

Fig. 1

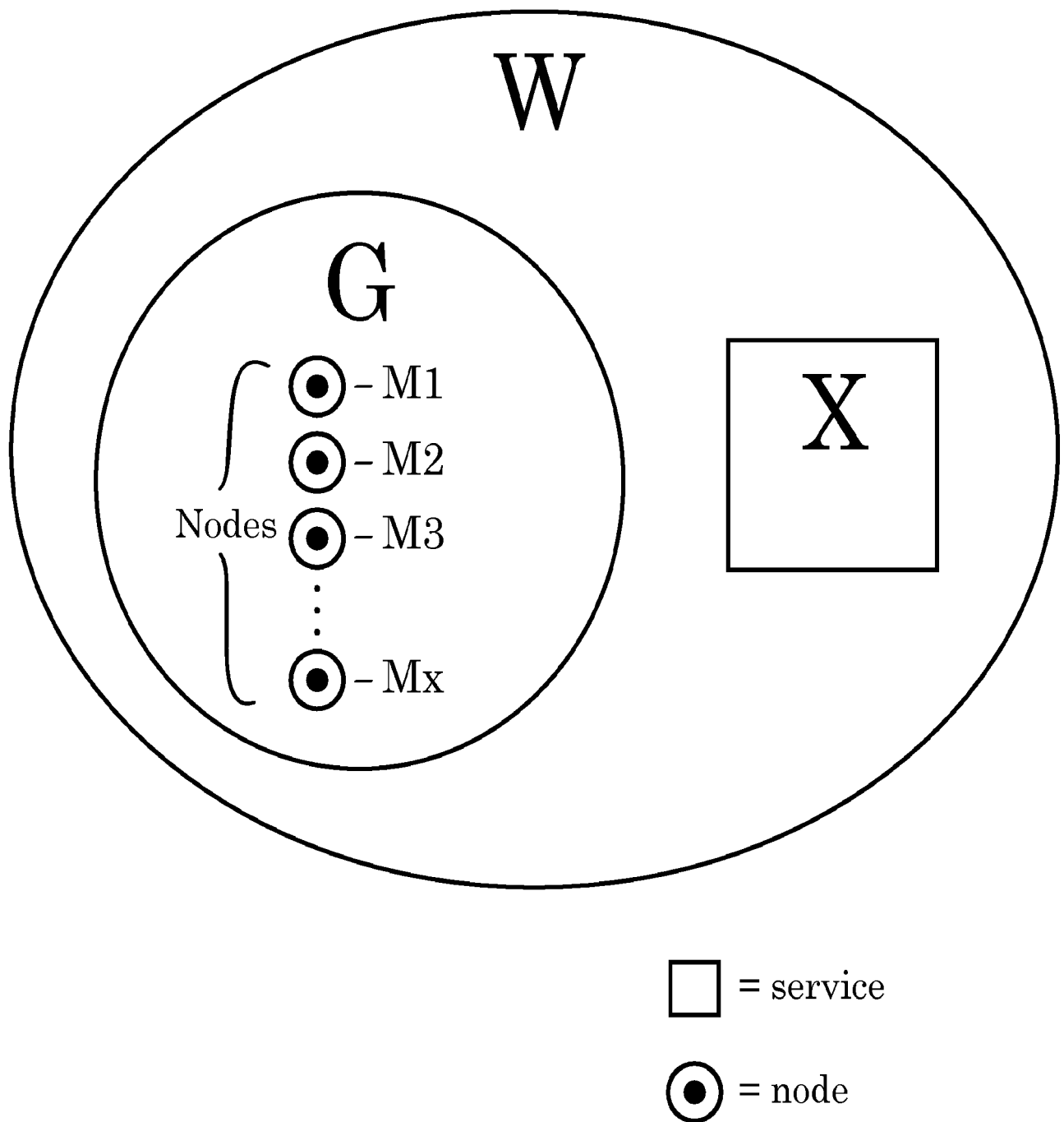


Fig. 2

		Signer						
		M1	M2	M3	M4	M5	...	MN
Signed	M1	M1S1	M2S1	M3S1	M4S1	M5S1		MNS1
	M2	M1S2	M2S2	M3S2	M4S2	M5S2		MNS2
	M3	M1S3	M2S3	M3S3	M4S3	M5S3		MNS3
	M4	M1S4	M2S4	M3S4	M4S4	M5S4		MNS4
	M5	M1S5	M2S5	M3S5	M4S5	M5S5		MNS5
	⋮							
	MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN

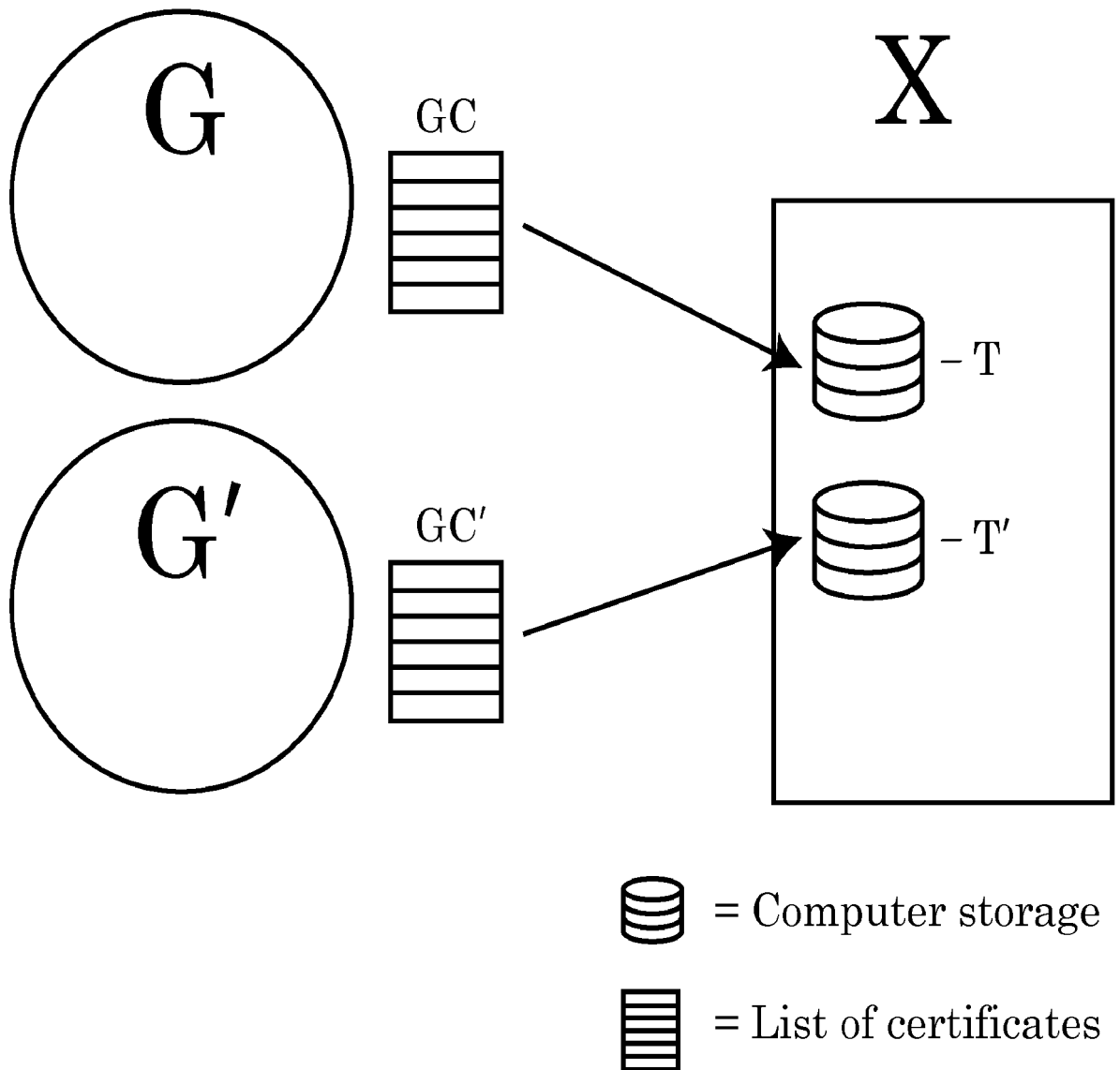
Fig. 3

		Signer						
		M1	M2	M3	M4	M5	M6	M7
Signed	M1	M1S1				M5S1	M6S1	M7S1
	M2	M1S2	M2S2				M6S2	M7S2
	M3	M1S3	M2S3	M3S3				M7S3
	M4	M1S4	M2S4	M3S4	M4S4			
	M5		M2S5	M3S5	M4S5	M5S5		
	M6			M3S6	M4S6	M5S6	M6S6	
	M7				M4S7	M5S7	M6S7	M7S7

Assume
N = 7

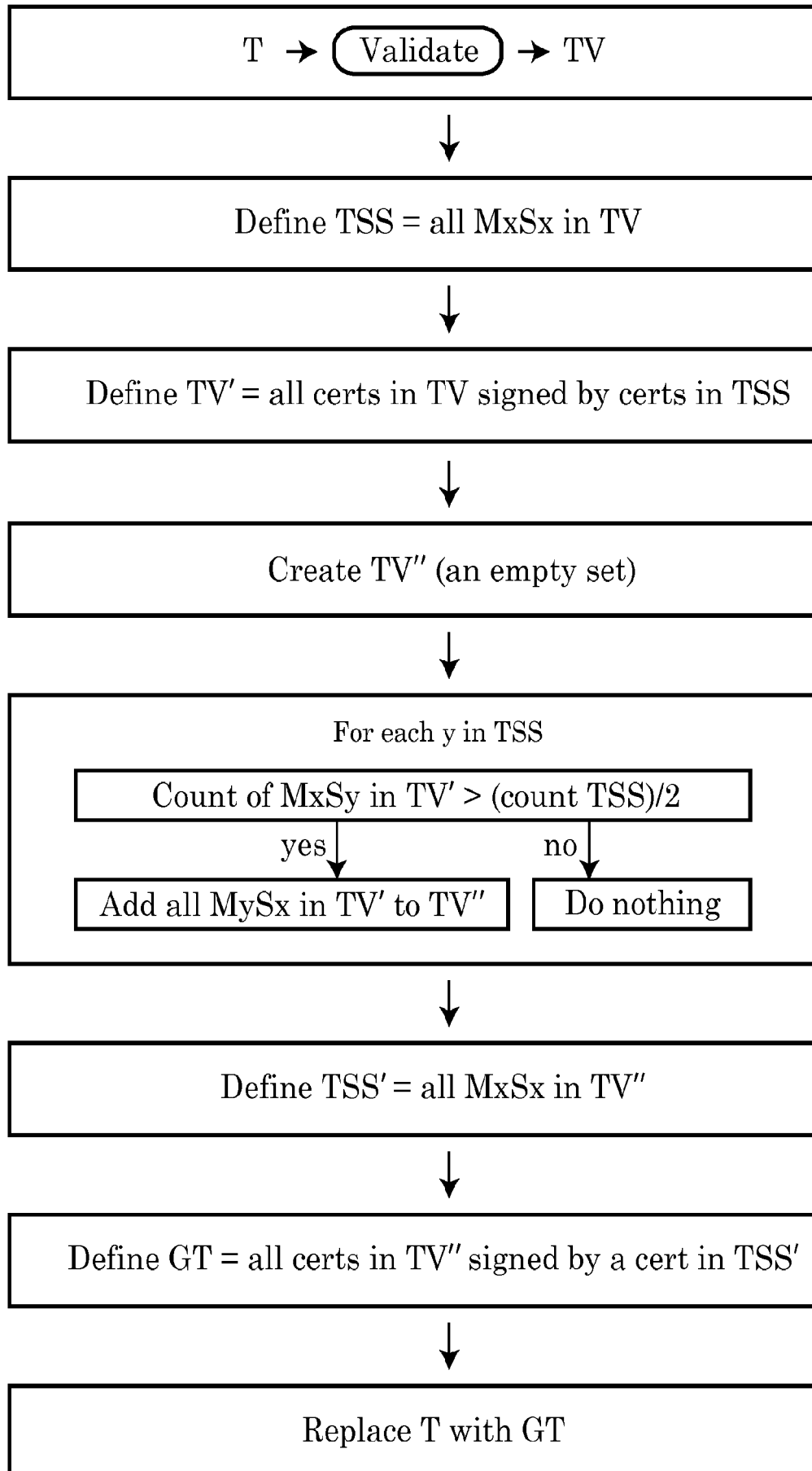
Each M_x signs (N/2 rounded down) other certificates.

Fig. 4



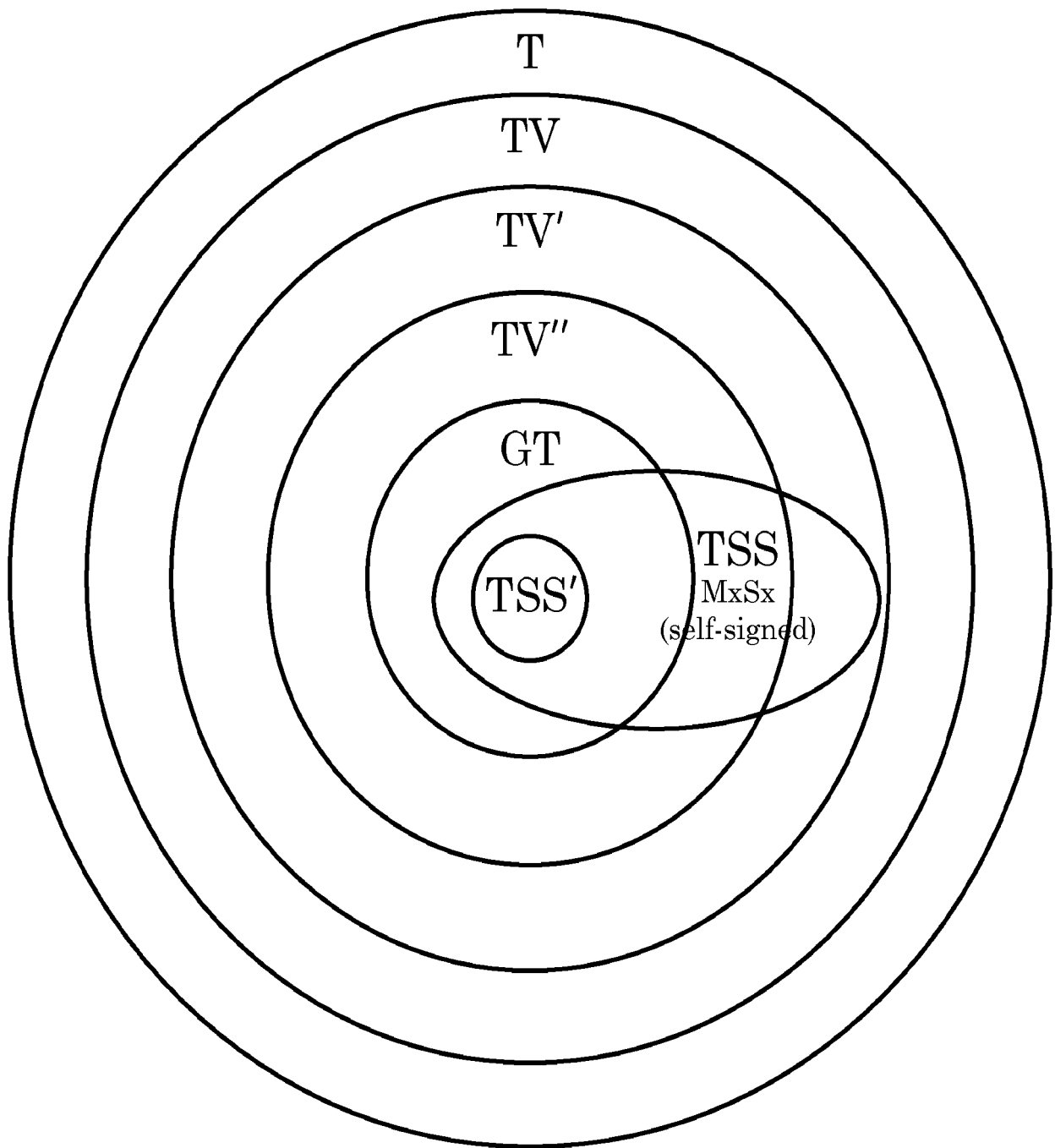
5/21

Fig. 5.1



6/21

Fig. 5.2



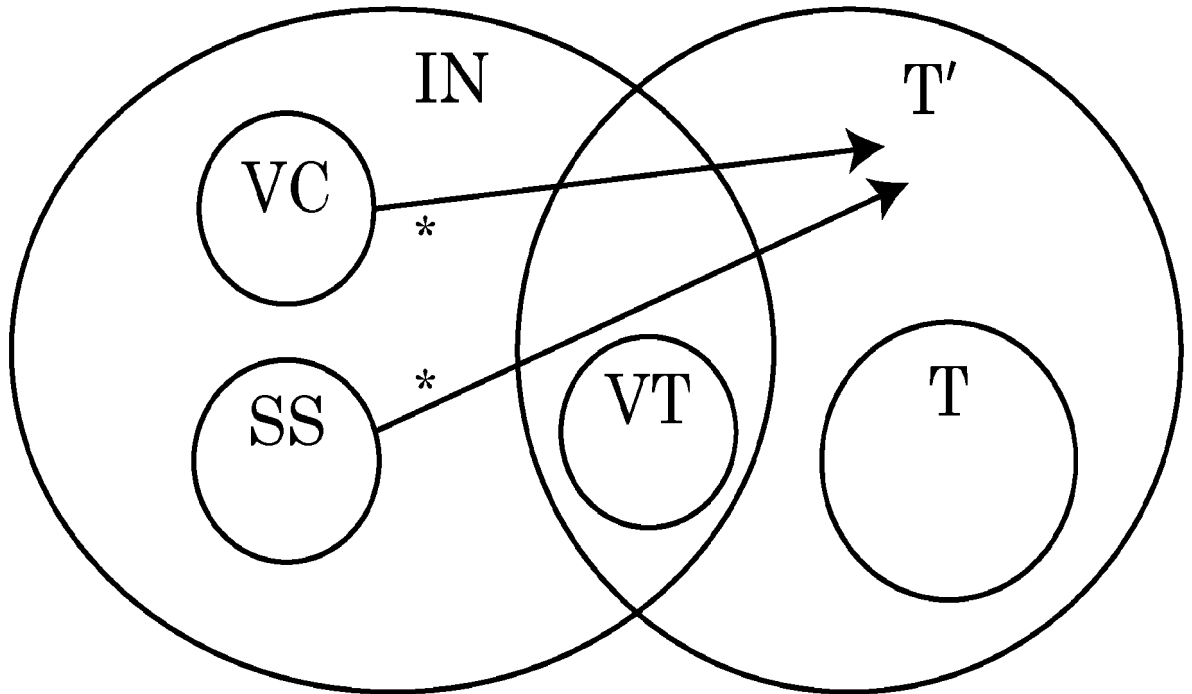
7/21

Fig. 6

		Signer							
		M1	M2	M3	M4	M5	...	MN	<i>Mp</i>
Signed	M1	M1S1	M2S1	M3S1	M4S1	M5S1		MNS1	<i>MpS1</i>
	M2	M1S2	M2S2	M3S2	M4S2	M5S2		MNS2	<i>MpS2</i>
	M3	M1S3	M2S3	M3S3	M4S3	M5S3		MNS3	<i>MpS3</i>
	M4	M1S4	M2S4	M3S4	M4S4	M5S4		MNS4	<i>MpS4</i>
	M5	M1S5	M2S5	M3S5	M4S5	M5S5		MNS5	<i>MpS5</i>
	⋮								
	MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN	<i>MpSN</i>
	<i>Mp</i>	<i>M1Sp</i>	<i>M2Sp</i>	<i>M3Sp</i>	<i>M4Sp</i>	<i>M5Sp</i>		<i>MNSp</i>	<i>MpSp</i>

where $p = N+1$

Fig. 7



T = trusted store for G in X

VC = all MySx in IN for y successful in step 2d

T' = all certs in IN or T signed by MxSx in T

SS = all MxSx in IN

IN = certs to be added

VT = all certs in T' targeting MxSp in B

* If Step 2f successful

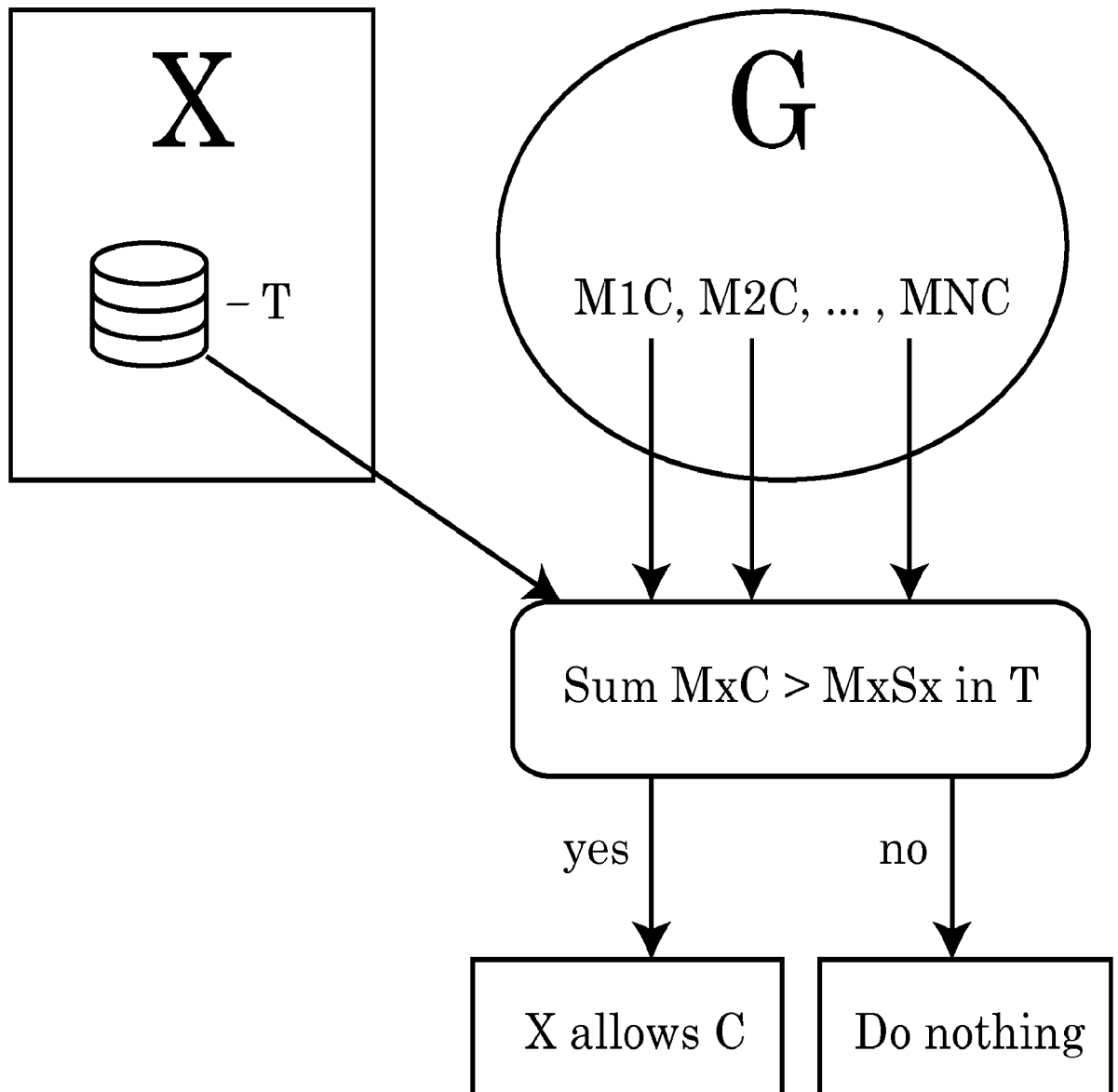
9/21

Fig. 8

		Signer						
		M1	M2	M3	M4	M5	...	MN
Signed	M1	M1S1	M2S1	M3S1	M4S1	M5S1		MNS1
	M2	<i>M1R2</i>	M2S2	M3S2	M4S2	M5S2		MNS2
	M3	M1S3	M2S3	M3S3	M4S3	M5S3		MNS3
	M4	M1S4	M2S4	M3S4	M4S4	M5S4		MNS4
	M5	M1S5	M2S5	M3S5	M4S5	M5S5		MNS5
	⋮							
	MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN

$M_xR_y = \text{revoked}$

Fig. 9



11/21

Fig. 10

```

{"keys":[
  {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
  {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
  {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
  {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
  {"id":"M5","key":"BC1048A392076098","algorithm":"ECC"},
  {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
  {"id":"M7","key":"83F0760992CDABE8","algorithm":"DSA"},
],
"hash_type":"SHA256",
"notvalidbefore":0,
"expiry":4000000000,
"clock":1,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDKFJSELKRJSELKJFDL",
  "M5":"KFJSELKRJSEFJLEHRISKHLKFJSELKRJSEKJDKLDJ",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M7":"DKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLSLDJ",
  }
}

```

In this figure using JSON syntax, G is represented by the list “keys”. Signatures for each M_x in G are found in “signatures”.

12/21

Fig. 11

```

{"keys":[
  {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
  {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
  {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
  {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
  {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
  {"id":"M8","key":"760DABE992C83F08","algorithm":"DSA"},
],
"hash_type":"SHA256",
"notvalidbefore":0,
"expiry":4000000000,
"clock":2,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDKFJSELKRJSELKJFDL",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M8":"FJLFELKJFDLKKLSJEHRISKHLKJDDKSELKRJSLDJ",
  }
}

```

In this figure using JSON syntax, M5 has been removed from G, and M8 has been added, assuming this document replaces that found in Figure 11.

13/21

Fig. 12

```

{"keys":[
  { "keys":[
    {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
    {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
    {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
    ],
    "quorum":2,
  },
  { "keys":[
    {"id":"M5","key":"BC1048A392076098","algorithm":"ECC"},
    {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
    {"id":"M7","key":"83F0760992CDABE8","algorithm":"DSA"},
    ],
    "quorum":2,
  },
  { "keys":[
    {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
    ],
    "quorum":1,
  },
],
"hash_type":"SHA256",
"notvalidbefore":0,
"quorum":2,
"expiry":4000000000,
"clock":1,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDKFJSELKRJSELKJFDL",
  "M5":"KFJSELKRJSEFJLEHRISKHLKFJSELKRJSEKJDKLDJ",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M7":"DKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLSLDJ",
  }
}

```

In this figure using JSON syntax the “keys” field contains objects that each have an algebra for calculating the replacement .

Fig. 13

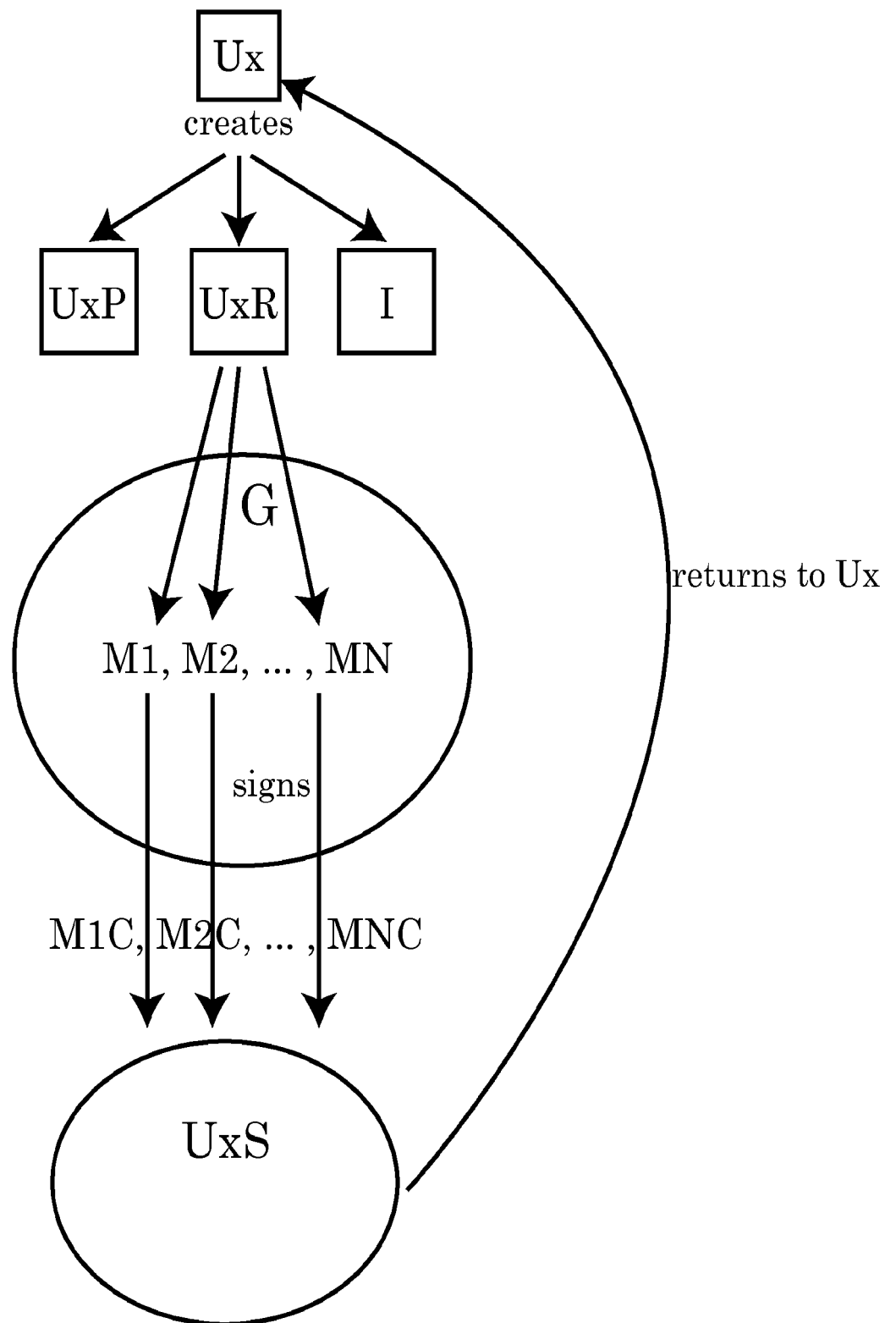


Fig. 14

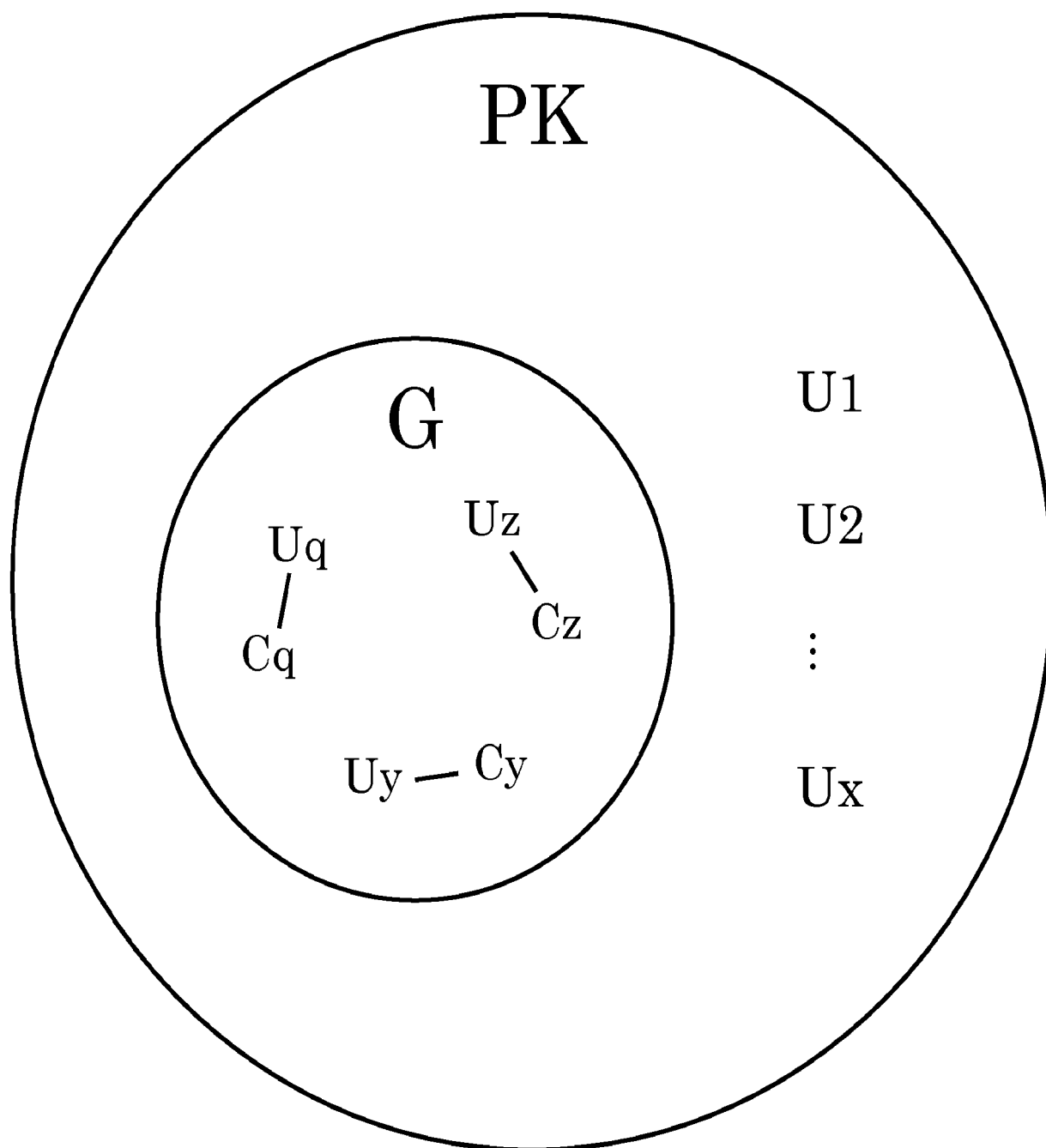


Fig. 15

```
{ "credential": {  
  "id": 4,  
  "policies": {  
    "renewal": "(1 and 2) or (3 and 2) or (1 and 3)",  
    "dataAccess": "(1 and 2) or (3 and 2) or (1 and 3)",  
    "authentication": "(1 and 2) or (3 and 2) or (1 and 3)"  
  },  
  "nonce": "SLKJa2r36FkRHS"  
},  
"hash_type": "SHA256",  
"signatureCA": "DLKSJLSEKRRHSELKJKDFLFLFKJS"  
}
```

In this figure, Ux with id 4 has set 3 policies, one for renewal, one for dataAccess and one for authentication with the CA. The CA has signed this document, specifically the “credential” sub-object string.

Fig. 16

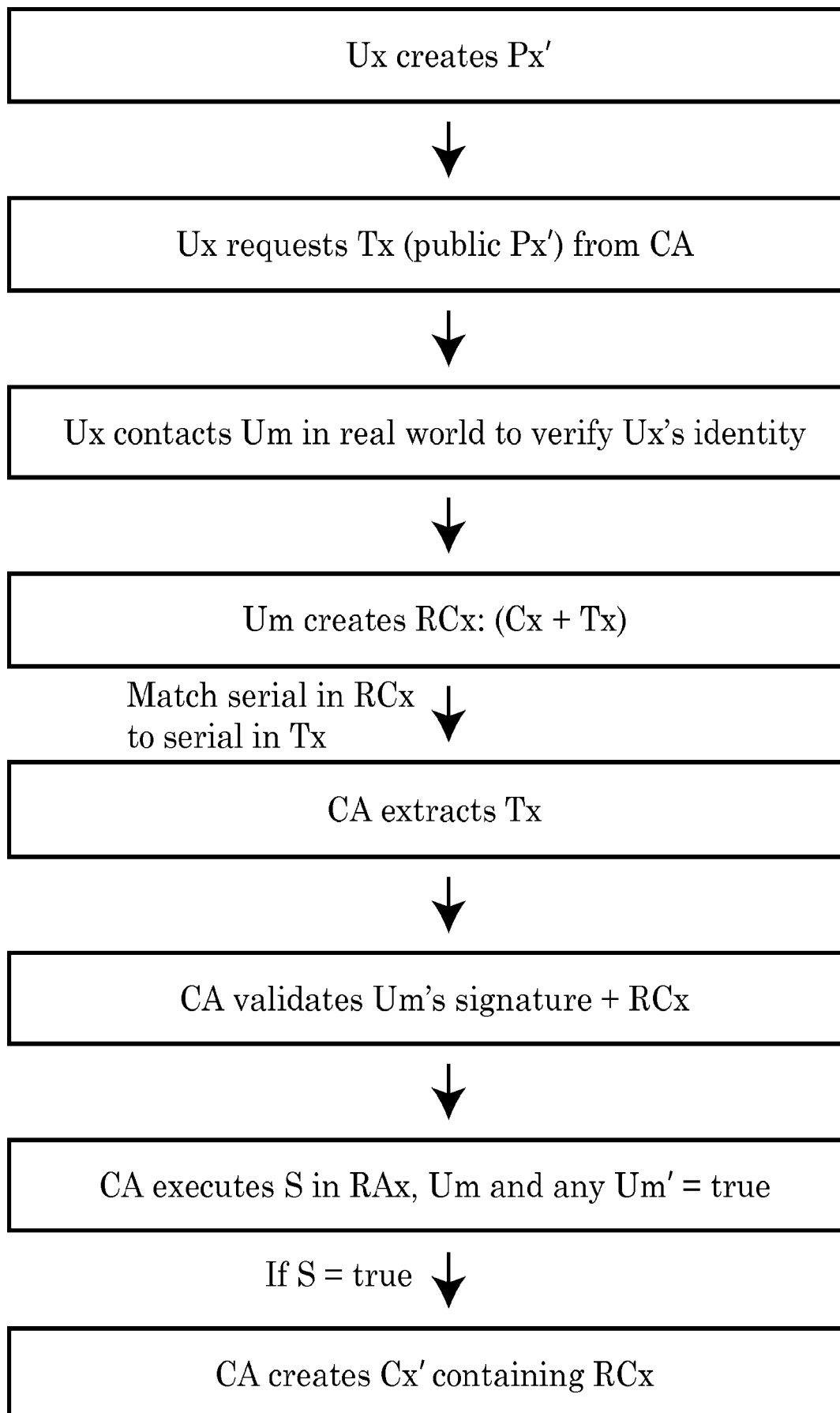


Fig. 17

```
"renewal":{
  "id":4,
  "serial":850293,
  "nonce":"SLKJSELRKHS"
},
"hash_type":"SHA256",
"vouchers":[
  {"id":1,"signature":"DLKSJLSEKRRHSELKJKDFLFLKJS"},
  {"id":2,"signature":"AWEOTQPNVXIMNERIOWJFLDIK"}
]
```

In this figure, U_x has id 4, and U_m 1 and 2 have signatures of the “renewal” sub-object string present in this document instead of sending two identical documents with different signatures.

Fig. 18

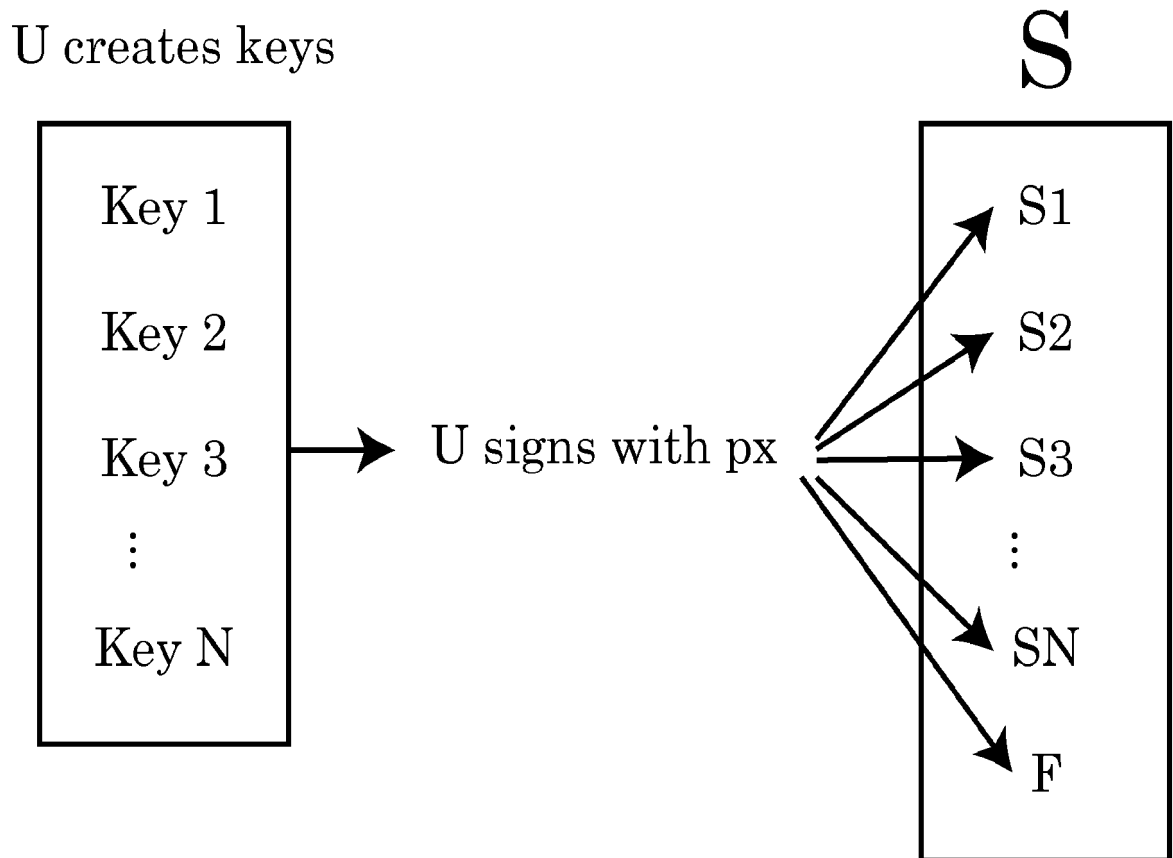


Fig. 19

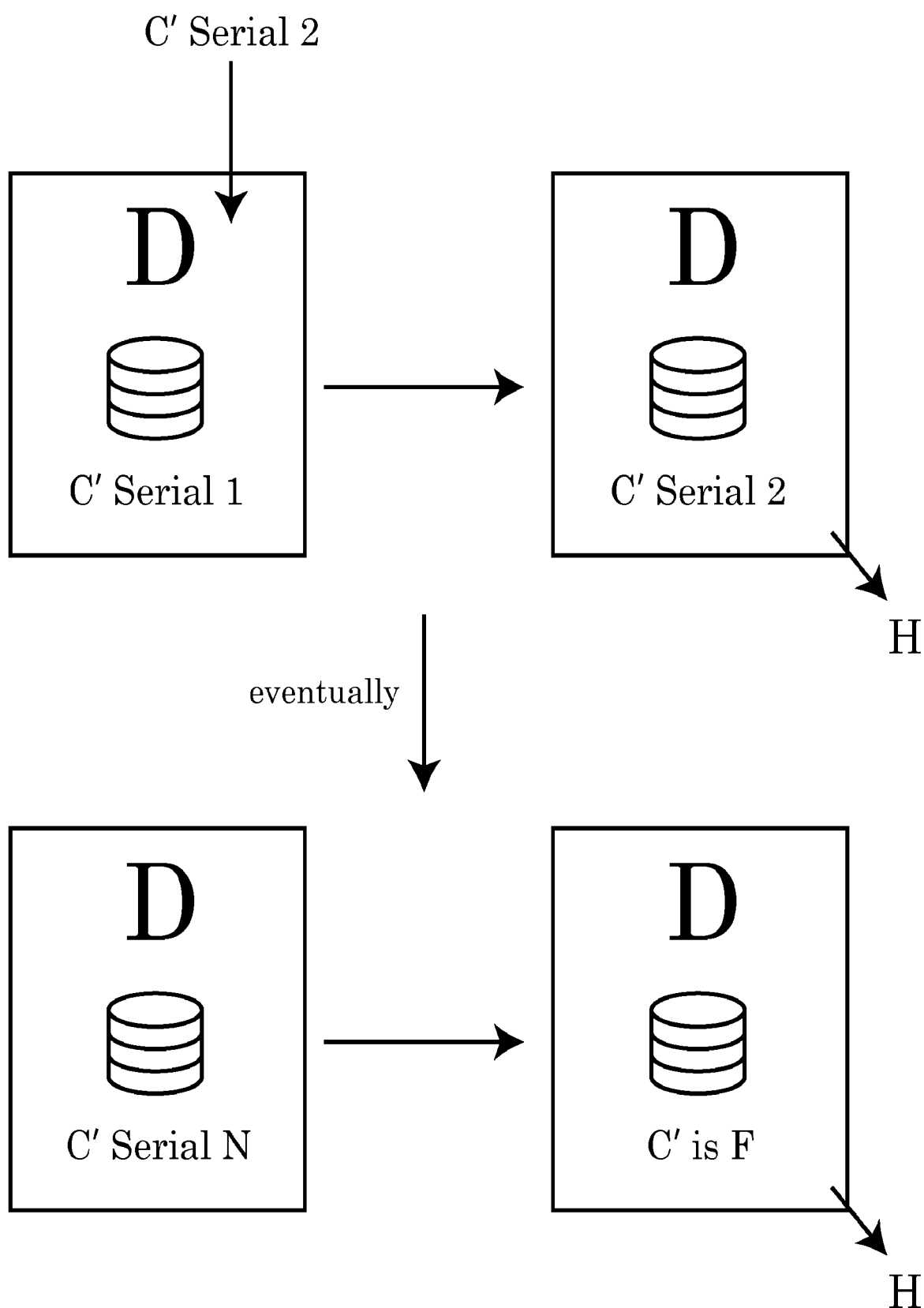


Fig. 20

