US009607158B2

(12) **United States Patent**
Yamamoto et al.

(10) **Patent No.: US 9,607,158 B2**
(45) **Date of Patent: *Mar. 28, 2017**

(54) **PROXY COMPUTING SYSTEM, COMPUTING APPARATUS, CAPABILITY PROVIDING APPARATUS, PROXY COMPUTING METHOD, CAPABILITY PROVIDING METHOD, PROGRAM, AND RECORDING MEDIUM**

(75) Inventors: **Go Yamamoto**, Tokyo (JP); **Tetsutaro Kobayashi**, Tokyo (JP)

(73) Assignee: **NIPPON TELEGRAPH AND TELEPHONE CORPORATION**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 748 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **13/881,111**

(22) PCT Filed: **Oct. 25, 2011**

(86) PCT No.: **PCT/JP2011/074546**

§ 371 (c)(1),
(2), (4) Date: **Jul. 9, 2013**

(87) PCT Pub. No.: **WO2012/057134**

PCT Pub. Date: **May 3, 2012**

(65) **Prior Publication Data**

US 2013/0318360 A1 Nov. 28, 2013

(30) **Foreign Application Priority Data**

Oct. 26, 2010 (JP) 2010-239342
Jan. 14, 2011 (JP) 2011-005899
(Continued)

(51) **Int. Cl.**
G06F 21/60 (2013.01)
H04L 9/00 (2006.01)
H04L 9/30 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 21/602*** (2013.01); ***H04L 9/008*** (2013.01); ***H04L 9/3013*** (2013.01);
(Continued)

(58) **Field of Classification Search**
CPC G06F 21/602; H04L 9/008; H04L 9/3013; H04L 2209/76; H04L 2209/08; H04L 9/3073; H04L 9/304
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,509,728 B1 * 1/2003 Uchino G01R 23/16 324/76.13
7,707,420 B1 * 4/2010 Little H04L 9/3247 380/30
(Continued)

FOREIGN PATENT DOCUMENTS

EP 2 525 341 A1 11/2012

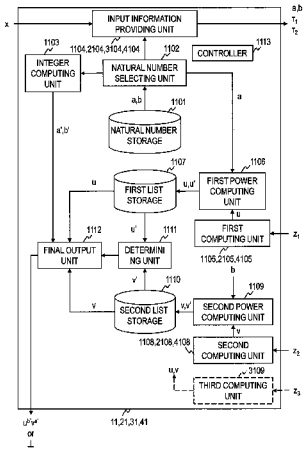
OTHER PUBLICATIONS

Elgamal, T., "A Public Key Cryptosystem and a signature scheme based on Discrete Logarithms", IEEE Transactions on Information Theory V. IT-31, No. 4 or CRYPTO, vol. 84, pp. 10-18, (1985).
(Continued)

Primary Examiner — Farid Homayounmehr
Assistant Examiner — Sanchit Sarker
(74) *Attorney, Agent, or Firm* — Oblon, McClelland, Maier & Neustadt, L.L.P.

(57) **ABSTRACT**

A computing apparatus outputs τ_1 and τ_2 corresponding to a ciphertext x, a capability providing apparatus uses τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets the result of the computation as z_1, uses τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets the result of the computation as z_2, the computing apparatus generates a
(Continued)



computation result $u=f(x)^b x_1$ from z_1, generates a computation result $v=f(x)^a x_2$ from z_2, and outputs $u^{b'}v^{a'}$ if the computation results u and v satisfy a particular relation, where G and H are groups, f(x) is a function for obtaining an element of the group G for $x \epsilon H$, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, and x_2 is a realization of the random variable X_2.

61 Claims, 45 Drawing Sheets

(30) **Foreign Application Priority Data**

Mar. 31, 2011 (JP) 2011-077779
Apr. 12, 2011 (JP) 2011-088002

(52) **U.S. Cl.**
CPC ***H04L 9/3073*** (2013.01); *H04L 2209/08* (2013.01); *H04L 2209/76* (2013.01)

(58) **Field of Classification Search**
USPC 713/189
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,949,614	B1 *	2/2015	Jernigan, IV	H04L 63/00 711/114
9,037,623	B2 *	5/2015	Yamamoto	H04L 9/22 708/250
2007/0169179	A1 *	7/2007	Narad	H04L 45/16 726/4
2009/0080658	A1 *	3/2009	Waters	G06F 21/6218 380/277
2009/0129600	A1 *	5/2009	Brickell	H04L 9/3073 380/282
2010/0070768	A1 *	3/2010	Furukawa	H04L 9/3073 713/171
2010/0082973	A1 *	4/2010	Brickell	H04L 9/3073 713/155
2010/0098253	A1 *	4/2010	Delerablee	H04L 9/3073 380/259
2010/0329454	A1 *	12/2010	Takashima	H04L 9/3073 380/44
2011/0194698	A1 *	8/2011	Asano	H04L 9/0643 380/282
2011/0200186	A1 *	8/2011	Ghouti	H04L 9/3066 380/43
2012/0323981	A1 *	12/2012	Yamamoto	H04L 9/22 708/250
2013/0318360	A1 *	11/2013	Yamamoto	G06F 21/602 713/189
2015/0358162	A1 *	12/2015	Kobayashi	H04L 9/3013 713/189

OTHER PUBLICATIONS

Gentry, C., "A Simple BGN-type Crytosystem from LWE", Advances in Cryptology—Eurocrypt, Total 16 Pages, (2010).
Menezes, A. J., "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, ISBN, Total 15 Pages, (1993).
Boyen, X. et al., "Identity Based Crryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems", Network Working Group, pp. 1-63, (Dec. 2007).
Dent, A. W., "Hidden Pairings and Trapdoor DDH Groups", ANTS, LNCS 4076, Total 17 Pages, (2006).
Miller, V., "Short Programs for functions on Curves", URL: http://crypto.stanford.edu/miller/miller/pdf, pp. 2-7, (May 6, 1986).
Miyaji, A., et al.,"New Explicit Conditions of Elliptic Curve Traces for FR-Reduction," IEICE Trans. Fundamentals., vol. E84-A, No. 5, pp. 1234-1243, (May 2001).
Barreto, P. S. L. M., et al., "Constructing Elliptic Curves with Prescribed Embedding Degrees," SCN 2002, LNCS 2576, pp. 257-267, (2003).
Dupont, R., et al., "Building curves with arbitrary small MOV degree over finite prime fields", URL: http://eprint.iacr.org/2002/094/, pp. 1-13, (Jul. 18, 2002).
Buchmann, J., "Introduction to Cryptography", Springer Verlag, pp. 52-56, (Jul. 2001) (with partial English translation).
International Search Report Issued Nov. 22, 2011 in PCT/JP11/074546 Filed Oct. 25, 2011.
Supplementary Partial European Search Report issued Sep. 9, 2016 in Patent Application 11836267.2.
Go Yamamoto, et al., "On Self-Correctors for Homomorphic Functions" SCIS 2010, The 2010 Symposium on Cryptography and Information Security, XP008172377, Jan. 19, 2010, pp. 1-6.
Yevgeniy Dodis, et al., "Chosen-Ciphertext Security of Multiple Encryption" Theory of cryptography: Second Theory of Cryptography Conference, XP047029371, Feb. 10, 2005, pp. 188-209.
Marcos Kiwi, et al., "Exact and Approximate Testing/Correcting of Algebraic Functions: A Survey" Correct System Design, Springer International Publishing, vol. 2292, XP055290782, Jan. 1, 2002, pp. 30-83.
Extended European Search Report issued Dec. 13, 2016 in Patent Application No. 11836267.2.

* cited by examiner

FIG.1

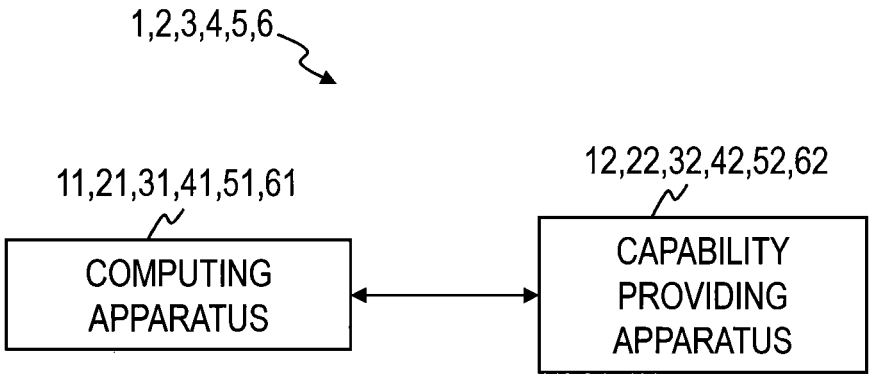

1,2,3,4,5,6
11,21,31,41,51,61
COMPUTING APPARATUS
12,22,32,42,52,62
CAPABILITY PROVIDING APPARATUS

FIG.2

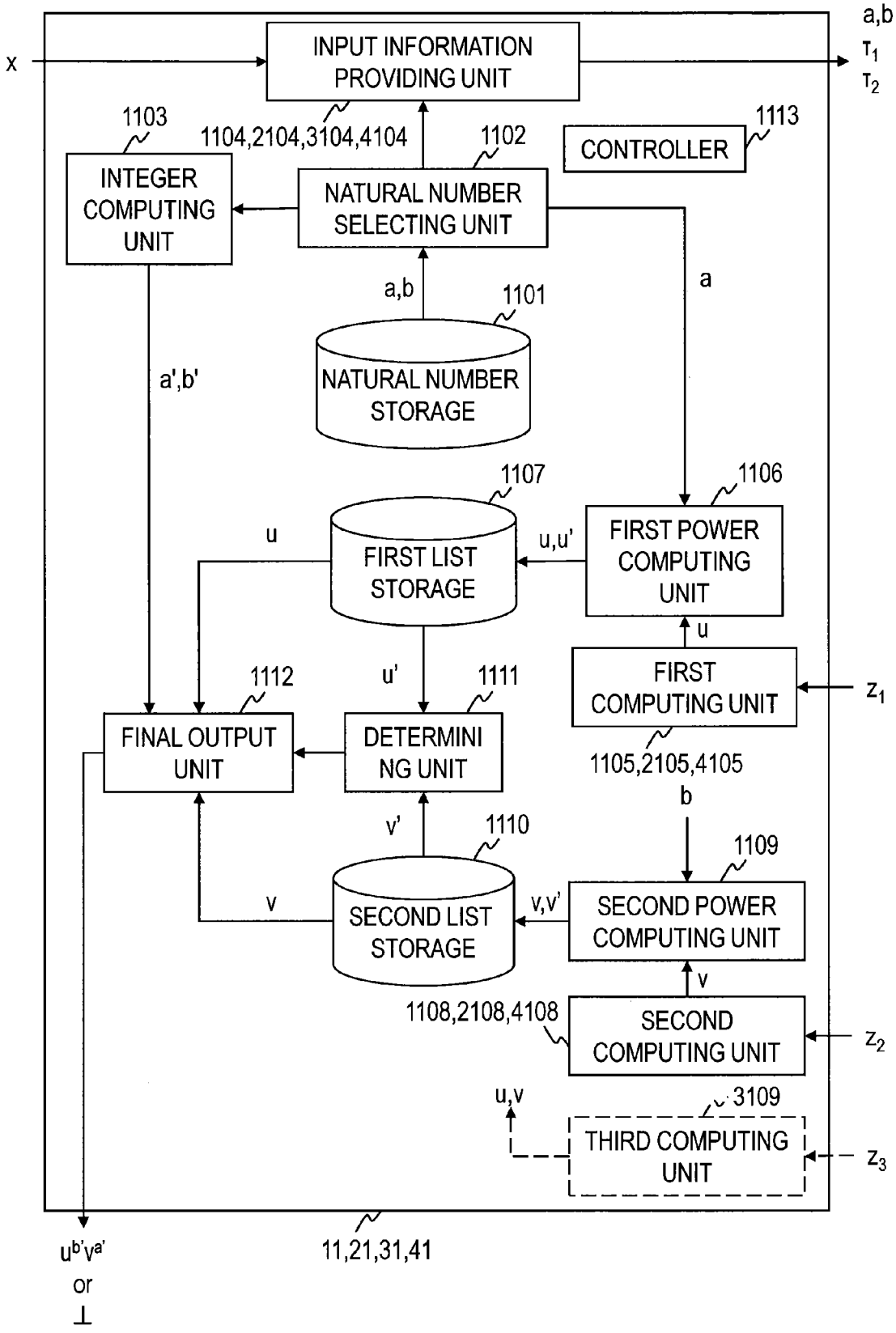

x
INPUT INFORMATION PROVIDING UNIT
a,b
τ_1
τ_2
1103
1104,2104,3104,4104
1102
CONTROLLER
1113
INTEGER COMPUTING UNIT
NATURAL NUMBER SELECTING UNIT
a,b
1101
a
a',b'
NATURAL NUMBER STORAGE
1107
1106
u
FIRST LIST STORAGE
u,u'
FIRST POWER COMPUTING UNIT
u
1112
u'
1111
FIRST COMPUTING UNIT
z_1
FINAL OUTPUT UNIT
DETERMINING UNIT
1105,2105,4105
b
v'
1110
1109
v
SECOND LIST STORAGE
v,v'
SECOND POWER COMPUTING UNIT
v
1108,2108,4108
SECOND COMPUTING UNIT
z_2
u,v
3109
THIRD COMPUTING UNIT
z_3
11,21,31,41
$u^{b'}v^{a'}$
or
$\perp$

FIG.3

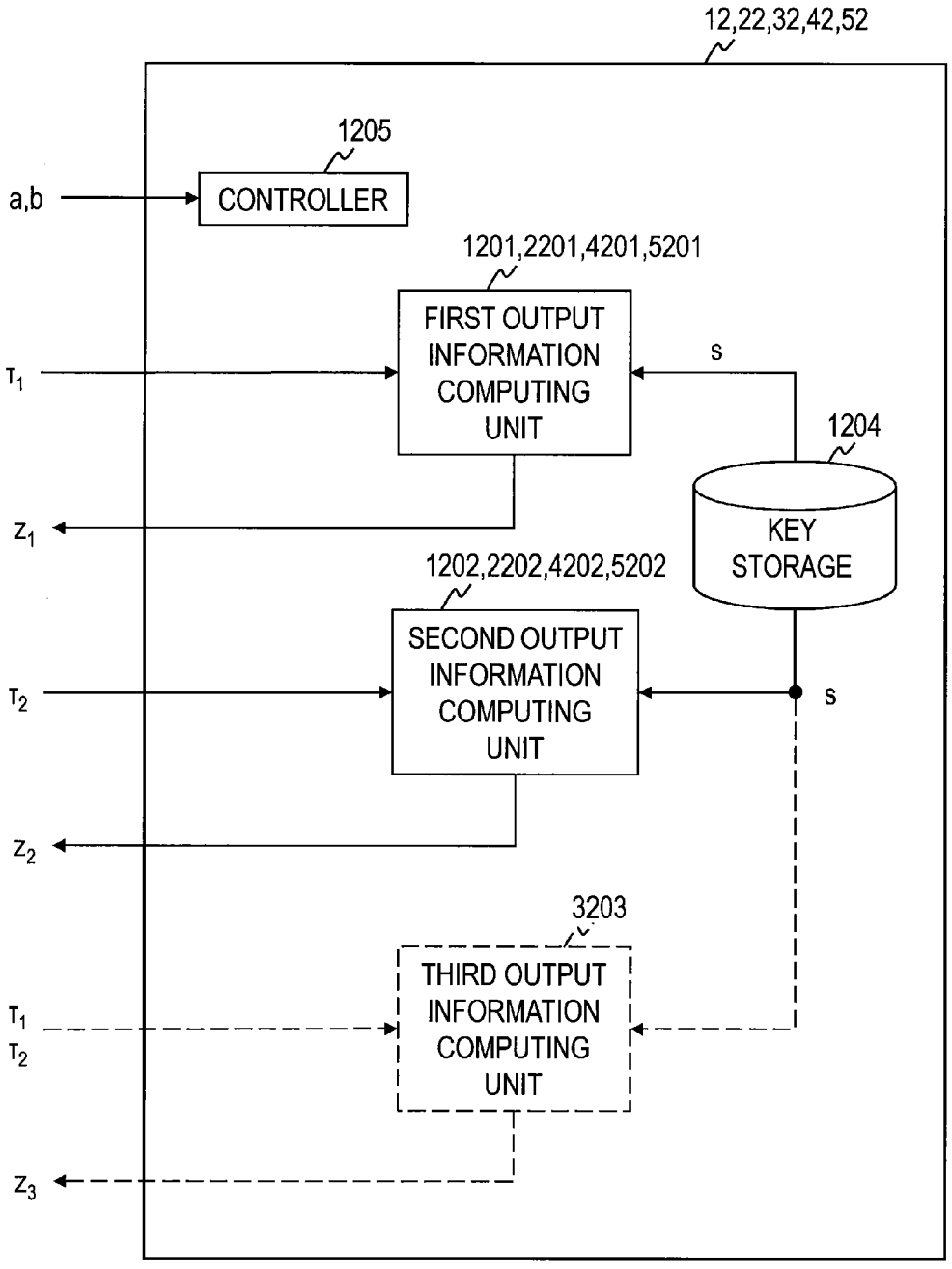

12,22,32,42,52
1205
a,b
CONTROLLER
1201,2201,4201,5201
FIRST OUTPUT INFORMATION COMPUTING UNIT
s
T1
1204
KEY STORAGE
Z1
1202,2202,4202,5202
SECOND OUTPUT INFORMATION COMPUTING UNIT
T2
s
Z2
3203
THIRD OUTPUT INFORMATION COMPUTING UNIT
T1
T2
Z3

FIG.4

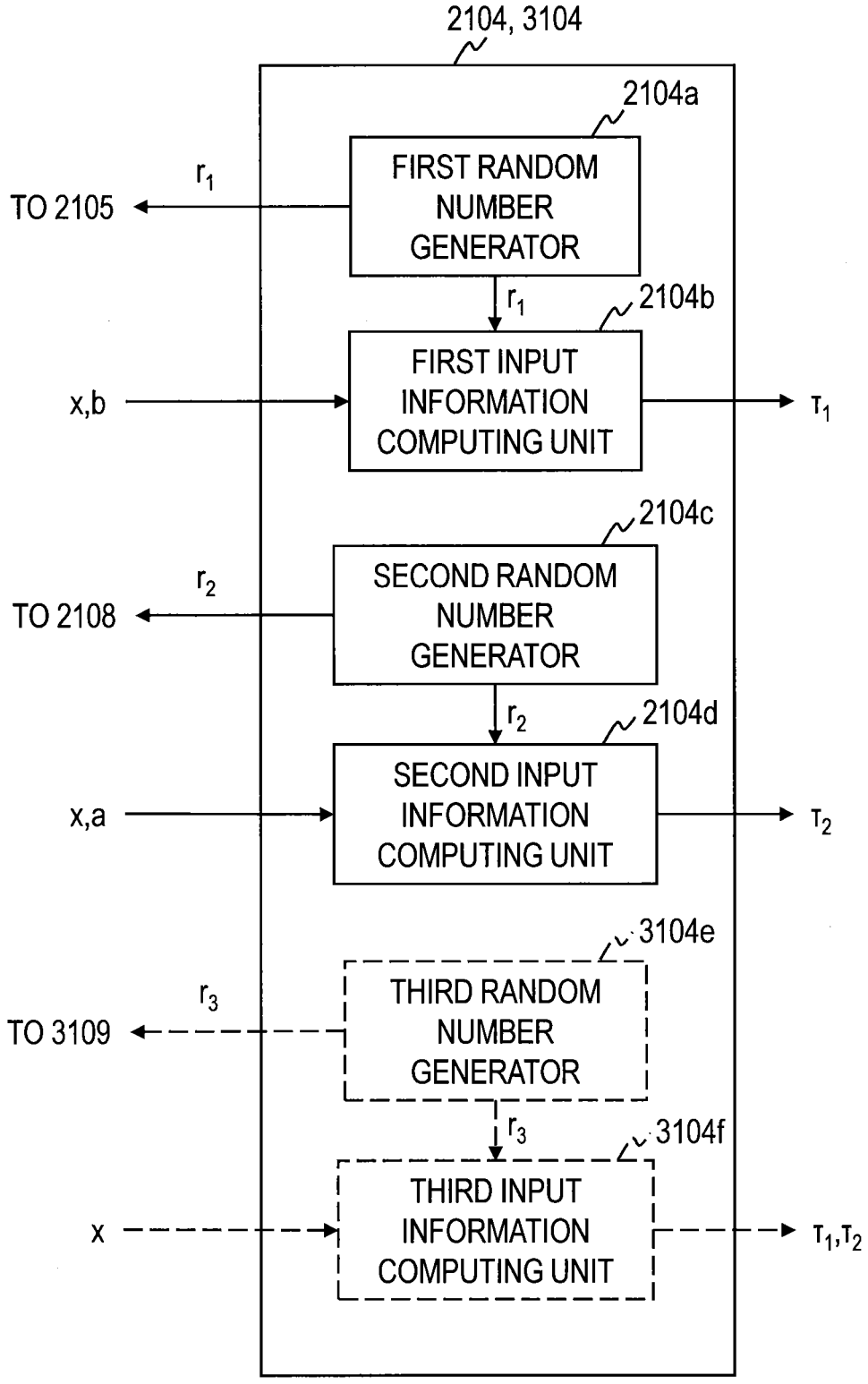

2104, 3104
2104a
FIRST RANDOM NUMBER GENERATOR
r1
TO 2105
r1
2104b
FIRST INPUT INFORMATION COMPUTING UNIT
x,b
T1
2104c
SECOND RANDOM NUMBER GENERATOR
r2
TO 2108
r2
2104d
SECOND INPUT INFORMATION COMPUTING UNIT
x,a
T2
3104e
THIRD RANDOM NUMBER GENERATOR
r3
TO 3109
r3
3104f
THIRD INPUT INFORMATION COMPUTING UNIT
x
T1,T2

FIG.5

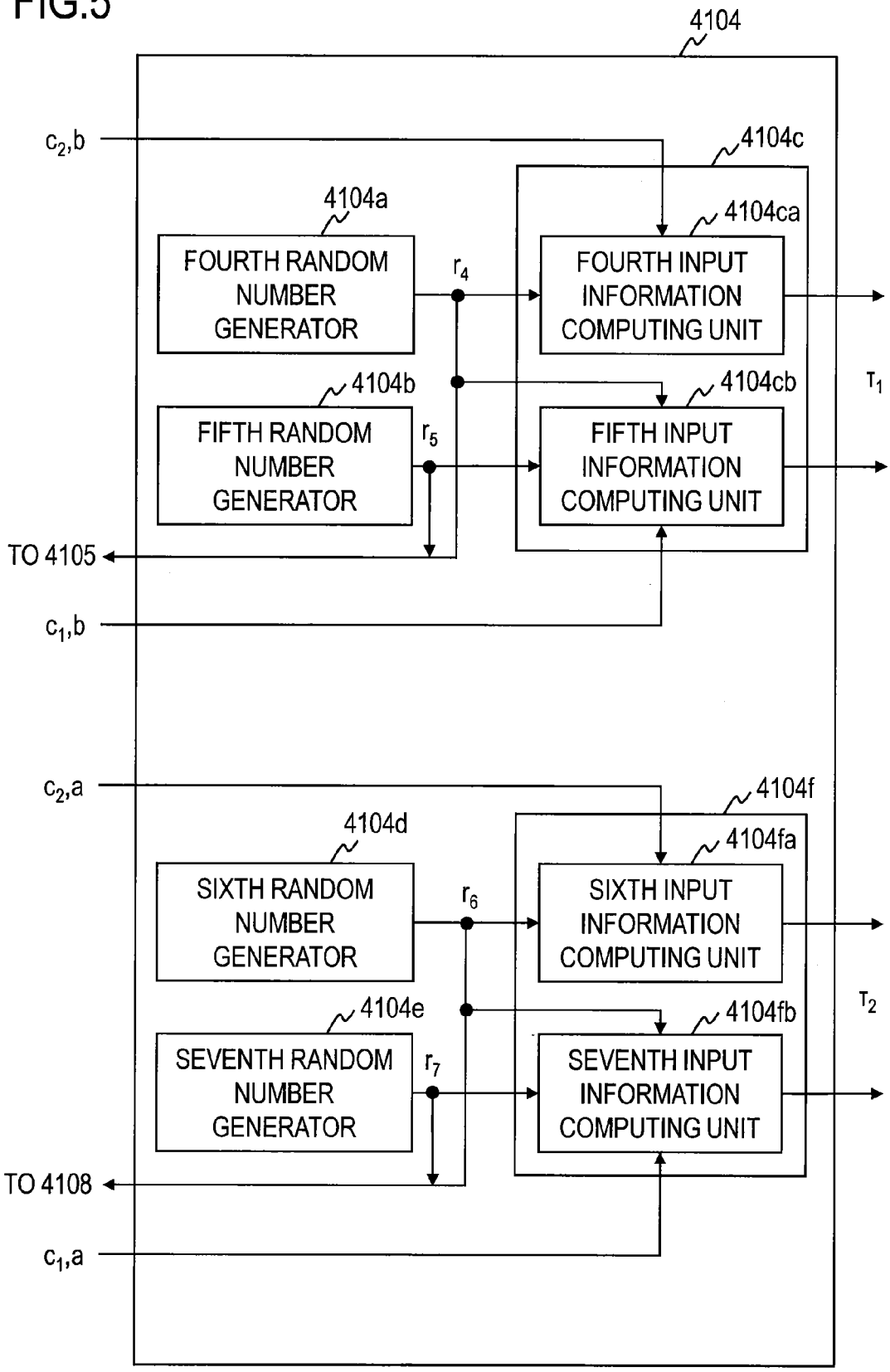

4104
c_2,b
4104c
4104a
4104ca
FOURTH RANDOM NUMBER GENERATOR
r_4
FOURTH INPUT INFORMATION COMPUTING UNIT
4104b
4104cb
T_1
FIFTH RANDOM NUMBER GENERATOR
r_5
FIFTH INPUT INFORMATION COMPUTING UNIT
TO 4105
c_1,b
c_2,a
4104f
4104d
4104fa
SIXTH RANDOM NUMBER GENERATOR
r_6
SIXTH INPUT INFORMATION COMPUTING UNIT
4104e
4104fb
T_2
SEVENTH RANDOM NUMBER GENERATOR
r_7
SEVENTH INPUT INFORMATION COMPUTING UNIT
TO 4108
c_1,a

FIG.6

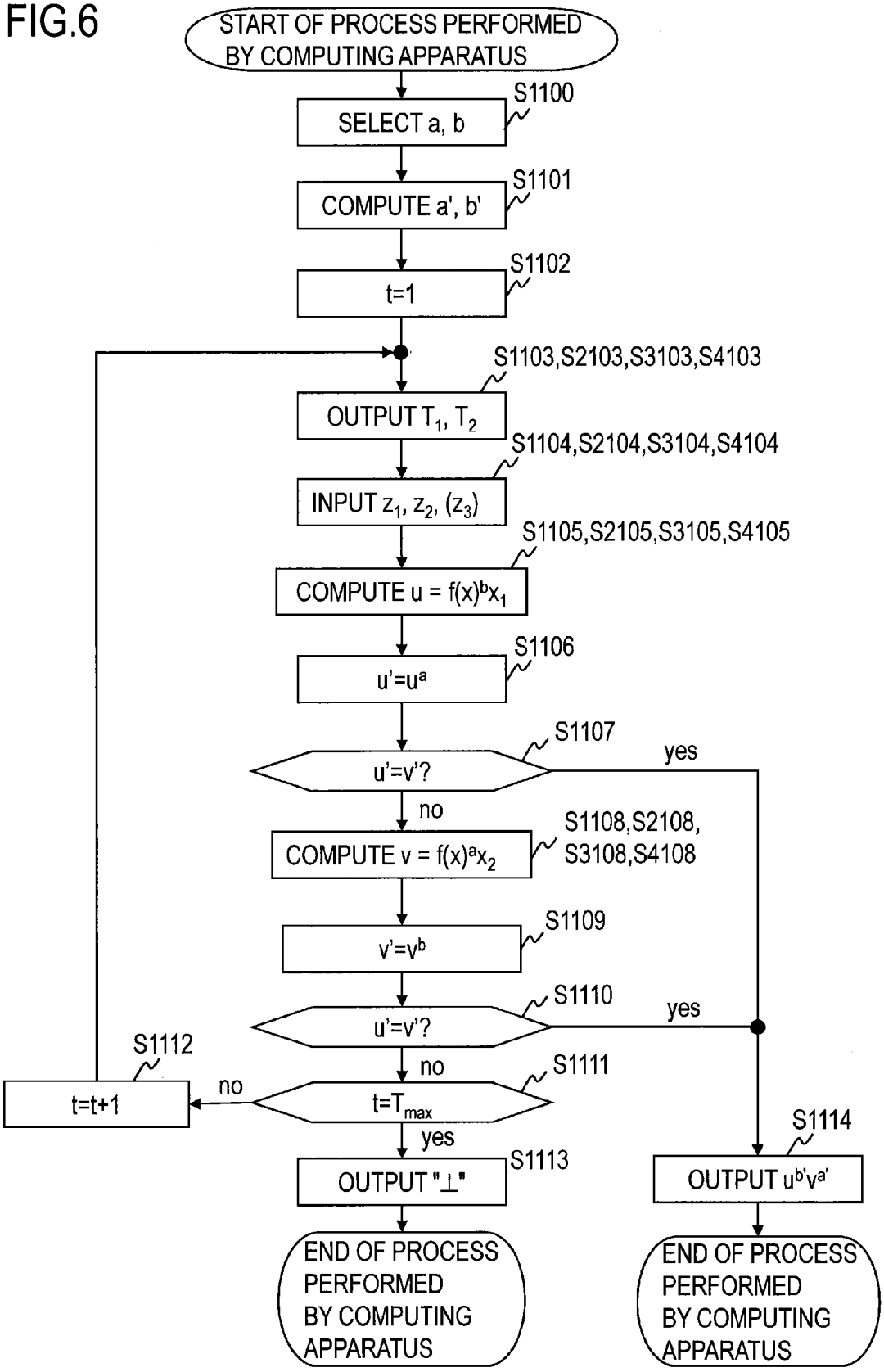

START OF PROCESS PERFORMED BY COMPUTING APPARATUS
S1100
SELECT a, b
S1101
COMPUTE a', b'
S1102
t=1
S1103,S2103,S3103,S4103
OUTPUT T_1, T_2
S1104,S2104,S3104,S4104
INPUT z_1, z_2, (z_3)
S1105,S2105,S3105,S4105
COMPUTE $u = f(x)^b x_1$
S1106
$u'=u^a$
S1107
u'=v'?
yes
no
S1108,S2108, S3108,S4108
COMPUTE $v = f(x)^a x_2$
S1109
$v'=v^b$
S1110
u'=v'?
yes
no
S1111
$t=T_{max}$
no
S1112
t=t+1
yes
S1113
OUTPUT "⊥"
END OF PROCESS PERFORMED BY COMPUTING APPARATUS
S1114
OUTPUT $u^{b'}v^{a'}$
END OF PROCESS PERFORMED BY COMPUTING APPARATUS

FIG.7

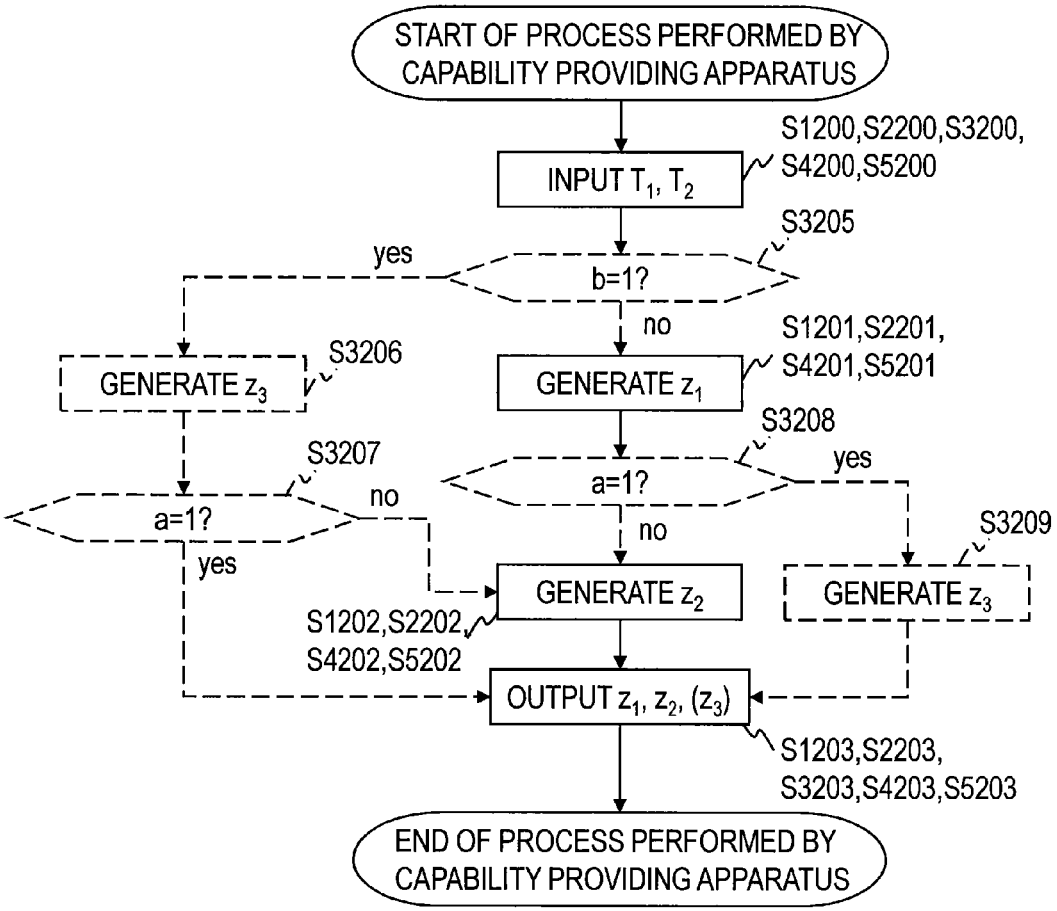

START OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS
S1200,S2200,S3200, S4200,S5200
INPUT T_1, T_2
S3205
b=1?
yes
no
S3206
GENERATE z_3
S1201,S2201, S4201,S5201
GENERATE z_1
S3208
a=1?
yes
no
S3207
a=1?
no
yes
S3209
GENERATE z_2
GENERATE z_3
S1202,S2202, S4202,S5202
OUTPUT z_1, z_2, (z_3)
S1203,S2203, S3203,S4203,S5203
END OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS

FIG.8

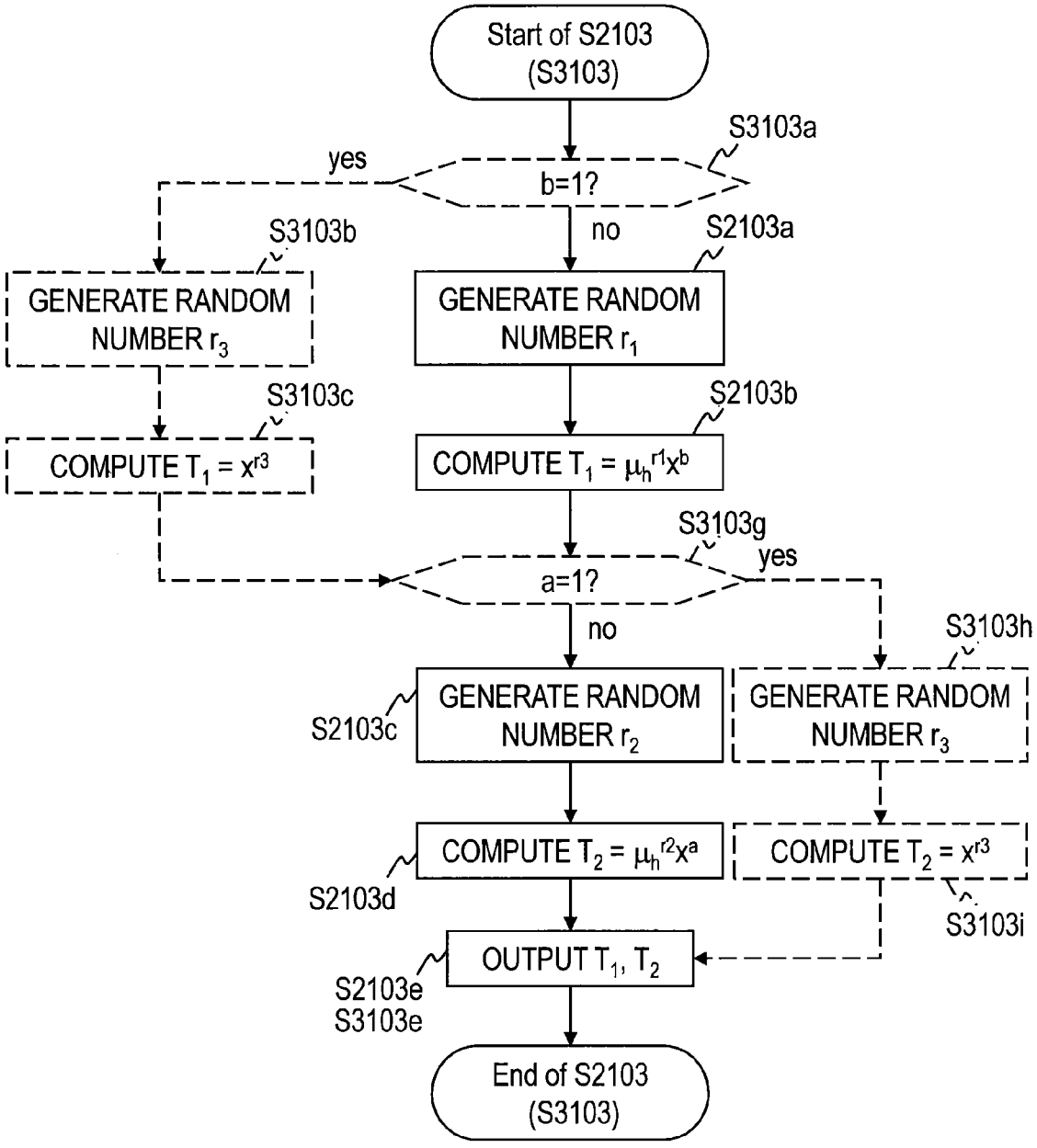

Start of S2103
(S3103)
S3103a
b=1?
yes
no
S3103b
GENERATE RANDOM NUMBER r_3
S2103a
GENERATE RANDOM NUMBER r_1
S3103c
COMPUTE $T_1 = x^{r3}$
S2103b
COMPUTE $T_1 = \mu_h{}^{r1}x^b$
S3103g
a=1?
yes
no
S3103h
S2103c
GENERATE RANDOM NUMBER r_2
GENERATE RANDOM NUMBER r_3
COMPUTE $T_2 = \mu_h{}^{r2}x^a$
S2103d
COMPUTE $T_2 = x^{r3}$
S3103i
OUTPUT T_1, T_2
S2103e
S3103e
End of S2103
(S3103)

FIG.9

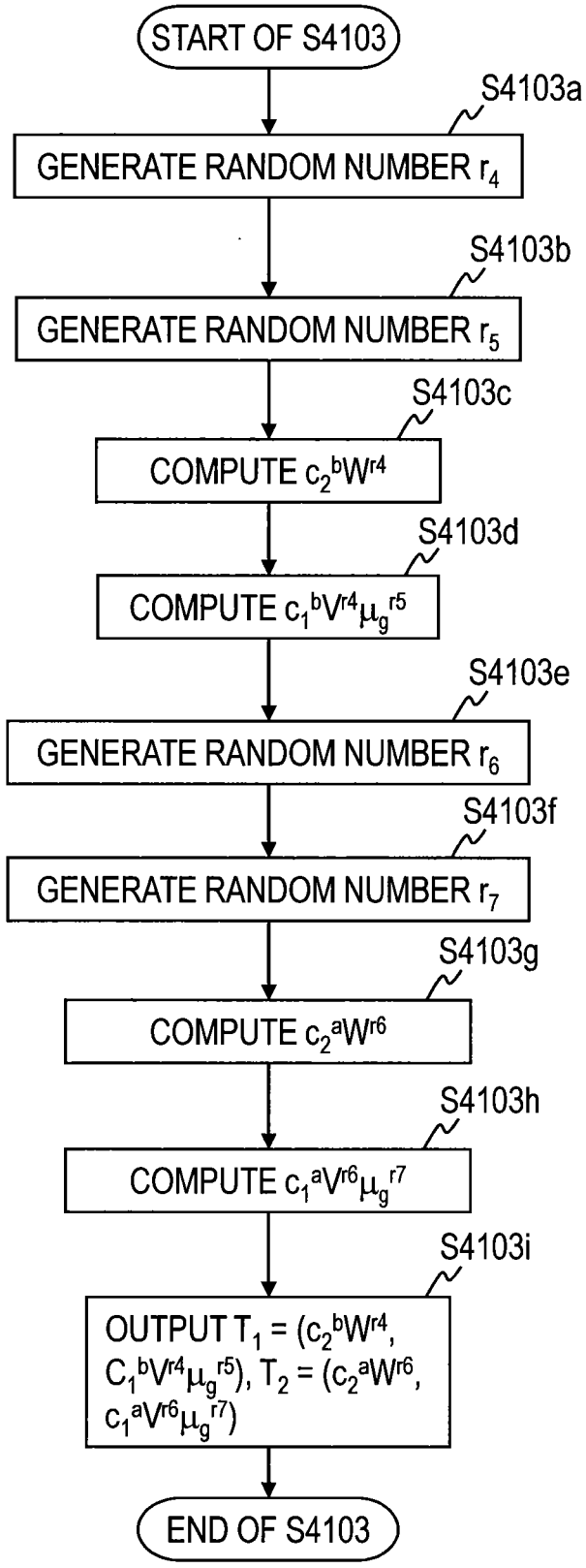

START OF S4103
S4103a
GENERATE RANDOM NUMBER r_4
S4103b
GENERATE RANDOM NUMBER r_5
S4103c
COMPUTE $c_2^{b}W^{r4}$
S4103d
COMPUTE $c_1^{b}V^{r4}\mu_g^{r5}$
S4103e
GENERATE RANDOM NUMBER r_6
S4103f
GENERATE RANDOM NUMBER r_7
S4103g
COMPUTE $c_2^{a}W^{r6}$
S4103h
COMPUTE $c_1^{a}V^{r6}\mu_g^{r7}$
S4103i
OUTPUT $T_1 = (c_2^{b}W^{r4}, C_1^{b}V^{r4}\mu_g^{r5})$, $T_2 = (c_2^{a}W^{r6}, c_1^{a}V^{r6}\mu_g^{r7})$
END OF S4103

FIG.10

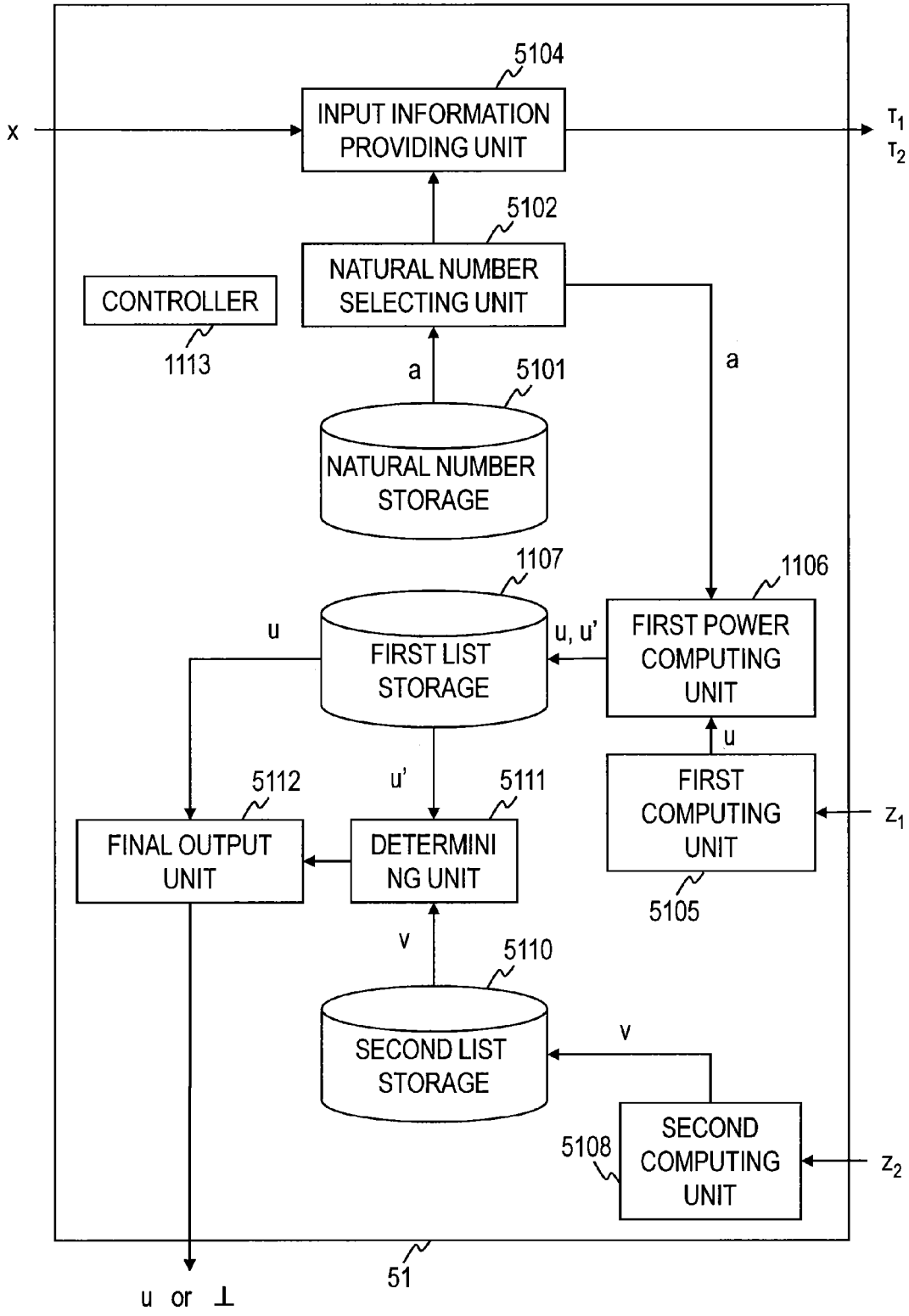

5104
INPUT INFORMATION PROVIDING UNIT
x
τ_1
τ_2
5102
NATURAL NUMBER SELECTING UNIT
CONTROLLER
1113
a
5101
NATURAL NUMBER STORAGE
a
1107
1106
u
FIRST LIST STORAGE
u, u'
FIRST POWER COMPUTING UNIT
u
5112
u'
5111
FINAL OUTPUT UNIT
DETERMINING UNIT
FIRST COMPUTING UNIT
z_1
5105
v
5110
SECOND LIST STORAGE
v
5108
SECOND COMPUTING UNIT
z_2
51
u or ⊥

FIG.11

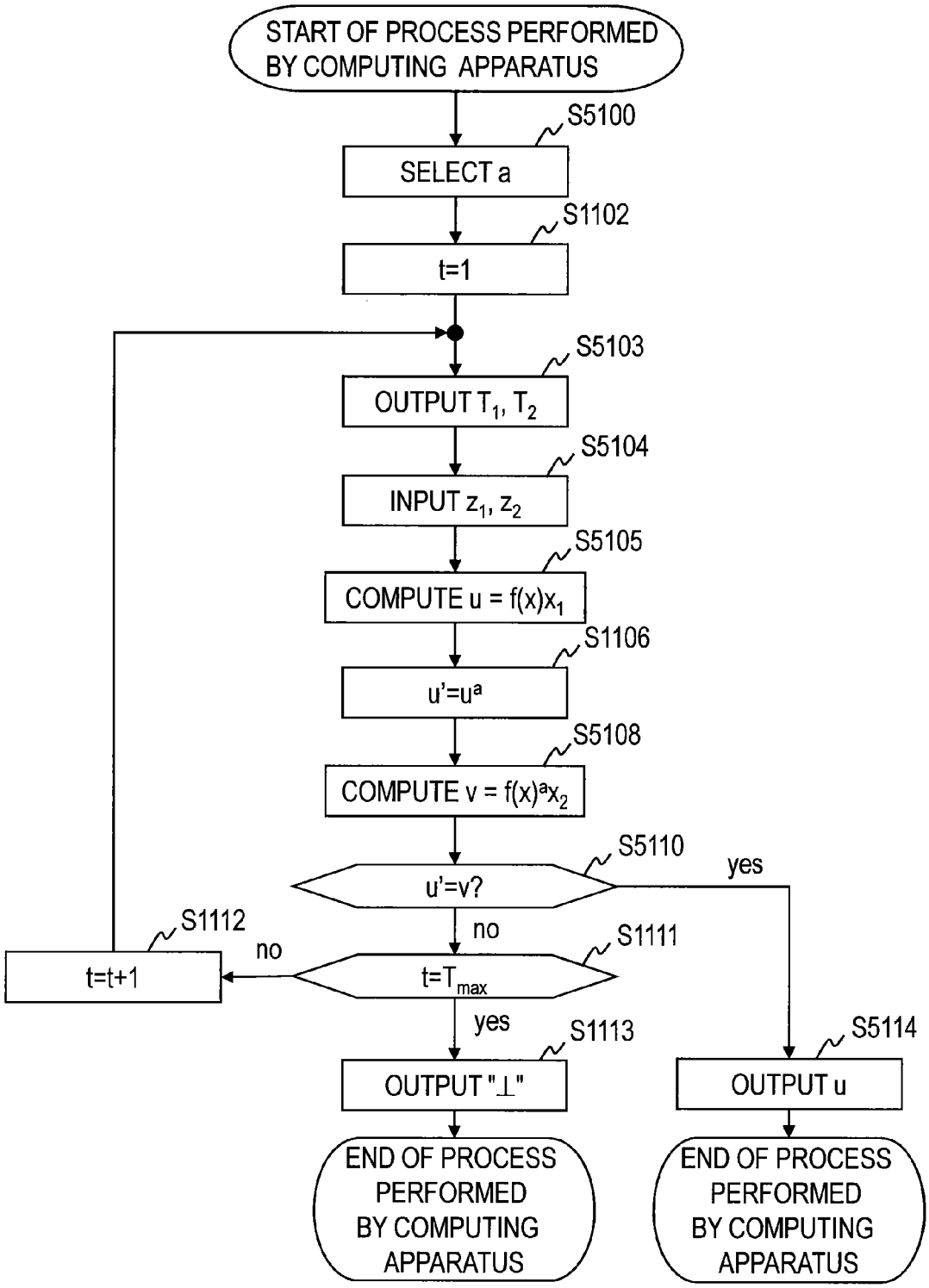

START OF PROCESS PERFORMED BY COMPUTING APPARATUS
S5100
SELECT a
S1102
t=1
S5103
OUTPUT T_1, T_2
S5104
INPUT z_1, z_2
S5105
COMPUTE $u = f(x)x_1$
S1106
$u'=u^a$
S5108
COMPUTE $v = f(x)^a x_2$
S5110
u'=v?
yes
no
S1111
$t=T_{max}$
no
S1112
t=t+1
yes
S1113
OUTPUT "⊥"
S5114
OUTPUT u
END OF PROCESS PERFORMED BY COMPUTING APPARATUS
END OF PROCESS PERFORMED BY COMPUTING APPARATUS

FIG.12

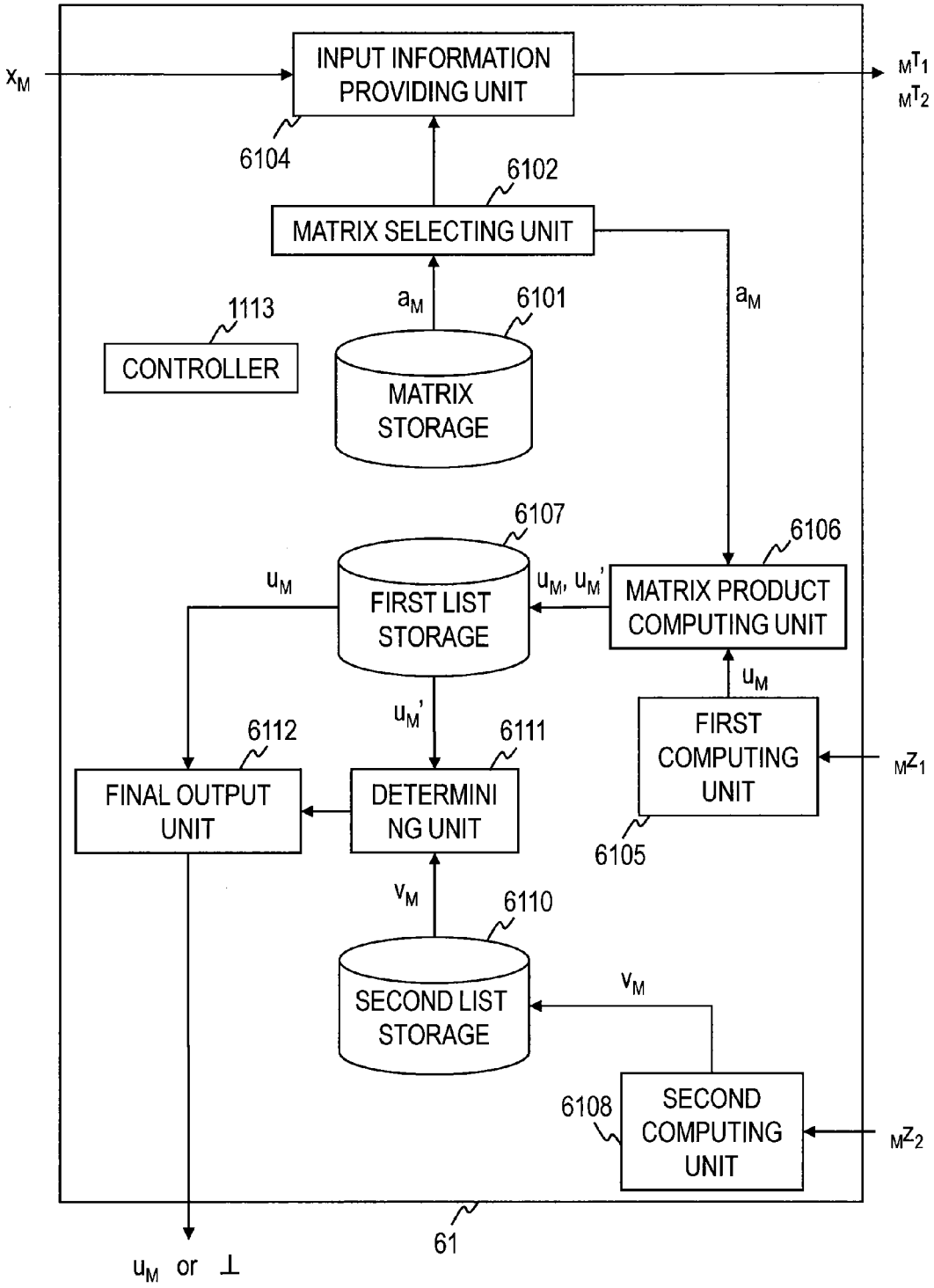

x_M
INPUT INFORMATION PROVIDING UNIT
$_MT_1$
$_MT_2$
6104
6102
MATRIX SELECTING UNIT
a_M
a_M
1113
6101
CONTROLLER
MATRIX STORAGE
6107
6106
u_M
u_M, u_M'
FIRST LIST STORAGE
MATRIX PRODUCT COMPUTING UNIT
u_M
u_M'
6112
6111
FIRST COMPUTING UNIT
$_MZ_1$
FINAL OUTPUT UNIT
DETERMINING UNIT
6105
v_M
6110
SECOND LIST STORAGE
v_M
6108
SECOND COMPUTING UNIT
$_MZ_2$
61
u_M or ⊥

FIG.13

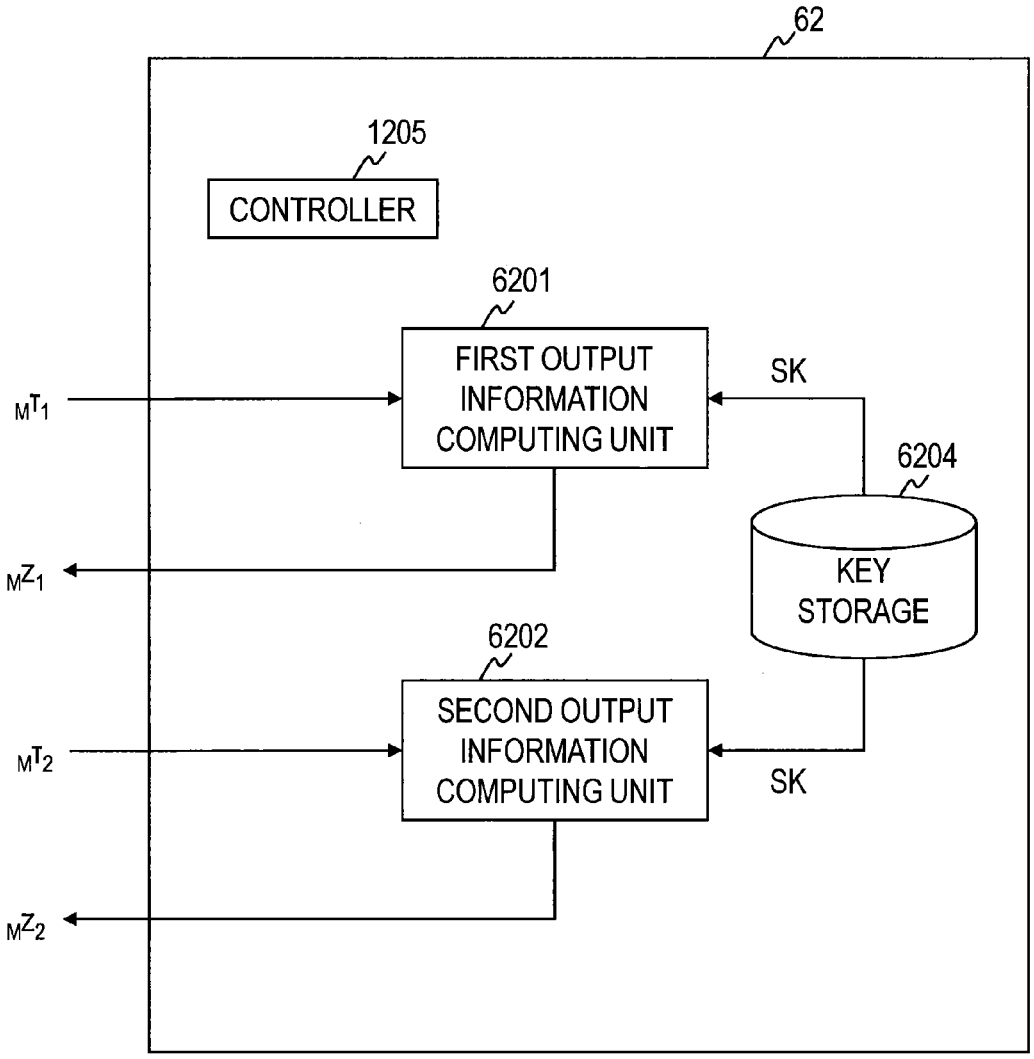

62
1205
CONTROLLER
6201
FIRST OUTPUT INFORMATION COMPUTING UNIT
SK
6204
KEY STORAGE
$_MT_1$
$_MZ_1$
6202
SECOND OUTPUT INFORMATION COMPUTING UNIT
SK
$_MT_2$
$_MZ_2$

FIG.14

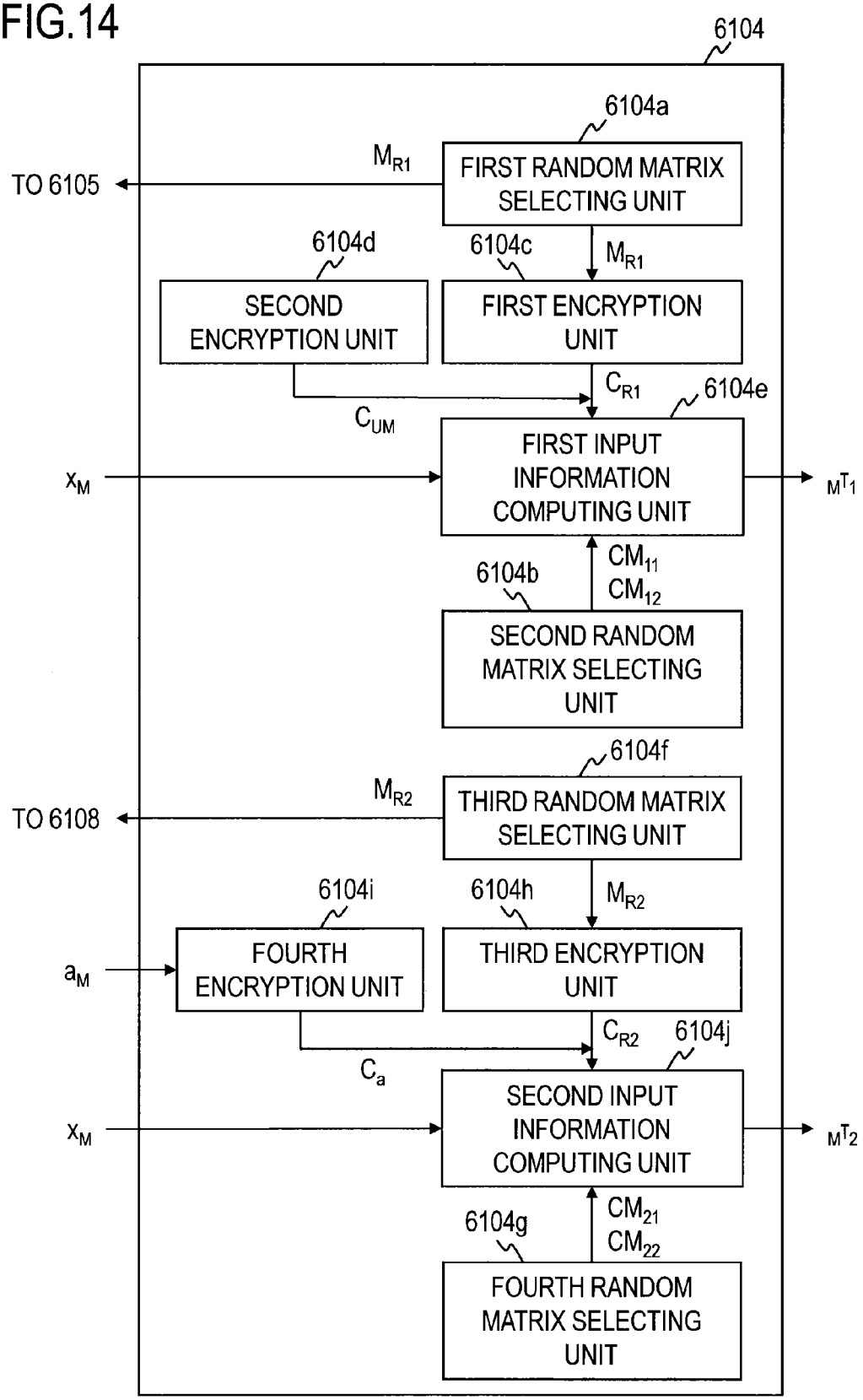

6104
6104a
M_{R1}
TO 6105
FIRST RANDOM MATRIX SELECTING UNIT
6104d
6104c
M_{R1}
SECOND ENCRYPTION UNIT
FIRST ENCRYPTION UNIT
C_{R1}
6104e
C_{UM}
x_M
FIRST INPUT INFORMATION COMPUTING UNIT
M^T_1
CM_{11}
CM_{12}
6104b
SECOND RANDOM MATRIX SELECTING UNIT
6104f
M_{R2}
TO 6108
THIRD RANDOM MATRIX SELECTING UNIT
6104i
6104h
M_{R2}
a_M
FOURTH ENCRYPTION UNIT
THIRD ENCRYPTION UNIT
C_{R2}
6104j
C_a
x_M
SECOND INPUT INFORMATION COMPUTING UNIT
M^T_2
CM_{21}
CM_{22}
6104g
FOURTH RANDOM MATRIX SELECTING UNIT

FIG.15

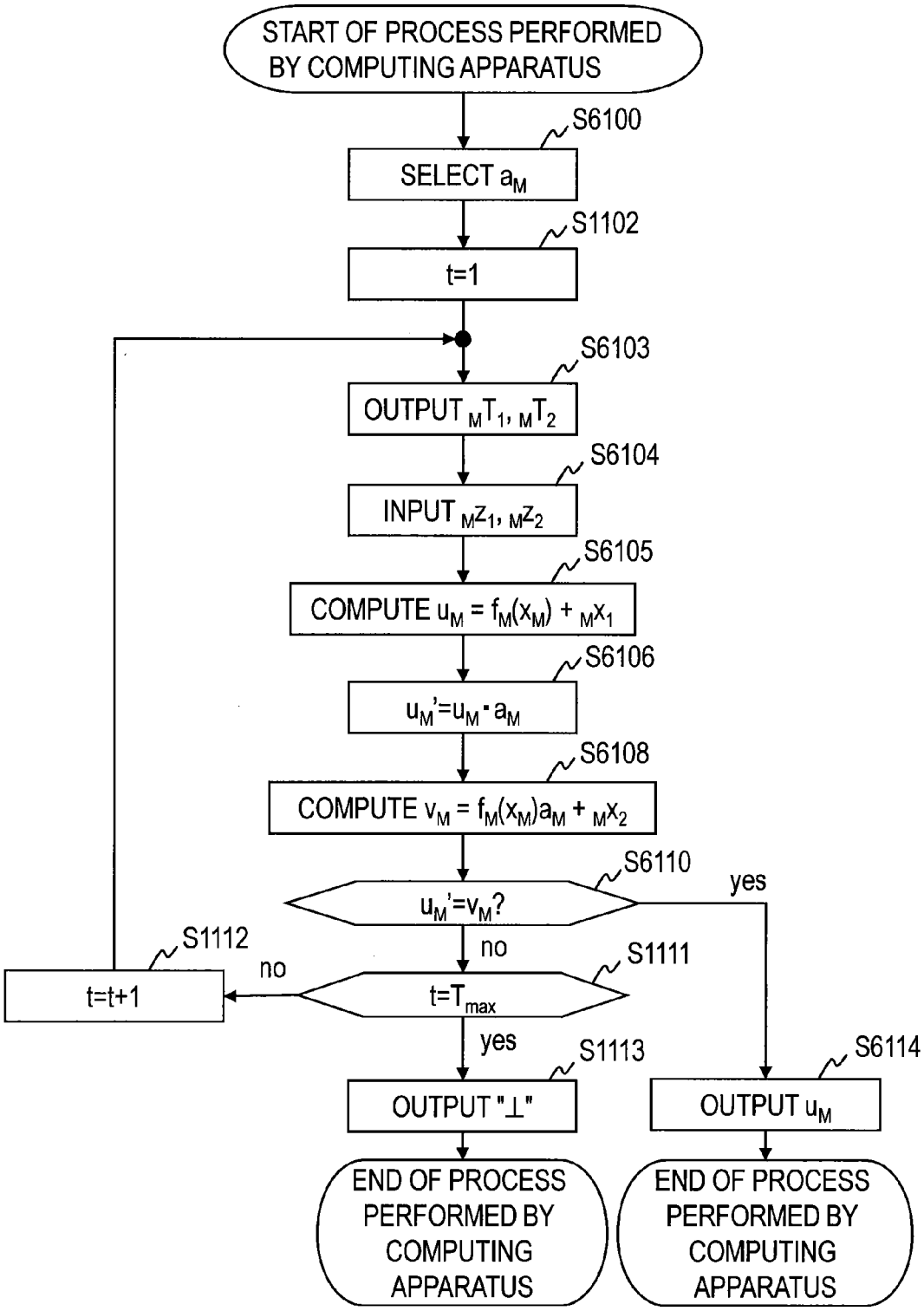

START OF PROCESS PERFORMED BY COMPUTING APPARATUS
S6100
SELECT a_M
S1102
t=1
S6103
OUTPUT ${}_MT_1$, ${}_MT_2$
S6104
INPUT ${}_Mz_1$, ${}_Mz_2$
S6105
COMPUTE $u_M = f_M(x_M) + {}_Mx_1$
S6106
$u_M' = u_M \cdot a_M$
S6108
COMPUTE $v_M = f_M(x_M)a_M + {}_Mx_2$
S6110
$u_M' = v_M$?
yes
no
S1111
$t = T_{max}$
no
S1112
t=t+1
yes
S1113
OUTPUT "⊥"
S6114
OUTPUT u_M
END OF PROCESS PERFORMED BY COMPUTING APPARATUS
END OF PROCESS PERFORMED BY COMPUTING APPARATUS

FIG.16

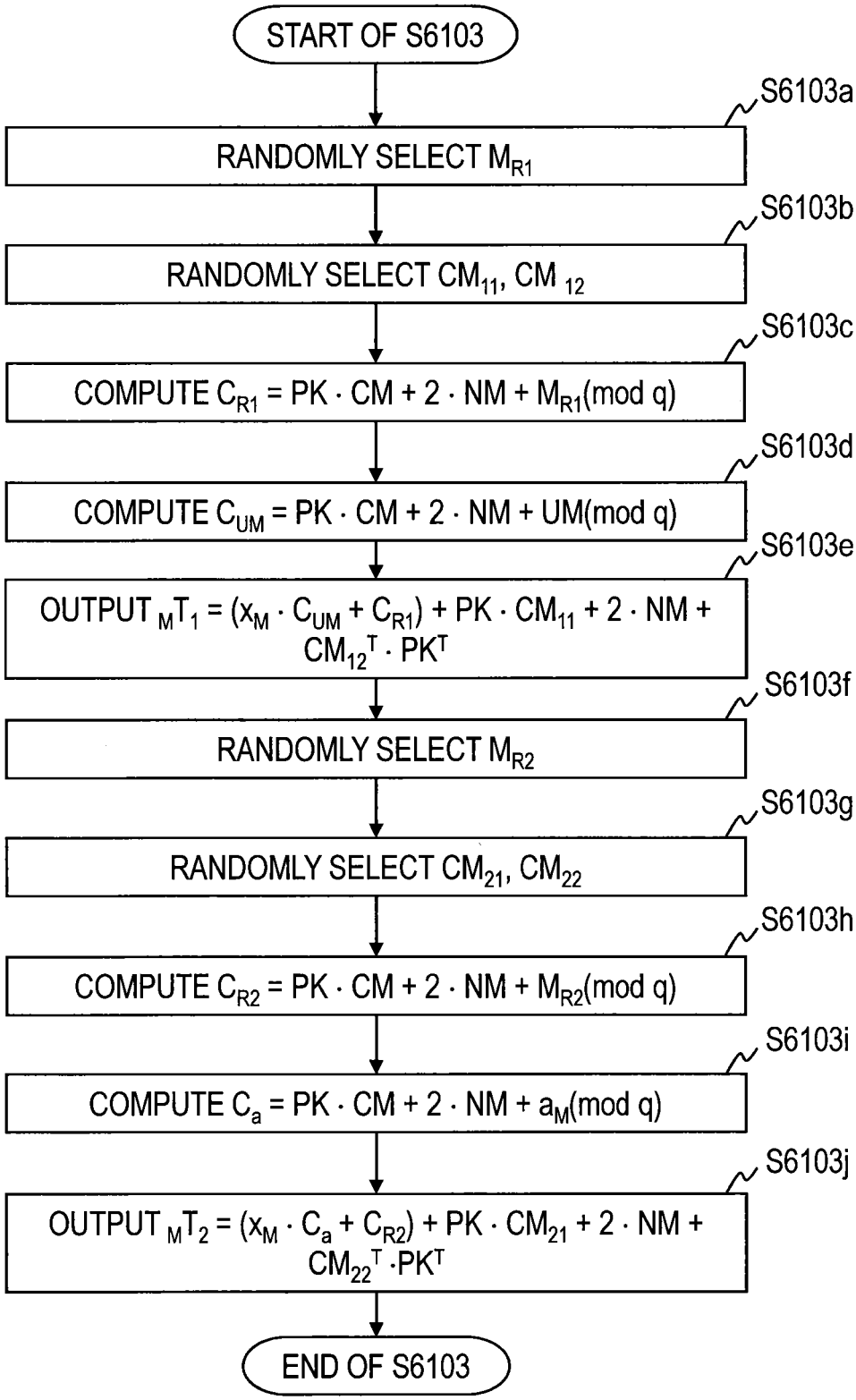

START OF S6103
S6103a
RANDOMLY SELECT M_{R1}
S6103b
RANDOMLY SELECT CM_{11}, CM_{12}
S6103c
COMPUTE $C_{R1} = PK \cdot CM + 2 \cdot NM + M_{R1}(\text{mod } q)$
S6103d
COMPUTE $C_{UM} = PK \cdot CM + 2 \cdot NM + UM(\text{mod } q)$
S6103e
OUTPUT $_MT_1 = (x_M \cdot C_{UM} + C_{R1}) + PK \cdot CM_{11} + 2 \cdot NM + CM_{12}^T \cdot PK^T$
S6103f
RANDOMLY SELECT M_{R2}
S6103g
RANDOMLY SELECT CM_{21}, CM_{22}
S6103h
COMPUTE $C_{R2} = PK \cdot CM + 2 \cdot NM + M_{R2}(\text{mod } q)$
S6103i
COMPUTE $C_a = PK \cdot CM + 2 \cdot NM + a_M(\text{mod } q)$
S6103j
OUTPUT $_MT_2 = (x_M \cdot C_a + C_{R2}) + PK \cdot CM_{21} + 2 \cdot NM + CM_{22}^T \cdot PK^T$
END OF S6103

FIG.17

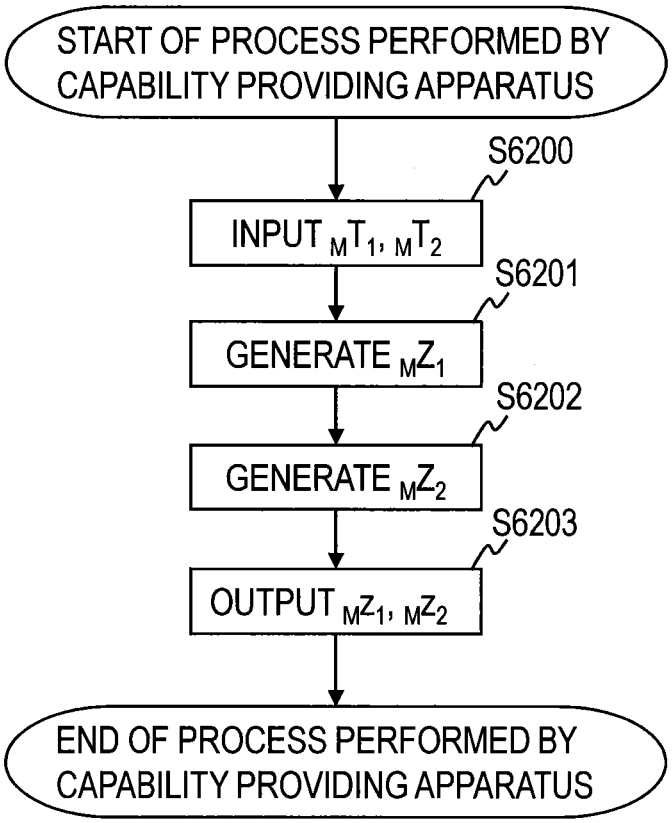

START OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS
S6200
INPUT $_{M}T_{1}$, $_{M}T_{2}$
S6201
GENERATE $_{M}Z_{1}$
S6202
GENERATE $_{M}Z_{2}$
S6203
OUTPUT $_{M}Z_{1}$, $_{M}Z_{2}$
END OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS

FIG.18

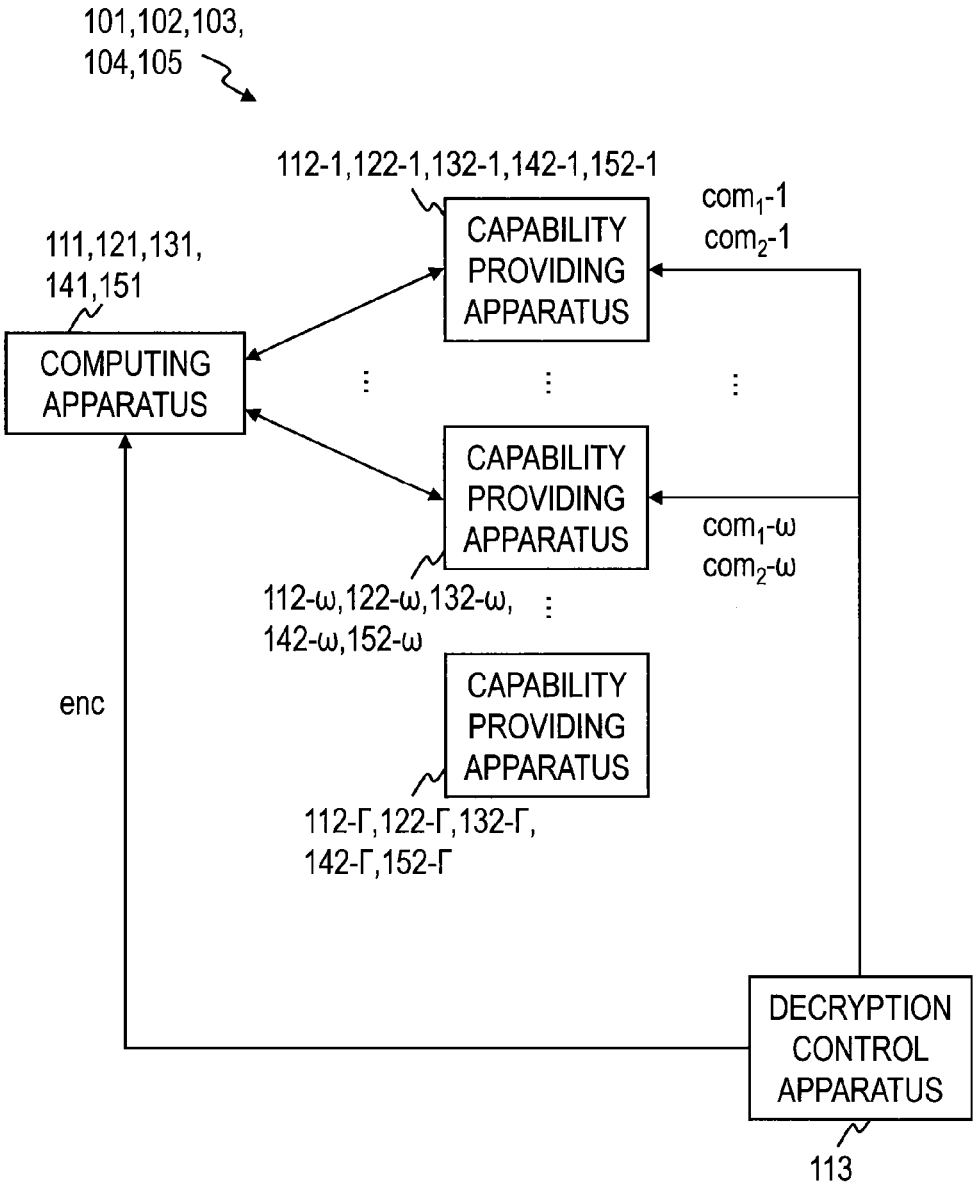

101,102,103,
104,105
112-1,122-1,132-1,142-1,152-1
CAPABILITY PROVIDING APPARATUS
com1-1
com2-1
111,121,131,
141,151
COMPUTING APPARATUS
CAPABILITY PROVIDING APPARATUS
com1-ω
com2-ω
112-ω,122-ω,132-ω,
142-ω,152-ω
CAPABILITY PROVIDING APPARATUS
enc
112-Γ,122-Γ,132-Γ,
142-Γ,152-Γ
DECRYPTION CONTROL APPARATUS
113

FIG.19

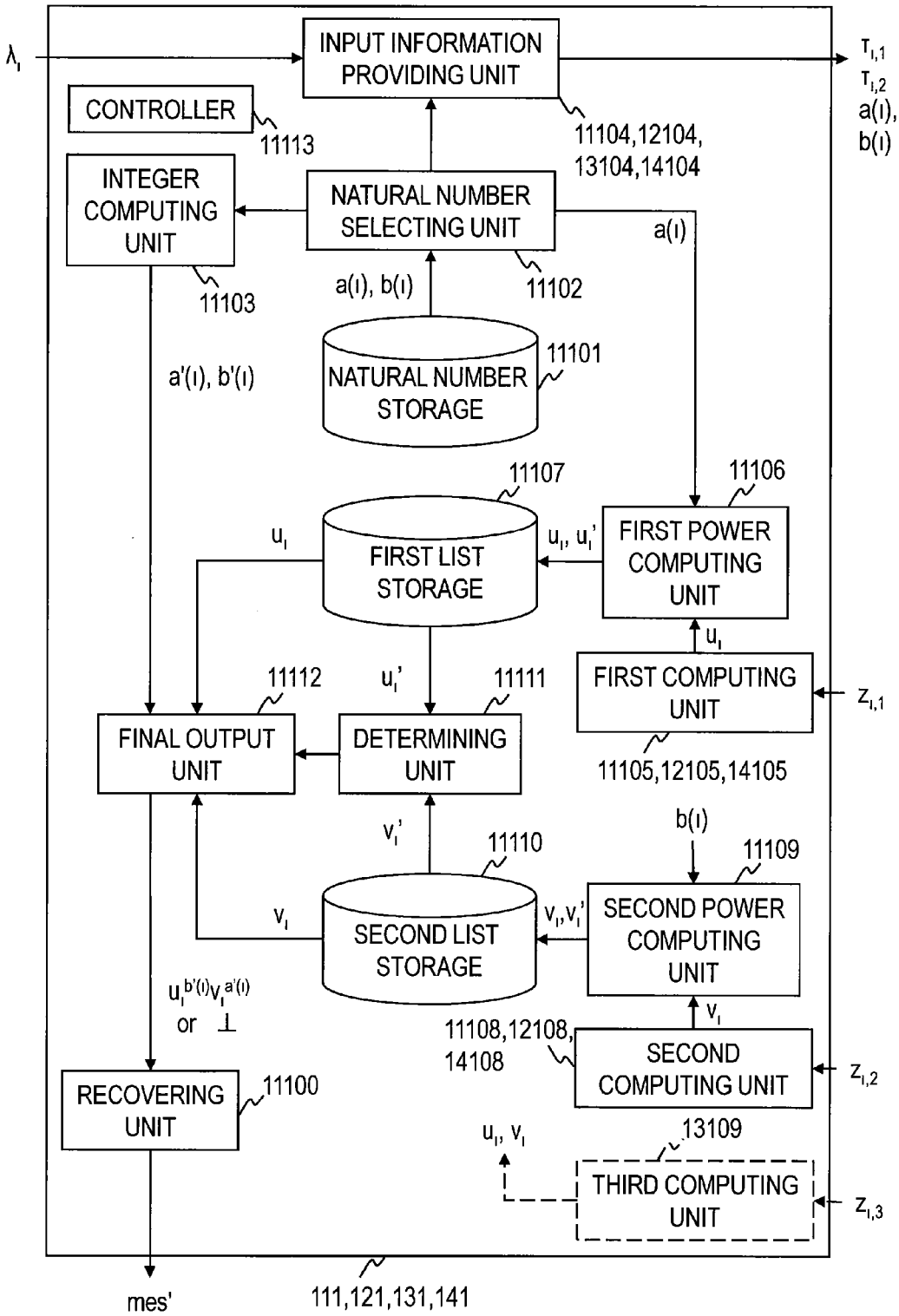

λ_I
INPUT INFORMATION PROVIDING UNIT
τ_I,1
τ_I,2
a(I),
b(I)
CONTROLLER
11113
11104,12104,
13104,14104
INTEGER COMPUTING UNIT
NATURAL NUMBER SELECTING UNIT
a(I)
11103
a(I), b(I)
11102
a'(I), b'(I)
NATURAL NUMBER STORAGE
11101
11107
11106
u_I
FIRST LIST STORAGE
u_I, u_I'
FIRST POWER COMPUTING UNIT
u_I
11112
u_I'
11111
FIRST COMPUTING UNIT
z_I,1
FINAL OUTPUT UNIT
DETERMINING UNIT
11105,12105,14105
v_I'
11110
b(I)
11109
v_I
SECOND LIST STORAGE
v_I,v_I'
SECOND POWER COMPUTING UNIT
u_I^b'(I)v_I^a'(I)
or ⊥
v_I
11108,12108,
14108
SECOND COMPUTING UNIT
z_I,2
RECOVERING UNIT
11100
u_I, v_I
13109
THIRD COMPUTING UNIT
z_I,3
mes'
111,121,131,141

FIG.20

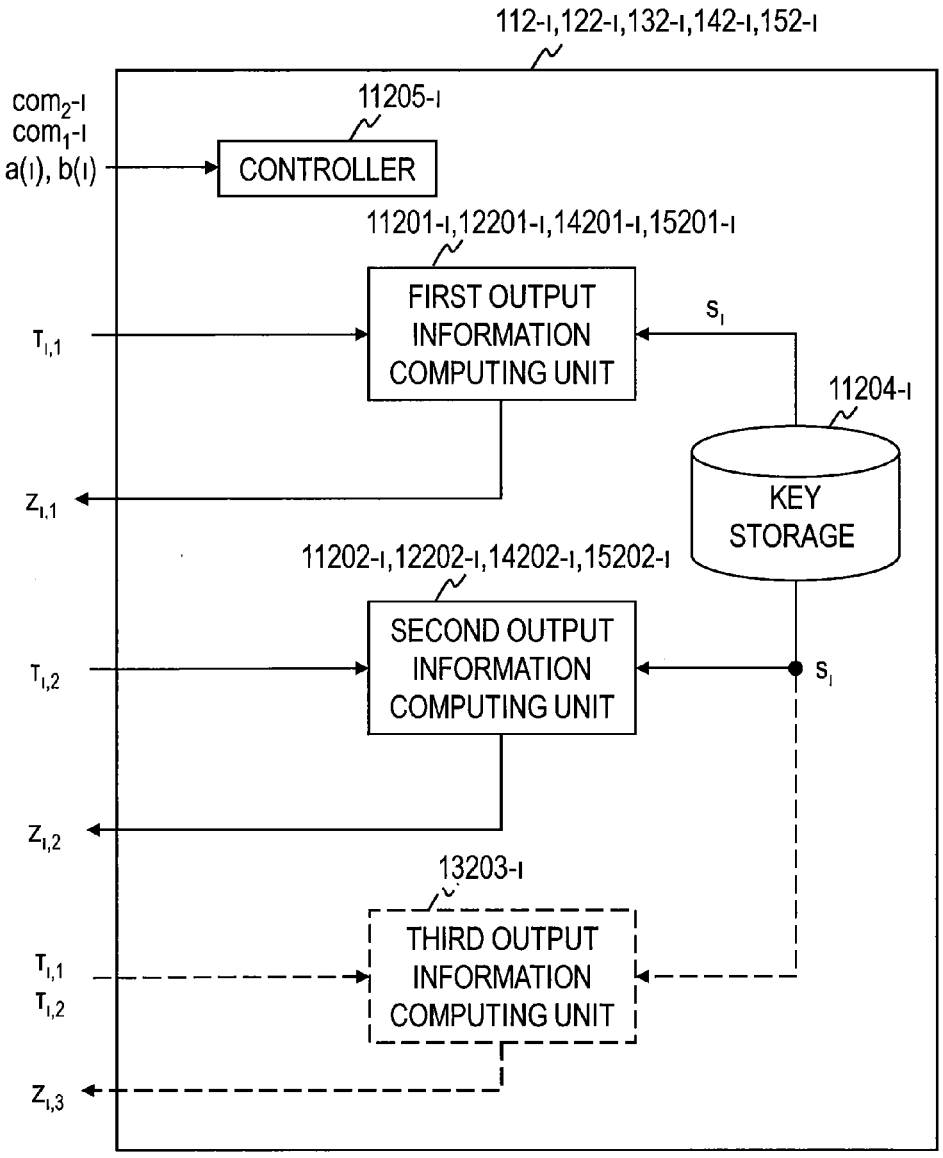

112-ι,122-ι,132-ι,142-ι,152-ι
com2-ι
com1-ι
a(ι), b(ι)
11205-ι
CONTROLLER
11201-ι,12201-ι,14201-ι,15201-ι
FIRST OUTPUT INFORMATION COMPUTING UNIT
sι
Tι,1
11204-ι
KEY STORAGE
Zι,1
11202-ι,12202-ι,14202-ι,15202-ι
SECOND OUTPUT INFORMATION COMPUTING UNIT
Tι,2
sι
Zι,2
13203-ι
THIRD OUTPUT INFORMATION COMPUTING UNIT
Tι,1
Tι,2
Zι,3

FIG.21

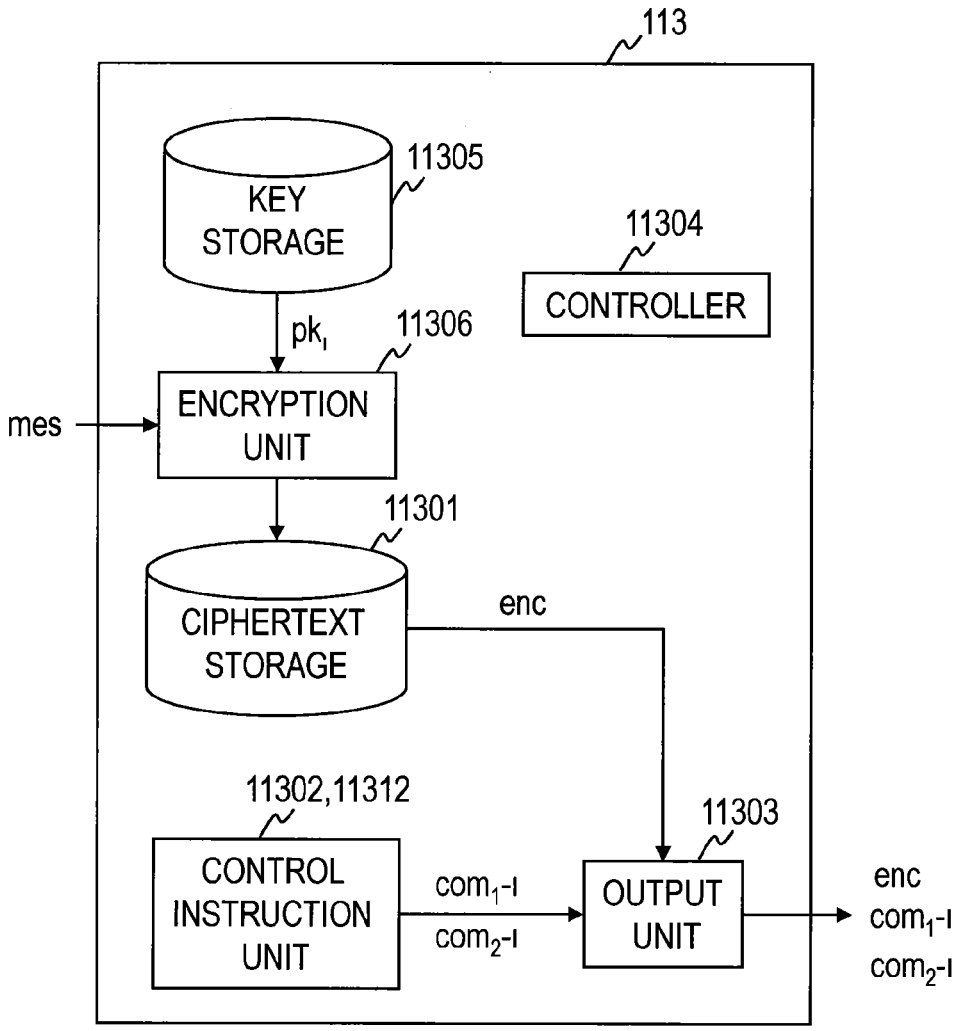

113
11305
KEY STORAGE
11304
CONTROLLER
pk_i
11306
ENCRYPTION UNIT
mes
11301
CIPHERTEXT STORAGE
enc
11302,11312
CONTROL INSTRUCTION UNIT
com_1-I
com_2-I
11303
OUTPUT UNIT
enc
com_1-I
com_2-I

FIG.22

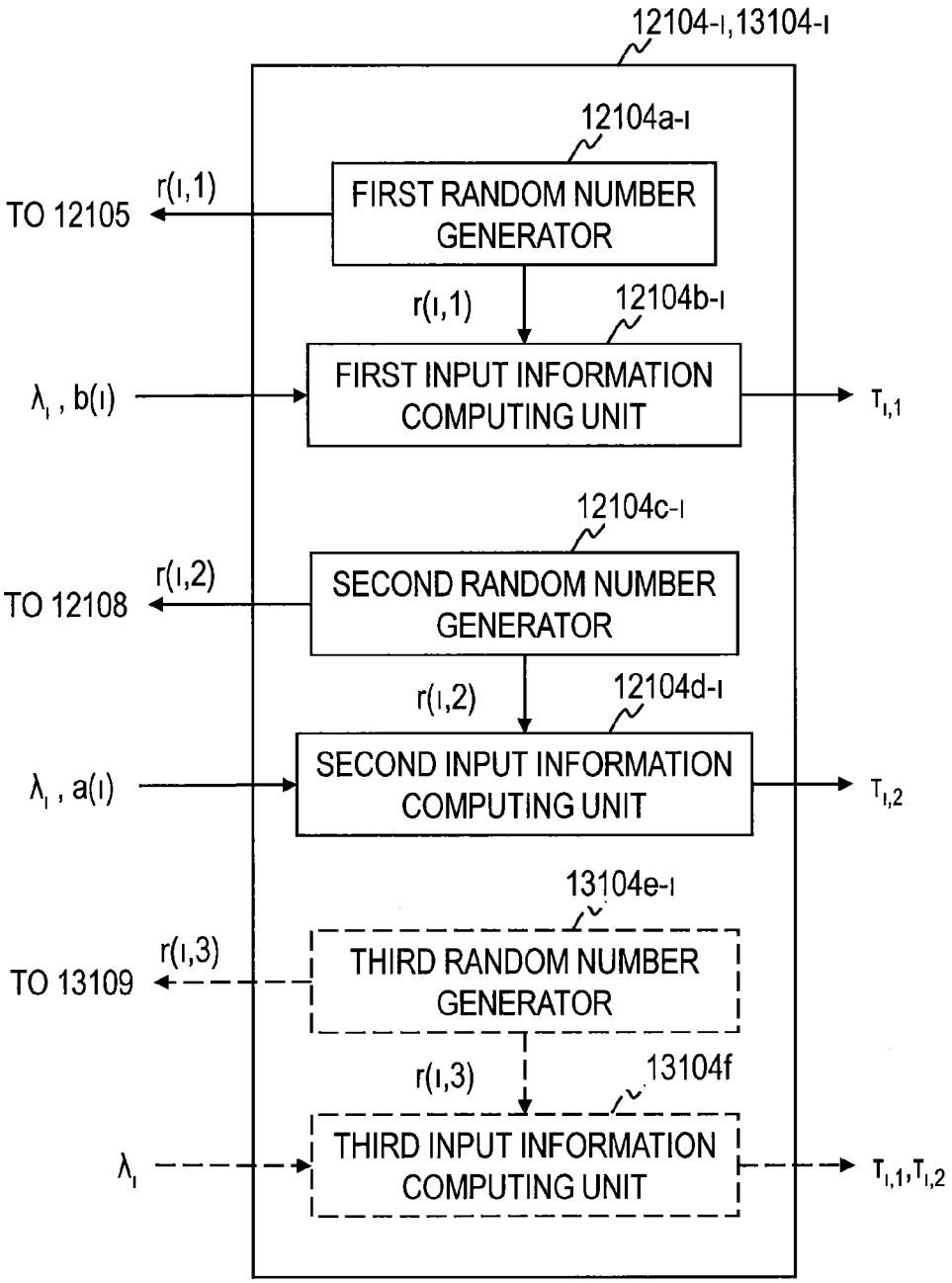

12104-I,13104-I
12104a-I
FIRST RANDOM NUMBER GENERATOR
r(I,1)
TO 12105
r(I,1)
12104b-I
FIRST INPUT INFORMATION COMPUTING UNIT
λI , b(I)
TI,1
12104c-I
SECOND RANDOM NUMBER GENERATOR
r(I,2)
TO 12108
r(I,2)
12104d-I
SECOND INPUT INFORMATION COMPUTING UNIT
λI , a(I)
TI,2
13104e-I
THIRD RANDOM NUMBER GENERATOR
r(I,3)
TO 13109
r(I,3)
13104f
THIRD INPUT INFORMATION COMPUTING UNIT
λI
TI,1,TI,2

FIG.23

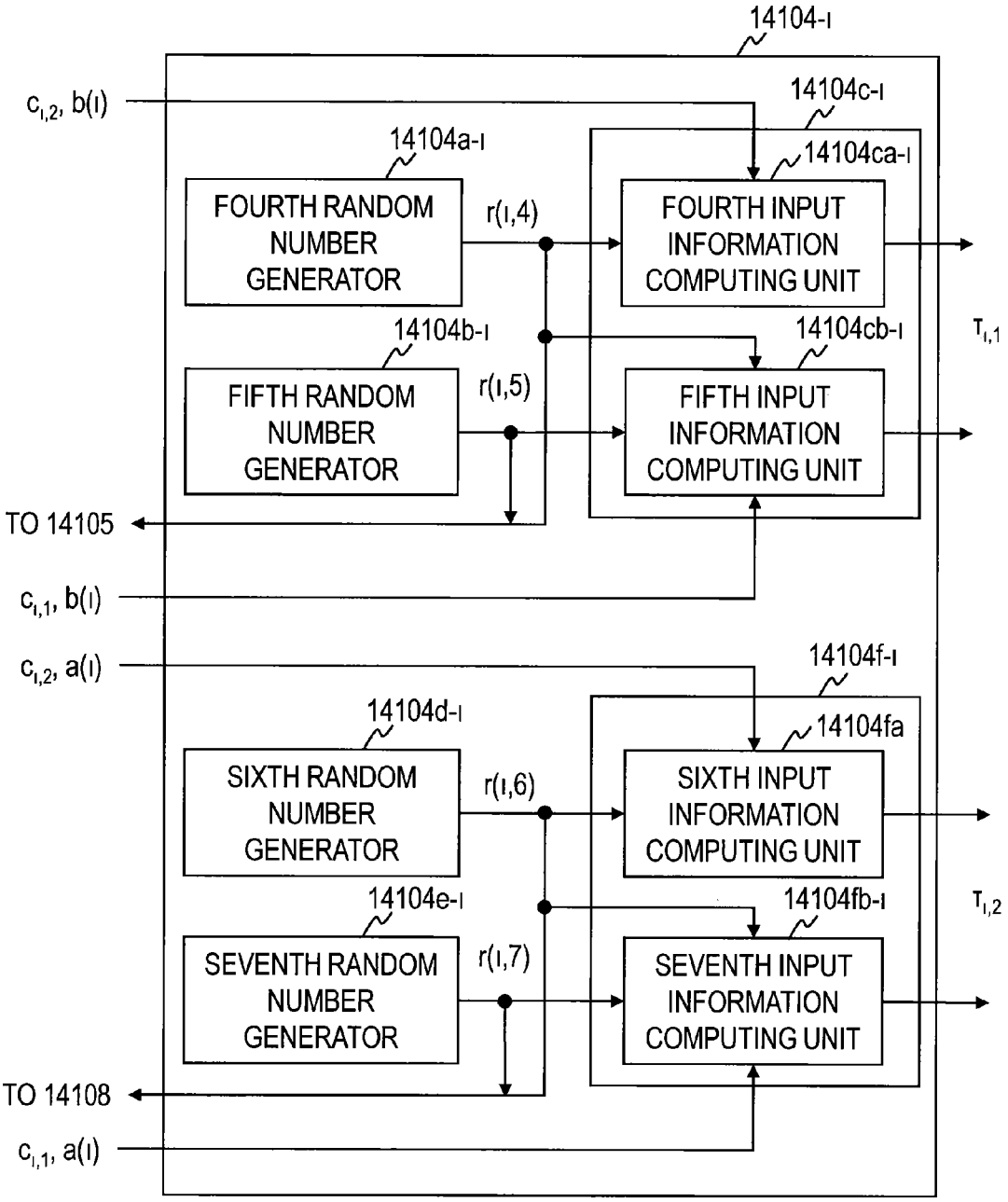

14104-I
$c_{I,2}$, b(I)
14104c-I
14104a-I
14104ca-I
FOURTH RANDOM NUMBER GENERATOR
r(I,4)
FOURTH INPUT INFORMATION COMPUTING UNIT
14104b-I
14104cb-I
$T_{I,1}$
FIFTH RANDOM NUMBER GENERATOR
r(I,5)
FIFTH INPUT INFORMATION COMPUTING UNIT
TO 14105
$c_{I,1}$, b(I)
$c_{I,2}$, a(I)
14104f-I
14104d-I
14104fa
SIXTH RANDOM NUMBER GENERATOR
r(I,6)
SIXTH INPUT INFORMATION COMPUTING UNIT
14104e-I
14104fb-I
$T_{I,2}$
SEVENTH RANDOM NUMBER GENERATOR
r(I,7)
SEVENTH INPUT INFORMATION COMPUTING UNIT
TO 14108
$c_{I,1}$, a(I)

FIG.24

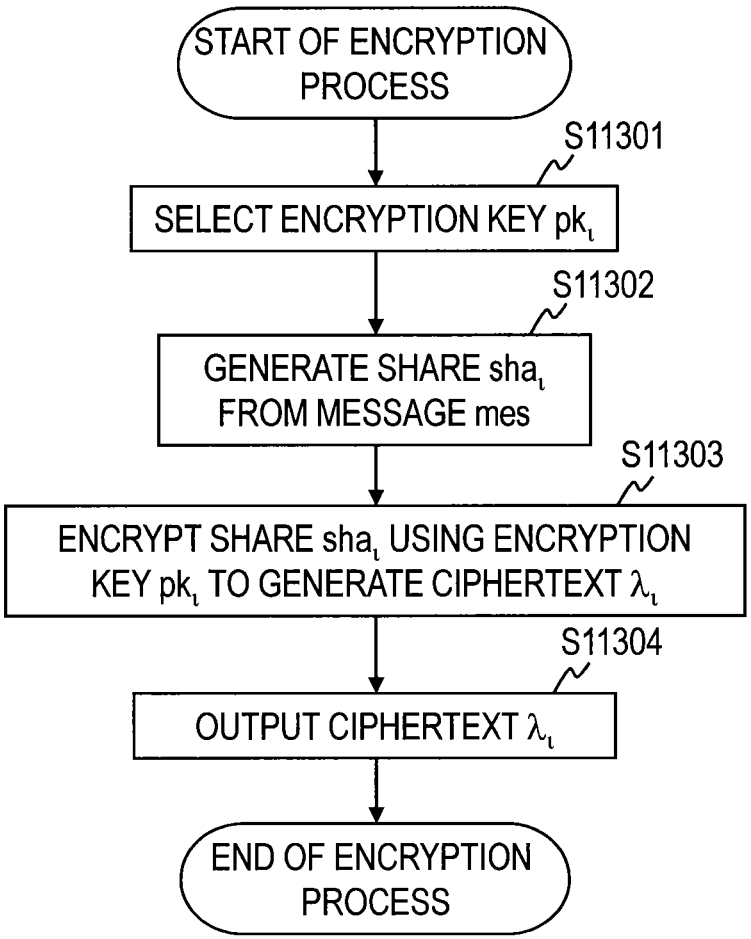

START OF ENCRYPTION PROCESS
S11301
SELECT ENCRYPTION KEY pk_ι
S11302
GENERATE SHARE sha_ι FROM MESSAGE mes
S11303
ENCRYPT SHARE sha_ι USING ENCRYPTION KEY pk_ι TO GENERATE CIPHERTEXT λ_ι
S11304
OUTPUT CIPHERTEXT λ_ι
END OF ENCRYPTION PROCESS

FIG.25

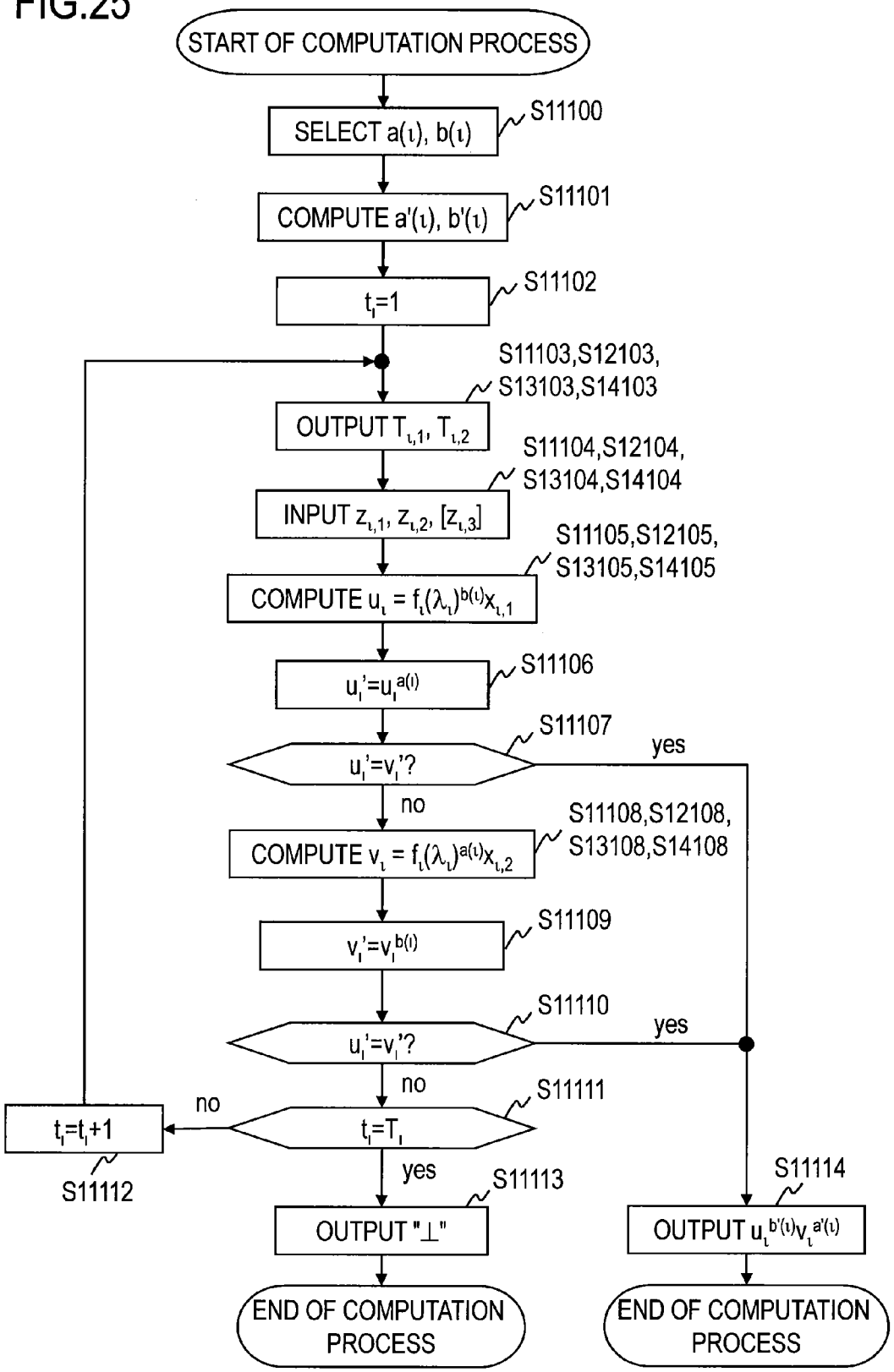

START OF COMPUTATION PROCESS
SELECT a(ι), b(ι)
S11100
COMPUTE a'(ι), b'(ι)
S11101
$t_ι$=1
S11102
S11103,S12103,
S13103,S14103
OUTPUT $T_{ι,1}$, $T_{ι,2}$
S11104,S12104,
S13104,S14104
INPUT $z_{ι,1}$, $z_{ι,2}$, [$z_{ι,3}$]
S11105,S12105,
S13105,S14105
COMPUTE $u_ι = f_ι(λ_ι)^{b(ι)}x_{ι,1}$
$u_ι'=u_ι^{a(ι)}$
S11106
S11107
$u_ι'=v_ι'$?
yes
no
S11108,S12108,
S13108,S14108
COMPUTE $v_ι = f_ι(λ_ι)^{a(ι)}x_{ι,2}$
$v_ι'=v_ι^{b(ι)}$
S11109
S11110
$u_ι'=v_ι'$?
yes
no
S11111
$t_ι=T_ι$
no
$t_ι=t_ι+1$
S11112
yes
S11113
OUTPUT "⊥"
END OF COMPUTATION PROCESS
S11114
OUTPUT $u_ι^{b'(ι)}v_ι^{a'(ι)}$
END OF COMPUTATION PROCESS

FIG.26

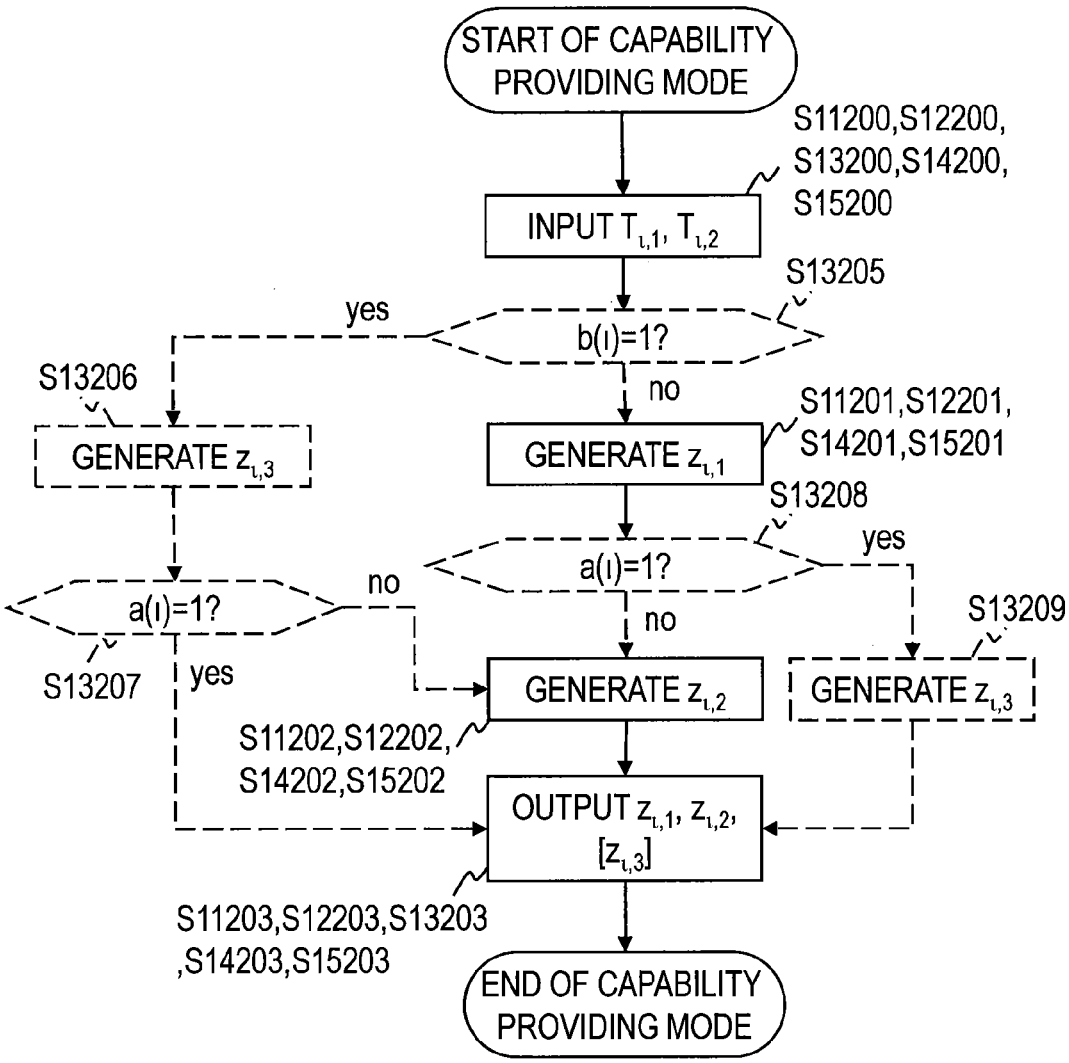

START OF CAPABILITY PROVIDING MODE
S11200,S12200, S13200,S14200, S15200
INPUT $T_{t,1}$, $T_{t,2}$
S13205
b(I)=1?
yes
no
S13206
GENERATE $z_{t,3}$
S11201,S12201, S14201,S15201
GENERATE $z_{t,1}$
S13208
a(I)=1?
yes
no
a(I)=1?
no
yes
S13207
S13209
GENERATE $z_{t,2}$
GENERATE $z_{t,3}$
S11202,S12202, S14202,S15202
OUTPUT $z_{t,1}$, $z_{t,2}$, [$z_{t,3}$]
S11203,S12203,S13203 ,S14203,S15203
END OF CAPABILITY PROVIDING MODE

FIG.27

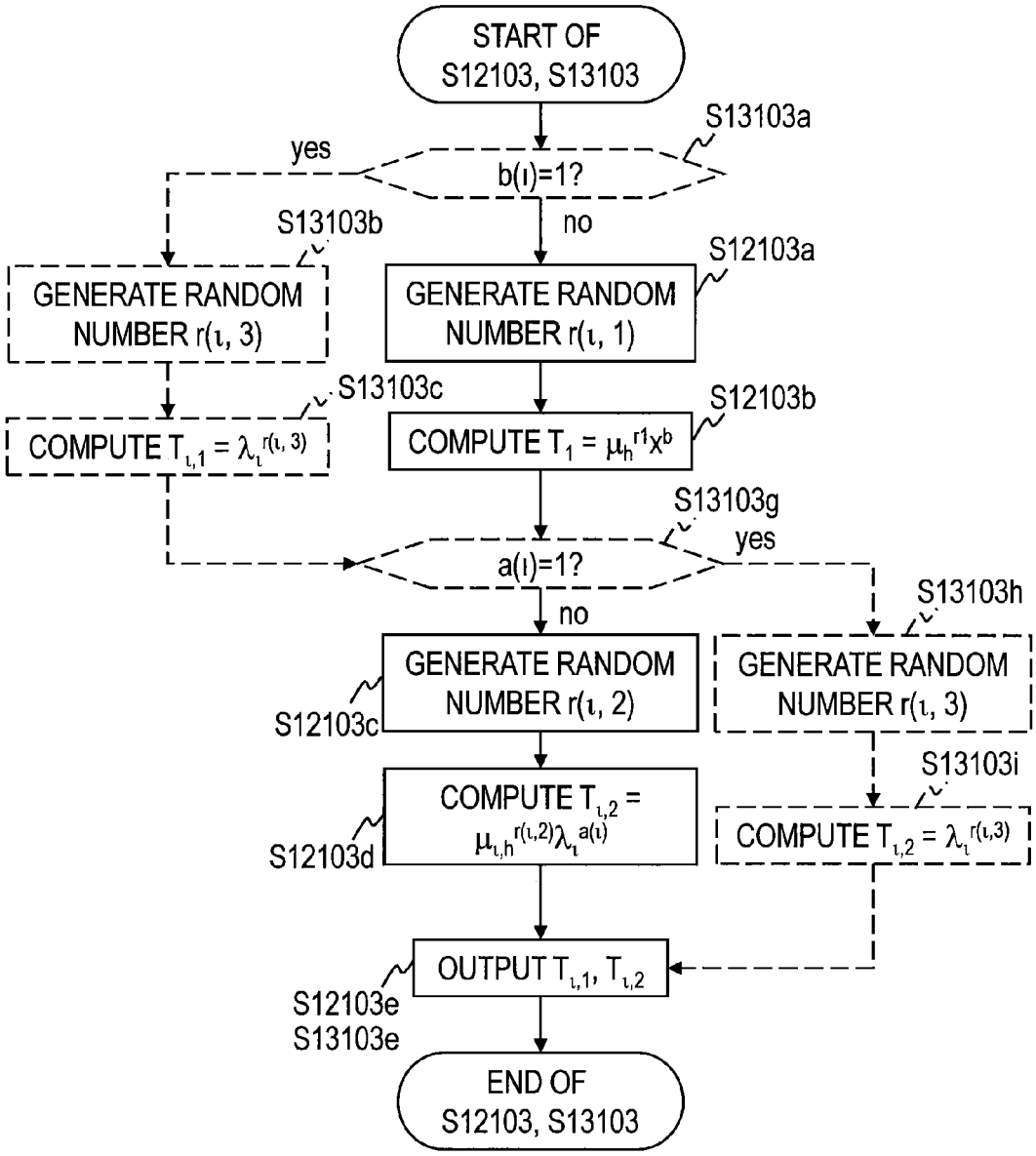

START OF S12103, S13103
S13103a
b(ι)=1?
yes
no
S13103b
GENERATE RANDOM NUMBER r(ι, 3)
S12103a
GENERATE RANDOM NUMBER r(ι, 1)
S13103c
COMPUTE $T_{\iota,1} = \lambda_\iota^{r(\iota,3)}$
S12103b
COMPUTE $T_1 = \mu_h^{r1} x^b$
S13103g
a(ι)=1?
yes
no
S13103h
GENERATE RANDOM NUMBER r(ι, 2)
S12103c
GENERATE RANDOM NUMBER r(ι, 3)
S13103i
COMPUTE $T_{\iota,2} = \mu_{\iota,h}^{r(\iota,2)} \lambda_\iota^{a(\iota)}$
S12103d
COMPUTE $T_{\iota,2} = \lambda_\iota^{r(\iota,3)}$
OUTPUT $T_{\iota,1}$, $T_{\iota,2}$
S12103e
S13103e
END OF S12103, S13103

FIG.28

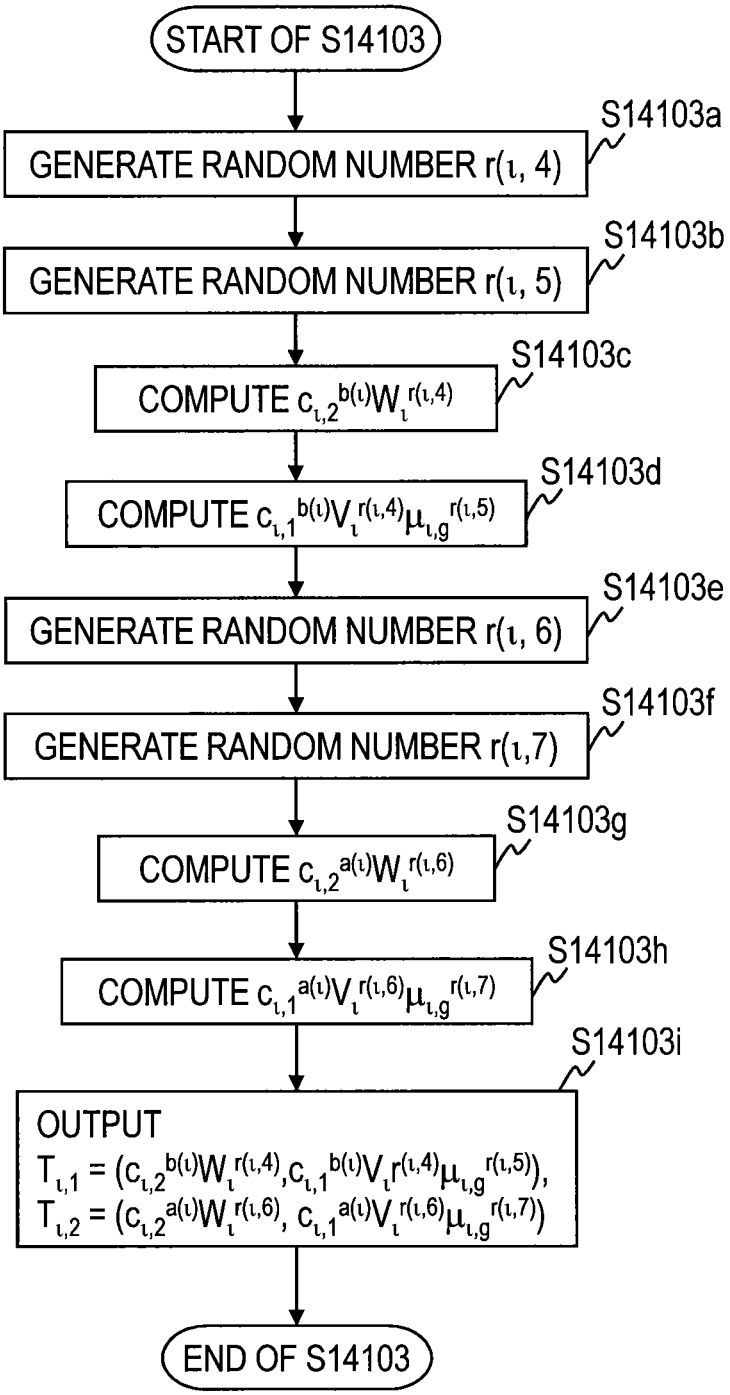

START OF S14103
S14103a
GENERATE RANDOM NUMBER r(ι, 4)
S14103b
GENERATE RANDOM NUMBER r(ι, 5)
S14103c
COMPUTE $c_{\iota,2}^{b(\iota)}W_{\iota}^{r(\iota,4)}$
S14103d
COMPUTE $c_{\iota,1}^{b(\iota)}V_{\iota}^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}$
S14103e
GENERATE RANDOM NUMBER r(ι, 6)
S14103f
GENERATE RANDOM NUMBER r(ι,7)
S14103g
COMPUTE $c_{\iota,2}^{a(\iota)}W_{\iota}^{r(\iota,6)}$
S14103h
COMPUTE $c_{\iota,1}^{a(\iota)}V_{\iota}^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)}$
S14103i
OUTPUT
$T_{\iota,1} = (c_{\iota,2}^{b(\iota)}W_{\iota}^{r(\iota,4)}, c_{\iota,1}^{b(\iota)}V_{\iota}^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)})$,
$T_{\iota,2} = (c_{\iota,2}^{a(\iota)}W_{\iota}^{r(\iota,6)}, c_{\iota,1}^{a(\iota)}V_{\iota}^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)})$
END OF S14103

FIG.29

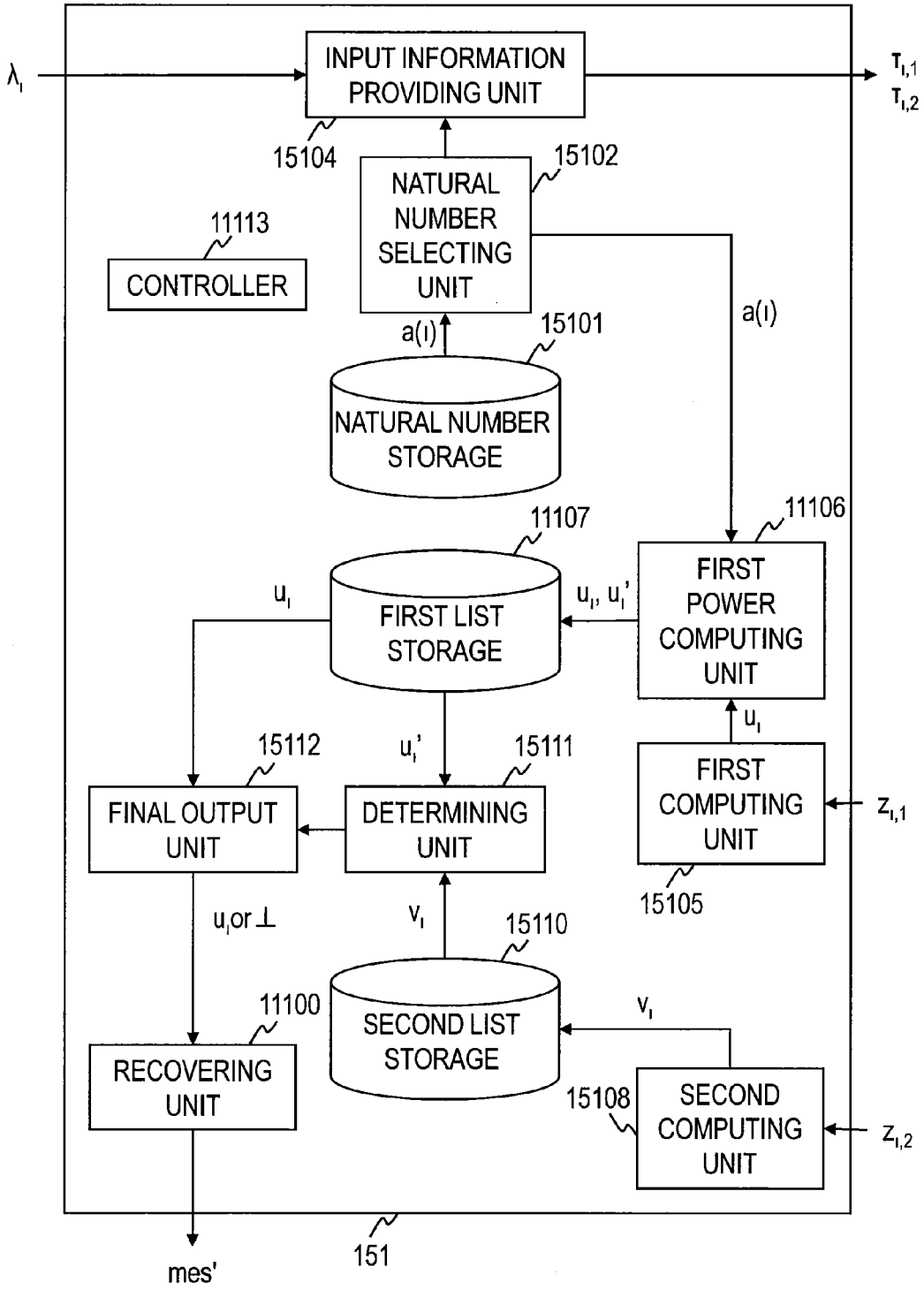

λ_I
INPUT INFORMATION PROVIDING UNIT
$T_{I,1}$
$T_{I,2}$
15104
15102
NATURAL NUMBER SELECTING UNIT
11113
CONTROLLER
a(I)
15101
NATURAL NUMBER STORAGE
a(I)
11107
11106
u_I
FIRST LIST STORAGE
u_I, u_I'
FIRST POWER COMPUTING UNIT
u_I
15112
u_I'
15111
FINAL OUTPUT UNIT
DETERMINING UNIT
FIRST COMPUTING UNIT
$z_{I,1}$
15105
u_I or ⊥
v_I
15110
11100
SECOND LIST STORAGE
v_I
RECOVERING UNIT
15108
SECOND COMPUTING UNIT
$z_{I,2}$
mes'
151

FIG.30

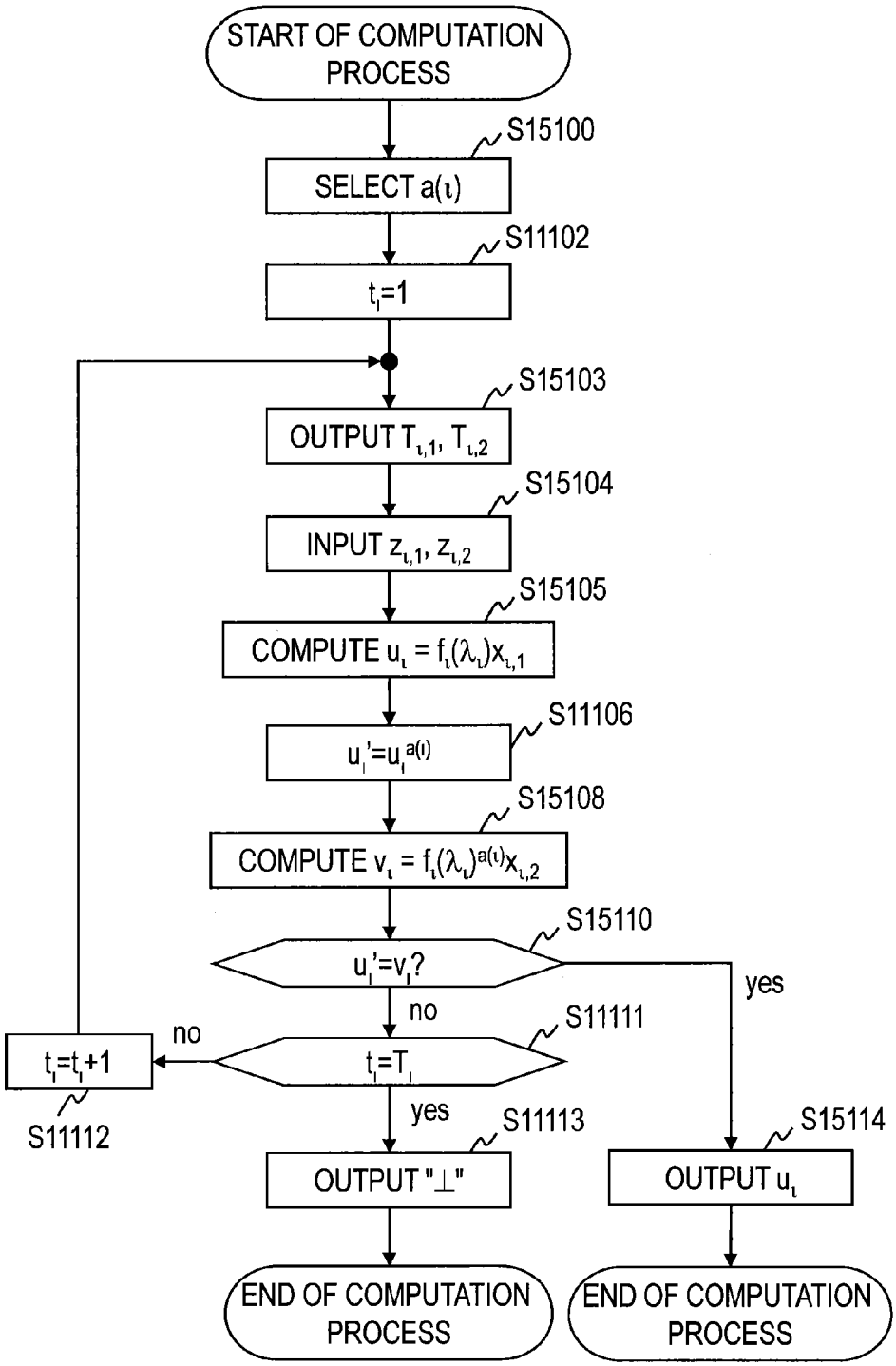

START OF COMPUTATION PROCESS
S15100
SELECT a(ι)
S11102
$t_ι$=1
S15103
OUTPUT $T_{ι,1}$, $T_{ι,2}$
S15104
INPUT $z_{ι,1}$, $z_{ι,2}$
S15105
COMPUTE $u_ι = f_ι(\lambda_ι)x_{ι,1}$
S11106
$u_ι'=u_ι^{a(ι)}$
S15108
COMPUTE $v_ι = f_ι(\lambda_ι)^{a(ι)}x_{ι,2}$
S15110
$u_ι'=v_ι$?
yes
no
S11111
$t_ι=T_ι$
no
$t_ι=t_ι+1$
S11112
yes
S11113
OUTPUT "⊥"
S15114
OUTPUT $u_ι$
END OF COMPUTATION PROCESS
END OF COMPUTATION PROCESS

FIG.31

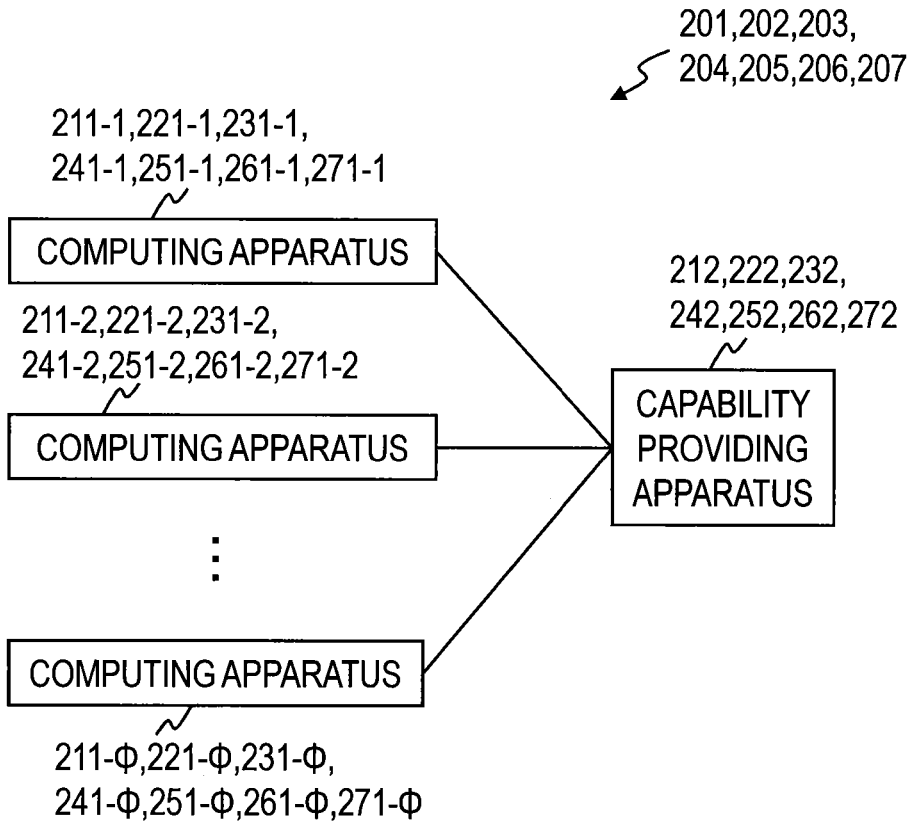

201,202,203,
204,205,206,207
211-1,221-1,231-1,
241-1,251-1,261-1,271-1
COMPUTING APPARATUS
211-2,221-2,231-2,
241-2,251-2,261-2,271-2
COMPUTING APPARATUS
COMPUTING APPARATUS
211-Φ,221-Φ,231-Φ,
241-Φ,251-Φ,261-Φ,271-Φ
212,222,232,
242,252,262,272
CAPABILITY PROVIDING APPARATUS

FIG.32

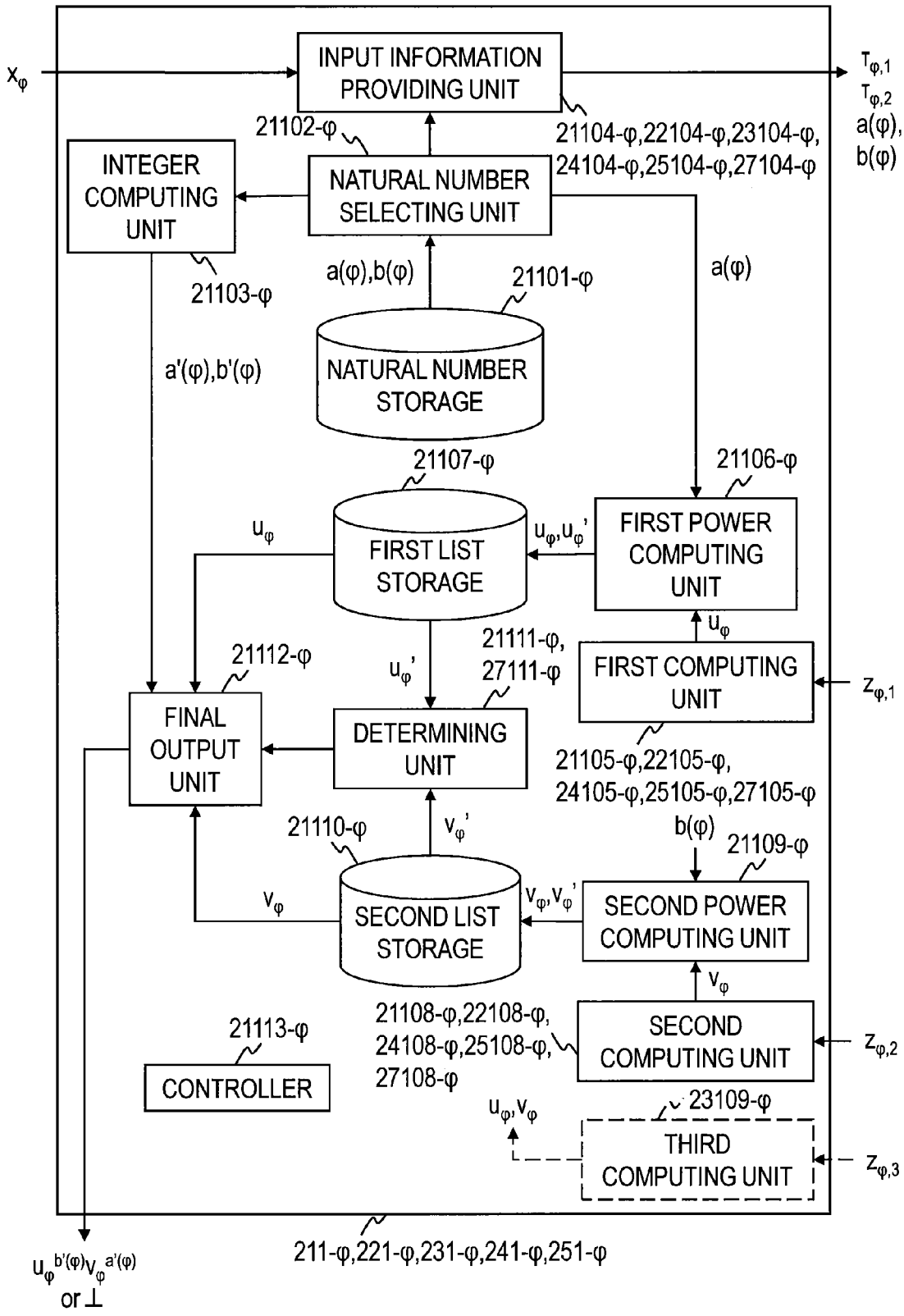

INPUT INFORMATION PROVIDING UNIT
x_φ
$T_{\varphi,1}$
$T_{\varphi,2}$
a(φ),
b(φ)
21102-φ
21104-φ,22104-φ,23104-φ,
24104-φ,25104-φ,27104-φ
INTEGER COMPUTING UNIT
NATURAL NUMBER SELECTING UNIT
21103-φ
a(φ),b(φ)
21101-φ
a(φ)
a'(φ),b'(φ)
NATURAL NUMBER STORAGE
21107-φ
21106-φ
u_φ
FIRST LIST STORAGE
u_φ,u_φ'
FIRST POWER COMPUTING UNIT
u_φ
21112-φ
u_φ'
21111-φ,
27111-φ
FIRST COMPUTING UNIT
$z_{\varphi,1}$
FINAL OUTPUT UNIT
DETERMINING UNIT
21105-φ,22105-φ,
24105-φ,25105-φ,27105-φ
b(φ)
21110-φ
v_φ'
21109-φ
v_φ
SECOND LIST STORAGE
v_φ,v_φ'
SECOND POWER COMPUTING UNIT
v_φ
21113-φ
21108-φ,22108-φ,
24108-φ,25108-φ,
27108-φ
SECOND COMPUTING UNIT
$z_{\varphi,2}$
CONTROLLER
u_φ,v_φ
23109-φ
THIRD COMPUTING UNIT
$z_{\varphi,3}$
211-φ,221-φ,231-φ,241-φ,251-φ
$u_\varphi^{b'(\varphi)}v_\varphi^{a'(\varphi)}$
or ⊥

FIG.33

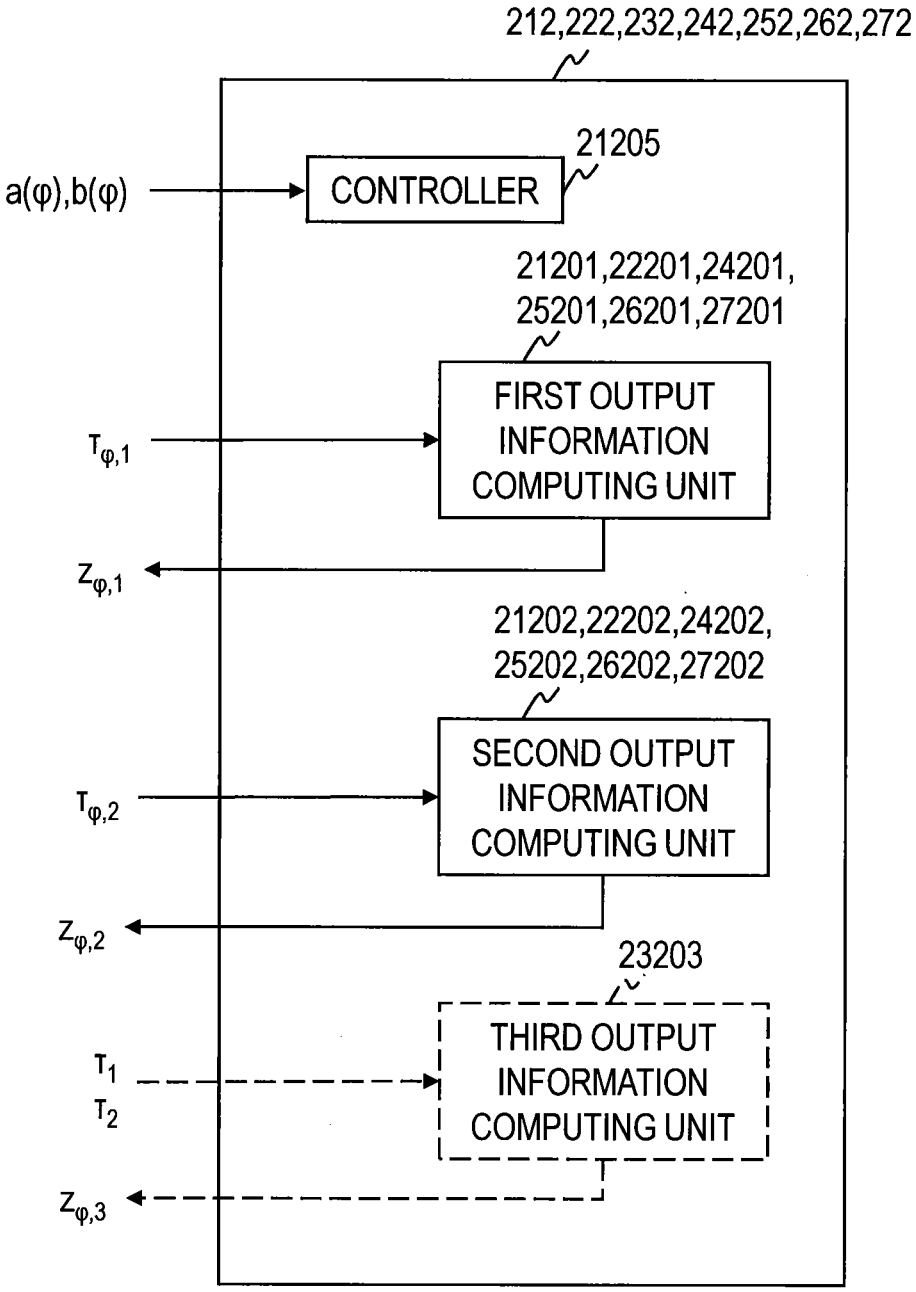

212,222,232,242,252,262,272
21205
a(φ),b(φ)
CONTROLLER
21201,22201,24201,
25201,26201,27201
FIRST OUTPUT INFORMATION COMPUTING UNIT
$T_{\varphi,1}$
$Z_{\varphi,1}$
21202,22202,24202,
25202,26202,27202
SECOND OUTPUT INFORMATION COMPUTING UNIT
$T_{\varphi,2}$
$Z_{\varphi,2}$
23203
THIRD OUTPUT INFORMATION COMPUTING UNIT
T_1
T_2
$Z_{\varphi,3}$

FIG.34

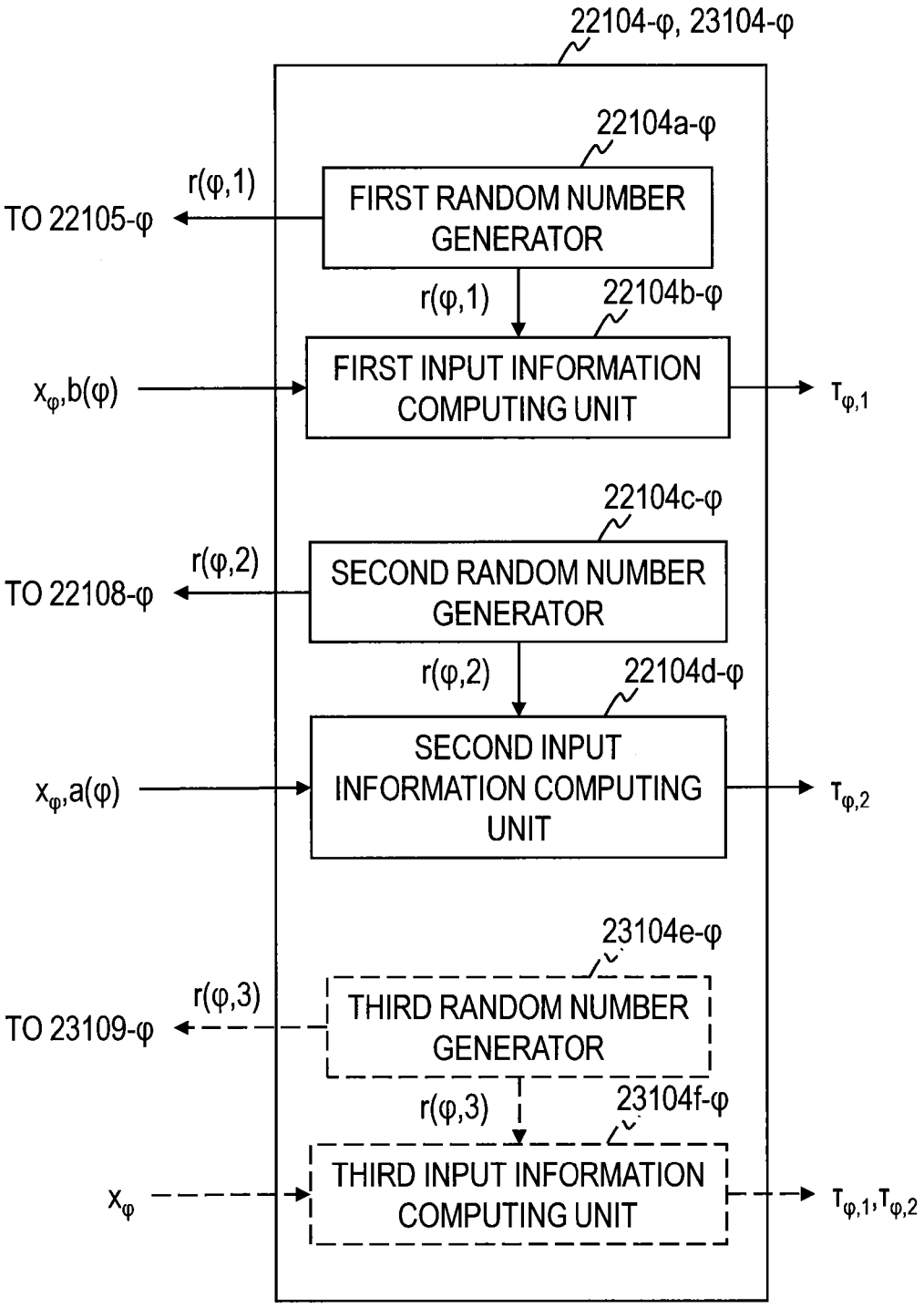

22104-φ, 23104-φ
22104a-φ
r(φ,1)
TO 22105-φ
FIRST RANDOM NUMBER GENERATOR
r(φ,1)
22104b-φ
$x_φ$,b(φ)
FIRST INPUT INFORMATION COMPUTING UNIT
$T_{φ,1}$
22104c-φ
r(φ,2)
TO 22108-φ
SECOND RANDOM NUMBER GENERATOR
r(φ,2)
22104d-φ
$x_φ$,a(φ)
SECOND INPUT INFORMATION COMPUTING UNIT
$T_{φ,2}$
23104e-φ
r(φ,3)
TO 23109-φ
THIRD RANDOM NUMBER GENERATOR
r(φ,3)
23104f-φ
$x_φ$
THIRD INPUT INFORMATION COMPUTING UNIT
$T_{φ,1}$,$T_{φ,2}$

FIG.35

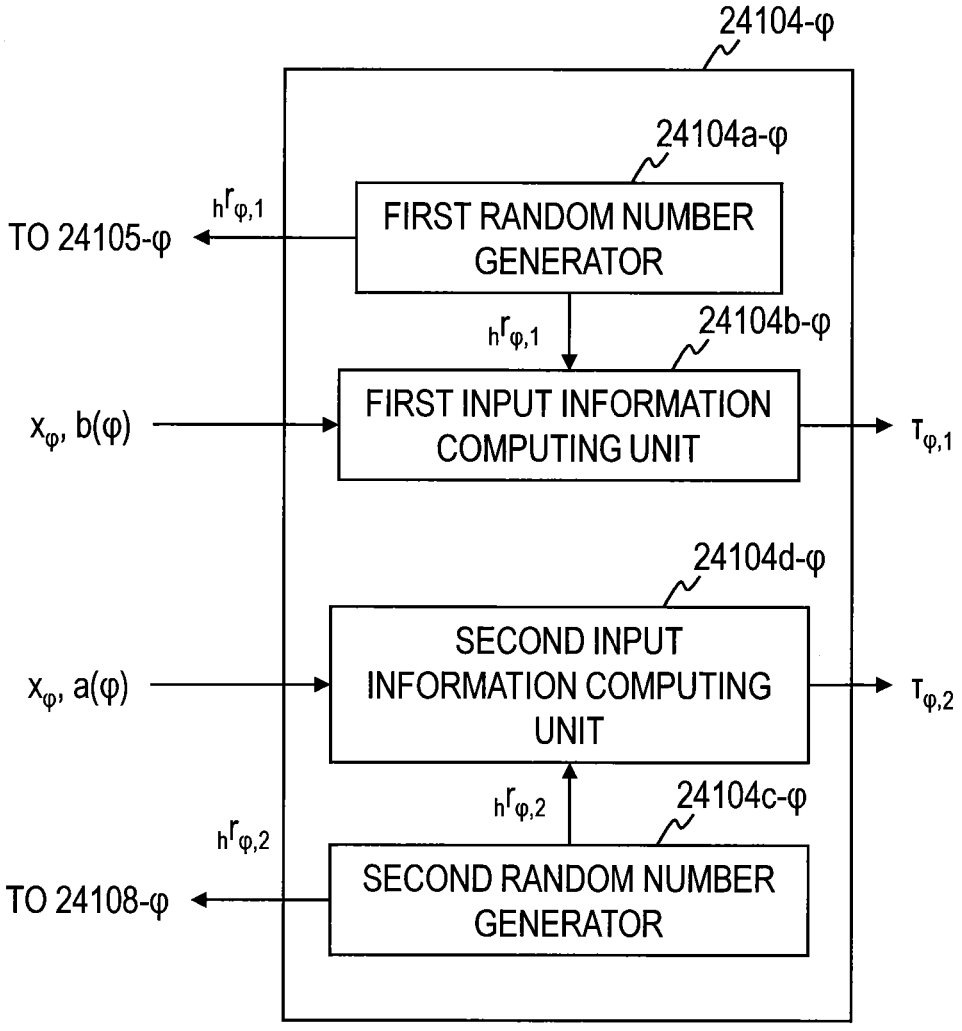

24104-φ
24104a-φ
FIRST RANDOM NUMBER GENERATOR
hrφ,1
TO 24105-φ
hrφ,1
24104b-φ
FIRST INPUT INFORMATION COMPUTING UNIT
xφ, b(φ)
Tφ,1
24104d-φ
SECOND INPUT INFORMATION COMPUTING UNIT
xφ, a(φ)
Tφ,2
hrφ,2
24104c-φ
hrφ,2
SECOND RANDOM NUMBER GENERATOR
TO 24108-φ

FIG.36

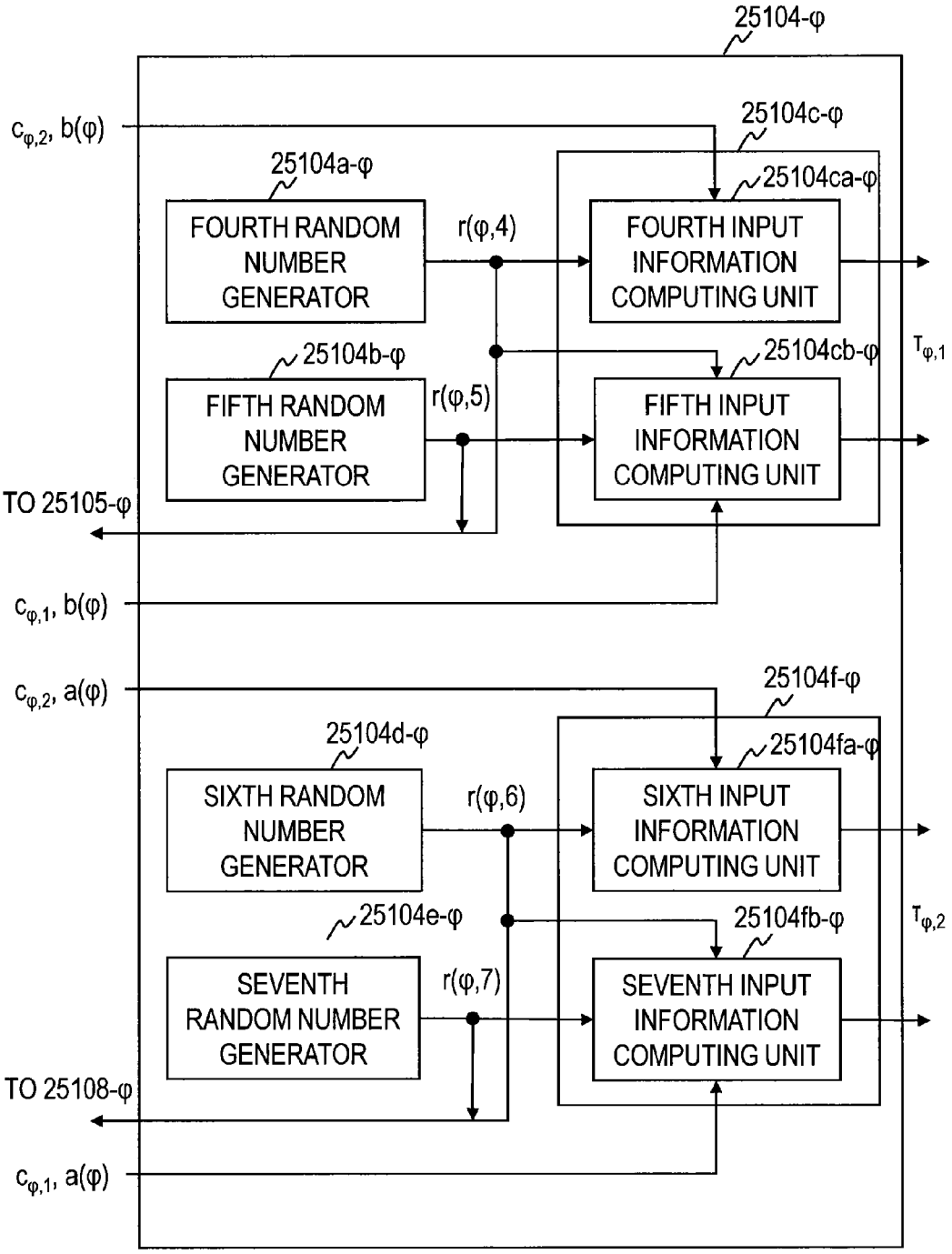

25104-φ
$c_{\varphi,2}$, b(φ)
25104c-φ
25104a-φ
25104ca-φ
FOURTH RANDOM NUMBER GENERATOR
r(φ,4)
FOURTH INPUT INFORMATION COMPUTING UNIT
25104b-φ
25104cb-φ
$T_{\varphi,1}$
FIFTH RANDOM NUMBER GENERATOR
r(φ,5)
FIFTH INPUT INFORMATION COMPUTING UNIT
TO 25105-φ
$c_{\varphi,1}$, b(φ)
$c_{\varphi,2}$, a(φ)
25104f-φ
25104d-φ
25104fa-φ
SIXTH RANDOM NUMBER GENERATOR
r(φ,6)
SIXTH INPUT INFORMATION COMPUTING UNIT
25104e-φ
25104fb-φ
$T_{\varphi,2}$
SEVENTH RANDOM NUMBER GENERATOR
r(φ,7)
SEVENTH INPUT INFORMATION COMPUTING UNIT
TO 25108-φ
$c_{\varphi,1}$, a(φ)

FIG.37

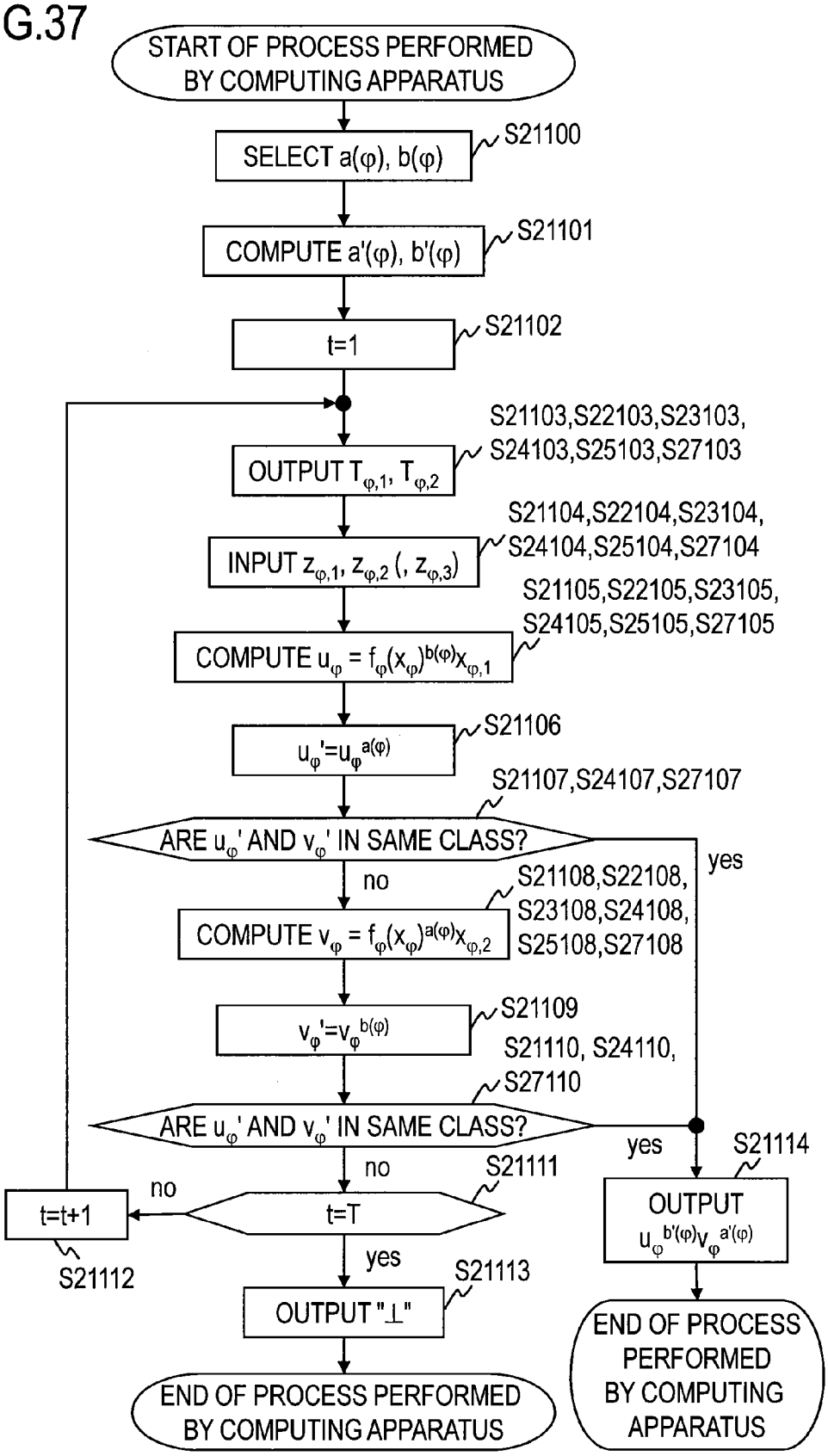

START OF PROCESS PERFORMED BY COMPUTING APPARATUS
SELECT a(φ), b(φ)
S21100
COMPUTE a'(φ), b'(φ)
S21101
t=1
S21102
OUTPUT $T_{\varphi,1}$, $T_{\varphi,2}$
S21103,S22103,S23103, S24103,S25103,S27103
INPUT $z_{\varphi,1}$, $z_{\varphi,2}$ (, $z_{\varphi,3}$)
S21104,S22104,S23104, S24104,S25104,S27104
COMPUTE $u_\varphi = f_\varphi(x_\varphi)^{b(\varphi)}x_{\varphi,1}$
S21105,S22105,S23105, S24105,S25105,S27105
$u_\varphi' = u_\varphi^{a(\varphi)}$
S21106
ARE u_φ' AND v_φ' IN SAME CLASS?
S21107,S24107,S27107
yes
no
COMPUTE $v_\varphi = f_\varphi(x_\varphi)^{a(\varphi)}x_{\varphi,2}$
S21108,S22108, S23108,S24108, S25108,S27108
$v_\varphi' = v_\varphi^{b(\varphi)}$
S21109
ARE u_φ' AND v_φ' IN SAME CLASS?
S21110, S24110, S27110
yes
S21114
OUTPUT $u_\varphi^{b'(\varphi)}v_\varphi^{a'(\varphi)}$
END OF PROCESS PERFORMED BY COMPUTING APPARATUS
no
t=T
S21111
no
t=t+1
S21112
yes
OUTPUT "⊥"
S21113
END OF PROCESS PERFORMED BY COMPUTING APPARATUS

FIG.38

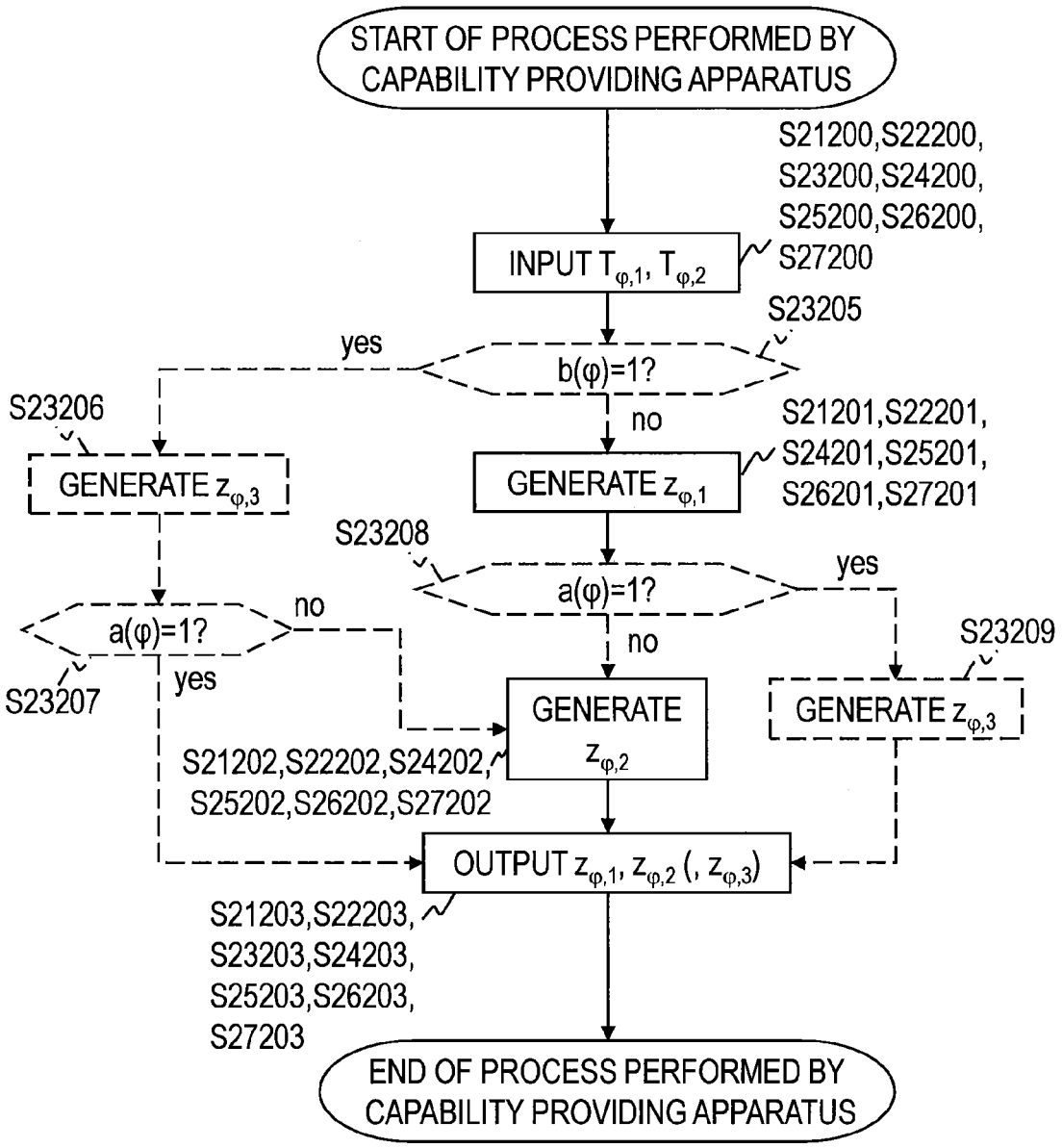

START OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS
S21200,S22200, S23200,S24200, S25200,S26200, S27200
INPUT $T_{\varphi,1}$, $T_{\varphi,2}$
S23205
b(φ)=1?
yes
no
S23206
GENERATE $z_{\varphi,3}$
S21201,S22201, S24201,S25201, S26201,S27201
GENERATE $z_{\varphi,1}$
S23208
a(φ)=1?
yes
no
S23209
GENERATE $z_{\varphi,3}$
a(φ)=1?
no
yes
S23207
GENERATE $z_{\varphi,2}$
S21202,S22202,S24202, S25202,S26202,S27202
OUTPUT $z_{\varphi,1}$, $z_{\varphi,2}$ (, $z_{\varphi,3}$)
S21203,S22203, S23203,S24203, S25203,S26203, S27203
END OF PROCESS PERFORMED BY CAPABILITY PROVIDING APPARATUS

FIG.39

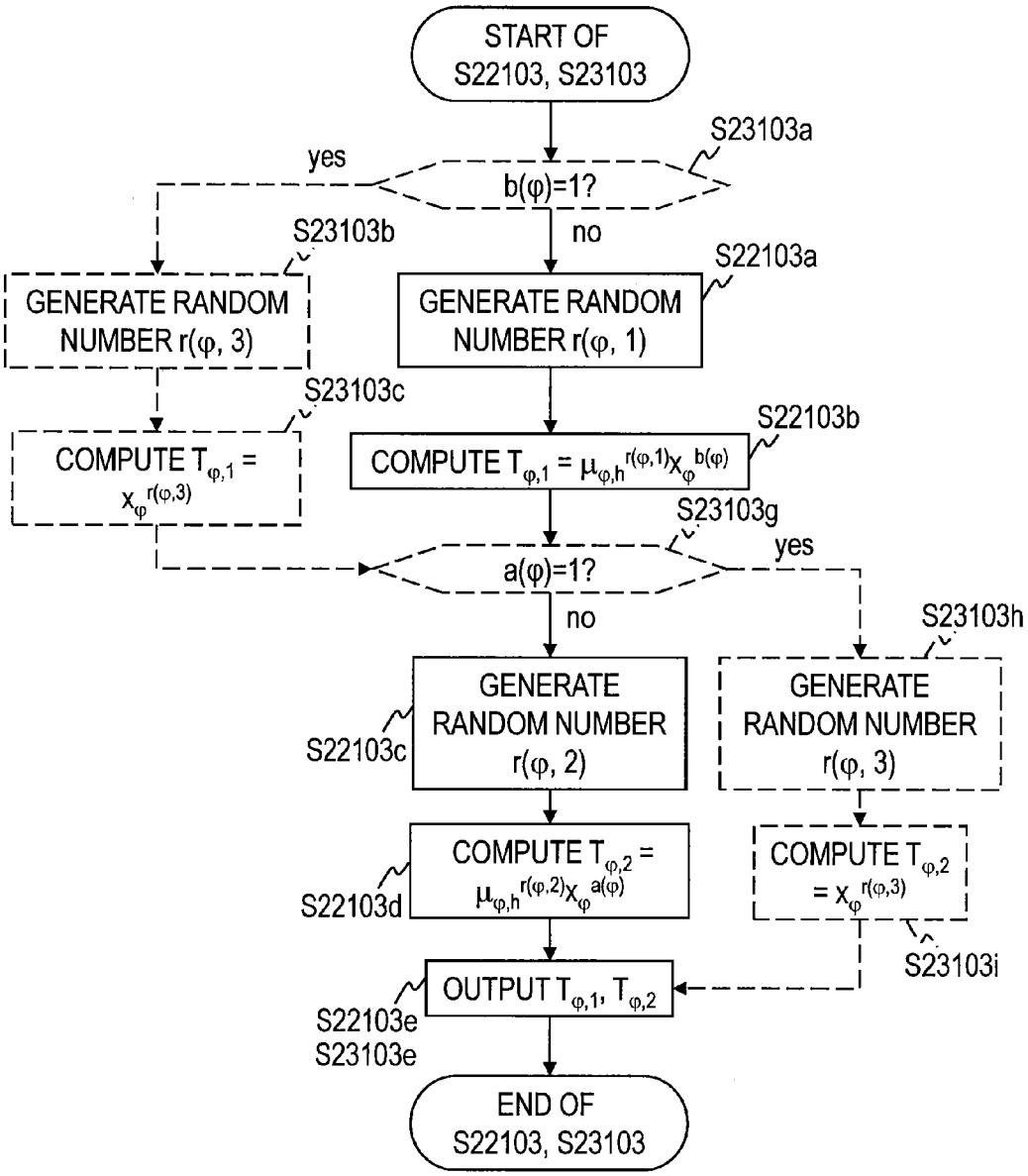

START OF S22103, S23103
S23103a
b(φ)=1?
yes
no
S23103b
GENERATE RANDOM NUMBER r(φ, 3)
S22103a
GENERATE RANDOM NUMBER r(φ, 1)
S23103c
COMPUTE $T_{\varphi,1}$ = $x_{\varphi}^{r(\varphi,3)}$
S22103b
COMPUTE $T_{\varphi,1}$ = $\mu_{\varphi,h}^{r(\varphi,1)} x_{\varphi}^{b(\varphi)}$
S23103g
a(φ)=1?
yes
no
S23103h
S22103c
GENERATE RANDOM NUMBER r(φ, 2)
GENERATE RANDOM NUMBER r(φ, 3)
S22103d
COMPUTE $T_{\varphi,2}$ = $\mu_{\varphi,h}^{r(\varphi,2)} x_{\varphi}^{a(\varphi)}$
COMPUTE $T_{\varphi,2}$ = $x_{\varphi}^{r(\varphi,3)}$
S23103i
S22103e
S23103e
OUTPUT $T_{\varphi,1}$, $T_{\varphi,2}$
END OF S22103, S23103

FIG.40

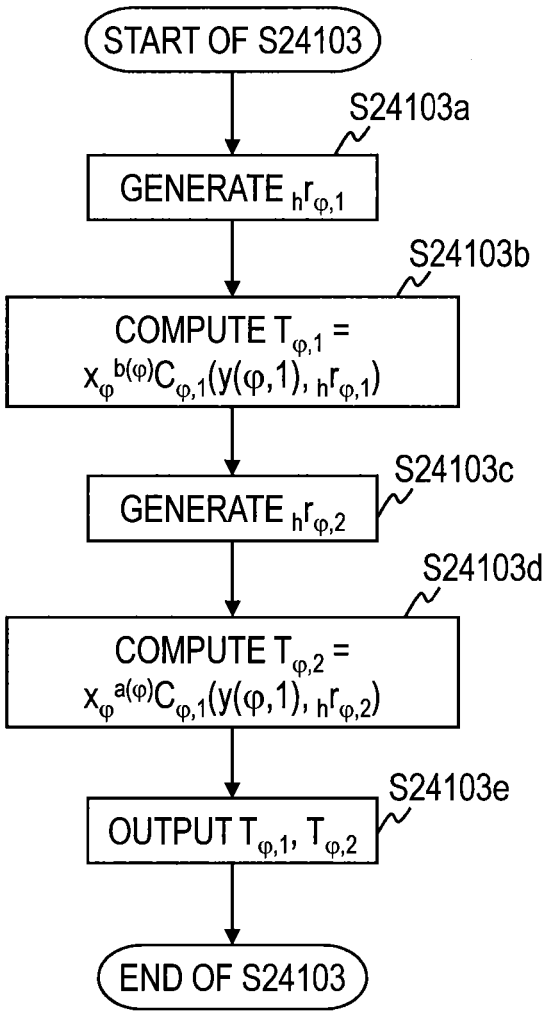

START OF S24103
S24103a
GENERATE $_{h}r_{\varphi,1}$
S24103b
COMPUTE $T_{\varphi,1}$ = $x_{\varphi}^{b(\varphi)}C_{\varphi,1}(y(\varphi,1), {}_{h}r_{\varphi,1})$
S24103c
GENERATE $_{h}r_{\varphi,2}$
S24103d
COMPUTE $T_{\varphi,2}$ = $x_{\varphi}^{a(\varphi)}C_{\varphi,1}(y(\varphi,1), {}_{h}r_{\varphi,2})$
S24103e
OUTPUT $T_{\varphi,1}$, $T_{\varphi,2}$
END OF S24103

FIG.41

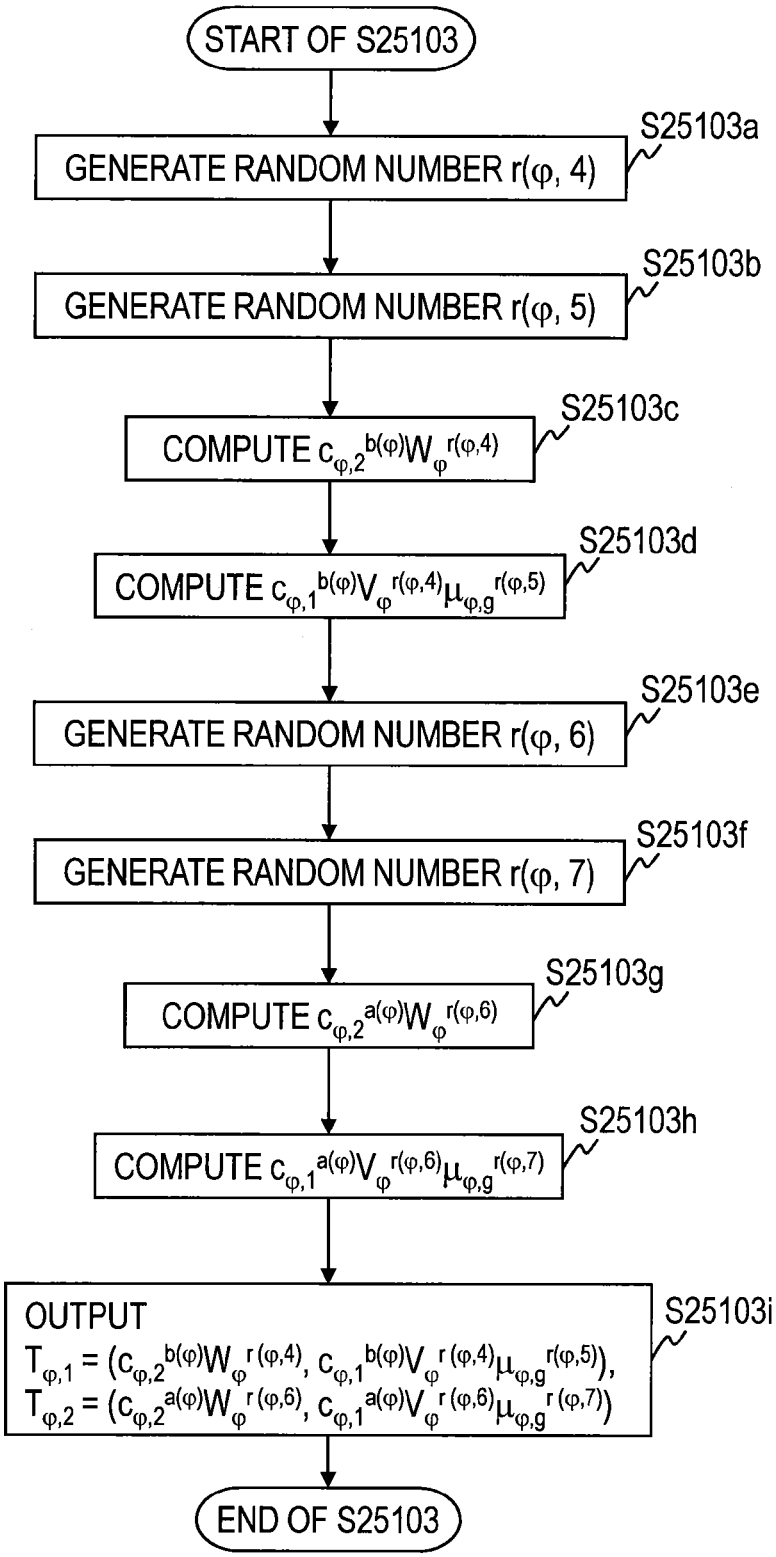

START OF S25103
GENERATE RANDOM NUMBER r(φ, 4)
S25103a
GENERATE RANDOM NUMBER r(φ, 5)
S25103b
COMPUTE $c_{\varphi,2}^{b(\varphi)}W_{\varphi}^{r(\varphi,4)}$
S25103c
COMPUTE $c_{\varphi,1}^{b(\varphi)}V_{\varphi}^{r(\varphi,4)}\mu_{\varphi,g}^{r(\varphi,5)}$
S25103d
GENERATE RANDOM NUMBER r(φ, 6)
S25103e
GENERATE RANDOM NUMBER r(φ, 7)
S25103f
COMPUTE $c_{\varphi,2}^{a(\varphi)}W_{\varphi}^{r(\varphi,6)}$
S25103g
COMPUTE $c_{\varphi,1}^{a(\varphi)}V_{\varphi}^{r(\varphi,6)}\mu_{\varphi,g}^{r(\varphi,7)}$
S25103h
OUTPUT
$T_{\varphi,1} = (c_{\varphi,2}^{b(\varphi)}W_{\varphi}^{r(\varphi,4)}, c_{\varphi,1}^{b(\varphi)}V_{\varphi}^{r(\varphi,4)}\mu_{\varphi,g}^{r(\varphi,5)})$,
$T_{\varphi,2} = (c_{\varphi,2}^{a(\varphi)}W_{\varphi}^{r(\varphi,6)}, c_{\varphi,1}^{a(\varphi)}V_{\varphi}^{r(\varphi,6)}\mu_{\varphi,g}^{r(\varphi,7)})$
S25103i
END OF S25103

FIG.42

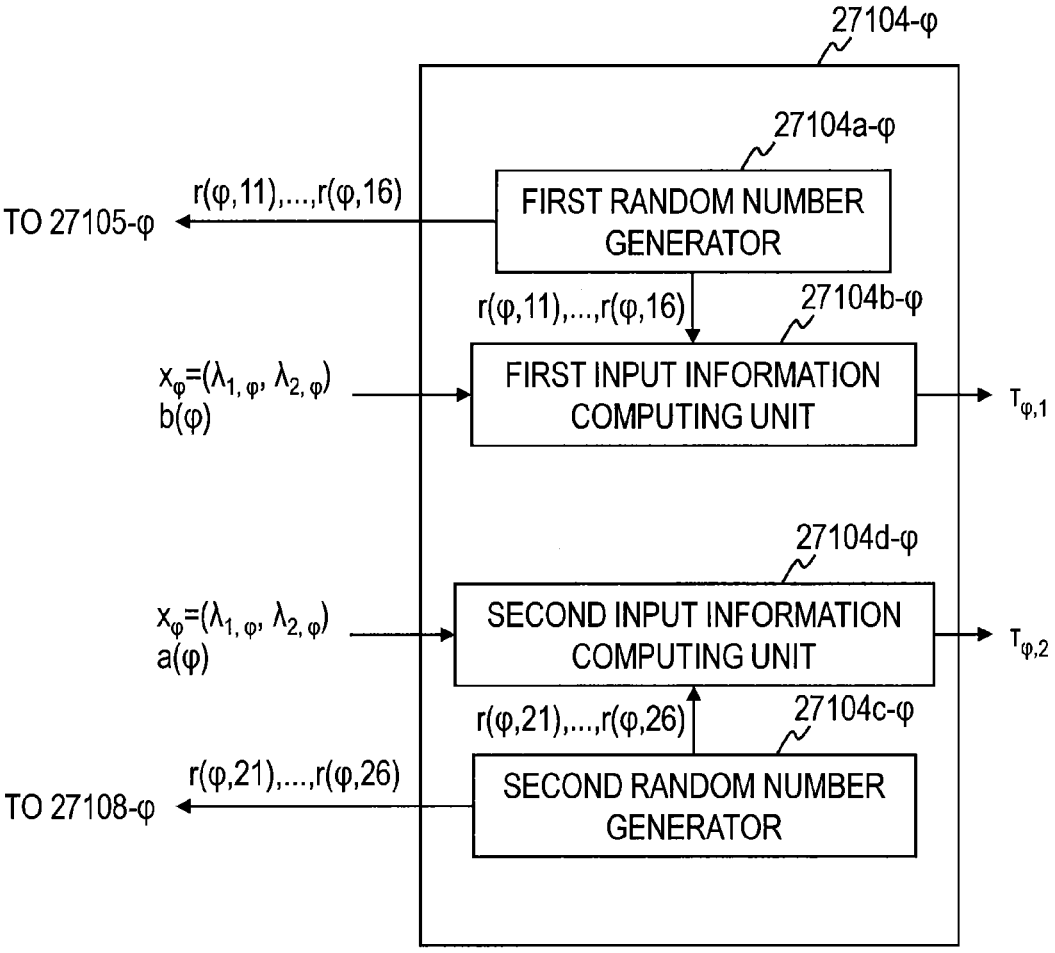

27104-φ
27104a-φ
FIRST RANDOM NUMBER GENERATOR
r(φ,11),...,r(φ,16)
TO 27105-φ
r(φ,11),...,r(φ,16)
27104b-φ
FIRST INPUT INFORMATION COMPUTING UNIT
$x_\varphi=(\lambda_{1,\varphi}, \lambda_{2,\varphi})$
b(φ)
$T_{\varphi,1}$
27104d-φ
SECOND INPUT INFORMATION COMPUTING UNIT
$x_\varphi=(\lambda_{1,\varphi}, \lambda_{2,\varphi})$
a(φ)
$T_{\varphi,2}$
r(φ,21),...,r(φ,26)
27104c-φ
SECOND RANDOM NUMBER GENERATOR
r(φ,21),...,r(φ,26)
TO 27108-φ

FIG.43

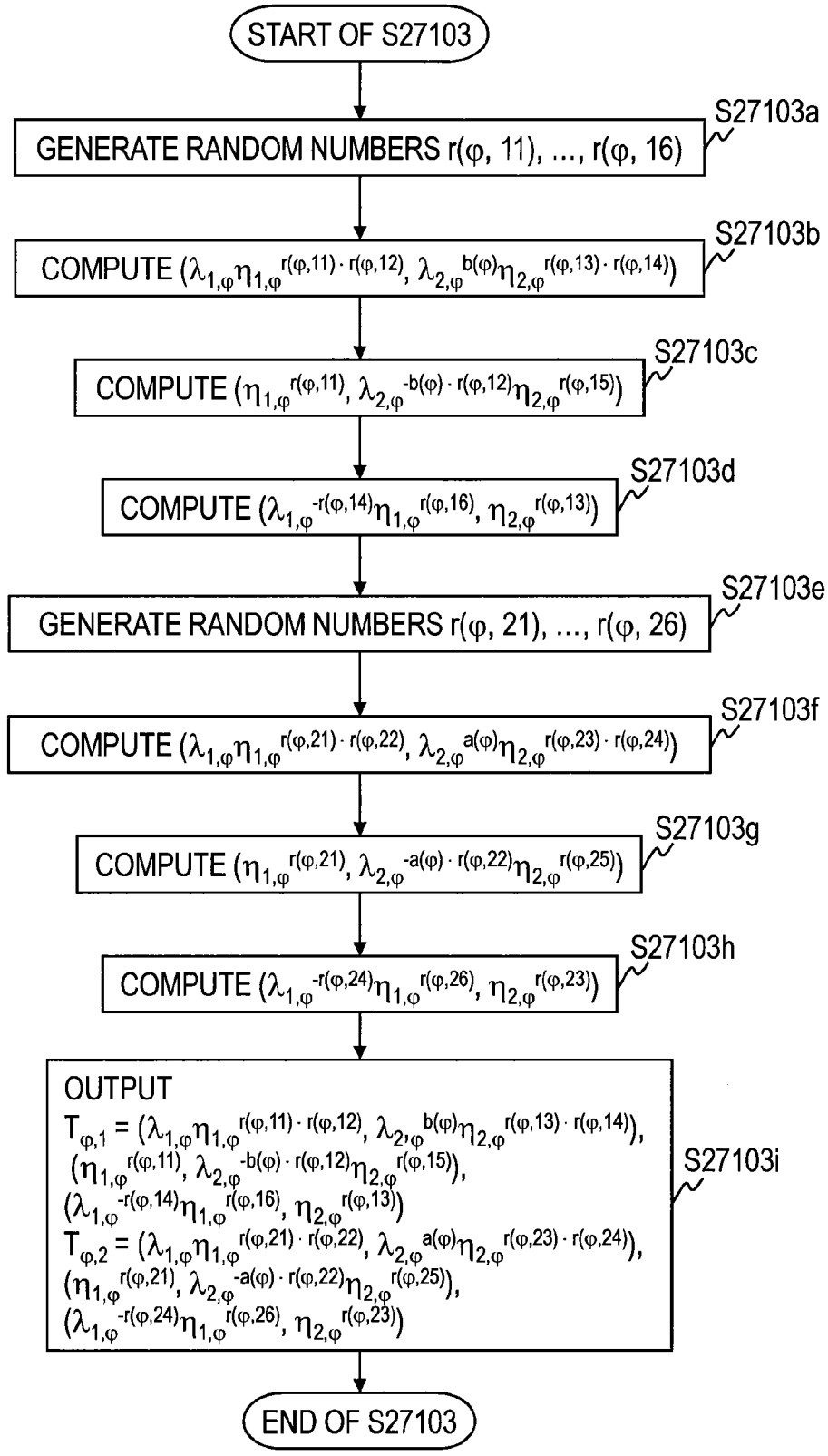

START OF S27103
GENERATE RANDOM NUMBERS r(φ, 11), ..., r(φ, 16)
S27103a
COMPUTE $(\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,11)\cdot r(\varphi,12)}, \lambda_{2,\varphi}^{b(\varphi)}\eta_{2,\varphi}^{r(\varphi,13)\cdot r(\varphi,14)})$
S27103b
COMPUTE $(\eta_{1,\varphi}^{r(\varphi,11)}, \lambda_{2,\varphi}^{-b(\varphi)\cdot r(\varphi,12)}\eta_{2,\varphi}^{r(\varphi,15)})$
S27103c
COMPUTE $(\lambda_{1,\varphi}^{-r(\varphi,14)}\eta_{1,\varphi}^{r(\varphi,16)}, \eta_{2,\varphi}^{r(\varphi,13)})$
S27103d
GENERATE RANDOM NUMBERS r(φ, 21), ..., r(φ, 26)
S27103e
COMPUTE $(\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,21)\cdot r(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{r(\varphi,23)\cdot r(\varphi,24)})$
S27103f
COMPUTE $(\eta_{1,\varphi}^{r(\varphi,21)}, \lambda_{2,\varphi}^{-a(\varphi)\cdot r(\varphi,22)}\eta_{2,\varphi}^{r(\varphi,25)})$
S27103g
COMPUTE $(\lambda_{1,\varphi}^{-r(\varphi,24)}\eta_{1,\varphi}^{r(\varphi,26)}, \eta_{2,\varphi}^{r(\varphi,23)})$
S27103h
OUTPUT
$T_{\varphi,1} = (\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,11)\cdot r(\varphi,12)}, \lambda_{2,\varphi}^{b(\varphi)}\eta_{2,\varphi}^{r(\varphi,13)\cdot r(\varphi,14)}),$
$(\eta_{1,\varphi}^{r(\varphi,11)}, \lambda_{2,\varphi}^{-b(\varphi)\cdot r(\varphi,12)}\eta_{2,\varphi}^{r(\varphi,15)}),$
$(\lambda_{1,\varphi}^{-r(\varphi,14)}\eta_{1,\varphi}^{r(\varphi,16)}, \eta_{2,\varphi}^{r(\varphi,13)})$
$T_{\varphi,2} = (\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,21)\cdot r(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{r(\varphi,23)\cdot r(\varphi,24)}),$
$(\eta_{1,\varphi}^{r(\varphi,21)}, \lambda_{2,\varphi}^{-a(\varphi)\cdot r(\varphi,22)}\eta_{2,\varphi}^{r(\varphi,25)}),$
$(\lambda_{1,\varphi}^{-r(\varphi,24)}\eta_{1,\varphi}^{r(\varphi,26)}, \eta_{2,\varphi}^{r(\varphi,23)})$
S27103i
END OF S27103

FIG.44

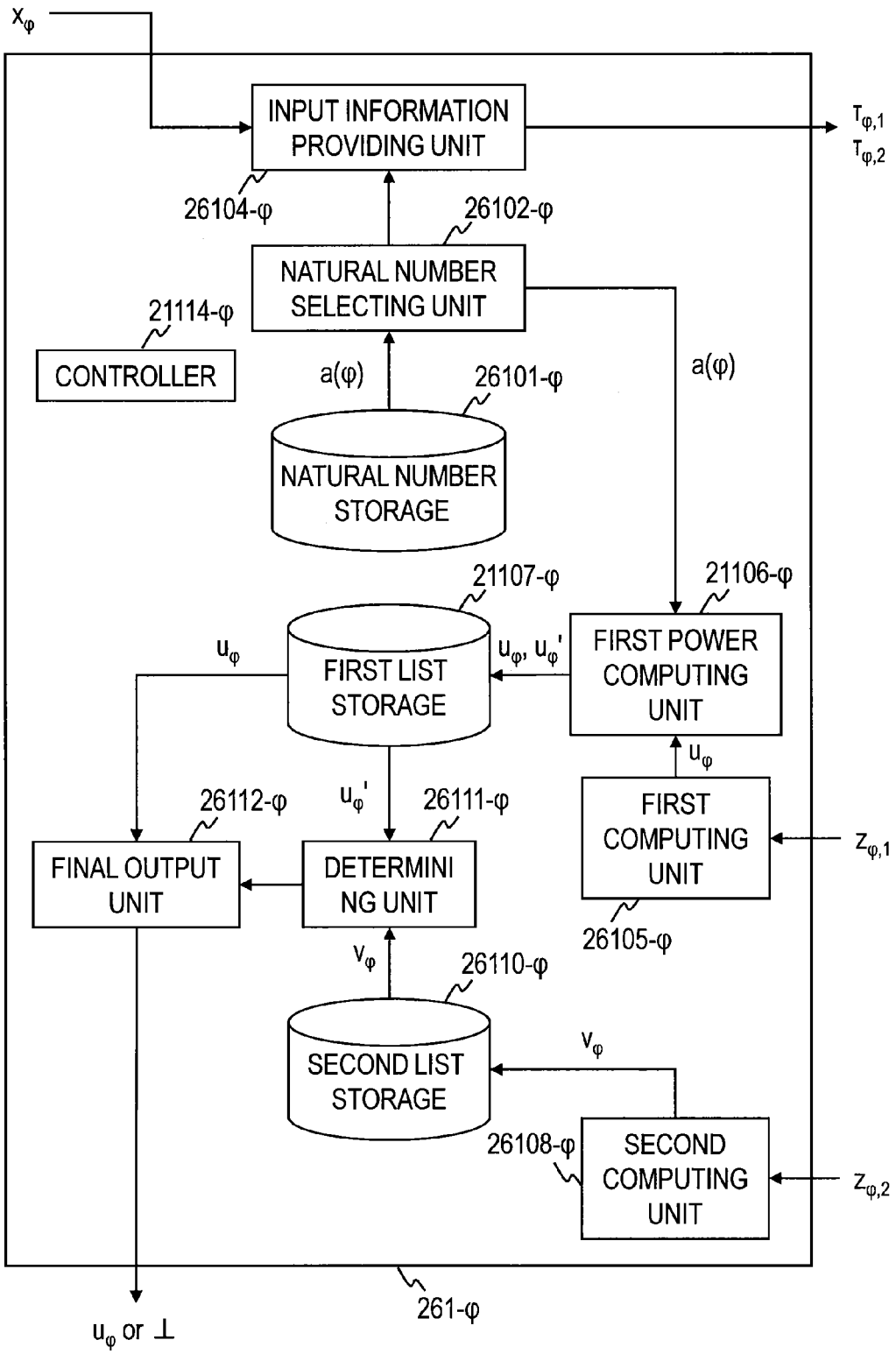
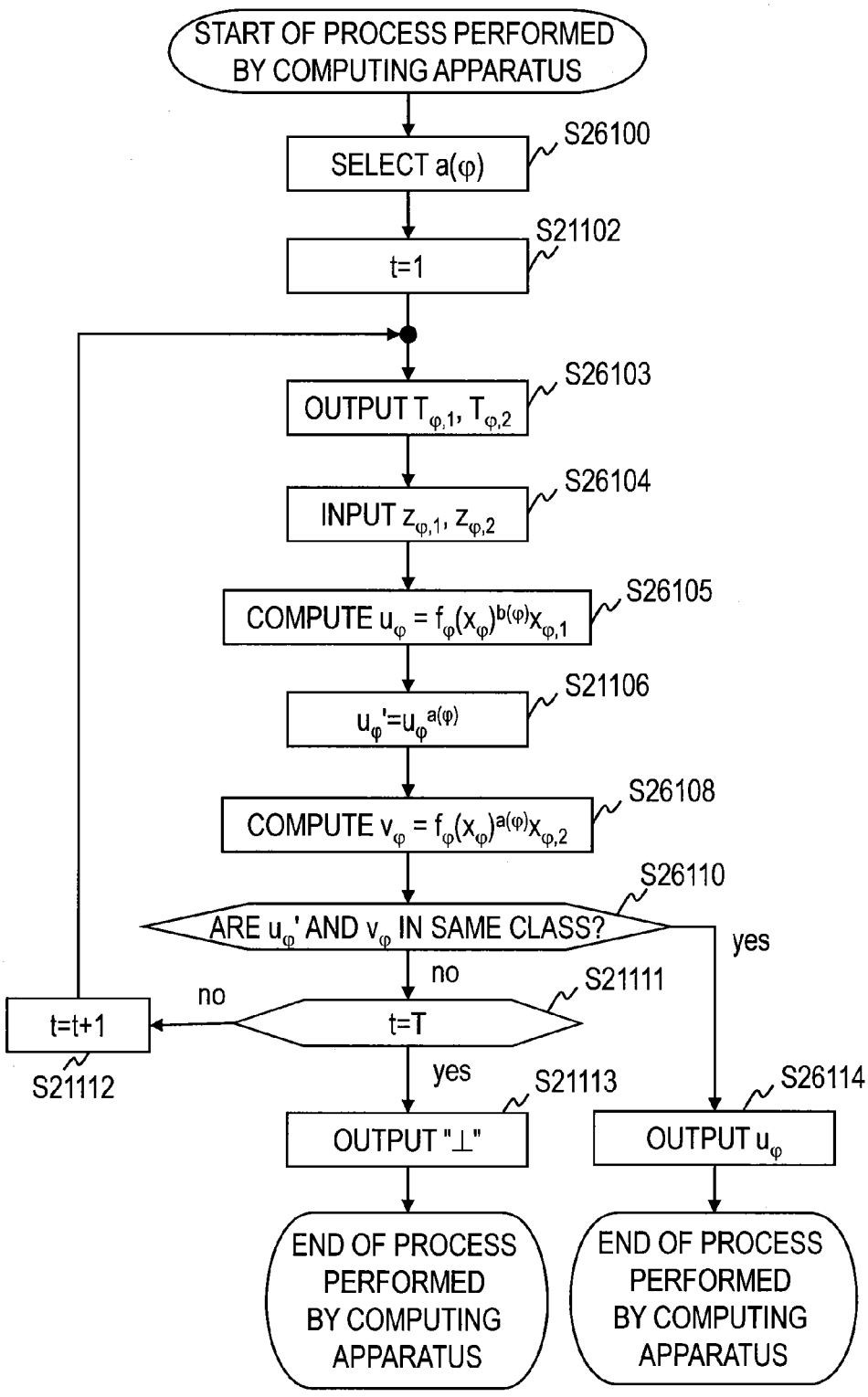

x_φ
INPUT INFORMATION PROVIDING UNIT
$T_{\varphi,1}$
$T_{\varphi,2}$
26104-φ
26102-φ
NATURAL NUMBER SELECTING UNIT
21114-φ
CONTROLLER
a(φ)
26101-φ
a(φ)
NATURAL NUMBER STORAGE
21107-φ
21106-φ
u_φ
FIRST LIST STORAGE
u_φ, u_φ'
FIRST POWER COMPUTING UNIT
u_φ
26112-φ
u_φ'
26111-φ
FIRST COMPUTING UNIT
$z_{\varphi,1}$
FINAL OUTPUT UNIT
DETERMINI NG UNIT
26105-φ
v_φ
26110-φ
SECOND LIST STORAGE
v_φ
26108-φ
SECOND COMPUTING UNIT
$z_{\varphi,2}$
261-φ
u_φ or ⊥

FIG.45


START OF PROCESS PERFORMED BY COMPUTING APPARATUS
SELECT a(φ)
S26100
t=1
S21102
OUTPUT $T_{\varphi,1}$, $T_{\varphi,2}$
S26103
INPUT $z_{\varphi,1}$, $z_{\varphi,2}$
S26104
COMPUTE $u_\varphi = f_\varphi(x_\varphi)^{b(\varphi)}x_{\varphi,1}$
S26105
$u_\varphi' = u_\varphi^{a(\varphi)}$
S21106
COMPUTE $v_\varphi = f_\varphi(x_\varphi)^{a(\varphi)}x_{\varphi,2}$
S26108
ARE u_φ' AND v_φ IN SAME CLASS?
S26110
yes
no
t=T
S21111
no
t=t+1
S21112
yes
OUTPUT "⊥"
S21113
OUTPUT u_φ
S26114
END OF PROCESS PERFORMED BY COMPUTING APPARATUS
END OF PROCESS PERFORMED BY COMPUTING APPARATUS

PROXY COMPUTING SYSTEM, COMPUTING APPARATUS, CAPABILITY PROVIDING APPARATUS, PROXY COMPUTING METHOD, CAPABILITY PROVIDING METHOD, PROGRAM, AND RECORDING MEDIUM

TECHNICAL FIELD

The present invention relates to a technique to perform computation using a result of computation performed on another apparatus.

BACKGROUND ART

Decryption of a ciphertext encrypted using an encryption scheme such as public key cryptography or common key cryptography requires a particular decryption key (see Non-patent literature 1, for example). In one of existing methods for a first apparatus that does not have a decryption key to obtain a result of decryption of a ciphertext, a second apparatus that has a decryption key provides the decryption key to the first apparatus and the first apparatus uses the decryption key to decrypt the ciphertext. In another existing method for the first apparatus to obtain a result of decryption of a ciphertext, the first apparatus provides the ciphertext to the second apparatus and the second apparatus decrypts the ciphertext and provides the result of decryption to the first apparatus.

PRIOR ART LITERATURE

Non-Patent Literature

Non-patent literature 1: Taher Elgamal, A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp. 469-472 or CRYPTO 84, pp. 10-18, Springer-Verlag

SUMMARY OF THE INVENTION

Problems to be Solved by the Invention

However, in the method in which the second apparatus provides a decryption key to the first apparatus, the decryption key needs to be taken out from the second apparatus to the outside, which poses security concerns. On the other hand, in the method in which the first apparatus provides a ciphertext to the second apparatus and the second apparatus decrypts the ciphertext, the first apparatus cannot verify the validity of decryption performed by the second apparatus. These problems can be generalized to other processing besides decryption. That is, there was not a technique for the second apparatus to provide only a computing capability to the first apparatus without leaking secrete information so that the first apparatus uses the computing capability to correctly perform computations.

Means to Solve the Problems

According to the present invention, a computing apparatus outputs first input information τ_1 and second input information τ_2 that are elements of a group H and correspond to a ciphertext x, a capability providing apparatus uses the first input information τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability to provide the result of the computation as first output information z_1 and uses the second input information τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability to provide the result of the computation as second output information z_2, the computing apparatus generates a computation result $u=f(x)^b x_1$ from the first output information z_1 and generates a computation result $v=f(x)^a x_2$ from the second output information z_2 and, when the computation results u and v satisfy $u^a=v^b$, outputs $u^{b'}v^{a'}$ for integers a' and b' that satisfy $a'a+b'b=1$, where G and H are groups, f(x) is a decryption function for decrypting the ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, and a and b are natural numbers relatively prime to each other.

Effects of the Invention

According to the present invention, the capability providing apparatus provides only a computing capability to the computing apparatus without leaking secret information and the computing apparatus can use the computing capability to correctly perform computations.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating a configuration of a proxy computing system of an embodiment;

FIG. **2** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **3** is a block diagram illustrating a configuration of a capability providing apparatus of an embodiment;

FIG. **4** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **5** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **6** is a flowchart illustrating a process performed by a computing apparatus of an embodiment;

FIG. **7** is a flowchart illustrating a process performed by a capability providing apparatus of an embodiment;

FIG. **8** is a flowchart illustrating a process at step S**2103** (S**3103**);

FIG. **9** is a flowchart illustrating a process at step S**4103**;

FIG. **10** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **11** is a flowchart illustrating a process performed by a computing apparatus of an embodiment;

FIG. **12** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **13** is a block diagram illustrating a configuration of a capability providing apparatus of an embodiment;

FIG. **14** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **15** is a flowchart illustrating a process performed by a computing apparatus of an embodiment;

FIG. **16** is a flowchart illustrating an example of a process at step S**6103**;

FIG. **17** is a flowchart illustrating a process performed by a capability providing apparatus of an embodiment;

FIG. **18** is a block diagram illustrating a configuration of a proxy computing system of an embodiment;

FIG. **19** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **20** is a block diagram illustrating a configuration of a capability providing apparatus of an embodiment;

FIG. **21** is a block diagram illustrating a configuration of a decryption control apparatus of an embodiment;

FIG. **22** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **23** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **24** is a flowchart illustrating an encryption process of an embodiment;

FIG. **25** is a flowchart illustrating a decryption process of an embodiment;

FIG. **26** is a flowchart illustrating a decryption capability providing process of an embodiment;

FIG. **27** is a flowchart illustrating a process at step S**12103** (S**13103**);

FIG. **28** is a flowchart illustrating a process at step S**14103**;

FIG. **29** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **30** is a flowchart illustrating a decryption process of an embodiment;

FIG. **31** is a block diagram illustrating a configuration of a proxy computing system of an embodiment;

FIG. **32** is a block diagram illustrating a configuration of a computing apparatus of an embodiment;

FIG. **33** is a block diagram illustrating a configuration of a capability providing apparatus of an embodiment;

FIG. **34** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **35** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **36** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **37** is a flowchart illustrating a process performed by a computing apparatus of an embodiment;

FIG. **38** is a flowchart illustrating a process performed by a capability providing apparatus of an embodiment;

FIG. **39** is a flowchart illustrating a process at step S**22103** (S**23103**);

FIG. **40** is a flowchart illustrating a process at step S**24103**;

FIG. **41** is a flowchart illustrating a process at step S**25103**;

FIG. **42** is a block diagram illustrating a configuration of an input information providing unit of an embodiment;

FIG. **43** is a flowchart illustrating a process at step S**27103**;

FIG. **44** is a block diagram illustrating a configuration of a computing apparatus of an embodiment; and

FIG. **45** is a flowchart illustrating a process performed by a computing apparatus of an embodiment.

DETAILED DESCRIPTION OF THE EMBODIMENTS

Embodiments of the present invention will be described below with reference to drawings.

First Embodiment

A first embodiment of the present invention will be described.

<Configuration>

As illustrated in FIG. **1**, a proxy computing system **1** of the first embodiment includes, for example, a computing apparatus **11** which does not have a decryption key and a capability providing apparatus **12** which has a decryption key. The computing apparatus **11** requests the capability providing apparatus **12** to provide a capability of decrypting a ciphertext and uses the capability of decryption provided from the capability providing apparatus **12** to decrypt the ciphertext. The computing apparatus **11** and the capability providing apparatus **12** are configured to be able to exchange information. For example, the computing apparatus **11** and the capability providing apparatus **12** are configured to be able to exchange information through a transmission line, a network, a portable recording medium and/or other medium.

As illustrated in FIG. **2**, the computing apparatus **11** of the first embodiment includes, for example, a natural number storage **1101**, a natural number selecting unit **1102**, an integer computing unit **1103**, an input information providing unit **1104**, a first computing unit **1105**, a first power computing unit **1106**, a first list storage **1107**, a second computing unit **1108**, a second power computing unit **1109**, a second list storage **1110**, a determining unit **1111**, a final output unit **1112**, and a controller **1113**. Examples of the computing apparatus **11** include a device having a computing function and a memory function, such as a card reader-writer apparatus and a mobile phone, and a well-known or specialized computer that includes a CPU (central processing unit) and a RAM (random-access memory) in which a special program is loaded.

As illustrated in FIG. **3**, the capability providing apparatus **12** of the first embodiment includes, for example, a first output information computing unit **1201**, a second output information computing unit **1202**, a key storage **1204** and a controller **1205**. Examples of the capability providing apparatus **12** include a tamper-resistant module such as an IC card and an IC chip, a device having computing and memory functions, such as a mobile phone, and a well-known or specialized computer including a CPU and a RAM in which a special program is loaded.

<Processes>

Processes of this embodiment will be described below. For the processes, let G and H be groups (for example commutative groups), f(x) be a decryption function for decrypting a ciphertext x, which is an element of the group H, with a particular decryption key s to obtain an element of the group G, generators of the groups G and H be μ_g and μ_h, respectively, X_1 and X_2 be random variables having values in the group G, x_1 be a realization of the random variable X_1, and x_2 be a realization of the random variable X_2. It is assumed here that a plurality of pairs of natural numbers a, b that are relatively prime to each other (a, b) are stored in the natural number storage **1101** of the computing apparatus **11**. The term "natural number" means an integer greater than or equal to 0. Let I be a set of pairs of relatively prime natural numbers that are less than the order of the group G, then it can be considered that pairs (a, b) of natural numbers a, b corresponding to a subset S of I are stored in the natural number storage **1101**. It is also assumed that a particular decryption key s is stored in the key storage **1204** of the capability providing apparatus **12** in a secure manner. Processes of the computing apparatus **11** are performed under the control of the controller **1113** and processes of the capability providing apparatus **12** are performed under the control of the controller **1205**.

As illustrated in FIG. **6**, first, the natural number selecting unit **1102** of the computing apparatus **11** (FIG. **2**) randomly reads one pair of natural numbers (a, b) from a plurality of pairs of natural numbers (a, b) stored in the natural number storage **1101**. At least part of information on the read pair of natural numbers (a, b) is sent to the integer computing unit

1103, the input information providing unit 1104, the first power computing unit 1106, and the second power computing unit 1109 (step S1100).

The integer computing unit 1103 uses the sent pair of natural numbers (a, b) to compute integers a', b' that satisfy the relation a'a+b'b=1. Since the natural numbers a and b are relatively prime to each other, the integers a' and b' that satisfy the relation a'a+b'b=1 definitely exist. Information on the pair of natural numbers (a', b') is sent to the final output unit 1112 (step S1101).

The controller 1113 set t=1 (step S1102).

The input information providing unit 1104 generates and outputs first input information τ_1 and second input information τ_2 which are elements of the group H and each of which corresponds to the input ciphertext x. Preferably, each of the first input information τ_1 and the second input information τ_2 is information whose relation with the ciphertext x is scrambled. This enables the computing apparatus 11 to conceal the ciphertext x from the capability providing apparatus 12. Preferably, the first input information τ_1 of this embodiment further corresponds to the natural number b selected by the natural number selecting unit 1102 and the second input information τ_2 further corresponds to the natural number a selected by the natural number selecting unit 1102. This enables the computing apparatus 11 to evaluate the decryption capability provided by the capability providing apparatus 12 with a high degree of accuracy (step S1103).

As exemplified in FIG. 7, the first input information τ_1 is input in the first output information computing unit 1201 of the capability providing apparatus 12 (FIG. 3) and the second input information τ_2 is input in the second output information computing unit 1202 (step S1200).

The first output information computing unit 1201 uses the first input information τ_1 and the decryption key s stored in the key storage 1204 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets the result of the computation as first output information z_1 (step S1201). The second output information computing unit 1202 uses the second input information τ_2 and the decryption key s stored in the key storage 1204 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets the result of the computation as second output information z_2 (step S1202). Note that the "certain probability" is a probability less than 100%. An example of the "certain probability" is a nonnegligible probability and an example of the "nonnegligible probability" is a probability greater than or equal to $1/\psi(k)$, where $\psi(k)$ is a polynomial that is a weakly increasing function (non-decreasing function) for a security parameter k. That is, the first output information computing unit 1201 and the second output information computing unit 1202 output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit 1201 may or may not be $f(\tau_1)$ and the result of the computation by the second output information computing unit 1202 may or may not be $f(\tau_2)$.

The first output information computing unit 1201 outputs the first output information z_1 and the second output information computing unit 1202 outputs the second output information z_2 (step S1203).

Returning to FIG. 6, the first output information z_1 is input in the first computing unit 1105 of the computing apparatus 11 (FIG. 2) and the second output information z_2 is input in the second computing unit 1108. The first output information z_1 and the second output information z_2 are equivalent to the decryption capability provided by the capability providing apparatus 12 to the computing apparatus 11 (step S1104).

The first computing unit 1105 generates a computation result $u=f(x)^b x_1$ from the first output information z_1. Here, generating (computing) $f(x)^b x_1$ means computing a value of a formula defined as $f(x)^b x_1$. Any intermediate computation method may be used, provided that the value of the formula $f(x)^b x_1$ can be eventually computed. The same applies to computations of the other formulae that appear herein. In the first embodiment, a computation defined at a group is expressed multiplicatively. That is, "α^b" for $\alpha \in G$ means that a computation defined at the group G is applied b times to α and "$\alpha_1\alpha_2$" for α_1, $\alpha_2 \in G$ means that a computation defined at the group G is performed on operands α_1 and α_2 (the same applies to second to fifth embodiments described later). The result u of the computation is sent to the first power computing unit 1106 (step S1105).

The first power computing unit 1106 computes $u'=u^a$. The pair of the result u of the computation and u' computed on the basis of the result of the computation, (u, u'), is stored in the first list storage 1107 (step S1106).

The determining unit 1111 determines whether or not there is one that satisfies u'=v' among the pairs (u, u') stored in the first list storage 1107 and the pairs (v, v') stored in the second list storage 1110 (step S1107). If no pair (v, v') is stored in the second list storage 1110, the process at step S1107 is omitted and a process at step S1108 is performed. If there is one that satisfies u'=v', the process proceeds to step S1114; if there is not one that satisfies u'=v', the process proceeds to step S1108.

At step S1108, the second computing unit 1108 generates a computation result $v=f(x)^a x_2$ from the second output information z_2. The result v of the computation is sent to the second power computing unit 1109 (step S1108).

The second power computing unit 1109 computes $v'=v^b$. The pair (v, v') of the result v of the computation and v' computed on the basis of the computation result is stored in the second list storage 1110 (step S1109).

The determining unit 1111 determines whether or not there is one that satisfies u'=v' among the pairs (u, u') stored in the first list storage 1107 and the pairs (v, v') stored in the second list storage 1110 (step S1110). If there is one that satisfies u'=v', the process proceeds to step S1114. If there is not one that satisfies u'=v', the process proceeds to step S1111.

At step S1111, the controller 1113 determines whether or not $t=T_{max}$ (step S1111). Here, T_{max} is a predetermined natural number. If $t=T_{max}$, the controller 1113 outputs information indicating that the computation is impossible, for example the symbol "$\perp$" (step S1113) and the process ends. If not $t=T_{max}$, the controller 1113 increments t by 1, that is, sets t=t+1 (step S1112) and the process returns to step S1103.

The information indicating the computation is impossible (the symbol "$\perp$" in this example) means that the level of reliability that the capability providing apparatus 12 correctly performs computation is lower than a criterion defined by T_{max}. In other words, the capability providing apparatus 12 was unable to perform a correct computation in T_{max} trials.

At step S1114, the final output unit 1112 uses u and v that correspond to u' and v' that are determined to satisfy u'=v' to calculate and output $u^{b'}v^{a'}$ (step S1114). The $u^{b'}v^{a'}$ thus computed will be a result f(x) of decryption of the ciphertext x with the particular decryption key s with a high probability (the reason why $u^{b'}v^{a'}=f(x)$ with a high probability will be described later). Therefore, the process described above is repeated multiple times and the value obtained with the

highest frequency among the values obtained at step S1114 can be provided as the result of decryption. As will be described, $u^{b'}v^{a'}=f(x)$ can result with an overwhelming probability, depending on settings. In that case, the value obtained at step S1114 can be directly provided as the result of decryption.

<<Reason Why $u^{b'}v^{a'}=f(x)$ with a High Probability>>

Let X be a random variable having a value in the group G For $w\in G$, an entity that returns wx' corresponding to a sample x' according to the random variable X in response to each request is called a sampler having an error X for w.

For $w\in G$, an entity that returns $w^a x'$ corresponding to a sample x' according to a random variable X whenever a natural number a is given is called a randomizable sampler having an error X for w. The randomizable sampler functions as the sampler if used with a=1.

The combination of the input information providing unit **1104**, the first output information computing unit **1201** and the first computing unit **1105** of this embodiment is a randomizable sampler having an error X_1 for f(x) (referred to as the "first randomizable sampler") and the combination of the input information providing unit **1104**, the second output information computing unit **1202** and the second computing unit **1108** is a randomizable sampler having an error X_2 for f(x) (referred to as the "second randomizable sampler").

The inventor has found that if u'=v' holds, that is, if $u^a=v^b$ holds, it is highly probable that the first randomizable sampler has correctly computed $u=f(x)^b$ and the second randomizable sampler has correctly computed $v=f(x)^a$ (x_1 and x_2 are identity elements e_g of the group G). For simplicity of explanation, this will be proven in a fifth embodiment.

When the first randomizable sampler correctly computes $u=f(x)^b$ and the second randomizable sampler correctly computes $v=f(x)^a$ (when x_1 and x_2 are identity elements e_g of the group G), then $u^{b'}v^{a'}=(f(x)^b x_1)^{b'}(f(x)^a x_2)^{a'}=(f(x)^b e_g)^{b'}(f(x)^a e_g)^{a'}=f(x)^{bb'}e_g{}^{b'}f(x)^{aa'}e_g{}^{a'}=f(x)^{(bb'+aa')}=f(x)$.

For $(q_1, q_2)\in I$, a function π_i is defined by $\pi_i(q_1, q_2)=q_i$ for each of i=1, 2. Let $L=\min(\#\pi_1(S), \#\pi_2(S))$, where $\#\cdot$ is the order of a set $\cdot$. If the group G is a cyclic group or a group whose order is difficult to compute, it can be expected that the probability that an output other than "⊥" of the computing apparatus **11** is not f(x) is at most approximately $T_{max}{}^2 L/\#S$ within a negligible error. If L/#S is a negligible quantity and T_{max} is a quantity approximately equal to a polynomial order, the computing apparatus **11** outputs a correct f(x) with an overwhelming probability. An example of S that results in a negligible quantity of L/#S is $S=\{(1, d)|d\in[2, |G|-1]\}$, for example.

Second Embodiment

A proxy computing system of a second embodiment is an example that embodies the first randomizable sampler and the second randomizable sampler described above. The following description will focus on differences from the first embodiment and repeated description of commonalities with the first embodiment will be omitted. In the following description, elements labeled with the same reference numerals have the same functions and the steps labeled with the same reference numerals represent the same processes.

<Configuration>

As illustrated in FIG. **1**, the proxy computing system **2** of the second embodiment includes a computing apparatus **21** in place of the computing apparatus **11** and a capability providing apparatus **22** in place of the capability providing apparatus **12**.

As illustrated in FIG. **2**, the computing apparatus **21** of the second embodiment includes, for example, a natural number storage **1101**, a natural number selecting unit **1102**, an integer computing unit **1103**, an input information providing unit **2104**, a first computing unit **2105**, a first power computing unit **1106**, a first list storage **1107**, a second computing unit **2108**, a second power computing unit **1109**, a second list storage **1110**, a determining unit **1111**, a final output unit **1112** and a controller **1113**. As illustrated in FIG. **4**, the input information providing unit **2104** of this embodiment includes, for example, a first random number generator **2104***a*, a first input information computing unit **2104***b*, a second random number generator **2104***c*, and a second input information computing unit **2104***d*.

As illustrated in FIG. **3**, the capability providing apparatus **22** of the second embodiment includes, for example, a first output information computing unit **2201**, a second output information computing unit **2202**, a key storage **1204**, and a controller **1205**.

<Processes>

Processes of this embodiment will be described below. In the second embodiment, a decryption function f(x) is a homomorphic function, a group H is a cyclic group, and a generator of the group H is μ_h, the order of the group H is K_H, and $v=f(\mu_h)$. The rest of the assumptions are the same as those in the first embodiment, except that the computing apparatus **11** is replaced with the computing apparatus **21** and the capability providing apparatus **12** is replaced with the capability providing apparatus **22**.

As illustrated in FIGS. **6** and **7**, a process of the second embodiment is the same as the process of the first embodiment except that steps S1103 through S1105, S1108, and S1200 through S1203 of the first embodiment are replaced with steps S2103 through S2105, S2108, and S2200 through S2203, respectively. In the following, only processes at steps S2103 through S2105, S2108, and S2200 through S2203 will be described.

<<Process at Step S2103>>

The input information providing unit **2104** of the computing apparatus **21** (FIG. **2**) generates and outputs first input information τ_1 and second input information τ_2 which are elements of the group H and each of which corresponds to an input ciphertext x (Step S2103 of FIG. **6**). A process at step S2103 of this embodiment will be described with reference to FIG. **8**.

The first random number generator **2104***a* (FIG. **4**) generates a uniform random number r_1 that is a natural number greater than or equal to 0 and less than K_H. The generated random number r_1 is sent to the first input information computing unit **2104***b* and the first computing unit **2105** (step S2103*a*). The first input information computing unit **2104***b* uses the input random number r_1, the ciphertext x and a natural number b to compute first input information $\tau_1=\mu_h{}^{r1}x^b$ (step S2103*b*). Here, the superscript r1 on μ_h represents r_1. When a notational $\alpha^{\beta\gamma}$ is used herein in this way, $\beta\gamma$ represents β_γ, namely β with subscript γ, where α is a first letter, β is a second letter, and γ is a number.

The second random number generator **2104***c* generates a uniform random number r_2 that is a natural number greater than or equal to 0 and less than K_H. The generated random number r_2 is sent to the second input information computing unit **2104***d* and the second computing unit **2108** (step S2103*c*). The second input information computing unit

2104*d* uses the input random number r_2, the ciphertext x, and a natural number a to compute second input information $\tau_2=\mu_h^{r2}x^a$ (step S**2103***d*).

The first input information computing unit **2104***b* and the second input information computing unit **2104***d* output the first input information τ_1 and the second input information τ_2 thus generated (step S**2103***e*). Note that the first input information τ_1 and the second input information τ_2 in this embodiment are information whose relation with the ciphertext x is scrambled using random numbers r_1, r_2, respectively. This enables the computing apparatus **21** to conceal the ciphertext x from the capability providing apparatus **22**. The first input information τ_1 of this embodiment further corresponds to the natural number b selected by the natural number selecting unit **1102** and the second input information τ_2 further corresponds to the natural number a selected by the natural number selecting unit **1102**. This enables the computing apparatus **21** to evaluate the decryption capability provided by the capability providing apparatus **22** with a high degree of accuracy.

<<Processes at Steps S**2200** Through S**2203**>>

As illustrated in FIG. **7**, first, the first input information $\tau_1=\mu_h^{r1}x^b$ is input in the first output information computing unit **2201** of the capability providing apparatus **22** (FIG. **3**) and the second input information $\tau_2=\mu_h^{r2}x^a$ is input in the second output information computing unit **2202** (step S**2200**).

The first output information computing unit **2201** uses the first input information $\tau_1=\mu_h^{r1}x^b$ and a decryption key s stored in the key storage **1204** to correctly compute $f(\mu_h^{r1}x^b)$ with a probability greater than a certain probability and sets the result of the computation as first output information z_1. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **2201** may or may not be $f(\mu_h^{r1}x^b)$ (step S**2201**).

The second output information computing unit **2202** uses the second input information $\tau_2=\mu_h^{r2}x^a$ and the decryption key s stored in the key storage **1204** to correctly compute $f(\mu_h^{r2}x^a)$ with a probability greater than a certain probability and sets the result of the computation as second output information z_2. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **2202** may or may not be $f(\mu_h^{r2}x^a)$ (step S**2202**). The first output information computing unit **2201** outputs the first output information z_1 and the second output information computing unit **2202** outputs the second output information z_2 (step S**2203**).

<<Processes at Steps S**2104** and S**2105**>>

Returning to FIG. **6**, the first output information z_1 is input in the first computing unit **2105** of the computing apparatus **21** (FIG. **2**) and the second output information z_2 is input in the second computing unit **2108**. The first output information z_1 and the second output information z_2 are equivalent to the decryption capability provided by the capability providing apparatus **22** to the computing apparatus **21** (step S**2104**).

The first computing unit **2105** uses the input random number r_1 and the first output information z_1 to compute z_1v^{-r1} and sets the result of the computation as u. The result u of the computation is sent to the first power computing unit **1106**. Here, $u=z_1v^{-r1}=f(x)^bx_1$. That is, z_1v^{-r1} is an output of a randomizable sampler having an error X_1 for f(x). The reason will be described later (step S**2105**).

<<Process at Step S**2108**>>

The second computing unit **2108** uses the input random number r_2 and the second output information z_2 to compute z_2v^{-r2} and sets the result of the computation as v. The result v of the computation is sent to the second power computing unit **1109**. Here, $v=z_2v^{-r2}=f(x)^ax_2$. That is, z_2v^{-r2} is an output of a randomizable sampler having an error X_2 for f(x). The reason will be described later (step S**2108**).

<<Reason Why z_1v^{-r1} and z_2v^{-r2} are Outputs of Randomizable Samplers Having Errors X_1 and X_2, Respectively, for f(x)>>

Let c be a natural number, R and R' be random numbers, and $B(\mu_h^Rx^c)$ be the result of computation performed by the capability providing apparatus **22** using $\mu_h^Rx^c$. That is, the results of computations that the first output information computing unit **2201** and the second output information computing unit **2202** return to the computing apparatus **21** are $z=B(\mu_h^Rx^c)$. A random variable X that has a value in the group G is defined as $X=B(\mu_h^{R'})f(\mu_h^{R'})^{-1}$.

Then, $zv^{-R}=B(\mu_h^Rx^c)f(\mu_h)^{-R}=Xf(\mu_h^Rx^c)f(\mu_h)^{-R}=Xf(\mu_h)^Rf(x)^cf(\mu_h)^{-R}=f(x)^cX$. That is, zv^{-R} is an output of a randomizable sampler having an error X for f(x).

The expansion of formula given above uses the properties such that $X=B(\mu_h^{R'})f(\mu_h^{R'})^{-1}=B(\mu_h^Rx^c)f(\mu_h^Rx^c)^{-1}$ and that $B(\mu_h^Rx^c)=Xf(\mu_h^Rx^c)$. The properties are based on the fact that the function f(x) is a homomorphic function and R and R' are random numbers.

Therefore, considering that a and b are natural numbers and r_1 and r_2 are random numbers, z_1v^{-r1} and z_2v^{-r2} are, likewise, outputs of randomizable samplers having errors X_1 and X_2, respectively, for f(x).

Third Embodiment

A third embodiment is a variation of the second embodiment and computes a value of u or v by using samplers described above when a=1 or b=1. The amounts of computations performed by samplers are in general smaller than the amounts of computations by randomizable samplers. Using samplers instead of randomizable samplers for computations when a=1 or b=1 can reduce the amounts of computations by a proxy computing system. The following description will focus on differences from the first and second embodiments and repeated description of commonalities with the first and second embodiments will be omitted.

<Configuration>

As illustrated in FIG. **1**, a proxy computing system **3** of the third embodiment includes a computing apparatus **31** in place of the computing apparatus **21** and a capability providing apparatus **32** in place of the capability providing apparatus **22**.

As illustrated in FIG. **2**, the computing apparatus **31** of the third embodiment includes, for example, a natural number storage **1101**, a natural number selecting unit **1102**, an integer computing unit **1103**, an input information providing unit **3104**, a first computing unit **2105**, a first power computing unit **1106**, a first list storage **1107**, a second computing unit **2108**, a second power computing unit **1109**, a second list storage **1110**, a determining unit **1111**, a final output unit **1112**, a controller **1113**, and a third computing unit **3109**.

As illustrated in FIG. **3**, the capability providing apparatus **32** of the third embodiment includes, for example, a first output information computing unit **2201**, a second output information computing unit **2202**, a key storage **1204**, a controller **1205**, and a third output information computing unit **3203**.

<Processes>

Processes of this embodiment will be described below. Differences from the second embodiment will be described.

As illustrated in FIGS. **6** and **7**, a process of the third embodiment is the same as the process of the second embodiment except that steps **S2103** through **S2105**, **S2108**, and **S2200** through **S2203** of the second embodiment are replaced with steps **S3103** through **S3105**, **S3108**, **S3200** through **S3203**, and **S3205** through **3209**, respectively. The following description will focus on processes at steps **S3103** through **S3105**, **S3108**, **S3200** through **S3203**, and **S3205** through **S3209**.

<<Process at Step S3103>>

The input information providing unit **3104** of the computing apparatus **31** (FIG. **2**) generates and outputs first input information τ_1 and second input information τ_2 which are elements of a group H and each of which corresponds to an input ciphertext x (step **S3103** of FIG. **6**).

A process at step **S3103** of this embodiment will be described below with reference to FIG. **8**.

The controller **1113** (FIG. **2**) controls the input information providing unit **3104** according to natural numbers (a, b) selected by the natural number selecting unit **1102**.

Determination is made by the controller **1113** as to whether b is equal to 1 (step **S3103***a*). If it is determined that $b\neq1$, the processes at steps **S2103***a* and **2103***b* described above are performed and the process proceeds to step **S3103***g*.

On the other hand, if it is determined at step **S3103***a* that b=1, the third random number generator **3104***e* generates a random number r_3 that is a natural number greater than or equal to 0 and less than K_H. The generated random number r_3 is sent to the third input information computing unit **3104***f* and the third computing unit **3109** (step **S3103***b*). The third input information computing unit **3104***f* uses the input random number r_3 and a ciphertext x to compute x^{r3} and sets x^{r3} as first input information τ_1 (step **S3103***c*). Then the process proceeds to step **S3103***g*.

At step **S3103***g*, determination is made by the controller **1113** as to whether a is equal to 1 (step **S3103***g*). If it is determined that $a\neq1$, the processes at steps **S2103***c* and **S2103***d* described above are performed.

On the other hand, if it is determined at step **S3103***g* that a=1, the third random number generator **3104***e* generates a random number r_3 that is a natural number greater than or equal to 0 and less than K_H. The generated random number r_3 is sent to the third input information computing unit **3104***f* (step **S3103***h*). The third input information computing unit **3104***f* uses the input random number r_3 and the ciphertext x to compute x^{r3} and sets x^{r3} as second input information τ_2 (step **S3103***i*).

The first input information computing unit **2104***b*, the second input information computing unit **2104***d*, and the third input information computing unit **3104***f* output the first input information τ_1 and the second input information τ_2 thus generated along with information on the corresponding natural numbers (a, b) (step **S3103***e*). Note that the first input information τ_1 and the second input information τ_2 in this embodiment are information whose relation with the ciphertext x is scrambled using random numbers r_1, r_2 and r_3. This enables the computing apparatus **31** to conceal the ciphertext x from the capability providing apparatus **32**.

<<Processes at S3200 through S3203 and S3205 through S3209>>

Processes at **S3200** through **S3203** and **S3205** through **S3209** of this embodiment will be described below with reference to FIG. **7**.

The controller **1205** (FIG. **3**) controls the first output information computing unit **2201**, the second output information computing unit **2202** and the third output information computing unit **3203** according to input natural numbers (a, b).

Under the control of the controller **1205**, the first input information $\tau_1=\mu_h{}^{r1}x^b$ when $b\neq1$ is input in the first output information computing unit **2201** of the capability providing apparatus **32** (FIG. **3**) and the second input information $\tau_2=\mu_h{}^{r2}x^a$ when $a\neq1$ is input in the second output information computing unit **2202**. The first input information $\tau_1=x^{r3}$ when b=1 and the second input information $\tau_2=x^{r3}$ when a=1 are input in the third output information computing unit **3203** (step **S3200**).

Determination is made by the controller **1113** as to whether b is equal to 1 (step **S3205**). If it is determined that $b\neq1$, the process at step **S2201** described above is performed. Then, determination is made by the controller **1113** as to whether a is equal to 1 (step **S3208**). If it is determined that $a\neq1$, the process at step **S2202** described above is performed and then the process proceeds to step **S3203**.

On the other hand, if it is determined at step **S3208** that a=1, the third output information computing unit **3203** uses the second input information $\tau_2=x^{r3}$ to correctly compute $f(x^{r3})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information z_3. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **3203** may or may not be $f(x^{r3})$ (step **S3209**). Then the process proceeds to step **S3203**.

If it is determined at step **S3205** that b=1, the third output information computing unit **3203** uses the first input information $\tau_1=x^{r3}$ to correctly compute $f(x^{r3})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information z_3. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **3203** may or may not be $f(x^{r3})$ (step **S3206**).

Then, determination is made by the controller **1113** as to whether a is equal to 1 (step **S3207**). If it is determined that a=1, the process proceeds to step **S3203**; if it is determined that $a\neq1$, the process proceeds to step **S2202**.

At step **S3203**, the first output information computing unit **2201**, which has generated the first output information z_1, outputs the first output information z_1, the second output information computing unit **2202**, which has generated the second output information z_2, outputs the second output information z_2, and the third output information computing unit **3202**, which has generated the third output information z_3, outputs the third output information z_3 (step **S3203**).

<<Processes at Steps S3104 and S3105>>

Returning to FIG. **6**, under the control of the controller **1113**, the first output information z_1 is input in the first computing unit **2105** of the computing apparatus **31** (FIG. **2**), the second output information z_2 is input in the second computing unit **2108**, and the third output information z_3 is input in the third computing unit **3109** (step **S3104**).

If $b\neq1$, the first computing unit **2105** performs the process at step **S2105** described above to generate u; if b=1, the third computing unit **3109** computes $z_3{}^{1/r3}$ and sets the result of the computation as u. The result u of the computation is sent to the first power computing unit **1106**. Here, if b=1, then $u=z_3{}^{1/r3}=f(x)x_3$. That is, $z_3{}^{1/r3}$ serves as a sampler having an error X_3 for f(x). The reason will be described later (step **S3105**).

<<Process at Step S3108>>

If $a \neq 1$, the second computing unit **2108** performs the process at S2108 described above to generate v; if a=1, the third computing unit **3109** computes $z_3^{1/r3}$ and sets the result of the computation as v. The result v of the computation is sent to the second power computing unit **1109**. Here, if a=1, then $v=z_3^{1/r3}=f(x)x_3$. That is, $z_3^{1/r3}$ serves as a sampler having an error X_3 for f(x). The reason will be described later (step S3108).

Note that if $z_3^{1/r3}$, that is, the radical root of z_3, is hard to compute, u and/or v may be calculated as follows. The third computing unit **3109** may store each pair of random number r_3 and z_3 computed on the basis of the random number r_3 in a storage, not depicted, in sequence as (α_1, β_1), (α_2, β_2), . . . , (α_m, β_m), ... Here, m is a natural number. The third computing unit **3109** may compute $\gamma_1, \gamma_2, \ldots, \gamma_m$ that satisfies $\gamma_1\alpha_1+\gamma_2\alpha_2+ \ldots +\gamma_m\alpha_m=1$ when the least common multiple of $\alpha_1, \alpha_2, \ldots, \alpha_m$ is 1, where $\gamma_1, \gamma_2, \ldots, \gamma_m$ are integers. The third computing unit **3109** may then use the resulting $\gamma_1, \gamma_2, \ldots, \gamma_m$ to compute $\Pi_{i=1}^{m}\beta_i^{\gamma i}=\beta_1^{\gamma 1}\beta_2^{\gamma 2} \ldots \beta_m^{\gamma m}$ and may set the results of the computation as u and/or v.

<<Reason why $z_3^{1/r3}$ Serves as a Sampler Having an Error X_3 for f(x)>>

Let R be a random number and $B(x^R)$ be the result of computation performed by the capability providing apparatus **32** using x^R. That is, let $z=B(x^R)$ be the results of computations returned by the first output information computing unit **2201**, the second output information computing unit **2202**, and the third output information computing unit **3203** to the computing apparatus **31**. Furthermore, a random variable X having a value in the group G is defined as $X=B(x^R)^{1/R}f(x)^{-1}$.

Then, $z^{1/R}=B(x^R)^{1/R}=Xf(x)=f(x)X$. That is, $z^{1/R}$ serves as a sampler having an error X for f(x).

The expansion of formula given above uses the properties such that $X=B(x^R)^{1/R}f(x^R)^{-1}$ and that $B(x^R)^{1/R}=Xf(x^R)$. The properties are based on the fact that R is a random number.

Therefore, considering that r_3 is a random number, $z^{1/R}$ serves as a sampler having an error X_3 for f(x) likewise.

Fourth Embodiment

A proxy computing system of a fourth embodiment is another example that embodies the first and second randomizable samplers described above. Specifically, the proxy computing system embodies an example of first and second randomizable samplers in the case where H=G×G, the decryption function f(x) is a decryption function of ElGamal encryption, that is, $f(c_1, c_2)=c_1c_1^{-s}$ for a decryption key s and a ciphertext $x=(c_1, c_2)$. The following description will focus on differences from the first embodiment and repeated description of commonalities with the first embodiment will be omitted.

As illustrated in FIG. **1**, the proxy computing system **4** of the fourth embodiment includes a computing apparatus **41** in place of the computing apparatus **11** and a capability providing apparatus **42** in place of the capability providing apparatus **12**.

As illustrated in FIG. **2**, the computing apparatus **41** of the fourth embodiment includes, for example, a natural number storage **1101**, a natural number selecting unit **1102**, an integer computing unit **1103**, an input information providing unit **4104**, a first computing unit **4105**, a first power computing unit **1106**, a first list storage **1107**, a second computing unit **4108**, a second power computing unit **1109**, a second list storage **1110**, a determining unit **1111**, a final output unit **1112**, and a controller **1113**. As illustrated in FIG. **5**, the input information providing unit **4104** of this embodiment includes, for example, a fourth random number generator **4104***a*, a fifth random number generator **4104***b*, a first input information computing unit **4104***c*, a sixth random number generator **4104***d*, a seventh random number generator **4104***e*, and a second input information computing unit **4104***f*. The first input information computing unit **4104***c* includes, for example, a fourth input information computing unit **4104***ca* and a fifth input information computing unit **4104***cb*. The second input information computing unit **4104***f* includes, for example, a sixth input information computing unit **4104***fa* and a seventh input information computing unit **4104***fb*.

As illustrated in FIG. **3**, the capability providing apparatus **42** of the fourth embodiment includes, for example, a first output information computing unit **4201**, a second output information computing unit **4202**, a key storage **1204**, and a controller **1205**.

<Processes>

Processes of this embodiment will be described below. In the fourth embodiment, it is assumed that a group H is the direct product group G×G of a group the group G, the group G is a cyclic group, a ciphertext $x=(c_1, c_2)\in H$, $f(c_1, c_2)$ is a homomorphic function, a generator of the group G is μ_g, the order of the group G is K_G, a pair of a ciphertext $(V, W)\in H$ and a text $f(V, W)=Y\in G$ decrypted from the ciphertext for the same decryption key s is preset in the computing apparatus **41** and the capability providing apparatus **42**, and the computing apparatus **41** and the capability providing apparatus **42** can use the pair.

As illustrated in FIGS. **6** and **7**, a process of the fourth embodiment is the same as the process of the first embodiment except that steps S1103 through S1105, S1108, and S1200 through S1203 of the first embodiment are replaced with steps S4103 through S4105, S4108, and S4200 through S4203, respectively. In the following, only processes at steps S4103 through S4105, S4108, and S4200 through S4203 will be described.

<<Process at Step S4103>>

The input information providing unit **4104** of the computing apparatus **41** (FIG. **2**) generates and outputs first input information τ_1 which is an element of the group H and corresponds to an input ciphertext $x=(c_1, c_2)$ and second input information τ_2 which is an element of the group H and corresponds to the ciphertext $x=(c_1, c_2)$ (step S4103 of FIG. **6**). A process at step S4103 of this embodiment will be described below with reference to FIG. **9**.

The fourth random number generator **4104***a* (FIG. **5**) generates a uniform random number r_4 that is a natural number greater than or equal to 0 and less than K_G. The generated random number r_4 is sent to the fourth input information computing unit **4104***ca*, the fifth input information computing unit **4104***cb*, and the first computing unit **4105** (step S4103*a*). The fifth random number generator **4104***b* generates a uniform random number r_5 that is a natural number greater than or equal to 0 and less than K_G. The generated random number r_5 is sent to the fifth input information computing unit **4104***cb* and the first computing unit **4105** (step S4103*b*).

The fourth input information computing unit **4104***ca* uses a natural number b selected by the natural number selecting unit **1102**, c_2 included in the ciphertext x, and the random number r_4 to compute fourth input information $c_2^bW^4$ (step S4103*c*). The fifth input information computing unit **4104***cb* uses the natural number b selected by the natural number

selecting unit **1102**, c_1 included in the ciphertext x, and random numbers r_4 and r_5 to compute fifth input information $c_1{}^{b}V^{r4}\mu_g{}^{r5}$ (step S**4103***d*).

The sixth random number generator **4104***d* generates a uniform random number r_6 that is a natural number greater than or equal to 0 and less than K_G. The generated random number r_6 is sent to the sixth input information computing unit **4104***fa*, the seventh input information computing unit **4104***fb*, and the second computing unit **4108** (step S**4103***e*). The seventh random number generator **4104***e* generates a uniform random number r_7 that is a natural number greater than or equal to 0 and less than K_G. The generated random number r_7 is sent to the sixth input information computing unit **4104***fa* and the second computing unit **4108** (step S**4103***f*).

The sixth input information computing unit **4104***fa* uses a natural number a selected by the natural number electing unit **1102**, c_2 included in the ciphertext x, and the random number r_6 to compute sixth input information $c_2{}^{a}W^{r6}$ (step S**4103***g*). The seventh input information computing unit **4104***fb* uses the natural number a selected by the natural number selecting unit **1102**, c_1 included in the ciphertext x, and the random number r_7 to compute seventh input information $c_1{}^{a}V^{r6}\mu_g{}^{r7}$ (step S**4103***h*).

The first input information computing unit **4104***c* outputs the fourth input information $c_2{}^{b}W^{r4}$ and the fifth input information $c_1{}^{b}V^{r4}\mu_g{}^{r5}$ generated as described above as first input information $\tau_1=(c_2{}^{b}W^{r4}, c_1{}^{b}V^{r4}\mu_g{}^{r5})$. The second input information computing unit **4104***f* outputs the sixth input information $c_2{}^{a}W^{r6}$ and the seventh input information $c_1{}^{a}V^{r6}\mu_g{}^{r7}$ generated as described above as second input information $\tau_2=(c_2{}^{a}W^{r6}, c_1{}^{a}V^{r6}\mu_g{}^{r7})$ (step S**4103***i*).

<<Processes at Steps S**4200** Through S**4203**>>

As illustrated in FIG. **7**, first, the first input information $\tau_1=(c_2{}^{b}W^{r4}, c_1{}^{b}V^{r4}\mu_g{}^{r5})$ is input in the first output information computing unit **4201** of the capability providing apparatus **42** (FIG. **3**) and the second input information $\tau_2=(c_2{}^{a}W^{r6}, c_1{}^{a}V^{r6}\mu_g{}^{r7})$ is input in the second output information computing unit **4202** (step S**4200**).

The first output information computing unit **4201** uses the first input information $\tau_1=(c_2{}^{b}W^{r4}, c_1{}^{b}V^{r4}\mu_g{}^{r5})$ and the decryption key s stored in the key storage **1204** to correctly compute $f(c_1{}^{b}V^{r4}\mu_g{}^{r5}, c_2{}^{b}W^{r4})$ with a probability greater than a certain probability and sets the result of the computation as first output information z_1. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **4201** may or may not be $f(c_1{}^{b}V^{r4}\mu_g{}^{r5}, c_2{}^{b}W^{r4})$ (step S**4201**).

The second output information computing unit **4202** can correctly compute $f(c_1{}^{a}V^{r6}\mu_g{}^{r7}, c_2{}^{a}W^{r6})$ with a probability greater than a certain probability by using the second input information $\tau_2=(c_2{}^{a}W^{r6}, c_1{}^{a}V^{r6}\mu_g{}^{r7})$ and the decryption key s stored in the key storage **1204** and sets the result of the computation as second output information z_2. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **4202** may or may not be $f(c_1{}^{a}V^{r6}\mu_g{}^{r7}, c_2{}^{a}W^{r6})$ (step S**4202**). The first output information computing unit **4201** outputs the first output information z_1 and the second output information computing unit **4202** outputs the second output information z_2 (step S**4203**).

<<Processes at Steps S**4104** and S**4105**>>

Returning to FIG. **6**, the first output information z_1 is input in the first computing unit **4105** of the computing apparatus **41** (FIG. **2**) and the second output information z_2 is input in the second computing unit **4108** (step S**4104**).

The first computing unit **4105** uses the input first output information z_1 and random numbers r_4 and r_5 to compute $z_1Y^{-r4}\mu_g{}^{-r5}$ and sets the result of the computation as u (step S**4105**). The result u of the computation is sent to the first power computing unit **1106**. Here, $u=z_1Y^{-r4}\mu_g{}^{-r5}=f(c_1, c_2)^{b}x_1$. That is, $z_1Y^{-r4}\mu_g{}^{-r5}$ is an output of a randomizable sampler having an error X_1 for $f(c_1, c_2)$. The reason will be described later.

<<Process at Step S**4108**>>

The second computing unit **4108** uses the input second output information z_2 and random numbers r_6 and r_7 to compute $z_2Y^{-r6}\mu_g{}^{-r7}$ and sets the result of the computation as v. The result v of the computation is sent to the second power computing unit **1109**. Here, $v=z_2Y^{-r6}\mu_g{}^{-r7}=f(c^1, c_2)^{a}x_2$. That is, $Z_2Y^{-r6}\mu^{-r7}$ is an output of a randomizable sampler having an error X_2 for $f(c_1, c_2)$. The reason will be described later.

<<Reason Why $z_1Y^{-r4}\mu_g{}^{-r5}$ and $z_2Y^{-r6}\mu_g{}^{-r7}$ are Outputs of Randomizable Samplers Having Errors X_1 and X_2, Respectively, for $f(c_1, c_2)$>>

Let c be a natural number, R_1, R_2, $R_{1'}$ and $R_{2'}$ be random numbers, and $B(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})$ be the result of computation performed by the capability providing apparatus **42** using $c_1{}^{c}V^{R1}\mu_g{}^{R2}$ and $c_2{}^{c}W^{R1}$. That is, the first output information computing unit **4201** and the second output information computing unit **4202** return $z=B(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})$ as the results of computations to the computing apparatus **41**. Furthermore, a random variable X having a value in a group G is defined as $X=B(V^{R1'}\mu_g{}^{R2'}, W^{R1'})f(V^{R1'}\mu_g{}^{R2'}, W^{R1'})^{-1}$.

Here, $zY^{-R1}\mu_g{}^{-R2}=B(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})Y^{-R1}\mu_g{}^{-R2}=Xf(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})Y^{-R1}\mu_g{}^{-R2}=Xf(c_1, c_2)^{c}f(V, W)^{R1}f(\mu_g, e_g)^{R2}Y^{-R1}\mu_g{}^{-R2}=Xf(c_1, c_2)^{c}Y^{R1}\mu_g{}^{R2}Y^{-R1}\mu_g{}^{-R2}=f(c_1c_2)^{c}X$. That is, $zY^{-R1}\mu_g{}^{-R2}$ is an output of a randomizable sampler having an error X for f(x). Note that e_g is an identity element of the group G.

The expansion of formula given above uses the properties such that $X=B(V^{R1'}\mu_g{}^{R2'}, W^{R1'})f(V^{R1'}\mu_g{}^{R2'}, W^{R1'})^{-1}=B(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})f(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})$ and that $B(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})=Xf(c_1{}^{c}V^{R1}\mu_g{}^{R2}, c_2{}^{c}W^{R1})$. The properties are based on the fact that R_1, R_2, $R_{1'}$ and $R_{2'}$ are random numbers.

Therefore, considering that a and b are natural numbers and r_4, r_5, r_6 and r_7 are random numbers, $z_1Y^{-r4}\mu_g{}^{-r5}$ and $z_2Y^{-r6}\mu_g{}^{-r7}$ are, likewise, outputs of randomizable samplers having errors X_1 and X_2, respectively, for $f(c_1, c_2)$.

Fifth Embodiment

In the embodiments described above, a plurality of pairs (a, b) of natural numbers a and b that are relatively prime to each other are stored in the natural number storage **1101** of the computing apparatus and the pairs (a, b) are used to perform the processes. However, one of a and b may be a constant. For example, a may be fixed at 1 or b may be fixed at 1. In other words, one of the first randomizable sampler and the second randomizable sampler may be replaced with a sampler. If one of a and b is a constant, the process for selecting the constant a or b is unnecessary, the constant a or b is not input in the processing units and each processing units can treat it as a constant in computations. If a or b set as a constant is equal to 1, $f(x)=u^{b'}V^{a'}$ can be obtained as f(x)=v or f(x)=u without using a' or b'.

A fifth embodiment is an example of such a variation, in which b is fixed at 1 and the second randomizable sampler is replaced with a sampler. The following description will focus on differences from the first embodiment. Specific

examples of the first randomizable sampler and the sampler are similar to those described in the second to fourth embodiments and therefore description of the first randomizable sampler and the sampler will be omitted.

<Configuration>

As illustrated in FIG. **1**, a proxy computing system **5** of the fifth embodiment includes a computing apparatus **51** in place of the computing apparatus **11** of the first embodiment and a capability providing apparatus **52** in place of the capability providing apparatus **12**.

As illustrated in FIG. **10**, the computing apparatus **51** of the fifth embodiment includes, for example, a natural number storage **5101**, a natural number selecting unit **5102**, an input information providing unit **5104**, a first computing unit **5105**, a first power computing unit **1106**, a first list storage **1107**, a second computing unit **5108**, a second list storage **5110**, a determining unit **5111**, a final output unit **1112**, and a controller **1113**.

As illustrated in FIG. **3**, the capability providing apparatus **52** of the fifth embodiment includes, for example, a first output information computing unit **5201**, a second output information computing unit **5202**, a key storage **1204**, and a controller **1205**.

<Processes>

Processes of this embodiment will be described below. For the processes, let G and H be groups (for example commutative groups), f(x) be a decryption function for decrypting a ciphertext x, which is an element of the group H, with a particular decryption key s to obtain an element of the group G, generators of the groups G and H be μ_g and μ_h, respectively, X_1 and X_2 be random variables having values in the group G, x_1 be a realization of the random variable X_1, and x_2 be a realization of the random variable X_2. It is assumed here that a plurality of natural numbers a are stored in the natural number storage **5101** of the computing apparatus **51**.

As illustrated in FIG. **11**, first, the natural number selecting unit **S102** of the computing apparatus **51** (FIG. **10**) randomly reads one natural number a from among the plurality of natural numbers a stored in the natural number storage **5101**. Information on the read natural number a is sent to the input information providing unit **5104** and the first power computing unit **1106** (step S**5100**).

The controller **1113** sets t=1 (step S**1102**).

The input information providing unit **5104** generates and outputs first input information τ_1 and second input information τ_2 which are elements of the group H and each of which corresponds to an input ciphertext x. Preferably, the first input information τ_1 and the second input information τ_2 are information whose relation with the ciphertext x is scrambled. This enables the computing apparatus **51** to conceal the ciphertext x from the capability providing apparatus **52**. Preferably, the second input information τ_2 of this embodiment further corresponds to the natural number a selected by the natural number selecting unit **5102**. This enables the computing apparatus **51** to evaluate the decryption capability provided by the capability providing apparatus **52** with a high degree of accuracy (step S**5103**). A specific example of the pair of the first input information τ_1 and the second input information τ_2 is a pair of first input information τ_1 and the second input information τ_2 of any of the second to fourth embodiments when b=1.

As illustrated in FIG. **7**, the first input information τ_1 is input in the first output information computing unit **5201** of the capability providing apparatus **52** (FIG. **3**) and the second input information τ_2 is input in the second output information computing unit **5202** (step S**5200**).

The first output information computing unit **5201** uses the first input information τ_1 and the decryption key s stored in the key storage **1204** to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets the result of the computation as first output information z_1 (step S**5201**). The second output information computing unit **5202** uses the second input information τ_2 and the decryption key s stored in the key storage **1204** to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets the result of the computation as second output information z_2 (step S**5202**). That is, the first output information computing unit **5201** and the second output information computing unit **5202** output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **5201** may or may not be $f(\tau_1)$ and the result of the computation by the second output information computing unit **5202** may or may not be $f(\tau_2)$. A specific example of the pair of the first output information z_1 and the second output information z_2 is a pair of first output information z_1 and the second output information z_2 of any of the second to fourth embodiments when b=1.

The first output information computing unit **5201** outputs the first output information z_1 and the second output information computing unit **5202** outputs the second output information z_2 (step S**5203**).

Returning to FIG. **11**, the first output information z_1 is input in the first computing unit **5105** of the computing apparatus **51** (FIG. **10**) and the second output information z_2 is input in the second computing unit **5108**. The first output information z_1 and the second output information z_2 are equivalent to the decryption capability provided by the capability providing apparatus **52** to the computing apparatus **51** (step S**5104**).

The first computing unit **5105** generates a computation result $u=f(x)x_1$ from the first output information z_1. A specific example of the computation result u is a result u of computation of any of the second to fourth embodiments when b=1. The result u of the computation is sent to the first power computing unit **1106** (step S**5105**).

The first power computing unit **1106** computes $u'=u^a$. The pair of the result u of the computation and u' computed on the basis of the result of the computation, (u, u'), is stored in the first list storage **1107** (step S**1106**).

The second computing unit **5108** generates a computation result $v=f(x)^a x_2$ from the second output information z_2. A specific example of the result v of the computation is a result v of the computation of any of the second to fourth embodiments. The result v of the computation is stored in the second list storage **5110** (step S**5108**).

The determining unit **5111** determines whether or not there is one that satisfies u'=v among the pairs (u, u') stored in the first list storage **1107** and v stored in the second list storage **5110** (step S**5110**). If there is one that satisfies u'=v, the process proceeds to step S**5114**; if there is not one that satisfies u'=v, the process proceeds to step S**1111**.

At step S**1111**, the controller **1113** determines whether or not $t=T_{max}$(step S**1111**). Here, T_{max} is a predetermined natural number. If $t=T_{max}$, the controller **1113** outputs information indicating that the computation is impossible, for example the symbol "⊥" (step S**1113**), then the process ends. If not $t=T_{max}$, the controller **1113** increments t by 1, that is, sets t=t+1 (step S**1112**), then the process returns to step S**5103**.

At step S**5114**, the final output unit **1112** outputs u corresponding to u' that has been determined to satisfy u'=v (step S**5114**). The obtained u is equivalent to $u^{b'}v^{a'}$ when b=1

in the first to fourth embodiments. That is, u thus obtained can be a result f(x) of decryption of the ciphertext x with a particular decryption key s with a high probability. Therefore, the process described above is repeated multiple times and the value that has most frequently obtained among the values obtained at step **S5114** can be chosen as the decryption result. As will be described later, u=f(x) can result with an overwhelming probability, depending on settings. In that case, the value obtained at step **S5114** can be directly provided as a result of decryption.

<<Reason why Decryption Result f(x) can be Obtained>>

The reason why a decryption result f(x) can be obtained on the computing apparatus **51** of this embodiment will be described below. Terms required for the description will be defined first.

Black-box:

A black-box $F(\tau)$ of $f(\tau)$ is a processing unit that takes an input of $\tau\epsilon H$ and outputs $z\epsilon G$. In this embodiment, each of the first output information computing unit **5201** and the second output information computing unit **5202** is equivalent to the black box $F(\tau)$ for the decryption function $f(\tau)$. A black-box $F(\tau)$ that satisfies $z=f(\tau)$ for an element $\tau\epsilon_U H$ arbitrarily selected from a group H and $z=F(\tau)$ with a probability greater than $\delta(0<\delta\leq 1)$, that is, a black-box $F(\tau)$ for $f(\tau)$ that satisfies

$$Pr[z=f(\tau)|\tau\epsilon_u H, z=F(\tau)]>\delta \quad (1)$$

is called a δ-reliable black-box $F(\tau)$ for $f(\tau)$. Here, δ is a positive value and is equivalent to the "certain probability" stated above.

Self-corrector:

A self-corrector $C^F(x)$ is a processing unit that takes an input of $x\epsilon H$, performs computation by using a black-box $F(\tau)$ for $f(\tau)$, and outputs $j\epsilon G\cup\perp$. In this embodiment, the computing apparatus **51** is equivalent to the self-corrector $C^F(x)$.

Almost self-corrector:

Assume that a self-corrector $C^F(x)$ that takes an input of $x\epsilon H$ and uses a δ-reliable black-box $F(\tau)$ for $f(\iota)$ to perform computation outputs a correct value $j=f(x)$ with a probability sufficiently greater than the provability with which the self-corrector $C^F(x)$ outputs an incorrect value $j\neq f(x)$.

That is, assume that a self-corrector $C^F(x)$ satisfies

$$Pr[j=f(x)|j=C^F(x), j\neq\perp]>Pr[j\neq f(x)|j=C^F(x), j\neq\perp]+\Delta \quad (2)$$

Here, Δ is a certain positive value $(0<\Delta<1)$. If this is the case, the self-corrector $C^F(x)$ is called an almost self-corrector. For example, for a certain positive value $\Delta'(0<\Delta'<1)$, if a self-corrector $C^F(x)$ satisfies

$$Pr[j=f(x)|j=C^F(x)]>(1/3)+\Delta'$$

$$Pr[j=\perp|j=C^F(x)]<1/3$$

$$Pr[j\neq f(x) \text{ and } j\neq\perp|j=C^F(x)]<1/3,$$

then the self-corrector $C^F(x)$ is an almost self-corrector. Examples of Δ' include $\Delta'=1/12$ and $\Delta'=1/3$.

Robust self-corrector:

Assume that a self-corrector $C^F(x)$ that takes an input of $x\epsilon H$ and uses a δ-reliable black-box $F(\tau)$ for $f(\tau)$ outputs a correct value $j=f(x)$ or $j=\perp$ with an overwhelming probability. That is, assume that for a negligible error $\xi(0\leq\xi<1)$, a self-corrector $C^F(x)$ satisfies

$$Pr[j=f(x) \text{ or } j=\perp|j=C^F(x)]>1-\xi \quad (3)$$

If this is the case, the self-corrector $C^F(x)$ is called a robust self-corrector. An example of the negligible error ξ is a function vale ξ(k) of a security parameter k. An example of the function value ξ(k) is a function value ξ(k) such that $\{\xi(k)\ p(k)\}$ converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function value ξ(k) include $\xi(k)=2^{-k}$ and $\xi(k)=2^{-\sqrt{k}}$.

A robust self-corrector can be constructed from an almost self-corrector. Specifically, a robust self-corrector can be constructed by executing an almost self-constructor multiple times for the same x and selecting the most frequently output value, except ⊥, as j. For example, an almost self-corrector is executed $O(\log(1/\xi))$ times for the same x and the value most frequently output is selected as j to construct robust self-corrector. Here, $O(\cdot)$ represents O notation.

Pseudo-free action:

An upper bound of the probability

$$Pr[\alpha^a=\beta \text{ and } \alpha\neq e_g|a\epsilon_u\Omega, \alpha\epsilon X_1, \beta\epsilon X_2] \quad (4)$$

of satisfying $\alpha^a=\beta$ for all possible X_1 and X_2 is called a pseudo-free indicator of a pair (G, Ω) and is represented as P(G, Ω)), where G is a group, Ω is a set of natural numbers $\Omega=\{0, \ldots, M\}$ (M is a natural number greater than or equal to 1), α and β are realizations $\alpha\epsilon X_1$ $(\alpha\neq e_g)$ and $\beta\epsilon X_2$ of random variables X_1 and X_2 that have values in the group G, and $a\epsilon Q$. If a certain negligible function ζ(k) exists and

$$P(G,\Omega)<\zeta(k) \quad (5),$$

then a computation defined by the pair (G; Ω) is called a pseudo-free action. Note that a computation defined by a group is expressed multiplicatively in the fifth embodiment. That is, "α^a" for $\alpha\epsilon G$ means that a computation defined at the group G is applied a times to α. An example of the negligible function ζ(k) is such that $\{\zeta(k)p(k)\}$ converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function ζ(k) include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. For example, if the probability of Formula (4) is less than $O(2^{-k})$ for a security parameter k, a computation defined by the pair (C, Ω) is a pseudo-free action. For example, if the number of the elements $|\Omega\cdot\alpha|$ of a set $\Omega\cdot a=\{a(\alpha)|a\epsilon\Omega\}$ exceeds 2^k for any $\alpha\epsilon G$ where $\alpha\neq e_g$, a computation defined by the pair (G, Ω) can be a pseudo-free action. There are many such examples. For example, if the group G is a residue group Z/pZ modulo prime p, the prime p is the order of 2^k, the set $\Omega=\{0, \ldots, p-2\}$, $a(\alpha)$ is $\alpha^a\epsilon Z/pZ$, and $\alpha\neq e_g$, then $\Omega\cdot\alpha=\{\alpha^a|a=0, \ldots, p-2\}=\{e_g, \alpha^1, \ldots, \alpha^{p-2}\}$ and $|\Omega\cdot\alpha|=p-1$. If a certain constant C exists and k is sufficiently large, $|\Omega\cdot\alpha|>C2^k$ is satisfied because the prime p is the order of 2^k. Here, the probability of Formula (4) is less than $C^{-1}2^{-k}$ and a computation defined by such pair (G, Ω) is a pseudo-free action.

δ^γ-reliable randomizable sampler:

A randomizable sampler that whenever a natural number a is given, uses the δ-reliable black-box $F(\tau)$ for $f(\tau)$ and returns $w^a x'$ corresponding to a sample x' that depends on a random number X for $w\epsilon G$ and in which the probability that $w^a x'=w^a$ is greater than δ^γ (γ is a positive constant), that is,

$$Pr[w^a x'=w^a]>\delta^\gamma \quad (6)$$

is satisfied, is called a δ^γ-reliable randomizable sampler. The combination of the input information providing unit **5104**, the second output information computing unit **5202**, and the second computing unit **5108** of this embodiment is a δ^γ-reliable randomizable sampler for w=f(x).

The definitions given above will be used to describe the reason why a decryption result f(x) can be obtained by using the computing apparatus **51** of this embodiment.

At step **S5110** of this embodiment, determination is made as to whether u'=v, that is, whether $u^a=v$. Since the combination of the input information providing unit **5104**, the

second output information computing unit **5202**, and the second computing unit **5108** of this embodiment is a δ^{γ}-reliable randomizable sampler (Formula (6)), $u^{a}=v$ holds (Yes at step **S5110**) with an asymptotically large probability if T_{max} is greater than a certain value determined by k, δ and γ. For example, Markov's inequality shows that if $T_{max}\geq 4/\delta^{\gamma}$, the probability that $u^{a}=v$ holds (Yes at step **S5110**) is greater than ½.

Since $u=f(x)x_1$ and $v=f(x)^{a}x_2$ in this embodiment, ${x_1}^{a}=x_2$ holds if $u^{a}=v$ holds. ${x_1}^{a}=x_2$ holds if $x_1=x_2=e_g$ or $x_1\neq e_g$. If $x_1=x_2=e_g$, then $u=f(x)$ and therefore u output at step **S5114** is a correct decryption result f(x). On the other hand, if $x_1\neq e_g$, then $u\neq f(x)$ and therefore u output at step **S5114** is not a correct decryption result f(x).

If a computation defined by a pair (G, Ω) of a group G and a set Ω to which a natural number a belongs is a pseudo-free action or $T_{max}{}^{2}P(G, \Omega)$ is asymptotically small for a pseudo-free index P(G, Ω), the probability that $x_1\neq e_g$ (Formula (4)) when $u^{a}=v$ is asymptotically small. Accordingly, the probability that $x_1=e_g$ when $u^{a}=v$ is asymptotically large. Therefore, if a computation defined by a pair (G, Ω) is a pseudo-free action or $T_{max}{}^{2}P(G, \Omega)$ is asymptotically small, the probability that an incorrect decryption result f(x) is output when $u^{a}=v$ is sufficiently smaller than the probability that a correct decryption result f(x) is output when $u^{a}=v$. In this case, it can be said that the computing apparatus **51** is an almost self-corrector (see Formula (2)). Therefore, a robust self-corrector can be constructed from the computing apparatus **51** as described above and a correct decryption result f(x) can be obtained with an overwhelming probability. If a computation defined by (G, Ω) is a pseudo-free action, the probability that an incorrect decryption result f(x) is output when $u^{a}=v$ is also negligible. In that case, the computing apparatus **51** outputs a correct decryption result f(x) or ⊥ with an overwhelming probability.

Note that "η(k') is asymptotically small" means that k_0 is determined for an arbitrary constant ρ and the function value η(k') for any k' that satisfies $k_0<k'$ for k_0 is less than ρ. An example of k' is a security parameter k.

"η(k') is asymptotically large" means that k_0 is determined for an arbitrary constant ρ and the function value 1−η(k') for any k' that satisfies $k_0<k'$ for k_0 is less than ρ.

<<δ^{γ}-Reliable Randomizable Sampler and Security>>

Consider the following attack.

- A black-box F(τ) or a part of the black-box F(τ) intentionally outputs an invalid z or a value output from the black-box F(τ) is changed to an invalid z.
- $w^{a}x'$ corresponding to the invalid z is output from the randomizable sampler.
- $w^{a}x'$ corresponding to the invalid z increases the probability with which the self-corrector $C^{F}(x)$ outputs an incorrect value even though $u^{a}=v$ holds (Yes at step **S5110**) in the self-corrector $C^{F}(x)$.

This attack is possible if the probability distribution $D_a=w^{a}x'w^{-a}$ of an error of $w^{a}x'$ output from the randomizable sampler for a given natural number a depends on the natural number a. For example, if tampering is made so that v output from the second computing unit **5108** is $f(x)^{a}{x_1}^{a}$, then $u^{a}=v$ always holds regardless of the value of x_1. Therefore, it is desirable that the probability distribution $D_a=w^{a}x'w^{-a}$ of an error of $w^{a}x'$ output from the randomizable sampler for a given natural number a do not depend on the natural number a.

Alternatively, it is desirable that the randomizable sampler be such that a probability distribution D that has a value in a group G that cannot be distinguished from the probability distribution $D_a=w^{a}x'w^{-a}$ of an error of $w^{a}x'$ for any element $a\in^{\forall}\Omega$ of a set Ω exists (the probability distribution D_a and the probability distribution D are statistically close to each other). Note that the probability distribution D does not depend on a natural number a. That the probability distribution D_a and the probability distribution D cannot be distinguished from each other means that the probability distribution D_a and the probability distribution D cannot be distinguished from each other by a polynomial time algorithm. For example, if

$$\Sigma_{g\in G}|Pr[g\in D]-Pr[g\in D_a]|<\zeta \tag{7}$$

is satisfied for negligible ζ(0≤ζ1), the probability distribution D_a and the probability distribution D cannot be distinguished from each other by the polynomial time algorithm. An example of negligible ζ is a function value ζ(k) of the security parameter k. An example of the function value ζ(k) is a function value such that {ζ(k)p(k)} converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function ζ(k) include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. These facts also apply to the first to fourth embodiments which use natural numbers a and b.

Sixth Embodiment

A sixth embodiment is a mode in which the present invention is applied to decryption of the GHV encryption scheme (see Reference literature 1 "C, Genrty, S. Halevi and V. Vaikuntanathan, 'A Simple BGNType Cryptosystem from LWE,' Advances in Cryptology—EUROCRYPT 2010, LNCS 6110, pp. 506-522, Springer-Verlag, 2010", for example), which is a type of lattice-based cryptography. The following description will focus on differences from the embodiments described above.

<Configuration>

As illustrated in FIG. **1**, a proxy computing system **6** of the sixth embodiment includes a computing apparatus **61** in place of the computing apparatus **11** of the first embodiment and a capability providing apparatus **62** in place of the capability providing apparatus **12**.

As illustrated in FIG. **12**, the computing apparatus **61** of the sixth embodiment includes, for example, a matrix storage **6101**, a matrix selecting unit **6102**, an input information providing unit **6104**, a first computing unit **6105**, a matrix product computing unit **6106**, a first list storage **6107**, a second computing unit **6108**, a second list storage **6110**, a determining unit **6111**, a final output unit **6112**, and a controller **1113**.

As illustrated in FIG. **14**, the input information providing unit **6104** of this embodiment includes, for example, a first random matrix selecting unit **6104***a*, a second random matrix selecting unit **6104***b*, a first encryption unit **6104***c*, a second encryption unit **6104***d*, a first input information computing unit **6104***e*, a third random matrix selecting unit **6104***f*, a fourth random matrix selecting unit **6104***g*, a third encryption unit **6104***h*, a fourth encryption unit **6104***i*, and a second input information computing unit **6104***j*.

As illustrated in FIG. **13**, the capability providing apparatus **62** of the sixth embodiment includes, for example, a first output information computing unit **6201**, a second output information computing unit **6202**, a key storage **6204**, and a controller **1205**.

<Processes>

Processes of this embodiment will be described below. In this embodiment, let G_M be a set of ι×ι matrices, H_M be a set of ι×ι matrices, ${}_{M}X_1$ and ${}_{M}X_2$ be random variables having values in the set G_M, ${}_{M}x_1$ be a realization of the random variable ${}_{M}X_1$, ${}_{M}x_2$ be a realization of the random

variable ${}_{M}X_2$, and a_M be an element of the set H_M. In this embodiment, PK be a $\iota\times K$ matrix which is an encryption key (public key), SK be a decryption key (secret key) which is a $\iota\times\iota$ matrix that satisfies $PK\cdot SK=0$, CM be a $\kappa\times\iota$ matrix, NM be a $\iota\times\iota$ matrix, UM be a $\iota\times\iota$ unit matrix, PT be a plaintext $PT\epsilon G_M$ which is an element of the set G_M, x_M be a ciphertext $x_M\epsilon H_M$ which is an element of the set H_M, ENC_M be an encryption function for encrypting the plaintext PT which is an element of the set G_M to obtain the ciphertext $x_M\epsilon H_M$, and $f_M(x_M)$ be a decryption function for decrypting the ciphertext $x_M\epsilon H_M$ with a particular decryption key SK to obtain the plaintext PT which is an element of the set G_M. The decryption function $f_M(x_M)$ is a homomorphic function. For example, let G_M be a set of $\iota\times\iota$ matrices $(Z/2Z)^{\iota\times\iota}$, H_M be a set of $\iota\times\iota$ matrices $(Z/qZ)^{\iota\times\iota}$, the encryption key PK be a $\iota\times\kappa$ matrix $(Z/qZ)^{\iota\times\kappa}$, the decryption key SK be a $\iota\times\iota$ matrix $(Z/qZ)^{\iota\times\iota}$, CM be a randomly selected $\kappa\times\iota$ matrix $(Z/qZ)^{\kappa\times\iota}$, NM be a $\iota\times\iota$ matrix $(Z/qZ)^{\iota\times\iota}$ according to a Gaussian distribution, UM be a $\iota\times\iota$ unit matrix $(Z/2Z)^{\iota\times\iota}$, the encryption function $ENC_M(PT)$ be $PK\cdot CM+2\cdot NM+PT(\text{mod } q)$, and the decryption function $f_M(x_M)$ be $SK^{-1}\{SK\cdot x_M\cdot SK^T(\text{mod } q)\}(SK^T)^{-1}(\text{mod } 2)$. Here, κ, ι and q are positive integers, $\cdot^T$ is the transposed matrix of $\cdot$, and $(Z/qZ)^{\kappa\times\iota}$ is a matrix of κ rows and ι columns having members of a factor ring Z/qZ modulo q as elements. In the sixth embodiment, the product between matrices α_1 and α_2 is written as $\alpha_1\cdot\alpha_2$ and the sum of matrices α_1 and α_2 is written as $\alpha_1+\alpha_2$. A matrix that is equal to the each element of a matrix α by a natural number β is written as $\beta\cdot\alpha$.

It is assumed in this embodiment that a plurality of matrices $a_M\epsilon H_M$ are stored in the matrix storage **6101** of the computing apparatus **61** (FIG. **12**) and a decryption key SK is stored in the key storage **6204** of the capability providing apparatus **62** (FIG. **13**) in a secure manner. As illustrated in FIG. **15**, first, the matrix selecting unit **6102** of the computing apparatus **61** (FIG. **12**) uniformly randomly selects and reads one matrix a_M from among the plurality of matrices stored in the matrix storage **6101**. Information of the read matrix a_M is sent to the input information providing unit **6104** and the matrix product computing unit **6106** (step S**6100**).

The controller **1113** sets t=1 (step S**1102**).

The input information providing unit **6104** generates and outputs first input information ${}_M\tau_1$ and second input information ${}_M\tau_2$ which are elements of the set H_M and each of which corresponds to an input ciphertext x_M. Preferably, the first input information ${}_M\tau_1$ and the second input information ${}_M\tau_2$ are information whose relation with the ciphertext x_M is scrambled. This enables the computing apparatus **61** to conceal the ciphertext x_M from the capability providing apparatus **62**. The second input information ${}_M\tau_2$ further corresponds to an element a_M. This enables the computing apparatus **61** to evaluate the decryption capability provided by the capability providing apparatus **62** with a high degree of accuracy (step S**6103**). A specific example of step S**6103** will be described below with reference to FIG. **16**.

[Specific Example of Step S**6103**]

The first random matrix selecting unit **6104***a* of the input information providing unit **6104** (FIG. **14**) uniformly randomly selects an element M_{R1} of the set G_M (step **6103***a*). The selected M_{R1} is sent to the first encryption unit **6104***c* and the first computing unit **6105** (step S**6103***a*). The second random matrix selecting unit **6104***b* selects uniform and random matrices CM_{11} and $CM_{12}\epsilon(Z/qZ)^{\kappa\times\iota}$ of $\kappa\times\iota$. The selected CM_{11} and CM_{12} are sent to the first input information computing unit **6104***e* (step S**6103***b*). The first encryption unit **6104***c* uses the public key PK to generate a first ciphertext $C_{R1}=PK\cdot CM+2\cdot NM+M_{R1}$ (mod q) which is a ciphertext $ENC_M(M_{R1})$ of M_{R1}. The first ciphertext C_{R1} is sent to the first input information computing unit **6104***e* (step S**6103***c*). The second encryption unit **6104***d* uses the public key PK to generate a second ciphertext $C_{UM}=PK\cdot CM+2\cdot NM+UM$ (mod q) which is a ciphertext $ENC_M(UM)$ of the unit matrix UM. The second ciphertext C_{UM} is sent to the first input information computing unit **6104***e* (step S**6103***d*). The first input information computing unit **6104***e* further takes an input of the ciphertext x_M. The first input information computing unit **6104***e* obtains and outputs $(x_M\cdot C_{UM}+C_{R1})+PK\cdot CM_{11}+2\cdot NM+CM_{12}{}^T\cdot PK^T$ as first input information ${}_M\tau_1$. Note that the order of the products of the matrices is not particularly specified. That is, the first input information computing unit **6104***e* may compute $Re(C_x)=C_x+PK\cdot CM_{11}+2\cdot NM+CM_{12}{}^T\cdot PK^T$ where $C_x=x_M\cdot C_{UM}+C_{R1}$, to generate the first input information ${}_M\tau_1$ or may compute $Re(C_x)$ where $C_x=C_{UM}\cdot x_M+C_{R1}$ to generate the first input information ${}_M\tau_1$ (step S**6103***e*).

The third random matrix selecting unit **6104***f* uniformly randomly selects an element M_{R2} of the set G_M. The selected M_{R2} is sent to the third encryption unit **6104***h* and the second computing unit **6108** (step S**6103***f*). The fourth random matrix selecting unit **6104***g* selects random matrices CM_{21} and $CM_{22}\epsilon(Z/qZ)^{\kappa\times\iota}$ of $\kappa\times\iota$. The selected CM_{21} and CM_{22} are sent to the second input information computing unit **6104***j* (step S**6103***g*). The third encryption unit **6104***h* uses the public key PK to generate a third ciphertext $C_{R2}=PK\cdot CM+2\cdot NM+M_{R2}(\text{mod } q)$ which is a ciphertext $ENC_M(M_{R2})$ of M_{R2}. The third ciphertext C_{R2} is sent to the second input information computing unit **6104***j* (step S**6103***h*). Matrix a_M is input in the fourth encryption unit **6104***i*. The fourth encryption unit **6104***i* uses the public key PK to generate a fourth ciphertext $C_a=PK\cdot CM+2\cdot NM+a_M$ (mod q) which is a ciphertext $ENC_M(a_M)$ of the matrix a_M. The fourth ciphertext C_a is sent to the second input information computing unit **6104***j* (step S**6103***i*). The second input information computing unit **6104***j* further takes an input of the ciphertext x_M. The second input information computing unit **6104***j* obtains and outputs $(x_M\cdot C_a+C_{R2})+PK\cdot CM_{21}+2\cdot NM+CM_{22}{}^T\cdot PK^T$ as second input information ${}_M\tau_2$. The second input information computing unit **6104***j* may compute $Re(C_x)$ where $C_x=x_M\cdot C_a+C_{R2}$ to generate the second input information ${}_M\tau_2$ or may compute $Re(C_x)$ where $C_x=C_a\cdot x_M+C_{R2}$ to generate the second input information ${}_M\tau_2$ ((step S**6103***j*)/end of description of [Specific example of step S**6103**]).

As illustrated in FIG. **17**, the first input information ${}_M\tau_1$ is input in the first output information computing unit **6201** of the capability providing apparatus **62** (FIG. **13**) and the second input information ${}_M\tau_2$ is input in the second output information computing unit **6202** (step S**6200**).

The first output information computing unit **6201** uses the first input information ${}_M\tau_1$ and the decryption key SK stored in the key storage **6204** to correctly compute $f_M({}_M\tau_1)=SK^{-1}\{SK\cdot{}_M\tau_1\cdot SK^T(\text{mod } q)\}(SK^T)^{-1}(\text{mod } 2)$ with a probability greater than a certain probability and sets the obtained result of the computation as first output information ${}_Mz_1$ (step S**6201**). The second output information computing unit **6202** uses the second input information ${}_M\tau_2$ and the decryption key SK stored in the key storage **6204** to correctly compute $f_M({}_M\tau_2)=SK^{-1}\{SK\cdot{}_M\tau_2\cdot SK^T(\text{mod } q)\}(SK^T)^{-1}(\text{mod } 2)$ with a probability greater than a certain probability and sets the obtained result of the computation as second output information ${}_Mz_2$ (step S**6202**). That is, the first output information computing unit **6201** and the second output information computing unit **6202** outputs computation

results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **6201** may or may not be $f_M({}_M\tau_1)$ and the result of the computation by the second output information computing unit **6202** may or may not be $f_M({}_M\tau_2)$.

The first output information computing unit **6201** outputs the first output information ${}_Mz_1$ and the second output information computing unit **6202** outputs the second output information ${}_Mz_2$ (step S**6203**).

Returning to FIG. **15**, the first output information ${}_Mz_1$ is input in the first computing unit **6105** of the computing apparatus **61** (FIG. **12**) and the second output information ${}_Mz_2$ is input in the second computing unit **6108**. The first output information ${}_Mz_1$ and the second output information ${}_Mz_2$ are equivalent to the decryption capability provided by the capability providing apparatus **62** to the computing apparatus **61** (step S**6104**).

The first computing unit **5105** uses the first output information ${}_Mz_1$ to compute ${}_Mz_1-M_{R1}$ and sets the result of the computation as u_M. The result u_M of the computation is sent to the matrix product computing unit **6106**. Here, $u_M={}_MZ^1-M_{R1}=f_M(x_M)+{}_MX_1$. That is, u_M serves as a sampler having an error ${}_MX_1$ for $f_M(x_M)$. The reason will be described later (step S**6105**).

The matrix product computing unit **6106** obtains $u_{M'}=u_M\cdot a_M$. Note that the matrix product computing unit **6106** may compute $u_M\cdot a_M$ to obtain $u_{M'}$ or may compute $a_M\cdot u_M$ to obtain $u_{M''}$. The pair $(u_M, u_{M'})$ of the result u_M of the computation and $u_{M'}$ computed on the basis of the result of the computation is stored in the first list storage **6107** (step S**6106**).

The second computing unit **6108** uses the second output information ${}_Mz_2$ to compute ${}_Mz_2-M_{R2}$ and sets the result of the computation as v_M. The result v_M of the computation is stored in the second list storage **6110**. Here, $v_M={}_Mz_2-M_{R2}=f_M(x_M)\cdot a_M+{}_Mx_2$. That is, v_M is an output of a randomizable sampler having an error ${}_MX_2$ for $f_M(x_M)$. The reason will be described later (step S**6108**).

The determining unit **6111** determines whether or not there is one that satisfies $u_{M'}=v_M$ among the pairs $(u_M, u_{M'})$ stored in the first list storage **6107** and v_M stored in the second list storage **6110** (step S**6110**). If there is one that satisfies $u_{M'}=v_M$, the process proceeds to step S**6114**; if there is not one that satisfies $u_{M'}=v_M$, the process proceeds to step S**1111**.

At step S**1111**, the controller **1113** determines whether or not $t=T_{max}$ (step S**1111**). Here, T_{max} is a predetermined natural number. If $t=T_{max}$, the controller **1113** outputs information indicating that the computation is impossible, for example, the symbol "⊥" (step S**1113**), then the process ends. If not $t=T_{max}$, the controller **1113** increments t by 1, that is, sets $t=t+1$ (step S**1112**), then the process returns to step S**6103**.

At step S**6114**, the final output unit **6112** outputs u_M corresponding to $u_{M'}$ that has been determined to satisfy $u_{M'}=v_M$ (step S**6114**). The u_M thus obtained can be a decryption result $f_M(x_M)$ resulting from decrypting the ciphertext x_M with the decryption key SK with a high probability (the reason will be described later). Therefore, the process described above is repeated multiple times and the value most frequently obtained among the values obtained at step S**6114** can be chosen as the decryption result. Depending on settings, $u_M=f_M(x_M)$ can result with an overwhelming probability. In that case, the value obtained at step S**6114** can be directly provided as the result of decryption.

<<Reason Why ${}_Mz_1-M_{R1}$ and ${}_Mz_2-M_{R2}$ are Outputs of a Sampler and Randomizable Sampler that have Errors ${}_MX_1$ and ${}_MX_2$, Respectively, for $f_M(x_M)$>>

Because of the homomorphy of $f_M(x_M)$, $f_M(x_M\cdot C_a+C_{R2})=f_M(x_M)\cdot f_M(C_a)+f_M(C_{R2})=f_M(x_M)\cdot a_M+M_{R2}$ is satisfied, $f_M(x_M)\cdot a_M=f_M(x_M\cdot C_a+C_{R2})-M_{R2}=f_M({}_M\tau_2)-M_{R2}$ is satisfied, and $M_{R2}=f_M({}_M\tau_2)-f_M(x_M)\cdot a_M$ is satisfied. Therefore, letting ${}_Mz_2=F_M({}_M\tau_2)$, then ${}_Mz_2-M_{R2}=F_M({}_M\tau_2)-f_M({}_M\tau_2)+f_M(x_M)\cdot a_M=f_M(x_M)\cdot a_M+\{F_M({}_M\tau_2)-f_M({}_M\tau_2)\}$ is satisfied. Because of the uniform randomness of CM_{21}, CM_{22} and M_{R2} corresponding to ${}_M\tau_2$, ${}_Mz_2-M_{R2}$ is statistically close to $f_M(x_M)\cdot a_M+{}_Mx_2$. Here, ${}_Mx_2$ is a realization of the random variable ${}_MX_2=F_M(ENC_M({}_MU_2))-{}_MU_2$ (${}_MU_2$ uniformly randomly distributes on G_M). Therefore, ${}_Mz_2-M_{R2}$ is an output of a randomizable sampler having an error ${}_MX_2$ for $f_M(x_M)$.

Likewise, $f_M(x_M\cdot C_{UM}+C_{R1})=f_M(x_M)\cdot f_M(C_{UM})+f_M(C_{R1})=f_M(x_M)\cdot UM+M_{R1}$ is satisfied, $f_M(x_M)=f_M(x_M\cdot C_{UM}+C_M)-M_{R1}=f_M({}_M\tau_1)-M_{R1}$ is satisfied, and $M_{R1}=f_M({}_M\tau_1)-f_M(x_M)$ is satisfied. Therefore, letting ${}_MZ_1=F_M({}_M\tau_1)$, then ${}_Mz_1-M_{R1}=F_M({}_M\tau_1)-f_M({}_M\tau_1)+f_M(x_M)=f_M(x_M)+\{F_M({}_M\tau_1)-f_M({}_M\tau_1)\}$ is satisfied. Because of the uniform randomness of CM_{11}, CM_{12} and M_{R1} corresponding to ${}_M\tau_1$, ${}_Mz_1-M_{R1}$ is statistically close to $f_M(x_M)+{}_Mx_1$. Here, ${}_Mx_1$ is a realization of the random variable ${}_MX_1=F_M(ENC_M({}_MU_1))-{}_MU_1$ (${}_MU_1$ uniformly randomly distributes on G_M). Therefore, the above-described configuration which outputs ${}_Mz_1-M_{R1}$ serves as a sampler having an error ${}_MX_1$ for $f_M(x_M)$.

<<Reason why Decryption Result $f_M(x_M)$ can be Obtained>>

For the same reason described in the section <<Reason why decryption result f(x) can be obtained>> in the fifth embodiment, a correct decryption result $f_M(x_M)$ can be obtained in the sixth embodiment as well. However, since the sixth embodiment deals with matrices, G and H in the section <<Reason why decryption result f(x) can be obtained>> in the fifth embodiment are replaced with G_M and H_M, f(x) is replaced with $f_M(x_M)$, τ is replaced with ${}_M\tau$, F(τ) is replaced with $F_M({}_M\tau)$, z is replaced with ${}_Mz$, x is replaced with x_M, X_1 and X_2 are replaced with ${}_MX_1$ and ${}_MX_2$, x_1 and x_2 are replaced with ${}_Mx_1$ and ${}_Mx_2$, e_g is replaced with a unit matrix ${}_Me_g$ of ι×ι, and multiplicative expressions are replaced with additive expressions (for example $\alpha^\beta\gamma$ is replaced with $\alpha\cdot\beta+\gamma$). Furthermore, "pseudo-free action" in the sixth embodiment is defined as follows.

Pseudo-free action:

An upper bound of the probability

$$Pr[\alpha_M\cdot a_M \text{ and } \alpha_M\neq{}_Me_g|a_M\in_U\Omega_M,\alpha_M\in_M X_1,\beta_M\in_M X_2]$$

of satisfying $\alpha_M\cdot a_M=\beta_M$ for all possible ${}_MX_1$ and ${}_MX_2$ is called a pseudo-free indicator of a pair (G_M, Ω_M) and is represented as $P(G_M, \Omega_M)$, where G_M is a matrix, Ω_M is a set of matrices $\Omega_M=\{0_M, \ldots, M_M\}$, α_M and β_M are realizations $\alpha_M\in_M X_1$ ($\alpha_M\neq{}_Me_g$) and $\beta_M\in_M X_2$ of random variables ${}_MX_1$ and ${}_MX_2$ on G_M, and $a_M\in\Omega_M$. If a certain negligible function ζ(k) exists and

$$P(G_M,\Omega_M)<\zeta(k)$$

then a computation defined by the pair (G_M, Ω_M) is called a pseudo-free action.

[Variations of First to Sixth Embodiments]

As has been described above, the capability providing apparatus provides first output information z_1 and second output information z_2 to the computing apparatus without providing a decryption key and the computing apparatus outputs $u^{b'}v^{a'}$ in the embodiments described above. The probability of $u^{b'}v^{a'}$ being the decryption value of the ciphertext x is high. Thus, the capability providing apparatus

can provide the decryption capability to the computing apparatus without providing a decryption key.

Note that the present invention is not limited to the embodiments described above. For example, the random variables X_1, X_2 and X_3 may be the same or different. Similarly, the random variables ${}_MX_1$ and ${}_MX_2$ may be the same or different.

Each of the first random number generator, the second random number generator, the third random number generator, the fourth random number generator, the fifth random number generator, the sixth random number generator and the seventh random number generator generates uniform random numbers to achieve the highest security of the proxy computing system. However, if the level of security required is not so high, at least some of the first random number generator, the second random number generator, the third random number generator, the fourth random number generator, the fifth random number generator, the sixth random number generator and the seventh random number generator may generate random numbers that are not uniform random numbers. Similarly, a non-uniform random matrix may be selected in the sixth embodiment instead of uniformly randomly selecting a matrix. While it is desirable from the computational efficiency point of view that random numbers which are natural numbers greater than or equal to 0 and less than K_H or random numbers that are natural numbers greater than or equal to 0 and less than K_G be selected as in the embodiments described above, random numbers that are natural numbers greater than or equal to K_H or K_G may be selected instead.

The process of the capability providing apparatus may be performed multiple times each time the computing apparatus provides first input information τ_1 and second input information τ_2 which are elements of a group H and correspond to the same a and b to the capability providing apparatus. This enables the computing apparatus to obtain a plurality of pieces of first output information z_1, second output information z_2, and third output information z_3 each time the computing apparatus provides first input information τ_1 and the second input information τ_2 to the capability providing apparatus. Consequently, the number of exchanges and the amount of communication between the computing apparatus and the capability providing apparatus can be reduced. The same applies to the first input information ${}_M\tau_1$ and the second input information ${}_M\tau_2$ of the sixth embodiment.

The computing apparatus may provide a plurality of pieces of first input information τ_1 and second input information τ_2 to the capability providing apparatus at once and may obtain a plurality of pieces of corresponding first output information z_1, second output information z_2 and third output information z_3 at once. This can reduce the number of exchanges between the computing apparatus and the capability providing apparatus. The same applies to the first input information ${}_M\tau_1$ and the second input information ${}_M\tau_2$ of the sixth embodiment.

Check may be made to see whether u and v obtained at the first computing unit and the second computing unit of any of the first to fifth embodiments are elements of the group G. If u and v are elements of the group the process described above may be continued; if u or v is not an element of the group information indicating that the computation is impossible, for example, the symbol "⊥" may be output. Similarly, check may be made to see whether u_M and v_M obtained at the first computing unit and the second computing unit of the sixth embodiment is an element of G_M. If u_M and v_M are elements of G_M, the process described above may be continued; if u_M or G_M is not an element of G_M, information indicating that the computation is impossible, for example, the symbol "⊥" may be output.

The units of the computing apparatus may exchange data among them directly or through a memory, which is not depicted. Similarly, the units of the capability providing apparatus may exchange data among them directly or through a memory, which is not depicted.

Furthermore, the processes described above may be performed in time sequence as described, or may be performed in parallel with one another or individually, depending on the throughput of the apparatuses that performs the processes or as needed. It would be understood that other modifications can be made without departing from the spirit of the present invention.

Seventh Embodiment

A seventh embodiment of the present invention will be described.

<Configuration>

As illustrated in FIG. **18**, a proxy computing system **101** of the seventh embodiment includes, for example, a computing apparatus **111** that not have a decryption key, capability providing apparatuses **112-1**, . . . , **112-Γ** (Γ is an integer greater than or equal to 2) that has decryption keys $s_1, \ldots, s_\Gamma$, respectively, and a decryption control apparatus **113** that controls a decryption capability of the computing apparatus **111**. The decryption control apparatus **113** controls a decryption capability provided by the capability providing apparatuses **112-1**, . . . , **112-Γ** to the computing apparatus **111** and the computing apparatus **111** uses the decryption capability provided by the capability providing apparatus **112-1**, . . . , **112-Γ** to decrypt a ciphertext. The computing apparatus **111**, the capability providing apparatuses **112-1**, . . . , **112-Γ**, and the decryption control apparatus **113** are configured so that information can be exchanged between them. For example, the computing apparatus **111**, the probability providing apparatuses **112-1**, . . . , **112-Γ**, and the decryption control apparatus **113** are capable of exchange information through a transmission line, a network, a portable recording medium, and/or other medium.

As illustrated in FIG. **19**, the computing apparatus **111** of the seventh embodiment includes, for example, a natural number storage **11101**, a natural number selecting unit **11102**, an integer computing unit **11103**, an input information providing unit **11104**, a first computing unit **11105**, a first power computing unit **11106**, a first list storage **11107**, a second computing unit **11108**, a second power computing unit **11109**, a second list storage **11110**, a determining unit **11111**, a final output unit **11112**, a recovering unit **11100**, and a controller **11113**. Examples of the computing apparatus **111** include a device having a computing function and a memory function, such as a card reader-writer apparatus and a mobile phone, and a well-known or specialized computer that includes a CPU (central processing unit) and a RAM (random-access memory) in which a special program is loaded.

As illustrated in FIG. **20**, the capability providing apparatus **112-1** (ι=1, . . . ω, where ω is an integer greater than or equal to 2 and less than or equal to Γ) of this embodiment includes, for example, a first output information computing unit **11201-ι**, a second output information computing unit **11202-ι**, a key storage **11204-ι**, and a controller **11205-ι**. Examples of the capability providing apparatus **112-ι** include a tamper-resistant module such as an IC card and an IC chip, a device having computing and memory functions, such as a mobile phone, and a well-known or specialized

computer including a CPU and a RAM in which a special program is loaded. As will be described later, the capability providing apparatuses **112-1**, . . . , **112-ω** are selected from the capability providing apparatuses **112-1**, . . . , **112-Γ**. If there are capability providing apparatuses **112-(ι+1)**, . . . , **112-Γ**, the capability providing apparatuses **112-(ι+1)**, . . . , **112-Γ** have the same configuration as the capability providing apparatus **112-ι**.

As illustrated in FIG. **21**, the decryption control apparatus **113** of the seventh embodiment includes, for example, an ciphertext storage **11301**, a control instruction unit **11302**, an output unit **11303**, a controller **11304**, a key storage **11305**, and an encryption unit **11306**. Examples of the decryption control apparatus **113** include a device having computing and memory functions, such as a mobile phone, and a well-known or specialized computer that includes a CPU and a RAM in which a special program is loaded.

<Processes>

Processes of this embodiment will be described below. For the processes, let G_ι and H_ι be groups (for example commutative groups), ω be an integer greater than or equal to 2, $\iota=1, \ldots, \omega$, $f_\iota(\lambda_\iota)$ be a decryption function for decrypting a ciphertext λ_ι which is an element of the group H_ι with a particular decryption key s_ι to obtain an element of the group Gι, generators of the groups G_ι and H_ι be $\mu_{\iota,g}$ and $\mu_{\iota,h}$, respectively, $X_{\iota,1}$ and $X_{\iota,2}$ be random variables having values in the group G_ι, $x_{\iota,1}$ be a realization of the random variable $X_{\iota,1}$, and $x_{\iota,2}$ be a realization of the random variable $X_{\iota,2}$. Note that ω in this embodiment is a constant. It is assumed here that a plurality of pairs of natural numbers a(ι) and b(ι) that are relatively prime to each other (a(ι), b(ι)) are stored in the natural number storage **11101** of the computing apparatus **111**. The term "natural number" means an integer greater than or equal to 0. Let I_ι be a set of pairs of relatively prime natural numbers that are less than the order of the group G_ι, then it can be considered that pairs (a(ι), b(ι)) of natural numbers a(ι) and b(ι) corresponding to a subset S_ι of I_ι are stored in the natural number storage **11101**. It is also assumed that a particular decryption key sι is stored in the key storage **12104** of the capability providing apparatus **112-ι** in a secure manner. It is assumed that encryption keys $pk_1, \ldots, pk_\Gamma$ corresponding to the decryption keys $s_1, \ldots, s_\Gamma$, respectively, are stored in the key storage **11305** of the decryption control apparatus **113**. Examples of the decryption keys s_ι and encryption keys pk_ι are secret keys and public keys of public key cryptography. Processes of the computing apparatus **111** are performed under the control of the controller **11113**, processes of the capability providing apparatus **112-ι** are performed under the control of the controller **11205-ι**, and processes of the decryption control apparatus **113** are performed under the control of the controller **11304**.

<Encryption Process>

As illustrated in FIG. **24**, first, a message mes is input in the encryption unit **11306** of the decryption control apparatus **113** (FIG. **21**). The encryption unit **11306** randomly selects ω encryption keys $pk_1, \ldots, pk_\omega$ from the encryption keys $pk_1, \ldots, pk_\Gamma$ (step **S11301**). The encryption unit **11306** generates ω shares $sha_1, \ldots, sha_\omega$ from the message mes(step **S11302**). A method for generating the shares $sha_1, \ldots, sha_\omega$ will be described below.

Example 1 of Shares

Shares $sha_1, \ldots, sha_\omega$ are generated so that a bit combination value $sha_1| \ldots |sha_\omega$ of the ω shares $sha_1, \ldots, sha_\omega$ is the message mes.

Example 2 of Shars

Shares $sha_1, \ldots, sha_\omega$ are generated so that the exclusive OR of the ω shares $sha_1, \ldots, shar_\omega$ is the message mes.

Example 3 of Shars

The message mes is secret-shared by a secret sharing scheme such as Shamir's secret sharing to generate shares $sha_1, \ldots, sha_\omega$ (End of description of examples of the method for generating shares).

Then, the encryption unit **11306** encrypts the share sha_ι with an encryption key pk_ι to generate a ciphertext λ_ι for each of $\iota=1, \ldots, \omega$. The generated ciphertexts $\lambda_1, \ldots, \lambda_\omega$ are stored in the ciphertext storage **11301** (step **S11303**).

Then, the ciphertexts $\lambda_1, \ldots, \lambda_\omega$ stored in the ciphertext storage **11301** are output from the output unit **11301** and input in the computing apparatus **111** (FIG. **19**) (step **S11304**). The ciphertexts $\lambda_1, \ldots, \lambda_\omega$ may or may not be sent at a time.

<Decryption Process>

A decryption process for decrypting a ciphertext λ_ι according to this embodiment will be described with reference to FIG. **25**. The process described below is performed for each of $\iota=1, \ldots, \omega$.

First, the natural number selecting unit **11102** of the computing apparatus **111** (FIG. **19**) randomly reads one pair (a(ι), b(ι)) of natural numbers from among a plurality of pairs of natural numbers (a(ι), b(ι)) stored in the natural number storage **11101**. At least part of information on the read pair of natural numbers (a(ι), b(ι)) is sent to the integer computing unit **11103**, the input information providing unit **11104**, the first power computing unit **11106**, and the second power computing unit **11109** (step **S11100**).

The integer computing unit **11103** uses the sent pair of natural numbers (a(ι), b(ι)) to compute integers a'(ι) and b'(ι) that satisfy the relation a'(ι)a(ι)+b'(ι) b(ι)=1. Since the natural numbers a(ι) and b(ι) are relatively prime to each other, the integers a'(ι) and b'(ι) that satisfy the relation a'(ι) a(ι)+b'(ι)b(ι)=1 definitely exist. Methods for computing such integers are well known. For example, a well-known algorithm such as the extended Euclidean algorithm may be used to compute the integers a'(ι) and b'(ι). Information on the pair of natural numbers (a'(ι), b'(ι)) is sent to the final output unit **11112** (step **S11101**).

The controller **11113** sets $t_\iota=1$ (step **S11102**).

The input information providing unit **11104** of the computing apparatus **111** generates and outputs first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ which are elements of a group H_ι and each of which corresponds to the input ciphertext λ_ι. Preferably, the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ are information whose relation with the ciphertext λ_ι is scrambled. This enables the computing apparatus **111** to conceal the ciphertext λ_ι from the capability providing apparatus **112-ι**. Preferably, the first input information $\tau_{\iota,1}$ of this embodiment further corresponds to the natural number b(ι) selected by the natural number selecting unit **11102** and the second input information $\tau_{\iota,2}$ further corresponds to the natural number a(ι) selected by the natural number selecting unit **11102**. This enables the computing apparatus **111** to evaluate the decryption capability provided by the capability providing apparatus **112-ι** with a high degree of accuracy (step **S11103**).

As illustrated in FIG. **26**, the first input information $\tau_{\iota,1}$ is input in the first output information computing unit **11201-ι** of the capability providing apparatus **112-ι** (FIG. **20**) and the

second input information $\tau_{\iota,2}$ is input in the second output information computing unit **11202**-ι (step S**11200**).

The first output information computing unit **11201** uses the first input information $\tau_{\iota,1}$ and the decryption key s_ι stored in the key storage **11204**-ι to correctly compute $f_\iota(\tau_{\iota,1})$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\iota,1}$ (step S**11201**). The second output information computing unit **11202**-ι uses the second input information $\tau_{\iota,2}$ and the decryption key s_ι stored in the key storage **11204**-ι to correctly computes $f_\iota(\tau_{\iota,2})$ with a probability greater than a certain probability and sets the result of the computation as second output information $z_{\iota,2}$ (step S**11202**). Note that the "certain probability" is a probability less than 100%. An example of the "certain probability" is a nonnegligible probability and an example of the "nonnegligible probability" is a probability greater than or equal to $1/\psi(k)$, where $\psi(k)$ is a polynomial that is a weakly increasing function (non-decreasing function) for a security parameter k. That is, the first output information computing unit **11201**-ι and the second output information computing unit **11202**-ι can output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **11201**-ι may or may not be $f_\iota(\tau_{\iota,1})$ and the result of the computation by the second output information computing unit **11202**-ι may or may not be $f_\iota(\tau_{\iota,2})$. The first output information computing unit **11201**-ι outputs the first output information $z_{\iota,1}$ and the second output information computing unit **11202**-ι outputs the second output information $z_{\iota,2}$ (step S**11203**).

Returning to FIG. **25**, the first output information $z_{\iota,1}$ is input in the first computing unit **11105** of the computing apparatus **111** (FIG. **19**) and the second output information $z_{\iota,2}$ is input in the second computing unit **11108**. The first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ are equivalent to the decryption capability provided by the capability providing apparatus **112**-ι to the computing apparatus **111** (step S**11104**).

The first computing unit **11105** generates computation result $u_\iota=f_\iota(\lambda_\iota)^{b(\iota)}x_{\iota,1}$ from the first output information $z_{\iota,1}$. Here, generating (computing) $f_\iota(\lambda_\iota)^{b(\iota)}{}_{x\iota,1}$ means computing a value of a formula defined as $f_\iota(\lambda_\iota)^{b(\iota)}x_{\iota,1}$. Any intermediate computation method may be used, provided that the value of the formula $f_\iota(\lambda_\iota)^{b(\iota)}{}_{x\iota,1}$ can eventually be computed. The same applies to computations of the other formulae that appear herein. The result u_ι of the computation is sent to the first power computing unit **11106** (step S**11105**).

The first power computing unit **11106** computes $u_\iota'=u_\iota^{a(\iota)}$. The pair of the result u_ι of the computation and u_ι' computed on the basis of the result of the computation, (u_ι, u_ι'), is stored in the first list storage **11107** (step S**11106**).

The determining unit **11111** determines whether or not there is a pair that satisfies $u_\iota'=v_\iota'$ among the pairs (u_ι, u_ι') stored in the first list storage **11107** and the pairs (v_ι, v_ι') stored in the second list storage **11110** (step S**11107**). If no pair (v_ι, v_ι') is stored in the second list storage **11110**, the process at step S**11107** is omitted and the process at step S**11108** is performed. If there is a pair that satisfies $u_\iota'=v_\iota'$, the process proceeds to step S**11114**; if there is not a pair that satisfies $u_\iota'=v_\iota'$, the process proceeds to step S**11108**.

At step S**11108**, the second computing unit **11108** generates a computation result $v_\iota=f_\iota(\lambda_\iota)^{a(\iota)}x_{\iota,2}$ from the second output information $z_{\iota,2}$. The result v_ι of the computation is sent to the second power computing unit **11109** (step S**11108**).

The second power computing unit **11109** computes $v_\iota'=v_\iota^{b(\iota)}$. The pair of the result v_ι of the computation and v_ι' computed on the basis of the computation result, (v_ι, v_ι'), is stored in the second list storage **11110** (step S**11109**).

The determining unit **11111** determines whether or not there is a pair that satisfies $u_\iota'=v_\iota'$ among the pairs (u_ι, u_ι') stored in the first list storage **11107** and the pairs (v_ι, v_ι') stored in the second list storage **11110** (step S**11110**). If there is a pair that satisfies $u_\iota'=v_\iota'$, the process proceeds to step S**11114**. If there is not a pair that satisfies $u_\iota'=v_\iota'$, the process proceeds to step S**11111**.

At step S**11111**, the controller **11113** determines whether or not $t_\iota=T_\iota$ (step S**11111**). Here, T_ι is a predetermined natural number. If $t_\iota=T_\iota$, the final output unit **11112** outputs information indicating that the computation is impossible, for example the symbol "⊥" (step S**11113**) and the process ends. If not $t_\iota=T_\iota$, the controller **11113** increments t_ι by 1, that is, sets $t_\iota=t_\iota+1$ (sets $t_\iota+1$ as a new t_ι) (step S**11112**) and the process returns to step S**11103**.

The information indicating the computation is impossible (the symbol "⊥" in this example) means that the reliability that the capability providing apparatus **112**-ι correctly performs computation is lower than a criterion defined by T. In other words, it means that the capability providing apparatus **112**-ι was unable to perform a correct computation in T_ι trials.

At step S**11114**, the final output unit **11112** uses u_ι and v_ι that correspond to u_ι' and v_ι' that are determined to satisfy $u_\iota'=v_\iota'$ to calculate and output $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ (step S**11114**). The $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ thus computed will be a result $f_\iota(\lambda_\iota)$ of decryption of the ciphertext λ_ι with the particular decryption key s_ι with a high probability (the reason why $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}=f_\iota(\lambda_\iota)$ with a high probability will be described later). Therefore, the process described above is repeated multiple times and the value obtained with the highest frequency among the values obtained at step S**11114** can be provided as the result of decryption $f_\iota(\lambda)$. As will be described later, $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}=f_\iota(\lambda_\iota)$can result with an overwhelming probability, depending on settings. In that case, the value obtained at step S**11114** can be directly provided as the result of decryption $f_\iota(\lambda_\iota)$.

The decryption results $f_\iota(\lambda_\iota)$ obtained by performing the process described above on each of ι=1, . . . , ω are input in the recovering unit **11100**. The recovering unit **11100** uses $f_\iota(\lambda_\iota)=u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ for each ι=1, . . . , ω to perform a recovering process for obtaining a recovered value that can be recovered only if all of decrypted values that can be obtained by decrypting a ciphertext λ_ι for each ι=1, . . . , ω with the decryption key s_ι are obtained. For example, if the shares have been generated by <<Example 1 of shares>> described above, the recovering unit **11100** generates a bit combination value $f_1(\lambda_1)|\ldots|f_\omega(\lambda_\omega)$ as the recovered value mes'. For example if the shares have been generated by <<Example 2 of shares>> described above, the recovering unit **11100** generates the exclusive OR of the decryption results $f_1(\lambda_1), \ldots, f_\omega(\lambda_\omega)$ as the recovered value mes'. For example, if the shares have been generated by <<Example 3 of shares>> described above, the recovering unit **11100** generates the recovered value mes' from the decryption results $f_1(\lambda_1), \ldots, f_\omega(\lambda_\omega)$ by using a recovering method corresponding to the secret sharing scheme.

If all of the decryption results $f_1(\lambda_1), \ldots, f_\omega(\lambda_\omega)$ are correct, the recovered value mes' obtained by the recovering unit **11100** is equal to the message mes. On the other hand, if all of the decryption results $f_1(\lambda_1), \ldots, f_\omega(\lambda_\omega)$ are incorrect, the probability that the recovered value mes' obtained by the recovering unit **11100** is equal to the message mes is negligibly small.

<<Reason Why $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}=f_\iota(\lambda_\iota)$ with High Probability>>

For simplicity of notation, ι is omitted in the following description.

Let X be a random variable having a value in the group G. For w∈G, an entity that returns wx' corresponding to a sample x' according to the random variable X in response to each request is called a sampler having an error X for w.

For w∈G, an entity that returns w^ax' corresponding to a sample x' according to a random variable X whenever a natural number a is given is called a randomizable sampler having an error X for w. The randomizable sampler functions as the sampler if used with a=1.

The combination of the input information providing unit **11104**, the first output information computing unit **11201** and the first computing unit **11105** of this embodiment is a randomizable sampler having an error X_1 for f(λ) (referred to as the "first randomizable sampler") and the combination of the input information providing unit **11104**, the second output information computing unit **11202** and the second computing unit **11108** is a randomizable sampler having an error X_2 for f(λ) (referred to as the "second randomizable sampler").

The inventor has found that if u'=v' holds, that is, if $u^a=v^b$ holds, it is highly probable that the first randomizable sampler has correctly computed $u=f(\lambda)^b$ and the second randomizable sampler has correctly computed $v=f(\lambda)^a$(x_1 and x_2 are identity elements e_g of the group G). For simplicity of explanation, this will be proven in an eleventh embodiment.

When the first randomizable sampler correctly computes $u=f(\lambda)^b$ and the second randomizable sampler correctly computes $v=f(\lambda)^a$ (when x_1 and x_2 are identity elements e_g of the group G), then $u^{b'}v^{a'}=(f(\lambda)^b x_1)^{b'}(f(\lambda)^a x_2)^{a'}=(f(\lambda)^b e_g)^{b'}(f(\lambda)^a e_g)^{a'}=f(\lambda)^{bb'}e_g{}^{b'}f(\lambda)^{aa'}e_g{}^{a'}=f(\lambda)^{(bb'+aa')}=f(\lambda)$.

For $(q_1, q_2)\in I$, a function π_i is defined by $\pi_i(q_i, q_2)=q_i$ for each of i=1, 2. Let L=min (#π_1(S), #π_2(S)), where #· is the order of a set ·. If the group G is a cyclic group or a group whose order is difficult to compute, it can be expected that the probability that an output other than "⊥" of the computing apparatus **111** is not f(λ) is at most approximately $T^2L/\#S$ within a negligible error. If L/#S is a negligible quantity and T is a quantity approximately equal to an polynomial order, the computing apparatus **111** outputs a correct $f_\iota(\lambda)$ with an overwhelming probability. An example of S that results in a negligible quantity of L/#S is $S=\{(1, d)|d\in[2, |G|-1]\}$.

<Decryption Control Process>

A decryption control process of this embodiment will be described below.

When the decryption control apparatus **113** controls the decryption process performed by the computing apparatus **111**, the decryption control apparatus **113** outputs a decryption control instruction that controls the decryption process of the computing apparatus **111** to all of the capability providing apparatuses **112**-ι. The capability providing apparatuses **112**-ι in which the decryption control instruction is input controls whether to output both of first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ according to the input decryption control instruction. The computing apparatus **111** cannot decrypt the ciphertext λ_ι unless the first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ are provided. Therefore, the decryption capability of the computing apparatus **111** can be controlled by controlling whether or not to output both of the first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$. Exemplary methods for controlling the decryption process will be described below.

Example 1 of Method for Controlling Decryption Process

In example 1 of the method for controlling the decryption process, decryption control instructions include a decryption restricting instruction com_1-ι for restricting the decryption capability of the computing apparatus **111**. When the decryption restricting instruction com_1-ι is input in the controller **11205**-ι of the capability providing apparatus **112**-ι, the controller **11205**-ι, prevents output of both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$.

To restrict the decryption capability of the computing apparatus **111**, the control instruction unit **11302** of the decryption control apparatus **113** (FIG. **21**) outputs the decryption restricting instruction com_1-ι for all ι. The decryption restricting instruction com_1-ι is output from the output unit **11303** to the capability providing apparatus **112**-ι.

The controller **11205**-ι of the capability providing apparatus **112**-ι (FIG. **20**) determines whether or not the decryption restricting instruction com_1-ι has been input. If the decryption restriction instruction com_1-ι has not been input in the controller **11205**-ι, the controller **11205**-ι does not perform the decryption control process. On the other hand, if the decryption restriction instruction com_1-ι has been input in the controller **11205**-ι, the controller **11205**-ι performs control to prevent output of both of the first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ (decryption restriction mode).

In the decryption restriction mode, the controller **11205**-ι prevents the first output information computing unit **11201**-ι from outputting first output information $z_{\iota,1}$ and also prevents the second output information computing unit **11202**-ι from outputting second output information $z_{\iota,2}$. An example of control to prevent output of the first output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$ is control to prevent output of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ without preventing generation of the output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$. Another example of control to prevent output of the first output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$ is control to causing the first output information computing unit **11201**-ι and the second output information computing unit **11202**-ι to output dummy information instead of the first output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$. Note that an example of the dummy information is a random number or other information that is independent of the ciphertexts λ_ι. Another example of control to prevent output of the first output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$ is control to prevent generation of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$. If control to prevent generation of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is performed, information required for generating the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ may optionally be nullified or removed. For example, the decryption key s_ι stored in the key storage **11204**-ι may optionally be nullified or removed.

If output of both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is prevented, the capability providing apparatus **112**-ι outputs neither of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ at

step S11203. Accordingly, the computing apparatus **111** can obtain neither of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ at step S11104 and cannot compute computation results u_ι and v_ι. Therefore a correct decryption result $f_\iota(\lambda)$ cannot be obtained. If correct decryption results $f_\iota(\lambda)$ cannot be obtained for all ι, the probability that a recovered value mes' obtained by the recovering unit **11100** will be equal to the message mes is negligibly small. Thus, the decryption capability of the computing apparatus **111** can be restricted.

Example 2 of Method for Controlling Decryption Process

In example 2 of the method for controlling the decryption process, the decryption control instructions include a decryption permitting instruction com_2-ι for removing restriction on the decryption capability of the computing apparatus **111**. When the decryption permitting instruction com_2-ι is input in the controller **11205**-ι of the capability providing apparatus **112**-ι, the controller **11205**-ι permits output of at least one of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$. Example 2 of the method for controlling the decryption process is performed when, for example, output of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is to be permitted again after output of both of first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ is prevented by example 1 of the method for controlling the decryption process. In that case, if information required for generating the first output information $z_{\iota,1}$ and/or the second output information $z_{\iota,2}$ has been nullified or removed, the decryption permitting instruction com_2-ι may include the information and the information may be re-set in the capability providing apparatus **112**-ι. Example 2 of the method for controlling the decryption process may also be performed to permit output of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ when, for example, output of both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is prevented in an initial state.

To remove restriction on the decryption capability of the computing apparatus **111**, the control instruction unit **11302** of the decryption control apparatus **113** (FIG. **21**) outputs the decryption permitting instruction com_2-ι for all ι. The decryption permitting instruction com_2-ι is output from the output unit **11303** to the capability providing apparatus **112**-ι.

The controller **112054** of the capability providing apparatus **112**-ι (FIG. **20**) determines whether or not the decryption permitting instruction com_2-ι is input. If the decryption permitting instruction com_2-ι is not input in the controller **11205**-ι, the controller **11205**-ι does not perform decryption control process. On the other hand, if the decryption permitting instruction com_2-ι is input in the controller **11205**-ι, the controller **11205**-ι performs control to permit output of both of first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ (decryption permission mode).

If output of both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is permitted, the capability providing apparatus **112**-ι outputs both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ at step S11203. Accordingly, the computing apparatus **111** can obtain both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ at step S11104 and can compute u_ι or v_ι as a computation result. Therefore a correct decryption result can be obtained with a high probability. Thus, the restriction on the decryption capability of the computing apparatus **111** can be removed.

Example 3 of Method for Controlling Decryption Process

In examples 1 and 2 of the method for controlling the decryption process, each decryption control instruction corresponds to any one of ι (corresponds to the decryption function f_ι) and the controller **11205**-ι of the capability providing apparatus **112**-ι controls whether to output all of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ that correspond to the decryption control instruction (corresponds to the decryption function f_ι corresponding to the decryption control instruction). However, a decryption control instruction may correspond to a plurality of ι and the controller **11205**-ι of the capability providing apparatus **112**-ι may control whether to output the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ corresponding to the decryption control instruction.

Eighth Embodiment

A proxy computing system of an eighth embodiment is an example that embodies the first randomizable sampler and the second randomizable sampler described above. The following description will focus on differences from the seventh embodiment and repeated description of commonalities with the seventh embodiment, including the decryption control process, will be omitted. In the following description, elements labeled with the same reference numerals have the same functions and the steps labeled with the same reference numerals represent the same processes.

<Configuration>

As illustrated in FIG. **18**, the proxy computing system **102** of the eighth embodiment includes a computing apparatus **121** in place of the computing apparatus **111** and capability providing apparatuses **122-1**, . . . , **122**-Γ in place of the capability providing apparatuses **112-1**, . . . , **112**-Γ.

As illustrated in FIG. **19**, the computing apparatus **121** of the eighth embodiment includes, for example, a natural number storage **11101**, a natural number selecting unit **11102**, an integer computing unit **11103**, an input information providing unit **12104**, a first computing unit **12105**, a first power computing unit **11106**, a first list storage **11107**, a second computing unit **12108**, a second power computing unit **11109**, a second list storage **11110**, a determining unit **11111**, a final output unit **11112** and a controller **11113**. As illustrated in FIG. **22**, the input information providing unit **12104** of this embodiment includes, for example, a first random number generator **12104***a*, a first input information computing unit **12104***b*, a second random number generator **12104***c*, and a second input information computing unit **12104***d*.

As illustrated in FIG. **20**, the capability providing apparatus **122**-ι of the eighth embodiment (ι=1, . . . , ω, where ω is an integer greater than or equal to 2 and less than or equal to Γ) includes, for example, a first output information computing unit **12201**-ι, a second output information computing unit **12202**-ι, a key storage **11204**-ι, and a controller **11205**-ι. If there are capability providing apparatuses **122**-(ι+1), . . . , **122**-Γ, the capability providing apparatuses **122**-(ι+1), . . . , **122**-Γ have the same configuration as the capability providing apparatus **122**-ι.

<Decryption Process>

A decryption process of this embodiment will be described below. In the eighth embodiment, a decryption

function f_ι is a homomorphic function, a group H is a cyclic group, and a generator of the group H is $\mu_{\iota,h}$, the order of the group H is $K_{\iota,H}$, and $v_\iota=f_\iota(\mu_{\iota,h})$. An example of cryptography in which the decryption function f_ι is a homomorphic function is RSA cryptography. The rest of the assumptions are the same as those in the seventh embodiment, except that the computing apparatus **111** is replaced with the computing apparatus **121** and the capability providing apparatuses **112-1**, . . . , **112-Γ** are replaced with the capability providing apparatuses **122-1**, . . . , **122-Γ**.

As illustrated in FIGS. **25** and **26**, the process of the eighth embodiment is the same as the process of the seventh embodiment except that steps S**11103** through S**11105**, S**11108**, and S**11200** through S**11203** of the seventh embodiment are replaced with steps S**12103** through S**12105**, S**12108**, and S**12200** through S**12203**, respectively. In the following, only processes at steps S**12103** through S**12105**, S**12108**, and S**12200** through S**12203** will be described.

<<Process at Step S**12103**>>

The input information providing unit **12104** of the computing apparatus **121** (FIG. **19**) generates and outputs first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ which corresponds to an input ciphertext λ_ι (step S**12103** of FIG. **25**). A process at step S**12103** of this embodiment will be described with reference to FIG. **27**.

The first random number generator **12104***a* (FIG. **22**) generates a uniform random number r(ι, 1) that is a natural number greater than or equal to 0 and less than $K_{\iota,H}$. The generated random number r(ι, 1) is sent to the first input information computing unit **12104***b* and the first computing unit **12105** (step S**12103***a*). The first input information computing unit **12104***b* uses the input random number r(ι, 1), the ciphertext λ_ι and a natural number b(ι) to compute first input information $\tau_{\iota,1}=\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)}$ (step S**12103***b*).

The second random number generator **12104***c* generates a uniform random number r(ι, 2) that is a natural number greater than or equal to 0 and less than $K_{\iota,H}$. The generated random number r(ι, 2) is sent to the second input information computing unit **12104***d* and the second computing unit **12108** (step S**12103***c*). The second input information computing unit **12104***d* uses the input random number r(ι, 2), the ciphertext λ_ι, and a natural number a(ι) to compute second input information $\tau_{\iota,2}\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)}$ (step S**12103***d*).

The first input information computing unit **12104***b* and the second input information computing unit **12104***d* output the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ thus generated (step S**12103***e*). Note that the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ in this embodiment are information whose relation with the ciphertext λ_ι is scrambled by using random numbers r(ι, 1) and r(ι, 2), respectively. This enables the computing apparatus **121** to conceal the ciphertext λ_ι from the capability providing apparatus **122**-ι. The first input information $\tau_{\iota,1}$ in this embodiment further corresponds to the natural number b(ι) selected by the natural number selecting unit **11102** and the second input information $\tau_{\iota,2}$ further corresponds to the natural number a(ι) selected by the natural number selecting unit **11102**. This enables the computing apparatus **121** to evaluate the decryption capability provided by the capability providing apparatus **122**-ι with a high degree of accuracy.

<<Processes at Steps S**12200** through S**12203**>>

As illustrated in FIG. **26**, first, the first input information $\tau_{\iota,1}=\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)}$ is input in the first output information computing unit **12201**-ι of the capability providing apparatus **122**-ι (FIG. **20**) and the second input information $\tau_{\iota,2}=\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)}$ is input in the second output information computing unit **12202**-ι (step S**12200**).

The first output information computing unit **12201**-ι uses the first input information $\tau_{\iota,1}=\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)}$ and a decryption key s_ι stored in the key storage **11204**-ι to correctly compute $f_\iota(\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)})$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\iota,1}$. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **12201**-ι may or may not be $f_\iota(\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)})$ (step S**12201**).

The second output information computing unit **12202**-ι uses the second input information $\tau_{\iota,2}=\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)}$ and the decryption key s_ι stored in the key storage **11204**-ι, to correctly compute $f_\iota(\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)})$ with a probability greater than a certain probability and provides the result of the computation as second output information $z_{\iota,2}$. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **12202**-ι may or may not be $f_\iota(\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)})$ (step S**12202**).

The first output information computing unit **12201**-ι outputs the first output information $z_{\iota,1}$ and the second output information computing unit **12202**-ι outputs the second output information $z_{\iota,2}$ (step S**12203**).

<<Processes at Steps S**12104** and S**12105**>>

Returning to FIG. **25**, the first output information $z_{\iota,1}$ is input in the first computing unit **12105** of the computing apparatus **121** (FIG. **19**) and the second input information $z_{\iota,2}$ is input in the second computing unit **12108**. The first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ are equivalent to the decryption capability provided by the capability providing apparatus **122**-ι to the computing apparatus **121** (step S**12104**).

The first computing unit **12105** uses the input random number r(ι, 1) and the first output information $z_{\iota,1}$ to compute $z_{\iota,1}v_\iota{}^{-r(\iota,1)}$ and sets the result of the computation as u_ι. The result u_ι of the computation is sent to the first power computing unit **11106**. Here, $u_\iota=z^{\iota,1}v_\iota{}^{-r(\iota,1)}=f_\iota(\lambda_\iota)^{b(\iota)}x_{\iota,1}$. That is, $z_{\iota,1}v_\iota{}^{-r(\iota,1)}$ is an output of a randomizable sampler having an error $X_{\iota,1}$ for $f_\iota(\lambda_\iota)$. The reason will be described later (step S**12105**).

<<Process at Step S**12108**>>

The second computing unit **12108** uses the input random number r(ι, 2) and the second output information $z_{\iota,2}$ to compute $z_{\iota,2}v_\iota{}^{-r(\iota,2)}$ and sets the result of the computation as v_ι. The result v_ι of the computation is sent to the second power computing unit **11109**. Here, $v_\iota=z_{\iota,2}v_\iota{}^{-r(\iota,2)}=f_\iota(\lambda_\iota)^{a(\iota)}x_{\iota,2}$. That is, $z_{\iota,2}v_\iota{}^{-r(\iota,2)}$ is an output of a randomizable sampler having an error $X_{\iota,2}$ for $f_\iota(\lambda_\iota)$. The reason will be described later (step S**12108**).

<<Reason Why $z_{\iota,1}v_\iota{}^{-r(\iota,1)}$ and $z_{\iota,2}v_\iota{}^{-r(\iota,2)}$ are Outputs of Randomizable Samplers Having Errors $X_{\iota,1}$ and $X_{\iota,2}$, Respectively, for $f_\iota(\lambda_\iota)$>>

Let c be a natural number, R be a random number, and $B(\mu_h{}^R\lambda^c)$ be the result of computation performed by the capability providing apparatus **122** using $\mu_h{}^R\lambda^c$. That is, the results of computations that the first output information computing unit **12201**-ι and the second output information computing unit **12202**-ι return to the computing apparatus **121** are $z=B(\mu_h{}^R\lambda^c)$. A random variable X that has a value in the group G is defined as $X=B(\mu_h{}^{R'})f(\mu_h{}^{R'})^{-1}$.

Then, $zv^{-R}=B(\mu_h{}^R\lambda^c)f(\mu_h)^{-R}=Xf(\mu_h{}^R\lambda^c)f(\mu_h)^{-R}=Xf(\mu_h{}^R\lambda^c)f(\mu_h)^{-R}=Xf(\mu_h)^Rf(\lambda)^cf(\mu_h)^{-R}=f(\lambda)^cX$. That is, zv^{-R} is an output of a randomizable sampler having an error X for $f(\lambda)$.

The expansion of formula given above uses the properties such that $X=B(\mu_h{}^{R'})f(\mu_h{}^{R'})^{-1}=B(\mu_h{}^R\lambda^c)f(\mu_h{}^R\lambda^c)^{-1}$ and that

$B(\mu_h{}^R\lambda^c)=Xf(\mu_h{}^R\lambda^c)$. The properties are based on the fact that the function f_ι is a homomorphic function and R is a random number.

Therefore, considering that $a(\iota)$ and $b(\iota)$ are natural numbers and $r(\iota, 1)$ and $r(\iota, 2)$ are random numbers, $z_{\iota,1}v_\iota{}^{-r(\iota,1)}$ and $z_{\iota,2}v_\iota{}^{-r(\iota,2)}$ are, likewise, outputs of randomizable samplers having errors $X_{\iota,1}$ and $X_{\iota,2}$, respectively, for $f_\iota(\lambda_\iota)$.

Ninth Embodiment

A ninth embodiment is a variation of the eighth embodiment and computes a value of u_ι or v_ι by using samplers described above when $a(\iota)=1$ or $b(\iota)=1$. The amounts of computation performed by samplers in general are smaller than randomizable samplers. Using samplers instead of randomizable samplers for computations when $a(\iota)=1$ or $b(\iota)=1$ can reduce the amount of computation by the proxy computing system. The following description will focus on differences from the seventh and eighth embodiments and repeated description of commonalities with the seventh and eighth embodiments, including the decryption control process, will be omitted.

<Configuration>

As illustrated in FIG. **18**, a proxy computing system **103** of the ninth embodiment includes a computing apparatus **131** in place of the computing apparatus **121** and capability providing apparatuses **132-1**, . . . , **132-Γ** in place of the capability providing apparatuses **122-1**, . . . , **122-Γ**.

As illustrated in FIG. **19**, the computing apparatus **131** of the ninth embodiment includes, for example, a natural number storage **11101**, a natural number selecting unit **11102**, an integer computing unit **11103**, an input information providing unit **12104**, a first computing unit **12105**, a first power computing unit **11106**, a first list storage **11107**, a second computing unit **12108**, a second power computing unit **11109**, a second list storage **11110**, a determining unit **11111**, a final output unit **11112**, a controller **11113**, and a third computing unit **13109**.

As illustrated in FIG. **20**, the capability providing apparatus **132-ι** of the ninth embodiment includes, for example, a first output information computing unit **12201-ι**, a second output information computing unit **12202-ι**, a key storage **11204-ι**, a controller **11205-ι**, and a third output information computing unit **13203-ι**.

<Decryption Process>

A decryption process of this embodiment will be described below. Differences from the eighth embodiment will be described.

As illustrated in FIGS. **25** and **26**, the process of the ninth embodiment is the same as the process of the eighth embodiment except that steps **S12103** through **S12105**, **S12108**, and **S12200** through **S12203** of the eighth embodiment are replaced with steps **S13103** through **S13105**, **S13108**, **S12200** through **S12203**, and **S13205** through **13209**, respectively. The following description will focus on processes at steps **S13103** through **S13105**, **S13108**, **S12200** through **S12203**, and **S13205** through **S13209**.

<<Process at Step S13103>>

The input information providing unit **13104** of the computing apparatus **131** (FIG. **19**) generates and outputs first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ each of which corresponds to a input ciphertext λ_ι (step **S13103** of FIG. **25**).

A process at **S13103** of this embodiment will be described below with reference to FIG. **27**.

The controller **11113** (FIG. **19**) controls the input information providing unit **13104** according to natural numbers $(a(\iota), b(\iota))$ selected by the natural number selecting unit **11102**.

Determination is made by the controller **11113** as to whether b is equal to 1 (step **S13103***a*). If it is determined that $b\neq 1$, the processes at steps **S12103***a* and **12103***b* described above are performed and the process proceeds to step **S13103***g*.

On the other hand, if it is determined at step **S13103***a* that $b(\iota)=1$, the third random number generator **13104***e* generates a random number $r(\iota 3)$ that is a natural number greater than or equal to 0 and less than $K_{\iota,H}$. The generated random number $r(\iota, 3)$ is sent to the third input information computing unit **13104***f* and the third computing unit **13109** (step **S13103***b*). The third input information computing unit **13104***f* uses the input random number $r(\iota, 3)$ and the ciphertext λ_ι to compute $\lambda_\iota{}^{r(\iota,3)}$ and sets it as first input information $\tau_{\iota,1}$ (step **S13103***c*). Then the process proceeds to step **S13103***g*.

At step **S13103***g*, determination is made by the controller **11113** as to whether $a(\iota)$ is equal to 1 (step **S13103***g*). If it is determined that $a(\iota)\neq 1$, the processes at steps **S12103***c* and **S12103***d* described above are performed.

On the other hand, if it is determined at step **S13103***g* that $a(\iota)=1$, the third random number generator **3104***e* generates a random number $r(\iota, 3)$ that is a natural number greater than or equal to 0 and less than $K_{\iota,H}$. The generated random number $r(\iota, 3)$ is sent to the third input information computing unit **13104***f* (step **S13103***h*). The third input information computing unit **13104***f* uses the input random number $r(\iota, 3)$ and the ciphertext λ_ι to compute $\lambda_\iota{}^{r(\iota,3)}$ and sets it as second input information $t_{\iota,2}$ (step **S13103***i*).

The first input information computing unit **12104***b*, the second input information computing unit **12104***d*, and the third input information computing unit **13104***f* output the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ thus generated along with information on corresponding natural numbers $(a(\iota), b(\iota))$ (step **S13103***e*). Note that the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ in this embodiment are information whose relation with the ciphertext λ_ι is scrambled by using the random numbers $r(\iota, 1)$, $r(\iota, 2)$, and $r(\iota, 3)$, respectively. This enables the computing apparatus **131** to conceal the ciphertext λ_ι from the capability providing apparatus **132-ι**.

<<Processes at S12200 Through S12203 and S13205 Through S13209>>

Processes at **S12200** through **S12203** and **S13205** through **S13209** of this embodiment will be described below with reference to FIG. **26**.

The controller **112054** (FIG. **20**) controls the first output information computing unit **12201-ι**, the second output information computing unit **12202-ι**, and the third output information computing unit **13203-ι** according to input natural numbers $(a(\iota), b(\iota))$.

Under the control of the controller **11205-ι**, the first input information $\tau_{\iota,1}=\mu_{\iota,h}{}^{r(\iota,1)}\lambda_\iota{}^{b(\iota)}$ when $b(\iota)\neq 1$ is input in the first output information computing unit **12201-ι** of the capability providing apparatus **132-ι** (FIG. **20**) and the second input information $\tau_{\iota,2}=\mu_{\iota,h}{}^{r(\iota,2)}\lambda_\iota{}^{a(\iota)}$ when $a(\iota)\neq 1$ is input in the second output information computing unit **12202-ι**. The first input information $\tau_{\iota,1}=\lambda_\iota{}^{r(\iota,3)}$ when $b(\iota)=1$ and the second input information $\tau_{\iota,2}=\lambda_\iota{}^{r(\iota,3)}$ when $a(\iota)=1$ are input in the third output information computing unit **13203-ι** (step **S13200**).

Determination is made by the controller **11113** as to whether $b(\iota)$ is equal to 1 (step **S13205**). If it is determined

that $b(\iota)\neq 1$, the process at step S**12201** described above is performed. Then, determination is made by the controller **11113** as to whether $a(\iota)$ is equal to 1 (step S**13208**). If it is determined that $a(\iota)\neq 1$, the process at step S**12202** described above is performed and then the process proceeds to step S**13203**.

On the other hand, if it is determined at step S**13208** that $a(\iota)=1$, the third output information computing unit **13203**-ι uses the second input information $\tau_{\iota,2}=\lambda_\iota^{r(\iota,3)}$ to correctly compute $f_\iota(\lambda_\iota^{r(\iota,3)})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information $z_{\iota,3}$. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **13203**-ι may or may not be $f_\iota(\lambda_\iota^{r(\iota,3)})$ (step S**13209**). Then the process proceeds to step S**13203**.

If it is determined at step S**13205** that $b(\iota)=1$, the third output information computing unit **13203**-ι uses the second input information $\tau_{\iota,1}=\lambda_\iota^{r(\iota,3)}$ to correctly compute $f_\iota(\lambda_\iota^{r(\iota,3)})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information $z_{\iota,3}$. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **13203**-ι may or may not be $f_\iota(\lambda_\iota^{r(\iota,3)})$ (step S**13206**).

Then, determination is made by the controller **11113** as to whether $a(\iota)$ is equal to 1 (step S**13207**). If it is determined that $a(\iota)=1$, the process proceeds to step S**13203**; if it is determined $a(\iota)\neq 1$, the process proceeds to step S**12202**.

At step S**13203**, the first output information computing unit **12201**-ι, which has generated the first output information $z_{\iota,1}$, outputs the first output information $z_{\iota,1}$, the second output information computing unit **12202**-ι, which has generated the second output information $z_{\iota,2}$, outputs the second output information $z_{\iota,2}$, and the third output information computing unit **13203**-ι, which has generated the third output information $z_{\iota,3}$, outputs the third output information $z_{\iota,3}$ (step S**13203**).

<<Processes at Steps S**13104** and S**13105**>>

Returning to FIG. **25**, under the control of the controller **11113**, the first output information $z_{\iota,1}$ is input in the first computing unit **12105** of the computing apparatus **131** (FIG. **19**), the second output information $z_{\iota,2}$ is input in the second computing unit **12108**, and the third output information $z_{\iota,3}$ is input in the third computing unit **13109** (step S**13104**).

If $b(\iota)\neq 1$, the first computing unit **12105** performs the process at step S**12105** described above to generate u_ι; if $b(\iota)\neq 1$, the third computing unit **13109** computes $z_{\iota,3}^{1/r(\iota,3)}$ and sets the result of the computation as u_ι. The result u_ι of the computation is sent to the first power computing unit **11106**. Here, if $b(\iota)=1$, then $u_\iota=z_{\iota,3}^{1/r(\iota,3)}=f_\iota(\lambda_\iota)\lambda_{\iota,3}$. That is, $z_{\iota,3}^{1/r(\iota,3)}$ serves as a sampler having an error $X_{\iota,3}$ for $f_\iota(\lambda_\iota)$. The reason will be described later (step S**13105**).

<<Process at Step S**13108**>>

If $a(\iota)\neq 1$, the second computing unit **12108** performs the process at step S**12108** to generate v_ι; if $a(\iota)=1$, the third computing unit **13109** computes $z_{\iota,3}^{1/r(\iota,3)}$ and sets the result of the computation as v_ι. The result v_ι of the computation is sent to the second power computing unit **11109**. Here, if $a(\iota)=1$, then $v_\iota=z_{\iota,3}^{1/r(\iota,3)}=f_\iota(\lambda_\iota)x_{\iota,3}$. That is, $z_{\iota,3}^{1/r(\iota,3)}$ serves as a sampler having an error $X_{\iota,3}$ for $f_\iota(\lambda_\iota)$. The reason will be described later (step S**13108**).

Note that if $z_3^{1/r(\iota,3)}$, that is, the radical root of $z_{\iota,3}$, is hard to compute, u_ι and/or v_ι may be calculated as follows. The third computing unit **13109** may store each pair of the random number $r(\iota, 3)$ and $z_{\iota,3}$ computed on the basis of the random number $r(\iota, 3)$ in a storage, not depicted, in sequence as (α_1, β_2), (α_2,β_2), . . . , $(\alpha_{m(\iota)}, \beta_{m(\iota)})$, ... Here, $m(\iota)$ is a natural number greater than or equal to 1. The third computing unit **13109** may compute $\gamma_1, \gamma_2, \ldots, \gamma_{m(\iota)}$ that satisfies $\gamma_1\alpha_1+\gamma_2\alpha_2+\ldots+\gamma_{m(\iota)}\alpha_{m(\iota)}=1$ if the least common multiple of $\alpha_1, \alpha_2, \ldots, \alpha_{m(\iota)}$ is 1, where $\gamma_1, \gamma_2, \ldots, \gamma_{m(\iota)}$ are integers. The third computing unit **13109** then may use the resulting $\gamma_1, \gamma_2, \ldots, \gamma_{m(\iota)}$ to compute $\Pi_{i=1}^{m(\iota)}\beta_i^{\gamma i}=\beta_1^{\gamma 1}\beta_2^{\gamma 2}\ldots\beta_{m(\iota)}^{\gamma m(\iota)}$ and may set the results of the computation as u_ι and/or v_ι. Note that when a notation $\alpha^{\beta\gamma}$ is used herein in this way, $\beta\gamma$ represents β_γ, namely β with subscript γ, where α is a first letter, β is a second letter, and γ is a number.

<<Reason Why $z_{\gamma,3}^{1/r(\iota,3)}$ Serves as a Sampler Having an Error $X_{\iota,3}$ for $f_\iota(\lambda_\iota)$>>

Let R and R' be random numbers and $B(\lambda^R)$ be the result of computation performed by the capability providing apparatus **132**-ι by using λ^R. That is, let $z=B(x^R)$ be the results of computations returned by the first output information computing unit **12201**-ι, the second output information computing unit **12202**-ι, and the third output information computing unit **13203**-ι to the computing apparatus **131**. Furthermore, a random variable X having a value in the group G is defined as $X=B(\lambda^R)^{1/R}f(\lambda)^{-1}$.

Then $z^{1/R}=B(\lambda^R)^{1/R}=Xf(\lambda)=f(\lambda)X$. That is, $z^{1/R}$ serves as a sampler having an error X for $f(\lambda)$.

The expansion of formula given above uses the properties such that $X=B(\lambda^R)^{1/R}f(\lambda^R)^{-1}$ and that $B(\lambda^R)^{1/R}Xf(\lambda^R)$. The properties are based on the fact that R and R' are random numbers.

Therefore, considering that $r(\iota, 3)$ is a random number, $z_{\iota,3}^{1/r(\iota,3)}$ serves as a sampler having an error $X_{\iota,3}$ for $f_\iota(\lambda_\iota)$, likewise.

Tenth Embodiment

A proxy computing system of a tenth embodiment is another example that embodies the first and second randomizable samplers described above. Specifically, the proxy computing system embodies an example of the first and second randomizable samplers in the case where $H_\iota=G_\iota\times G_\iota$ and the decryption function f_ι is a decryption function of ElGamal encryption, that is, $f_\iota(c_{\iota,1}, c_{\iota,2})=c_{\iota,1}c_{\iota,2}^{-s\iota}$ for a decryption key s_ι and a ciphertext $\lambda_\iota=(c_{\iota,1}, c_{\iota,2})$. The following description will focus on differences from the seventh embodiment and repeated description of commonalities with the seventh embodiment, including the decryption control process, will be omitted.

As illustrated in FIG. **18**, the proxy computing system **104** of the tenth embodiment includes a computing apparatus **141** in place of the computing apparatus **111** and capability providing apparatuses **142-1**, . . . , **142-Γ** in place of the capability providing apparatuses **112-1**, . . . , **112-Γ**.

As illustrated in FIG. **19**, the computing apparatus **141** of the tenth embodiment includes, for example, a natural number storage **11101**, a natural number selecting unit **11102**, an integer computing unit **11103**, an input information providing unit **14104**, a first computing unit **14105**, a first power computing unit **11106**, a first list storage **11107**, a second computing unit **14108**, a second power computing unit **11109**, a second list storage **11110**, a determining unit **11111**, a final output unit **11112**, and a controller **11113**. As illustrated in FIG. **23**, the input information providing unit **14104** of this embodiment includes, for example, a fourth random number generator **14104***a*, a fifth random number generator **14104***b*, a first input information computing unit **14104***c*, a sixth random number generator **14104***d*, a seventh random number generator **14104***e*, and a second input infor-

mation computing unit **14104***f*. The first input information computing unit **14104***c* includes, for example, a fourth input information computing unit **14104***ca* and a fifth input information computing unit **14104***cb*. The second input information computing unit **14104***f* includes, for example, a sixth input information computing unit **14104***fa* and a seventh input information computing unit **14104***fb*.

As illustrated in FIG. **20**, the capability providing apparatus **142**-ι of the tenth embodiment includes, for example, a first output information computing unit **14201**-ι, a second output information computing unit **14202**-ι, a key storage **11204**-ι, and a controller **11205**-ι. If there are capability providing apparatuses **142**-(ι+1), . . . , **42**-Γ, the capability providing apparatuses **142**-(ι+1), . . . , **142**-Γ have the same configuration as the capability providing apparatus **142**-ι.

<Decryption Process>

A decryption process of this embodiment will be described below.

In the tenth embodiment, it is assumed that a group H, is the direct product group $G_\iota \times G_\iota$ of a group G_ι, the group G_ι is a cyclic group, a ciphertext $\lambda_\iota=(c_{\iota,1}, c_{\iota,2}) \in H_\iota$, $f_\iota(c_{\iota,1},c_{\iota,2})$ is a homomorphic function, a generator of the group G_ι is $\mu_{\iota,g}$, the order of the group G_ι is $K_{\iota,G}$, a pair of a ciphertext $(V_\iota, W_\iota) \in H_\iota$ and a decrypted text $f_\iota(V_\iota, W_\iota)=Y_\iota \in G_\iota$ decrypted from the ciphertext for the same decryption key s_ι is preset in the computing apparatus **141** and the capability providing apparatus **142**-ι, and the computing apparatus **141** and the capability providing apparatus **142**-ι can use the pair.

As illustrated in FIGS. **25** and **26**, the process of the tenth embodiment is the same as the process of the seventh embodiment except that steps S**11103** through S**11105**, S**11108**, and S**11200** through S**11203** of the seventh embodiment are replaced with steps S**14103** through S**14105**, S**14108**, and S**14200** through S**14203**, respectively. In the following, only processes at steps S**14103** through S**14105**, S**14108**, and S**14200** through S**14203** will be described.

<<Process at Step S**14103**>>

The input information providing unit **14104** of the computing apparatus **141** (FIG. **19**) generates and outputs first input information $\tau_{\iota,1}$ corresponding to an input ciphertext $\lambda_\iota=(c_{\iota,1}, c_{\iota,2})$ and second input information $\tau_{\iota,2}$ corresponding to the ciphertext $\lambda_\iota=(c_{\iota,1}, c_{\iota,2})$ (step S**14103** of FIG. **25**). A process at step S**14103** of this embodiment will be described below with reference to FIG. **28**.

The fourth random number generator **14104***a* (FIG. **23**) generates a uniform random number r(ι,4) that is a natural number greater than or equal to 0 and less than $K_{\iota,G}$. The generated random number r(ι, 4) is sent to the fourth input information computing unit **14104***ca*, the fifth input information computing unit **14104***cb*, and the first computing unit **14105** (step S**14103***a*). The fifth random number generator **14104***b* generates a uniform random number r(ι, 5) that is a natural number greater than or equal to 0 and less than $K_{\iota,G}$. The generated random number r(ι, 5) is sent to the fifth input information computing unit **14104***cb* and the first computing unit **14105** (step S**14103***b*).

The fourth input information computing unit **14104***ca* uses a natural number b(ι) selected by the natural number selecting unit **11102**, $c_{\iota,2}$ included in the ciphertext λ_ι, and the random number r(ι, 4) to compute fourth input information $c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)}$(step S**14103***c*). The fifth input information computing unit **14104***cb* uses the natural number b(ι) selected by the natural number selecting unit **11102**, $c_{\iota,1}$ included in the ciphertext λ_ι, and random numbers r(ι, 4) and r(ι, 5) to compute fifth input information $c_{\iota,1}{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)}$(step S**14103***d*).

The sixth random number generator **14104***d* generates a uniform random number r(ι, 6) that is a natural number greater than or equal to 0 and less than $K_{\iota,G}$. The generated random number r(ι, 6) is sent to the sixth input information computing unit **14104***fa*, the seventh input information computing unit **14104***fb*, and the second computing unit **14108** (step S**14103***e*). The seventh random number generator **14104***e* generates a uniform random number r(ι 7) that is a natural number greater than or equal to 0 and less than $K_{\iota,G}$. The generated random number r(ι, 7) is sent to the seventh input information computing unit **14104***fb* and the second computing unit **14108** (step S**14103***f*).

The sixth input information computing unit **14104***fa* uses a natural number a(ι) selected by the natural number selecting unit **11102**, $c_{\iota,2}$ included in the ciphertext λ_ι, and the random number r(ι, 6) to compute sixth input information $c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)}$(step S**14103***g*). The seventh input information computing unit **14104***fb* uses a natural number a(ι) selected by the natural number selecting unit **11102**, $c_{\iota,1}$ included in the ciphertext λ_ι, and the random numbers r(ι, 6) and r(ι, 7) to compute seventh input information $c_{\iota,1}{}^{a(\iota)}V_\iota{}^{(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}$ (step S**14103***h*).

The first input information computing unit **14104***c* outputs the fourth input information $c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)}$ and the fifth input information $c_{\iota,1}{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)}$ generated as described above as first input information $\tau_{\iota,1}=(c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)}, c_\iota{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)})$. The second input information computing unit **14104***f* outputs the sixth input information $c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)}$ and the seventh input information $c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}$ generated as described above as second input information $\tau_{\iota,2}=(c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)}, c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}$ (step S**14103***i*).

<<Processes at Steps S**14200** Through S**14203**>>

As illustrated in FIG. **26**, first, the first input information $\tau_{\iota,1}=(c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)}, c_{\iota,1}{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)})$ is input in the first output information computing unit **14201**-ι of the capability providing apparatus **142**-ι (FIG. **20**) and the second input information $\tau_{\iota,2}$ $(c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)}, c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_g{}^{r(\iota,7)})$ is input in the second output information computing unit **14202**-ι (step S**14200**).

The first output information computing unit **14201**-ι uses the first input information $\tau_{\iota,1}=((c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)}, c_{\iota,1}{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)})$ and the decryption key s_ι stored in the key storage **11204**-ι to correctly compute $f_\iota(c_{\iota,1}{}^{b(\iota)}V_\iota{}^{r(\iota,4)}\mu_{\iota,g}{}^{r(\iota,5)}, c_{\iota,2}{}^{b(\iota)}W_\iota{}^{r(\iota,4)})$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\iota,1}$. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **14201**-ι may or may not be $f_\iota(c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}, c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)})$ (step S**14201**).

The second output information computing unit **14202**-ι can correctly compute $f_\iota(c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}, c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)})$ with a probability greater than a certain probability by using the second input information $\tau_{\iota,2}=(c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)}, c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)})$ and the decryption key s_ι stored in the key storage **11204**-ι and provides the result of the computation as second input information $z_{\iota,2}$. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **14202**-ι may or may not be $f_\iota(c_{\iota,1}{}^{a(\iota)}V_\iota{}^{r(\iota,6)}\mu_{\iota,g}{}^{r(\iota,7)}, c_{\iota,2}{}^{a(\iota)}W_\iota{}^{r(\iota,6)})$ (step S**14202**).

The first output information computing unit **14201**-ι outputs the first output information $z_{\iota,1}$ and the second output information computing unit **14202**-ι outputs the second output information $z_{\iota,2}$ (step S**14203**).

<<Processes at Steps S14104 and S14105>>

Returning to FIG. **25**, the first output information $z_{\iota,1}$ is input in the first computing unit **14105** of the computing apparatus **141** (FIG. **19**) and the second output information $z_{\iota,2}$ is input in the second computing unit **14108** (step S**14104**).

The first computing unit **14105** uses the input first output information $z_{\iota,1}$ and the random numbers $r(\iota, 4)$ and $r(\iota, 5)$ to compute $z_{\iota,1}Y_{\iota}^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ and sets the result of the computation as u_{ι} (step S**14105**). The result u_{ι} of the computation is sent to the first power computing unit **11106**. Here, $u_{\iota}=z_{\iota,1}Y_{\iota}^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}=f_{\iota}(c_{\iota,1}, c_{\iota,2})^{b(\iota)}x_{\iota,1}$. That is, $z_{\iota,1}Y_{\iota}^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ is an output of a randomizable sampler having an error $X_{\iota,1}$ for $f_{\iota}(c_{\iota,1}, c_{\iota,2})$. The reason will be described later.

<<Process at Step S14108>>

The second computing unit **14108** uses the input second output information $z_{\iota,2}$ and the random numbers $r(\iota, 6)$ and $r(\iota, 7)$ to compute $z_{\iota,2}Y_{1}^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}$ and sets the result of the computation as v_{ι}. The result v_{ι} of the computation is sent to the second power computing unit **11109**. Here, $v_{\iota}=z_{\iota,2}Y_{\iota}^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}=f_{\iota}(c_{\iota,1}, c_{\iota,2})^{a(\iota)}x_{\iota,2}$. That is, $z_{\iota,2}Y_{\iota}^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}$ is an output of a randomizable sampler having an error $X_{\iota,2}$ for $f_{\iota}(c_{\iota,1}, c_{\iota,2})$. The reason will be described later.

<<Reason Why $z_{\iota,1}Y^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ and $z_{\iota,2}Y_{\iota}^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}$ are Outputs of Randomizable Samplers Having Errors $X_{\iota,1}$ and $X_{\iota,2}$, Respectively, for $(c_{\iota,1}, c_{\iota,2})$>>

Let c be a natural number, R_1, R_2, $R_{1'}$ and $R_{2'}$ be random numbers, and $B(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})$ be the result of computation performed by the capability providing apparatus **142**-ι by using $c_1{}^{c}V^{R1}\mu_g^{R2}$ and $c_2{}^{c}W^{R1}$. That is, the first output information computing unit **14201**-ι and the second output information computing unit **14202**-ι return $z=B(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})$ as the results of computations to the computing apparatus **141**. A random variable X having a value in a group G is defined as $X=B(V^{R1'}\mu_g^{R2'}, W^{R1'})f(V^{R1'}\mu_g^{R2'}, W^{R1'})^{-1}$.

Then, $zY^{-R1}\mu_g^{-R2}=B(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})Y^{-R1}\mu_g^{-R2}=Xf(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})Y^{-R1}\mu_g^{-R2}=Xf(c_1,c_2)^{c}f(V,W)^{R1}f(\mu_g,e_g)^{R2}Y^{-R1}\mu_g^{-R2}=Xf(c_1, c_2)^{c}Y_{R1}\mu_g^{R2}Y^{-R1}\mu_g^{-R2}=f(c_1,c_2)^{c}X$. That is, $zY^{-R1}\mu_g^{-R2}$ is an output of a randomizable sampler having an error X for f(x). Note that e_g is an identity element of the group G.

The expansion of formula given above uses the properties such that $X=B(V^{R1'}\mu_g^{R2'}, W^{R1'})f(V^{R1'}\mu_g^{R2'}, W^{R1'})^{-1}=B(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{2}W^{R1})f(c_c{}^{c}V^{R1}{}_g^{R2}, c_2{}^{c}W^{R1})$ and that $B(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})=Xf(c_1{}^{c}V^{R1}\mu_g^{R2}, c_2{}^{c}W^{R1})$. The properties are based on the fact that R_1, R_2, $R_{1'}$ and $R_{2'}$ are random numbers.

Therefore, considering that $a(\iota)$ and $b(\iota)$ are natural numbers and $r(\iota, 4)$, $r(\iota, 5)$, $r(\iota, 6)$ and $r(\iota, 7)$ are random numbers, $z_{\iota,1}Y_{\iota}^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ and $z_{\iota,2}Y_{\iota}^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}$ are outputs of randomizable samplers having errors $X_{\iota,1}$ and $X_{\iota,2}$, respectively, for $f_{\iota}(c_{\iota,1}, c_{\iota,2})$.

Eleventh Embodiment

In the seventh to tenth embodiments described above, a plurality of pairs $(a(\iota), b(\iota))$ of natural numbers $a(\iota)$ and $b(\iota)$ that are relatively prime to each other are stored in the natural number storage **11101** of the computing apparatus and these pairs $(a(\iota), b(\iota))$ are used to perform processes. However, one of $a(\iota)$ and $b(\iota)$ may be a constant. For example, $a(\iota)$ may be fixed at 1 or $b(\iota)$ may be fixed at 1. Which of natural number $a(\iota)$ and $b(\iota)$ is a constant may be varied depending on t. In other words, one of the first randomizable sampler and the second randomizable sampler may be replaced with a sampler. If one of $a(\iota)$ and $b(\iota)$ is a constant, the need for the process for selecting the constant $a(\iota)$ or $b(\iota)$ is eliminated, the constant $a(\iota)$ or $b(\iota)$ is not input in the processing units and the processing units can treat it as a constant to perform computations. If $a(\iota)$ or $b(\iota)$ set as a constant is equal to 1, $f_{\iota}(\lambda_{\iota})=u_{\iota}^{b'(\iota)}v_{\iota}^{a'(\iota)}$ can be obtained as $f_{\iota}(\lambda_{\iota})=v_{\iota}$ or $f_{\iota}(\lambda_{\iota})=u_{\iota}$ without using $a'(\iota)$ or $b'(\iota)$.

An eleventh embodiment is an example of such a variation, in which $b(\iota)$ is fixed at 1 and the second randomizable sampler is replaced with a sampler. The following description will focus on differences from the seventh embodiment and description of commonalities with the seventh embodiment, including the decryption control process, will be omitted. Specific examples of the first randomizable sampler and the sampler are similar to those described in the eighth to tenth embodiments and therefore description of the first randomizable sampler and the sampler will be omitted.

<Configuration>

As illustrated in FIG. **18**, a proxy computing system **105** of the eleventh embodiment includes a computing apparatus **151** in place of the computing apparatus **111** of the seventh embodiment and capability providing apparatuses **152-1**, . . . , **152**-Γ in place of the capability providing apparatuses **112-1**, . . . , **112**-Γ.

As illustrated in FIG. **29**, the computing apparatus **151** of the eleventh embodiment includes, for example, a natural number storage **15101**, a natural number selecting unit **15102**, an input information providing unit **15104**, a first computing unit **15105**, a first power computing unit **11106**, a first list storage **11107**, a second computing unit **15108**, a second list storage **15110**, a determining unit **15111**, a final output unit **15112**, and a controller **11113**.

As illustrated in FIG. **20**, the capability providing apparatus **152**-ι of the eleventh embodiment includes, for example, a first output information computing unit **15201**-ι, a second output information computing unit **15202**-ι, a key storage **11204**-ι, and a controller **11205**-ι. If there are capability providing apparatuses **152**-(ι+1), . . . , **52**-Γ, the capability providing apparatuses **152**-(ι+1), . . . , **52**-Γ have the same configuration as the capability providing apparatus **152**-ι.

<Decryption Process>

A decryption process of this embodiment will be described below. For the decryption process, let G_{ι} and H_{ι} be groups (for example commutative groups), $f_{\iota}(\lambda_{\iota})$ be a decryption function for decrypting a ciphertext λ_{ι}, which is an element of the group H_{ι}, with a particular decryption key s_{ι} to obtain an element of the group G_{ι}, generators of the groups G_{ι} and H_{ι} be $\mu_{\iota,g}$ and $\mu_{\iota,h}$, respectively, $X_{\iota,1}$ and $X_{\iota,2}$ be random variables having values in the group G_{ι}, $x_{\iota,1}$ be a realization of the random variable $X_{\iota,1}$, and $x_{\iota,2}$ be a realization of the random variable $X_{\iota,2}$. It is assumed here that a plurality of natural numbers $a(\iota)$ are stored in the natural number storage **15101** of the computing apparatus **151**.

As illustrated in FIG. **30**, first, the natural number selecting unit **15102** of the computing apparatus **151** (FIG. **29**) randomly reads one natural number $a(\iota)$ from among the plurality of natural numbers $a(\iota)$ stored in the natural number storage **15101**. Information on the read natural number $a(\iota)$ is sent to the input information providing unit **15104** and the first power computing unit **11106** (step S**15100**).

The controller **11113** sets $t_{\iota}=1$ (step S**11102**).

The input information providing unit **15104** generates and outputs first input information $t_{\iota,1}$ and second input information $t_{\iota,2}$ each of which corresponds to the input ciphertext λ_{ι}. Preferably, the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ are information whose relation with

the ciphertext λ_ι is scrambled. This enables the computing apparatus **151** to conceal the ciphertext λ_ι from the capability providing apparatus **152**-ι. Preferably, the second input information $t_{\iota,2}$ of this embodiment further corresponds to the natural number a(ι) selected by the natural number selecting unit **15102**. This enables the computing apparatus **151** to evaluate the decryption capability provided by the capability providing apparatus **152**-ι with a high degree of accuracy (step **S15103**). A specific example of the pair of the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ is a pair of first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ of any of the eighth to tenth embodiments when b(ι)=1.

As illustrated in FIG. **26**, the first input information $\tau_{\iota,1}$ is input in the first output information computing unit **15201**-ι of the capability providing apparatus **152**-ι (FIG. **20**) and the second input information $\tau_{\iota,2}$ is input in the second output information computing unit **15202**-ι (step **S15200**).

The first output information computing unit **15201**-ι uses the first input information $\tau_{\iota,1}$ and the decryption key s_ι stored in the key storage **11204**-ι to correctly compute $f_\iota(\tau_{\iota,1})$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\iota,1}$ (step **S15201**). The second output information computing unit **15202**-ι uses the second input information $\tau_{\iota,2}$ and the decryption key s_ι stored in the key storage **11204**-ι to correctly compute $f_\iota(\tau_{\iota,2})$ with a probability greater than a certain probability and sets the result of the computation as second output information $z_{\iota,2}$ (step **S15202**). That is, the first output information computing unit **15201**-ι and the second output information computing unit **15202**-ι output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **15201**-ι may or may not be $f_\iota(\tau_{\iota,1})$ and the result of the computation by the second output information computing unit **15202**-ι may or may not be $f_\iota(T_{\iota,2})$. A specific example of the pair of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is a pair of first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ of any of the eighth to tenth embodiments when b(ι)=1.

The first output information computing unit **15201**-ι outputs the first output information $z_{\iota,1}$ and the second output information computing unit **15202**-ι outputs the second output information $z_{\iota,2}$ (step **S15203**).

Returning to FIG. **30**, the first output information $z_{\iota,1}$ is input in the first computing unit **15105** of the computing apparatus **151** (FIG. **29**) and the second output information $z_{\iota,2}$ is input in the second computing unit **15108**. The first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ are equivalent to the decryption capability provided by the capability providing apparatus **152**-ι to the computing apparatus **151** (step **S15104**).

The first computing unit **15105** generates a computation result $u_\iota=f_\iota(\lambda_\iota)x_{\iota,1}$ from the first output information $z_{\iota,1}$. A specific example of the computation result u_ι is a result u_ι of computation of any of the eighth to tenth embodiments when b(ι)=1. The computation result u, is sent to the first power computing unit **11106** (step **S15105**).

The first power computing unit **11106** computes $u_\iota'=u_\iota^{a(\iota)}$. The pair of the result u_ι of the computation and u_ι' computed on the basis of the result of the computation, (u_ι, u_ι'), is stored in the first list storage **11107** (step **S11106**).

The second computing unit **15108** generates a computation result $v_\iota=f_\iota(\lambda_\iota)^{a(\iota)}x_{\iota,2}$ from the second output information $z_{\iota,2}$. A specific example of the result v_ι of the computation is a result v_ι of the computation in any of the eighth to tenth embodiments. The result v_ι of the computation is stored in the second list storage **15110** (step **S15108**).

The determining unit **15111** determines whether or not there is one that satisfies $u_\iota'=v_\iota$ among the pairs (u_ι, u_ι') stored in the first list storage **11107** and v_ι stored in the second list storage **15110** (step **S15110**). If there is one that satisfies $u_\iota'=v_\iota$, the process proceeds to step **S15114**; if there is not one that satisfies $u_\iota'=v_\iota$, the process proceeds to step **S11111**.

At step **S11111**, the controller **11113** determines whether or not $t_\iota=T_\iota$ (step **S11111**). Here, T_ι is a predetermined natural number. If $t_\iota=T_\iota$, the final output unit **15112** outputs information indicating that the computation is impossible, for example, the symbol "⊥" (step **S11113**), then the process ends. If not $t_\iota=T_\iota$, the controller **11113** increments t_ι by 1, that is, sets $t_\iota=t_\iota+1$ (step **S11112**), then the process returns to step **S15103**.

At step **S15114**, the final output unit **15112** outputs u_ι corresponding to u_ι' determined to satisfy $u_\iota'=v_\iota$ (step **S15114**). The obtained u, is equivalent to $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ in the seventh to tenth embodiments when b(ι)=1. That is, u_ι thus obtained can be the decryption result $f_\iota(\lambda_\iota)$ resulting from decrypting the ciphertext λ_ι with the particular decryption key s_ι with a high probability. Therefore, the process described above is repeated multiple times and the value that has most frequently obtained among the values obtained at step **S15114** can be chosen as the decryption result $f_\iota(\lambda_\iota)$. As will be described later, $u_\iota=f_\iota(\lambda_\iota)$ can result with an overwhelming probability, depending on settings. In that case, the value obtained at step **S15114** can be directly provided as a result of decryption $f_\iota(\lambda_\iota)$. The rest of the process is as described in the seventh embodiment.

<<Reason why Decryption Result $f_\iota(\lambda_\iota)$ can be Obtained>>

The reason why a decryption result $f_\iota(\lambda_\iota)$ can be obtained on the computing apparatus **151** of this embodiment will be described below. For simplicity of notation, ι will be omitted in the following description. Terms required for the description will be defined first.

Black-box:

A black-box F(τ) of f(τ) is a processing unit that takes an input of τ∈H and outputs z∈G. In this embodiment, each of the first output information computing unit **15201** and the second output information computing unit **15202** is equivalent to the black box F(τ) for the decryption function f(τ). A black-box F(τ) that satisfies z=f(τ) for an element $\tau\in_U H$ arbitrarily selected from a group H and z=F(τ) with a probability greater than δ (0<δ≤1), that is, a black-box F(τ) for f(τ) that satisfies

$$Pr[z=f(\tau)|\tau\in_U H, z=F(\tau)]>\delta \tag{8}$$

is called a δ-reliable black-box F(τ) for f(τ). Here, δ is a positive value and is equivalent to the "certain probability" stated above.

Self-corrector

A self-corrector $C^F(\lambda)$ is a processing unit that takes an input of λ∈H, performs computation by using a black-box F(τ) for f(τ), and outputs j∈G∪⊥. In this embodiment, the computing apparatus **151** is equivalent to the self-corrector $C^F(\lambda)$.

Almost self-corrector:

Assume that a self-corrector $C^F(\lambda)$ that takes an input of λ∈H and uses a δ-reliable black-box F(τ) for f(τ) to perform computation outputs a correct value j=f(λ) with a probability sufficiently greater than the provability with which the self-corrector CF(λ) outputs an incorrect value j≠f(λ). That is, assume that a self-corrector $C^F(\lambda)$ satisfies

$$Pr[j=f(\lambda)|j=C^F(\lambda), j\neq\perp]>Pr[j\neq f(\lambda)|j=C^F(\lambda), j\neq\perp]+\Delta \tag{9}$$

Here, Δ is a certain positive value (0<Δ<1). If this is the case, the self-corrector $C^F(\lambda)$ is called an almost self-corrector. For example, for a certain positive value Δ' (0<Δ'<1), if a self-corrector $C^F(\lambda)$ satisfies

$$Pr[j=f(\lambda)|j=C^F(\lambda)]>(1/3)+\Delta'$$

$$Pr[j=\perp|j=C^F(\lambda)]<1/3$$

$$Pr[j\neq f(\lambda) \text{ and } j\neq\perp|j=C^F(\lambda)]<1/3,$$

then the self-corrector $C^F(\lambda)$ is an almost self-corrector. Examples of Δ' include Δ'=1/12 and Δ'=1/3.

Robust self-corrector:

Assume that a self-corrector $C^F(\lambda)$ that takes an input of $\lambda\in H$ and uses a δ-reliable black-box $F(\tau)$ for $f(\tau)$ outputs a correct value $j=f(\lambda)$ or $j=\perp$ with an overwhelming probability. That is, assume that for a negligible error ξ ($0\leq\xi<1$), a self-corrector $C^F(\lambda)$ satisfies

$$Pr[j=f(\lambda) \text{ or } j=\perp|j=C^F(\lambda)]>1-\xi \quad (10)$$

If this is the case, the self-corrector $C^F(\lambda)$ is called a robust self-corrector. An example of the negligible error ξ is a function vale ξ(k) of a security parameter k. An example of the function value ξ(k) is a function value ξ(k) such that {ξ(k)p(k)} converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function value ξ(k) include $\xi(k)=2^{-k}$ and $\xi(k)=2^{-\sqrt{k}}$.

A robust self-corrector can be constructed from an almost self-corrector. Specifically, a robust self-corrector can be constructed by executing an almost self-corrector multiple times for the same λ and selecting the most frequently output value, except ⊥, as j. For example, an almost self-corrector is executed O(log(1/ξ)) times for the same λ and the value most frequently output is selected as j, thereby a robust self-corrector can be constructed. Here, O(·) represents O notation.

Pseudo-free action:

An upper bound of the probability

$$Pr[\alpha^a=\beta \text{ and } \alpha\neq e_g|a\in_U\Omega,\alpha\in X_1,\beta\in X_2] \quad (11)$$

of satisfying $\alpha^a=\beta$ for all possible X_1 and X_2 is called a pseudo-free indicator of a pair (G, Ω) and is represented as P(G, Ω), where G is a group, Ω is a set of natural numbers $\Omega=\{0, \ldots, M\}$ (M is a natural number greater than or equal to 1), α and β are realizations $\alpha\in X_1$ ($\alpha\neq e_g$) and $\beta\in X_2$ of random variables X_1 and X_2 that have values in the group G, and $a\in\Omega$. If a certain negligible function ζ(k) exists and

$$P(G,\Omega)<\zeta(k) \quad (12),$$

then a computation defined by the pair (G, Ω) is called a pseudo-free action. Note that "α" means that a computation defined at the group G is applied a times to α. An example of the negligible function ζ(k) is such that {ζ(k)p(k)} converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function ζ(k) include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. For example, if the probability of Formula (11) is less than $O(2^{-k})$ for a security parameter k, a computation defined by the pair (G, Ω) is a pseudo-free action. For example, if the number of the elements $|\Omega\cdot\alpha|$ of a set $\Omega\cdot\alpha=\{a(\alpha)|a\in\Omega\}$ exceeds 2^k for any $^\forall\alpha\in G$ where $\alpha\neq e_g$, a computation defined by the pair (G, Ω) is a pseudo-free action. Note that $a(\alpha)$ represents the result of a given computation on a and α. There are many such examples. For example, if the group G is a residue group Z/pZ modulo prime p, the prime p is the order of 2^k, the set $\Omega=\{0, \ldots, p-2\}$, $a(\alpha)$ is $\alpha^a\in Z/pZ$, and $\alpha\neq e_g$, then $\Omega\cdot\alpha=\{\alpha^a|a=0, \ldots, p-2\}=\{e_g, \alpha^1, \ldots, \alpha^{P-2}\}$ and $|\Omega\cdot\alpha|=p-1$. If a certain constant C exists and k is sufficiently large, $|\Omega\cdot\alpha|>C2^k$ is satisfied because the prime p is the order of 2^k. Here, the probability of Formula (11) is less than $C^{-1}2^{-k}$ and a computation defined by such pair (G, Ω) is a pseudo-free action.

δ^γ-reliable randomizable sampler:

A randomizable sampler that whenever a natural number a is given, uses the δ-reliable black-box $F(\tau)$ for $f(\tau)$ and returns $w^a x'$ corresponding to a sample x' that depends on a random variable X for $w\in G$ and in which the probability that $w^a x'=w^a$ is greater than δ^γ (γ is a positive constant), that is,

$$Pr[w^a x'=w^a]>\delta\gamma \quad (13)$$

is satisfied, is called a δ^γ-reliable randomizable sampler. The combination of the input information providing unit **15104**, the second output information computing unit **15202**, and the second computing unit **15108** of this embodiment is a δ^γ-reliable randomizable sampler for $w=f(\lambda)$.

The definitions given above will be used to describe the reason why a decryption result $f(\lambda)$ can be obtained by using the computing apparatus **151** of this embodiment.

At step **S15110** of this embodiment, determination is made as to whether u'=v, that is, whether $u^a=v$. Since the combination of the input information providing unit **15104**, the second output information computing unit **15202**, and the second computing unit **15108** of this embodiment is a δ^γ-reliable randomizable sampler (Formula (13)), $u^a=v$ holds (Yes at step **S15110**) with an asymptotically large probability if T is greater than a certain value determined by k, δ and γ. For example, Markov's inequality shows that if $T\geq 4/\delta^\gamma$, the probability that $u^a=v$ holds (Yes at step **S15110**) is greater than ½.

Since $u=f(\lambda)x_1$ and $v=f(\lambda)^a x_2$ in this embodiment, $x_1{}^a=x_2$ holds if $u^a=v$ holds. $x_1{}^a=x_2$ holds if $x_1=x_2=e_g$ or $x\neq e_g$. If $x_1=x_2=e_g$, then $u=f(\lambda)$ and therefore u output at step **S15114** is a correct decryption result $f(\lambda)$. On the other hand, if $x_1\neq e_g$, then $u\neq f(\lambda)$ and therefore u output at step **S15114** is not a correct decryption result $f(\lambda)$.

If an computation defined by a pair (Q, Ω) of a group G and a set Ω to which a natural number a belongs is a pseudo-free action or $T^2P(G, \Omega)$ is asymptotically small for a pseudo-free index P(G, Ω), the probability that $x_1\neq e_g$ (Formula (11)) when $u^a=v$ is asymptotically small. Accordingly, the probability that $x_1=e_g$ when $u^a=v$ is asymptotically large. Therefore, if a computation defined by a pair (G, Ω) is a pseudo-free action or $T^2P(G, \Omega)$ is asymptotically small, the probability that an incorrect decryption result $f(\lambda)$ is output when $u^a=v$ is sufficiently smaller than the probability that a correct decryption result $f(\lambda)$ is output when $u^a=_v$. In this case, it can be said that the computing apparatus **151** is an almost self-corrector (see Formula (9)). Therefore, a robust self-corrector can be constructed from the computing apparatus **151** as described above and a correct decryption result $f(\lambda)$ can be obtained with an overwhelming probability. If a computation defined by (G, Ω) is a pseudo-free action, the probability that an incorrect decryption result $f(\lambda)$ is output when $u^a=v$ is also negligible. In that case, the computing apparatus **151** outputs a correct decryption result $f(\lambda)$ or ⊥ with an overwhelming probability.

Note that "η(k') is asymptotically small" means that k_0 is determined for an arbitrary constant ρ and the function value η(k') for any k' that satisfies $k_0<k'$ for k_0 is less than ρ. An example of k' is a security parameter k.

"η(k') is asymptotically large" means that k_0 is determined for an arbitrary constant ρ and the function value 1−η(k') for any k' that satisfies $k_0<k'$ for k_0 is less than ρ.

Note that the proof given above also proves that "if u'=v' holds, it is highly probable that the first randomizable

sampler has correctly computed $u=f(\lambda)^b$ and the second randomizable sampler has correctly computed $v=f(\lambda)^a$ (x_1 and x_2 are identity elements e_g of the group G)" stated in the seventh embodiment, as can be seen by replacing a with a/b.

<<δ^γ-Reliable Randomizable Sampler and Security>>

Consider the following attack.

A black-box $F(\tau)$ or part of the black-box $F(\tau)$ intentionally outputs an invalid z or a value output from the black-box $F(\tau)$ is changed to an invalid z.

$w^a x'$ corresponding to the invalid z is output from the randomizable sampler.

$w^a x'$ corresponding to the invalid z increases the probability with which the self-corrector $C^F(\lambda)$ outputs an incorrect value even though $u^a=v$ holds (Yes at step **S15110**) in the self-corrector $C^F(\lambda)$.

Such an attack is possible if the probability distribution $D_a=w^a x' w^{-a}$ of an error of $w^a x'$ output from the randomizable sampler for a given natural number a depends on the natural number a. For example, if tampering is made so that v output from the second computing unit **15108** is $f(\lambda)^a x_1{}^a$, always $u^a=v$ holds regardless of the value of x_1. Therefore, it is desirable that the probability distribution $D_a=w^a x' w^{-a}$ of an error of $w^a x'$ output from the randomizable sampler for a given natural number a do not depend on the natural number a.

Alternatively, it is desirable that the randomizable sampler be such that a probability distribution D that has a value in a group G that cannot be distinguished from the probability distribution $D_a=w^a x' w^{-a}$ of an error of $w^a x'$ for any element $a \in^\forall \Omega$ of a set Ω exists (the probability distribution D_a and the probability distribution D are statistically close to each other). Note that the probability distribution D does not depend on a natural number a. That the probability distribution D_a and the probability distribution D cannot be distinguished from each other means that the probability distribution D_a and the probability distribution D cannot be distinguished from each other by a polynomial time algorithm. For example, if

$$\Sigma_{g\in G}|Pr[g\epsilon D]-Pr[g\epsilon D_a]|<\zeta \quad (14)$$

is satisfied for negligible $\zeta(0\leq\zeta 1)$, the probability distribution D_a and the probability distribution D cannot be distinguished from each other by the polynomial time algorithm. An example of negligible ζ is a function value $\zeta(k)$ of the security parameter k. An example of the function value $\zeta(k)$ is a function value such that $\{\zeta(k)p(k)\}$ converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function $\zeta(k)$ include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. These facts also apply to the seventh to tenth embodiments which use natural numbers a and b.

[Variations of Seventh to Eleventh Embodiment]

In the seventh to eleventh embodiments, when the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ are provided to the computing apparatus, the computing apparatus can output $u_\iota{}^{b'(\iota)} v_\iota{}^{a'(\iota)}$ with a certain probability. $u_\iota{}^{b'(\iota)} v_\iota{}^{a'(\iota)}$ will be the decryption value of the ciphertext λ_ι. On the other hand, if neither of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ is provided to the computing apparatus, the computing apparatus cannot obtain the decryption value of the ciphertext λ_ι.

The capability providing apparatus in any of the seventh to eleventh embodiment can control whether or not to output both of the first output information $z_{\iota,1}$ and second output information $z_{\iota,2}$ to control the ciphertext decryption capability of the computing apparatus without providing a decryption key to the computing apparatus.

The present invention is not limited to the embodiments described above. For example, while ω is an integer greater than or equal to 2 in the embodiments described above, ω may be 1. That is, a configuration in which only one capability providing apparatus exists may be used. In that case, the computing apparatus does not need to include a recovering unit and may directly output a value output from the final output unit. For example, the system of any of the first to fifth embodiments described above may further include a decryption control apparatus, the decryption control apparatus may output a decryption control instruction for controlling the decryption process of the computing apparatus to the capability providing apparatus, and the capability providing apparatus may control whether or not to output both of first output information z_1 and second output information z_2 from the first output information computing unit and the second output information computing unit according to the decryption control instruction. For example, the system of the sixth embodiment described above may further include a decryption control apparatus, the decryption control apparatus may output a decryption control instruction for controlling the decryption process of the computing apparatus to the capability providing apparatus, and the capability providing apparatus may control whether or not to output first output information ${}_M z_1$ and second output information ${}_M z_2$ from the first output information computing unit and the second output information computing unit according to the decryption control instruction.

Random variables $X_{\iota,1}$, $X_{\iota,2}$ and $X_{\iota,3}$ may or may not be the same.

Each of the first random number generator, the second random number generator, the third random number generator, the fourth random number generator, the fifth random number generator, the sixth random number generator and the seventh random number generator generates uniform random numbers to increase the security of the proxy computing system. However, if the level of security required is not so high, at least some of the first random number generator, second random number generator, the third random number generator, the fourth random number generator, the fifth random number generator, the sixth random number generator and the seventh random number generator may generate random numbers that are not uniform random numbers. While it is desirable from the computational efficiency point of view that random numbers that are natural numbers greater than or equal to 0 and less than $K_{\iota,H}$ or random numbers that are natural numbers greater than or equal to 0 and less than $K_{\iota,G}$ be selected as in the embodiments described above, random numbers that are natural numbers greater than or equal to $K_{\iota,H}$ or $K_{\iota,G}$ may be selected instead.

The process of the capability providing apparatus may be performed multiple times each time the computing apparatus provides first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ corresponding to the same $a(\iota)$ and $b(\iota)$ to the capability providing apparatus. This enables the computing apparatus to obtain a plurality of pieces of first output information $z_{\iota,1}$, second output information $z_{\iota,2}$, and third output information $z_{\iota,3}$ each time the computing apparatus provides first input information $\tau_{\iota,1}$, and the second input information $\tau_{\iota,2}$ to the capability providing apparatus. Consequently, the number of exchanges and the amount of communication between the computing apparatus and the capability providing apparatus can be reduced.

The computing apparatus may provide a plurality of pieces of the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$ to the capability providing apparatus at once

and may obtain a plurality of pieces of corresponding first output information $z_{\iota,1}$, second output information $z_{\iota,2}$ and third output information $z_{\iota,3}$ at once. This can reduce the number of exchanges between the computing apparatus and the capability providing apparatus.

While ω in the embodiments is a constant, ω may be a variable, provided that a value of ω can be shared in the proxy computing system.

The units of the computing apparatus may exchange data directly or through a memory, which is not depicted. Similarly, the units of the capability providing apparatus may exchange data directly or through a memory, which is not depicted.

Check may be made to see whether u, and v, obtained at the first computing unit and the second computing unit of any of the embodiments are elements of the group G_ι. If they are elements of the group G_ι, the process described above may be continued; if u_ι or v_ι is not an element of the group G_ι, information indicating that the computation is impossible, for example the symbol "⊥", may be output.

Furthermore, the processes described above may be performed not only in time sequence as is written or may be performed in parallel with one another or individually, depending on the throughput of the apparatuses that perform the processes or requirements. A plurality of capability providing apparatus may be configured in a single apparatus. It would be understood that other modifications can be made without departing from the spirit of the present invention.

Twelfth Embodiment

A twelfth embodiment of the present invention will be described. In this embodiment, Φ computing apparatuses (Φ is an integer greater than or equal to 2) shares one capability providing apparatus to perform computations and the capability providing apparatus receives payment for the capability. However, this does not limit the present invention and Φ computing apparatuses may share a plurality of capability providing apparatuses to perform computations.

<Configuration>

As illustrated in FIG. **31**, a proxy computing system **201** of the twelfth embodiment includes, for example, Φ computing apparatuses **211**-ϕ (ϕ=1, . . . , Φ) and one capability providing apparatus **212**. The apparatuses are configured to be able to exchange data between them. For example, the apparatuses can exchange information through a transmission line, a network, a portable recording medium and other media.

In this embodiment, the capability providing apparatus **212** provides the capability of computing a function f_ϕ that maps elements of a group H_ϕ to elements of a group G_ϕ (the computing capability) to each computing apparatus **211**-ϕ. Each computing apparatus **211**-ϕ pays the capability providing apparatus **212** for the capability. The computing apparatus **211**-ϕ uses the provided capability to compute an element $f_\phi(x_\phi)$ of the group G_ϕ that corresponds to an element x_ϕ of the group H_ϕ.

As illustrated in FIG. **32**, the computing apparatus **211**-ϕ of the twelfth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **21104**-ϕ, a first computing unit **21105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **21108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **21111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ. The computing apparatus **211**-ϕ performs processes under the control of the controller **21113**-ϕ. Examples of the computing apparatuses **211**-ϕ include devices having computing and memory functions, such as well-known or specialized computers, that include a CPU (central processing unit) and a RAM (random-access memory) in which a special program is loaded, server devices, a gateway devices, card reader-writer apparatuses and mobile phones.

As illustrated in FIG. **33**, the capability providing apparatus **212** of the twelfth embodiment includes, for example, a first output information computing unit **21201**, a second output information computing unit **21202**, and a controller **21205**. The capability providing apparatus **212** performs processes under the control of the controller **21205**. Examples of the capability providing apparatus **212** include a well-known or specialized computer, a device including computing and memory functions, such as a mobile phone, and a tamper-resistant module such as an IC card and an IC chip, that include a CPU and a RAM in which a special program is loaded.

<Assumptions for Processes>

Let G_ϕ, H_ϕ be groups (for example commutative groups), $X_{\phi,1}$ and $X_{\phi,2}$ be random variables having values in the group G_ϕ, $x_{\phi,1}$ be a realization of the random variable $X_{\phi,1}$, $x_{\phi,2}$ be a realization of the random variable $X_{\phi,2}$, f_ϕ be a function that maps an element of the group H_ϕ to an element of the group G_ϕ, and a(ϕ) and b(ϕ) be natural numbers that are relatively prime to each other. Specific examples of f_ϕ include an encryption function, a decryption function, a re-encryption function, an image processing function, and a speech processing function. G_ϕ may be equal to H_ϕ or G_ϕ may be unequal to H_ϕ. All of the groups G_ϕ (ϕ=1, . . . , Φ) may be the same or at least some groups $G_{\phi'}$ may differ from the other groups G_ϕ, where ϕ'≠ϕ. All of the groups H_ϕ (ϕ=1, . . . , Φ) may be the same or at some groups $H_{\phi'}$ may differ from the other groups H_ϕ, where ϕ'≠ϕ. In the following description, computations on the groups G_ϕ and H_ϕ are multiplicatively expressed. a(ϕ) and b(ϕ) are natural numbers that are relatively prime to each other. The term "natural number" refers to an integer greater than or equal to 0. A set whose members are elements $f_\phi(M_\phi)^{a(\phi)b(\phi)}$ of the group G_ϕ that correspond to elements M_ϕ of the group H_ϕ is referred to as a "class $CL_\phi(M_\phi)$ corresponding to elements M_ϕ". Here, $CL_\phi(M_\phi)$ and $CL_{\phi'}(M_{\phi'})(\phi'\neq\phi)$ are classes different from each other. If a function $f_\phi(M_\phi)$ is an injective function for an element M_ϕ, only one member belongs to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ for a pair of a(ϕ) and b(ϕ). If only one member belongs to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ for each pair of a(ϕ) and b(ϕ), that two values belong to the class $CL_\phi(M_\phi)$ corresponding to the same M_ϕ is equivalent to that the two values are equal to each other. That is, if the function $f_\phi(M_\phi)$ is a injective function for the element M_ϕ, determination as to whether two values belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ or not can be made by determining whether the two values are equal to each other or not. On the other hand, if the function $f_\phi(M_\phi)$ is not an injective function for the element M_ϕ(for example if the function f_ϕ is an encryption function of probabilistic encryption such as ElGamal encryption), a plurality of members belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ for the pair of a(ϕ) and b(ϕ) because a plurality of ciphertexts correspond to a pair of a plaintext and an encryption key.

It is assumed here that a plurality of pairs (a(ϕ), b(ϕ)) of natural numbers a(ϕ) and b(ϕ) are stored in the natural number storage **21101**-ϕ of each computing apparatus **211**-ϕ

(FIG. **32**). Let I_ϕ be a set of pairs of relatively prime natural numbers that are less than the order of the group G_ϕ. Then it can be considered that pairs $(a(\phi), b(\phi))$ of natural numbers $a(\phi)$ and $b(\phi)$ corresponding to a subset S_ϕ of I_ϕ are stored in the natural number storage **21101**-ϕ.

<Processes>

Processes performed by the computing apparatus **211**-ϕ by using the capability providing apparatus **212** will be described. The processes may be performed by any one of the computing apparatuses **211**-ϕ occupying the capability providing apparatus **212** for a certain period of time or may be performed concurrently by a plurality of computing apparatuses **211**-ϕ accessing the capability providing apparatus **212**.

An element x_ϕ of the group H_ϕ is input in the input information providing unit **21104**-ϕ of a computing apparatus **211**-ϕ (FIG. **32**). If the element x_ϕ has been already input in the input information providing unit **21104**-ϕ, the input operation may be omitted.

As illustrated in FIG. **37**, the natural number selecting unit **21102**-ϕ of the computing apparatus **211**-ϕ in which the element x_ϕ is input in the input information providing unit **21104**-ϕ randomly reads one pair of natural numbers $(a(\phi), b(\phi))$ from among the plurality of pairs of natural numbers $(a(\phi), b(\phi))$ stored in the natural number storage **21101**-ϕ. At least some of information on the read pair of natural numbers $(a(\phi), b(\phi))$ is sent to the integer computing unit **21103**-ϕ, the input information providing unit **21104**-ϕ, the first power computing unit **21106**-ϕ and the second power computing unit **21109**-ϕ (step S**21100**).

The integer computing unit **21103**-ϕ uses the sent pair of natural numbers $(a(\phi), b(\phi))$ to computer integers $a'(\phi)$ and $b'(\phi)$ that satisfy the relation $a'(\phi)a(\phi)+b'(\phi)b(\phi)=1$. Since the natural numbers $a(\phi)$ and (ϕ) are relatively prime to each other, the integers $a'(\phi)$ and $b'(\phi)$ that satisfy the relation $a'(\phi)a(\phi)+b'(\phi)b(\phi)=1$ definitely exist. Methods for computing such integers are well known. For example, a well-known algorithm such as the extended Euclidean algorithm may be used to compute the integers $a'(\phi)$ and $b'(\phi)$. Information on the pair of natural numbers $(a'(\phi), b'(\phi))$ is sent to the final output unit **21112**-ϕ (step S**21101**).

The controller **21113**-ϕ sets t=1 (step S**21102**).

The input information providing unit **21104**-ϕ generates and outputs first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ each of which corresponds to the input element x_ϕ. Preferably, the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ are information whose relation with the element x_ϕ is scrambled. This enables the computing apparatus **211**-ϕ to conceal the element x_ϕ from the capability providing apparatus **212**. Preferably, the first input information $\tau_{\phi,1}$ of this embodiment further corresponds to the natural number $b(\phi)$ selected by the natural number selecting unit **21102**-ϕ and the second input information $\tau_{\phi,2}$ further corresponds to the natural number $a(\phi)$ selected by the natural number selecting unit **21102**-ϕ. This enables the computing apparatus **211**-ϕ to evaluate the computation capability provided by the capability providing apparatus **212** with a high degree of accuracy (step S**21103**).

As illustrated in FIG. **38**, the first input information $\tau_{\phi,1}$ is input in the first output information computing unit **21201** of the capability providing apparatus **212** (FIG. **33**) and the second input information $\tau_{\phi,2}$ is input in the second output information computing unit **21202** (step S**21200**).

The first output information computing unit **21201** uses the first input information $\tau_{\phi,1}$ to correctly compute $f_\phi(\tau_{\phi,1})$ with a probability greater than a certain probability and sets the obtained result of the computation as first output information $z_{\phi,1}$ (step S**21201**). The second output information computing unit **21202** uses the second input information $\tau_{\phi,2}$ to correctly computes $f_\phi(\tau_{\phi,2})$ with a probability greater than a certain probability and sets the obtained result of the computation as second output information $z_{\phi,2}$ (step S**21202**). Note that the "certain probability" is a probability less than 100%. An example of the "certain probability" is a nonnegligible probability and an example of the "nonnegligible probability" is a probability greater than or equal to $1/\psi(k)$, where $\psi(k)$ is a polynomial that is a weakly increasing function (non-decreasing function) for a security parameter k. That is, the first output information computing unit **21201** and the second output information computing unit **21202** can output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **21201** may or may not be $f_\phi(\tau_{\phi,1})$ and the result of the computation by the second output information computing unit **21202** may or may not be $f_\phi(\tau_{\phi,2})$. The first output information computing unit **21201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **21202** outputs the second output information $z_{\phi,2}$ (step S**21203**).

Returning to FIG. **37**, the first output information $z_{\phi,1}$ is input in the first computing unit **21105**-ϕ of the computing apparatus **211**-ϕ (FIG. **32**) and the second output information $z_{\phi,2}$ is input in the second computing unit **21108**-ϕ. The first output information $z_{\phi,1}$ and the second output information $z_{\phi,2}$ are equivalent to the computation capability provided by the capability providing apparatus **212** to the computing apparatus **211**-ϕ (step S**21104**). A user of the computing apparatus **211**-ϕ to which the computation capability has been provided pays the capability providing apparatus **212** for the computation capability. The payment may be made through a well-known electronic payment process, for example.

The first computing unit **21105**-ϕ generates computation result $u_\phi=f_\phi(x_\phi)^{b(\phi)}x_{\phi,1}$ from the first output information $z_{\phi,1}$. Here, generating (computing) $f_\phi(x_\phi)^{b(\phi)}x_{\phi,1}$ means computing a value of a formula defined as $f_\phi(x_\phi)^{b(\phi)}x_{\phi,1}$. Any intermediate computation method may be used that can eventually compute the value of the formula $f_\phi(x_\phi)^{b(\phi)}x_{\phi,1}$. The same applies to computations of the other formulae that appear herein. The result u_ϕ of the computation is sent to the first power computing unit **21106**-ϕ (step S**21105**).

The first power computing unit **21106**-ϕ computes $u_\phi'=u_\phi^{a(\phi)}$. The pair of the result u_ϕ of the computation and u_ϕ' computed on the basis of the result of the computation, (u_ϕ, u_ϕ'), is stored in the first list storage **21107**-ϕ (step S**21106**).

The determining unit **21111**-ϕ determines whether or not there is a pair of u_ϕ' and v_ϕ' that belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ among the pairs (u_ϕ, u_ϕ') stored in the first list storage **21107**-ϕ and the pairs (v_ϕ, v_ϕ') stored in the second list storage **21110**-ϕ. Whether $M_\phi=x_\phi$ does not need to be determined (The same applies to the following determinations.). In other words, the determining unit **21111**-ϕ determines whether there is a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ. For example, if the function $f_\phi(M_\phi)$ is an injective function for the element M_ϕ, the determining unit **21111**-ϕ determines whether or not $u_\phi'=v_\phi'$ (step S**21107**). If a pair (v_ϕ, v_ϕ') is not stored in the second list storage **21110**-ϕ, the process at step S**21107** is omitted and the process at step S**21108** is performed. If there is a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21114**. If there is not a pair of u_ϕ' and v_ϕ' that belong to the

class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21108**.

At step S**21108**, the second computing unit **21108**-ϕ generates a computation result $v_\phi=f_\phi(x_\phi)^{a(\phi)}x_{\phi,2}$ from the second output information $z_{\phi,2}$. The result v_ϕ of the computation is sent to the second power computing unit **21109**-ϕ (step S**21108**).

The second power computing unit **21109**-ϕ computes $v_\phi'=v_\phi^{b(\phi)}$. The pair of the result v_ϕ of the computation and v_ϕ' computed on the basis of the computation result, (v_ϕ, v_ϕ'), is stored in the second list storage **21110**-ϕ (step S**21109**).

The determining unit **21111**-ϕ determines whether or not there is a pair of u_ϕ' and v_ϕ' that belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ among the pairs (u_ϕ, u_ϕ') stored in the first list storage **21107**-ϕ and the pairs (v_ϕ, v_ϕ') stored in the second list storage **21110**-ϕ. For example, if the function $f_\phi(M_\phi)$ is an injective function for the element M_ϕ, the determining unit **21111**-ϕ determines whether or not $u_\phi'=v_\phi'$ (step S**21110**). If there is a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21114**. If there is not a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element $M_\phi$the process proceeds to step S**21111**.

At step S**21111**, the controller **21113**-ϕ determines whether or not t=T (step S**21111**). Here, T is a predetermined natural number. T may have the same value for all of ϕ or the value of T in the self-correction process for ϕ and the value of T in the self-correction process for ϕ' (ϕ'≠ϕ, ϕ'=1, . . . , Φ) may be different. If t=T, the final output unit **21112**-ϕ outputs information indicating that the computation is impossible, for example the symbol "⊥" (step S**21113**) and the process ends. If not t=T, the controller **21113**-ϕ increments t by 1, that is, sets t=t+1 (sets t+1 as a new t) (step S**21112**) and the process returns to step S**21103**.

The information indicating that the computation is impossible (the symbol "⊥" in this example) means that the reliability that the capability providing apparatus **212** correctly performs computation is lower than a criterion defined by T. In other words, the capability providing apparatus **212** was unable to perform a correct computation in T trials.

At step S**21114**, the final output unit **21112**-ϕ uses u_ϕ and v_ϕ that correspond to the pair of u_ϕ' and v_ϕ' that are determined to belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ to calculate and output $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}$, and then the process ends (step S**21114**).

The $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}$ thus computed is equal to $f_\phi(x_\phi)\in G_\phi$ with a high probability (the reason why $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)$ with a high probability will be described later). Therefore, $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)$ results with a given reliability (such as a probability) or greater by repeating at least the process for ϕ described above multiple times and selecting the value $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}$ obtained with the highest frequency among the values obtained at step S**21114**. As will be described later, $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)$ can result with an overwhelming probability, depending on settings. This also holds true even if the capability providing apparatus **212** does not necessarily return a correct response. Therefore, the computing apparatus **211**-ϕ does not perform verification for confirming the validity of the capability providing apparatus **212**. Even if a process performed between another computing apparatus **211**-ϕ' (ϕ'≠ϕ) and the capability providing apparatus **212** affects the process performed between the computing apparatus **211**-ϕ and the capability providing apparatus **212**, the computing apparatus **211**-ϕ can obtain a correct computation result $f_\phi(x_\phi)$ if the capability providing apparatus **212** returns a correct solution with a probability greater than a certain probability.

<<Reason Why $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)$ with High Probability>>

Let X_ϕ be a random variable having a value in the group G_ϕ. For $w_\phi\in G_\phi$, an entity that returns $w_\phi x_\phi'$ corresponding to a sample x_ϕ' according to a random variable X_ϕ in response to each request is called a sampler having an error X_ϕ for w_ϕ.

For $w_\phi\in G_\phi$, an entity that returns $w_\phi^{a(\Phi)}x_\phi'$ corresponding to a sample x_ϕ' according to a random variable X_ϕ whenever a natural number $a(\phi)$ is given is called a randomizable sampler having an error X_ϕ for w_ϕ. The randomizable sampler functions as a sampler if used with $a(\phi)=1$.

The combination of the input information providing unit **21104**-ϕ, the first output information computing unit **21201** and the first computing unit **21105**-ϕ of this embodiment is a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(x_\phi)$ (referred to as the "first randomizable sampler") and the combination of the input information providing unit **21104**-ϕ, the second output information computing unit **21202** and the second computing unit **21108**-ϕ is a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(x_\phi)$ (referred to as the "second randomizable sampler").

The inventor has found that if u_ϕ' and v_ϕ' belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, it is highly probable that the first randomizable sampler has correctly computed $u(=f_\phi(x_\phi)^{b(\Phi)}$ and the second randomizable sampler have correctly computed $v_\phi=f_\phi(x_\phi)^{a(\Phi)}$ ($x_{\phi,1}$ and $x_{\phi,2}$ are identity elements $e_{\phi,g}$ of the group G_ϕ). For simplicity of explanation, this will be proven in an eighteenth embodiment.

When the first randomizable sampler correctly computes $u_\phi=f_\phi(x_\phi)^{b(\Phi)}$ and the second randomizable sampler correctly computes $v_\phi=f_\phi(x_\phi)^{a(\Phi)}$ (when $x_{\phi m1}$ and $x_{\phi,2}$ are identity elements $e_{\phi,g}$ of the group G_ϕ), then $u_\phi^{b'(\Phi)}=(f_\phi(x_\phi)^{b(\Phi)}x_{\phi,1})^{b'(\Phi)}=(f_\phi(x_\phi)^{b(\Phi)}e_{\phi,g})^{b'(\Phi)}=f_\phi(x_\phi)^{b(\Phi)b'(\Phi)}$ and $v_\phi^{a'(\Phi)}=(f_\phi(x_\phi)^{a(\Phi)}x_{\phi,2})^{a'(\Phi)}=(f_\phi(x_\phi)^{a(\Phi)}e_{\phi,g})^{a'(\Phi)}=f_\phi(x_\phi)^{a(\Phi)a'(\Phi)}$. Therefore, if the function $f_\phi(M_\phi)$ is an injective function for the element M_ϕ, then $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)^{(b(\Phi)b'(\Phi)+a(\Phi)a'(\Phi))}=f_\phi(x_\phi)$. On the other hand, if the function $f_\phi(M_\phi)$ is not an injective function for the element M_ϕ but if a homomorphic function, then $u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}=f_\phi(x_\phi)^{(b(\Phi)b'(\Phi)+a(\Phi)a'(\Phi))}=f_\phi(x_\phi)$.

For $(q_1, q_2)\in I$, a function π_i is defined as $\pi_i(q_1, q_2)=q_i$ for each of i=1, 2. Let L=min $(\#\pi_1(S), \#\pi_2(S))$, where $\#\cdot$ is the order of a set ·. If the group G_ϕ is a cyclic group or a group whose order is difficult to compute, it can be expected that the probability that an output of the computing apparatus **211**-ϕ other than "⊥" is not $f_\phi(x_\phi)$ is at most approximately $T^2L/\#S$ within a negligible error. If L/#S is a negligible quantity and T is a quantity approximately equal to a polynomial order, the computing apparatus **211**-ϕ outputs a correct $f_\phi(x_\phi)$ with an overwhelming probability. An example of S that results in a negligible quantity of L/#S is $S=\{(1, d)|d\in[2,|G_\phi|-1]\}$.

Thirteenth Embodiment

A proxy computing system of an thirteenth embodiment is an example that embodies the first randomizable sampler and the second randomizable sampler described above. The following description will focus on differences from the twelfth embodiment and repeated description of commonalities with the twelfth embodiment will be omitted. In the following description, elements labeled with the same ref-

erence numerals have the same functions and the steps labeled with the same reference numerals represent the same processes.

<Configuration>

As illustrated in FIG. **31**, the proxy computing system **202** of the thirteenth embodiment includes a computing apparatus **221**-ϕ in place of the computing apparatus **211**-ϕ and a capability providing apparatus **222** in place of the capability providing apparatus **212**.

As illustrated in FIG. **32**, the computing apparatus **221**-ϕ of the thirteenth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **22104**-ϕ, a first computing unit **22105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **22108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **21111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ. As illustrated in FIG. **34**, the input information providing unit **22104**-ϕ of this embodiment includes, for example, a first random number generator **22104***a*-ϕ, a first input information computing unit **22104***b*-ϕ, a second random number generator **22104***c*-ϕ, and a second input information computing unit **22104***d*-ϕ.

As illustrated in FIG. **33**, the capability providing unit **222** of the thirteenth embodiment includes, for example, a first output information computing unit **22201**, a second output information computing unit **22202**, and a controller **21205**.

<Assumptions for Processes>

In the thirteenth embodiment, a function f_ϕ is a homomorphic function, a group H_ϕ is a cyclic group, a generator of the group H_ϕ is $\mu_{\phi,h}$, the order of the group H_ϕ is $K_{\phi,H}$, and $\nu_\phi=f_\phi(\mu_{\phi,h})$. The rest of the assumptions are the same as those in the twelfth embodiment, except that the computing apparatus **211**-ϕ is replaced with the computing apparatus **221**-ϕ and the capability providing apparatus **212** is replaced with the capability providing apparatuses **222**.

<Process>

As illustrated in FIGS. **37** and **38**, the process of the thirteenth embodiment is the same as the process of the twelfth embodiment except that steps S**21103** through S**21105**, S**21108**, and S**21200** through S**21203** of the twelfth embodiment are replaced with steps S**22103** through S**22105**, S**22108**, and S**22200** through S**22203**, respectively. In the following, only processes at steps S**22103** through S**22105**, S**22108**, and S**22200** through S**22203** will be described.

<<Process at Step S**22103**>>

The input information providing unit **22104**-T of the computing apparatus **221**-ϕ (FIG. **32**) generates and outputs first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ each of which corresponds to an input element x_ϕ (step S**22103** of FIG. **37**). A process at step S**22103** of this embodiment will be described below with reference to FIG. **39**.

The first random number generator **22104***a*-ϕ (FIG. **34**) generates a uniform random number r(ϕ, 1) that is a natural number greater than or equal to 0 and less than $K_{\phi,H}$. The generated random number r(ϕ, 1) is sent to the first input information computing unit **22104***b*-ϕ and the first computing unit **22105**-ϕ (step S**22103***a*). The first input information computing unit **22104***b*-ϕ uses the input random number r(ϕ, 1), the element x_ϕ and a natural number b(ϕ) to compute first input information $\tau_{\phi,1}=\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)}$ (step S**22103***b*).

The second random number generator **22104***c*-ϕ generates a uniform random number r(ϕ, 2) that is a natural number greater than or equal to 0 and less than $K_{\phi,H}$. The generated random number r(ϕ, 2) is sent to the second input information computing unit **22104***d*-ϕ and the second computing unit **22108**-ϕ (step S**22103***c*). The second input information computing unit **22104***d*-ϕ uses the input random number r(ϕ, 2), the element x_ϕ, and a natural number a(ϕ) to compute second input information $\tau_{\phi,2}=\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)}$ (step S**22103***d*).

The first input information computing unit **22104***b*-ϕ and the second input information computing unit **22104***d*-ϕ output the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ thus generated (step S**22103***e*). Note that the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ of this embodiment are information whose relation with the element x_ϕ is scrambled using random numbers r(ϕ, 1) and r(ϕ, 2), respectively. This enables the computing apparatus **221**-ϕ to conceal the element x_ϕ from the capability providing apparatus **222**. The first input information $t_{\phi,1}$ of this embodiment further corresponds to the natural number b(ϕ) selected by the natural number selecting unit **21102**-ϕ and the second input information $\tau_{\phi,2}$ further corresponds to the natural number a(ϕ) selected by the natural number selecting unit **21102**-ϕ. This enables the computing apparatus **221**-ϕ to evaluate the computing capability provided by the capability providing apparatus **222** with a high degree of accuracy.

<<Processes at Steps S**22200** through S**22203**>>

As illustrated in FIG. **38**, first, the first input information $\tau_{\phi,1}=\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)}$ is input in the first output information computing unit **22201** of the capability providing apparatus **222** (FIG. **33**) and the second input information $\tau_{\phi,2}=\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)}$ is input in the second output information computing unit **22202** (step S**22200**).

The first output information computing unit **22201** uses the first input information $\tau_{\phi,1}=\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)}$ to correctly compute $f_\phi(\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)})$ with a probability greater than a certain probability and sets the obtained result of the computation as first output information $z_{\phi,1}$. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **22201** may or may not be $f_\phi(\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)})$ (step S**22201**).

The second output information computing unit **22202** uses the second input information $\tau_{\phi,2}=\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)}$ to correctly compute $f_\phi(\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)})$ with a probability greater than a certain probability and sets the obtained result of the computation as second output information $z_{\phi,2}$. The result of the computation may or may not be correct. That is, the result of computation by the second output information computing unit **22202** may or may not be $f_\phi(\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)})$ (step S**22202**).

The first output information computing unit **22201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **22202** outputs the second output information $z_{\phi,2}$ (step S**22203**).

<<Processes at Steps S**22104** and S**22105**>>

Returning to FIG. **37**, the first output information $z_{\phi,1}$ is input in the first computing unit **22105**-ϕ of the computing apparatus **221**-ϕ (FIG. **32**) and the second input information $z_{\phi,2}$ is input in the second computing unit **22108**-ϕ. The first output information $z_{\phi,1}$ and the second output information $z_{\phi,2}$ are equivalent to the computing capability provided by the capability providing apparatus **222** to the computing apparatus **221**-ϕ (step S**22104**).

The first computing unit **22105**-ϕ uses the input random number r(ϕ, 1) and the first output information $z_{\phi,1}$ to compute $z_{\phi,1}\nu_\phi{}^{-r(\phi,1)}$ and sets the result of the computation as u_ϕ. The result u_ϕ of the computation is sent to the first

power computing unit **21106**-ϕ. Here, $u_\phi=z_{\phi,1}v_\phi^{-r(\phi,1)}=f_\phi(x_\phi)^{b(\phi)}x_{\phi,1}$. That is, $z_{\phi,1}v_\phi^{-r(\phi,1)}$ is an output of a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(x_\phi)$. The reason will be described later (step **S22105**).

<<Process at Step **S22108**>>

The second computing unit **22108**-ϕ uses the input random number r(ϕ, 2) and the second output information $z_{\phi,2}$ to compute $z_{\phi,2}v_\phi^{-r(\phi,2)}$ and sets the result of the computation as v_ϕ. The result $v_\phi$of the computation is sent to the secondpower computing unit **21109**-ϕ. Here, $v_\phi=z_{\phi,2}v_\phi^{-r(\phi,2)}=f_\phi(x_\phi)^{a(\phi)}x_{\phi,2}$. That is, $z_{\phi,2}v_\phi^{-r(\phi,2)}$ is an output of a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(x_\phi)$. The reason will be described later (step **S22108**).

<<Reason Why $z_{\phi,1}v_\phi^{-r(\phi,1)}$ and $z_{\phi,2}v_\phi^{-r(\phi,2)}$ are Outputs of Randomizable Samplers Having Errors $X_{\phi,1}$ and $X_{\phi,2}$, Respectively, for $f_\phi(x_\phi)$>>

Let c be a natural number, R and R' be random numbers, and $B(\mu_{\phi,h}{}^R x_\phi{}^c)$ be the result of computation performed by the capability providing apparatus **222** using $\mu_{\phi,h}{}^R x_\phi{}^c$. That is, the results of computations that the first output information computing unit **22201** and the second output information computing unit **22202** return to the computing apparatus **221**-ϕ are $z_\phi=B(\mu_{\phi,h}{}^R x_\phi{}^c)$. A random variable X_ϕ that has a value in the group G_ϕ is defined as $X_\phi=B(\mu_{\phi,h}{}^{R'})f_\phi(\mu_{\phi,h}{}^{R'})^{-1}$.

Then, $z_\phi v_\phi^{-R}=B(\mu_{\phi,h}{}^R x_\phi{}^c)f(\mu_{\phi,h})^{-R}=X_\phi f_\phi(\mu_{\phi,h}{}^R x_\phi{}^c)f_\phi(\mu_{\phi,h})^{-R}=X_\phi f_\phi(\mu_{\phi,h})^R f_\phi(x_\phi)^c f_\phi(\mu_{\phi,h})^{-R}=f_\phi(x_\phi)^c X_\phi$. That is, $z_\phi v_\phi^{-R}$ is an output of a randomizable sampler having an error X_ϕ for $f_\phi(x_\phi)$.

The expansion of formula given above uses the properties such that $X_\phi=B(\mu_{\phi,h}{}^{R'})f_\phi(\mu_{\phi,h}{}^{R'})^{-1}=B(\mu_{\phi,h}{}^R x_\phi{}^c)f_\phi(\mu_{\phi,h}{}^R x_\phi{}^c)f_\phi(\mu_{\phi,h}{}^R x_\phi{}^c)^{-1}$ and that $B(\mu_{\phi,h}{}^R x_\phi{}^c)=X_\phi f_\phi(\mu_{\phi,h}{}^R x_\phi{}^c)$. The properties are based on the fact that the function f_ϕ is a homomorphic function and R and R' are random numbers.

Therefore, considering that a(ϕ) and b(ϕ) are natural numbers and r(ϕ, 1) and r(ϕ, 2) are random numbers, $z_{\phi,1}v^{-r(\phi,1)}$ and $z_{\phi,2}v^{-r(\phi,2)}$ are, likewise, outputs of randomizable samplers having errors $X_{\phi,1}$ and $X_{\phi,2}$, respectively, for $f_\phi(x_\phi)$.

Fourteenth Embodiment

A fourteenth embodiment is a variation of the thirteenth embodiment and computes a value of $u_\phi$or v_ϕ by using samplers described above when a(ϕ)=1 or b(ϕ)=1. The amounts of computations performed by samplers in general are smaller than the amounts of computations by randomizable samplers. Using samplers instead of randomizable samplers for computations when a(ϕ)=1 or b(ϕ)=1 can reduce the amounts of computations by the proxy computing system. The following description will focus on differences from the twelfth and thirteenth embodiments and repeated description of commonalities with the twelfth and thirteenth embodiments will be omitted.

<Configuration>

As illustrated in FIG. **31**, a proxy computing system **203** of the fourteenth embodiment includes a computing apparatus **231**-ϕ in place of the computing apparatus **221**-ϕ and a capability providing apparatus **232** in place of the capability providing apparatus **222**.

As illustrated in FIG. **32**, the computing apparatus **231**-ϕ of the fourteenth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **22104**-ϕ, a first computing unit **22105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **22108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **21111**-ϕ, a final output unit **21112**-ϕ, a controller **21113**-ϕ, and a third computing unit **23109**-ϕ.

As illustrated in FIG. **33**, the capability providing apparatus **232** of the fourteenth embodiment includes, for example, a first output information computing unit **22201**, a second output information computing unit **22202**, a controller **21205**, and a third output information computing unit **23203**.

<Processes>

Processes of this embodiment will be described below. What follows is a description of differences from the thirteenth embodiment.

As illustrated in FIGS. **37** and **38**, a process of the fourteenth embodiment is the same as the process of the thirteenth embodiment except that steps **S22103** through **S22105**, **S22108**, and **S22200** through **S22203** of the thirteenth embodiment are replaced with steps **S23103** through **S23105**, **S23108**, **S22200** through **S22203**, and **S23205** through **S3209**, respectively. The following description will focus on processes at steps **S23103** through **S23105**, **S23108**, **S22200** through **S22203**, and **S23205** through **S23209**.

<<Process at Step **S23103**>>

The input information providing unit **23104**-ϕ of the computing apparatus **231**-ϕ (FIG. **32**) generates and outputs first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ each of which corresponds to an input element x_ϕ(step **S23103** of FIG. **37**).

A process at **S23103** of this embodiment will be described below with reference to FIG. **39**.

The controller **21113**-ϕ (FIG. **32**) controls the input information providing unit **23104**-ϕ according to natural numbers (a(ϕ), b(ϕ)) selected by the natural number selecting unit **21102**-ϕ.

Determination is made by the controller **21113**-ϕ as to whether b(ϕ) is equal to 1 (step **S23103***a*). If it is determined that b(ϕ)≠1, the processes at steps **S22103***a* and **22103***b* described above are performed and the process proceeds to step **S23103***g*.

On the other hand, if it is determined at step **S23103***a* that b(ϕ)=1, the third random number generator **23104***e*-ϕ generates a random number r(ϕ, 3) that is a natural number greater than or equal to 0 and less than $K_{\phi,H}$. The generated random number r(ϕ, 3) is sent to the third input information computing unit **23104***f*-ϕ and the third computing unit **23109**-ϕ (step **S23103***b*). The third input information computing unit **23104***f*-ϕ uses the input random number r(ϕ, 3) and the element x_ϕ to compute $x_\phi{}^{r(\phi,3)}$ and sets it as first input information $\tau_{\phi,1}$ (step **S23103***c*). Then the process proceeds to step **S23103***g*.

At step **S23103***g*, determination is made by the controller **21113***a*-ϕ as to whether a(ϕ) is equal to 1 (step **S23103***g*). If it is determined that a(ϕ)≠1, the processes at steps **S22103***c* and **S22103***d* described above are performed.

On the other hand, if it is determined at step **S23103***g* that a(ϕ)=1, the third random number generator **23104***e*-ϕ generates a random number r(ϕ, 3) that is a natural number greater than or equal to 0 and less than $K_{\phi,H}$. The generated random number r(ϕ, 3) is sent to the third input information computing unit **23104***f*-ϕ (step **S23103***h*). The third input information computing unit **23104***f*-ϕ uses the input random number r(ϕ, 3) and the element x_ϕ to compute $x_\phi{}^{r(\phi,3)}$ and sets it as second input information $\tau_{\phi,2}$ (step **S23103***i*).

The first input information computing unit **22104***b*-ϕ, the second input information computing unit **22104***d*-ϕ, and the third input information computing unit **23104***f*-ϕ output the first input information $\tau_{\phi,1}$ and the second input information

$\tau_{\phi,2}$ thus generated along with information on the corresponding natural numbers (a(ϕ), b(ϕ)) (step S**23103***e*). Note that the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ in this embodiment are information whose relation with the element x_ϕ is scrambled using the random numbers r(ϕ, 1), r(ϕ, 2) and r(3, ϕ). This enables the computing apparatus **231**-ϕ to conceal the element x_ϕ from the capability providing apparatus **232**.

<<Processes at S**22200** Through S**22203** and S**23205** Through S**23209**>>

Processes at S**22200** through S**22203** and S**23205** through S**23209** of this embodiment will be described below with reference to FIG. **38**.

The controller **21205** (FIG. **33**) controls the first output information computing unit **22201**, the second output information computing unit **22202** and the third output information computing unit **23203** according to input natural numbers (a(ϕ), b(ϕ)).

Under the control of the controller **21205**, the first input information $\tau_{\phi,1}=\mu_{\phi,h}{}^{r(\phi,1)}x_\phi{}^{b(\phi)}$ when b(ϕ)≠1 is input in the first output information computing unit **22201** of the capability providing apparatus **232** (FIG. **33**) and the second input information $\tau_{\phi,2}=\mu_{\phi,h}{}^{r(\phi,2)}x_\phi{}^{a(\phi)}$ when a(λ)≠1 is input in the second output information computing unit **22202**. The first input information $\tau_{\phi,1}=x_\phi{}^{r(\phi,3)}$ when b(ϕ)=1 and the second input information $\tau_{\phi,2}=x_\phi{}^{r(\phi,3)}$ when a(ϕ)=1 are input in the third output information computing unit **23203** (step S**23200**).

Determination is made by the controller **21113**-ϕ as to whether b(ϕ) is equal to 1 (step S**23205**). If it is determined that b(ϕ)≠1, the process at step S**22201** described above is performed. Then, determination is made by the controller **21113**-ϕ as to whether a(ϕ) is equal to 1 (step S**23208**). If it is determined that a(ϕ)≠1, the process at step S**22202** described above is performed and then the process proceeds to step S**23203**.

On the other hand, if it is determined at step S**23208** that a(ϕ)=1, the third output information computing unit **23203** uses the second input information $\tau_{\phi,2}=x_\phi{}^{r(\phi,3)}$ to correctly compute $f_\phi(x_\phi{}^{r(\phi,3)})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information $z_{\phi,3}$. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **23203** may or may not be $f_\phi(x_\phi{}^{r(\phi,3)})$ (step S**23209**). Then the process proceeds to step S**23203**.

If it is determined at step S**23205** that b(λ)=1, the third output information computing unit **23203** uses the second input information $\tau_{\phi,1}=x_\phi{}^{r(\phi,3)}$ to correctly compute $f_\phi(x_\phi{}^{r(\phi,3)})$ with a probability greater than a certain probability and sets the obtained result of the computation as third output information $z_{\phi,3}$. The result of the computation may or may not be correct. That is, the result of the computation by the third output information computing unit **23203** may or may not be $f_\phi(x_\phi{}^{r(\phi,3)})$ (step S**23206**).

Then, determination is made by the controller **21113**-ϕ as to whether a(ϕ) is equal to 1 (step S**23207**). If it is determined that a(ϕ)=1, the process proceeds to step S**23203**; if it is determined a(ϕ)≠1, the process proceeds to step S**22202**.

At step S**23203**, the first output information computing unit **22201**, which has generated the first output information $z_{\phi,1}$, outputs the first output information $z_{\phi,1}$, the second output information computing unit **22202**, which has generated the second output information $z_{\phi,2}$, outputs the second output information $z_{\phi,2}$, and the third output information computing unit **23202**, which has generated the third output information $z_{\phi,3}$, outputs the third output information $z_{\phi,3}$ (step S**23203**).

<<Processes at Steps S**23104** and S**23105**>>

Returning to FIG. **37**, under the control of the controller **21113**-ϕ, the first output information $z_{\phi,1}$ is input in the first computing unit **22105**-ϕ of the computing apparatus **231**-ϕ (FIG. **32**), the second output information $z_{\phi,2}$ is input in the second computing unit **22108**-ϕ, and the third output information $z_{\phi,3}$ is input in the third computing unit **23109**-ϕ (step S**23104**).

If b(ϕ)≠1, the first computing unit **22105**-ϕ performs the process at step S**22105** described above to generate u_ϕ; if b(ϕ)=1, the third computing unit **23109**-ϕ computes $Z_{\phi,3}{}^{1/r(\phi,3)}$ and sets the result of the computation as u_ϕ. The computation result u_ϕ is sent to the first power computing unit **21106**-ϕ. Here, if b(ϕ)=1, then $u_\phi=z_{\phi3}{}^{1/r(\phi,3)}=f_\phi(x_\phi)x_{\phi,3}$. That is, $z_{\phi,3}{}^{1/r(\phi,3)}$ serves as a sampler having an error $X_{\phi,3}$ for $f_\phi(x_\phi)$. The reason will be described later (step S**23105**).

<<Process at Step S**23108**>>

If a(ϕ)≠1, the second computing unit **22108**-ϕ performs the process at step S**22108** described above to generate v_ϕ; if a(ϕ)=1, the third computing unit **23109**-ϕ computes $z_{\phi,3}{}^{1/r(\phi,3)}$ and sets the result of the computation as v_ϕ. The computation result v_ϕ is sent to the second power computing unit **21109**-ϕ. Here, if a(ϕ)=1, then $v_\phi=z_{\phi,3}{}^{1/r(\phi,3)}=f_\phi(x_\phi)x_{\phi,3}$. That is, $z_{\phi,3}{}^{1/r(\phi,3)}$ serves as a sampler having an error $X_{\phi,3}$ for $f_\phi(x_\phi)$. The reason will be described later (step S**23108**).

Note that if $z_{\phi,3}{}^{1/r(\phi,3)}$, that is, the radical root of $z_{\phi,3}$, is hard to compute, u_ϕ and/or v_ϕ may be calculated as follows. The third computing unit **23109**-ϕ may store each pair of a random number r(ϕ, 3) and $z_{\phi,3}$ computed on the basis of that random number r(ϕ, 3) in a storage, not depicted, in sequence as $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_m, \beta_m), \ldots$, where m is a natural number. The third computing unit **23109**-ϕ may compute $\gamma_1, \gamma_2, \ldots \gamma_m$ that satisfy $\gamma_1\alpha_1+\gamma_2\alpha_2+\ldots+\gamma_m\alpha_m=1$ when the least common multiple of $\alpha_1, \alpha_2, \ldots, \alpha_m$ is 1, where $\gamma_1, \gamma_2, \ldots, \gamma_m$ are integers. The third computing unit **32109**-ϕ may then use the resulting $\gamma_1, \gamma_2, \ldots, \gamma_m$ to compute $\Pi_{i=1}{}^{m}\beta_i{}^{\gamma i}=\beta_1{}^{\gamma 1}\beta_2{}^{\gamma 2}\ldots\beta_m{}^{\gamma m}$ and may set the results of the computation as u_ϕ and/or v_ϕ.

<<Reason Why $z_{\phi,3}{}^{1/r(\phi,3)}$ Serves as a Sampler Having an Error $X_{\phi,3}$ for $f_\phi(x_\phi)$>>

Let R be a random number and $B(x_\phi{}^R)$ be the result of computation performed by the capability providing apparatus **232** using $x_\phi{}^R$. That is, let $z_\phi=B(x_\phi{}^R)$ be computation results returned by the first output information computing unit **22201**, the second output information computing unit **22202**, and the third output information computing unit **23203** to the computing apparatus **231**-ϕ. Furthermore, a random variable X_ϕ having a value in the group G_ϕ is defined as $X_\phi=B(x_\phi{}^R)^{1/R}f_\phi(x_\phi)^{-1}$.

Then, $z_\phi{}^{1/R}=B(x_\phi{}^R)^{1/R}=X_\phi f_\phi(x_\phi)=f_\phi(x_\phi)X_\phi$. That is, $z_\phi{}^{1/R}$ serves as a sampler having an error X_ϕ for $f_\phi(x_\phi)$.

The expansion of formula given above uses the properties such that $X_\phi=B(x_\phi{}^R)^{1/R}f_\phi(x_\phi{}^R)^{-1}$ and that $\beta(x_\phi{}^R)^{1/R}=X_\phi f_\phi(x_\phi{}^R)$. The properties are based on the fact that R is a random number.

Therefore, considering that r(ϕ, 3) is a random number, $z_\phi{}^{1/R}$ serves as a sampler having an error $x_{\phi,3}$ for $f_\phi(x_\phi)$, likewise.

Fifteenth Embodiment

A proxy computing system of a fifteenth embodiment is another example that embodies the first and second randomizable samplers described above. Specifically, in this

embodiment, $f_\phi(x_\phi)$ is a function for converting an element $x_\phi=C_{\phi,1}(y(\phi,1), m_\phi)$ of a group H_ϕ which is a first ciphertext to a second ciphertext $f_\phi(x_\phi)=C_{\phi,2}(y(\phi,2), m_\phi)$ which is an element of a group G_ϕ. Here, the first ciphertext $C_{\phi,1}(y(\phi, 1), m_\phi)$ is a ciphertext obtained by encrypting a plaintext m_ϕ with a first encryption key $y(\phi, 1)$ according to a first encryption scheme $ENC_{\phi,1}$ and the second ciphertext $C_{\phi,2}(y(\phi, 2), m_\phi)$ is a ciphertext obtained by encrypting the plaintext m_ϕ with a second encryption key $y(\phi, 2)$ according to a second encryption scheme $ENC_{\phi,2}$. The second encryption scheme $ENC_{\phi,2}$ is the ElGamal encryption and the function $f_\phi(x_\phi)$ is a homomorphic function. The first encryption scheme $ENC_{\phi,1}$ may be any encryption scheme; the first encryption scheme $ENC_{\phi,1}$ may be probabilistic encryption such as the ElGamal encryption or may be deterministic encryption such as RSA encryption.

The following description will focus on differences from the twelfth embodiment and repeated description of commonalities with the twelfth embodiment will be omitted.

As illustrated in FIG. **31**, a proxy computing system **204** of the fifteenth embodiment includes a computing apparatus **241**-ϕ in place of the computing apparatus **221**-ϕ and a capability providing apparatus **242** in place of the capability providing apparatus **212**.

As illustrated in FIG. **32**, the computing apparatus **241**-ϕ of the fifteenth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **24104**-ϕ, a first computing unit **24105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **24108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **24111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ. As illustrated in FIG. **35**, the input information providing unit **24104**-ϕ of this embodiment includes, for example, a first random number generator **24104***a*-ϕ, a first input information computing unit **24104***b*-ϕ, a second random number generator **24104***c*-ϕ, and a second input information computing unit **24104***d*-ϕ.

As illustrated in FIG. **33**, the capability providing apparatus **242** of the fifteenth embodiment includes, for example, a first output information computing unit **24201**, a second output information computing unit **24202**, and a controller **21205**.

<Assumptions for Processes>

In the fifteenth embodiment, the group G_ϕ is the direct product group $G_{\phi,1}\times G_{\phi,2}$ of cyclic groups $G_{\phi,1}$ and $G_{\phi,2}$, $\mu_{\phi,g1}$ is a generator of the group $G_{\phi,1}$, $\mu_{\phi,g2}$ is a generator of the group $G_{\phi,2}$, the second encryption key $y(\phi, 2)$ is $\mu_{\phi,g2}^{s(\phi,2)}$, an element $C_{\phi,2}(y(\phi, 2), m_\phi)$ is $(\mu_{\phi,g1}^{r(\Phi)}, m_\phi y(\phi, 2)^{r(\Phi)})$ $\in G_{\phi,1}\times G_{\phi,2}$, $r(\phi)$ is an integer random number, a value $u_\phi^{a(\Phi)}$ is $(c_{\phi,1u}, c_{\phi,2u})\in G_{\phi,1}\times G_{\phi,2}$, and a value $v_\phi^{b(\Phi)}$ is $(c_{\phi,1v}, c_{\phi,2v})$ $\in G_{\phi,1}\times G_{\phi,2}$. $G_{\phi,1}$ may be equal to $G_{\phi,2}$ or may be unequal to $G_{\phi,2}$. The first encryption scheme $ENC_{\phi,1}$ may be any encryption scheme as stated above. If the first encryption scheme $ENC_{\phi,1}$ is the ElGamal encryption, the group H_ϕ is the direct product group $H_{\phi,1}\times H_{\phi,2}$ of cyclic groups $H_{\phi,1}$ and $H_{\phi,2}$, $r'(\phi)$ is an integer random number, $\mu_{\phi,h1}$ is a generator of the group $H_{\phi,1}$, $\mu_{\phi,h2}$ is a generator of the group $H_{\phi,2}$, the first encryption key $y(\phi, 1)$ is $\mu_{\phi,h2}^{s(\phi,1)}$, the first ciphertext $C_{\phi,1}(y(\phi, 1), m_\phi)$ is $(\mu_{\phi,h1}^{r'(\Phi)}, m_\phi y(\phi, 1)^{r'(\Phi)})\in H_{\phi,1}\times H_{\phi,2}$. $H_{\phi,1}$ may be equal to $H_{\phi,2}$ or may be unequal to $H_{\phi,2}$.

Note that if $A=(\alpha_1, \alpha_2)\in G_{\phi,1}\times G_{\phi,2}$, $B=(\beta_1, \beta_2)\in G_{\phi,1}\times G_{\phi,2}$, and ϵ is a natural number, then A^ϵ represents $(\alpha_1^\epsilon, \alpha_2^\epsilon)$, $A^{-\epsilon}$ represents $(\alpha_1^{-\epsilon}, \alpha_2^{-\epsilon})$, and AB represents $(\alpha_1\beta_1, \alpha_2\beta_2)$. Similarly, if ϵ is a natural number, $A=(\alpha_1, \alpha_2)\in H_{\phi,1}\times H_{\phi,2}$, and $B=(\beta_1, \beta_2)\in H_{\phi,1}\times H_{\phi,2}$, then A^ϵ represents $(\alpha a_1^\epsilon, \alpha_2^\epsilon)$, $A^{-\epsilon}$ represents $(\alpha_1^{-\epsilon}, \alpha_2^{-\epsilon})$, and AB represents $(\alpha_1\beta_1, \alpha_2\beta_2)$. $e_\phi(\alpha, \beta)$ is a bilinear map that gives an element of a cyclic group $G_{\phi,T}$ for $(\alpha, \beta)\in G_{\phi,1}\times G_{\phi,2}$. Examples of the bilinear map include functions and algorithms for performing pairing computations such as Weil pairing and Tate pairing (see Reference literature 2: Alfred J. Menezes, "ELLIPTIC CURVE PUBLIC KEY CRYPTOSYSTEMS", KLUWER ACADEMIC PUBLISHERS, ISBN 0-7923-9368-6, pp. 61-81 and Reference literature 3: RFC 5091, "Identity-Based Cryptography Standard (IBCS) #1", Supersingular Curve Implementations of the BF and BB1 Cryptosystems, for example).

<Processes>

As illustrated in FIGS. **37** and **38**, a process of the fifteenth embodiment is the same as the process of the twelfth embodiment except that steps **S21103** through **S21105**, **S21107**, **S21108**, **S21110**, and **S21200** through **S21203** of the twelfth embodiment are replaced with steps **S24103** through **S24105**, **S241017**, **S24108**, **S24110**, and **S24200** through **S24203**, respectively. In the following, only processes at steps **S24103** through **S24105**, **S24107**, **S24108**, **S24110**, and **S24200** through **S24203** will be described.

<<Process at Step S24103>>

The input information providing unit **24104**-ϕ of the computing apparatus **241**-ϕ (FIG. **32**) generates and outputs first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ corresponding to an input element $x_\phi=C_{\phi,1}(y(\phi, 1), m_\phi)$ (step **S24103** of FIG. **37**). The process at step **S24103** of this embodiment will be described below with reference to FIG. **40**.

The first random number generator **24104***a*-ϕ (FIG. **35**) generates an arbitrary element ${}_hr_{\phi,1}\in H_\phi$ of the group H_ϕ. In this embodiment an element ${}_hr_{\phi,1}$ is randomly and uniformly selected from the group H_ϕ (uniform random number). The generated element ${}_hr_{\phi,1}$ is sent to the first input information computing unit **2410413**-ϕ and the first computing unit **24105**-ϕ (step **S24103***a*).

The first input information computing unit **24104***b*-ϕ uses a natural number $b(\phi)$ selected by the natural number selecting unit **21102**-ϕ, the element x_ϕ, the element ${}_hr_{\phi,1}$, and the first encryption key $y(\phi, 1)$ to compute $x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_hr_{\phi,1})$ as first input information $\tau_{\phi,1}$ (step **S24103***b*).

The second random number generator **24104***c*-ϕ generates an arbitrary element ${}_hr_{\phi,2}\in H_\phi$ of the group H_ϕ. In this embodiment an element ${}_hr_{\phi,2}$ is randomly and uniformly selected from the group H_ϕ (uniform random number). The generated element ${}_hr_{\phi,2}$ is sent to the second input information computing unit **24104***d*-ϕ and the second computing unit **24108**-ϕ (step **S24103***c*).

The second input information computing unit **24104***b*-ϕ uses a natural number $a(\phi)$ selected by the natural number selecting unit **21102**-ϕ, the element x_ϕ, the element ${}_hr_{\phi,2}$, and the first encryption key $y(\phi, 1)$ to compute $x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_hr_{\phi,2})$ as second input information $\tau_{\phi,2}$ (step **S24103***d*).

The first input information computing unit **24104***b*-ϕ outputs the first input information $\tau_{\phi,1}=x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_hr_{\phi,1})$ computed as described above. The second input information computing unit **24104***d*-ϕ outputs the second input information $\tau_{\phi,2}=x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_hr_{\phi,2})$ computed as described above (step **S24103***e*).

<<Processes at Steps S24200 Through S24203>>

As illustrated in FIG. **38**, first, the first input information $\tau_{\phi,1}=x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_hr_{\phi,1})$ is input in the first output information computing unit **24201** of the capability providing apparatus **242** (FIG. **33**) and the second input informa-

tion $\tau_{\phi,2}=x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,2})$ is input in the second output information computing unit **24202** (step S**24200**).

The first output information computing unit **24201** uses the first input information $\tau_{\phi,1}=x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,1})$, a first decryption key s(ϕ, 1) corresponding to the first encryption key y(ϕ, 1), and the second encryption key y(ϕ, 2) to correctly compute $f_\phi(x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,1}))$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\phi,1}$. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **24201** may or may not be $f_\phi(x_\phi^{b(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,1}))$ (step S**24201**).

Note that the function f_ϕ in this embodiment is a homomorphic function for encrypting a value that can be obtained by decrypting a ciphertext with the first decryption key s(ϕ, 1) according to the first encryption scheme $ENC_{\phi,1}$ with the second encryption key y(ϕ, 2) according to the ElGamal encryption. For example, if both of the first encryption scheme $ENC_{\phi,1}$ and the second encryption scheme $ENC_{\phi,2}$ are the ElGamal encryption, the function f_ϕ is a homomorphic function for encrypting a value that can be obtained by decrypting a ciphertext with the first decryption key s(ϕ, 1) according to the ElGamal encryption with the second encryption key y(ϕ, 2) according to the ElGamal encryption.

The second output information computing unit **24202** can use the second input information $\tau_{\phi,2}=x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,2})$, the first decryption key s(ϕ, 1), and the second encryption key y(ϕ, 2) to correctly compute $f_\phi(x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,2}))$ with a probability greater than a certain probability and sets the result of the computation as second output information $z_{\phi,2}$. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **24202** may or may not be $f_\phi(x_\phi^{a(\Phi)}C_{\phi,1}(y(\phi, 1), {}_h r_{\phi,2}))$ (step S**24202**).

The first output information computing unit **24201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **24202** outputs the second output information $z_{\phi,2}$ (step S**24203**).

<<Processes at Steps S**24104** and S**24105**>>

Returning to FIG. **37**, the first output information $z_{\phi,1}$ is input in the first computing unit **24105**-ϕ of the computing apparatus **241**-ϕ (FIG. **32**) and the second output information $z_{\phi,2}$ is input in the second computing unit **24108**-ϕ (step S**24104**).

The first computing unit **24105**-ϕ uses the input first output information $z_{\phi,1}$, the element ${}_h r_{\phi,1}$, and the second encryption key y(ϕ, 2) to compute $z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,1}))^{-1}$ and sets the result of the computation as u_ϕ (step S**24105**). The result u_ϕ of the computation is sent to the first power computing unit **21106**-ϕ. Here, $u_\phi=z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,1}))^{-1}=f_\phi(x_\phi)^{b(\Phi)}x_{\phi,1}$. That is, $z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,1}))^{-1}$ is an output of a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(x_\phi)$. The reason will be described later.

<<Process at Step S**24108**>>

The second computing unit **24108**-T uses the input second output information $z_{\phi,2}$, the element ${}_h r_{\phi,2}$, and the second encryption key y(ϕ, 2) to compute $z_{\phi,2}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,2}))^{-1}$ and sets the result of the computation as v_ϕ. The result v_ϕ of the computation is sent to the second power computing unit **21109**-ϕ. Here, $v_\phi=z_{\phi,2}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,2}))^{-1}=f_\phi(x_\phi)^{a(\Phi)}x_{\phi,2}$. That is, $z_{\phi,2}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,2}))^{-1}$ is an output of a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(x_\phi)$. The reason will be described later.

<<Process at Step S**24107**>>

The determining unit **24111**-ϕ determines whether or not there is a pair of u_ϕ' and v_ϕ' that belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ among the pairs (u_1, u_ϕ') stored in the first list storage **21107**-ϕ and the pairs (v_ϕ, v_ϕ') stored in the second list storage **21110**-ϕ. The determining unit **24111**-ϕ of this embodiment determines whether or not there is a pair that satisfies the relation $e_\phi(\mu_{\phi,g1}, c_{\phi,2u})/e_\phi(c_{\phi,1u}, y(\phi,2))=e_\phi(\mu_{\phi,g1}, c_{\phi,2v})/e_\phi(c_{\phi,1v}, y(\phi, 2))$ for $u_1'=(c_{\phi,1u}, c_{\phi,2u})$ and $v_\phi'=(c_{\phi,1v}, c_{\phi,2v})$ (step S**24107**). The reason why the determination as to whether u_ϕ' and v_ϕ' belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ can be made by determining whether u_ϕ' and v_ϕ' satisfy the relation will be described later.

If a pair (v_ϕ, v_ϕ') is not stored in the second list storage **21110**-ϕ, a process at step S**21108** is performed without performing the process at step S**24107**. If there is a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ (if there is a pair of u_ϕ' and v_ϕ' that satisfy the relation given above), the process proceeds to step S**21114**. If there is not a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ the process proceeds to step S**24108**.

<<Process at Step S**24110**>>

The determining unit **24111**-ϕ determines whether there is a pair of u_ϕ' and v_ϕ' that belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ among the pairs (u_ϕ, u_ϕ') stored in the first list storage **21107**-ϕ and the pairs (v_ϕ, v_ϕ') stored in the second list storage **24110**-ϕ. The determining unit **24111**-ϕ of this embodiment determines whether there is a pair that satisfies the relation $e_\phi(\mu_{\phi,g1}, c_{\phi,2u})/e_\phi(c_{\phi,1u}, y(\phi, 2))=e_\phi(\mu_{\phi,g1}, c_{\phi,2v})/e_\phi(c_{\phi,1v}, y(\phi, 2))$ for $u_\phi'=(c_{\phi,1u}, c_{\phi,2u})$ and $v_\phi'=(c_{\phi,1v}, c_{\phi,2v})$ (step S**24110**). If there is a pair of u_ϕ' and v_ϕ' that belong the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21114**. If there is not a pair of u_ϕ' and v_ϕ' that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21111**.

Note that using special groups $G_{\phi,1}$, $G_{\phi,2}$ and the bilinear map e_ϕ can allow only a permitted person to cause the computing apparatus **241**-ϕ to perform the processes of steps S**24107** and S**24110**. Details of this will be described later.

<<Reason Why $z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,1}))^{-1}$ and $z_{\phi,2}(C_{\phi,2}(y(\phi, 2), {}_h r_{\phi,2}))^{-1}$ are Outputs of Randomizable Samplers Having Errors $X_{\phi,1}$ and $X_{\phi,2}$, Respectively, for $f_\phi(x_\phi)$>>

Suppose that an arbitrary element ${}_h r_{\phi,1}$ is fixed, then the relation given below holds for a probability distribution over a probability space of random numbers r(ϕ) in elements $C_{\phi,2}(y(\phi, 2), m_\phi)=(\mu_{\phi,g1}{}^{r(\Phi)}, m_\phi y(\phi, 2)^{r(\Phi)})$:

$$[z_{\varphi,2}(C_{\varphi,2}(y(\varphi, 2), {}_h r_{\varphi,2}))^{-1}] =$$
$$[f_\varphi(x_\varphi^{a(\varphi)}C_{\varphi,2}(y(\varphi, 1), {}_h r_{\varphi,2}))(C_{\varphi,2}(y(\varphi, 2), {}_h r_{\varphi,2}))^{-1}] =$$
$$[f_\varphi(C_{\varphi,2}(y(\varphi, 1), D_{\varphi,1}(s(\varphi, 1), x_\varphi)^{a(\varphi)}{}_h r_{\varphi,2}))(C_{\varphi,2}(y(\varphi, 2), {}_h r_{\varphi,2}))^{-1}] =$$
$$[f_\varphi(C_{\varphi,2}(y(\varphi, 1), D_{\varphi,1}(s(\varphi, 1), x_\varphi)^{a(\varphi)}{}_h r_{\varphi,2}))$$
$$C_{\varphi,2}(y(\varphi, 2), D_{\varphi,1}(s(\varphi, 1), x_\varphi)^{a(\varphi)}{}_h r_{\varphi,2})^{-1}C_{\varphi,2}(y(\varphi, 2),$$
$$D_{\varphi,1}(s(\varphi, 1), x_\varphi)a^{(\varphi)})] = [f_\varphi(C_{\varphi,2}(y(\varphi, 1), {}_h r'_{\varphi,2}))$$
$$C_{\varphi,2}(y(\varphi, 2), {}_h r'_{\varphi,2})^{-1}C_{\varphi,2}(y(\varphi, 2), D_{\varphi,1}(s(\varphi, 1), x_\varphi)^{a(\varphi)})]$$

where $[\psi_1]=[\psi_2]$ means that ψ_1 is equal to ψ_2 as probability distributions over probability spaces of random numbers r. $D_{\phi,1}(s(\phi, 1), x_\phi)$ represents a function for decrypting elements $x_\phi$with the first decryption key s(ϕ, 1) according to a first encryption scheme $ENC_{\phi,1}$. It is assumed that ${}_h r_{\phi,2}'=D_{\phi,1}(s(\phi, 1), x_\phi)^{a(\Phi)}{}_h r_{\phi,2}$.

Therefore, assuming that both of the random number $r(\phi)$ and the uniform random element ${}_{g}r_{\phi}$ on the group G_{ϕ} are probability spaces and a random variable having a value in the group G_{ϕ} is $X_{\phi,2}=f_{\phi}(C_{\phi,2}(y(\phi, 1), {}_{h}r_{\phi,2}'))C_{\phi,2}(y(\phi, 2), {}_{h}r_{\phi,2}')^{-1}$, then the relation given below holds for $x_{\phi,2}\in X_{\phi,2}$:

$$z_{\varphi,2}(C_{\varphi,2}(y(\varphi, 2), {}_{h}r_{\varphi,2}))^{-1} = x_{\varphi,2}C_{\varphi,2}(y(\varphi, 2), D_{\varphi,1}(s(\varphi, 1), x_{\varphi})^{a(\varphi)}) = x_{\varphi,2}f_{\varphi}(x_{\varphi}^{a(\varphi)}) = f_{\varphi}(x_{\varphi})^{a(\varphi)}x_{\varphi 2}$$

Likewise, the relation $z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_{h}r_{\phi,1}))^{-1}=f_{\phi}(x_{\phi})^{b(\phi)}x_{\phi,1}$ holds for $x_{\phi,1}\in X_{\phi,1}$. Therefore, $z_{\phi,1}(C_{\phi,2}(y(\phi, 2), {}_{h}r_{\phi,1}))^{-1}$ and $z_{\phi,2}(C_{\phi,2}(y(\phi, 2), {}_{h}r_{\phi,2}))^{-1}$ are outputs of randomizable samples having errors $X_{\phi,1}$ and $X_{\phi,2}$, respectively, for $f_{\phi}(x_{\phi})$.

<<Why Determining Whether Relation $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2u})/e_{\phi}(c_{\phi,1u}, y(\phi, 2))=e_{\phi}(\mu_{\phi,g1}, c_{\phi,2v})/e_{\phi}(c_{\phi,1v}, y(\phi, 2))$ is Satisfied can Determine Whether u_{ϕ}' and v_{ϕ}' Belong to Class $CL_{\phi}(M_{\phi})$ Corresponding to the Same M_{ϕ}>>

Assume that u_{ϕ}' and v_{ϕ}' belong to a class $CL_{\phi}(M_{\phi})$ corresponding to the same element $M_{\phi}=x_{\phi}$. Then, it is highly probable that the first randomizable sampler has correctly computed $u_{\phi}=f(x_{\phi})^{b(\phi)}$ and that the second randomizable sampler has correctly computed $v_{\phi}=f_{\phi}(x_{\phi})^{a(\phi)}$ (that is, $x_{\phi,1}$ and $x_{\phi,2}$ are identity elements $e_{\phi,g}$ of the group G_{ϕ}). Accordingly, it is highly probable that $u_{\phi}'=(\mu_{\phi,g1}{}^{r''(\phi)}, m_{\phi}y(\phi, 2)^{r''(\phi)})$ and $v_{\phi}'=(\mu_{\phi,g1}{}^{r'''(\phi)}, {}^{m}{}_{\phi}y(\phi, 2)^{r'''(\phi)})$, where $r''(\phi)$ and $r'''(\phi)$ are values that are determined by a pair of a random number component of ElGamal encryption and a natural number selected by the natural number selecting unit **21102**-ϕ. Then, from the properties of the bilinear map e_{ϕ}, it is highly probable for $u_{\phi}'=(c_{\phi,1u}, c_{\phi,2u})$, the following is satisfied:

$$\begin{aligned} &e_{\varphi}(\mu_{\varphi,g1}, c_{\varphi,2u})/e_{\varphi}(c_{\varphi,1u}, y(\varphi, 2)) = \\ &\quad e_{\varphi}\left(\mu_{\varphi,g1}, m_{\varphi}y(\varphi, 2)^{r''(\varphi)}\right)/e_{\varphi}\left(\mu_{\varphi,g1}^{r''(\varphi)}, y(\varphi, 2)\right) = \\ &\quad e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi}y(\varphi, 2))^{r''(\varphi)}/e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2))^{r''(\varphi)} = \\ &\quad e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi})e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2))/e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2)) = e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi}). \end{aligned}$$

For $v_{\phi}'=(c_{\phi,1v}, c_{\phi,2v})$, it is highly probable that the following is satisfied:

$$\begin{aligned} &e_{\varphi}(\mu_{\varphi,g1}, c_{\varphi,2v})/e_{\varphi}(c_{\varphi,1v}, y(\varphi, 2)) = \\ &\quad e_{\varphi}\left(\mu_{\varphi,g1}, m_{\varphi}y(\varphi, 2)^{r'''(\varphi)}\right)/e_{\varphi}\left(\mu_{\varphi,g1}^{r'''(\varphi)}, y(\varphi, 2)\right) = \\ &\quad e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi}y(\varphi, 2))^{r'''(\varphi)}/e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2))^{r'''(\varphi)} = \\ &\quad e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi})e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2))/e_{\varphi}(\mu_{\varphi,g1}, y(\varphi, 2)) = e_{\varphi}(\mu_{\varphi,g1}, m_{\varphi}). \end{aligned}$$

Therefore, if u_{ϕ}' and v_{ϕ}' belong to the class $CL_{\phi}(M_{\phi})$ corresponding to the same element M_{ϕ}, it is highly probable that the relation $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2u})/e_{\phi}(c_{\phi,1u}, y(\phi, 2))=e_{\phi}(\mu_{\phi,g1}, c_{\phi,2v})/e_{\phi}(c_{\phi,1v}, y(\phi, 2))$ is satisfied.

Next, assume that u_{ϕ}' and v_{ϕ}' belong to classes corresponding to different elements. That is, u_{ϕ}' belongs a class $CL_{\phi}(m_{\phi,u})$ corresponding to an element $m_{\phi,u}$ and v_{ϕ}' belongs to a class $CL_{\phi}(m_{\phi, v})$ corresponding to an element $m_{\phi,v}$ $(m_{\phi,v}\neq m_{\phi,u})$. Then, $u_{\phi}'=(\mu_{\phi,g1}{}^{r''(\phi)}, m_{\phi,u}y(\phi, 2)^{r''(\phi)})x_{\phi,1}$ and $v_{\phi}'=(\mu_{\phi,g1}{}^{r'''(\phi)}, m_{\phi,v}y(\phi, 2)^{r'''(\phi)})x_{\phi,2}$. Accordingly, $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2u})/e_{\phi}(c_{\phi,1u}, y(\phi, 2))=e_{\phi}(\mu_{\phi,g1}, m_{\phi,u})x_{\phi,1}$ is satisfied for $u_{\phi}'=(c_{\phi,1u}, c_{\phi,2u})$ and $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2v})/e_{\phi}(c_{\phi,1v}, y(\phi, 2))=e_{\phi}(\mu_{\phi,g1}, m_{\phi,v})x_{\phi,2}$ is satisfied for $v_{\phi}'=(c_{\phi,1v}, c_{\phi,2v})$. Therefore, if u_{ϕ}' and v_{ϕ}' belong to classes corresponding to different elements, it is highly probable that the relation $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2u})/e_{\phi}(c_{\phi,1u}, y(\phi, 2))\neq e_{\phi}(\mu_{\phi,g1}, c_{\phi,2v})/e_{\phi}(c_{\phi,1v}, y(\phi, 2))$ is satisfied.

<<Special Groups $G_{\phi,1}$, $G_{\phi,2}$ and Bilinear Map e_{ϕ}>>

Setting the following restriction on the constructions of groups $G_{\phi,1}$, $G_{\phi,2}$ and the bilinear map e_{ϕ} can allow only a permitted person to cause the computing apparatus **241**-ϕ to perform the processes of steps S**24107** and S**24110**. Details of this will be described below.

In this special example, N_{ϕ} is the composite number of primes ω_{ϕ} and primes τ_{ϕ}, groups $G_{\phi,1}$ and $G_{\phi,2}$ are subgroups consisting of points on a first elliptic curve $E_{\phi,1}$ defined on a factor ring $Z/N_{\phi}Z$ modulo composite number N_{ϕ}, $G_{\phi,1\omega}$ and $G_{\phi,2\omega}$ are subgroups consisting of points on a second elliptic curve $E_{\phi,2}$ defined on a factor ring $Z/\omega_{\phi}Z$ modulo prime ω_{ϕ}, $G_{\phi,1\iota}$ and $G_{\phi,2\iota}$ are subgroups consisting of points on a third elliptic curve $E_{\phi,3}$ defined on a factor ring $Z/\iota_{\phi}Z$ modulo prime ι_{ϕ}, $e_{\phi}(\alpha, \beta)$ is a bilinear map that yields an element of a cyclic group $G_{\phi,T}$ for $(\alpha, \beta)\in G_{\phi,1}\times G_{\phi,2}$, $e_{\phi,\omega}(\alpha_{\omega}, \beta_{\omega})$ is a second bilinear map that yields an element of a cyclic group $G_{\phi,T\omega}$ for $(\alpha_{\omega}, \beta_{\omega})\in G_{\phi,1\omega}\times G_{\phi,2\omega}$, $e_{\phi,\iota}(\alpha_{\iota}, \beta_{\iota})$ is a third bilinear map that yields an element of a cyclic group $G_{\phi,T\iota}$ for $(\alpha_{\iota}, \beta_{\iota})\in G_{\phi,1\iota}\times G_{\phi,2\iota}$, HM_{ϕ} is an isomorphism map that maps a point on the first elliptic curve $E_{\phi,1}$ to a point on the second elliptic curve $E_{\phi,2}$, and a point on the third elliptic curve $E_{\phi,3}$, and HM_{ϕ}^{-1} is the inverse map of the isomorphism map HM_{ϕ}.

In this example, the bilinear map $e_{\phi}(\alpha, \beta)$ is defined on the first elliptic curve $E_{\phi,1}$ defined on the factor ring $Z/N_{\phi}Z$. However, there is not a known method for computing the bilinear map $e_{\phi}(\alpha, \beta)$ defined on an elliptic curve defined on a factor ring in polynomial time nor a method for constructing a bilinear map $e_{\phi}(\alpha, \beta)$ defined on an elliptic curve defined on a factor ring that can be computed in polynomial time (Reference literature 4: Alexander W. Dent and Steven D. Galbraith, "Hidden Pairings and Trapdoor DDH Groups," ANTS 2006, LNCS 4076, pp. 436-451, 2006). In such a setting, the determining unit **24111** cannot determine whether or not the relation $e_{\phi}(\mu_{\phi,g1}, c_{\phi,2u})/e_{\phi}(c_{\phi,1u}, y(\phi, 2))=e_{\phi}(\mu_{\phi,g1}, c_{\phi,2v})/e_{\phi}(c_{\phi,1v}, y(\phi, 2))$ is satisfied by directly computing the bilinear map e_{ϕ} on the factor ring $Z/N_{\phi}Z$.

On the other hand, as for $e_{\phi,\omega}(\alpha_{\omega},\beta_{\omega})$ and $e_{\phi,\iota}(\alpha_{\iota}, \beta_{\iota})$ defined on elliptic curves defined on residual fields $Z/\omega_{\phi}Z$ and $Z/t_{\phi}Z$, there are pairings such as Weil pairing and Tate pairing which can be computed in polynomial time (see Reference literatures 2 and 3, for example). Algorithms for computing such $e_{\phi,\omega}(\alpha_{\omega},\beta_{\omega})$ and $e_{\phi,\iota}(\alpha_{\iota}, \beta_{\iota})$ in polynomial time, such as Miller's algorithm, are well known (Reference literature 5: V. S. Miller, "Short Programs for functions on Curves," 1986, Internet <http://crypto.standford.edu/miller/miller.pdf>). Furthermore, methods for constructing elliptic curves and cyclic groups for efficiently computing such $e_{\phi,\omega}(\alpha_{\omega}, \beta_{\omega})$ and $e_{\phi,\iota}(\alpha_{\iota}, \beta_{\iota})$ are also well known (See, for example, Reference literatures 3 and 6: A Miyaji, M. Nakabayashi, S. Takano, "New explicit conditions of elliptic curve Traces for FR-Reduction," IEICE Trans. Fundamentals, vol. E84-A, no. 05, pp. 1234-1243, May 2001", Reference literature 7: P. S. L. M. Barreto, B. Lynn, M. Scott, "Constructing elliptic curves with prescribed embedding degrees," Proc. SCN 2002, LNCS 2576, pp. 257-267, Springer-Verlag, 2003", Reference literature 8: R. Dupont, A. Enge, F. Morain, "Building curves with arbitrary small MOV degree over finite prime fields," http://eprintiacr.org/2002/094).

It is well known that, based on the Chinese remainder theorem, there is an isomorphic map that maps from a factor

ring $Z/N_\phi Z$ modulo composite number N_ϕ ($N_\phi=\omega_\phi\cdot\iota_\phi$) to the direct product of a residue field $Z/\omega_\phi Z$ and a residue field $Z/\iota_\phi Z$, and that there is an isomorphic map that maps from the direct map of a residue field $Z/\omega_\phi Z$ and a residue field Z/ι_ϕ Z to a factor ring Z/N_ϕ Z (Reference literature 9: Johannes Buchmann "Introduction to Cryptography", Springer Verlag Tokyo, (2001/07), ISBN-10: 4431708669 ISBN-13, pp. 52-56). That is, there are an isomorphic map HM_ϕ that maps a point on the first elliptic curve $E_{\phi,1}$ to a point on the second elliptic curve $E_{\phi,2}$ and a point on the third elliptic curve $E_{\phi,3}$, and its inverse map HM_ϕ^{-1}. To take an example, an isomorphic map that maps an element κ mod N_ϕ of a factor ring $Z/N_\phi Z$ to an element κ mod ω_ϕ of a residue field $Z/\omega_\phi Z$ and an element κ mod ι_ϕ of a residue field $Z/\iota_\phi Z$ can be HM_ϕ and a map that maps an element κ_ω mod ω_ϕ of a residue field $Z/\omega_\phi Z$ and an element κ_ι mod ι_ϕ of a residue field $Z/\iota_\phi Z$ to a an element $\kappa_\omega\iota_\phi\iota_\phi'+\kappa_\iota\omega_\phi\omega_\phi'$ mod N_ϕ of factor ring $Z/N_\phi Z$ can be HM_ϕ^{-1}. Here, ω_ϕ' and t_ϕ' are natural numbers that satisfy $\omega_\phi\omega_\phi'+\iota_\phi\iota_\phi'=1$. Such ω_ϕ' and ι_ϕ' can be easily generated by using the extended Euclidean algorithm. From the relation $\omega_\phi\omega_\phi'+\iota_\phi\iota_\phi'=1$, application of HM_ϕ to $\kappa_\omega\iota_\phi\iota_\phi'+\kappa_\iota\omega_\phi\omega_\phi'$ mod N yields

$$\kappa_\omega\iota_\phi\iota_\phi'+\kappa_\iota\omega_\phi\omega_\phi' \bmod \omega_\phi=\kappa_\omega\iota_\phi\iota_\phi' \bmod \omega_\phi=\kappa_\omega(1-\omega_\phi\omega_\phi') \bmod \omega_\phi=\kappa_\omega \bmod \omega_\phi \in Z/\omega_\phi Z$$

$$\kappa_\omega\iota_\phi\iota_\phi'+\kappa_\iota\omega_\phi\omega_\phi' \bmod \iota_\phi=\kappa_\iota\omega_\phi\omega_\phi' \bmod \iota_\phi=\kappa_\iota(1-\iota_\phi\iota_\phi') \bmod \iota_\phi=\kappa_\iota \bmod \iota_\phi \in Z/\iota_\phi Z$$

Thus, it can be seen that the mapping in this example is between HM_ϕ and HM_ϕ^{-1}.

Therefore, if values resulting from factorization of the composite number N_ϕ into primes are given, that is, values of primes ω_ϕ and ι_ϕ are given, the determining unit **24111** can compute the bilinear map $e_\phi(\alpha, \beta)$ on the first elliptic curve $E_{\phi,1}$ defined on the factor ring $Z/N_\phi Z$ by performing the following process of steps A to D.

(Step A) The determining unit **24111** uses the isomorphic map HM_ϕ to map a point $\alpha\in G_{\phi,1}$ on the first elliptic curve $E_{\phi,1}$ defined on the factor ring $Z/N_\phi Z$ to a point $\theta_\omega(\alpha)\in G_{\phi,1\omega}$ on the second elliptic curve $E_{\phi,2}$ defined on the residue field $Z/\omega_\phi Z$ and a point $\theta_\iota(\alpha)\in G_{\phi,1\iota}$on the third elliptic curve $E_{\phi,3}$ defined on the residue field $Z/\iota_\phi Z$.

(Step B) The determining unit **24111** uses the isomorphic map HM_ϕ to map a point $\beta\in G_{\phi,2}$ on the first elliptic curve $E_{\phi,1}$ defined on the factor ring $Z/N_\phi Z$ to a point $\theta_\omega(\beta)\in G_{\phi,2\omega}$ on the second elliptic curve $E_{\phi,2}$ defined on the residue field $Z/\omega_\phi Z$ and a point $\theta_\iota(\beta)\in G_{\phi,2\iota}$ on the third elliptic curve $E_{\phi,3}$ defined on the residue field $Z/\iota_\phi Z$.

(Step C) The determining unit **24111** obtains $e_{\phi,\omega}(\theta_\omega(\alpha), \theta_\omega(\beta))$ and $e_{\phi,\iota}(\theta_\iota(\alpha), \theta_\iota(\beta))$ on the second elliptic curve $E_{\phi,2}$ and the third elliptic curve $E_{\phi,3}$.

(Step D) The determining unit **24111** applies the inverse map HM_ϕ^{-1} to the obtained results of the computations $e_{\phi,\omega}(\theta_\omega(\alpha), \theta_\omega(\beta))$ and $e_{\phi,\iota}(\theta_\iota(\alpha), \theta_\iota(\beta))$ to obtain a value $e_\phi(\alpha, \beta)$.

Thus, if values of primes ω_ϕ and ι_ϕ are given, the determining unit **24111** can compute $e_\phi(\mu_{\phi,g1}, c_{\phi,2u})$, $e_\phi(c_{\phi,1u}, y(\phi, 2))$, $e_\phi(\mu_{\phi,g1}, c_{\phi,2v})$, and $e_\phi(c_{\phi,1v}, y(\phi, 2))$ by following steps A through D to determine whether or not the relation $e_\phi(\mu_{\phi,g1}, c_{\phi,2u})/e_\phi(e_{\phi,1u}, y(\phi, 2))=e_\phi(\mu_{\phi,g1}, c_{\phi,2v})/e_\phi(c_{\phi,1v}, y(\phi, 2))$ is satisfied.

On the other hand, no method for factorizing a large composite number N_ϕ into primes in polynomial time is known. Therefore, in this setting, the determining unit **24111** to which at least one of the primes ω_ϕ and ι_ϕ is not given cannot determine whether or not the relation $e_\phi(\mu_{\phi,g1}, c_{\phi,2u})/e_\phi(c_{\phi,1u}, y(\phi,2))=e_\phi(\mu_{\phi,g1}, c_{\phi,2v})/e_\phi(c_{\phi,1v}, y(\phi, 2))$ is satisfied.

Using the special groups $G_{\phi,1}$, $G_{\phi,2}$ and the bilinear map e_ϕ described above can allow only a person who knows at least one of the primes ω_ϕ and ι_ϕ to cause the computing apparatus **241**-ϕ to perform the processes of steps **S24107** and **S24110**.

Sixteenth Embodiment

A proxy computing system of a sixteenth embodiment is another example that embodies the first randomizable sampler and the second randomizable sampler described above. Specifically, the proxy computing system embodies an example of the first randomizable sampler and the second randomizable sampler in which H_ϕ is the direct product $G_\phi\times G_\phi$ of groups G_ϕ which are cyclic groups, a function $f_\phi(x_\phi)$ is a decryption function of the ElGamal encryption, that is, $f_\phi(c_{\phi,1}, c_{\phi,2})=c_{\phi,1}c_{\phi,2}^{-s(\phi)}$ *for an element* $x_\phi=(c_{\phi,1}, c_{\phi,2})$ which is a ciphertext and a decryption key $s(\phi)$. The following description will focus on differences from the twelfth embodiment and description of commonalities with the twelfth embodiment will be omitted.

As illustrated in FIG. **31**, the proxy computing system **205** of the sixteenth embodiment includes a computing apparatus **251**-ϕ in place of the computing apparatus **211**-ϕ and a capability providing apparatus **252** in place of the capability providing apparatus **212**.

As illustrated in FIG. **32**, the computing apparatus **251**-ϕ of the sixteenth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **25104**-ϕ, a first computing unit **25105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **25108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **21111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ. As illustrated in FIG. **36**, the input information providing unit **25104**-ϕ of this embodiment includes, for example, a fourth random number generator **25104***a*-ϕ, a fifth random number generator **25104***b*-ϕ, a first input information computing unit **25104***c*-ϕ, a sixth random number generator **25104***d*-ϕ, a seventh random number generator **25104***e*-ϕ, and a second input information computing unit **25104***f*-ϕ. The first input information computing unit **25104***c*-ϕ includes, for example, a fourth input information computing unit **25104***ca*-ϕ and a fifth input information computing unit **25104***cb*-ϕ. The second input information computing unit **25104***f*-ϕ includes, for example, a sixth input information computing unit **25104***fa*-ϕ and a seventh input information computing unit **25104***fb*-ϕ.

As illustrated in FIG. **33**, the capability providing apparatus **252** of the sixteenth embodiment includes, for example, a first output information computing unit **25201**, the second output information computing unit **25202**, and a controller **21205**.

<Processes>

Processes of this embodiment will be described below. In the sixteenth embodiment, it is assumed that a group $H_\phi=G_\phi\times G_\phi$, an element $x_\phi=c_{\phi,1}, c_{\phi,2})\in H_\phi$, $f_\phi(c_{\phi,1}, c_{\phi,2})$ is a homomorphic function, a generator of the group G_ϕ is $\mu_{\phi,g}$, the order of the group G_ϕ is $K_{\phi,G}$, a pair of a ciphertext $(V_\phi, W_\phi)\in H_\phi$ and a decrypted text $f_\phi(V_\phi, W_\phi)=Y_\phi\in G_\phi$ decrypted from the ciphertext for the same decryption key $s(\phi)$ is preset in the computing apparatus **251**-ϕ and the capability providing apparatus **252**, and the computing apparatus **251**-ϕ and the capability providing apparatus **252** can use the pair.

As illustrated in FIGS. **37** and **38**, a process of the sixteenth embodiment is the same as the process of the

twelfth embodiment except that steps **S21103** through **S21105**, **S21108**, and **S21200** through **S21203** of the twelfth embodiment are replaced with steps **S25103** through **S25105**, **S25108**, and **S25200** through **S25203**, respectively. In the following, only processes at steps **S25103** through **S25105**, **S25108**, and **S25200** through **S25203** will be described.

<<Process at Step S25103>>

The input information providing unit **25104**-ϕ of the computing apparatus **251**-ϕ (FIG. **32**) generates and outputs first input information $\tau_{\phi,1}$ corresponding to an input element $x_\phi=(c_{\phi,1}, c_{\phi,2})$ and second input information $\tau_{\phi,2}$ corresponding to the input element $x_\phi=(c_{\phi,1},c_{\phi,2})$ (step **S25103** of FIG. **37**). A process at step **S25103** of this embodiment will be described below with reference to FIG. **41**.

The fourth random number generator **25104***a*-ϕ (FIG. **36**) generates a uniform random number r(ϕ, 4) that is a natural number greater than or equal to 0 and less than $K_{\phi,G}$. The generated random number r(ϕ, 4) is sent to the fourth input information computing unit **25104***ca*-ϕ, the fifth input information computing unit **25104***cb*-ϕ, and the first computing unit **25105**-ϕ (step **S25103***a*). The fifth random number generator **25104***b*-ϕ generates a uniform random number r(ϕ, 5) that is a natural number greater than or equal to 0 and less than $K_{\phi,G}$. The generated random number r(ϕ, 5) is sent to the fifth input information computing unit **25104***cb*-ϕ and the first computing unit **25105**-ϕ (step **S25103***b*).

The fourth input information computing unit **25104***ca*-ϕ uses a natural number b(ϕ) selected by the natural number selecting unit **21102**-ϕ, $c_{\phi,2}$ included in the element x_ϕ, and the random number r(ϕ, 4) to compute fourth input information $c_{\phi,2}{}^{b(\phi)}W^{r(\phi,4)}$ (step **S25103***c*). The fifth input information computing unit **25104***cb*-ϕ uses the natural number b(ϕ) selected by the natural number selecting unit **21102**-ϕ, $c_{\phi,1}$ included in the element x_ϕ, and random numbers r(ϕ, 4) and r(ϕ5) to compute fifth input information $c_{\phi,1}{}^{b(\phi)}V^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)}$ (step **S25103***d*).

The sixth random number generator **25104***d*-ϕ generates a uniform random number r(ϕ, 6) that is a natural number greater than or equal to 0 and less than $K_{\phi,G}$. The generated random number r(ϕ, 6) is sent to the sixth input information computing unit **25104***fa*-ϕ, the seventh input information computing unit **251041***b*-ϕ, and the second computing unit **25108**-ϕ (step **S25103***e*). The seventh random number generator **25104***e*-ϕ generates a uniform random number r(ϕ, 7) that is a natural number greater than or equal to 0 and less than $K_{\phi,G}$. The generated random number r(ϕ, 7) is sent to the seventh input information computing unit **25104***fb*-ϕ and the second computing unit **25108**-ϕ (step **S251030**.

The sixth input information computing unit **25104***fa*-ϕ uses a natural number a(ϕ) selected by the natural number selecting unit **21102**-ϕ, $c_{\phi,2}$ included in the element x_ϕ, and the random number r(ϕ, 6) to compute sixth input information $c_{\phi,2}{}^{a(\phi)}W^{r(\phi,6)}$(step **S25103***g*). The seventh input information computing unit **25104***fb*-ϕ uses the natural number a(ϕ) selected by the natural number selecting unit **21102**-ϕ, $c_{\phi,1}$ included in the element x_ϕ, and the random numbers r(ϕ, 6) and r(ϕ, 7) to compute seventh input information $c_{\phi,1}{}^{a(\phi)}V^{r(\phi,6)}\mu_g{}^{r(\phi,7)}$(step **S25103***h*).

The first input information computing unit **25104***c*-ϕ outputs the fourth input information $c_{\phi,2}{}^{b(\phi)}W^{r(\phi,4)}$ and the fifth input information $c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)}$ generated as described above as first input information $\tau_{\phi,1}=(c_{\phi,2}{}^{b(\phi)}W_\phi{}^{r(\phi,4)}, c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)})$. The second input information computing unit **25104***f*-ϕ outputs the sixth input information $c_{\phi,2}{}^{a(\phi)}W^{r(\phi,6)}$ and the seventh input information $c_{\phi,1}{}^{a(\phi)}V_\phi{}^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)}$ generated as described above as second input information $\tau_{\phi,2}=(c_{\phi,2}{}^{a(\phi)}W_\phi{}^{r(\phi,6)}, c_{\phi,1}{}^{a(\phi)}V_\phi{}^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)})$ (step **S25103***i*).

<<Processes at Steps S25200 through S25203>>

As illustrated in FIG. **38**, first, the first input information $\tau_{\phi,1}=(c_{\phi,2}{}^{b(\phi)}W_\phi{}^{r(\phi,4)}, c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)})$ is input in the first output information computing unit **25201** of the capability providing apparatus **252** (FIG. **33**) and the second input information $\tau_{\phi,2}=(c_{\phi,2}{}^{a(\phi)}W^{r(\phi,6)}, c_{\phi,1}{}^{a(\phi)}V^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)})$ is input in the second output information computing unit **25202** (step **S25200**).

The first output information computing unit **25201** uses the first input information $\tau_{\phi,1}=(c_{\phi,2}{}^{b(\phi)}W_\phi{}^{r(\phi,4)}, c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)})$ and the decryption key s(ϕ) to correctly compute $f_\phi(c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)}, c_{\phi,2}{}^{b(\phi)}W_\phi{}^{r(\phi,4)})$ with a probability greater than a certain probability and sets the result of the computation as first output information $z_{\phi,1}$. The result of the computation may or may not be correct. That is, the result of the computation by the first output information computing unit **25201** may or may not be $f_\phi(c_{\phi,1}{}^{b(\phi)}V_\phi{}^{r(\phi,4)}\mu_{\phi,g}{}^{r(\phi,5)}, c_{\phi,2}{}^{b(\phi)}W_\phi{}^{r(\phi,4)})$(step **S25201**).

The second output information computing unit **25202** can use the second input information $\tau_{\phi,2}=(c_{\phi,2}{}^{a(\phi)}W_\phi{}^{r(\phi,6)}, c_{\phi,1}{}^{a(\phi)}V_\phi{}^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)})$ and the decryption key s(ϕ) to correctly compute $f_\phi(c_{\phi,1}{}^{a(\phi)}V_\phi{}^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)}, c_{\phi,2}{}^{a(\phi)}W_\phi{}^{r(\phi,6)})$ with a probability greater than a certain probability and sets the result of the computation as second output information $z_{\phi,2}$. The result of the computation may or may not be correct. That is, the result of the computation by the second output information computing unit **25202** may or may not be $f_\phi(c_{\phi,1}{}^{a(\phi)}V_\phi{}^{r(\phi,6)}\mu_{\phi,g}{}^{r(\phi,7)}, c_{\phi,2}{}^{a(\phi)}W_\phi{}^{r(\phi,6)})$ (step **S25202**).

The first output information computing unit **25201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **25202** outputs the second output information $z_{\phi,2}$ (step **S25203**).

<<Processes at Steps S25104 and S25105>>

Returning to FIG. **37**, the first output information $z_{\phi,1}$ is input in the first computing unit **25105**-ϕ of the computing apparatus **251**-ϕ (FIG. **32**) and the second output information $z_{\phi,2}$ is input in the second computing unit **25108**-ϕ (step **S25104**).

The first computing unit **25105**-ϕ uses the input first output information $z_{\iota,1}$ and the random numbers r(ϕ, 4) and r(ϕ, 5) to compute $z_{\phi,1}Y^{-r(\phi,4)}\mu_{\phi,g}{}^{-r(\phi,5)}$ and sets the result of the computation as μ_ϕ (step **S25105**). The result u_ϕ of the computation is sent to the first power computing unit **21106**-ϕ. Here, $u_\phi=z_{\phi,4}Y_\phi{}^{-r(\phi,4)}\mu_{\phi,g}{}^{-r(\phi,5)}=f_\phi(c^{\phi,1}, c_{\phi,2})^{b(\phi)}x_{\phi,1}$. That is, $z_{\phi,4}Y_\phi{}^{-r(\phi,4)}\mu_{\phi,g}{}^{-r(\phi,5)}$ is an output of a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(c_{\phi,1}, c_{\phi,2})$. The reason will be described later.

<<Process at Step S25108>>

The second computing unit **25108**-ϕ uses the input second output information $z_{\phi,2}$ and the random numbers r(ϕ,6) and r(ϕ,7) to compute $z_{\phi,2}Y_\phi{}^{-r(\phi,6)}{}_{\mu\phi,g}{}^{-r(\phi,7)}$ and sets the result of the computation as v_ϕ. The result v_ϕ of the computation is sent to the second power computing unit **21109**-ϕ. Here, $v_\phi=z_{\phi,5}Y_\phi{}^{-r(\phi,6)}\mu_{\phi,g}{}^{-r(\phi,7)}=f_\phi(c_{\phi,1}, c_{\phi,2})^{a(\phi)}x_{\phi,2}$. That is, $z_{\phi,5}Y_\phi{}^{-r(\phi,6)}\mu_{\phi,g}{}^{-r(\phi,7)}$ is an output of a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(c_{\phi,1}, c_{\phi,2})$. The reason will be described later.

<<Reason Why $z_{\phi,4}Y_\phi{}^{-r(\phi,4)}\mu_{\phi,g}{}^{-r(\phi,5)}$ *and* $z_{\phi,5}Y_\phi{}^{-r(\phi,6)}\mu_{\phi,g}{}^{-r(\phi,7)}$ are Outputs of Randomizable Samplers Having Errors $X_{\phi,1}$ and $X_{\phi,2}$, Respectively, for $f_\phi(c_{\phi,1}, c_{\phi,2})$>>

Let c be a natural number, R_1, R_2, R_1' and R_2' be random numbers, and $B(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})$ be the result of computation performed by the capability providing apparatus **252** using $c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}$ and $c_{\phi,2}{}^cW_\phi{}^{R1}$. That is, the first

output information computing unit **25201** and the second output information computing unit **25202** return $z_\phi=B(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})$ as the results of computations to the computing apparatus **251**-ϕ. Furthermore, a random variable X_ϕ having a value in a group G_ϕ is defined as $X_\phi=B(V_\phi{}^{R1'}\mu_{\phi,g}{}^{R2'}, W_\phi{}^{R1'})f_\phi(V_\phi{}^{R1'}\mu_{\phi,g}{}^{R2'}, W_\phi{}^{R1'})^{-1}$.

Here, $z_\phi Y_\phi{}^{-R1}\mu_{\phi,g}{}^{-R2}=B(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})Y_\phi{}^{-R1}\mu_{\phi,g}{}^{-R2}=X_\phi f_\phi(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})Y_\phi{}^{-R1}\mu\phi, g^{-R2}=X_\phi f_\phi(c_{\phi,1}, c_{\phi,2})^cT\phi(V_\phi, W_\phi)^{R1}f_\phi(\mu_{\phi,g}, e_{\phi,g})^{R2}Y_\phi{}^{-R1}\mu_{\phi,g}{}^{-R2}=X_\phi f_\phi(c_{\phi,1}, c_{\phi,2})^cY_\phi{}^{R1}\mu_{\phi,g}{}^{R2}Y_\phi{}^{-R1}\mu_{\phi,g}{}^{-R2}=f_\phi(c_{\phi,1}, c_{\phi,2})^cX_\phi$. That is, $z_\phi Y_\phi{}^{-R1}\mu_{\phi,g}{}^{-R2}$ is an output of a randomizable sampler having an error X_ϕ for $f_\phi(x_\phi)$. Note that $e_{\phi,g}$ is an identity element of the group G_ϕ.

The expansion of formula given above uses the properties such that $X_\phi=B(V_\phi{}^{R1'}\mu_{\phi,g}{}^{R2'}, W_\phi{}^{R1'})f_\phi(V_\phi{}^{R1'}\mu_{\phi,g}{}^{R2'}, W_\phi{}^{R1'})^{-1}=B(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi2}{}^cW_\phi{}^{R1})f_\phi(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})$ and that $B(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})=X_\phi f_\phi(c_{\phi,1}{}^cV_\phi{}^{R1}\mu_{\phi,g}{}^{R2}, c_{\phi,2}{}^cW_\phi{}^{R1})$. The properties are based on the fact that R_1, R_2, $R_{1'}$ and $R_{2'}$ are random numbers.

Therefore, considering that a(ϕ) and b(ϕ) are natural numbers and r(ϕ,4), r(ϕ, 5), r(ϕ, 6), and r(ϕ, 7) are random numbers, $z_{\phi,4}Y_\phi{}^{-r(\phi,4)}\mu_{\phi,g}{}^{-r(\phi,5)}$ and $z_{\phi,5}Y_\phi{}^{-r(\phi,6)}\mu_{\phi,g}{}^{-r(\phi,7)}$ are, likewise, outputs of randomizable samplers having errors $X_{\phi,1}$ and $X_{\phi,2}$, respectively, for $f_\phi(c_{\phi,1}, c_{\phi,2})$.

Seventeenth Embodiment

A proxy computing system of a seventeenth embodiment is another example that embodies the first randomizable sampler and the second randomizable sampler described above. Specifically, the proxy computing system embodies an example of a first randomizable sampler and a second randomizable sampler in which a group H_ϕ is the direct product $H_{1,\phi}\times H_{2,\phi}$ of cyclic groups $H_{1,\phi}$ and $H_{2,\phi}$, a generator of the cyclic group $H_{1,\phi}$ is $\eta_{1,\phi}$, a generator of the cyclic group $H_{2,\phi}$ is $\eta_{2,\phi}$, f_ϕ is a bilinear map that maps a pair of an element of the cyclic group $H_{1,\phi}$ and an element of the cyclic group $H_{2,\phi}$ to an element of a cyclic group G_ϕ, an element x_ϕ of a cyclic group H_ϕ is a pair of an element $\lambda_{1,\phi}$ of the cyclic group $H_{1,\phi}$ and an element $\lambda_{2,\phi}$ of the cyclic group $H_{2,\phi}$, and $\Omega_\phi=f_\phi(\eta_{1,\phi}, \eta_{2,\phi})$. Examples of the bilinear map f_ϕ include functions and algorithms for computing pairings such as Weil pairing and Tate pairing. The following description will focus on differences from the twelfth embodiment and repeated description of commonalities with the twelfth embodiment will be omitted.

As illustrated in FIG. **31**, the proxy computing system **207** of the seventeenth embodiment includes a computing apparatus **271**-ϕ in place of the computing apparatus **211**-ϕ and a capability providing apparatus **272** in place of the capability providing apparatus **212**.

As illustrated in FIG. **32**, the computing apparatus **271**-ϕ of the seventeenth embodiment includes, for example, a natural number storage **21101**-ϕ, a natural number selecting unit **21102**-ϕ, an integer computing unit **21103**-ϕ, an input information providing unit **27104**-ϕ, a first computing unit **27105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **27108**-ϕ, a second power computing unit **21109**-ϕ, a second list storage **21110**-ϕ, a determining unit **27111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ. As illustrated in FIG. **42**, the input information providing unit **27104**-ϕ of this embodiment includes, for example, a first random number generator **27104***a*-ϕ, a second random number generator **27104***c*-ϕ, a first input information computing unit **27104***b*-ϕ, and a second input information computing unit **27104***c***1**-ϕ.

As illustrated in FIG. **33**, the capability providing apparatus **272** of the seventeenth embodiment includes, for example, a first output information computing unit **27201**, the second output information computing unit **27202**, and a controller **21205**.

<Processes>

Processes of this embodiment will be described below. In the seventeenth embodiment, it is assumed that a group H_ϕ is the direct product $H_{1,\phi}\times H_{2,\phi}$ of cyclic groups $H_{1,\phi}$ and $H_{2,\phi}$, a generator of the cyclic group $H_{1,\phi}$ is $\eta_{1,\phi}$, a generator of the cyclic group $H_{2,\phi}$ is $\eta_{2,\phi}$, f_ϕ is a bilinear map that maps a pair of an element of the cyclic group $H_{1,\phi}$ and an element of the cyclic group $H_{2,\phi}$ to an element of a cyclic group G_ϕ, and an element x_ϕ of a group H_ϕ is a pair of an element $\lambda_{1,\phi}$ of the cyclic group $H_{1,\phi}$ and an element $\lambda_{2,\phi}$ of the cyclic group $H_{2,\phi}$,$\Omega_\phi=f_\phi(\eta_{1,\phi}, \eta_{2,\phi})$. Here, $\Omega_\phi=f_\phi(\eta_{1,\phi}, \eta_{2,\phi})$ is computed beforehand.

As illustrated in FIGS. **37** and **38**, a process of the seventeenth embodiment is the same as the process of the twelfth embodiment except that steps **S21103** through **S21105**, **S21107**, **S21108**, **S21110**, and **S21200** through **S21203** of the twelfth embodiment are replaced with steps **S27103** through **S27105**, **S27107**, **S27108**, **S27110**, and **S27200** through **S27203**, respectively. In the following, only processes at steps **S27103** through **S27105**, **S27107**, **S27108**, **S27110** and **S27200** through **S27203** will be described.

<<Process at Step S27103>>

The input information providing unit **27104**-ϕ of the computing apparatus **271**-ϕ (FIG. **32**) generates and outputs first input information $\tau_{\phi,1}$ corresponding to an input pair $x_\phi=(\lambda_{1,\phi}, \lambda_{2,\phi})$ of an element $\lambda_{\iota,\phi}$ of the cyclic group $H_{1,\phi}$ and an element $\lambda_{2,\phi}$ of the cyclic group $H_{2,\phi}$ and second input information $\tau_{\phi,2}$ corresponding to $x_\phi=(\lambda_{1,\phi}, \lambda_{2,\phi})$ (step **S27103** of FIG. **37**). A process at step **S27103** of this embodiment will be described below with reference to FIG. **43**.

The first random number generator **27104***a*-ϕ (FIG. **42**) generates uniform random numbers r(ϕ, 11), r(ϕ, 12), r(ϕ, 13), r(ϕ,14), r(ϕ, 15), and r(ϕ, 16) that are natural numbers greater than or equal to 0 and less than or equal to $2^{\mu(k)+k}$. Here, μ(k) represents a function value of a security parameter k. The generated random numbers r(ϕ, 11), r(ϕ, 12), r(ϕ, 13), r(ϕ,14), r(ϕ, 15), and r(ϕ, 16) are sent to the first input information computing unit **27104***b*-ϕ and the first computing unit **27105**-ϕ (step **S27103***a*).

The first input information computing unit **27104***b*-ϕ uses a natural number b(ϕ) selected by the natural number selecting unit **21102**-ϕ, the input value $x_\phi=(\lambda_{1,\phi}, \lambda_{2,\phi})$, generators $\eta_{1,\phi}$ and $\eta_{2,\phi}$, and random numbers r(ϕ, 11), r(ϕ, 12), r(ϕ, 13), r(ϕ,14), r(ϕ, 15), and r(ϕ, 16) to compute $(\lambda_{1,\phi}\eta_{1,\phi}{}^{r(\phi,11)\cdot r(\phi,12)}, \lambda_{2,\phi}{}^{b(\phi)}\eta_{2,\phi}{}^{r(\phi,13)\cdot(\phi,14)})$, $(\eta_{1,\phi}{}^{r(\phi,11)}, \eta_{2,\phi}{}^{-b(\phi)\cdot r(\phi,12)}\eta_{2,\phi}{}^{r(\phi,15)})$ and $(\lambda_{1,\phi}{}^{-r(\phi,14)}\eta_{1,\phi}{}^{r(\phi,16)}, \eta_{2,\phi}{}^{r(\phi,13)})$ as first input information $\tau_{\phi,1}$ (steps **S27103***b* through **S27103***d*).

The second random number generator **27104***c*-ϕ (FIG. **42**) generates uniform random numbers r(ϕ, 21), r(ϕ, 22), r(ϕ, 23), r(ϕ,24), r(ϕ, 25) and r(ϕ, 26) that are natural numbers greater than or equal to 0 and less than or equal to $2^{\mu(k)+k}$. The generated random numbers r(ϕ, 21), r(ϕ, 22), r(ϕ, 23), r(ϕ,24), r(ϕ, 25) and r(ϕ, 26) are sent to the second input information computing unit **27104***d*-ϕ and the second computing unit **27108**-ϕ (step **S27103***e*).

The second input information computing unit **27104***d*-ϕ uses a natural number a(ϕ) selected by the natural number selecting unit **21102**-ϕ, the input value $x_\phi=(\lambda_{1,\phi}, \lambda_{2,\phi})$, the generators $\eta_{1,\phi}$ and $\eta_{2,\phi}$, and the random numbers r(ϕ, 21),

r(ϕ, 22), r(ϕ, 23), r(ϕ,24), r(ϕ, 25) and r(ϕ, 26) to compute $(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,21)\cdot r(\phi,22)}, \lambda_{2,\phi}^{a(\phi)}\eta_{2,\phi}^{r(\phi,23)\cdot(\phi,24)})$, $(\eta_{1,\phi}^{r(\phi,21)}, \lambda_{2,\phi}^{-a(\phi)\cdot r(\phi,22)}\eta_{2,\phi}^{r(\phi,25)})$ and $(\lambda_{1,\phi}^{-r(\phi,24)}\eta_{1,\phi}^{r(\phi,26)}, \eta_{2,\phi}^{r(\phi,23)})$ as second input information $\tau_{\phi,2}$ (steps **S27103***f* through **S27103***h*).

The first input information computing unit **27104***b*-ϕ outputs $(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,11)\cdot r(\phi,12)}, \lambda_{2,\phi}^{b(\phi)}\eta_{2,\phi}^{r(\phi,13)\cdot(\phi,14)})$, $(\eta_{1,\phi}^{r(\phi,11)}, \lambda_{2,\phi}^{-b(\phi)\cdot r(\phi,12)}\eta_{2,\phi}^{r(\phi,15)})$, and $(\lambda_{1,\phi}^{-r(\phi,14)}\eta_{1,\phi}^{r(\phi,16)}, \eta_{2,\phi}^{r(\phi,13)})$ as the first input information $\tau_{\phi,1}$. The second input information computing unit **27104***d*-ϕ outputs $(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,21)\cdot r(\phi,22)}, \lambda_{2,\phi}^{a(\phi)}\eta_{2,\phi}^{r(\phi,23)\cdot r(\phi,24)})$, $(\eta_{1,\phi}^{r(\phi,21)}, \lambda_{2,\phi}^{-a(\phi)\cdot r(\phi,22)}\eta_{2,\phi}^{r(\phi,25)})$ and $(\lambda_{1,\phi}^{-r(\phi,24)}\eta_{1,\phi}^{r(\phi,26)}, \eta_{2,\phi}^{r(\phi,23)})$ as the second input information $\tau_{\phi,2}$ (step **S27103***i*).

<<Processes at Steps **S27200** through **S27203**>>

As illustrated in FIG. **38**, $(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,11)\cdot r(\phi,12)}, \lambda_{2,\phi}^{b(\phi)}\eta_{2,\phi}^{r(\phi,13)\cdot(\phi,14)})$, $(\eta_{1,\phi}^{r(\phi,11)}, \lambda_{2,\phi}^{-b(\phi)\cdot r(\phi,12)}\eta_{2,\phi}^{r(\phi,15)})$, and $(\lambda_{1,\phi}^{-r(\phi,14)}\eta_{1,\phi}^{r(\phi,16)}, \eta_{2,\phi}^{r(\phi,13)})$, which are the first input information $\tau_{\phi,1}$, are input in the first output information computing unit **27201** of the capability providing apparatus **272** (FIG. **33**). $(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,21)\cdot r(\phi,22)}, \lambda_{2,\phi}^{a(\phi)}\eta_{2,\phi}^{r(\phi,23)\cdot(\phi,24)})$, $(\eta_{1,\phi}^{r(\phi,21)}, \lambda_{2,\phi}^{-a(\phi)\cdot r(\phi,22)}\eta_{2,\phi}^{r(\phi,25)})$ and $(\lambda_{1,\phi}^{-r(\phi,24)}\eta_{1,\phi}^{r(\phi,26)}, \eta_{2,\phi}^{r(\phi,23)})$, which are the second input information $\tau_{\phi,2}$, are input in the second output information computing unit **27202** (step **S27200**).

The first output information computing unit **27201** uses the first input information $\tau_{\phi,1}$ to correctly compute $f_\phi(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,11)\cdot r(\phi,12)}, \lambda_{2,\phi}^{b(\phi)}\eta_{2,\phi}^{r(\phi,13)\cdot(\phi,14)})$, $f_\phi(\eta_{1,\phi}^{r(\phi,11)}, \lambda_{2,\phi}^{-b(\phi)\cdot r(\phi,12)}\eta_{2,\phi}^{r(\phi,15)})$, and $f_\phi(\lambda_{1,\phi}^{-r(\phi,14)}\eta_{1,\phi}^{r(\phi,16)}, \eta_{2,\phi}^{r(\phi,13)})$ with a probability greater than a certain probability and sets the obtained results of the computations, $z_{\phi,1,1}$, $z_{\phi,1,2}$ and $z_{\phi,1,3}$, as first output information $z_{\phi,1}$. The results of the computations may or may not be correct (step **S27201**).

The second output information computing unit **27202** uses the second input information $\tau_{\phi,2}$ to correctly compute $f_\phi(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,21)\cdot r(\phi,22)}, \lambda_{2,\phi}^{a(\phi)}\eta_{2,\phi}^{r(\phi,23)\cdot(\phi,24)})$, $f_\phi(\eta_{1,\phi}^{r(\phi,21)}, \lambda_{2,\phi}^{-a(\phi)\cdot r(\phi,22)}\eta_{2,\phi}^{r(\phi,25)})$ and $f_\phi(\lambda_{1,\phi}^{-r(\phi,24)}\eta_{1,\phi}^{r(\phi,26)}, \eta_{2,\phi}^{r(\phi,23)})$ with a probability greater than a certain probability and sets the obtained results of the computations, $z_{\phi,2,1}$, $z_{\phi,2,2}$ and $z_{\phi,2,3}$, as second output information $z_{\phi,2}$. The results of the computations may or may not be correct (step **S27202**). The first output information computing unit **27201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **27202** outputs the second output information $z_{\phi,2}$ (step **S27203**).

<Processes at Steps **S27104** and **S27105**>>

Returning to FIG. **37**, the first output information $z_{\phi,1}$ is input in the first computing unit **27105**-ϕ of the computing apparatus **271**-ϕ (FIG. **32**) and the second output information $z_{\phi,2}$ is input in the second computing unit **27108**-ϕ (step **S27104**).

The first computing unit **27105**-ϕ uses the input first output information $z_{\phi,1}=(z_{\phi,1,1}, z_{\phi,1,2}, z_{\phi,1,3})$ and the random numbers r(ϕ, 11), r(ϕ, 12), r(ϕ, 13), r(ϕ,14), r(ϕ, 15), and r(ϕ, 16) to compute $u_\phi=z_{\phi,1,1}z_{\phi,1,2}z_{\phi,1,3}\Omega_\phi^{-r(\phi,11)\cdot r(\phi,12)\cdot r(\phi,13)\cdot r(\phi,14)\cdot r(\phi,11)\cdot r(\phi,15)\cdot r(\phi,13)\cdot r(\phi,16)}$ to obtain the result u_ϕ of the computation (step **S27105**). The result u_ϕ of the computation is sent to the first power computing unit **21106**-ϕ. Here, $u_\phi=f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})^{b(\phi)}x_{\phi,1}$. That is, $z_{\phi,1,1}z_{\phi,1,2}z_{\phi,1,3}\Omega_\phi^{-r(\phi,11)\cdot r(\phi,12)\cdot r(\phi,13)\cdot r(\phi,14)\cdot r(\phi,11)\cdot r(\phi,15)\cdot r(\phi,13)\cdot r(\phi,16)}$ is an output of a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})$. The reason will be described later.

<<Process at Step **S27108**>>

The second computing unit **27108**-ϕ uses the input second output information $z_{\phi,2}=(z_{\phi,2,1}, z_{\phi,2,2}, z_{\phi,2,3})$ and the random numbers r(ϕ, 21), r(ϕ, 22), r(ϕ, 23), r(ϕ, 24), r(ϕ, 25), and r(ϕ, 26) to compute $v_\phi=z_{\phi,2,1}z_{\phi,2,2}z_{\phi,2,3}\Omega_\phi^{-r(\phi,21)\cdot r(\phi,22)\cdot r(\phi,23)\cdot r(\phi,24)\cdot r(\phi,21)\cdot r(\phi,25)\cdot r(\phi,23)\cdot r(\phi,26)}$ to obtain the result v_ϕ of the computation. The result v_ϕ of the computation is sent to the second power computing unit **21109**-ϕ. Here, $v_\phi=f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})^{a(\phi)}x_{\phi,2}$. That is, $z_{\phi,2,1}z_{\phi,2,2}z_{\phi,2,3}\Omega_\phi^{-r(\phi,21)\cdot r(\phi,22)\cdot r(\phi,23)\cdot r(\phi,24)\cdot r(\phi,21)\cdot r(\phi,25)\cdot r(\phi,23)\cdot r(\phi,26)}$ is an output of a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})$. The reason will be described later.

<<Processes at Steps **S27107** and **S27110**>>

At steps **S27107** and **S27110**, the determining unit **27111**-ϕ determines whether $u_\phi'=v_\phi'$. If it is determined at step **S27107** that $u_\phi'=v_\phi'$, the process proceeds to step **S21114**; otherwise, the process proceeds to step **S27108**. If it is determined at step **S27110** that $u_\phi'=v_\phi'$, the process proceeds to step **S21114**; otherwise, the process proceeds to step **S21111**.

<<Reason why $z_{\phi,1,1}z_{\phi,1,2}z_{\phi,1,3}\Omega_\phi^{-r(\phi,11)\cdot r(\phi,12)\cdot r(\phi,13)\cdot r(\phi,14)\cdot r(\phi,11)\cdot r(\phi,15)-r(\phi,13)\cdot r(\phi,16)}$ and $z_{\phi,2,1}z_{\phi,2,2}z_{\phi,2,3}\Omega_\phi^{-r(m,21)\cdot r(\phi,22)\cdot r(\phi,23)\cdot r(\phi,24)\cdot r(\phi,21)\cdot r(\phi,25)\cdot r(\phi,23)\cdot r(\phi,26)}$ are output of randomizable samplers having errors $X_{\phi,1}$, and $X_{\phi,2}$, respectively, for $f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})$>>

Because of the bilinearity of f_ϕ, the following relation holds for v_ϕ.

$$v_\varphi f_\varphi(\lambda_{1,\varphi}, \lambda_{2,\varphi})^{-a(\varphi)} = z_{\varphi,2,1}z_{\varphi,2,2}z_{\varphi,2,3}$$

$$\Omega_\varphi^{-r(\varphi,21)\cdot r(\varphi,22)\cdot r(\varphi,23)\cdot r(\varphi,24)-r(\varphi,21)\cdot r(\varphi,25)-r(\varphi,23)\cdot r(\varphi,26)} f_\varphi(\lambda_{1,\varphi}, \lambda_{2,\varphi})^{-a(\varphi)} =$$

$$z_{\varphi,2,1}z_{\varphi,2,2}z_{\varphi,2,3} f_\varphi\left(\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,21)\cdot r(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{r(\varphi,23)\cdot r(\varphi,24)}\right)^{-1}$$

$$f_\varphi\left(\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,21)\cdot(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{(\varphi,23)\cdot r(\varphi,24)}\right)$$

$$\Omega_\varphi^{-r(\varphi,21)\cdot r(\varphi,22)\cdot r(\varphi,23)\cdot r(\varphi,24)-r(\varphi,21)\cdot r(\varphi,25)-r(\varphi,23)\cdot r(\varphi,26)} f_\varphi(\lambda_{1,\varphi}, \lambda_{2,\varphi})^{-a(\varphi)} =$$

$$z'_{\varphi,2,1}z_{\varphi,2,2}z_{\varphi,2,3} f_\varphi\left(\lambda_{1,\varphi}\eta_{1,\varphi}{}^{r(\varphi,21)\cdot r(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{r(\varphi,23)\cdot r(\varphi,24)}\right)$$

$$\Omega_\varphi^{-r(\varphi,21)\cdot r(\varphi,22)\cdot r(\varphi,23)\cdot r(\varphi,24)-r(\varphi,21)\cdot r(\varphi,25)-r(\varphi,23)\cdot r(\varphi,26)}$$

$$f_\varphi(\lambda_{1,\varphi}, \lambda_{2,\varphi})^{-a(\varphi)} =$$

$$z'_{\varphi,2,1}z'_{\varphi,2,2}z'_{\varphi,2,3} f_\varphi\left(\lambda_{1,\varphi}\eta_{1,\varphi}^{r(\varphi,21)\cdot r(\varphi,22)}, \lambda_{2,\varphi}^{a(\varphi)}\eta_{2,\varphi}^{r(\varphi,23)\cdot r(\varphi,24)}\right)$$

$$f_\varphi\left(\eta_{1,\varphi}^{r(\varphi,21)}, \lambda_{2,\varphi}^{-a(\varphi)\cdot r(\varphi,22)}\eta_{2,\varphi}^{r(\varphi,25)}\right) f_\varphi\left(\lambda_{1,\varphi}^{-r(\varphi,24)}\eta_{1,\varphi}^{r(\varphi,26)}, \eta_{2,\varphi}^{r(\varphi,23)}\right)$$

$$\Omega_\varphi^{-r(\varphi,21)\cdot r(\varphi,22)\cdot r(\varphi,23)\cdot r(\varphi,24)-r(\varphi,21)\cdot r(\varphi,25)-r(\varphi,23)\cdot r(\varphi,26)}$$

$$f_\varphi(\lambda_{1,\varphi}, \lambda_{2,\varphi})^{-a(\varphi)} = z'_{\varphi,2,1}z'_{\varphi,2,2}z'_{\varphi,2,3}$$

Here, $z_{\phi,2,1}'=z_{\phi2,1}f_\phi(\lambda_{1,\phi}\eta_{1,\phi}^{r(\phi,21)\cdot r(\phi,22)}, \lambda_{2,\phi}^{a(\phi)}\eta_{2,\phi}^{r(\phi,23)\cdot r(\phi,24)})^{-1}$, $z_{\phi,2,2}'=z_{\phi,2,2}f_\phi(\eta_{1,\phi}^{r(\phi,21)}, \lambda_{2,\phi}^{-a(\phi)\cdot r(\phi,22)}\eta_{2,\phi}^{r(\phi,25)})^{-1}$, and $z_{\phi2,3}'=z_{\phi,2,3}f_\phi(\lambda_{1,\phi}^{-r(\phi,24)}\eta_{1,\phi}^{r(\phi,26)}, \eta_{2,\phi}^{r(\phi,23)})^{-1}$ are satisfied.

Each of $z_{\phi,2,1}'$, $z_{\phi,2,2}'$ and $z_{\phi,2,3}'$ is statistically close to a probability distribution that is independent of a(ϕ). Accordingly, a probability distribution formed by $v_\phi f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})^{-a(\Phi)}$ is statistically close to a certain probability distribution $X_{\phi,2}$ that is independent of a(ϕ). Therefore, v_ϕ is an output of a randomizable sampler having an error $X_{\phi,2}$ for $f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})$. Similarly, u_ϕ is an output of a randomizable sampler having an error $X_{\phi,1}$ for $f_\phi(\lambda_{1,\phi}, \lambda_{2,\phi})$.

Eighteenth Embodiment

In the embodiments described above, a plurality of pairs (a(ϕ), b(ϕ)) of natural numbers a(ϕ) and b(ϕ) that are relatively prime to each other are stored in the natural number storage **21101**-ϕ of the computing apparatus and the pairs (a(ϕ), b(ϕ)) are used to perform the processes. However, one of a(ϕ) and b(ϕ) may be a constant. For example, a(ϕ) may be fixed at 1 or b(ϕ) may be fixed at 1. In other words, one of the first randomizable sampler and the second randomizable sampler may be replaced with a sampler. If

one of $a(\phi)$ and $b(\phi)$ is a constant, the process for selecting the constant $a(\phi)$ or $b(\phi)$ is unnecessary, $a(\phi)$ or $b(\phi)$ as a constant is not input in the processing units, and the processing units can treat it as a constant in computations. If $a(\phi)$ or $b(\phi)$ as a constant is equal to 1, $f_\phi(x_\phi)=u_\phi^{b'(\Phi)}v_\phi^{a'(\Phi)}$ can be obtained as $f_\phi(x_\phi)=v_\phi$ or $f_\phi(x_\phi)=u_\phi$ without using $a'(\phi)$ or $b'(\phi)$.

An eighteenth embodiment is an example of such a variation, in which $b(\phi)$ is fixed at 1 and the second randomizable sampler is replaced with a sampler. The following description will focus on differences from the twelfth embodiment. Specific examples of the first randomizable sampler and the sampler are similar to those described in the thirteenth to seventeenth embodiments and therefore description of the first randomizable sampler and the sampler will be omitted.

<Configuration>

As illustrated in FIG. **31**, a proxy computing system **206** of the eighteenth embodiment includes a computing apparatus **261**-ϕ in place of the computing apparatus **211**-ϕ of the twelfth embodiment and a capability providing apparatus **262** in place of the capability providing apparatus **212**.

As illustrated in FIG. **44**, the computing apparatus **261**-ϕ of the eighteenth embodiment includes, for example, a natural number storage **26101**-ϕ, a natural number selecting unit **26102**-ϕ, an input information providing unit **26104**-ϕ, a first computing unit **26105**-ϕ, a first power computing unit **21106**-ϕ, a first list storage **21107**-ϕ, a second computing unit **26108**-ϕ, a second list storage **26110**-ϕ, a determining unit **26111**-ϕ, a final output unit **21112**-ϕ, and a controller **21113**-ϕ.

As illustrated in FIG. **33**, the capability providing apparatus **262** of the eighteenth embodiment includes, for example, a first output information computing unit **26201**, a second output information computing unit **26202**, and a controller **21205**.

<Assumptions for Processes>

No natural number $b(\phi)$ is stored in the natural number storage **26101**-ϕ of the computing apparatus **261**-ϕ and only a plurality of natural numbers $a(\phi)$ are stored. The rest of the assumptions are the same as those in any of the twelfth to seventeenth embodiments.

<Processes>

As illustrated in FIG. **45**, first, the natural number selecting unit **26102**-ϕ of the computing apparatus **261**-ϕ (FIG. **44**) randomly reads one natural number $a(\phi)$ from among the plurality of natural numbers $a(\phi)$ stored in the natural number storage **26101**-ϕ. Information on the read natural number $a(\phi)$ is sent to the input information providing unit **26104**-ϕ and the first power computing unit **21106**-ϕ (step S**26100**).

The controller **21113**-ϕ sets t=1 (step S**21102**).

The input information providing unit **26104**-ϕ generates and outputs first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ each of that corresponds to an input element x_ϕ. Preferably, the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ are information whose relation with the element x_ϕ is scrambled. This enables the computing apparatus **261**-ϕ to conceal the element x_ϕ from the capability providing apparatus **262**. Preferably, the second input information $\tau_{\phi,2}$ of this embodiment further corresponds to the natural number $a(\phi)$ selected by the natural number selecting unit **26102**-ϕ. This enables the computing apparatus **261**-ϕ to evaluate the computation capability provided by the capability providing apparatus **262** with a high degree of accuracy (step S**26103**). A specific example of a pair of the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ is a pair of first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ when $b(\phi)=1$ in any of the thirteenth to seventeenth embodiments.

As illustrated in FIG. **38**, the first input information $t_{\phi,1}$ is input in the first output information computing unit **26201** of the capability providing apparatus **262** (FIG. **33**) and the second input information $\tau_{\phi,2}$ is input in the second output information computing unit **26202** (step S**26200**).

The first output information computing unit **26201** uses the first input information $\tau_{\phi,1}$ to correctly compute $f_\phi(\tau_{\phi,1})$ with a probability greater than a certain probability and sets the obtained result of the computation as first output information $z_{\phi,1}$ (step S**26201**). The second output information computing unit **26202** uses the second input information $\tau_{\phi,2}$ to correctly compute $f_t(\tau_{\phi,2})$ with a probability greater than a certain probability and sets the obtained result of the computation as second output information $z_{\phi,2}$ (step S**26202**). That is, the first output information computing unit **26201** and the second output information computing unit **26202** can output computation results that have an intentional or unintentional error. In other words, the result of the computation by the first output information computing unit **26201** may or may not be $f_\phi(\tau_{\phi,1})$ and the result of the computation by the second output information computing unit **26202** may or may not be $f_\phi(\tau_{\phi,2})$. A specific example of a pair of the first output information $z_{\phi,1}$ and the second output information $z_{\phi,2}$ is a pair of first output information $z_{\phi,1}$ and second output information $z_{\phi,2}$ when $b(\phi)=1$ in any of the thirteenth to seventeenth embodiments.

The first output information computing unit **26201** outputs the first output information $z_{\phi,1}$ and the second output information computing unit **26202** outputs the second output information $z_{\phi,2}$ (step S**26203**).

Returning to FIG. **45**, the first output information $z_{\phi,1}$ is input in the first computing unit **26105**-ϕ of the computing apparatus **261**-ϕ (FIG. **44**) and the second output information $z_{\phi,2}$ is input in the second computing unit **26108**-ϕ. The first output information $z_{\phi,1}$ and the second output information $z_{\phi,2}$ are equivalent to the computation capability provided by the capability providing apparatus **262** to the computing apparatus **261**-ϕ (step S**26104**).

The first computing unit **26105**-ϕ generates a computation result $u_\phi=f_\phi(x_\phi)x_{\phi,1}$ from the first output information $z_{\phi,1}$. A specific example of the result u_ϕ of the computation is the result u_ϕ of the computation in any of the thirteenth to seventeenth embodiments when $b(\phi)=1$. The result u_ϕ of the computation is sent to the first power computing unit **21106**-ϕ (step S**26105**).

The first power computing unit **21106**-ϕ computes $u_\phi'=u_\phi^{a(\Phi)}$. The pair of the result u_ϕ, of the computation and u_ϕ' computed on the basis of the result of the computation, (u_ϕ, u_ϕ'), is stored in the first list storage **21107**-ϕ (step S**21106**).

The second computing unit **26108**-ϕ generates a computation result $v_\phi=f_\phi(x_\phi)^{a(\Phi)}x_{\phi,2}$ from the second output information $z_{\phi,2}$. A specific example of the result v_ϕ of the computation is the result v_ϕ of the computation in any of the thirteenth to seventeenth embodiments. The result v_ϕ of the computation is stored in the second list storage **26110**-ϕ (step S**26108**).

The determining unit **26111**-ϕ determines whether or not there is a pair of u_ϕ' and v_ϕ that belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ among the pairs (u_ϕ, u_ϕ') stored in the first list storage **21107**-ϕ and v_ϕ stored in the second list storage **26110**-ϕ as in any of the twelfth to seventeenth embodiments (step S**26110**). If there is a pair of u_ϕ' and v_ϕ that belong to the class $CL_\phi(M_\phi)$ corresponding to the same M_ϕ, the process proceeds to step S26114. If there

is not a pair of u_ϕ' and v_ϕ that belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, the process proceeds to step S**21111**.

At step S**21111**, the controller **21113**-ϕ determines whether or not t=T (step S**21111**). T is a predetermined natural number. If t=T, the controller **21113**-ϕ outputs information indicating that the computation is impossible, for example the symbol "⊥" (step S**21113**) and the process ends. If not t=T, the controller **21113**-ϕ increments t by 1, that is, sets t=t+1 (step S**21112**) and the process returns to step S**26103**.

At step S**26114**, the final output unit **21112**-ϕ outputs u_ϕ corresponding to u_ϕ' included in the pair of u_ϕ' and v_ϕ determined to belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ (step S**26114**). The u_ϕ thus obtained corresponds to $u_\phi^{b'(\phi)}v_\phi^{a'(\phi)}$ in the twelfth to seventeenth embodiments when b(ϕ)=1. That is, u_ϕ thus obtained can be $f_\phi(x_\phi)$ with a high probability. Therefore, a predetermined reliability that the selected u_ϕ is equal to $f_\phi(x_\phi)$ or higher can be achieved by repeating at least the process described above multiple times and selecting the value u_ϕ obtained with the highest frequency among the values obtained at step S**26114**. As will be described, $u_\phi=f_\phi(x_\phi)$ can result with an overwhelming probability, depending on settings.

<<Reason Why $f_\phi(x_\phi)$ can be Obtained>>

The reason why a decryption result $f_\phi(x_\phi)$ can be obtained on the computing apparatus **261**-ϕ of this embodiment will be describe below. Terms required for the description will be defined first.

Black-box:

A black-box $F_\phi(\tau_\phi)$ of $f_\phi(t_\phi)$ is a processing unit that takes an input of $\tau_\phi \in H_\phi$ and outputs $z_\phi \in G_\phi$. In this embodiment, each of the first output information computing unit **26201** and the second output information computing unit **26202** is equivalent to the black box $F_\phi(\tau_\phi)$ for the decryption function $f_\phi(\tau_\phi)$. A black-box $F_\phi(\tau_\phi)$ that satisfies $z_\phi=f_\phi(\tau_\phi)$ for elements $\tau_\phi \in_U H_\phi$ and $z_\phi=F_\phi(\tau_\phi)$ arbitrarily selected from a group H_ϕ with a probability greater than δ (0<δ<1), that is, a black-box $F_\phi(\tau_\phi)$ for $f_\phi(\tau_\phi)$ that satisfies

$$Pr[z_\phi=f_\phi(\tau^\phi)|\tau_\phi \in_U H_\phi, z_\phi=F_\phi(\tau_\phi)]>\delta \tag{15}$$

is called the δ-reliable black-box $F_\phi(\tau_\phi)$ for $f_\phi(\tau_\phi)$. Here, δ is a positive value and is equivalent to the "certain probability" mentioned above.

Self-corrector:

A self-corrector $C^F(x_\phi)$ is a processing unit that takes an input of $x_\phi \in H_\phi$, performs computation by using a black-box $F_\phi(\tau_\phi)$ for $f_\phi(\tau_\phi)$ and outputs $j \in G \cup \perp$. In this embodiment, the computing apparatus **261**-ϕ is equivalent to the self-corrector $C^F(x_\phi)$.

Almost self-corrector:

Assume that a self-corrector $C^F(x_\phi)$ that takes an input of $x_\phi \in H_\phi$ and uses the δ-reliable black-box $F_\phi(\tau_\phi)$ for $f_\phi(\tau_\phi)$ outputs a correct value $j=f_\phi(x_\phi)$ with a probability sufficiently greater than the provability with which the self-corrector $C^F(x_\phi)$ outputs an incorrect value $j \neq f_\phi(x_\phi)$. That is, assume that a self-corrector $C^F(x_\phi)$ satisfies

$$Pr[j=f_\phi(x_\phi)|j=C^F(x_\phi), j\neq\perp]>Pr[j\neq f_\phi(x_\phi)|j=C^F(x_\phi), j\neq\perp]+\Delta \tag{16}$$

Here, Δ is a certain positive value (0<Δ<1). If this is the case, the self-corrector $C^F(x_\phi)$ is called an almost self-corrector. For example, for a certain positive value Δ' (0<Δ'<1), if a self-corrector $C^F(x_\phi)$ satisfies

$$Pr[j=f_\phi(x_\phi)|j=C^F(x_\phi)]>(1/3)+\Delta'$$

$$Pr[j=\perp|j=C^F(x_\phi)]<1/3$$

$$Pr[j\neq f_\phi(x_\phi) \text{ and } j\neq\perp|j=C^F(x_\phi)]<1/3,$$

then the self-corrector $C^F(x_\phi)$ is an almost self-corrector. Examples of Δ' include Δ'=1/12 and Δ'=1/3.

Robust self-corrector:

Assume that a self-corrector $C^F(x_\phi)$ that takes an input of $x_\phi \in H$ and uses a δ-reliable black-box $F_\phi(\tau_\phi)$ for $f_\phi(x_\phi)$ to output a correct value $j=f_\phi(x_\phi)$ or $j=\perp$ with an overwhelming probability. That is, assume that for a negligible error ξ (0≤ξ<1), a self-corrector $C^F(x_\phi)$ satisfies

$$Pr[j=f_\phi(x_\phi) \text{ or } j=\perp|j=C^F(x_\phi)]>1-\xi \tag{17}$$

If this is the case, the self-corrector $C^F(x_\phi)$ is called a robust self-corrector. An example of the negligible error ξ is a function value ξ(k) of a security parameter k. An example of the function value ξ(k) is a function value ξ(k) such that {ξ(k)p(k)} converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function value ξ(k) include $\xi(k)=2^{-k}$ and $\xi(k)=2^{-\sqrt{k}}$.

A robust self-corrector can be constructed from an almost self-corrector. Specifically, a robust self-corrector can be constructed by executing an almost self-constructor multiple times for the same x and selecting the most frequently output value, except ⊥, as j. For example, an almost self-corrector is executed O(log(1/ξ)) times for the same x and the value most frequently output is selected as j, thereby a robust self-corrector can be constructed. Here, O(·) represents O notation.

Pseudo-free action:

An upper bound of the probability

$$Pr[\alpha_\phi^{a(\phi)}=\beta_\phi \text{ and } \alpha_\phi\neq e_{\phi,g}|a(\phi)\in_U\Omega_\phi,\alpha_\phi\in X_{\phi,1},\beta_\phi\in X_{\phi,2}] \tag{18}$$

of satisfying $\alpha_\phi^{a(\phi)}=\beta_\phi$ for all possible $X_{\phi,1}$ and $X_{\phi,2}$ is called a pseudo-free indicator of a pair (G_ϕ, Ω_ϕ) and is represented as $P(G_\phi, \Omega_\phi)$, where $G_\phi$is a group G_ϕ, Ω_ϕ is a set of natural numbers $\Omega_\phi\{0, \ldots, M_\phi\}$ (M_ϕ is a natural number greater than or equal to 1), α_ϕ and β_ϕ are realizations $\alpha_\phi \in X_{\phi,1}$ ($\alpha_\phi \neq e_{\phi,g}$) and $\beta_\phi \in X_{\phi,2}$ of random variables $X_{\phi,1}$ and $X_{\phi,2}$ that have values in the group G_ϕ, and $a(\phi)\in\Omega_\phi$. If a certain negligible function ζ(k) exists and

$$P(Q_\phi,\Omega_\phi)<\zeta(k) \tag{19},$$

then a computation defined by the pair (G_ϕ, Ω_ϕ) is called a pseudo-free action. Note that "$\alpha_\phi^{a(\phi)}$" means that a computation defined on the group G_ϕ is applied a(ϕ) times to α_ϕ. An example of the negligible function ζ(k) is such that {ζ(k)p(k)} converges to 0 for a sufficiently large k, where p(k) is an arbitrary polynomial. Specific examples of the function ζ(k) include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. For example, if the probability of Formula (18) is less than $O(2^{-k})$ for a security parameter k, a computation defined by the pair (G_ϕ, Ω_ϕ) is a pseudo-free action. For example, if the number of the elements $|\Omega_\phi\cdot\alpha_\phi|$ of a set $\Omega_\phi\cdot\alpha_\phi=\{a(\phi)(\alpha_\phi)|a(\phi)\in\Omega_\phi\}$ exceeds 2^k for any $\alpha_\phi \in G_\phi$ where $\alpha_\phi \neq e_{\phi,g}$, a computation defined by the pair (G_ϕ, Ω_ϕ) is a pseudo-free action. Note that $a(\phi)(\alpha_\phi)$ represents the result of a given computation on a(ϕ) and α_ϕ. There are many such examples. For example, if the group G_ϕ is a residue group Z/pZ modulo prime p, the prime p is the order of 2^k, the set $\Omega_\phi=\{0, \ldots, p-2\}$, $a(\phi)(\alpha_\phi)$ is $\alpha_\phi^{a(\phi)}\in Z/pZ$, and $\alpha_\phi\neq e_{\phi,g}$, then $\Omega_\phi\cdot\alpha_\phi=\{\alpha_\phi^{a(\phi)}|a(\phi)=0, \ldots, p-2\}=\{e_{\phi,g}, \alpha_\phi, \ldots, \alpha_\phi^{p-2}\}$ and $|\Omega_\phi\cdot\alpha_\phi|=p-1$. If a certain constant C exists and k is sufficiently large, $|\Omega_\phi\cdot\alpha_\phi|>C2^k$ is satisfied because the prime p is the order of

2^k. Here, the probability of Formula (18) is less than $C^{-1}2^{-k}$ and a computation defined by such pair (G_ϕ, Ω_ϕ) is a pseudo-free action.

δ^γ-reliable randomizable sampler:

A randomizable sampler that whenever a natural number $a(\phi)$ is given, uses the black-box $F_\phi(\tau_\phi)$ for δ-reliable $f_\phi(\tau_\phi)$ and returns $w_\phi{}^{a(\Phi)}x_\phi'$ corresponding to a sample x_ϕ' that depends on a random variable X_ϕ for $w_\phi \epsilon G_\phi$, where the probability that $w_\phi{}^{a(\Phi)}x_\phi'=w_\phi{}^{a(\Phi)}$ is greater than δ^γ (γ is a positive constant), that is,

$$Prf[w_\phi{}^{a(\Phi)}x_\phi'=w_\phi{}^{a(\Phi)}]>\delta^\gamma \tag{20}$$

is satisfied, is called a δ^γ-reliable randomizable sampler. The combination of the input information providing unit **26104**-ϕ, the second output information computing unit **26202**, and the second computing unit **26108**-ϕ of this embodiment is a δ^γ-reliable randomizable sampler for $w_\phi=f_\phi(x_\phi)$.

The definitions given above will be used to describe the reason why $f_\phi(x_\phi)$ can be obtained on the computing apparatus **261**-ϕ. At step **S26110** of this embodiment, determination is made as to whether there is a pair of u_ϕ' and v_ϕ that belong to a class corresponding to the same element, that is, whether there is a pair of $u_\phi{}^{a(\Phi)}$ and v_ϕ that belong to a class corresponding to the same element. Since the combination of the input information providing unit **26104**-ϕ, the second output information computing unit **26202**, and the second computing unit **26108**-ϕ is a δ^γ-reliable randomizable sampler (Formula (20)), $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to a class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ with an asymptotically large probability if T is a large value greater than a certain value determined by k, δ and γ (Yes at step **S26110**). For example, Markov's inequality can be used to show that if $T\geq 4/\delta^\gamma$, the probability that $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to a class corresponding to the same element (Yes at step **S26110**) is greater than ½.

Since $u_\phi=f_\phi(x_\phi)x_{\phi,1}$ and $v_\phi=f_\phi(x_\phi)x_{\phi,2}$ in this embodiment, $u_\phi{}^{a(\Phi)}=v_\phi$ is satisfied and $x_{\phi,1}{}^{a(\Phi)}=x_{\phi,2}$ holds if the determination at step **S26110** is yes, provided that the function $f_\phi(x_\phi)$ is an injective function for the element x_ϕ. Even where the function $f_\phi(x_\phi)$ is not an injective function for an element x_ϕ, $x_{\phi,1}{}^{a(\Phi)}=x_{\phi,2}$ holds if the determination at step **S26110** is yes, provided that $f_\phi(x_\phi)$ is a homomorphic function.

$x_{\phi,1}{}^{a(\Phi)}=x_{\phi,2}$ holds if $x_{\phi,1}=x_{\phi,2}=e_{\phi,g}$ or $x_{\phi,1}\neq e_{\phi,g}$. If $x_{\phi,1}=x_{\phi,2}=e_{\phi,g}$, then $u_\phi=f_\phi(x_\phi)$ and therefore u_ϕ output at step **S26114** is a correct decryption result $f_\phi(x_\phi)$. On the other hand, if $x_{\phi,1}\neq e_{\phi,g}$, then $u_\phi\neq f_\phi(x_\phi)$ and therefore u_ϕ output at step **S26114** is not a correct decryption result $f_\phi(x_\phi)$.

If an computation defined by a pair (G_ϕ, Ω_ϕ) of a group G_ϕ and a set Ω_ϕ to which a natural number $a(\phi)$ belongs is pseudo-free action or $T^2P(G_\phi, \Omega_\phi)$ is asymptotically small for a pseudo-free index $P(G_\phi, \Omega_\phi)$, the probability that $x_{\phi,1}\neq e_{\phi,g}$ when $x_{\phi,1}{}^{a(\Phi)}=x_{\phi,2}$ (Formula (18)) is asymptotically small. Accordingly, the probability that $x_{\phi,1}=e_{\phi,g}$ when $x_{\phi,1}{}^{a(\Phi)}=x_{\phi,2}$ is asymptotically large. Therefore, if an computation defined by a pair (G_ϕ, Ω_ϕ) is a pseudo-free action or $T^2P(G_\phi, \Omega_\phi)$ is asymptotically small, the probability that an incorrect decryption result $f_\phi(x_\phi)$ is output when $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ is sufficiently smaller than the probability that a correct decryption result $f_\phi(x_\phi)$ is output when $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ. In this case, it can be said that the computing apparatus **261**-ϕ is an almost self-corrector (see Formula (16)). Therefore, a robust self-corrector can be constructed from the computing apparatus **261**-ϕ as described above and a correct decryption result $f_\phi(x_\phi)$ can be obtained with an overwhelming probability. If a computation defined by (G_ϕ, Ω_ϕ) is a pseudo-free action, the probability that an incorrect decryption result $f_\phi(x_\phi)$ is output when $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ is also negligible. In that case, the computing apparatus **261**-ϕ outputs a correct decryption result $f_\phi(x_\phi)$ or $\perp$ with an overwhelming probability.

Note that "$\eta(k')$ is asymptotically small" means that k_0 is determined for an arbitrary constant ρ and the function value $\eta(k')$ for any k' that satisfies $k_0<k'$ for k_0 is less than ρ. An example of k' is a security parameter k. "$\eta(k')$ is asymptotically large" means that k_0 is determined for an arbitrary constant ρ and the function value $1-\eta(k')$ for any k' that satisfies $k_0<k'$ for k_0 is less than ρ.

The proof given above also proves that "if u_ϕ' and v_ϕ' belong to the class $CL_\phi(M_\phi)$ corresponding to the same element M_ϕ, it is highly probable that the first randomizable sampler has correctly computed $u_\phi=f_\phi(x_\phi)^{b(\Phi)}$ and the second randomizable sampler has correctly computed $v_\phi=f_\phi(x_\phi)^{a(\Phi)}$ ($x_{\phi,1}$ and $x_{\phi,2}$ are identity elements $e_{\phi,g}$ of the group G_ϕ)" stated in the twelfth embodiment, as can be seen by replacing $a(\phi)$ with $a(\phi)/b(\phi)$.

<<δ^γ-Reliable Randomizable Sampler and Security>>

Consider the following attack.

A black-box $F_\phi(\tau_\phi)$ or part of the black-box $F_\phi(\tau_\phi)$ intentionally outputs an invalid z_ϕ or a value output from the black-box $F_\phi(T_\phi)$ is changed to an invalid z_ϕ.

$w_\phi{}^{a(\Phi)}x_\phi'$ corresponding to the invalid z_ϕ is output from the randomizable sampler.

$w_\phi{}^{a(\Phi)}x_\phi'$ corresponding to the invalid z_ϕ increases the probability with which the self-corrector $C^F(x_\phi)$ outputs an incorrect value even though $w_\phi{}^{a(\Phi)}x_\phi'$ corresponding to the invalid z_ϕ causes the self-corrector $C^F(x_\phi)$ to determine that $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to the class $CL_\phi(M_\phi)$ corresponding to the same M_ϕ (Yes at step **S26110**).

Such an attack is possible if the probability distribution $D_a=w_\phi{}^{a(\Phi)}x_\phi'w_\phi{}^{-a(\Phi)}$ of an error of $w_\phi{}^{a(\Phi)}x_\phi'$ output from the randomizable sampler for a given natural number $a(\phi)$ depends on the natural number $a(\phi)$. For example, if tampering is made so that v_ϕ output from the second computing unit **26108**-ϕ is $f_\phi(x_\phi)^{a(\Phi)}x_{\phi,1}{}^{a(\Phi)}$, always $u_\phi{}^{a(\Phi)}=v_\phi$ holds and it is determined that $u_\phi{}^{a(\Phi)}$ and v_ϕ belong to the class $CL_\phi(M_\phi)$ corresponding to the same M_ϕ regardless of the value of $x_{\phi,1}$. Therefore it is desirable that the probability distribution $D_a=w_\phi{}^{a(\Phi)}x_\phi'w_\phi{}^{-a(\Phi)}$ of an error of $w_\phi{}^{a(\Phi)}x_\phi'$ output from the randomizable sampler for a given natural number $a(\phi)$ do not depend on the natural number $a(\phi)$.

Alternatively, it is desirable that the randomizable sampler be such that there is a probability distribution D that has a value in a group G_ϕ and is indistinguishable from the probability distribution $D_a=w_\phi{}^{a(\Phi)}x_\phi'w_\phi{}^{-a(\Phi)}$ of an error of $w_\phi{}^{a(\Phi)}x_\phi'$ for any element $a(\phi)\epsilon{}^\forall\Omega_\phi$ of a set Ω_ϕ (the probability distribution D_a and the probability distribution D are statistically close to each other). Note that the probability distribution D is not dependent on the natural number $a(\phi)$. That the probability distribution D_a and the probability distribution D are indistinguishable from each other means that the probability distribution D_a and the probability distribution D cannot be distinguished from each other by a polynomial time algorithm. For example, if

$$\Sigma_{g\epsilon G}|Pr[g\epsilon D]-Pr[g\epsilon D_a]|<\zeta \tag{21}$$

is satisfied for negligible ζ ($0\leq\zeta\leq 1$), the probability distribution D_a and the probability distribution D cannot be distinguished from each other by a polynomial time algorithm. An example of negligible ζ is a function value $\zeta(k)$ of the security parameter k. An example of the function value

$\zeta(k)$ is a function value such that $\{\zeta(k)p(k)\}$ converges to 0 for a sufficiently large k where p(k) is an arbitrary polynomial. Specific examples of the function $\zeta(k)$ include $\zeta(k)=2^{-k}$ and $\zeta(k)=2^{-\sqrt{k}}$. These also apply to the twelfth to seventeenth embodiments which use natural numbers $a(\phi)$ and $b(\phi)$.

[Variations of the Twelfth to Eighteenth Embodiments]

In a variation of the twelfth to eighteenth embodiments, even though it is not guaranteed that the capability providing apparatus always performs correct computations, a value $u_\phi^{b'(\phi)}v_\phi^{a'(\phi)}$ obtained on each of the computing apparatuses ϕ will be $f_\phi(x_\phi)$ with a high probability when the capability providing apparatus correctly computes $f_\phi(\tau_{\phi,1})$ and $f_\phi(\tau_{\phi,2})$ with a probability greater than a certain probability. Therefore, each of the computing apparatuses ϕ can cause the capability providing apparatus to perform a computation without performing authentication and can obtain a correct result of the computation (for example, the result of decryption of a ciphertext) by using the result of the computation.

The present invention is not limited to the embodiments described above. For example, random variables $X_{\phi,1}$, $X_{\phi,2}$ and $X_{\phi,3}$ may or may not be the same.

Each of the random number generators generates uniform random numbers to increase the security of the proxy computing system to the highest level. However, if the level of security required is not so high, at least some of the random number generators may generate random numbers that are not uniform random numbers. While it is desirable from the computational efficiency point of view that random natural numbers selected are natural numbers greater than or equal to 0 and less than $K_{\phi,H}$ or natural numbers greater than or equal to 0 and less than or equal to $2^{\mu(k)+k}$ in the embodiments described above, random numbers that are natural numbers greater than or equal to $K_{\phi,H}$ or natural numbers greater than $2^{\mu(k)+k}$ may be selected instead. Here, μ is a function of k. For example, μ may be the length of an element of the group H_ϕ as a bit string.

The process of the capability providing apparatus may be performed multiple times each time the computing apparatus provides first input information $\tau_{\phi,1}$ and second input information $\tau_{\phi,2}$ corresponding to the same $a(\phi)$ and $b(\phi)$ to the capability providing apparatus. This enables the computing apparatus to obtain a plurality of pieces of first output information $z_{\phi,1}$, second output information $z_{\phi,2}$, and third output information $z_{\phi,3}$ each time the computing apparatus provides first input information $\tau_{\phi,1}$, and second input information $\tau_{\phi,2}$ to the capability providing apparatus. Consequently, the number of exchanges and the amount of communication between the computing apparatus and the capability providing apparatus can be reduced.

The computing apparatus may provide a plurality of pieces of the first input information $\tau_{\phi,1}$ and the second input information $\tau_{\phi,2}$ to the capability providing apparatus at once and may obtain a plurality of pieces of corresponding first output information $z_{\phi,1}$, second output information $z_{\phi,2}$ and third output information $z_{\phi,3}$ at once. This can reduce the number of exchanges between the computing apparatus and the capability providing apparatus.

The units of the computing apparatus may exchange data directly or through a memory, which is not depicted. Similarly, the units of the capability providing apparatus may exchange data directly or through a memory, which is not depicted.

Check may be made to see whether u_ϕ and v_ϕ obtained at the first computing unit and the second computing unit of any of the embodiments are elements of the group G_ϕ. They are elements of the group G_ϕ, the process described above may be continued; if u_ϕ or v_ϕ is not an element of the group G_ϕ, information indicating that the computation is impossible, for example the symbol "$\perp$", may be output.

Furthermore, the processes described above may be performed not only in time sequence as is written or may be performed in parallel with one another or individually, depending on the throughput of the apparatuses that perform the processes or requirements. It would be understood that other modifications can be made without departing from the spirit of the present invention.

If any of the configurations described above is implemented by a computer, the processes of the functions the apparatuses need to include are described by a program. The processes of the functions are implemented on the computer by executing the program on the computer. The program describing the processes can be recorded on a computer-readable recording medium. An example of the computer-readable recording medium is a non-transitory recording medium. Examples of such a recording medium include a magnetic recording device, an optical disc, a magneto-optical recording medium, and a semiconductor memory.

The program is distributed by selling, transferring, or lending a portable recording medium on which the program is recorded, such as a DVD or a CD-ROM. The program may be stored on a storage device of a server computer and transferred from the server computer to other computers over a network, thereby distributing the program.

A computer that executes the program first stores the program recorded on a portable recording medium or transferred from a server computer into a storage device of the computer. When the computer executes the processes, the computer reads the program stored on the recording medium of the computer and executes the processes according to the read program. In another mode of execution of the program, the computer may read the program directly from a portable recording medium and execute the processes according to the program or may execute the processes according to the program each time the program is transferred from the server computer to the computer. Alternatively, the processes may be executed using a so-called ASP (Application Service Provider) service in which the program is not transferred from a server computer to the computer but process functions are implemented by instructions to execute the program and acquisition of the results of the execution. Note that the program in this mode encompasses information that is provided for processing by an electronic computer and is equivalent to the program (such as data that is not direct commands to a computer but has the nature that defines processing of the computer).

While the apparatuses are configured by causing a computer to execute a predetermined program in the embodiments described above, at least some of the processes may be implemented by hardware.

This application is based upon and claims priority of Japanese Patent Applications No. 2010-239342, 2011-5899, 2011-88002, 2011-77779, the entire contents of which are incorporated by reference herein.

INDUSTRIAL APPLICABILITY

As has been described above, each of the computing apparatuses of the embodiments are capable of obtaining a correct result of computation by using the computation capability provided by the capability providing apparatus even if the capability providing apparatus is in a condition where the capability providing apparatus does not always perform a correct process. Accordingly, the computing appa-

ratus does not need to perform verification for confirming the validity of the capability providing apparatus. Furthermore, if a plurality of computing apparatuses share the capability providing apparatus, the computing apparatuses can obtain a correct result of computation.

Such a proxy computing system can be used in, for example, volunteer-based distributed computing, P2P computing services, computing services paid for with payments for advertisements, computing services that are provided as network services or public infrastructures, and network services that are substitutes for computing packages licensed in the form of libraries.

DESCRIPTION OF REFERENCE NUMERALS

1-5, **101-105**, **201-207**: Proxy computing system
11-61, **111-151**, **211-271**: Computing apparatus
12-62, **112-152**, **212-272**: Capability providing apparatus.

What is claimed is:

1. A proxy computing system comprising a computing apparatus and a capability providing apparatus,

wherein G and H are groups, f(x) is a decryption function for decrypting a certain ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a and b are natural numbers that are relatively prime to each other;

the computing apparatus comprises processing circuitry configured to implement an input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H;

the capability providing apparatus comprises processing circuitry configured to implement:

a first output information computing unit using the first input information τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets an obtained result of the computation as first output information z_1; and

a second output information computing unit using the second input information τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets an obtained result of the computation as second output information z_2; and

the processing circuitry of the computing apparatus is further configured to implement:

a first computing unit configured to generate a computation result $u=f(x)^b x_1$ from the first output information z_1, and generate a computation result $u'=u^a$;

a second computing unit configured to generate a computation result $v=f(x)^a x_2$ from the second output information z_2, and generate a computation result $v'=v^b$;

a determining unit configured to determine whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

a final output unit configured to output $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy u'=v' and thus $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

2. The proxy computing system according to claim **1**,

wherein the computing apparatus comprises a natural number selecting unit selecting at least one of the natural numbers a and b; and

the first input information τ_1 further corresponds to the natural number b and the second input information τ_2 further corresponds to the natural number a.

3. The proxy computing system according to claim **2**,

wherein the input information providing unit sets pieces of information whose relation with the ciphertext x is scrambled as the first input information τ_1 and the second input information τ_2.

4. The proxy computing system according to claim **1**,

wherein the input information providing unit sets pieces of information whose relation with the ciphertext x is scrambled as the first input information τ_1 and the second input information τ_2.

5. The proxy computing system according to any one of claims **1** to **4**;

wherein the decryption function f(x) is a homomorphic function, the group H is a cyclic group, a generator of the cyclic group H is μ_h, the order of the cyclic group H is K_H, and $v=f(\mu_h)$;

the input information providing unit comprises:

a first random number generator generating a random number r_1 that is a natural number greater than or equal to 0;

a first input information computing unit computing $\mu_h^{r1}x^b$ as the first input information τ_1;

a second random number generator generating a random number r_2 that is a natural number greater than or equal to 0; and

a second input information computing unit computing $\mu_h^{r2}x^a$ as the second input information τ_2;

the first output information computing unit uses the first input information $\mu_h^{r1}x^b$ to correctly compute $f(\mu_h^{r1}x^b)$ with a probability greater than a certain probability and sets an obtained result of the computation as the first output information z_1; and

the second output information computing unit uses the second input information $\mu_h^{r2}x^a$ to correctly compute $f(\mu_h^{r2}x^a)$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information z_2;

the first computing unit computes z_1v^{-r1} to obtain the computation result u; and

the second computing unit computes z_2v^{-r2} to obtain the computation result v.

6. The proxy computing system according to claim **5**,

wherein the first random number generator generates the random number r_1 when $b\neq 1$;

the first input information computing unit computes the $\mu_h^{r1}x^b$ as the first input information τ_1 when $b\neq 1$;

the first output information computing unit sets the computation result obtained by using the first input information $\mu_h^{r1}x^b$ as the first output information z_1 when $b\neq 1$;

the first computing unit computes z_1v^{-r1} to obtain the computation result u when $b\neq 1$;

the second random number generator generates the random number r_2 when $a\neq 1$;

the second input information computing unit computes $\mu_h^{r2}x^a$ as the second input information τ_2 when $a\neq 1$;

the second output information computing unit sets the computation result obtained by using the second input information $\mu_h^{r2}x^a$ as the second output information z_2 when $a\neq 1$;

the second computing unit computes z_2v^{-r2} to obtain the computation result v when $a\neq 1$;

the input information providing unit comprises:

a third random number generator generating a random number r_3 that is a natural number greater than or equal to 0; and

a third input information computing unit setting x^{r3} as the first input information τ_1 when b=1 and sets x^{r3} as the second input information τ_2 when a=1;

the capability providing apparatus comprises:

a third output information computing unit using the x^{r3} to correctly compute $f(x^{r3})$ with a probability greater than a certain probability and sets an obtained result of the computation as third output information z_3; and

a third computing unit setting $z_3^{1/r3}$ as the computation result u when b=1 and sets $z_3^{1/r3}$ as the computation result v when a=1.

7. The proxy computing system according to any one of claims **1** to **4**,

wherein the group H is a direct product group G×G of the group G, the decryption function f(x) is a homomorphic function, the group G is a cyclic group, a generator of the cyclic group G is μ_g, the order of the cyclic group G is K_G, $x=(c_1, c_2)$, (V, W) is an element of the group H, and f(V, W)=Y;

the input information providing unit comprises:

a fourth random number generator generating a random number r_4 that is a natural number greater than or equal to 0;

a fifth random number generator generating a random number r_5 that is a natural number greater than or equal to 0;

a first input information computing unit computing $c_2{}^bW^{r4}$ and $c_1{}^bV^{r4}\mu_g{}^{r5}$ as the first input information τ_1;

a sixth random number generator generating a random number r_6 that is a natural number greater than or equal to 0;

a seventh random number generator generating a random number r_7 that is a natural number greater than or equal to 0; and

a second input information computing unit computing $c_2{}^aW^{r6}$ and $c_1{}^aV^{r6}\mu_g{}^{r7}$ as the second input information τ_2;

the first output information computing unit uses the first input information $c_1{}^bV^{r4}\mu_g{}^{r5}$ and $c_2{}^bW^{r4}$ to correctly compute $f(c_1{}^bV^{r4}\mu_g{}^{r5}, c_2{}^bW^{r4})$ with a probability greater than a certain probability and sets an obtained result of the computation as the first output information z_1;

the second output information computing unit uses the second input information $c_1{}^aV^{r6}\mu_g{}^{r7}$ and $c_2{}^aW^{r6}$ to correctly compute $f(c_1{}^aV^{r6}\mu g^{r7}, c_2{}^aW^{r6})$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information z_2,

the first computing unit computes $z_1Y^{-r4}\mu g^{-r5}$ to obtain the computation result u; and

the second computing unit computes $z_2Y^{-r6}\mu_g{}^{-r7}$ to obtain the computation result v.

8. The proxy computing system according to any one of claims **1** to **4**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for $f(x)^a$ and does not depend on the natural number a.

9. The proxy computing system according to claim **5**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for $f(x)^a$ and does not depend on the natural number a.

10. The proxy computing system according to claim **7**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for $f(x)^a$ and does not depends on the natural number a.

11. The proxy computing system according to any one of claims **1** to **4**, wherein the natural number a or the natural number b is a constant.

12. The proxy computing system according to claim **5**, wherein the natural number a or the natural number b is a constant.

13. The proxy computing system according to claim **7**, wherein the natural number a or the natural number b is a constant.

14. The proxy computing system according to any one of the claims **1** to **4**, further comprising a decryption control apparatus,

wherein the decryption control apparatus comprises an output unit outputting a decryption control instruction controlling a decryption process of the computing apparatus to the capability providing apparatus; and

the capability providing apparatus comprises a controller controlling, in accordance with the decryption control instruction, whether or not to output both of the first output information z_1 and the second output information z_2 from the first output information computing unit and the second output information computing unit.

15. The proxy computing system according to claim **5**, further comprising a decryption control apparatus,

wherein the decryption control apparatus comprises an output unit outputting a decryption control instruction controlling a decryption process of the computing apparatus to the capability providing apparatus; and

the capability providing apparatus comprises a controller controlling, in accordance with the decryption control instruction, whether or not to output both of the first

output information z_1 and the second output information z_2 from the first output information computing unit and the second output information computing unit.

16. The proxy computing system according to claim **7** further comprising a decryption control apparatus,

wherein the decryption control apparatus comprises an output unit outputting a decryption control instruction controlling a decryption process of the computing apparatus to the capability providing apparatus; and

the capability providing apparatus comprises a controller controlling, in accordance with the decryption control instruction, whether or not to output both of the first output information z_1 and the second output information z_2 from the first output information computing unit and the second output information computing unit.

17. The proxy computing system according to claim **1**, further comprising a decryption control apparatus,

wherein G_ι and H_ι are groups, ω is an integer greater than or equal to 2, $\iota=1, \ldots, \omega$, $f_\iota(X_\iota)$ is a decryption function for decrypting a cipher text λ_ι which is an element of the group H_ι with a particular decryption key s_ι to obtain an element of the group G_ι, $X_{\iota,1}$ and $X_{\iota,2}$ are random variables having values in the group G_ι, $x_{\iota,1}$ is a realization of the random variable $x_{\iota,1}$, $x_{\iota,2}$ is a realization of the random variable $X_{\iota,2}$, $a(\iota)$ and $b(\iota)$ are natural numbers that are relatively prime to each other, the group G is a group G_1, the group H is a group H_1, the ciphertext x is a ciphertext λ_1, the decryption function f(x) is a decryption function $f_1(\lambda_1)$, the random variable X_1 is a random variable $X_{1,1}$, the random variable X_2 is a random variable $X_{1,2}$, the realization x_1 is a realization $x_{1,1}$, the realization x_2 is a realization $x_{1,2}$, the natural number a is a natural number a(1), and the natural number b is a natural number b(1);

the computing apparatus comprises an input information providing unit outputting first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ that correspond to the ciphertext λ_ι and are elements of the group H_ι;

the capability providing apparatus comprises:

a first output information computing unit using the first input information $\tau_{\iota,1}$ to correctly compute $f_\iota(\tau_{\iota,1})$ with a probability greater than a certain provability and outputting an obtained result of the computation as first output information $z_{\iota,1}$; and

a second output information computing unit using the second input information $\tau_{\iota,2}$ to correctly compute $f_\iota(\tau_{\iota,2})$ with a probability greater than a certain provability and outputting an obtained result of the computation as second output information $z_{\iota,2}$;

the first computing unit generates a computation result $u_\iota=f_\iota(\lambda_\iota)^{b(\iota)}x_{\iota,1}$ from the first output information $z_{\iota,1}$;

the second computing unit generates a computation result $v_\iota=f_\iota(\lambda_\iota)^{a(\iota)}x_{\iota,2}$ from the second output information $z_{\iota,2}$;

the final output unit outputs $u_\iota^{b'(\iota)}V_\iota^{a'(\iota)}$ for integers $a'(\iota)$ and $b'(\iota)$ when the computation results $u_\iota$and $v_\iota$satisfy $u_\iota^{a(\iota)}=v_\iota^{b(\iota)}$, the integers $a'(\iota)$ and $b'(\iota)$ satisfying $a'(\iota)a(\iota)+b'(\iota)b(\iota)=1$;

the decryption control apparatus comprises an output unit outputting a decryption control instruction controlling a decryption process of the computing apparatus to the capability providing apparatus;

the capability providing apparatus further comprises a controller controlling, in accordance with the decryption control instruction, whether or not to output both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ from the first output information computing unit and the second output information computing unit.

18. The proxy computing system according to claim **17**, wherein the decryption control instruction corresponds to any one of the decryption functions f_ι; and

the controller controls whether or not to output both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ that correspond to the decryption function f_ι corresponding to the decryption control instruction.

19. The proxy computing system according to claim **17**, wherein the computing apparatus further comprises a recovering unit using $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ for each $\iota=1, \ldots, \omega$ output from the final output unit to perform a recovering process for generating a recovered value that is recoverable only if all of decryption values that are obtainable by decrypting the ciphertext λ_ι for each of $\iota=1, \ldots, \omega$ with the decryption key s_ι are obtained.

20. The proxy computing system according to claim **18**, wherein the computing apparatus further comprises a recovering unit using $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ for each $\iota=1, \ldots, \omega$ output from the final output unit to perform a recovering process for generating a recovered value that is recoverable only if all of decryption values that are obtainable by decrypting the ciphertext λ_ι for each of $=1, \ldots, \omega$ with the decryption key s_ι are obtained.

21. The proxy computing system according to any one of claims **17** to **20**,

wherein the computing apparatus comprises a natural number selecting unit selecting at least part of the natural numbers $a(\iota)$ and $b(\iota)$; and

the first input information $\tau_{\iota,1}$ further corresponds to the natural number $b(\iota)$, the second input information $\tau_{\iota,2}$ further corresponds to the natural number $a(\iota)$.

22. The proxy computing system according to claim **21**, wherein the input information providing unit sets pieces of information whose relation with the ciphertext λ_ι is scrambled as the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$.

23. The proxy computing system according to any one of claims **17** to **20**,

wherein the input information providing unit sets pieces of information whose relation with the ciphertext λ_ι is scrambled as the first input information $\tau_{\iota,1}$ and the second input information $\tau_{\iota,2}$.

24. The proxy computing system according to any one of claims **17** to **20**,

wherein the decryption function f_ι is a homomorphic function, the group $H_\iota$is a cyclic group, a generator of the cyclic group $H_\iota$is $\mu_{\iota,h}$, the order of the cyclic group H_ι is $K_{\iota,H}$, and $v_\iota=f_\iota(\mu_{\iota,h})$;

the input information providing unit comprises:

a first random number generator generating a random number $r(\iota, 1)$ that is a natural number greater than or equal to 0;

a first input information computing unit computing $\mu_{\iota,h}^{r(\iota,1)}\lambda_\iota^{b(\iota)}$ as the first input information $\tau_{\iota,1}$;

a second random number generator generating a random number $(\iota, 2)$ that is a natural number greater than or equal to 0; and

a second input information computing unit computing $\mu_{\iota,h}^{r(\iota,2)}\lambda_\iota^{a(\iota)}$ as the second input information $\tau_{\iota,2}$;

the first output information computing unit uses the first input information $\mu_{\iota,h}^{r(\iota,1)}\lambda_\iota^{b(\iota)}$ to correctly compute $f_\iota(\mu_{\iota,h}^{r(\iota,1)}\lambda_\iota^{b(\iota)})$ with a probability greater than a certain

probability and sets an obtained result of the computation as the first output information $z_{\iota,1}$;

the second output information computing unit uses the second input information $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ to correctly compute $f_{\iota}(\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information $z_{\iota,2}$,

the first computing unit computes $z_{\iota,1}v_{\iota}^{-r(\iota,1)}$ to obtain the computation result u_{ι}; and

the second computing unit computes $z_{\iota,2}v_{\iota}^{-r(\iota,2)}$ to obtain the computation result v_{ι}.

25. The proxy computing system according to claim **24**,

wherein the first random number generator generates the random number r(ι, 1) when b(ι)≠1;

the first input information computing unit computes the first input $\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)}$ as the information $\tau_{\iota,1}$ when b(ι)≠1;

the first output information computing unit sets, as the first output information $z_{\iota,1}$, the computation result obtained by using the first input information $\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)}$ when b(ι)≠1;

the first computing unit computes $z_{\iota,1}v_{\iota}^{-r(\iota,1)}$ to obtain the computation result u_{ι} when b(ι)≠1;

the second random number generator generates the random number r(ι, 2) when a(ι)≠1;

the second input information computing unit computes $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ as the second input information $\tau_{\iota,2}$ when a(ι)≠1;

the second output information computing unit sets, as the second output information $z_{\iota,2}$, the computation result obtained by using the second input information $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ when a(ι)≠1;

the second computing unit computes $z_{\iota,2}v_{\iota}^{-r(\iota,2)}$ to obtain the computation result $v_{\iota}$when a(ι)≠1;

the input information providing unit comprises:

a third random number generator generating a random number r(ι, 3) that is a natural number greater than or equal to 0; and

a third input information computing unit setting $\lambda_{\iota}^{r(\iota,3)}$ as the first input information $\tau_{\iota,1}$ when b(ι)=1 and sets $\lambda_{\iota}^{r(\iota,3)}$ as the second input information $\tau_{\iota,2}$ when a(ι)=1;

the capability providing apparatus comprises a third output information computing unit using the $x^{r(\iota,3)}$ to correctly compute $f_{\iota}(\lambda_{\iota}^{r(\iota,3)})$ with a probability greater than a certain probability and sets an obtained result of the computation as third output information $z_{\iota,3}$; and

the computing apparatus comprises a third computing unit setting $z_{\iota,3}^{1/r(\iota,3)}$ as the computation result $u_{\iota}$when b(ι)=1 and setting $z_{\iota,3}^{1/r(\iota,3)}$ as the computation result $v_{\iota}$when a(ι)=1.

26. The proxy computing system according to claim **21**,

wherein the decryption function $f_{\iota}$is a homomorphic function, the group H_{ι} is a cyclic group, a generator of the cyclic group H_{ι} is $\mu_{\iota,h}$, the order of the cyclic group H_{ι} is $K_{\iota,H}$, and $v_{\iota}=f_{\iota}(\mu_{\iota,h})$;

the input information providing unit comprises:

a first random number generator generating a random number r(ι, 1) that is a natural number greater than or equal to 0;

a first input information computing unit computing $\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)}$ as the first input information $\tau_{\iota,1}$;

a second random number generator generating a random number r(ι, 2) that is a natural number greater than or equal to 0; and

a second input information computing unit computing $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ as the second input information $\tau_{\iota,2}$;

the first output information computing unit uses the first input information $\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)}$ to correctly compute $f_{\iota}(\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the first output information $z_{\iota,1}$;

the second output information computing unit uses the second input information $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ to correctly compute $f_{\iota}(\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information $z_{\iota,2}$,

the first computing unit computes $z_{\iota,1}v_{\iota}^{-r(\iota,1)}$ to obtain the computation result u_{ι}, and

the second computing unit computes $z_{\iota,2}v_{\iota}^{-r(\iota,2)}$ to obtain the computation result v_{ι}.

27. The proxy computing system according to claim **26**,

wherein the first random number generator generates a random number r(ι, 1) when b(ι)≠1;

the first input information computing unit computes the $\mu_{\iota,h}^{r(\iota,1)}\lambda_{\iota}^{b(\iota)}$ as the first input information $\tau_{\iota,1}$ when b(ι)≠1;

the first output information computing unit sets, as the first output information $z_{\iota,1}$, the computation result obtained by using the first input information $\mu_{\iota,h}^{r(\iota,1)}$ $\lambda_{z}^{b(\iota)}$ when b(ι)≠1;

the first computing unit computes $z_{\iota,1}v_{\iota}^{-r(\iota,1)}$ to obtain the computation result $u_{\iota}$when b(ι)≠1;

the second random number generator generates the random number r(ι, 2) when a(ι)≠1;

the second input information computing unit computes $\mu_{\iota,h}^{r(\iota,2)}\lambda_{\iota}^{a(\iota)}$ as the second input information $\tau_{\iota,2}$ when a(ι)≠1;

the second output information computing unit sets, as second output information $z_{\iota,2}$, the computation result obtained by using the second input information $\mu_{\iota,h}^{r(\iota,2)}\lambda_{z}^{a(\iota)}$ when a(ι)≠1;

the second computing unit computes $z_{\iota,2}v_{\iota}^{-r(\iota,2)}$ to obtain the computation result v_{ι} when a(ι)≠1;

the input information providing unit comprises:

a third random number generator generating a random number r(ι, 3) that is a natural number greater than or equal to 0; and

a third input information computing unit setting $\lambda_{\iota}^{r(\iota,3)}$ as the first input information $\tau_{\iota,1}$ when b(ι)=1 and sets $\lambda_{\iota}^{r(\iota,3)}$ as the second input information $\tau_{\iota,2}$ when a(ι)=1;

the capability providing apparatus comprises:

a third output information computing unit using the $X^{r(\iota,3)}$ to correctly compute $f_{\iota}(\lambda_{\iota}^{r(\iota,3)})$ with a probability greater than a certain probability and setting an obtained result of the computation as third output information $z_{\iota,3}$; and

the computing apparatus comprises a third computing unit setting $z_{\iota,3}^{1/r(\iota,3)}$ as the computation result $u_{\iota}$when b(ι)=1 and sets $z_{\iota,3}^{1/r(\iota,3)}$ as the computation result v_{ι} when a(ι)=1.

28. The proxy computing system according to any one of claims **17** to **20**,

wherein the group $H_{\iota}$is a direct product group $G_{\iota}\times G_{\iota}$, the decryption function $f_{\iota}$is a homomorphic function, the group $G_{\iota}$is a cyclic group, a generator of the cyclic group G_{ι} is $\mu_{\iota,g}$, the order of the cyclic group $G_{\iota}$is $K_{\iota,G}$, $\lambda_{\iota}=(c_{\iota,1}, c_{\iota,2})$, (V_{ι}, W_{ι}) is an element of the group H_{ι}, and $f_{\iota}(V_{\iota}, W_{\iota})=Y_{\iota}$;

the input information providing unit comprises:

a fourth random number generator generating a random number r(ι, 4) that is a natural number greater than or equal to 0;

a fifth random number generator generating a random number r(ι, 5) that is a natural number greater than or equal to 0;

a first input information computing unit computing $c_{\iota,2}^{b(\iota)}W_\iota^{r(\iota,4)}$ and $c_{\iota,1}^{b(\iota)}V_\iota^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}$ as the first input information $\tau_{\iota,1}$;

a sixth random number generator generating a random number r(ι, **6**) that is a natural number greater than or equal to 0;

a seventh random number generator generating a random number r(ι, 7) that is a natural number greater than or equal to 0; and

a second input information computing unit computing $c_{\iota,2}^{a(\iota)W^{r(\iota,6)}}$ and $c_{\iota,1}^{a(\iota)}V_\iota^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)}$ as the second input information $\tau_{\iota,2}$;

the first output information computing unit uses the first input information $c_{\iota,1}^{b(\iota)}V_\iota^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}$ and $c_{\iota,2}^{b(\iota)}W_\iota^{r(\iota,4)}$ to correctly compute $f_\iota(c_{\iota,1}^{b(\iota)V_\iota^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}}, c_{\iota,2}^{b(\iota)}W_\iota^{r(\iota,4)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the first output information $z_{\iota,1}$;

the second output information computing unit uses the second input information $c_{\iota,1}^{a(\iota)}V_\iota^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)}$ and $c_{\iota,2}^{a(\iota)}W_\iota^{r(\iota,6)}$ to correctly compute $f_\iota(c_{\iota,1}^{a(\iota)}V\iota^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)}, c_{\iota,2}^{a(\iota)}W_\iota^{r(\iota,6)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information $z_{\iota,2}$;

the first computing unit computes $z_{\iota,1}Y_\iota^{r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ to obtain the computation result u_ι; and

the second computing unit computes $z_{\iota,2}Y_\iota^{-r(\iota,6)}\mu_{\iota,g}^{-r(t,7)}$ to obtain the computation result v_ι.

29. The proxy computing system according to claim **21**,

wherein the group H_ι is a direct product group $G_\iota \times G_\iota$, the decryption function f_ι is a homomorphic function, the group G_ι is a cyclic group, a generator of the cyclic group $G_\iota$is $\mu_{\iota,g}$, the order of the cyclic group G_ι is $K_{\iota,G}$, $\lambda_\iota=(c_{\iota,1}, c_{\iota,2})$, (V_ι, W_ι) is an element of the group H_ι, and $f_\iota(V_\iota, W_\iota)=Y_\iota$;

the input information providing unit comprises:

a fourth random number generator generating a random number r(ι, 4) that is a natural number greater than or equal to 0;

a fifth random number generator generating a random number r(ι, 5) that is a natural number greater than or equal to 0;

a first input information computing unit computing $c_{\iota,2}^{b(\iota)}W_\iota^{r(\iota,4)}$ and $c_{\iota,1}^{b(\iota)}V_\iota^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}$ as the first input information $\tau_{\iota,1}$;

a sixth random number generator generating a sixth random number r(ι, 6) that is a natural number greater than or equal to 0;

a seventh random number generator generating a random number r(ι, 7) that is a natural number greater than or equal to 0; and

a second input information computing unit computing $c_{\iota,2}^{a(\iota)}W_\iota^{r(\iota,6)}$ and $c_{\iota,1}^{a(\iota)}V_\iota^{r(t,6)}\mu_{\iota,g}^{r(\iota,7)}$ as the second input information $\tau_{\iota,2}$;

the first output information computing unit uses the first input information $c_{\iota,1}^{b(\iota)}V_\iota^{r(\iota,4)})\mu_{\iota,g}^{r(t,5)}$ and $c_{t,2}^{b(\iota)}W_\iota^{r(\iota,4)}$ to correctly compute $f_\iota(c_{\iota,1}^{b(\iota)}V_\iota^{r(\iota,4)}\mu_{\iota,g}^{r(\iota,5)}, c_{\iota,2}^{b(\iota)}W_\iota^{t(\iota,4)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the first output information $z_{\iota,1}$;

the second output information computing unit uses the second input information $c_{\iota,1}^{a(\iota)}V_\iota^{r(\iota,6)}\mu\iota,g^{r(\iota,7)}$ and $c_{\iota,2}^{a(\iota)}W_\iota^{r(\iota,6)}$ to correctly compute $f_\iota(c_{\iota,1}^{a(\iota)}V_\iota^{r(\iota,6)}\mu_{\iota,g}^{r(\iota,7)}, c_{\iota,2}^{a(\iota)}W_\iota^{r(\iota,6)})$ with a probability greater than a certain probability and sets an obtained result of the computation as the second output information $z_{\iota,2}$;

the first computing unit computes $Z_{\iota,1}Y_\iota^{-r(\iota,4)}\mu_{\iota,g}^{-r(\iota,5)}$ to obtain the computation result u_ι; and

the second computing unit computes $z_{\iota,2}Y_\iota^{-r(\iota,6)}\mu_{\iota,g}^{-r(\iota,7)}$ to obtain the computation result v_ι.

30. The proxy computing system according to any one of claims **17** to **20**,

wherein a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ does not depend on the natural number b(ι) and/or a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ does not depend on the natural number a(ι), or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ and does not depend on the natural number b(ι), and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ and does not depend on the natural number a(ι).

31. The proxy computing system according to claim **24**,

wherein a probability distribution of an error of the computation result $u_\iota$for $f_\iota(\lambda_\iota)^{b(\iota)}$ does not depend on the natural number b(ι) and/or a probability distribution of an error of the computation result $v_\iota$for $f_\iota(\lambda_\iota)^{a(\iota)}$ does not depend on the natural number a(ι), or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ and does not depend on the natural number b(ι), and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ and does not depend on the natural number a(ι).

32. The proxy computing system according to claim **26**,

wherein a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ does not depend on the natural number b(ι) and/or a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ does not depend on the natural number a(ι), or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ and does not depend on the natural number b(ι), and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result $v_\iota$for $f_\iota(\lambda_\iota)^{a(\iota)}$ and does not depend on the natural number a(ι).

33. The proxy computing system according to claim **28**,

wherein a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ does not depend on the natural number b(ι) and/or a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ does not depend on the natural number a(ι), or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ and does not depend on the natural number b(ι), and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result $v_\iota$for $f_\iota(\lambda_\iota)^{a(\iota)}$ and does not depend on the natural number a(ι).

34. The proxy computing system according to claim **29**,

wherein a probability distribution of an error of the computation result $u_\iota$for $f_\iota(\lambda_\iota)^{b(\iota)}$ does not depend on the natural number b(ι) and/or a probability distribution of an error of the computation result $v_\iota$for $f_\iota(\lambda_\iota)^{a(\iota)}$ does not depend on the natural number a(ι), or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation

result u_ι for $f_\iota(\lambda_\iota)^{b(\iota)}$ and does not depend on the natural number b(ι), and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v_ι for $f_\iota(\lambda_\iota)^{a(\iota)}$ and does not depend on the natural number a(ι).

35. The proxy computing system according to any one of claims **17** to **20**,

wherein the natural number a(ι) or the natural number b(ι) is a constant.

36. The proxy computing system according to claim **24**, wherein the natural number a(ι) or the natural number b(ι) is a constant.

37. The proxy computing system according to claim **26**, wherein the natural number a(ι) or the natural number b(ι) is a constant.

38. The proxy computing system according to claim **28**, wherein the natural number a(ι) or the natural number b(ι) is a constant.

39. The proxy computing system according to claim **29**, wherein the natural number a(ι) or the natural number b(ι) is a constant.

40. A computing apparatus comprising processing circuitry configured to implement a first computing unit, a second computing unit, an input information providing unit, and a determining unit, and a final output unit,

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a and b are natural numbers that are relatively prime to each other;

the input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

the first computing unit generates a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generates a computation result $u'=u^a$;

the second computing unit generates a computation result $v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability, and generates a computation result $v'=v^b$;

the determining unit determines whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

the final output unit outputs $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

41. The computing apparatus according to claim **40**, further comprising a natural number selecting unit selecting at least one of the natural numbers a and b,

wherein the first input information τ_1 further corresponds to the natural number b and the second input information τ_2 further corresponds to the natural number a.

42. The computing apparatus according to claim **41**,

wherein the input information providing unit sets pieces of information whose relation with the ciphertext x is scrambled as the first input information τ_1 and the second input information τ_2.

43. The computing apparatus according to claim **40**,

wherein the input information providing unit sets pieces of information whose relation with the ciphertext x is scrambled as the first input information τ_1 and the second input information τ_2.

44. The computing apparatus according to any one of claims **40**, **41**, **42**, and **43**,

wherein the decryption function f(x) is a homomorphic function, the group H is a cyclic group, a generator of the cyclic group H is μ_h, the order of the cyclic group H is K_H, and $v=f(\mu_h)$;

the input information providing unit comprises:

a first random number generator generating a random number r_1 that is a natural number greater than or equal to 0;

a first input information computing unit computing $\mu_h^{r1}x^b$ as the first input information τ_1;

a second random number generator generating a random number r_2 that is a natural number greater than or equal to 0; and

a second input information computing unit computing $\mu_h^{r2}x^a$ as the second input information τ_2;

the first output information z_1 is a computation result obtained by correctly computing $f(\mu_h^{r1}x^b)$ with a probability greater than a certain probability;

the second output information z_2 is a computation result obtained by correctly computing $f(\mu_h^{r2}x^a)$ with a probability greater than a certain probability;

the computation result u is z_1v^{-r1}; and

the computation result v is z_2v^{-r2}.

45. The computing apparatus according to any one of claims **40**, **41**, **42**, and **43**,

wherein the group His a direct product group G×G, the decryption function f(x) is a homomorphic function, the group G is a cyclic group, a generator of the cyclic group G is μ_g, the order of the cyclic group G is K_G, $x=(c_1, c_2)$, (V, W) is an element of the group H, and f(V, W)=Y;

the input information providing unit comprises:

a fourth random number generator generating a random number r_4 that is a natural number greater than or equal to 0;

a fifth random number generator generating a random number r_5 that is a natural number greater than or equal to 0;

a first input information computing unit computing $c_2^bW^{r4}$ and $c_1^bV^{r4}\mu_g^{r5}$ as the first input information τ_1;

a sixth random number generator generating a random number r_6 that is a natural number greater than or equal to 0;

a seventh random number generator generating a random number r_7 that is a natural number greater than or equal to 0; and

a second input information computing unit computing $c_2^aW^{r6}$ and $c_1^aV^{r6}\mu_g^{r7}$ as the second input information τ_2;

the first output information z_1 is a computation result obtained by correctly computing $f(c_1{}^b V^{r4} \mu_g{}^{r5}, c_2{}^b W^{r4})$ with a probability greater than a certain probability;

the second output information z_2 is a computation result obtained by correctly computing $f(c_1{}^a V^{r6} \mu_g{}^{r7}, c_2{}^a W^{r6})$ with a probability greater than a certain probability;

the computation result u is $z_1 Y^{-r4} \mu_g{}^{-r5}$; and

the computation result v is $z_2 Y^{-r6} \mu_g{}^{-r7}$.

46. The computing apparatus according to any one of claims **40**, **41**, **42**, and **43**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for $f(x)^a$ and does not depend on the natural number a.

47. The computing apparatus according to claim **44**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for f(x)a and does not depend on the natural number a.

48. The computing apparatus according to claim **45**,

wherein a probability distribution of an error of the computation result u for $f(x)^b$ does not depend on the natural number b and/or a probability distribution of an error of the computation result v for $f(x)^a$ does not depend on the natural number a, or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result u for $f(x)^b$ and does not depend on the natural number b, and/or there is a probability distribution that is indistinguishable from a probability distribution of an error of the computation result v for $f(x)^a$ and does not depend on the natural number a.

49. The computing apparatus according to any one of claims **40**, **41**, **42**, and **43**,

wherein the natural number a or the natural number b is a constant.

50. The computing apparatus according to claim **44**, wherein the natural number a or the natural number b is a constant.

51. The computing apparatus according to claim **45**, wherein the natural number a or the natural number b is a constant.

52. A proxy computing method comprising the steps of:

on a computing apparatus, outputting first input information τ_1 and second input information τ_2 that correspond to a ciphertext x and are elements of a group H;

on a capability providing apparatus, using the first input information τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets an obtained result of the computation as first output information z_1;

on the capability providing apparatus, using the second input information τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets an obtained result of the computation as second output information z_2;

on the computing apparatus, generating a computation result $u=f(x)^b x_1$ from the first output information z_1 and generating a computation result $u'=u^a$;

on the computing apparatus, generating a computation result $v=f(x)^a x_2$ from the second output information z_2 and generating a computation result $v'=v^b$;

on the computing apparatus, determining whether or not there are values of u' and v' that satisfy u'=v' after every calculation of and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

on the computing apparatus, outputting $u^{b'} v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy u'=v' and thus $u^a=v^b$;

wherein G is a group, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a and b are natural numbers that are relatively prime to each other, and

wherein the computing apparatus determines that calculation of $u^{b'} v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

53. The proxy computing method according to claim **52**, further comprising the steps of:

by a decryption control apparatus, outputting a decryption control instruction controlling a decryption process of the computing apparatus to the capability providing apparatus; and

by the capability providing apparatus, controlling, in accordance with the decryption control instruction, whether or not to output both of the first output information z_1 and the second output information z_2 from the first output information computing unit and the second output information computing unit.

54. The proxy computing method according to claim **52**, wherein G_ι and H_ι are groups, co is an integer greater than or equal to 2, $\iota=1, \ldots, \omega$, $f_\iota(\lambda_\iota)$ is a decryption function for decrypting a ciphertext λ_ι which is an element of the group H_ι with a particular decryption key s, to obtain an element of the group G_ι, $X_{\iota,1}$ and $\lambda_{\iota,2}$ are random variables having values in the group G_ι, $x_{\iota,1}$ is a realization of the random variable $X_{\iota,1}$, $x_{\iota,2}$ is a realization of the random variable $X_{\iota,2}$, $a(\iota)$ and $b(\iota)$ are natural numbers that are relatively prime to each other, the group G is a group G_1, the group H is a group H_1, the ciphertext x is a ciphertext λ_1, the decryption function f(x) is a decryption function $f_1(\lambda_1)$, the random variable X_1 is a random variable $X_{1,1}$, the random variable X_2 is a random variable $X_{1,2}$, the realization x_1 is a realization $x_{1,1}$, the realization x_2 is a realization $x_{1,2}$, the natural number a is a natural number a(1), and the natural number b is a natural number b(1);

the proxy computing method comprises the steps of:

(A) on the computing apparatus, outputting first input information $\tau_{\iota,1}$ and second input information $\tau_{\iota,2}$ that correspond to the ciphertext λ_ι and are elements of the group H_ι;

(B) on the capability providing apparatus, using the first input information $\tau_{\iota,1}$ to correctly compute $f_\iota(\tau_{\iota,1})$ with a probability greater than a certain provability and outputting an obtained result of the computation as first output information $z_{\iota,1}$;

(C) on the capability providing apparatus, using the second input information $\tau_{\iota,2}$ to correctly compute $f_\iota(\tau_{\iota,2})$ with a probability greater than a certain provability and outputting an obtained result of the computation as second output information $z_{\iota,2}$;

(D) on the computing apparatus, generating a computation result $u_\iota=f_\iota(\lambda_\iota)^{b(\iota)x_{\iota,1}}$ from the first output information $z_{\iota,1}$;

(E) on the computing apparatus, generating a computation result $v_\iota=f_\iota(\lambda_\iota)^{a(\iota)}x_{\iota,2}$ from the second output information $z_{\iota,2}$; and

(F) on the computing apparatus, outputting $u_\iota^{b'(\iota)}v_\iota^{a'(\iota)}$ for integers a'(ι) and b'(ι) that satisfy a'(ι)a(ι)+b'(ι)b(ι)=1 when the computation results u_ι and v_ι satisfy $u_\iota^{a(\iota)}=v_\iota^{b(\iota)}$;

the steps (B) and (C) controlling whether or not to output both of the first output information $z_{\iota,1}$ and the second output information $z_{\iota,2}$ from the first output information computing unit and the second output information computing unit, in accordance with a decryption control instruction output from a decryption control apparatus for controlling a decryption process of the computing apparatus.

55. A computing method comprising the steps of:

by an input information providing unit, outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

by a first computing unit, generating a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generating a computation result $u'=u^a$;

by a second computing unit, generating a computation result $v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability, and generating a computation result $v'=v^b$;

by a determination unit, determining whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

by a final output unit, outputting $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy $u^a=v^b$;

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a and b are natural numbers that are relatively prime to each other,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

56. A non-transitory computer-readable recording medium storing a program for causing a computer to function as a computing apparatus comprising processing circuitry configured to implement a first computing unit, a second computing unit, an input information providing unit, and a determining unit, and a final output unit,

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a and b are natural numbers that are relatively prime to each other;

the input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

the first computing unit generates a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generates a computation result $u'=u^a$;

the second computing unit generates a computation result $v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability, and generates a computation result $v'=v^b$;

the determining unit determines whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

the final output unit outputs $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

57. A proxy computing system comprising a computing apparatus and a capability providing apparatus,

wherein G and H are groups, f(x) is a decryption function for decrypting a certain ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a is a natural number, and b=1;

the computing apparatus comprises processing circuitry configured to implement an input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H;

the capability providing apparatus comprises processing circuitry configured to implement:

a first output information computing unit using the first input information τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets an obtained result of the computation as first output information z_1; and

a second output information computing unit using the second input information τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets an obtained result of the computation as second output information z_2, and

the processing circuitry of the computing apparatus is further configured to implement:

a first computing unit configured to generate a computation result $u=f(x)^b x_1$ from the first output information z_1, and generate a computation result $u'=u^a$;

a second computing unit configured to generate a computation result $v'=v=f(x)^a x_2$ from the second output information z_2,

a determining unit configured to determine whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

a final output unit configured to output $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy u'=v' and thus $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

58. A computing apparatus comprising processing circuitry configured to implement a first computing unit, a second computing unit, an input information providing unit, and a determining unit, and a final output unit,

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a is a natural number, and b=1;

the input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

the first computing unit generates a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generates a computation result $u'=u^a$;

the second computing unit generates a computation result $v'=v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability;

the determining unit determines whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

the final output unit outputs $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

59. A proxy computing method comprising the steps of:

on a computing apparatus, outputting first input information τ_1 and second input information τ_2 that correspond to a ciphertext x and are elements of a group H;

on a capability providing apparatus, using the first input information τ_1 to correctly compute $f(\tau_1)$ with a probability greater than a certain probability and sets an obtained result of the computation as first output information z_1;

on the capability providing apparatus, using the second input information τ_2 to correctly compute $f(\tau_2)$ with a probability greater than a certain probability and sets an obtained result of the computation as second output information z_2;

on the computing apparatus, generating a computation result $u=f(x)^b x_1$ from the first output information z_1 and generating a computation result $u'=u^a$;

on the computing apparatus, generating a computation result $v'=v=f(x)^a x_2$ from the second output information z_2;

on the computing apparatus, determining whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

on the computing apparatus, outputting $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy u'=v' and thus $u^a=v^b$;

wherein G is a group, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a is a natural number, and b=1, and

wherein the computing apparatus determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

60. A computing method comprising the steps of:

by an input information providing unit, outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

by a first computing unit, generating a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generating a computation result $u'=u^a$;

by a second computing unit, generating a computation result $v'=v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability;

by a determination unit, determining whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

by a final output unit, outputting $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1 when the computation results u and v satisfy $u^a=v^b$;

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a is a natural number, and b=1,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

61. A non-transitory computer-readable recording medium storing a program for causing a computer to function as a computing apparatus comprising processing circuitry configured to implement a first computing unit, a second computing unit, an input information providing unit, and a determining unit, and a final output unit,

wherein G and H are groups, f(x) is a decryption function for decrypting a ciphertext x which is an element of the group H with a particular decryption key to obtain an element of the group G, X_1 and X_2 are random variables having values in the group G, x_1 is a realization of the random variable X_1, x_2 is a realization of the random variable X_2, a is a natural number, and b=1;

the input information providing unit outputting first input information τ_1 and second input information τ_2 that correspond to the ciphertext x and are elements of the group H to a capability providing apparatus,

the first computing unit generates a computation result $u=f(x)^b x_1$ from a first output information z_1 received from the capability providing apparatus, the first output information z_1 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_1)$ with a probability greater than a certain probability, and generates a computation result $u'=u^a$;

the second computing unit generates a computation result $v'=v=f(x)^a x_2$ from a second output information z_2 received from the capability providing apparatus, the second output information z_2 being a computation result obtained by the capability providing apparatus correctly computing $f(\tau_2)$ with a probability greater than a certain probability;

the determining unit determines whether or not there are values of u' and v' that satisfy u'=v' after every calculation of u' and v' for no more than a predetermined number of iterations of receiving newly calculated values of u and v from the calculating apparatus; and

the final output unit outputs $u^{b'}v^{a'}$ for integers a' and b' that satisfy a'a+b'b=1

when the computation results u and v satisfy $u^a=v^b$,

wherein the determining unit determines that calculation of $u^{b'}v^{a'}$ is impossible based on a calculation reliability of the capability providing apparatus being lower than a reference value when there are no values of u' and v' that satisfy u'=v' after the predetermined number of iterations are completed.

* * * * *