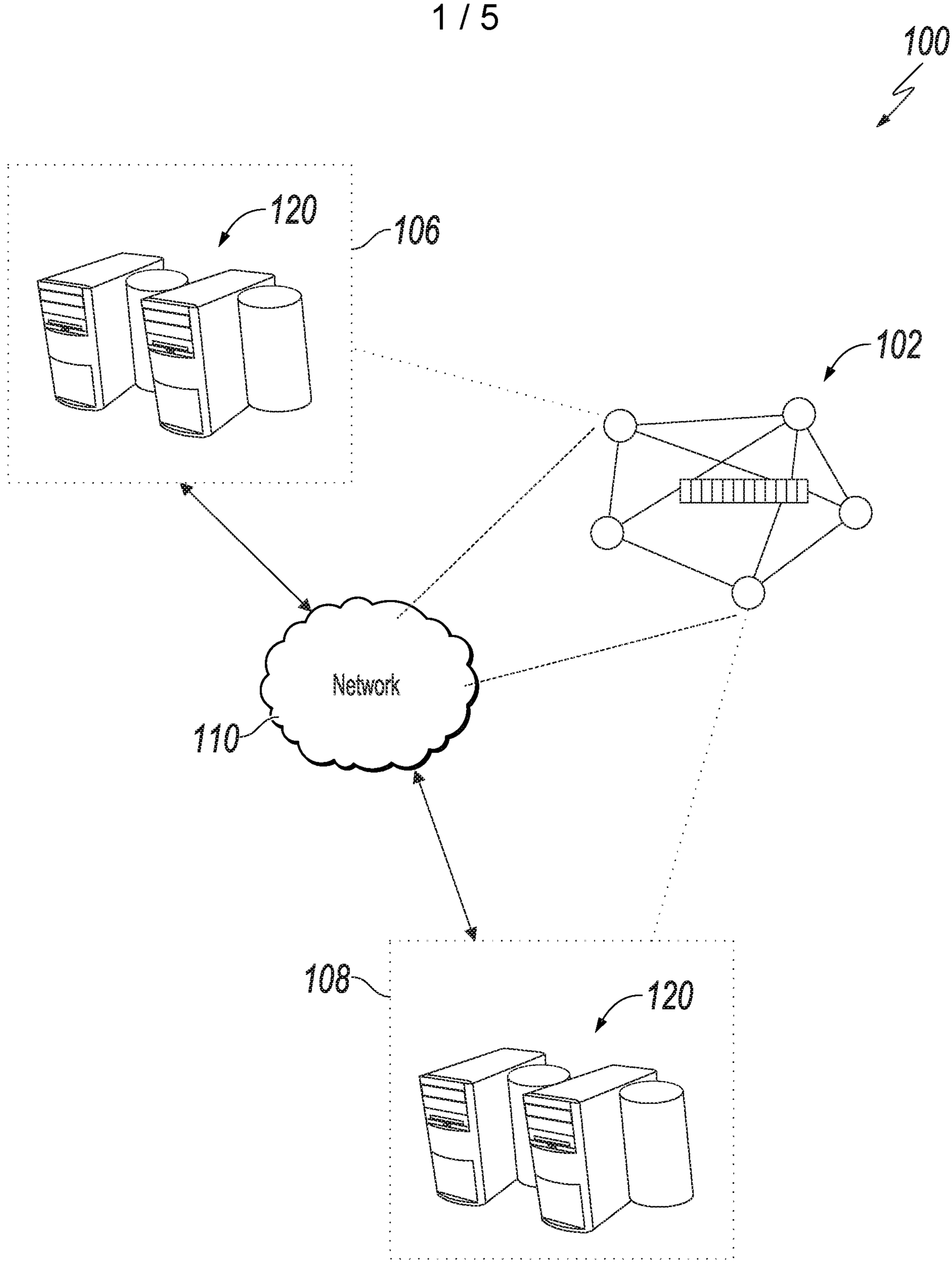

FIG. 1

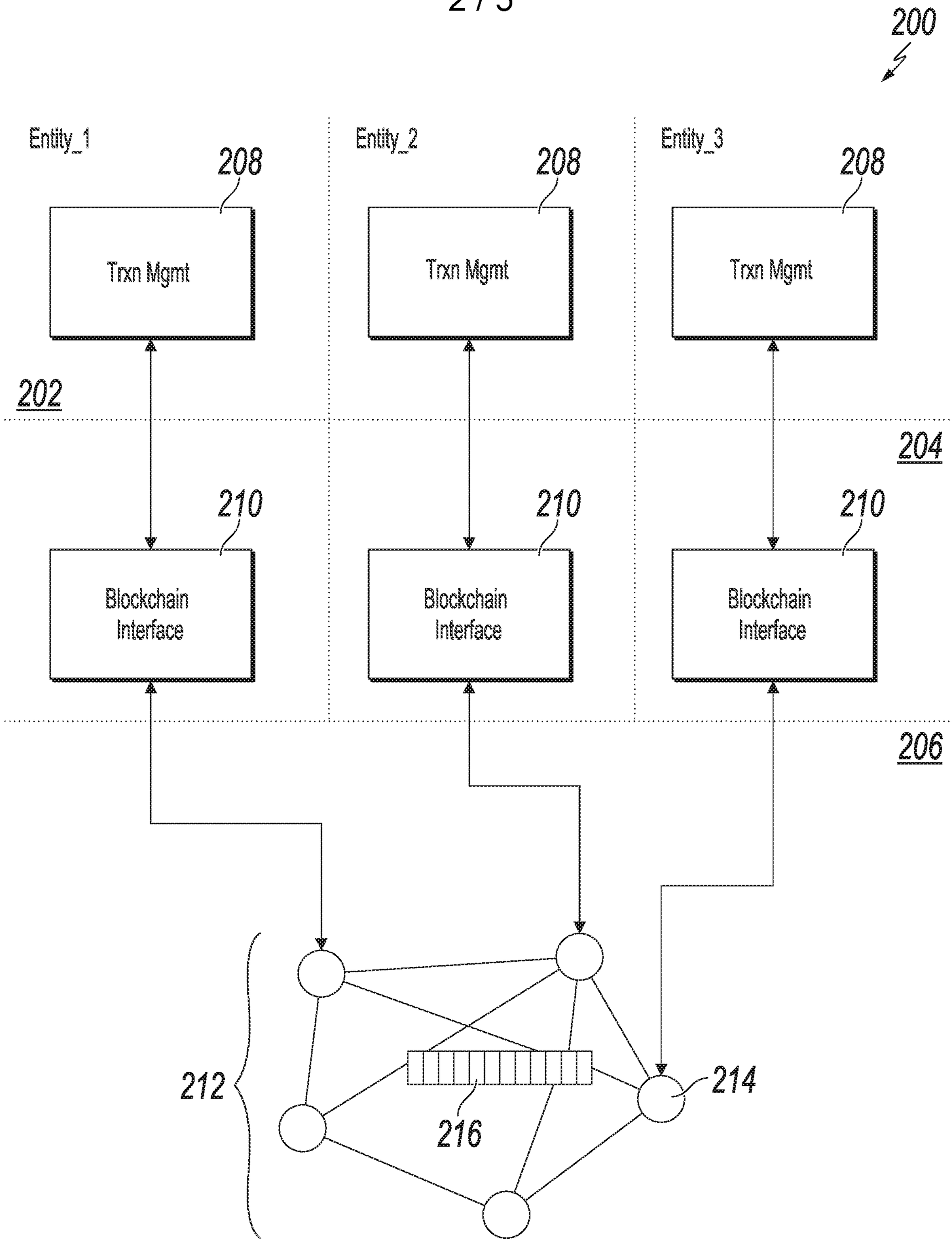
200
Entity_1
Entity_2
Entity_3
208
Trxn Mgmt
202
204
210
Blockchain Interface
206
212
214
216

FIG. 2

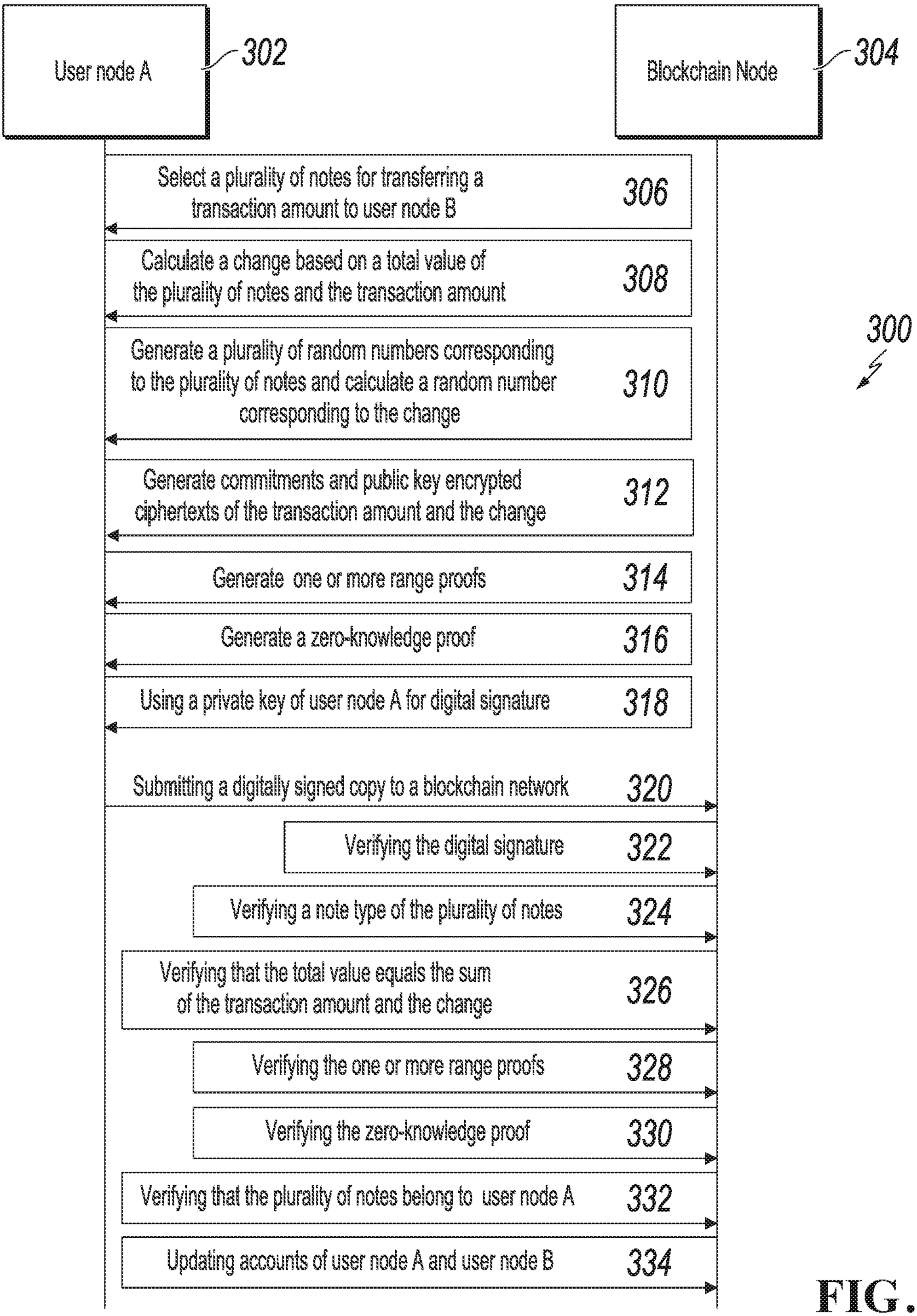
User node A
302
Blockchain Node
304
Select a plurality of notes for transferring a transaction amount to user node B
306
Calculate a change based on a total value of the plurality of notes and the transaction amount
308
Generate a plurality of random numbers corresponding to the plurality of notes and calculate a random number corresponding to the change
310
Generate commitments and public key encrypted ciphertexts of the transaction amount and the change
312
Generate one or more range proofs
314
Generate a zero-knowledge proof
316
Using a private key of user node A for digital signature
318
Submitting a digitally signed copy to a blockchain network
320
Verifying the digital signature
322
Verifying a note type of the plurality of notes
324
Verifying that the total value equals the sum of the transaction amount and the change
326
Verifying the one or more range proofs
328
Verifying the zero-knowledge proof
330
Verifying that the plurality of notes belong to user node A
332
Updating accounts of user node A and user node B
334
300

FIG. 3

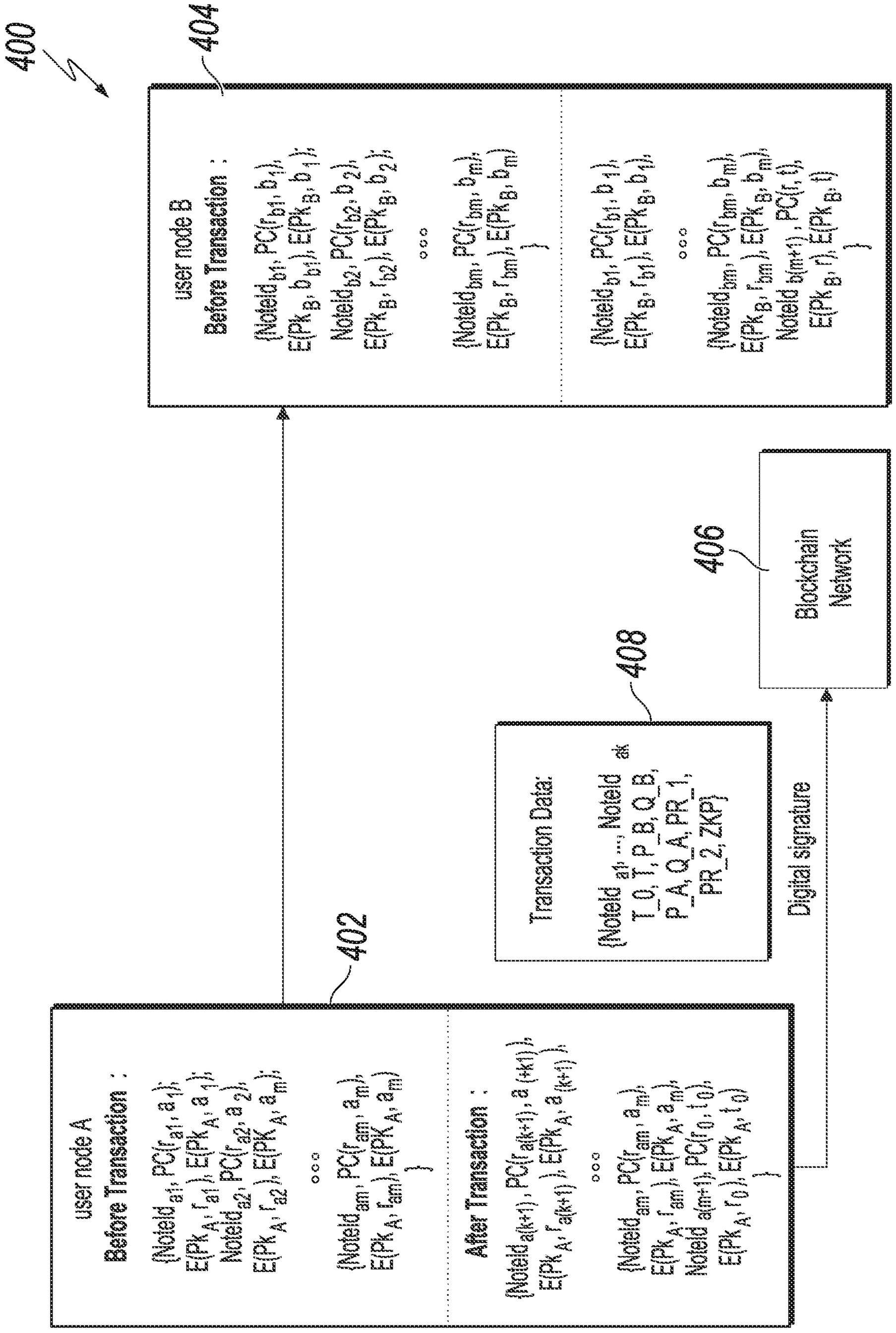
400
404
user node B
Before Transaction :
{NoteId$_{b1}$, PC(r_{b1}, b_{b1}),
E(Pk_B, b_{b1})}, E(Pk_B, b_1)};
NoteId$_{b2}$, PC(r_{b2}, b_2),
E(Pk_B, r_{b2}), E(Pk_B, b_2)};
°°°
{NoteId$_{bm}$, PC(r_{bm}, b_m),
E(Pk_B, r_{bm}), E(Pk_B, b_m)
}
{NoteId$_{b1}$, PC(r_{b1}, b_1)},
E(Pk_B, r_{b1}), E(Pk_B, b_1)},
°°°
{NoteId$_{bm}$, PC(r_{bm}, b_m)},
E(Pk_B, r_{bm}), E(Pk_B, b_m)},
NoteId b(m+1), PC(r, t),
E(Pk_B, r), E(Pk_B, t)
}
406
Blockchain Network
408
Transaction Data:
{NoteId a1, ..., NoteId ak
T_0, T_P_B, Q_B,
P_A, Q_A, PR_1,
PR_2, ZKP}
Digital signature
402
user node A
Before Transaction :
{NoteId$_{a1}$, PC(r_{a1}, a_1)},
E(Pk_A, r_{a1}), E(Pk_A, a_1)};
NoteId$_{a2}$, PC(r_{a2}, a_2),
E(Pk_A, r_{a2}), E(PK_A, a_m)};
°°°
{NoteId$_{am}$, PC(r_{am}, a_m),
E(Pk_A, r_{am}), E(PK_A, a_m)
}
After Transaction :
{NoteId$_{a(k+1)}$, PC($r_{a(k+1)}$, a (+k1)),
E(Pk_A, $r_{a(k+1)}$), E(Pk_A, a $_{(k+1)}$)};
°°°
{NoteId$_{am}$, PC(r_{am}, a_m),
E(Pk_A, r_{am}), E(Pk_A, a_m),
NoteId a(m+1), PC(r_0, t_0),
E(Pk_A, r_0), E(Pk_A, t_0)
}

FIG. 4

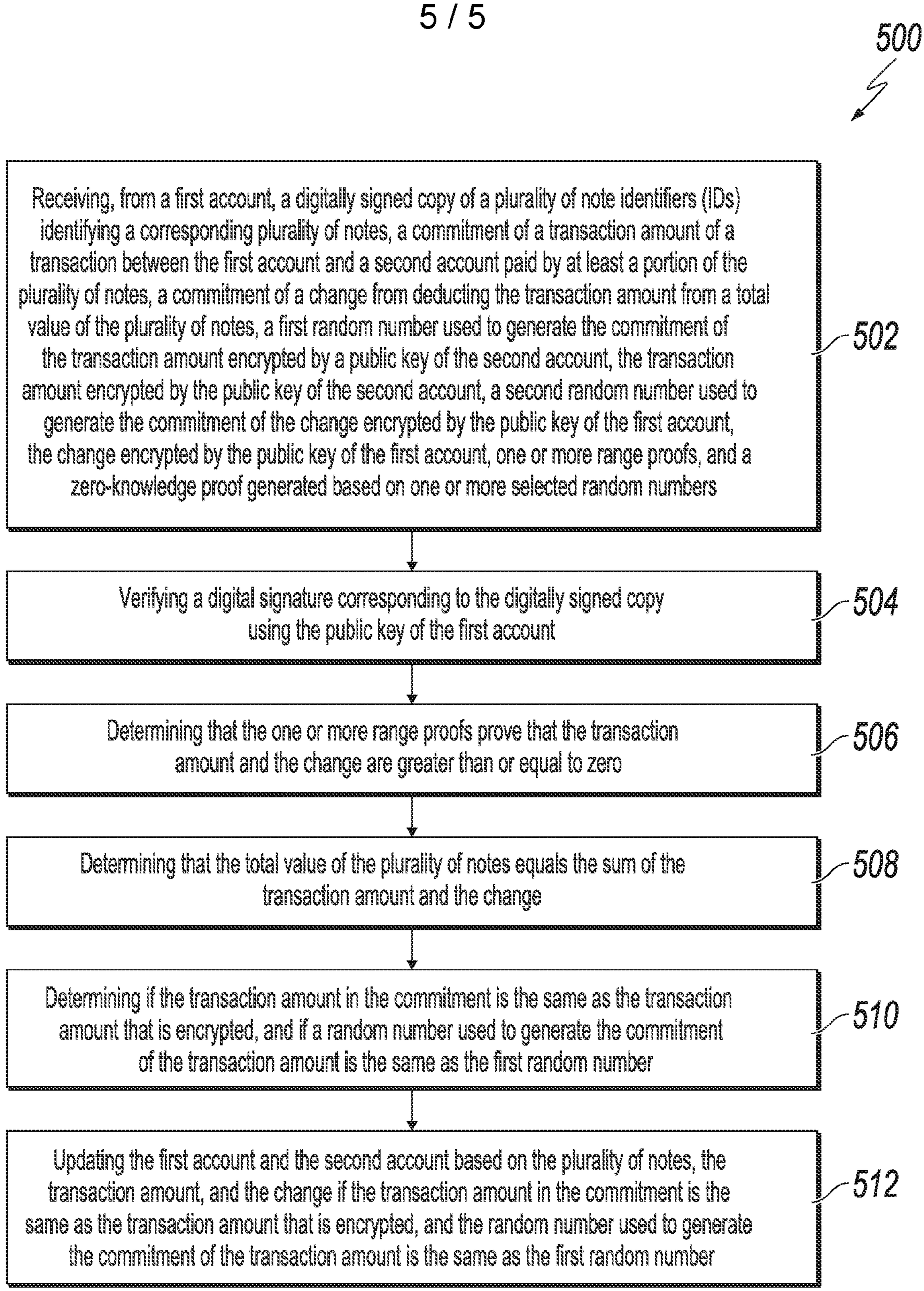
500
Receiving, from a first account, a digitally signed copy of a plurality of note identifiers (IDs) identifying a corresponding plurality of notes, a commitment of a transaction amount of a transaction between the first account and a second account paid by at least a portion of the plurality of notes, a commitment of a change from deducting the transaction amount from a total value of the plurality of notes, a first random number used to generate the commitment of the transaction amount encrypted by a public key of the second account, the transaction amount encrypted by the public key of the second account, a second random number used to generate the commitment of the change encrypted by the public key of the first account, the change encrypted by the public key of the first account, one or more range proofs, and a zero-knowledge proof generated based on one or more selected random numbers
502
Verifying a digital signature corresponding to the digitally signed copy using the public key of the first account
504
Determining that the one or more range proofs prove that the transaction amount and the change are greater than or equal to zero
506
Determining that the total value of the plurality of notes equals the sum of the transaction amount and the change
508
Determining if the transaction amount in the commitment is the same as the transaction amount that is encrypted, and if a random number used to generate the commitment of the transaction amount is the same as the first random number
510
Updating the first account and the second account based on the plurality of notes, the transaction amount, and the change if the transaction amount in the commitment is the same as the transaction amount that is encrypted, and the random number used to generate the commitment of the transaction amount is the same as the first random number
512

FIG. 5