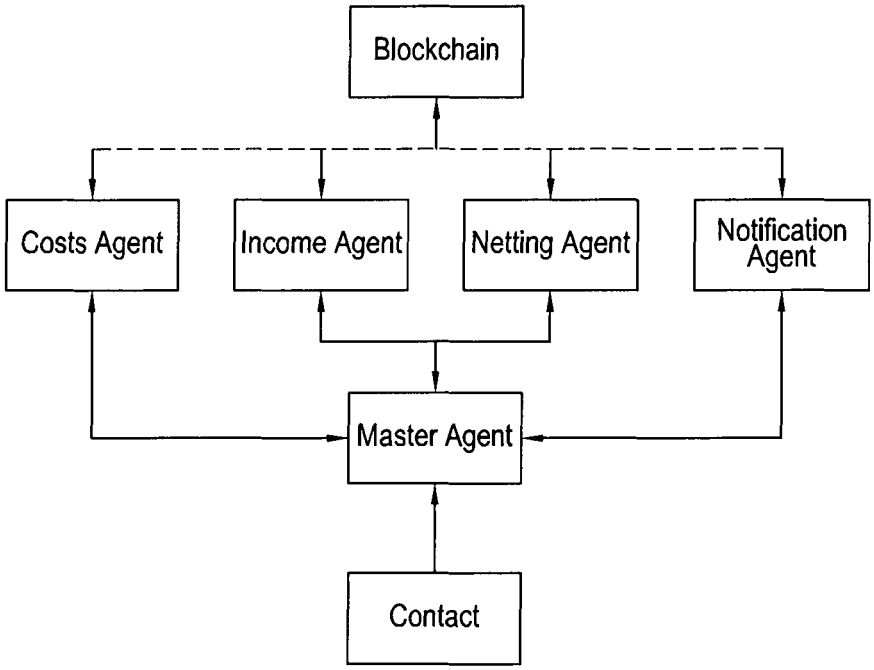

Blockchain
Costs Agent
Income Agent
Netting Agent
Notification Agent
Master Agent
Contact

Fig. 1

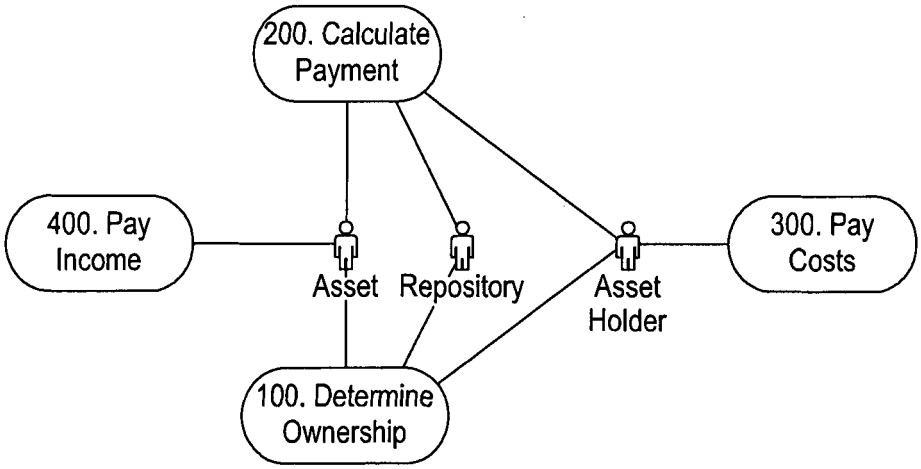

200. Calculate Payment
400. Pay Income
Asset
Repository
Asset Holder
300. Pay Costs
100. Determine Ownership

Fig. 2

For comparison: Generic Contract metadata format				
Field	**Sub-field**	**Bytes**	**Value**	**Comments**
Metadata1	ContractType	4		Coded value indicates type of contract.
	ContractPointer	16		IPv6 address of the actual contract file location
	ContractTypeData1	12		Format depends on value of ContractType. Padded with zeros
Metadata2	ContractHash	20		RIPEMD-160(SHA256(actual contract file addressed by ContractPointer))
	Jurisdiction	2		Legal juristiction covering the metadata
	ContractTypeData2	10		Format depends on value of ContractType. Padded with zeros

Fig. 3

Scenario Definition				
	Create a racehorse asset			
	and then pay winnings and collect costs			
Company Metadata				
Field	**Sub-field**	**Bytes**	**Value**	**Comments**
Asset Metadata A	ContractType	4	0x0000FF04	Indicates unit
	ContractPointer	16	xxxx.xxxx.xxxx.xxxx (...) .xxxx	Address of the *Asset Definition* file
	Padding	12		Spare
Asset Metadata B	ContractHash	20	################...	Hash of the *Asset Definition* file (**not** the tokenisation!)
	jurisdiction	2	EN	Specifies that the asset is covered by English law
	Options	2	0x0000	No options specified
	Padding	8		Spare

Fig. 4

Stable Seed Funding	
STABLE-S3-T1	Transaction-ID
Version number	Version number
1	Number of inputs
<Stable's current Satochi pool - assume 75000 Satochi	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Stable PubK-Stable	ScriptSig
Sequence number	Sequence number
2	Number of outputs
13,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
61,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 5

			Racehorse share issuance	
			RACEHORSE-S3-T1	Transaction-ID
			Version number	Version number
			1	Number of inputs
			STABLE-S3-T1	Prev Trans Output
			IDX-00	Prev Trans Output index
			Script length	Script length
			Sig-Racehorse PubK-RaceHorse	ScriptSig
			Sequence number	Sequence number
			4	Number of outputs
			5,000	Output value
Redeem script allowing the stable to onsell without restriction			Output script length	Output script length
OP_1 InvestMetaDataA InvestMetadataB PubK-Stable OP_3 OP_CHECKMULTISIG		>>>	OP_HASH160 <redeem script hash> OP_EQUAL	Output script
			3,000	Output value
Redeem script allowing Alice to onsell without restriction			Output script length	Output script length
OP_1 InvestMetaDataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG		>>>	OP_HASH160 <redeem script hash> OP_EQUAL	Output script
			2,000	Output value
Redeem script allowing Bob to onsell without restriction			Output script length	Output script length
OP_1 InvestMetaDataA InvestMetadataB PubK-Bob OP_3 OP_CHECKMULTISIG		>>>	OP_HASH160 <redeem script hash> OP_EQUAL	Output script
			2,000	Output value
Racehorse smart contract to support clause 28 in contract			Output script length	Output script length
OP_1 HorseMetaDataA HorseMetadataB PubK-Racehorse OP_3 OP_CHECKMULTISIG		>>>	OP_HASH160 <redeem script hash> OP_EQUAL	Output script
			LockTime	LockTime

Fig. 6

Stable Fee Payment from Alice	
RACEHORSE-S3-T2	Transaction-ID
Version number	Version number
1	Number of inputs
ALICE-S3-T1	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Alice PubK-Alice	ScriptSig
Sequence number	Sequence number
1	Number of outputs
540,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG **SIGHASH_ANYONECANPAY**	Output script
LockTime	LockTime
Stable Fee Payment from Bob	
RACEHORSE-S3-T3	Transaction-ID
Version number	Version number
1	Number of inputs
BOB-S3-T1	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Bob PubK-Bob	ScriptSig
Sequence number	Sequence number
1	Number of outputs
360,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG **SIGHASH_ANYONECANPAY**	Output script
LockTime	LockTime

Fig. 7

Racehorse Stable Fees	
RACEHORSE-S3-T6	Transaction-ID
Version number	Version number
2	Number of inputs
RACEHORSE-S3-T2	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Racehorse PubK-Racehorse	ScriptSig
RACEHORSE-S2-T4	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Racehorse PubK-Racehorse	ScriptSig
Sequence number	Sequence number
1	Number of outputs
8,999,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

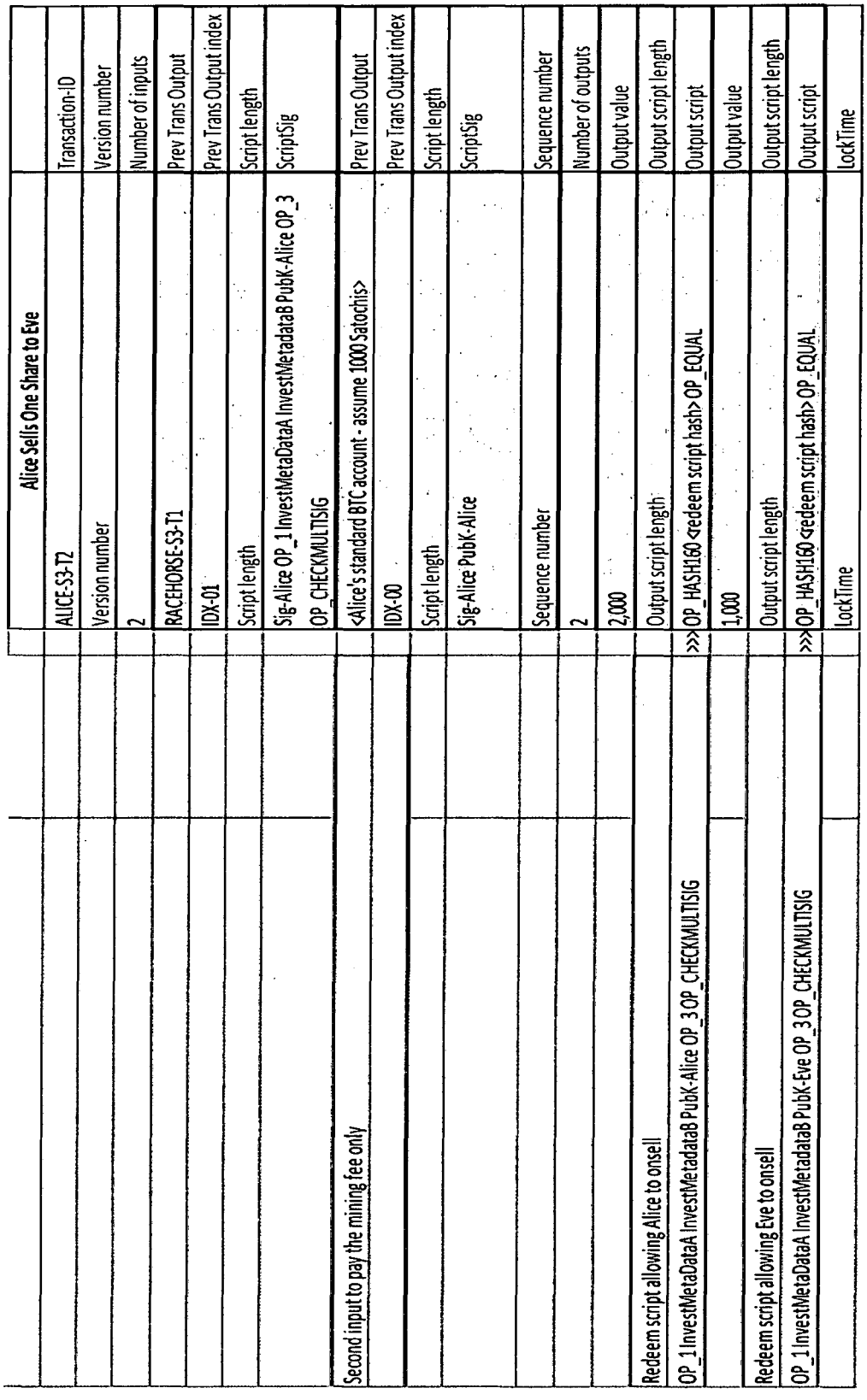
Fig. 8

Racehorse Winnings	
STABLE-S3-T2	Transaction-ID
Version number	Version number
1	Number of inputs
EXCHANGE-S3-T1	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Stable PubK-Stable	ScriptSig
Sequence number	Sequence number
1	Number of outputs
8,000,001,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Racehorse Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 9

Racehorse Winnings to Investors	
RACEHORSE-S3-T8	Transaction-ID
Version number	Version number
1	Number of inputs
STABLE-S3-T2	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Racehorse PubK-Racehorse	ScriptSig
Sequence number	Sequence number
3	Number of outputs
4,000,000,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Stable Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
2,400,000,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Alice Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
1,600,000,000	Output value
Output script length	Output script length
OP_DUP OP_HASH160 <PubK-Bob Hash> OP_EQUALVERIFY OP_CHECKSIG	Output script
LockTime	LockTime

Fig. 10



Alice Sells One Share to Eve	
ALICE-S3-T2	Transaction-ID
Version number	Version number
2	Number of inputs
RACEHORSE-S3-T1	Prev Trans Output
IDX-01	Prev Trans Output index
Script length	Script length
Sig-Alice OP_1 InvestMetaDataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG	ScriptSig
<Alice's standard BTC account - assume 1000 Satochis>	Prev Trans Output
IDX-00	Prev Trans Output index
Script length	Script length
Sig-Alice PubK-Alice	ScriptSig
Sequence number	Sequence number
2	Number of outputs
2,000	Output value
Output script length	Output script length
>>>OP_HASH160 <redeem script hash> OP_EQUAL	Output script
1,000	Output value
Output script length	Output script length
>>>OP_HASH160 <redeem script hash> OP_EQUAL	Output script
LockTime	LockTime

Second input to pay the mining fee only

Redeem script allowing Alice to onsell

OP_1 InvestMetaDataA InvestMetadataB PubK-Alice OP_3 OP_CHECKMULTISIG

Redeem script allowing Eve to onsell

OP_1 InvestMetaDataA InvestMetadataB PubK-Eve OP_3 OP_CHECKMULTISIG

Fig. 11

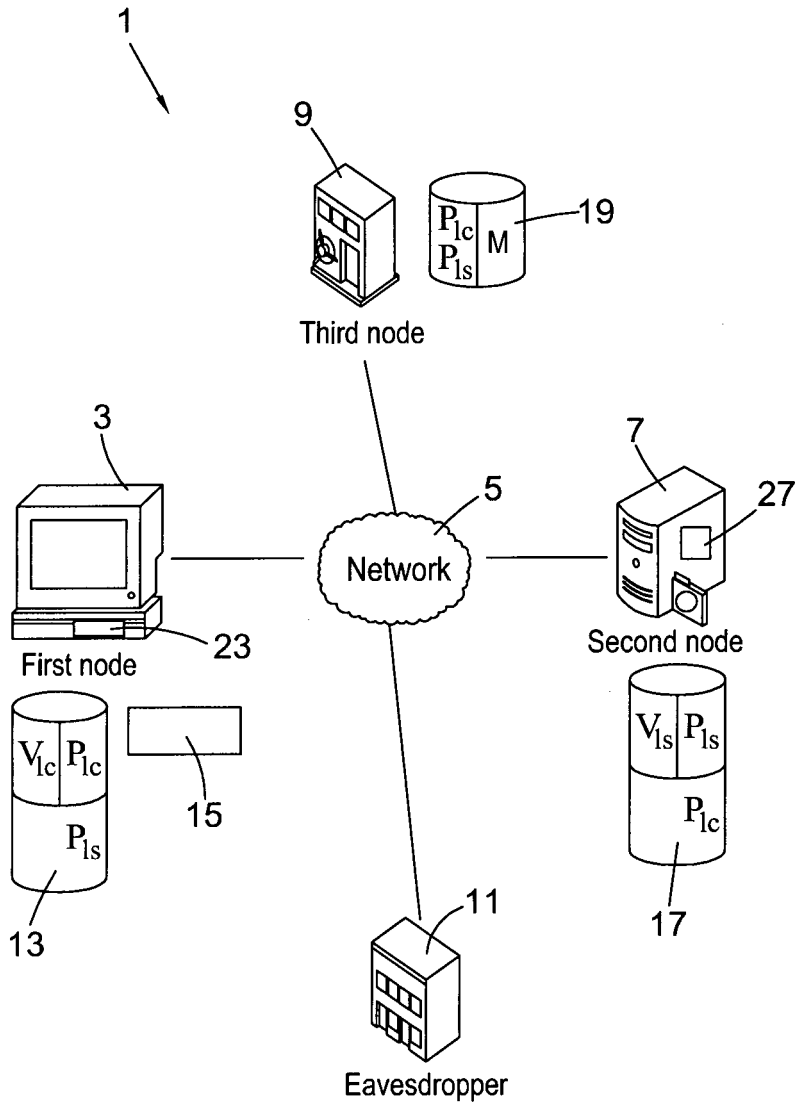

1
9
P_{lc}
P_{ls}
M
19
Third node
3
7
5
27
Network
First node
23
Second node
V_{lc}
P_{lc}
P_{ls}
15
V_{ls}
P_{ls}
P_{lc}
13
17
11
Eavesdropper

Fig. 12

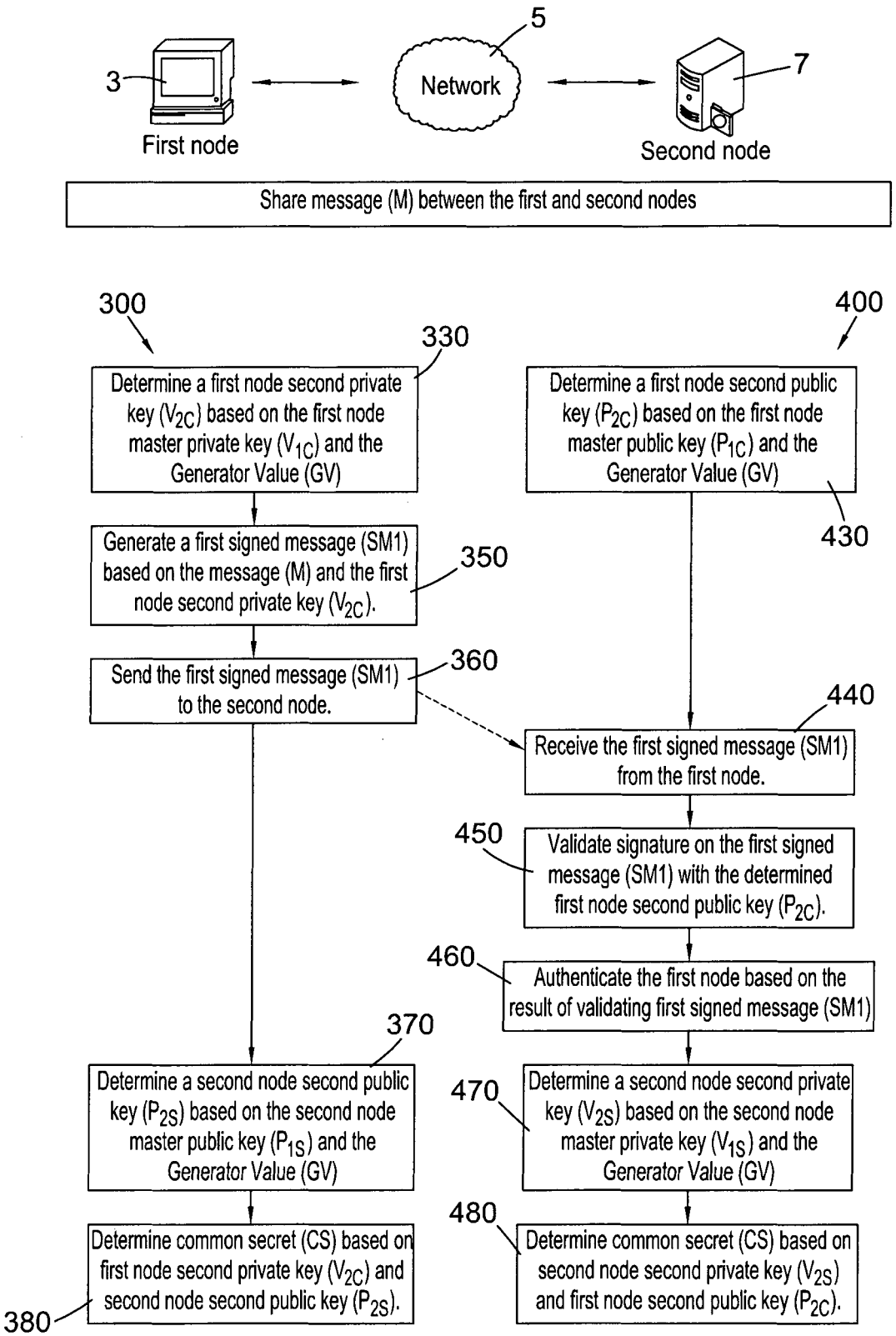

5
3
Network
7
First node
Second node
Share message (M) between the first and second nodes
300
330
Determine a first node second private key (V_{2C}) based on the first node master private key (V_{1C}) and the Generator Value (GV)
Generate a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}).
350
Send the first signed message (SM1) to the second node.
360
370
Determine a second node second public key (P_{2S}) based on the second node master public key (P_{1S}) and the Generator Value (GV)
Determine common secret (CS) based on first node second private key (V_{2C}) and second node second public key (P_{2S}).
380
400
Determine a first node second public key (P_{2C}) based on the first node master public key (P_{1C}) and the Generator Value (GV)
430
440
Receive the first signed message (SM1) from the first node.
450
Validate signature on the first signed message (SM1) with the determined first node second public key (P_{2C}).
460
Authenticate the first node based on the result of validating first signed message (SM1)
470
Determine a second node second private key (V_{2S}) based on the second node master private key (V_{1S}) and the Generator Value (GV)
480
Determine common secret (CS) based on second node second private key (V_{2S}) and first node second public key (P_{2C}).

Fig. 13

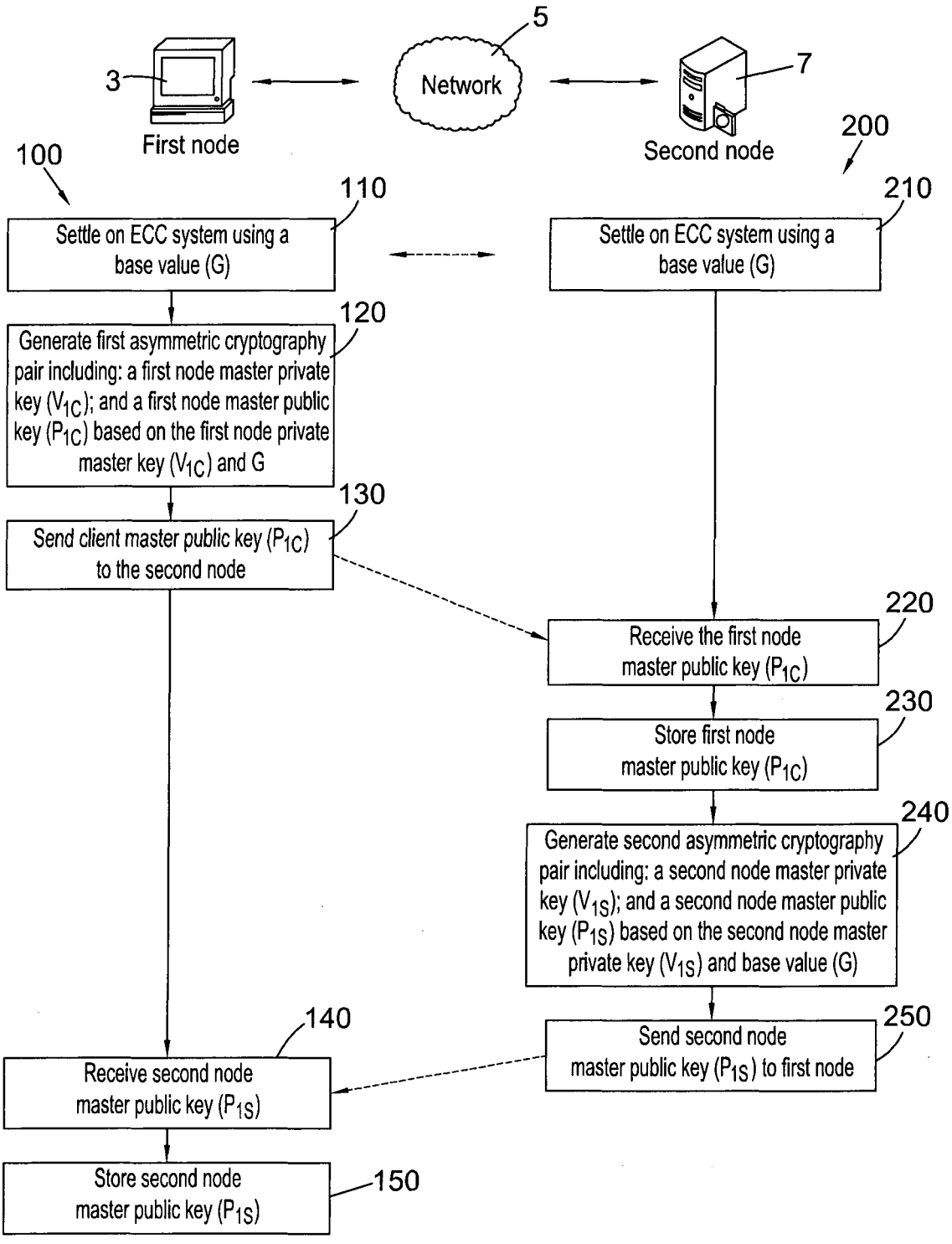

3
First node
5
Network
7
Second node
100
200
110
Settle on ECC system using a base value (G)
210
Settle on ECC system using a base value (G)
120
Generate first asymmetric cryptography pair including: a first node master private key (V_{1C}); and a first node master public key (P_{1C}) based on the first node private master key (V_{1C}) and G
130
Send client master public key (P_{1C}) to the second node
220
Receive the first node master public key (P_{1C})
230
Store first node master public key (P_{1C})
240
Generate second asymmetric cryptography pair including: a second node master private key (V_{1S}); and a second node master public key (P_{1S}) based on the second node master private key (V_{1S}) and base value (G)
250
Send second node master public key (P_{1S}) to first node
140
Receive second node master public key (P_{1S})
Store second node master public key (P_{1S})
150

Fig. 14

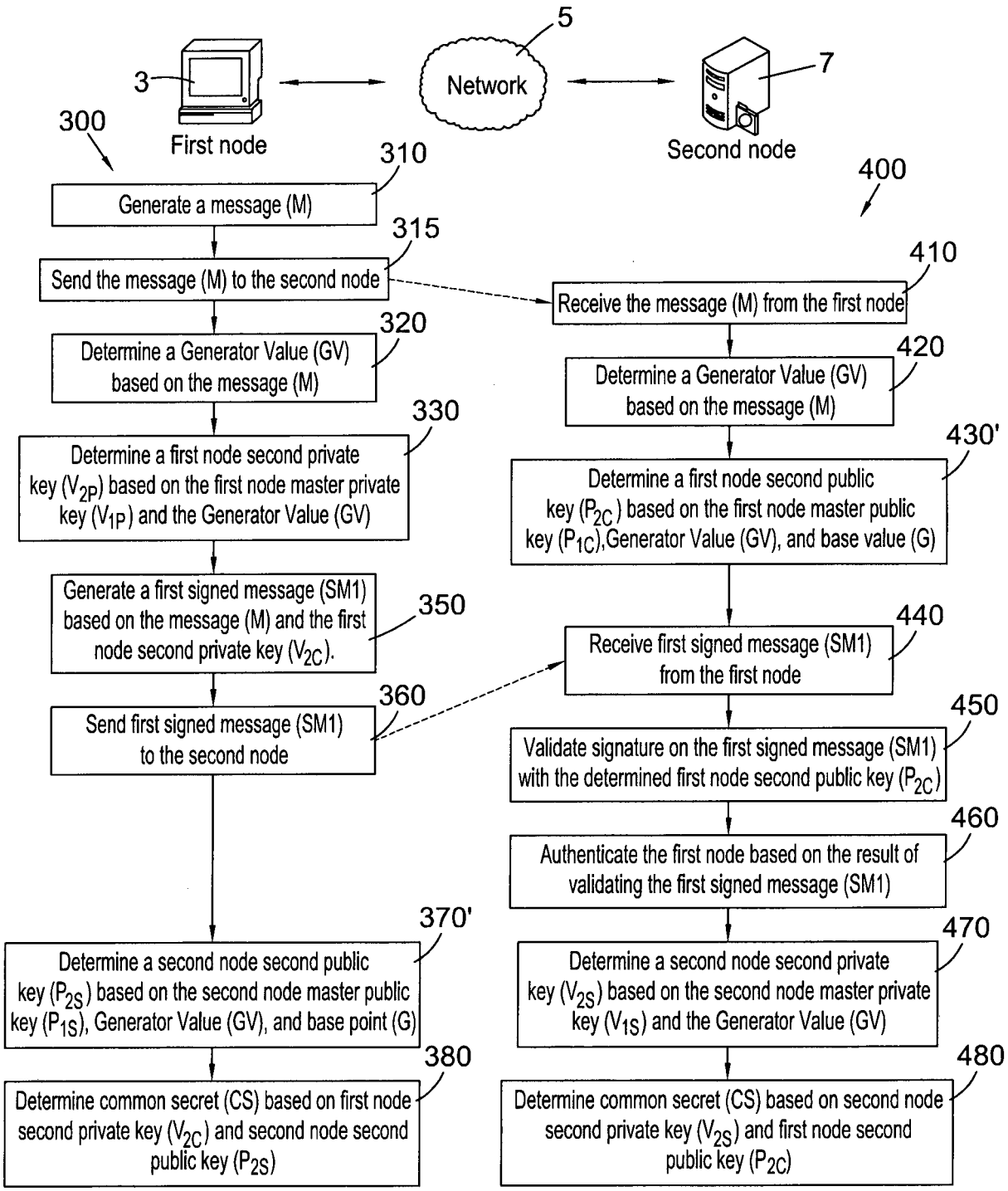

5
3
Network
7
300
First node
Second node
310
400
Generate a message (M)
315
410
Send the message (M) to the second node
Receive the message (M) from the first node
320
420
Determine a Generator Value (GV) based on the message (M)
Determine a Generator Value (GV) based on the message (M)
330
430'
Determine a first node second private key (V_{2P}) based on the first node master private key (V_{1P}) and the Generator Value (GV)
Determine a first node second public key (P_{2C}) based on the first node master public key (P_{1C}),Generator Value (GV), and base value (G)
Generate a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}).
350
440
Receive first signed message (SM1) from the first node
360
Send first signed message (SM1) to the second node
450
Validate signature on the first signed message (SM1) with the determined first node second public key (P_{2C})
460
Authenticate the first node based on the result of validating the first signed message (SM1)
370'
470
Determine a second node second public key (P_{2S}) based on the second node master public key (P_{1S}), Generator Value (GV), and base point (G)
Determine a second node second private key (V_{2S}) based on the second node master private key (V_{1S}) and the Generator Value (GV)
380
480
Determine common secret (CS) based on first node second private key (V_{2C}) and second node second public key (P_{2S})
Determine common secret (CS) based on second node second private key (V_{2S}) and first node second public key (P_{2C})

Fig. 15

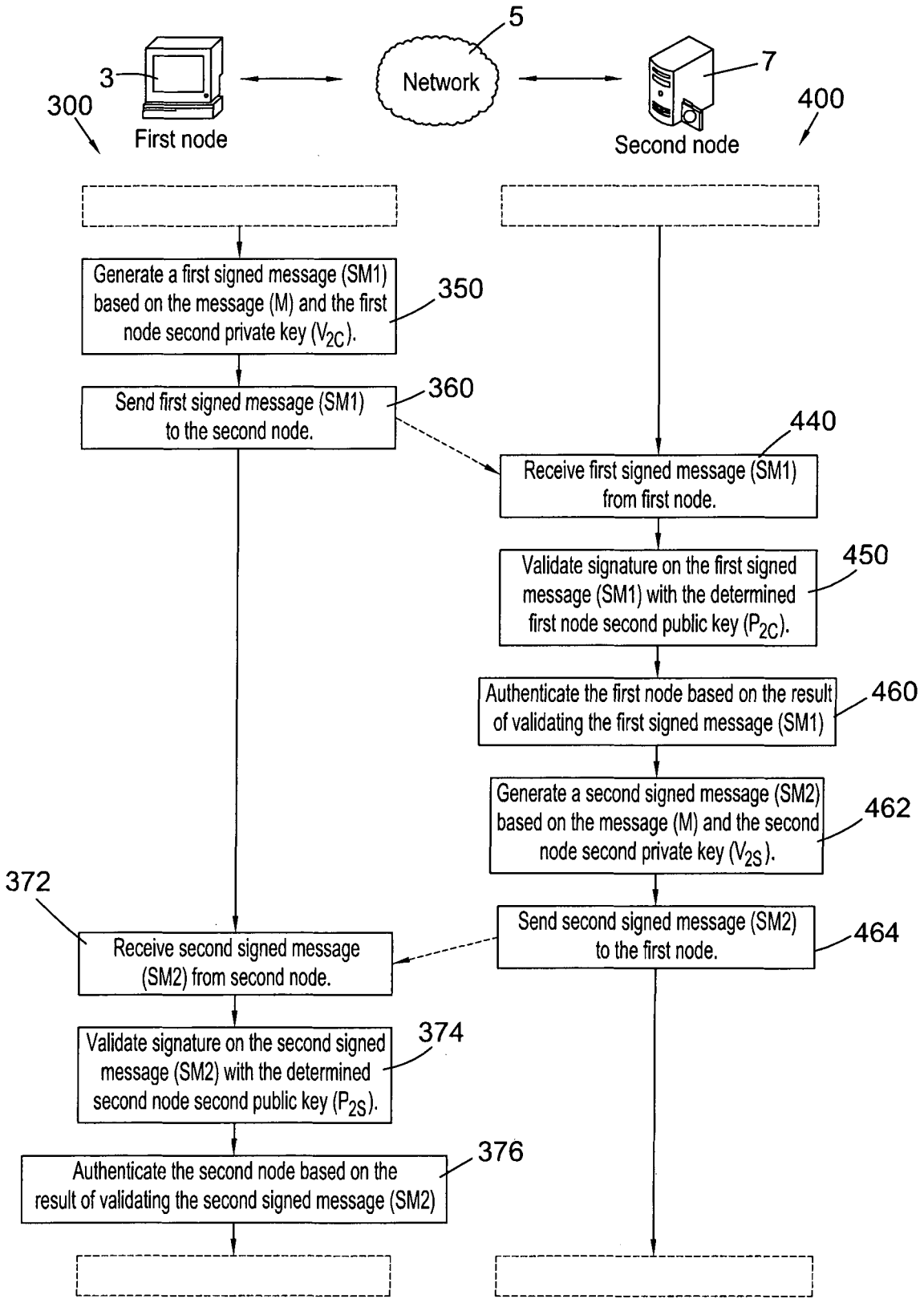

5
Network
3
300
First node
7
400
Second node
Generate a first signed message (SM1) based on the message (M) and the first node second private key (V_{2C}).
350
Send first signed message (SM1) to the second node.
360
440
Receive first signed message (SM1) from first node.
Validate signature on the first signed message (SM1) with the determined first node second public key (P_{2C}).
450
Authenticate the first node based on the result of validating the first signed message (SM1)
460
Generate a second signed message (SM2) based on the message (M) and the second node second private key (V_{2S}).
462
372
Receive second signed message (SM2) from second node.
Send second signed message (SM2) to the first node.
464
Validate signature on the second signed message (SM2) with the determined second node second public key (P_{2S}).
374
Authenticate the second node based on the result of validating the second signed message (SM2)
376

Fig. 16

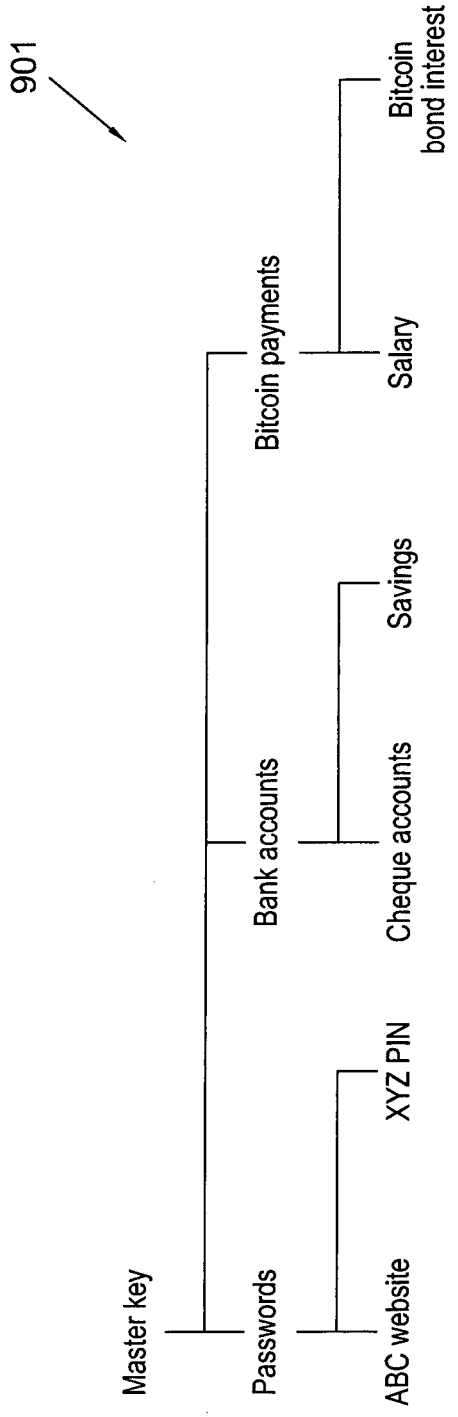

901
Master key
Passwords
ABC website
XYZ PIN
Bank accounts
Cheque accounts
Savings
Bitcoin payments
Salary
Bitcoin bond interest

Fig. 17

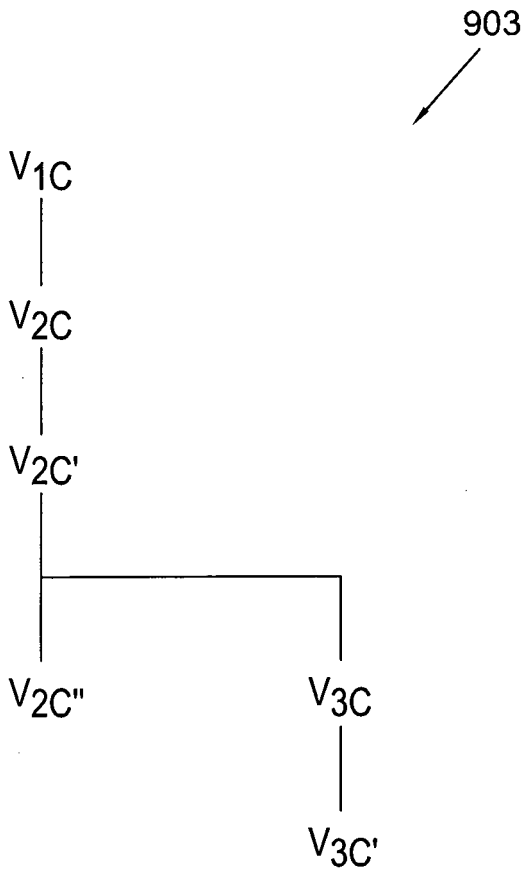


Fig. 18