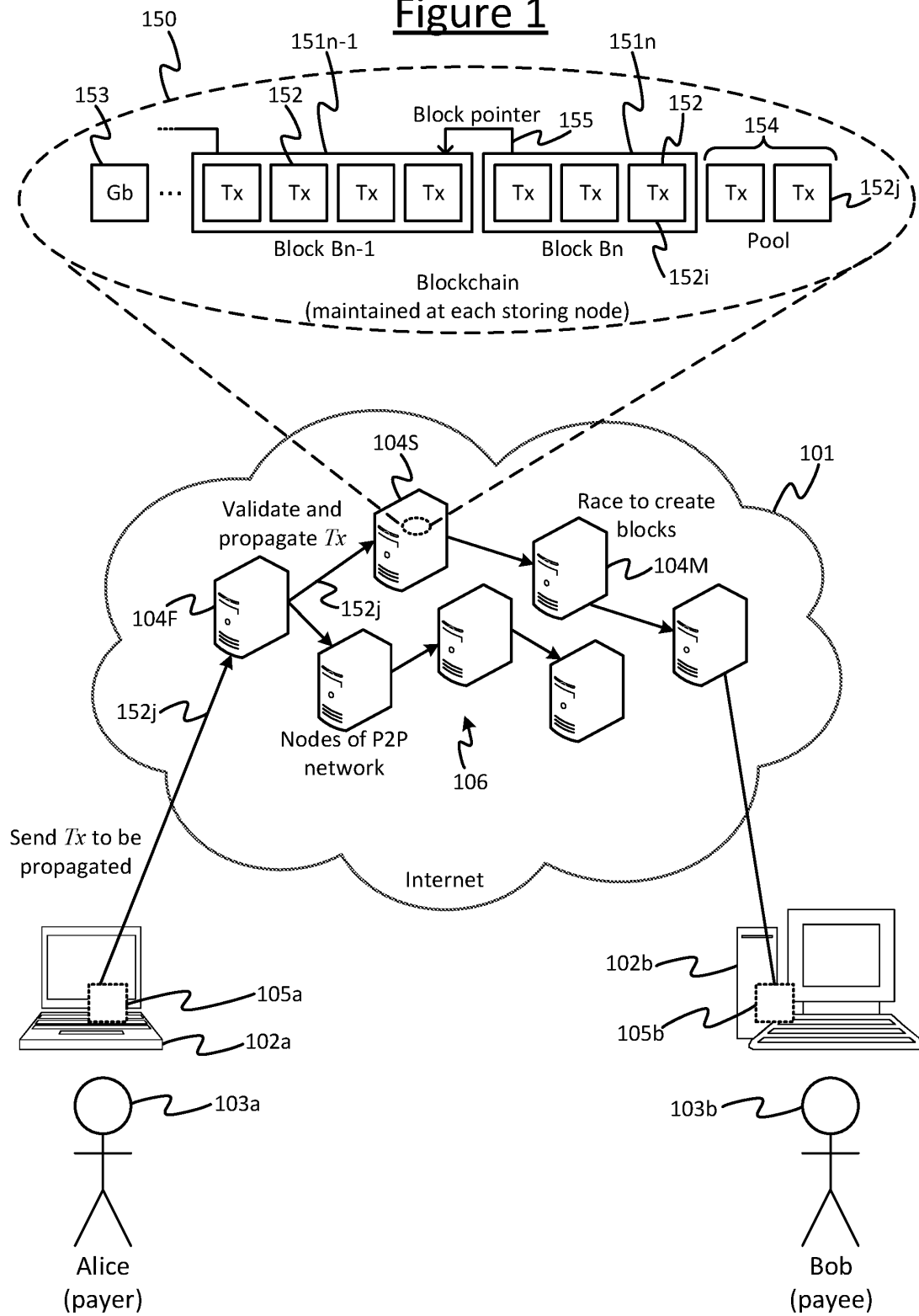
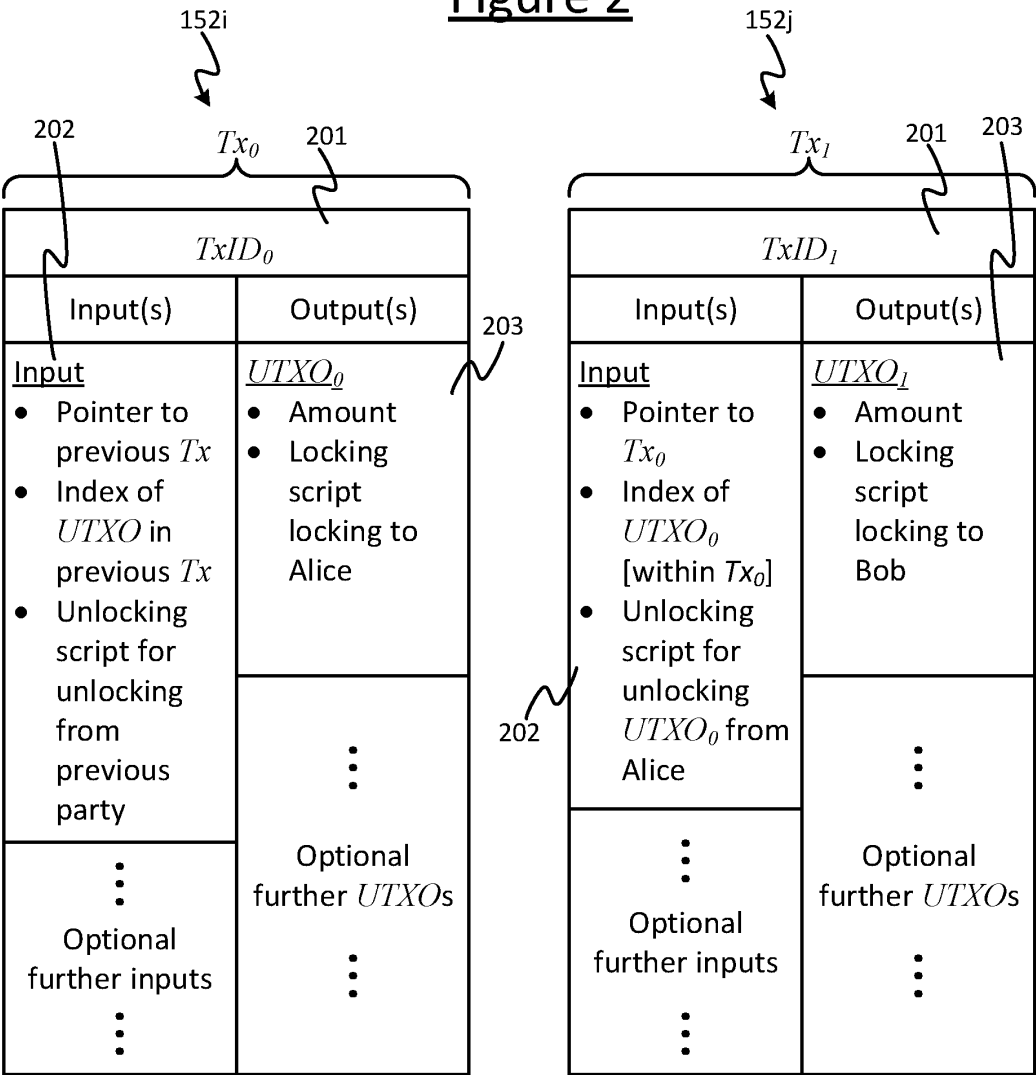


Figure 2



Transaction
from Alice to Bob



Validated by running: Alice's locking script (from output of Tx_0), together with Alice's unlocking script (as input to Tx_1). This checks that Tx_1 meets the condition(s) defined in Alice's locking script.

Figure 3

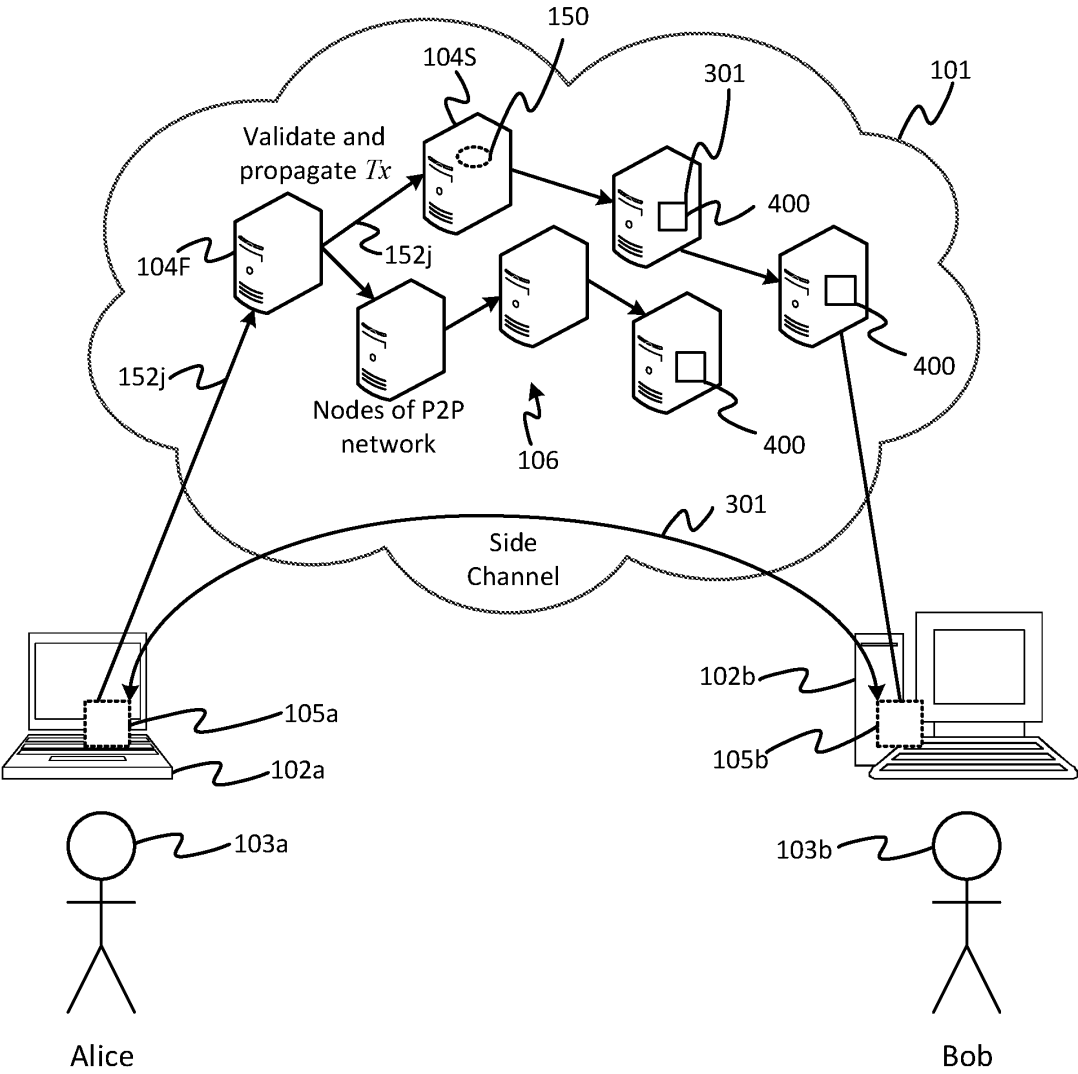


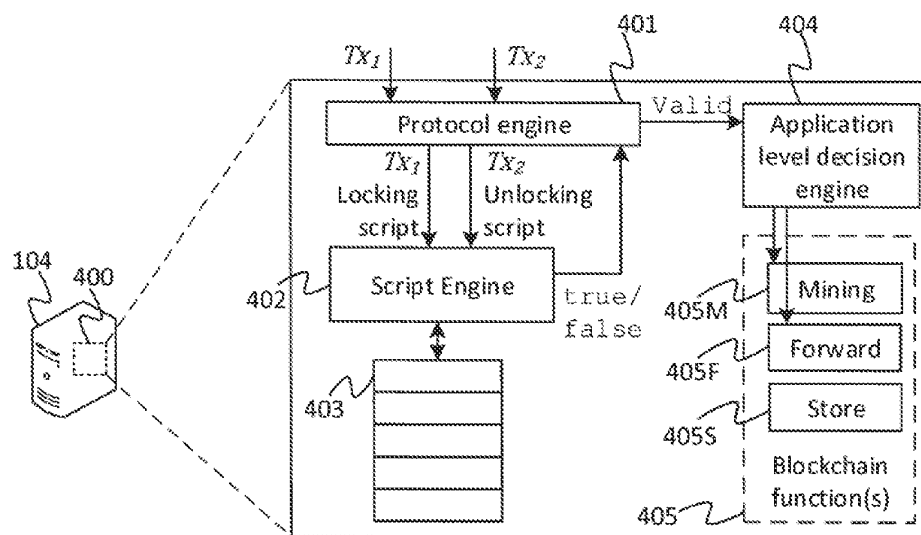
Figure 4

Figure 5

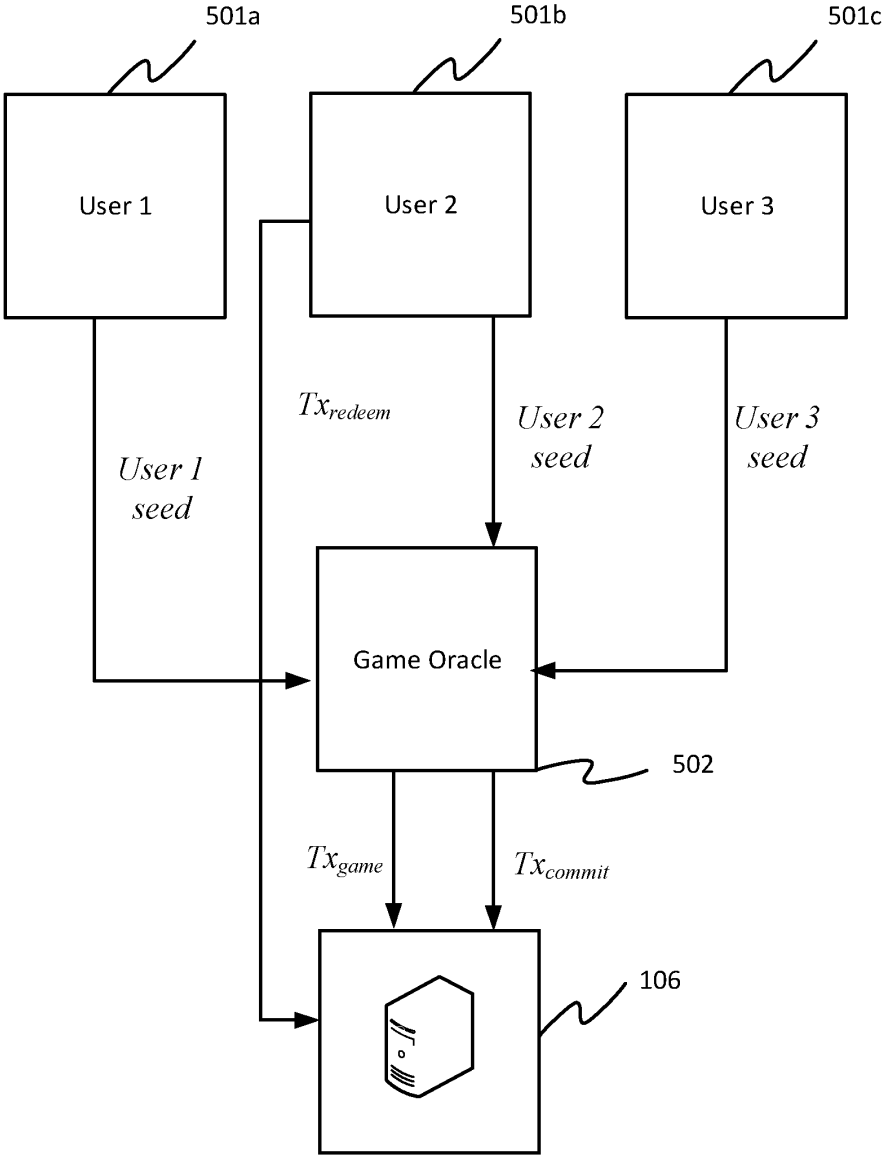


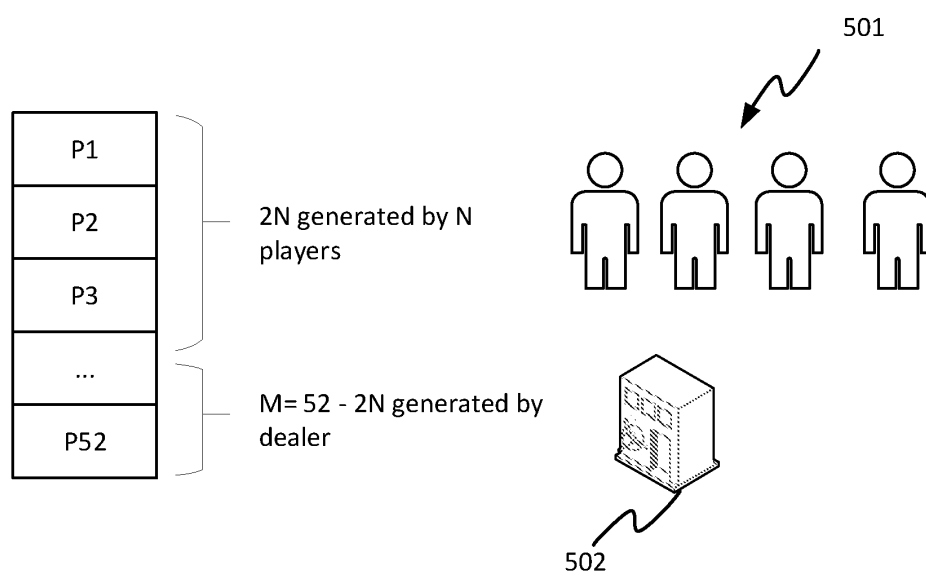
Figure 6

Figure 7

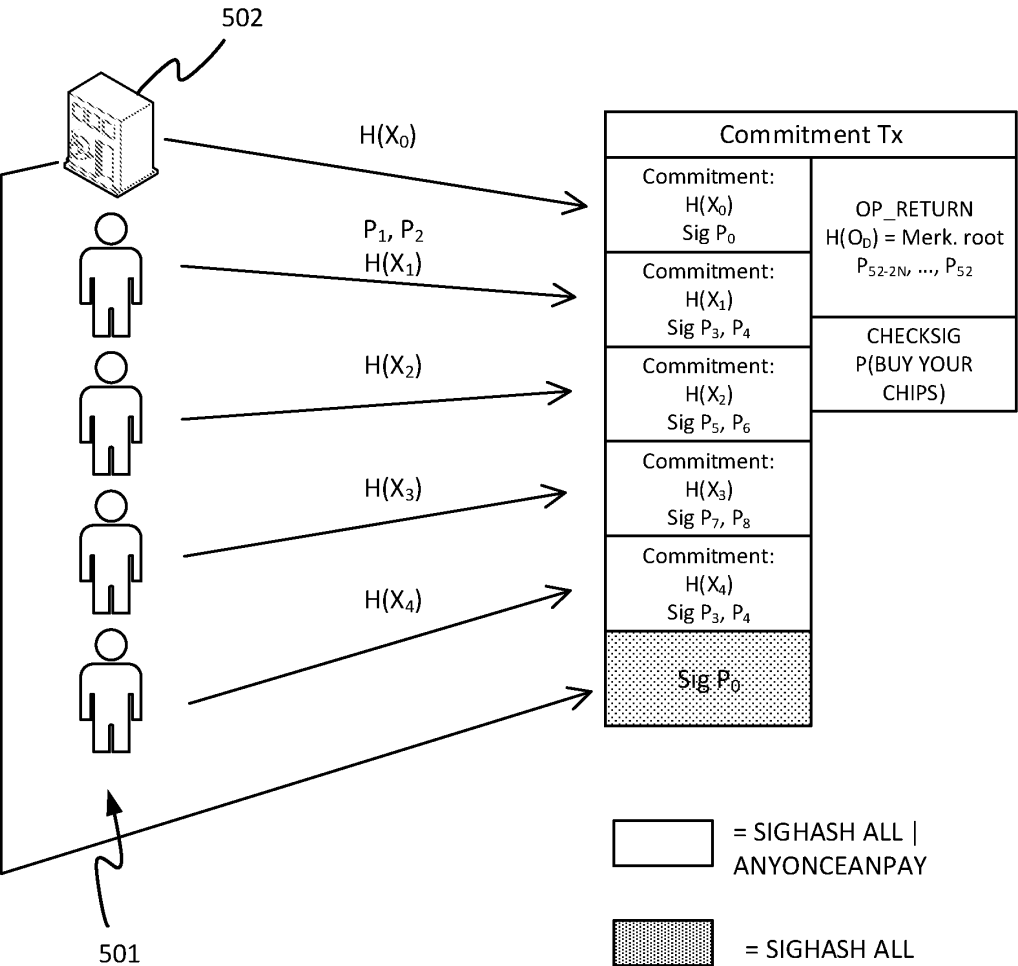


Figure 8

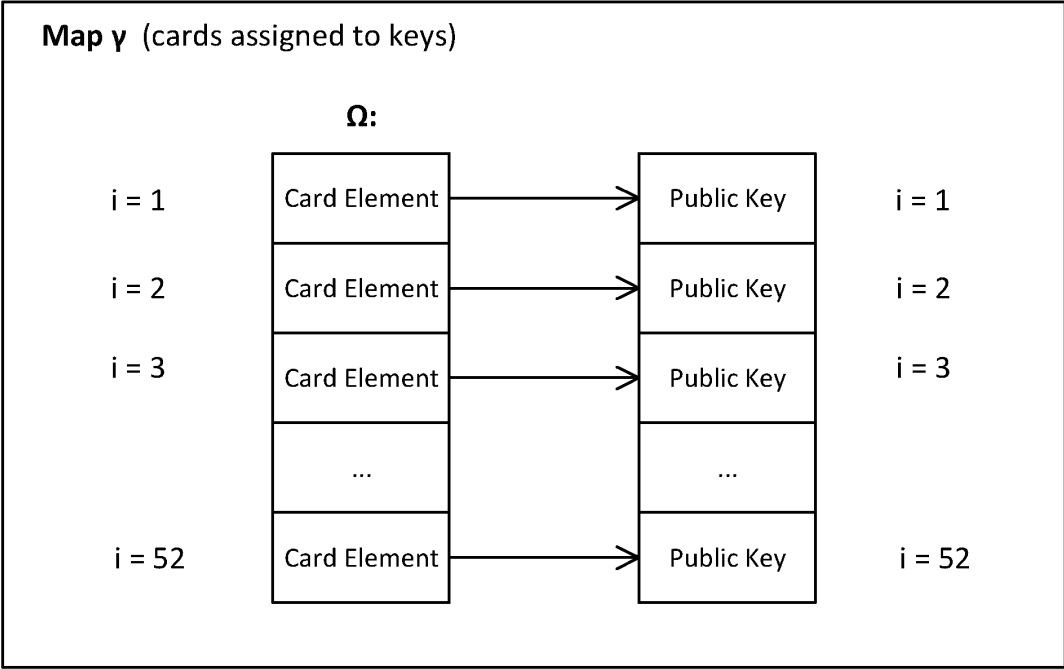


Figure 9

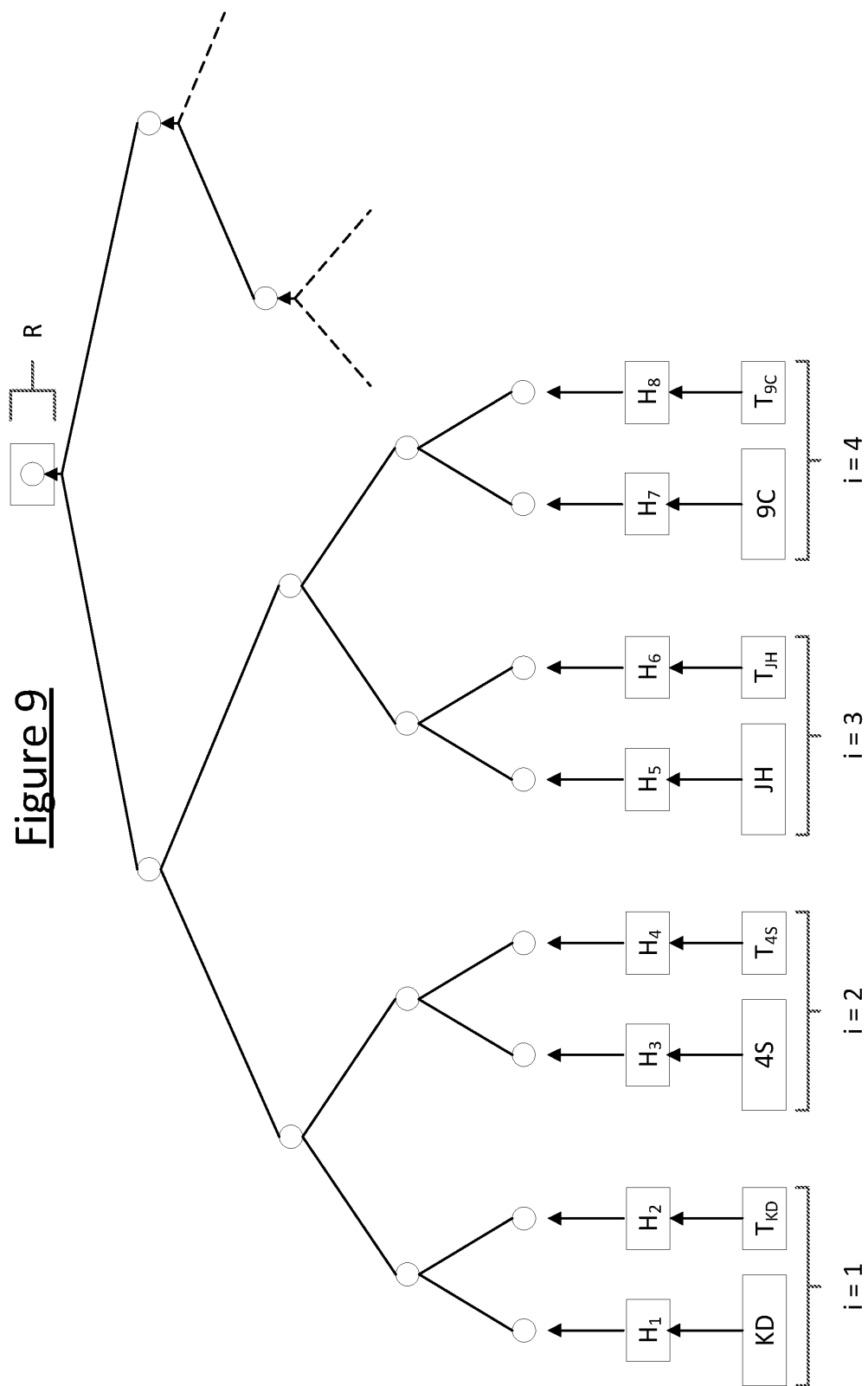


Figure 10

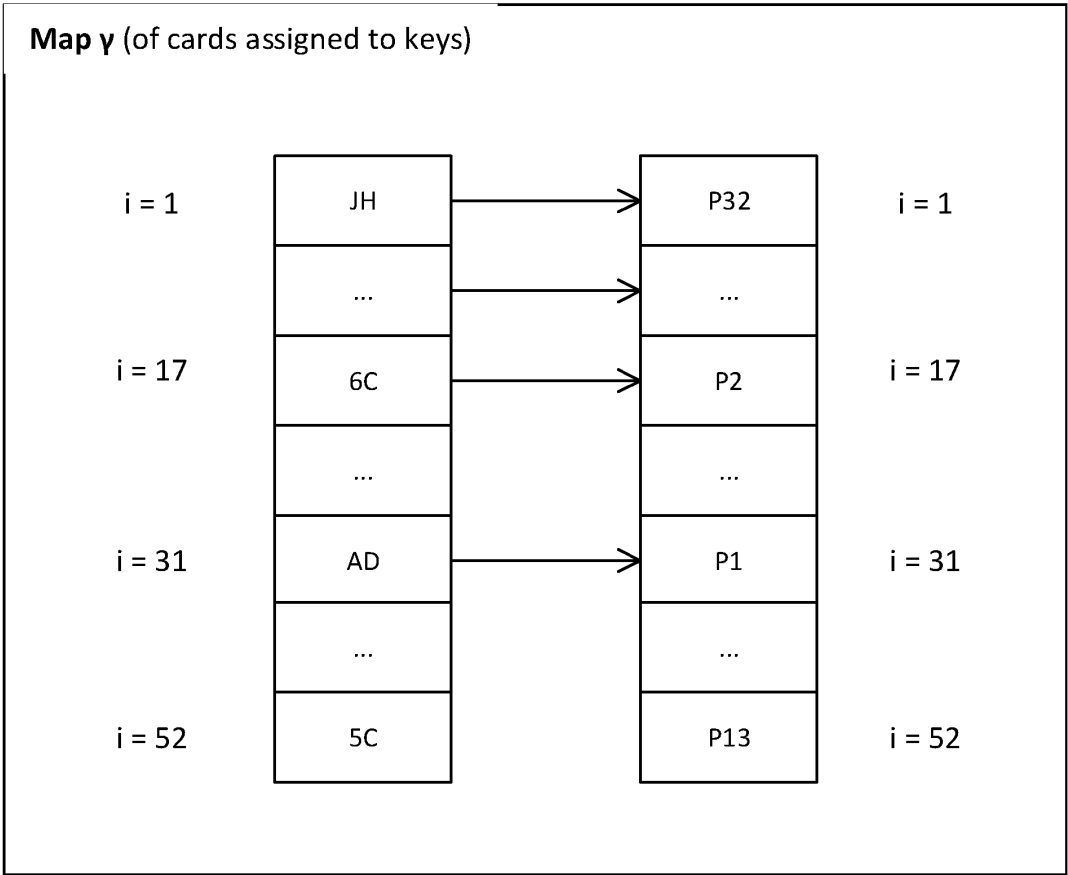


Figure 11

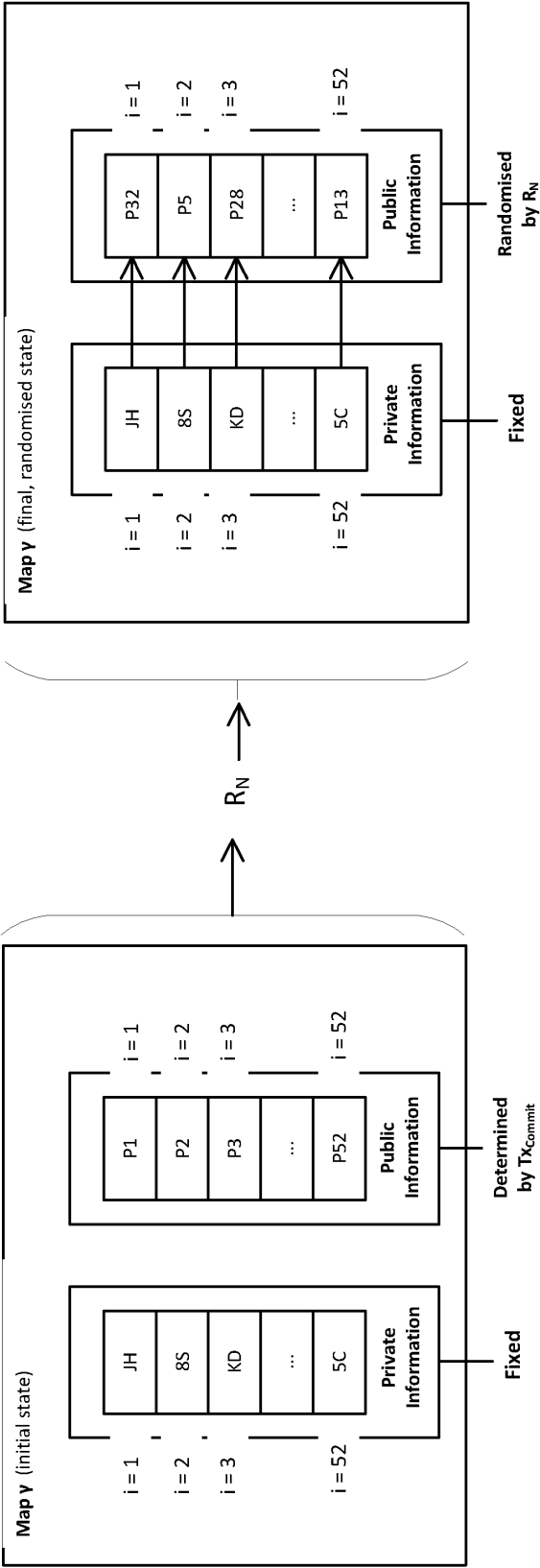
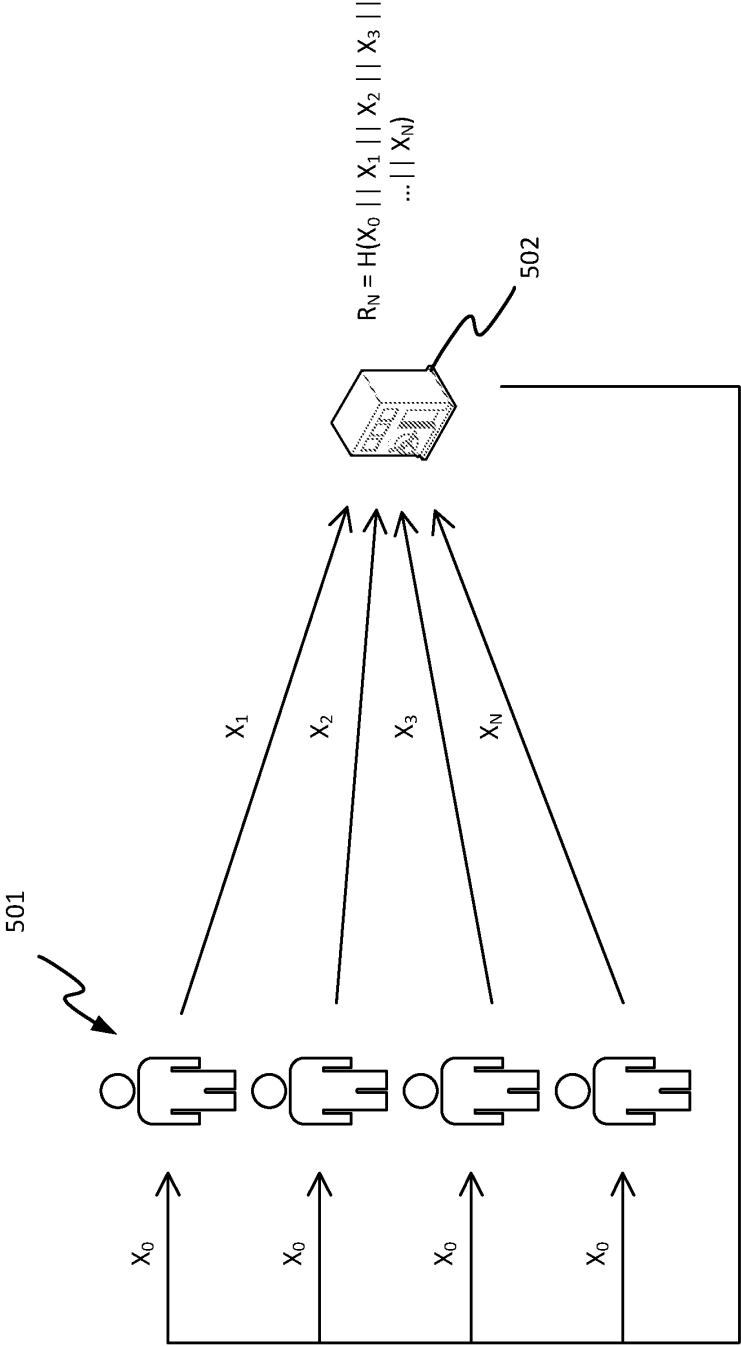


Figure 12



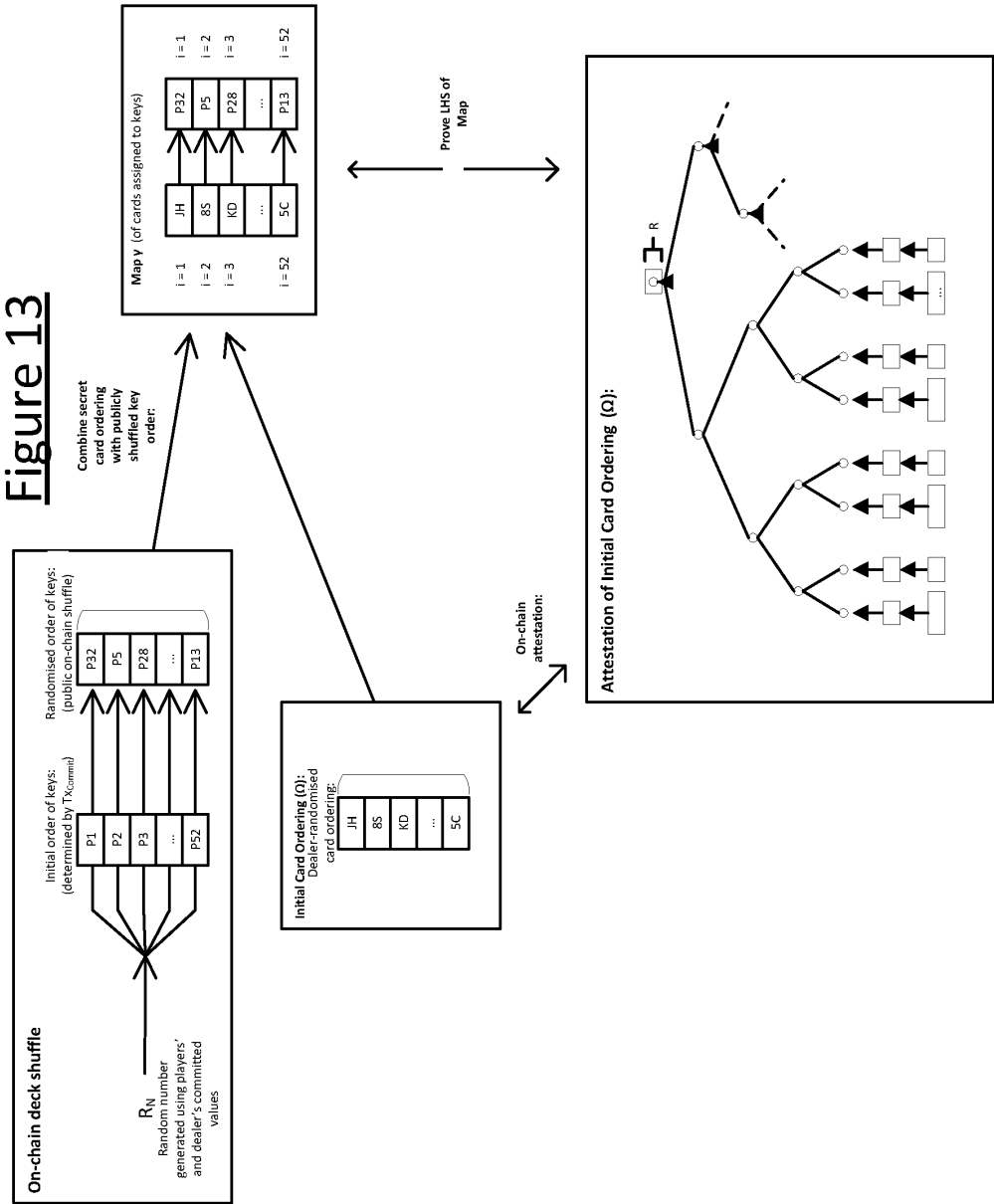


Figure 14

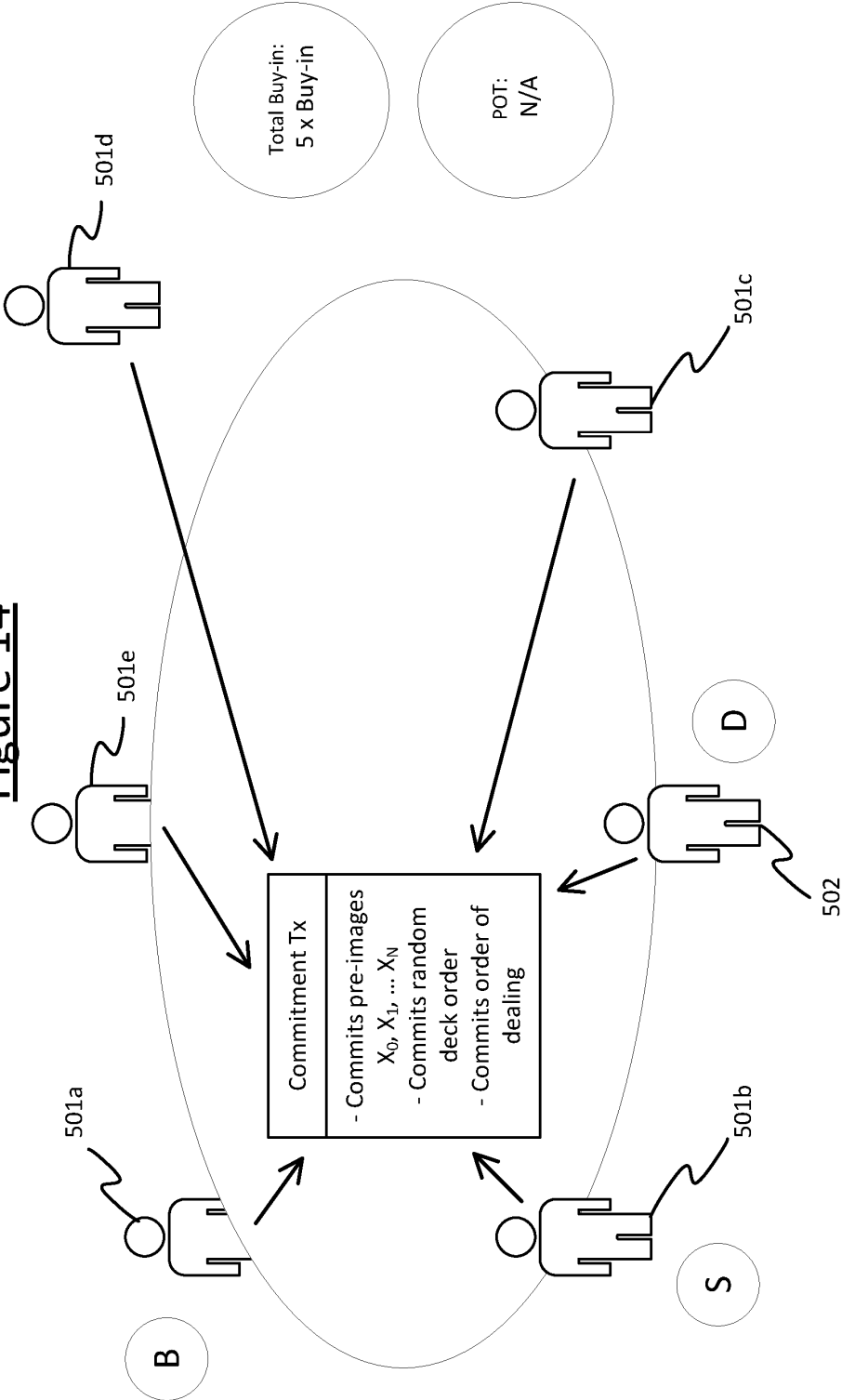


Figure 15

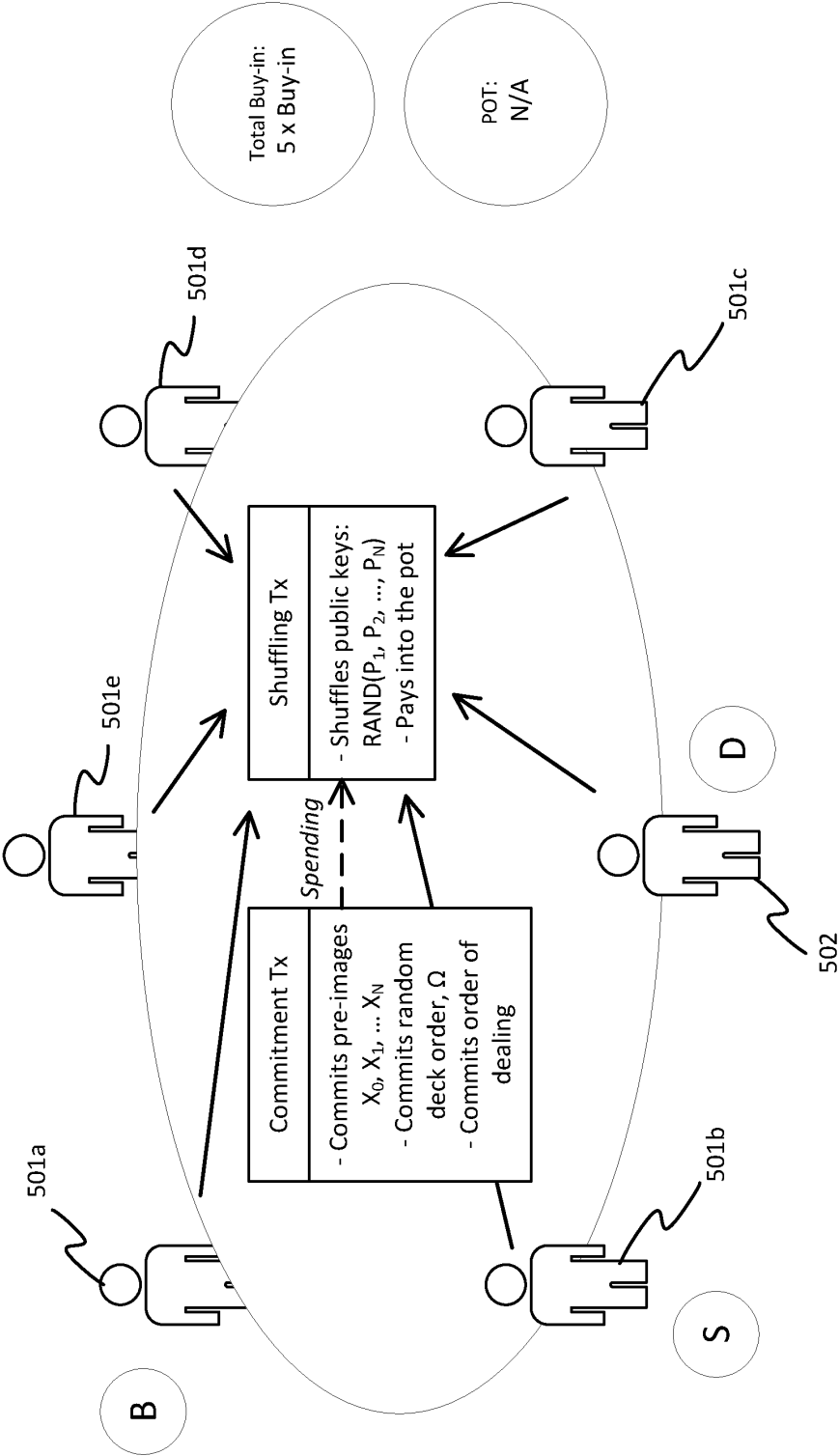


Figure 16

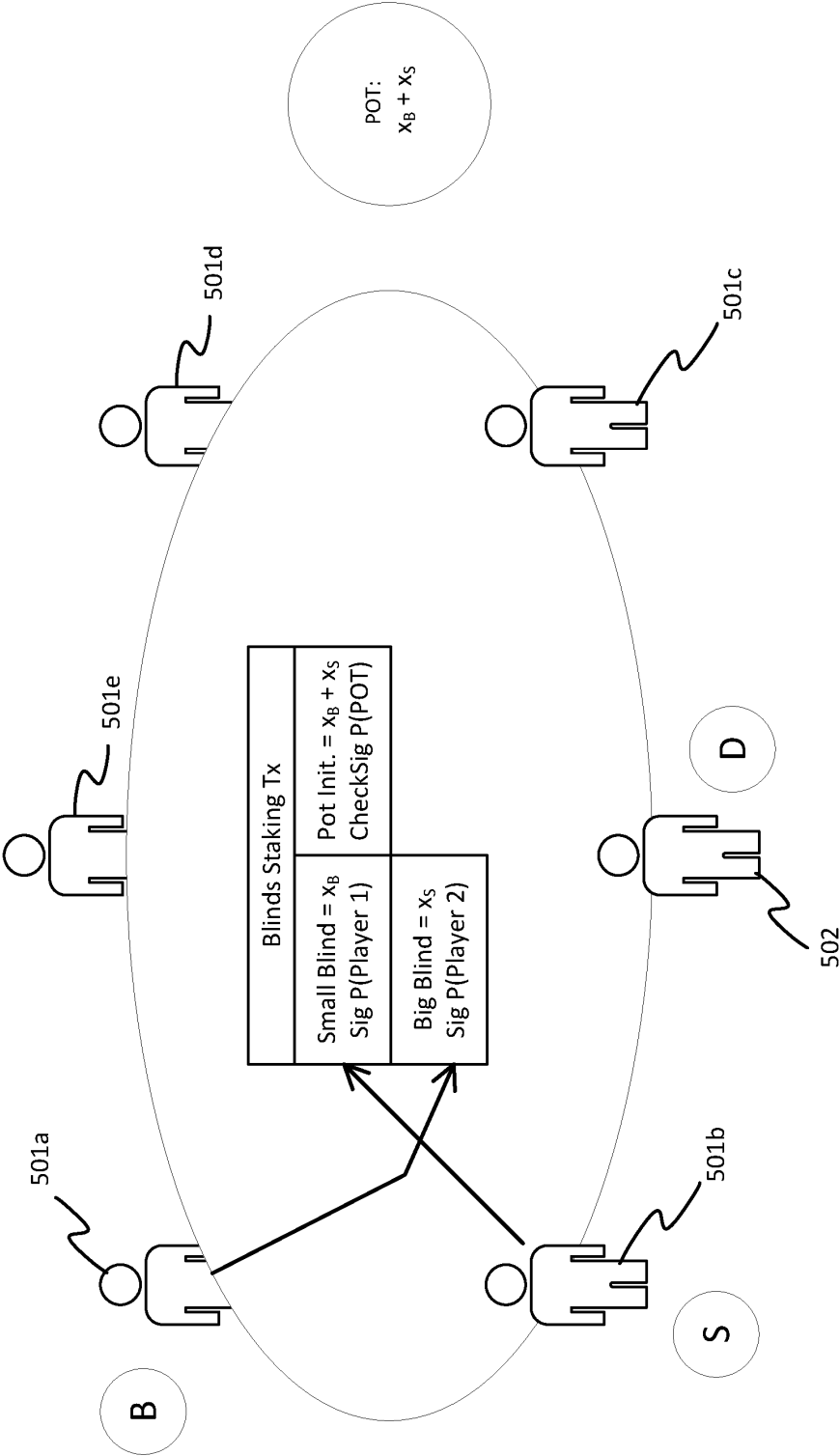


Figure 17

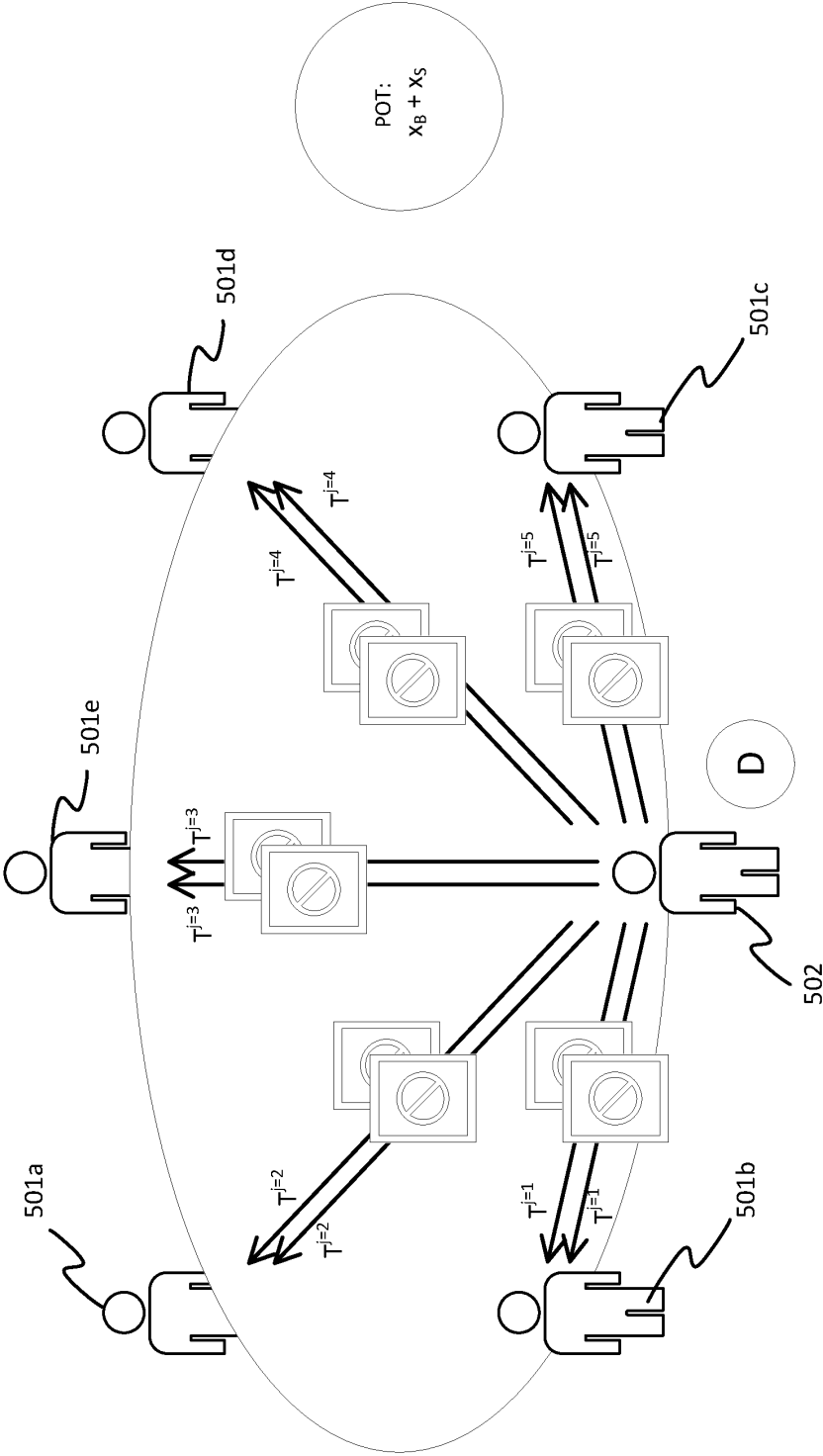


Figure 18

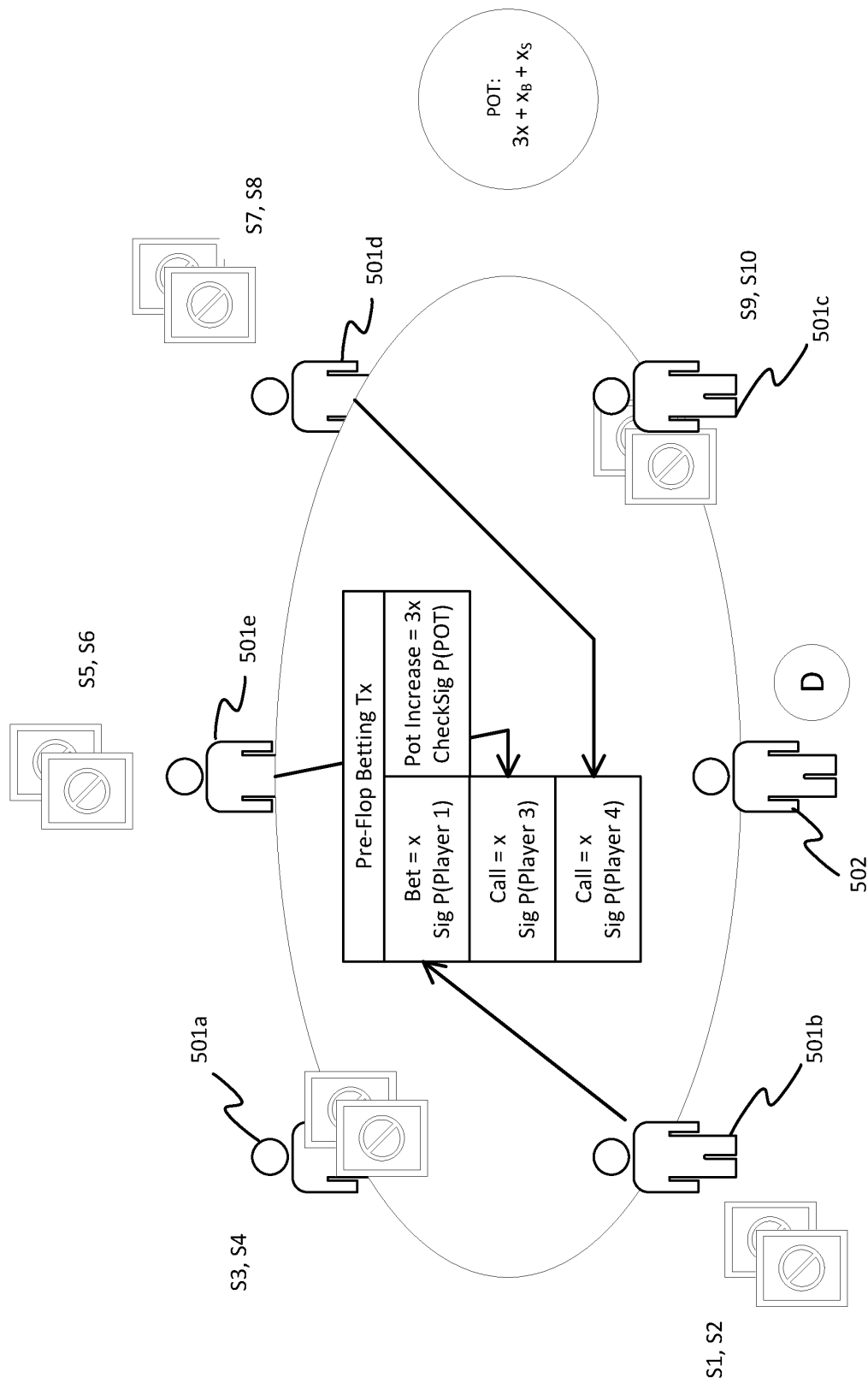


Figure 19

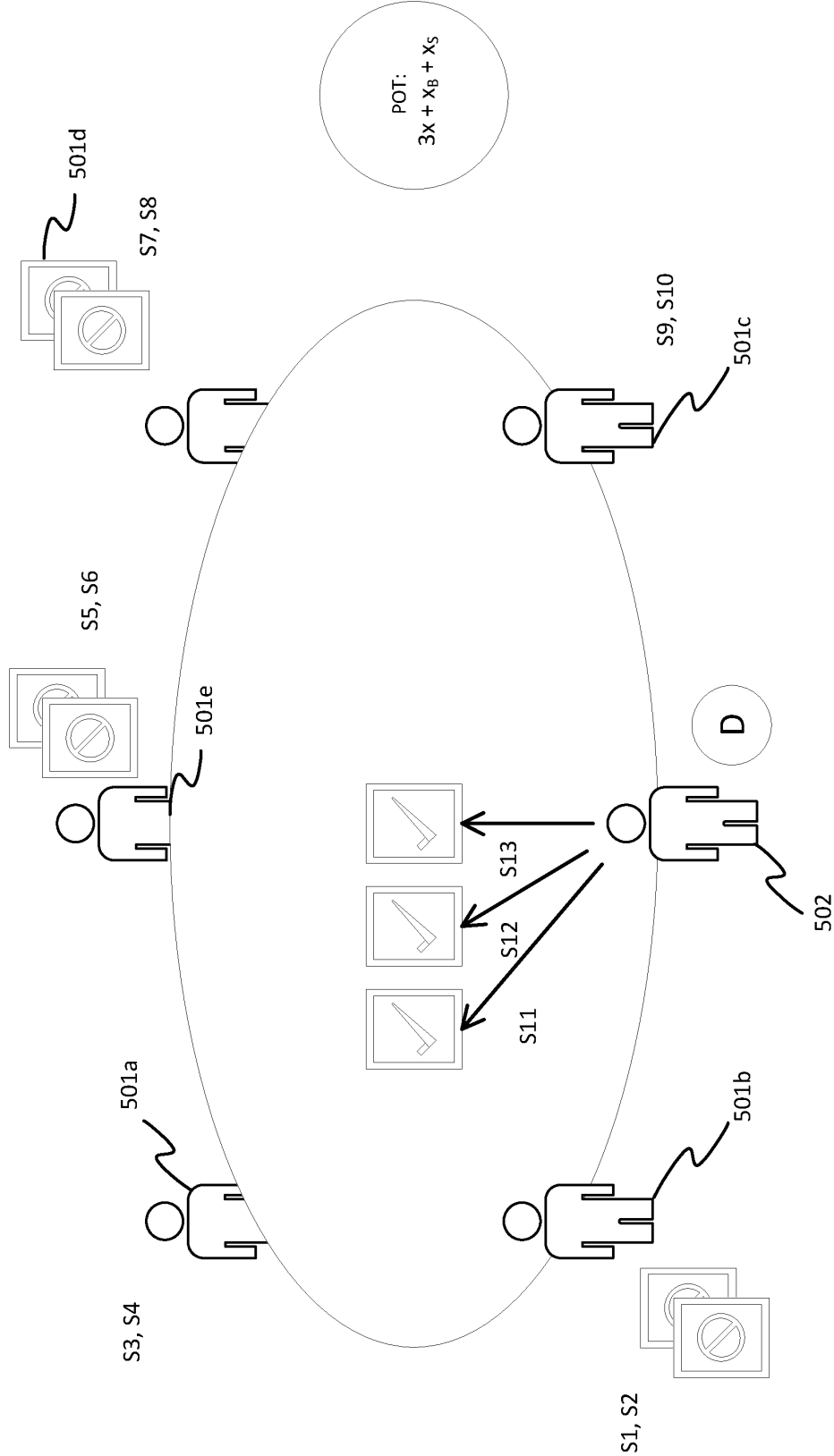


Figure 20

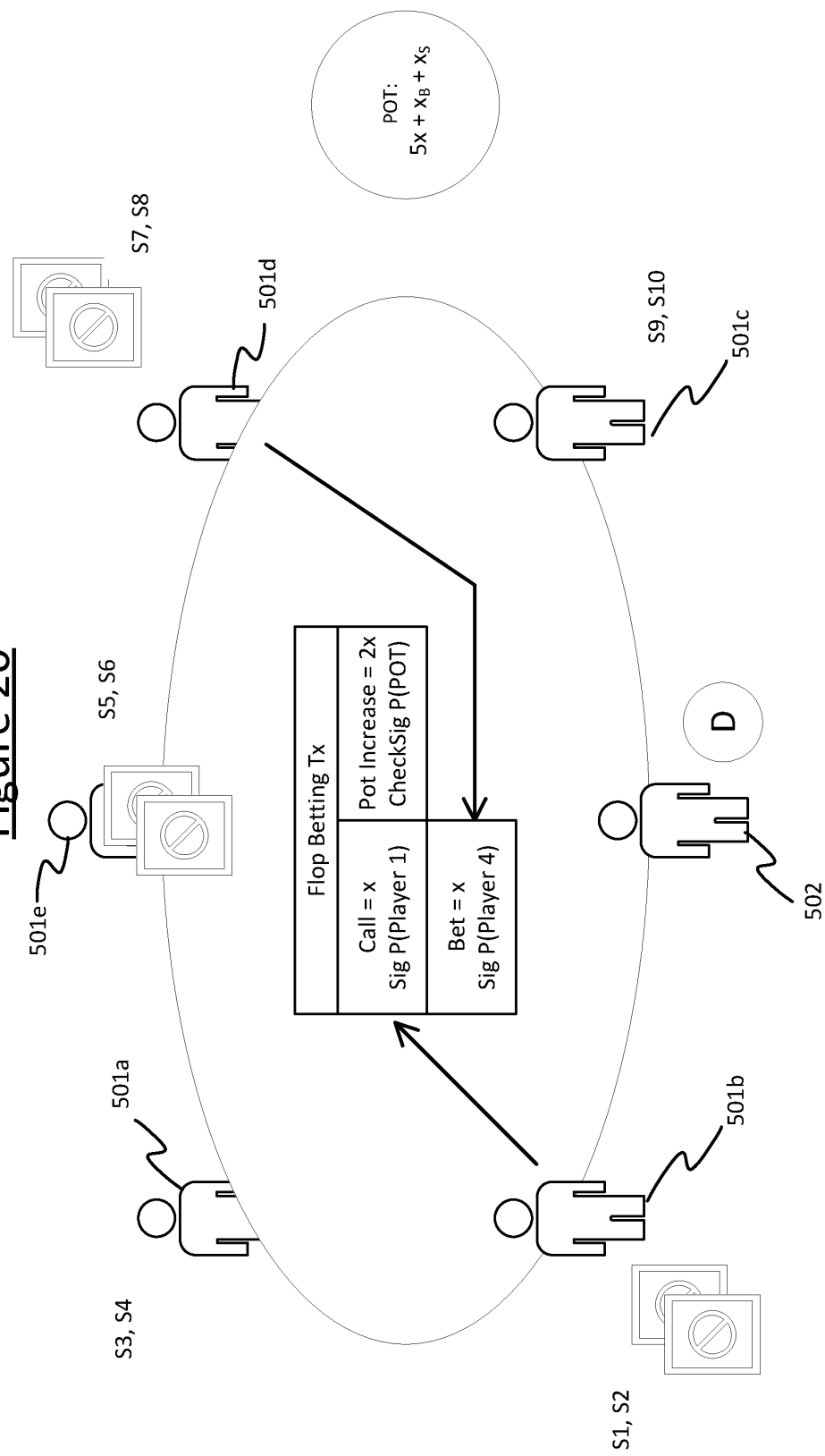


Figure 21

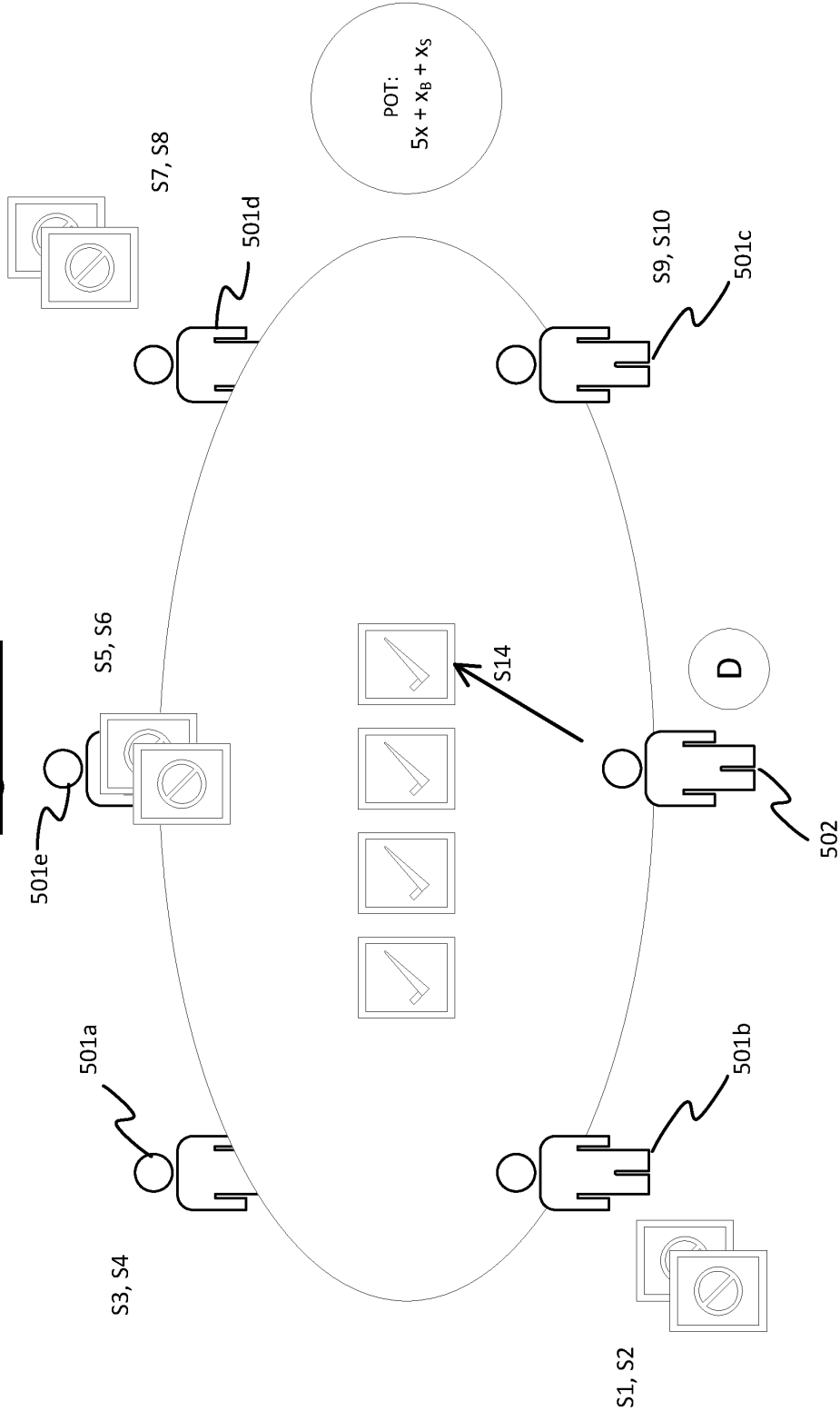


Figure 22

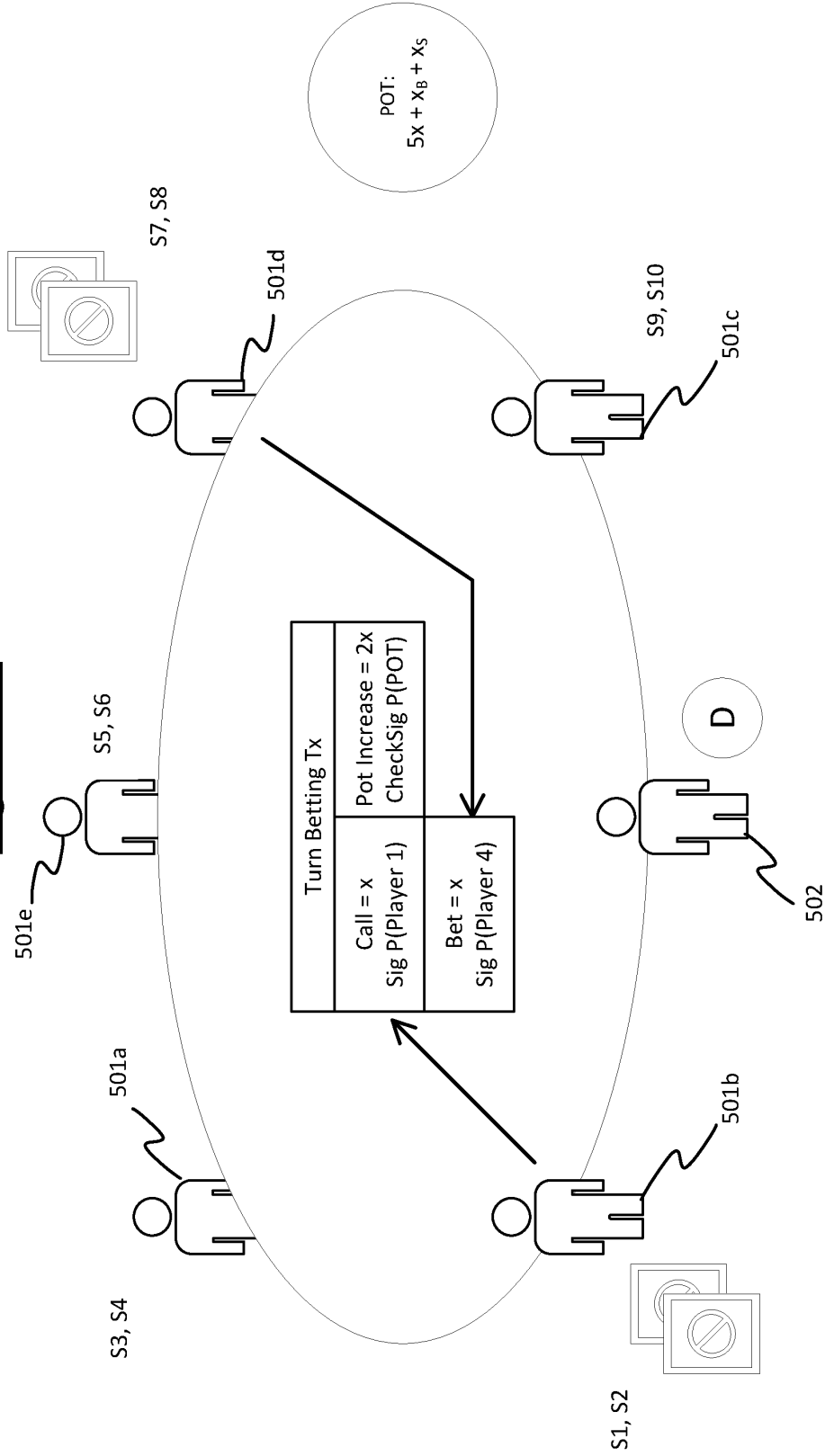


Figure 23

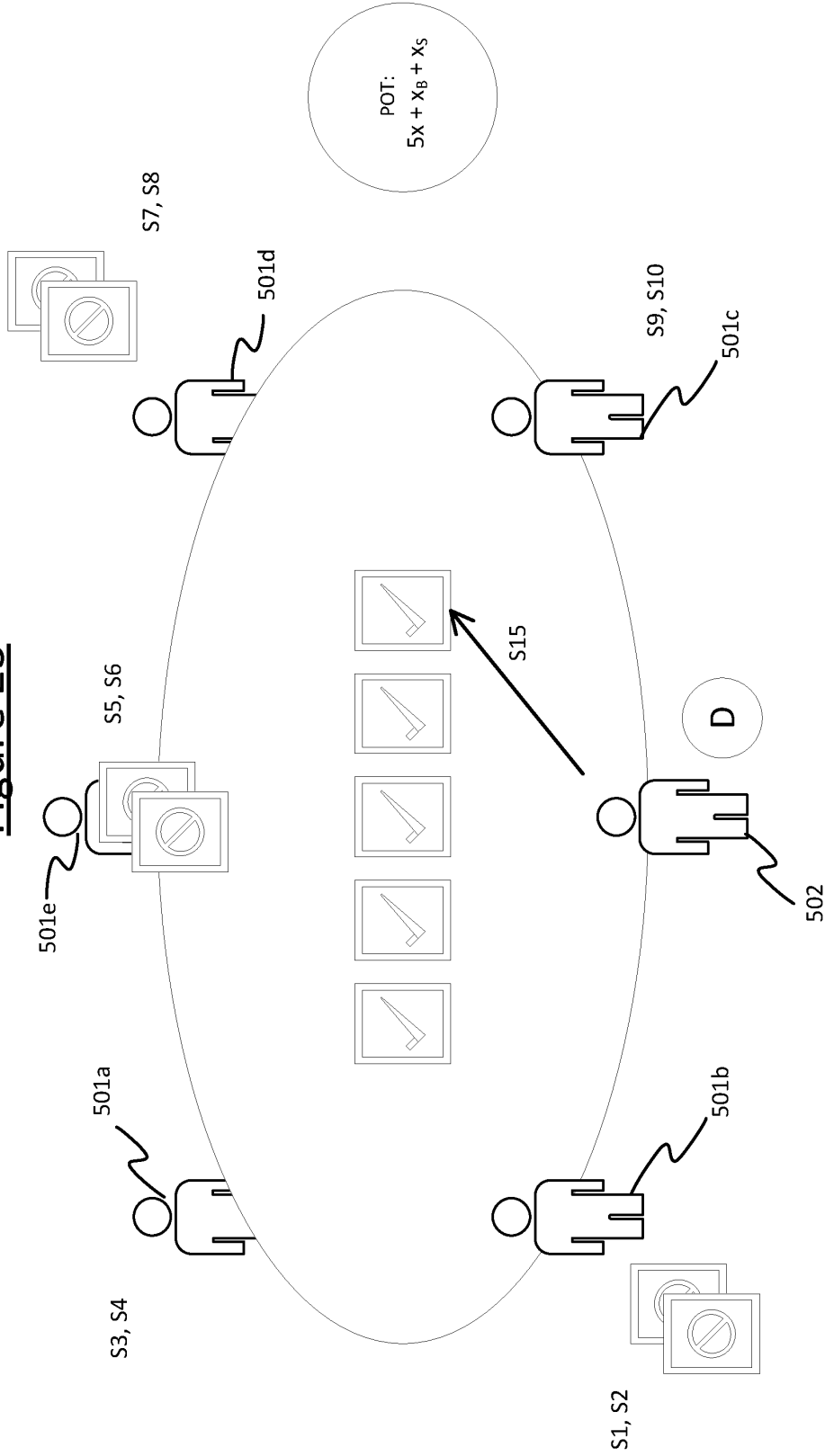


Figure 24

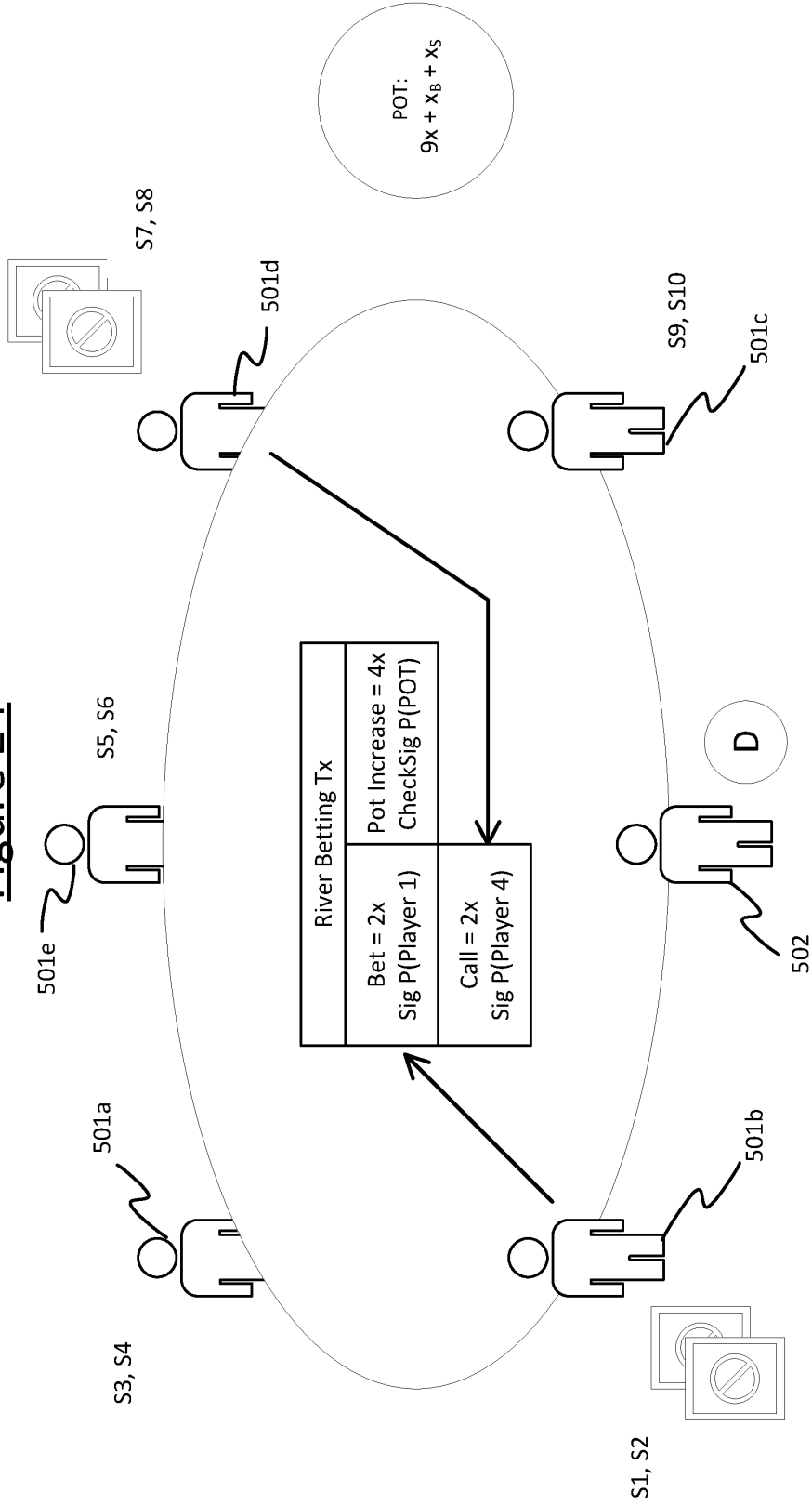


Figure 25

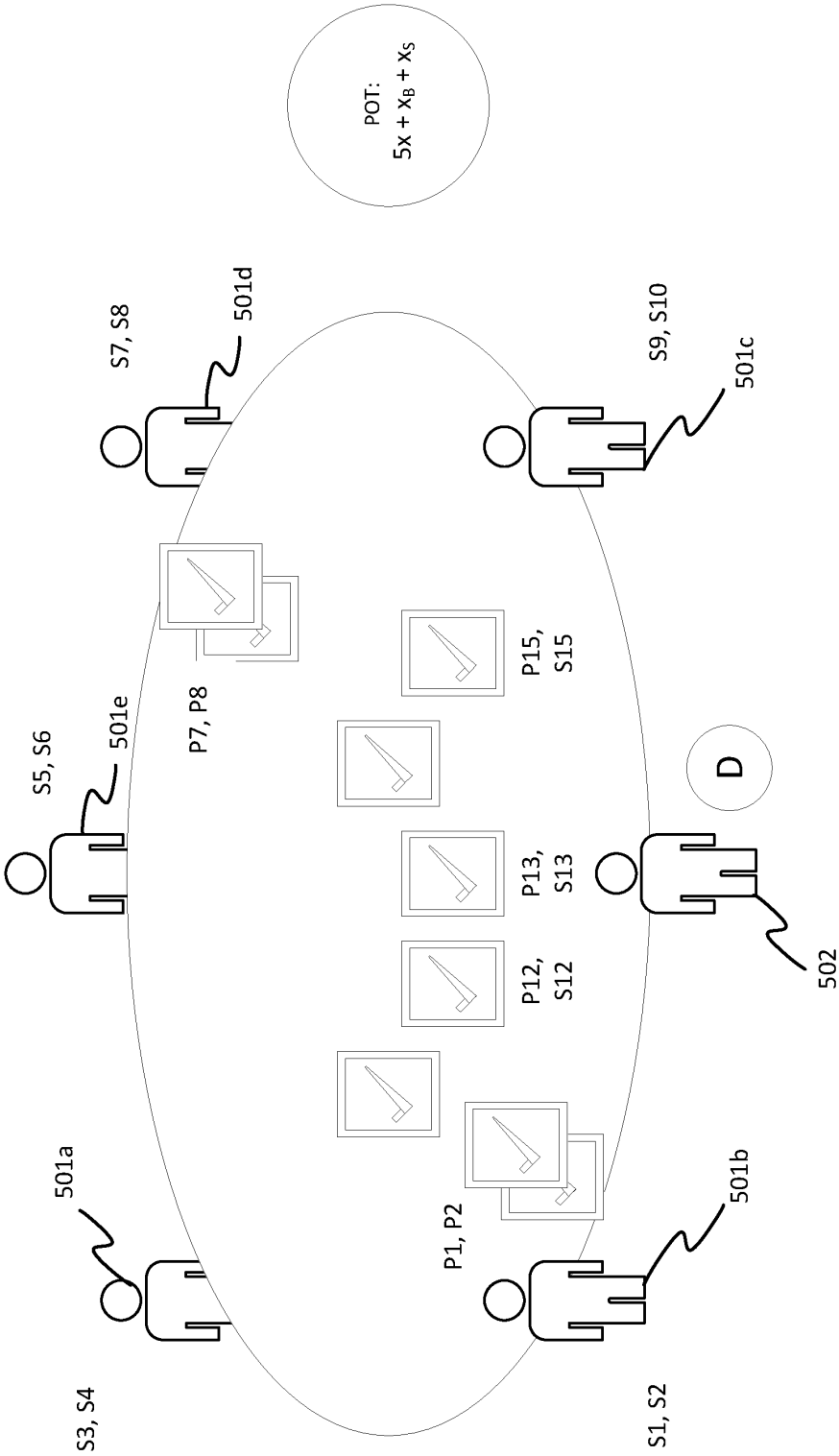


Figure 26

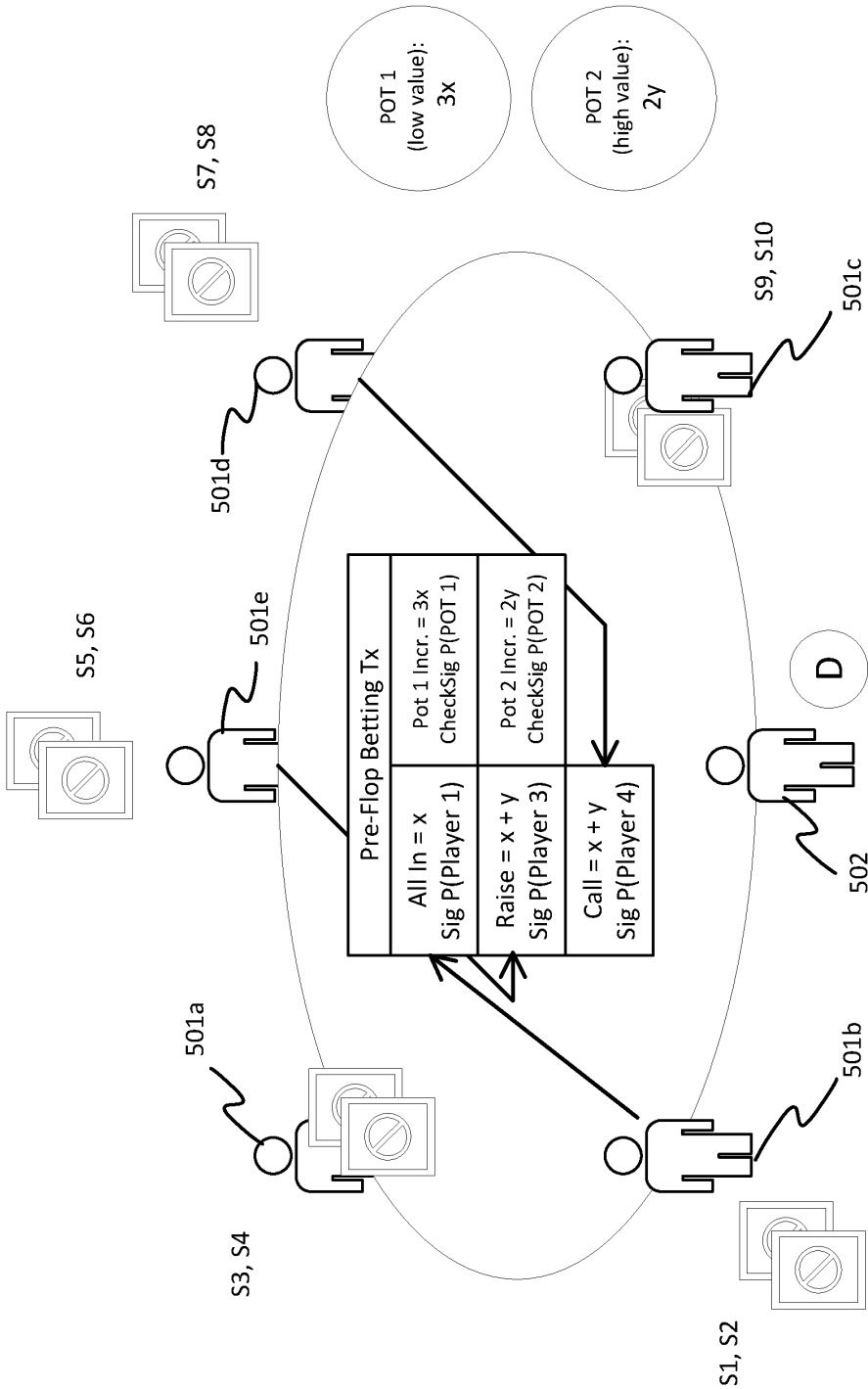


Figure 27

Table 1: Script to generate random number $\langle R_N \rangle$		
Stack	Script	Description
empty	$\langle s'_1 \rangle \langle s'_2 \rangle \dots \langle s'_N \rangle$ OP_ADD OP_ADD ... OP_ADD OP_HASH256 $\langle N \rangle$	
$\langle s'_1 \rangle \langle s'_2 \rangle \dots \langle s'_N \rangle$	OP_ADD OP_ADD ... OP_ADD OP_HASH256 $\langle N \rangle$ OP_MOD	Constants are added to the stack
$\langle s'_1 \rangle \langle s'_2 \rangle \dots \langle s'_{N-1} + s'_N \rangle$	OP_ADD ... OP_ADD OP_HASH256 $\langle N \rangle$ OP_MOD	Top two values on stack added
$\langle s'_1 \rangle \langle s'_2 \rangle \dots \langle s'_{N-2} + s'_{N-1} + s'_N \rangle$	... OP_ADD OP_HASH256 $\langle N \rangle$ OP_MOD	Top two values on stack added
$\langle \sum_i s'_i \rangle$	OP_HASH256 $\langle N \rangle$ OP_MOD	Repeated addition of top two values
$\langle H(\sum_i s'_i) \rangle$	$\langle N \rangle$ OP_MOD	Top value hashed twice with SHA-256
$\langle H(\sum_i s'_i) \rangle \langle N \rangle$	OP_MOD	Constants are added to the stack
$\langle R_N \rangle$	empty	Remainder of 2^{256} value divided by 1^{st} value computed; a random number

Figure 28

Table 2: Script to spend funds encumbered by the Oracle transaction		
AltStack	Stack	Script
empty	empty	$\langle \text{Sig } P_w \rangle$ OP_TOALTSTACK $\langle P_1 \rangle$ $\langle P_2 \rangle$... $\langle P_N \rangle$ OP_ROLL OP_TOALTSTACK OP_DROP ... OP_DROP OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
	$\langle \text{Sig } P_w \rangle$	OP_TOALTSTACK $\langle P_1 \rangle$ $\langle P_2 \rangle$... $\langle P_N \rangle$ OP_ROLL OP_TOALTSTACK OP_DROP ... OP_DROP OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
$\langle \text{Sig } P_w \rangle$	$\langle P_1 \rangle$ $\langle P_2 \rangle$... $\langle P_N \rangle$ $\langle R_w \rangle$	OP_ROLL OP_TOALTSTACK OP_DROP ... OP_DROP OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
$\langle \text{Sig } P_w \rangle$	$\langle P_1 \rangle$... $\langle P_N \rangle$ $\langle P_w \rangle$	OP_TOALTSTACK OP_DROP ... OP_DROP OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
$\langle \text{Sig } P_w \rangle$ $\langle P_w \rangle$	$\langle P_1 \rangle$... $\langle P_N \rangle$	OP_DROP ... OP_DROP OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
$\langle \text{Sig } P_w \rangle$ $\langle P_w \rangle$	empty	OP_FROMALTSTACK OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
$\langle \text{Sig } P_w \rangle$	$\langle P_w \rangle$	OP_FROMALTSTACK OP_SWAP OP_CHECKSIG
empty	$\langle P_w \rangle$ $\langle \text{Sig } P_w \rangle$	OP_SWAP
	$\langle \text{Sig } P_w \rangle$ $\langle P_w \rangle$	OP_CHECKSIG
	TRUE	empty
		Constants are added to the stack Winning signature {WSG} moved to altstack Winning public key {WPK} moved to top of stack WPK moved to altstack Remaining $N-1$ public keys dropped from stack WPK moved back to stack WSG moved back to stack Winning signature checked