

(12) 发明专利

(10) 授权公告号 CN 101300775 B

(45) 授权公告日 2012. 12. 19

(21) 申请号 200680040882. 8

(51) Int. Cl.

(22) 申请日 2006. 10. 24

H04L 9/08(2006. 01)

(30) 优先权数据

316105/2005 2005. 10. 31 JP

G09C 1/00(2006. 01)

(85) PCT申请进入国家阶段日

2008. 04. 30

(56) 对比文件

(86) PCT申请的申请数据

PCT/JP2006/321090 2006. 10. 24

JP 2002368735 A, 2002. 12. 20, 说明书第 [0033]-[0039] 段 .

(87) PCT申请的公布数据

W02007/052491 JA 2007. 05. 10

US 6658569 B1, 2003. 12. 02, 全文 .
US 6278783 B1, 2001. 08. 21, 摘要, 第 1 栏第 65 行 - 第 2 栏第 9 行 .

(73) 专利权人 松下电器产业株式会社

JP 2002368735 A, 2002. 12. 20, 说明书第 [0033]-[0039] 段 .

地址 日本大阪府

JP 2002368735 A, 2002. 12. 20, 说明书第 [0033]-[0039] 段 .

(72) 发明人 芳贺智之 佐藤太一 浅井理惠子

US 6278783 B1, 2001. 08. 21, 摘要, 第 1 栏第 65 行 - 第 2 样第 9 行 .

(74) 专利代理机构 永新专利商标代理有限公司

审查员 李萍

72002

代理人 许玉顺 胡建新

权利要求书 4 页 说明书 32 页 附图 17 页

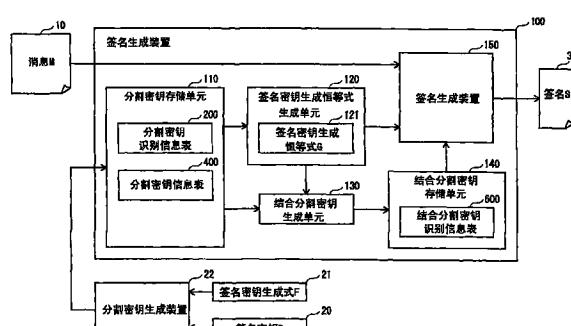
(54) 发明名称

安全处理装置、安全处理方法、加密信息嵌入方法、程序、存储介质和集成电路

(57) 摘要

本发明的目的在于提供一种安全处理装置、安全处理方法、加密信息嵌入方法。在对信息使用签名密钥实施签名的签名生成单元中，分割密钥存储单元存储已将签名密钥至少分割为 2 个的分割秘密密钥、从上述分割秘密密钥计算出上述签名密钥的签名密钥生成式 F 和签名生成式，签名密钥生成恒等式生成单元使用结合法则、分配法则和交换法则，生成与签名密钥生成式 F 得到相同结果的签名密钥生成恒等式 G，结合分割密钥生成单元生成需要给予签名密钥生成恒等式 G 作为自变量的、运算了上述分割秘密密钥后的结果的结合分割密钥，签名生成单元基于上述签名密钥生成恒等式 G 和上述分割秘密密钥，对上述信息实施签名。

CN 101300775 B



1. 一种安全处理装置,能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,该安全处理装置的特征在于,具有:

存储单元,存储有已预先将秘密信息至少分割为2个的分割秘密信息、第一秘密信息生成式和第一安全运算过程,在所述第一秘密信息生成式中作为自变量而被输入上述分割秘密信息,所述第一秘密信息生成式根据上述分割秘密信息计算出上述秘密信息,所述第一安全运算过程示出使用了上述分割秘密信息的安全运算的过程;

结合秘密信息生成单元,生成将至少2个上述分割秘密信息相互进行运算的结合秘密信息;

第一生成单元,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式,该第二秘密信息生成式以上述结合秘密信息为自变量;

第二生成单元,基于上述第二秘密信息生成式中包含的算子,生成示出与上述第一安全运算过程等效的运算过程的第二安全运算过程,该第二安全运算过程以上述结合秘密信息为自变量;以及

执行单元,对上述消息实施按照上述第二安全运算过程的安全运算;

将上述结合秘密信息和上述第二秘密信息生成式动态地决定,以使上述结合秘密信息和上述第二秘密信息生成式在对上述消息执行安全处理之前总是不固定。

2. 如权利要求1所述的安全处理装置,其特征在于,

上述第一秘密信息生成式包含1个以上的运算,

上述第一生成单元从上述第一秘密信息生成式中包含的运算中,随机选择与上述运算之间满足交换法则、结合法则、分配法则中的任一个的替代运算的某个运算,通过将选择的运算置换为上述替代运算,生成第二秘密信息生成式。

3. 如权利要求2所述的安全处理装置,其特征在于,

上述第一秘密信息生成式包含1个以上的运算,各运算包含多个操作数和示出操作数相互之间的计算内容的算子,

上述存储单元存储有示出上述第一秘密信息生成式涉及的上述操作数与上述算子的关系的属性信息,

上述第一生成单元使用上述属性信息,生成上述第二秘密信息生成式。

4. 如权利要求3所述的安全处理装置,其特征在于,

在上述第一秘密信息生成式中,向上述操作数输入自变量的值,上述自变量各自的值不重复地对应于上述分割秘密信息的某一个,

上述属性信息示出对应于能结合的多个操作数的算子,

上述第一生成单元基于上述属性信息,用上述算子结合上述第一秘密信息生成式中的上述多个操作数,

通过使用上述算子结合与由上述第一生成单元结合的各操作数相对应的分割秘密信息,来生成作为上述第二生成单元的自变量的结合秘密信息。

5. 如权利要求2所述的安全处理装置,其特征在于,

上述第二秘密信息生成式包含1个以上的运算,各运算包含多个操作数和示出操作数相互之间的计算内容的算子,

上述存储单元存储有属性信息,该属性信息表示上述第二秘密信息生成式涉及的上述

操作数与上述算子的关系，

上述第二生成单元使用上述属性信息，生成上述第二安全运算过程。

6. 如权利要求 1 所述的安全处理装置，其特征在于，

上述第一生成单元进一步生成随机数信息，生成包含了上述随机数信息的上述第二秘密信息生成式，

上述第二生成单元基于上述第二秘密信息生成式，使用上述结合秘密信息和上述随机数信息生成第二安全运算过程，该第二安全运算过程示出与上述第一安全运算过程等效的运算过程，

上述执行单元使用上述结合秘密信息和上述随机数信息，对上述消息实施按照上述第二安全运算过程的安全运算。

7. 如权利要求 1 所述的安全处理装置，其特征在于，

上述第一生成单元进一步生成使用了上述秘密信息的、不对运算处理的结果造成影响的冗余信息，使用该冗余信息生成上述第二秘密信息生成式。

8. 如权利要求 1 所述的安全处理装置，其特征在于，

上述存储单元存储有上述分割秘密信息和上述执行单元不使用的伪信息。

9. 如权利要求 1 所述的安全处理装置，其特征在于，

上述秘密信息是用于生成数字签名的签名密钥，

上述安全运算是对上述消息实施数字签名的签名生成运算。

10. 如权利要求 9 所述的安全处理装置，其特征在于，

上述签名生成运算是 RSA(Rivest Shamir Adleman) 签名生成处理。

11. 如权利要求 9 所述的安全处理装置，其特征在于，

上述签名生成运算是利用椭圆曲线数字签名方式的签名处理。

12. 如权利要求 11 所述的安全处理装置，其特征在于，

上述安全处理装置进一步具有生成随机数信息的随机数信息生成单元，

上述执行单元不直接利用随机数 k，而是使用至少 2 个上述随机数信息，进行上述椭圆曲线数字签名方式中的计算定义体 GF(p) 上的椭圆曲线的位数为 q 的基点 P 的随机数值 k 倍点的处理、和使用了定义体 GF(q) 上的 k 的倒数的值的处理。

13. 如权利要求 1 所述的安全处理装置，其特征在于，

上述秘密信息是公开密钥密码的秘密密钥，

上述执行单元进行利用了公开密钥和秘密密钥的公开密钥密码系统的处理，作为上述安全运算。

14. 如权利要求 13 所述的安全处理装置，其特征在于，

上述公开密钥密码是 RSA 密码。

15. 如权利要求 13 所述的安全处理装置，其特征在于，

上述公开密钥密码是椭圆曲线密码。

16. 如权利要求 1 所述的安全处理装置，其特征在于，

上述安全处理装置进一步包括：

取得单元，从外部取得上述第一生成单元的更新用数据；以及

更新单元，使用上述更新用数据，更新上述第一生成单元。

17. 如权利要求 1 所述的安全处理装置,其特征在于,

上述安全处理装置进一步包括 :

取得单元,从外部取得更新用的分割秘密信息;以及

更新单元,将上述存储单元中存储的分割秘密信息,更新为上述取得单元取得的更新用的分割秘密信息。

18. 一种用于安全处理装置的安全处理方法,能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,其特征在于,

上述安全处理装置具有存储单元,该存储单元存储有已预先将秘密信息至少分割为 2 个的分割秘密信息、第一秘密信息生成式和第一安全运算过程,在所述第一秘密信息生成式中作为自变量而被输入上述分割秘密信息,所述第一秘密信息生成式根据上述分割秘密信息计算出上述秘密信息,所述第一安全运算过程示出使用了上述分割秘密信息的安全运算的过程,

上述安全处理方法包括 :

结合秘密信息生成步骤,生成将至少 2 个上述分割秘密信息相互进行运算的结合秘密信息;

第一生成步骤,生成输出与上述第一秘密信息生成式相同的结果的第二秘密信息生成式,该第二秘密信息生成式以上述结合秘密信息为自变量;

第二生成步骤,基于上述第二秘密信息生成式中包含的算子,生成示出与上述第一安全运算过程等效的运算过程的第二安全运算过程,该第二安全运算过程以上述结合秘密信息为自变量;以及

执行步骤,对上述消息实施按照上述第二安全运算过程的安全运算;

将上述结合秘密信息和上述第二秘密信息生成式动态地决定,以使上述结合秘密信息和上述第二秘密信息生成式在对上述消息执行安全处理之前总是不固定。

19. 一种用于安全处理装置的集成电路,该安全处理装置能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,其特征在于,

上述安全处理装置具有存储单元,该存储单元存储有已预先将秘密信息至少分割为 2 个的分割秘密信息、第一秘密信息生成式和第一安全运算过程,在所述第一秘密信息生成式中作为自变量而被输入上述分割秘密信息,所述第一秘密信息生成式根据上述分割秘密信息计算出上述秘密信息,所述第一安全运算过程示出使用了上述分割秘密信息的安全运算的过程;

上述集成电路具有 :

结合秘密信息生成单元,生成将至少 2 个上述分割秘密信息相互进行运算的结合秘密信息;

第一生成单元,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式,该第二秘密信息生成式以上述结合秘密信息为自变量;

第二生成单元,基于上述第二秘密信息生成式中包含的算子,生成示出与上述第一安全运算过程等效的运算过程的第二安全运算过程,该第二安全运算过程以上述结合秘密信息为自变量;以及

执行单元,对上述消息实施按照上述第二安全运算过程的安全运算;

将上述结合秘密信息和上述第二秘密信息生成式动态地决定,以使上述结合秘密信息和上述第二秘密信息生成式在对上述消息执行安全处理之前总是不固定。

安全处理装置、安全处理方法、加密信息嵌入方法、程序、存储介质和集成电路

技术领域

[0001] 本发明涉及防止非法地篡改和分析程序的技术等。

背景技术

[0002] 以前，在检测数据的篡改等的目的中广泛使用着电子签名（以下称作签名）。该签名方法之一有 RSA(Rivest Shanir Adleman) 签名生成法。在 RSA 签名生成法中，通过对签名对象信息 M 使用签名密钥 d 进行 $S = M^d \bmod n$ 的运算，生成签名 S 。

[0003] 上述的签名密钥 d 是必须要保护的信息，以下将这样的信息称作秘密信息。此外，在本说明书中，记号 \wedge 表示乘方运算，记号 $*$ 表示乘法。

[0004] 在此，在执行上述的生成 RSA 签名的处理中，由于在计算机内的 RAM 和 CPU 的寄存器等存储器上出现了签名密钥 d 的值，因此，有通过分析这些存储器来非法取得签名密钥 d 的危险性。

[0005] 作为用于防止这样的非法取得签名密钥 d 的技术之一，在非专利文献 1 中公开了一种不使签名密钥 d 的值出现在存储器上而进行签名生成的方法。

[0006] 在非专利文献 1 的方法中，首先求满足签名密钥生成式 $d = (d1*d2)+d3$ 的 $d1$ 、 $d2$ 、 $d3$ 。在此， $d1$ 、 $d2$ 和 $d3$ 是上述的分割密钥，将根据签名密钥生成分割密钥称作签名密钥的分割。

[0007] 基于该分割密钥 ($d1$ 、 $d2$ 、 $d3$) 和签名密钥生成式，按照

[0008] $S1 = M^d1 \bmod n$

[0009] $S2 = S1^d2 \bmod n$

[0010] $S = S2*M^d3 \bmod n$ 的顺序进行运算。这样，不使用签名密钥 d 而得到与 RSA 签名生成式 $S = M^d \bmod n$ 相同的签名 S 。此外，在签名生成处理中，由于在存储器上不是出现签名密钥 d ，而是出现分割密钥 ($d1$ 、 $d2$ 、 $d3$)，因此，能够保护签名密钥 d 。

[0011] 非专利文献 1：“签名生成软件的随机数据搜索的抗篡改性评价”横滨国立大学本田洋之松本勉 SCIS2005

[0012] 但是，在上述的方法中，由于签名密钥生成式总是固定的，因此，能够通过对签名生成部进行静态分析来确定签名密钥生成式。

[0013] 另外，由于每次利用相同的分割密钥，因此，通过一边改变签名对象数据一边收集多次的签名生成时的随机数据，进行所谓的看收集到的多个随机数据相互之间的差分，抽出不变的数据的动态分析，就能够确定分割密钥。

[0014] 然后，就产生了若使用利用上述方法确定了的分割密钥和签名密钥生成式，就能确定签名密钥的问题。

发明内容

[0015] 鉴于上述问题，本发明的目的在于，提供一种即使是非法分析者进行了静态分析

和动态分析的情况下,也能够隐匿秘密信息的安全处理装置。

[0016] 为了解决上述问题,本发明的一种安全处理装置,得到与对消息实施安全运算的情况相同的运算结果,该安全运算使用秘密信息,具有:存储单元,存储有第一秘密信息生成式和第一安全处理过程,所述第一秘密信息生成式中输入将秘密信息至少分割为2个的分割秘密信息作为自变量,根据该分割秘密信息计算出上述秘密信息,所述第一安全运算过程示出使用了上述秘密信息的安全运算的过程;第一生成单元,作为自变量而输入运算了至少2个上述分割秘密信息的结果即结合秘密信息,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式;第二生成单元,基于上述第二秘密信息生成式中包含的算子,作为自变量而输入上述结合秘密信息,生成第二安全运算过程,该第二安全运算过程示出与上述第一安全运算过程等效的运算过程;执行单元,对上述消息实施按照上述第二安全运算过程的安全运算。

[0017] 本发明的安全处理装置通过具有上述结构,通过在每次执行安全处理时,生成上述结合秘密信息,生成第二安全运算过程,执行第二安全运算过程,由此得到与第一安全运算相同的结果,因此,取代上述秘密信息,在运算用的存储器上出现每次安全运算都成为不同值的结合秘密信息,每次执行安全处理时第二安全运算过程都不同,因此,利用静态分析和动态分析的上述秘密信息的确定变得困难。这样就能够隐匿上述秘密信息。

[0018] 此外,也可以上述第一秘密信息生成式包含1个以上的运算,上述第二生成单元从上述第一秘密信息生成式中包含的运算中,随机选择与上述运算之间满足交换法则、结合法则、分配法则中的任一个的替代运算的某个运算,通过将选择的运算置换为上述替代运算,生成第二秘密信息生成式。

[0019] 此外,也可以上述第一秘密信息生成式包含1个以上的运算,各运算包含多个操作数和示出操作数相互之间的计算内容的算子,上述存储单元存储有示出上述第一秘密信息生成式涉及的上述操作数与上述算子的关系的属性信息,上述第一生成单元使用上述属性信息,生成上述第二秘密信息生成式。

[0020] 此外,也可以在上述第一秘密信息生成式中,向上述操作数输入自变量的值,上述自变量的各个值不重复地对应于上述分割秘密信息的某一个,上述属性信息示出对应于能结合的多个操作数的算子,上述第一生成单元基于上述属性信息,用上述算子结合上述第一秘密信息生成式中的上述多个操作数,上述第二生成单元通过使用上述算子结合与上述已结合的各操作数相对应的分割秘密信息,来生成上述结合秘密信息。

[0021] 此外,也可以上述第二秘密信息生成式包含1个以上的运算,各运算包含多个操作数和示出操作数相互之间的计算内容的算子,上述存储单元存储有属性信息,该属性信息表示上述第二秘密信息生成式涉及的上述操作数与上述算子的关系,上述第二生成单元使用上述属性信息,生成上述第二安全运算过程。

[0022] 根据该结构,由于能够随机地生成得到与第一秘密信息生成式相同结果的第二秘密信息生成式,因此,能够使利用静态分析的秘密信息的确定变得困难。

[0023] 此外,也可以上述第一生成单元进一步生成随机数信息,生成包含了上述随机数信息的上述第二秘密信息生成式,上述第二生成单元基于上述第二秘密信息生成式,使用上述结合秘密信息和上述随机数信息生成第二安全运算过程,该第二安全运算过程示出与上述第一安全运算过程等效的运算过程,上述执行单元对上述消息使用上述结合秘密信息

和上述随机数信息,实施按照上述第二安全运算过程的安全运算。

[0024] 根据该结构,通过包含随机数信息,上述第二秘密信息生成式的分析就变得困难,能够使按照使用随机数信息执行的第二安全运算过程的安全运算的静态分析变得困难。

[0025] 此外,也可以上述第一生成单元进一步生成使用了上述秘密信息的、不对运算处理的结果造成影响的冗余信息,使用该冗余信息生成上述第二秘密信息生成式。

[0026] 根据该结构,由于使用冗余信息,因此能够使第二秘密信息生成式的静态分析变得困难。

[0027] 此外,也可以上述存储单元存储有上述分割秘密信息和上述执行单元不使用的伪信息。

[0028] 根据该结构,能够使基于存储装置中存储着的信息来静态分析安全处理变得困难。

[0029] 此外,也可以上述秘密信息是用于生成数字签名的签名密钥,上述安全运算是对上述消息实施数字签名的签名生成运算。

[0030] 根据该结构,在执行数字签名处理中,能够使利用静态分析的秘密信息的确定变得困难。

[0031] 此外,上述签名生成处理是 RSA(Rivest Shamir Adleman) 签名生成处理。

[0032] 根据该结构,在执行 RSA 签名生成处理中,能够使利用静态分析的秘密信息的确定变得困难。

[0033] 此外,上述签名生成处理也可以是利用椭圆曲线数字签名方式的签名处理。

[0034] 此外,也可以上述安全处理装置进一步具有生成随机数信息的随机数信息生成单元,上述执行单元不直接利用上述随机数 k,而是使用至少 2 个上述随机数信息,进行上述椭圆曲线数字签名方式中的计算定义体 GF(p) 上的椭圆曲线的位数 q 即基点 P 的随机数值 k 倍点的处理、和使用了定义体 GF(q) 上的 k 的逆数(逆数)的值的处理。

[0035] 根据该结构,在执行利用椭圆曲线数字签名方式的签名处理中,能够使利用静态分析的秘密信息的确定变得困难。

[0036] 此外,也可以上述秘密信息是上述公开密钥密码的上述秘密密钥,上述执行单元进行利用了公开密钥和秘密密钥的公开密钥密码系统的处理,作为上述安全运算。

[0037] 根据该结构,在执行利用公开密钥密码的安全处理中,能够使利用静态分析的秘密密钥的确定变得困难。

[0038] 此外,上述公开密钥密码也可以是 RSA 密码。

[0039] 根据该结构,在执行利用 RSA 密码的安全处理中,能够使利用静态分析的秘密密钥的确定变得困难。

[0040] 此外,上述公开密钥密码也可以是椭圆曲线密码。

[0041] 根据该结构,在执行利用椭圆曲线密码的安全处理中,能够使利用静态分析的秘密密钥的确定变得困难。

[0042] 此外,也可以上述安全处理装置进一步包括:取得单元,从外部取得上述第一生成单元的更新用数据;更新单元,使用上述更新用数据,更新上述第一生成单元。

[0043] 根据该结构,通过更新第一生成装置,即使是在执行安全处理时进行静态分析的情况下,也能够使秘密信息的确定变得困难。此外,能从装置外部调整所生成的分割密钥的

个数和结合分割密钥的个数,能灵活地设定安全强度。

[0044] 此外,也可以上述安全处理装置进一步包括:取得单元,从外部取得更新用的分割秘密信息;更新单元,将上述存储单元中存储的分割秘密信息,更新为上述取得单元取得的更新用的分割秘密信息。

[0045] 根据该结构,通过更新分割秘密信息,即使是在执行安全处理时进行静态分析的情况下,也能够使秘密信息的确定变得困难。

[0046] 此外,能从装置外部调整所生成的分割密钥的个数和结合分割密钥的个数,能灵活地设定安全强度。

[0047] 本发明的用于安全处理装置的安全处理方法,能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,上述安全处理装置存储单元,该存储单元存储有第一秘密信息生成式和第一安全处理过程,所述第一秘密信息生成式中输入将秘密信息至少分割为2个的分割秘密信息作为自变量,根据该分割秘密信息计算出上述秘密信息,所述第一安全处理过程示出使用了上述秘密信息的安全运算的过程,上述安全处理方法包括:第一生成步骤,作为自变量而输入运算了至少2个上述分割秘密信息的结果即结合秘密信息,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式;第二生成步骤,基于上述第二秘密信息生成式中包含的算子,作为自变量而输入上述结合秘密信息,生成第二安全运算过程,该第二安全运算过程示出与上述第一安全运算过程等效的运算过程;执行步骤,对上述消息实施按照上述第二安全运算过程的安全运算。

[0048] 本发明的用于安全处理装置的计算机程序,该安全处理装置能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,上述安全处理装置具有存储单元,该存储单元存储有第一秘密信息生成式和第一安全处理过程,所述第一秘密信息生成式中输入将秘密信息至少分割为2个的分割秘密信息作为自变量,根据该分割秘密信息计算出上述秘密信息,所述第一安全处理过程示出使用了上述秘密信息的安全运算的过程,上述计算机程序包括:第一生成步骤,作为自变量而输入运算了至少2个上述分割秘密信息的结果即结合秘密信息,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式;第二生成步骤,基于上述第二秘密信息生成式中包含的算子,作为自变量而输入上述结合秘密信息,生成第二安全运算过程,该第二安全运算过程示出与上述第一安全运算过程等效的运算过程;执行步骤,对上述消息实施按照上述第二安全运算过程的安全运算。

[0049] 本发明的记录介质是计算机可读取的记录介质,记录有上述计算机程序。

[0050] 本发明的用于安全处理装置的集成电路,该安全处理装置能够得到与对消息实施了安全运算的情况相同的运算结果,该安全运算使用秘密信息,其特征在于,上述安全处理装置具有存储单元,该存储单元存储有已预先将秘密信息至少分割为2个的分割秘密信息、第一秘密信息生成式和第一安全运算过程,在所述第一秘密信息生成式中作为自变量而输入上述分割秘密信息,所述第一秘密信息生成式根据上述分割秘密信息计算出上述秘密信息,所述第一安全运算过程示出使用了上述分割秘密信息的安全运算的过程;上述集成电路具有:结合秘密信息生成单元,生成将至少2个上述分割秘密信息相互进行运算的结合秘密信息;第一生成单元,作为自变量而输入上述结合秘密信息,生成在运算上与上述第一秘密信息生成式等效的第二秘密信息生成式;第二生成单元,基于上述第二秘密信息

生成式中包含的算子，作为自变量而输入上述结合秘密信息，生成示出与上述第一安全运算过程等效的运算过程的第二安全运算过程；执行单元，对上述消息实施按照上述第二安全运算过程的安全运算。

[0051] 根据该结构，通过在每次执行安全处理时，生成上述结合秘密信息，并生成第二安全运算过程，执行第二安全运算过程，由此得到与第一安全运算相同的结果，因此，取代上述秘密信息，在运算用的存储器上出现每个安全运算都不同值的结合秘密信息，每次执行安全处理时第二安全运算过程都不同，因此，利用静态分析的上述秘密信息的确定变得困难。这样就能够隐匿上述秘密信息。

[0052] 本发明的加密信息嵌入方法，将上述秘密信息加密后嵌入到进行使用了秘密信息的运算的安全处理装置中，其特征在于，包括：秘密信息加密步骤，使用将上述秘密信息变换为很难分析的状态的加密装置，将上述秘密信息进行加密；嵌入步骤，使用加密信息写入装置，将加密后的秘密信息嵌入到上述安全处理装置中。

[0053] 根据该结构，通过将秘密信息加密后嵌入到安全处理装置中，能够防止暴露秘密信息。

[0054] 此外，也可以上述加密装置具有输入单元，该输入单元接受用于决定上述秘密信息的加密方法的参数作为输入，上述秘密信息加密步骤包括：上述输入单元接受上述参数的接受步骤；用基于上述参数决定的加密方法，将上述秘密信息进行加密的加密步骤。

[0055] 根据该结构，通过从外部给予用于决定上述秘密信息的加密的方法的参数，就能够灵活地设定安全强度。

[0056] 此外，也可以上述参数是上述秘密信息的分割个数，上述加密步骤包括分割步骤，该分割步骤根据上述分割个数，将上述秘密信息分割为至少 2 个分割秘密信息。

[0057] 此外，也可以上述加密步骤进一步包括式生成步骤，该式生成步骤基于上述分割个数，生成至少包含上述分割个数的项的第一秘密信息生成式，上述分割步骤分割上述秘密信息，以便从上述第一秘密信息生成式计算出上述秘密信息。

[0058] 根据该结构，由于能从装置外部调整秘密信息的分割个数，因此，能够灵活地设定安全强度。

[0059] 此外，也可以上述参数是第一秘密信息生成式，上述加密步骤包括将上述秘密信息分割为至少 2 个分割秘密信息，以便从上述第一秘密信息生成式计算出上述秘密信息的步骤。

[0060] 根据该结构，由于能从装置外部给予第一秘密信息生成式，因此，能够灵活地设定安全强度。

[0061] 此外，也可以上述秘密信息是从密钥发行机关发行的秘密密钥，上述加密信息嵌入方法进一步具有：上述密钥发行机关用密钥发行机关的秘密密钥加密上述秘密密钥的加密步骤；上述加密装置用公开密钥对已加密了的上述密钥进行解密和验证的验证步骤，上述秘密信息加密步骤将验证后的上述秘密密钥进行加密，上述嵌入步骤包括：将已利用上述加密装置加密后的上述加密信息变换为二进制数据的二进制变换步骤；在安全处理装置中嵌入上述二进制数据的二进制嵌入步骤，所述安全处理装置具有使用上述加密信息对与使用了上述秘密信息的运算结果相同的结果进行运算的功能。

[0062] 根据该结构，能够保护从密钥发行机关交给制造厂的秘密密钥不在递交途中被盗

听和窜改。

附图说明

- [0063] 图 1 是示出本发明的实施方式 1 中的签名生成单元的结构概要的图。
- [0064] 图 2 是示出本发明的实施方式 1 中的分割密钥识别信息表的结构的图。
- [0065] 图 3 是用树形结构概念地表现了本发明的实施方式 1 中的签名密钥生成式的图。
- [0066] 图 4 是示出本发明的实施方式 1 中的分割密钥信息表的图。
- [0067] 图 5 是示出本发明的实施方式 1 中的签名生成的概要的流程图。
- [0068] 图 6 是本发明的实施方式 1 中的恒等式生成流程图。
- [0069] 图 7 是表示本发明的实施方式 1 中的利用了交换法则的随机洗牌处理的流程图。
- [0070] 图 8 是表示本发明的实施方式 1 中的利用了结合法则的随机分组处理的流程图。
- [0071] 图 9 是示出本发明的实施方式 1 中的随机洗牌和随机分组处理执行后的分割密钥信息表的图。
- [0072] 图 10 是示出本发明的实施方式 1 中的利用了矩阵的恒等式生成处理的概念的图。
- [0073] 图 11 是示出本发明的实施方式 1 中的结合分割密钥识别信息表的图。
- [0074] 图 12 是详细示出本发明的实施方式 1 中的签名生成处理的流程图。
- [0075] 图 13 是示出本发明的实施方式 1 中的图 12 的流程图的接续的图。
- [0076] 图 14 是说明本发明的实施方式 1 中的生成签名生成式的具体例的图。
- [0077] 图 15 是本发明的实施方式 2 中的 ECDSA 的流程图。
- [0078] 图 16 是本发明的实施方式 2 中的图 15 的 S1502 的详细流程图。
- [0079] 图 17 是本发明的实施方式 2 中的图 15 的 S1504 的详细流程图。
- [0080] 图 18 是示出本发明的实施方式 3 中的分割密钥嵌入工序的图。
- [0081] 图 19 是示出本发明的实施方式 4 中的签名密钥生成恒等式生成单元 120 的程序的更新概要的图。
- [0082] 图 20 是示出本发明的实施方式 4 中的签名密钥生成恒等式生成程序的更新的流程图。
- [0083] 附图标记的说明
- [0084] 10 消息 M
- [0085] 20 签名密钥 D
- [0086] 21 签名密钥生成式 F
- [0087] 22 分割密钥生成装置
- [0088] 30 签名 S
- [0089] 100 签名生成单元
- [0090] 110 分割密钥存储单元
- [0091] 120 签名密钥生成恒等式生成单元
- [0092] 121 签名密钥生成恒等式 G
- [0093] 130 结合分割密钥生成单元
- [0094] 140 结合分割密钥存储单元
- [0095] 150 签名生成单元

- [0096] 200 分割密钥识别信息表
- [0097] 201 分割密钥标识符
- [0098] 1000 矩阵
- [0099] 1001 结合处选择部
- [0100] 1900 网络
- [0101] 1901 更新服务器
- [0102] 1902 更新用的签名密钥生成恒等式生成程序
- [0103] 1903 更新用的签名密钥生成恒等式生成程序的篡改检测值
- [0104] 1910 收发单元
- [0105] 1920 签名密钥生成恒等式生成程序更新单元

具体实施方式

- [0106] 以下,参照附图,关于本发明的实施方式进行说明。
- [0107] (实施方式 1)
 - [0108] 关于使用本发明的实施方式 1 涉及的秘密信息进行安全处理的安全处理装置,以使用签名密钥进行签名生成的签名生成单元为例进行说明。
 - [0109] 本发明的一个实施方式涉及的签名生成装置,在对于输入的信息 M ,生成基于签名生成式 $S = M^d \bmod n$ 的签名 S 的情况下,不直接在存储器上出现签名密钥 d ,而是取代签名密钥 d ,在存储器上出现分割密钥来生成签名,并且,每次生成签名都动态地变更分割密钥的生成过程、分割密钥的值、签名的生成过程。
 - [0110] 这样,就能够对于非法分析者的静态分析和动态分析,隐匿秘密信息即签名密钥 d 。
 - [0111] <结构>
 - [0112] 图 1 是本实施方式 1 中的包括签名生成单元 100 的签名生成系统的概要图。
 - [0113] 签名生成系统包括:使用签名密钥生成式 F21,从签名密钥 D 生成后述的分割密钥的分割密钥生成装置 22;使用上述分割密钥生成签名的签名生成装置 100。
 - [0114] 消息 M10 是签名对象数据,是输入到签名生成单元 100 中的数据。再有,在图 1 中,消息 M10 是从外部向签名生成单元 100 输入的,但也可以是在签名生成单元 100 内生成的数据,也可以是签名密钥生成装置的存储器内存储着的程序代码和 / 或数据。
 - [0115] 签名密钥 D20 是在签名生成中利用的秘密密钥信息,是需要保护不被非法分析的信息。具体地说,在使用了公开密钥密码系统的 RSA 签名生成的情况下,设质数 p 和质数 q 的积为 n 时的 RSA 签名生成运算 $M^d \bmod n$ 中的变量 d ,相当于秘密密钥信息,向 d 代入签名密钥 D20。
 - [0116] 由于生成签名时不在存储器上出现签名密钥 D20 的本身的价值,因此,图 2 中示出的分割密钥 111(D1、D2、D3、...、Dn) 是分割密钥生成装置 22 基于签名密钥生成恒等式 F21,预先分割签名密钥 D20 的值来得到的分割密钥。在此,在签名密钥生成式 F21 与分割密钥 111(D1、D2、D3、...、Dn) 之间,具有通过使用分割密钥 111(D1、D2、D3、...、Dn) 计算签名密钥生成式 F21 能计算签名密钥 D20 的值的关系。
 - [0117] 下面,关于分割密钥生成装置 22 和签名生成单元 100 依次进行说明。

[0118] (1) 分割密钥生成装置 22

[0119] 分割密钥生成装置 22 从外部接受签名密钥生成式 F21 和签名密钥 D20 的输入, 基于签名密钥生成式 F21 和签名密钥 D20, 生成分割密钥识别信息表 200 和分割密钥信息表 400, 并将分割密钥识别信息表 200 和分割密钥信息表 400 写入到后述的签名生成装置 100 所具有的分割密钥存储单元 110 中。

[0120] 分割密钥生成装置 22 具体地说是由微处理机、ROM、RAM、硬盘单元、显示器单元、键盘、鼠标等构成的计算机系统。在上述 ROM 或者上述硬盘单元中存储有计算机程序, 被上述 RAM 读出上述计算机程序, 通过上述微处理机按照上述计算机程序进行工作, 分割密钥生成装置 22 实现其功能。

[0121] 在此, 作为一例, 假设签名密钥生成式 F21 是设 8 个变量 ($d_1 \sim d_8$) 为自变量的下式。

$$[0122] F(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8) = (d_1+d_2+d_3+d_4)*(d_5+d_6+d_7+d_8)$$

[0123] 在此, 在变量 d_1 中代入分割密钥 D_1 , 在变量 d_2 中代入分割密钥 D_2 , 同样地, 在变量 $d_3 \sim d_8$ 中分别代入分割密钥 $D_3 \sim D_8$ 。

[0124] 对分割密钥生成装置 22 输入签名密钥生成式 F21, 作为图 4 中示出的分割密钥信息表 400, 但在说明分割密钥信息表 400 之前, 使用图 3, 关于签名密钥生成式 F21 的表现形式进行说明。

[0125] 图 3 用树形结构概念地表现了签名密钥生成式 F21 “ $F(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8) = (d_1+d_2+d_3+d_4)*(d_5+d_6+d_7+d_8)$ ”。

[0126] 变量 d_1, d_2, d_3, d_4 与所谓加法组 1 的组信息相关联。

[0127] 在此, 所述组信息是在签名生成式 F21 中表示操作数 ($d_1 \sim d_8$) 与其他操作数进行何种运算 (例如, 加法、乘法等) 的关系的属性信息。此外, 将构成上述组的各操作数和更小的组等称作成员。

[0128] 例如, 在“加法组”中, 利用算子“(加法)”运算各成员, 在“乘法组”中, 利用“*(乘法)”运算各成员。

[0129] 再有, 关于表示由多个变量构成的组彼此之间的关系和利用何种算子运算组与其他变量的关系的属性信息, 也可以称作组信息。

[0130] 通过这样地给予所谓组信息的属性信息, 就能够识别各变量与签名密钥生成式 F21 的关系。

[0131] 具体地说, 在“加法组”中, 由于如上所述地用算子“(加法)”运算各成员, 因此, 加法组 1 就表现了 $d_1+d_2+d_3+d_4$ 的式子。此外, 变量 d_5, d_6, d_7, d_8 与所谓加法组 2 的组信息相关联。加法组 2 与加法组 1 的情况同样地表现了 $d_5+d_6+d_7+d_8$ 的式子。另外, 所述加法组 1 和加法组 2 作为组信息与乘法组 1 相关联。

[0132] 所谓“乘法组”的组信息如前所述示出了用算子“*(乘法)”运算该组的各成员, 乘法组 1 表示了加法组 1 和加法组 2 相乘的关系, 表现了 $(d_1+d_2+d_3+d_4)*(d_5+d_6+d_7+d_8)$ 的签名密钥生成式 F21。

[0133] 下面, 使用图 4, 关于将图 3 的签名密钥生成式 F21 的结构具体表示为信息的分割密钥信息表 400 进行说明。

[0134] 分割密钥信息表 400 如图 4 所示, 由分割密钥组标识符 401 和分割密钥组成员标

识符 402 构成。

[0135] 分割密钥组标识符 401 是识别上述组的标识符, 分割密钥组成员标识符 402 是示出各组中所属的成员的标识符。

[0136] 图 4 中, 在加法组 1 中分配着“AG001”作为分割密钥组标识符 401。在此, 由于在图 3 中示出了属于组“AG001”中的成员是变量 d1、d2、d3、d4, 因此, 在分割密钥信息表 400 中登记着识别各个变量的“id001”、“id003”、“id003”、“id004”, 作为分割密钥组成员标识符 402。

[0137] 此外, 在加法组 2 中分配着“AG002”作为分割密钥组标识符 401。在此, 由于属于组“AG002”中的成员是变量 d5、d6、d7、d8, 因此, 在分割密钥信息表 400 中登记着识别各个变量的“id005”、“id006”、“id007”、“id008”, 作为分割密钥组成员标识符 402。

[0138] 此外, 在乘法组 1 中分配着“MG001”作为分割密钥组标识符 401。由于属于组“MG001”中的成员是加法组 1 和加法组 2, 因此, 在分割密钥信息表 400 中登记着各个分割密钥组标识符 401 即“AG001”和“AG002”, 作为分割密钥组成员标识符 402。

[0139] 使用这样的数据结构, 签名密钥生成恒等式生成单元 120 通过参照分割密钥识别信息表 200 和分割密钥信息表 400, 能够知道签名密钥生成式 F21 的结构。

[0140] 下面, 关于分割密钥生成装置 22 基于签名密钥生成式 F21, 将签名密钥 D 分割为多个分割密钥的处理进行说明。

[0141] 在本实施方式中, 由于签名密钥生成式 F21 的输入变量的数量是 8 个, 因此, 分割密钥生成装置 22 将签名密钥 D20 分割为 8 个分割密钥 D1 ~ D8。

[0142] 其中, 在 D1 ~ D8 中, 随机选择满足 $F(D1, D2, D3, D4, D5, D6, D7, D8) = D$ 的值。

[0143] 在此, 在签名密钥生成式的自变量 d_n 中代入分割密钥 D_n ($n : 1 \sim 8$)。即, 在签名密钥生成式 F21 的自变量 d_1 中代入分割密钥 D1, 同样地在自变量 d_2 中代入分割密钥 D2, 在自变量 $d_3 \sim d_8$ 中代入分割密钥 D3 ~ D8。

[0144] 作为基于签名密钥生成式 F21, 从签名密钥 D 计算出分割密钥 D1 ~ D8 的方法的一例, 分割密钥生成装置 22 从用签名密钥生成式 F21 进行的最外侧的运算开始, 逐次细化运算的单位来进行。

[0145] 具体地说, 首先, 分割密钥生成装置 22 选择出 $D = R1 * R2$ 成立的随机值 R1、R2。

[0146] 接着, 选择出 $R1 = (D1 + D2 + D3 + D4)$ 成立的随机值 D1、D2、D3、D4, 选择出 $R2 = (D5 + D6 + D7 + D8)$ 成立的随机值 D5、D6、D7、D8。

[0147] 分割密钥生成装置 22 对选择出的各个分割密钥 D1 ~ D8 分配用于识别各分割密钥的识别信息即分割密钥标识符 201。

[0148] 作为一例, 分割密钥生成装置 22 对分割密钥 D1 分配“ID001”作为分割密钥标识符 201, 对分割密钥 D2 分配“ID002”作为分割密钥标识符 201, 同样地分别对分割密钥 D3、D4、D5、D6、D7、D8 分配“ID003”、“ID004”、“ID005”、“ID006”、“ID007”、“ID008”, 作为分割密钥标识符 201。

[0149] 分割密钥生成装置 22 生成图 2 中示出的、将分割密钥标识符 201 和分割密钥 111 的对应起来的分割密钥识别信息表 200, 将分割密钥识别信息表 200 写入到后述的签名生成单元 100 所具有的分割密钥存储单元 110 中。

[0150] (2) 签名生成单元 100

[0151] 签名生成单元 100 如图 1 所示,其结构包括分割密钥存储单元 110、签名密钥生成恒等式生成单元 120、结合分割密钥生成单元 130、结合分割密钥存储单元 140、签名生成单元 150。

[0152] 签名生成单元 100 具体地说是由微处理机、ROM、RAM、硬盘单元、显示器单元等构成的计算机系统。在上述 ROM 或者上述硬盘单元中存储有计算机程序,被上述 RAM 读出上述计算机程序,通过上述微处理机按照上述计算机程序进行工作,签名生成单元 100 实现其功能。

[0153] 分割密钥存储单元 110 是存储器(内存)和硬盘等存储设备,存储由分割密钥生成装置 22 写入的分割密钥识别信息表 200 和分割密钥信息表 400。

[0154] 签名密钥生成恒等式生成单元 120 生成与签名密钥生成式 F21 恒等的式子即签名密钥生成恒等式 121。在此,所述恒等是指在运算中具有等效的、运算结果相同的关系。

[0155] 以下,使用图,关于签名密钥生成恒等式生成单元 120 根据签名密钥生成式 F21 和签名密钥 D20,动态地生成与签名密钥生成式 F21 恒等的式子即签名密钥生成恒等式 G121 和如图 11 中所示的后述的结合分割密钥 141 的方法,详细地进行说明。

[0156] 再有,以后进行以生成 RSA 签名为例的说明。

[0157] 图 5 是流程图,示出动态地生成签名生成恒等式 G121 和后述的结合分割密钥 141,直到生成签名 S30 并输出为止的处理的概要。其中,使用图 6 ~ 图 11 的流程图详细地说明图 5 的各步骤。

[0158] 在图 5 中,首先,签名密钥生成恒等式生成单元 120 读取分割密钥存储单元 110 中存储的分割密钥信息表 400,生成与签名密钥生成式 F21 输出相同结果的签名密钥生成恒等式 G121(步骤 S501)。

[0159] 接着,签名密钥生成恒等式生成单元 120 基于在步骤 S501 中生成的签名密钥生成恒等式 G121,通过结合运算分割密钥存储单元 110 中存储的分割密钥 111,生成与分割密钥 111 不同的值的结合分割密钥 141,并将生成的结合分割密钥 141 写入到结合分割密钥存储单元 140 中(步骤 S502)。

[0160] 接着,在签名生成单元 150 中,使用签名对象消息 M10、签名密钥生成恒等式 G121 和结合分割密钥 141,计算出不使签名密钥 D20 的值出现在存储器中的、成为与 $M^d \bmod n$ 相同的结果的签名 S30,并输出签名 S30,结束签名生成处理(步骤 S503)。

[0161] 在此,说明书中“ A^B ”的记载表示 A 的 B 次方,“ $A \bmod B$ ”的记载表示用自然数 B 除 A 时的剩余。

[0162] 接着,使用图 6,关于上述的图 5 的步骤 S501 的详细内容进行说明。

[0163] 首先,签名密钥生成恒等式生成单元 120 读入分割密钥存储单元 110 中存储的分割密钥信息表 400,进行随机地重排各组中所属的成员的排列的随机洗牌处理(步骤 S601)。以后使用图 7,关于随机洗牌的详细过程进行叙述。

[0164] 接着,将随机洗牌后的分割密钥信息表 400 的各组的成员随机地分组(步骤 S602)。以后使用图 8 和图 9,关于随机分组的处理的详细内容进行叙述。

[0165] 接着,将在步骤 S602 中随机分组后的各组进行式展开,变形为后述的所谓矩阵表现的数据结构,随机选择展开后的各组中的结合处来进行结合。其结果,生成与签名密钥生成式 F21 恒等的签名密钥生成恒等式 121,所以将其输出(步骤 S603)。以后使用图 10,关

于步骤 S603 进行叙述。

[0166] 下面, 使用图 7, 关于步骤 S601 中的分割密钥组的成员的随机洗牌处理的详细内容进行说明。

[0167] 再有, 所述本实施方式中的随机洗牌是指利用交换法则, 将例如式子 $d_1+d_2+d_3+d_4$ 变换为恒等的式子。此外, 在加法等的情况下, 将式子 $1+2+3$ 的计算结果和 $3+2+1$ 的计算结果相同的关系、即即使调换式子中的操作数的顺序, 计算结果也不变的关系, 称作“交换法则成立”。

[0168] 首先, 签名密钥生成恒等式生成单元 120 取得分割密钥存储单元 110 中存储的分割密钥信息表 400 的组总数 N(步骤 S701)。

[0169] 在此, 设分割密钥信息表 400 表现为二维数组 Table。此外, Table[n][m] 与图 4 的分割密钥组成员标识符 401 相对应, 示出属于从上数第 n 行的组中的、从左数第 m 个的分割密钥组标识符。此外, 在仅写为 Table “n”的情况下, 示出属于从上数第 n 行的组中的分割密钥组标识符的集合。

[0170] 具体地说, 若用上述的 Table 的形式表示图 4 中示出的分割密钥信息表 400, 则如下:

[0171] Table[0] = {AG001, AG002}、

[0172] Table[1] = {id001, id002, id003, id004}、

[0173] Table[2] = {id005, id006, id007, id008}。

[0174] 再有, 在该例子和以下的例子中, 设 Table 的各索引 (index) 从 0 开始。

[0175] 组总数 N 是加法组、乘法组等的组的个数, 相当于 n 的最大值 +1。

[0176] 图 4 中示出的分割密钥信息表 400 的情况下, 组总数 N 为 3。

[0177] 接着, 用 0 初始化变量 i(步骤 S702)。

[0178] 在此, i 是变量, 对成为当前混洗对象的分割密钥组是第几个分割密钥组进行计数。

[0179] 接着, 产生随机数值, 代入到计数混洗次数的变量 sn 中 (步骤 S703)。

[0180] 在此, 混洗次数 sn 是变量, 表示以后对各分割密钥组进行几次混洗。在此, 通过利用随机数作为混洗次数 sn, 能够动态地设定混洗的次数, 进而能够动态地生成签名密钥生成恒等式。

[0181] 接着, 取得 Table[i] 中登记着的成员数, 即第 i 个组中登记着的成员数 M(步骤 S704)。

[0182] 接着, 产生 2 个 0 以上不足 M 的随机整数, 分别代入到变量 s1、s2 中 (步骤 S705)。

[0183] 在此, 通过利用随机整数作为 s1、s2, 能够动态地设定成为混洗对象的成员的位置, 进而能够动态地生成签名密钥生成恒等式。

[0184] 接着, 调换 Table[i][s1] 和 Table[i][s2] 的值, 减小 sn(步骤 S706)。

[0185] 即, 在调换了属于第 i 个组中的第 s1 个结构要素和第 s2 个结构要素之后, 将剩余混洗次数减 1。

[0186] 接着, 判定是否 $sn > 0$, 即、关于成为当前混洗对象的第 i 个分割密钥组, 是否还剩余有混洗次数 (步骤 S707)。

[0187] 若步骤 S707 的判定结果是“是”, 就进一步反复进行混洗, 向步骤 S705 返回。

[0188] 反之,若判定结果是“否”,就判定 i 是否等于 $N-1$ (步骤 S708)。即,判定关于全部组混洗是否已结束。

[0189] 若步骤 S708 的判定结果是“否”,就通过增加 i 后向步骤 S703 转移,对下一个分割密钥组进行混洗(步骤 S709)。反之,若步骤 S708 的判定结果是“是”,则对于全部分割密钥组的混洗已结束,因此就结束了随机洗牌处理。

[0190] 下面,参照图 8 的流程图,关于步骤 S602 中的分割密钥组的成员的随机分组处理的详细内容进行说明。

[0191] 所述随机分组是例如利用结合法则将式子 $(d_1+d_2+d_3)+d_4$ 变换为恒等式 $d_1+(d_2+d_3+d_4)$ 的处理。在此,例如式子 $(1+2)+3$ 与式子 $1+(2+3)$ 的结果是相同值,将这样 的关系,即、与一次进行计算的单位的分割方式无关而结果为相同值的关系,称作“结合法则成立”。

[0192] 再有,在图 8 中,也如说明图 5 时所述,用二维数组的 Table[n][m] 表现分割密钥信息表 400 并进行说明。

[0193] 首先,签名密钥生成恒等式生成单元 120 取得分割密钥存储单元 110 中存储的分割密钥信息表 400 的组总数 N (步骤 S801)。

[0194] 接着,用 0 初始化变量 i (步骤 S802)。在此, i 是变量,表示成为当前随机分组的对象的分割密钥组的号码。

[0195] 接着,将组分割位置列表 GP 设置为空列表(步骤 S803)。在此,组分割位置列表 GP 是表示在从前面开始的第几个位置上分割分割密钥组的成员的值的列表。

[0196] 接着,取得 Table[i] 中登记着的成员数 M ,随机选择 1 以上 M 以下的自然数,将选择的值作为组分割数 k (步骤 S804)。在此,所述组分割数是表示将分割密钥组分割为几个组的数。通过将组分割数 k 设定为随机自然数,就能够动态地设定式子的分割数,进而能够动态地生成签名密钥生成恒等式。

[0197] 接着,不重复地随机选择 $k-1$ 个 0 以上 $M-1$ 以下的整数,按降序重排选择的值,设置为组分割位置列表 GP(步骤 S805)。

[0198] 这样地,通过随机设定组分割位置列表 GP 的内容,就能够动态地设定式子的分割位置,进而能够动态地生成签名密钥生成恒等式。

[0199] 接着,在位于步骤 S804 中选择的组分割列表 GP 的位置上的成员的后面,分隔 Table 内的成员的组,分为 k 个组(步骤 S806)。

[0200] 接着,检验 i 是否等于 $N-1$ (步骤 S807)。即,关于全部分割密钥组,判定随机分组是否已结束。

[0201] 若步骤 S807 的判定结果是“否”,通过增加 i 并返回步骤 S803,进一步关于下一个分割密钥组进行随机分组(步骤 S808)。反之,若判定结果是“是”,就判断为关于全部分割密钥组的随机分组已结束,并结束随机分组。

[0202] 图 9 是示出了对图 4 的分割密钥信息表进行了图 7 的随机洗牌和图 8 的随机分组的处理之后的分割密钥信息表 400 的状态的图。

[0203] 图 9 中示出了按照图 7 的流程随机洗牌后的结果,在 MG001 中,将 Table[0][2] = {AG001, AG002} 的排列混洗为 {AG002, AG001},将 Table[1][4] = {id001, id002, id003, id004} 的排列混洗为 {id002, id001, id004, id003},将 Table[2][4] = {id005, id006,

id007, id008} 的排列混洗为 {id007, id006, id008, id005}。

[0204] 此外,示出了按照图 8 的流程随机分组后的结果,在 MG001 中,将分割密钥组成员标识符分组为 {AG002} 和 {AG001} 的 2 个,在 AG001 中分组为 {id002, id001} 和 {id004, id003} 的 2 个,在 AG002 中分组为 {id005, id006} 和 {id007, id008} 的 2 个。

[0205] 在如图 9 所示随机洗牌和随机分组后的情况下,利用交换法则和结合法则,将图 4 的分割密钥信息表 400 所示的签名密钥生成式 F21

[0206] $F(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8) = (d_1+d_2+d_3+d_4)*(d_5+d_6+d_7+d_8)$

[0207] 变换为与签名密钥生成式 F21 恒等的签名密钥生成恒等式

[0208] $G' (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8) = ((d_7+d_6)+(d_8+d_5)*((d_2+d_1)+(d_4+d_3)))$ 。

[0209] 接着,进一步使用后述的矩阵表现,将该签名密钥生成恒等式 G' 进行恒等变换。

[0210] 图 10(a) 将图 9 中生成的签名密钥生成恒等式 121 的右边 $((d_7+d_6)+(d_8+d_5))*((d_2+d_1)+(d_4+d_3))$ 予以矩阵表现。

[0211] 在此,所述矩阵表现是一种数据结构,为了利用分配法则制作与签名密钥生成式 F 恒等的式子 G,例如使 $(d_1+d_2)*(d_3+d_4)$ 与 $d_1*(d_3+d_4)+d_2*(d_3+d_4)$ 相等。在此,将例如使式子 $3*(2+1)$ 的值与式子 $3*2+3*1$ 的值相等这样的关系,即、式展开前的值与式展开后的值相等的关系,称作“分配法则成立”。

[0212] 在此,矩阵 1000 的各要素是示出关于随机分组后的各组彼此之间实施了式展开的情况下所产生的乘法的式子,分别示出使用分配法则展开了签名密钥生成恒等式 G' 时相乘的项。

[0213] 若示出更具体的例子,用 Matrix[2][2] 的 2×2 的数组表现矩阵 1000,则如下。

[0214] $Matrix[0][0] = (d_7+d_6)*(d_2+d_1)$

[0215] $Matrix[0][1] = (d_7+d_6)*(d_4+d_3)$

[0216] $Matrix[1][0] = (d_8+d_5)*(d_2+d_1)$

[0217] $Matrix[1][1] = (d_8+d_5)*(d_4+d_3)$

[0218] 在此,Matrix[n][m] 是最初的加法组中的第 n 个要素与下一个加法组中的第 m 个要素的积,分别示出了展开了签名密钥生成恒等式时相乘的各项彼此之间的积。再有,假设索引 n 和 m 的号码从 0 开始,在各加法组中,越位于左侧的要素,索引号越小。

[0219] 例如,Matrix[0][0] 成为在图 9 的表中最初的加法组即 AG001 中的最初成员 (d_7+d_6) 与第 2 个加法组即 AG002 中的最初成员 (d_2+d_1) 相乘后的值。再有,当然,按照加法组的数量,数组的维数 Matrix[n][m] “1”... 地增加。

[0220] 对于这样的矩阵表现的签名密钥生成恒等式 G,在上述的步骤 S603 中,对矩阵表现的各要素进行随机选择结合之处的处理。即,在矩阵表现中,由于属于相同行或者相同列的要素彼此之间是至少其中之一含有共通项的式子,因此,利用该性质,能够彼此结合属于相同行或者列中的式子。

[0221] 以下,示出更具体的例子。首先,在图 10(a) 中,假设如 1001 所图示选择了 Matrix[0][1] 和 Matrix[1][1] 作为结合处。在此,在被选择为结合处的 1001 的 Matrix[0][1] 和 Matrix[1][1] 的两个要素中,共同存在着 (d_4+d_3) 。因此,通过如

[0222] $(d_4+d_3)*((d_7+d_6)*(d_8+d_5))$

[0223] 这样地用 (d_4+d_3) 总括起来,结合处 1001 就能够结合。

[0224] 在此,通过如上所述地随机选择结合处,能够动态地决定结合处,进而能够动态地生成签名生成密钥恒等式 G。

[0225] 图 10(b) 用通常的式子的变形,例示了使用矩阵表现直到生成最终的签名密钥生成恒等式 G121 为止的式变形。

[0226] 图 9 中生成的签名密钥生成恒等式 G' 是

$$G' = ((d7+d6)+(d8+d5)) * ((d2+d1)+(d4+d3)) \dots \text{ (式 1)}$$

[0228] 若使用分配法则展开该式子,就如(式 2)所示。

$$G' = (d7+d6)*(d2+d1)+(d7+d6)*(d4+d3)$$

$$+ (d8+d5)*(d2+d1)+(d8+d5)*(d4+d3) \dots \text{ (式 2)}$$

[0231] 另外,作为在图 10(a) 的矩阵 1000 内通过步骤 S603 结合(式 2)的地方,若结合相当于随机指定的 1001 的项,则成为(式 3)。

$$G' = (d7+d6)*(d2+d1)+(d8+d5)*(d2+d1)$$

$$+ (d4+d3)*((d7+d6)*(d8+d5)) \dots \text{ (式 3)}$$

[0234] 该(式 3)成为经过图 6 的 S601 到 S603 而生成的签名密钥生成恒等式 G121。

[0235] 结合分割密钥生成单元 130 基于签名密钥生成恒等式 G121,将分割密钥存储单元 110 中存储的分割密钥彼此进行结合运算,生成不同于分割密钥 111 的值的结合分割密钥 141(CD1、CD2、...、CDm)。对生成后的各个结合分割密钥 141,赋予用于识别结合分割密钥 141 的结合分割密钥标识符 1101,生成将图 11 所示的结合分割密钥标识符 1101 和结合分割密钥 141 对应起来的结合分割密钥识别信息表 600,将结合分割密钥识别信息表 600 写入到结合分割密钥存储单元 140 中。

[0236] 在此,在签名密钥生成恒等式 G121 与结合分割密钥 141(CD1、CD2、...、CDm)之间,具有能够根据结合分割密钥 141(CD1、CD2、...、CDm)和签名密钥生成恒等式 121 计算出签名密钥 D 的值的关系。

[0237] 此外,所述结合运算是汇总多个值生成新的值的运算,作为具体的例子,有加法、乘法和其组合等。即,在所谓 $1+2=3$ 的运算中,“3”是对于所谓“1”和“2”的值实施了所谓“+”的结合运算后得到的值。

[0238] 具体地说,将如前所述地随机分组后的加法组的各要素 {id002, id001} 和 {id004, id003}、{id005, id006}、{id007, id008} 作为一组输入自变量。

[0239] 若按照签名密钥生成恒等式 G' 的(式 3)所示的最小括弧的单位生成结合分割密钥 141,则生成以下的 7 个结合分割密钥 141。

[0240] 在变量 cd1 中存储运算了 d7+d6 的结果值,在 cd2 中存储运算了 d2+d1 的结果值,在 cd3 中存储运算了 d8+d5 的结果值,在 cd4 中存储运算了 d2+d1 的结果值,在 cd5 中存储运算了 d4+d3 的结果值,在 cd6 中存储运算了 d7+d6 的结果值,在 cd7 中存储运算了 d8+d5 的结果值。即,下式成立。

$$G' (d1, d2, d3, d4, d5, d6, d7, d8)$$

$$= G(cd1, cd2, cd3, cd4, cd5, cd6, cd7)$$

[0243] 在此,在 $G' (d1, d2, d3, d4, d5, d6, d7, d8)$ 的自变量的变量 d1 中,代入具有对应于 id001 的分割密钥标识符 ID001 的分割密钥 D1,同样地,在变量 d2 中代入分割密钥 D2,在各个变量 d3、d4、d5、d6、d7、d8 中分别代入分割密钥 D3、D4、D5、D6、D7、D8。

[0244] 此外,在 G(cd1, cd2, cd3, cd4, cd5, cd6, cd7) 的自变量的变量 cd1 中代入结合分割密钥 D1,同样地,在变量 cd2 中代入结合分割密钥 CD2,在各个变量 cd3、cd4、cd5、cd6、cd7 中分别代入结合分割密钥 CD3、CD4、CD5、CD6、CD7。

[0245] 在此,CD1 = D7+D6 成立,CD2 = D2+D1 成立,CD3 = D8+D5 成立,CD4 = D2+D1 成立,CD5 = D4+D3 成立,CD6 = D7+D6 成立,CD7 = D8+D5 成立。

[0246] 在此,由于结合分割密钥 141 是运算的结果值,因此,难以仅从该值导出原来的分割密钥 111,能够隐匿分割密钥 111。

[0247] 图 11 示出了在各结合分割密钥 141(CD1、CD2、...、CD7) 中分别赋予了结合分割密钥标识符 1101(ID001、ID002、...、ID007)。

[0248] 从而,若将签名密钥生成恒等式 G' 即(式 3)作为使用了结合分割密钥 141 的表现即签名密钥生成恒等式 G121,则如下述的(式 4)所示。

[0249] $G(cd1, cd2, cd3, cd4, cd5, cd6, cd7)$

[0250] $= (cd1*cd2)+(cd3*cd4)+cd5*(cd6+cd7)\dots$ (式 4)

[0251] 在此,在(式 4)的自变量 cd1、cd2、...、cd7 中代入了结合分割密钥 141(CD1、CD2、...、CD7)的值的结果,与在(式 3)的自变量 d1、d2、...、d8 中代入了(D1、D2、...、D8)的结果相等,另外,结合分割密钥 141(CD1、CD2、...、CD7)是对分割密钥 111(D1、D2、...、D8)进行结合运算的结果,因此,(式 4)就成为能够得到与签名密钥生成式 F21 相同结果的式子,因此,签名生成单元 150 就利用(式 4)作为签名密钥生成恒等式 121。

[0252] 结合分割密钥存储单元 140 存储由结合分割密钥生成单元 130 生成签名时动态地生成的结合分割密钥 141。

[0253] 签名生成单元 150 将消息 M10 作为输入,使用结合分割密钥存储单元 140 中存储的结合分割密钥 141 生成签名 S30。若示出 RSA 签名生成中的例子,则不使用签名密钥 D20,而仅使用结合分割密钥 141 来生成与 $S = M^d \bmod n$ 相同运算结果的签名 S30。

[0254] 接着,使用图 12 和图 13,说明示出签名生成单元 150 使用了结合分割密钥 141 和签名密钥生成恒等式 G121(式 4)的签名生成的处理的流程。

[0255] 首先说明签名 S 的生成方法的概略。

[0256] 根据签名密钥生成恒等式 G121 的结构决定签名生成中使用的运算式。

[0257] 例如在签名密钥生成恒等式 G 的各自变量 cd1 ~ cd7 中代入 CD1 ~ CD7,在签名 S 是 $(CD1*CD2)+(CD3*CD4)+CD5*(CD6+CD7)$ 时,使用信息 M 和 n ($= p*q$),利用下述计算——

[0258] $S_0 = (M^CD1)^CD2 \bmod n$ 、

[0259] $S_1 = S_0 * ((M^CD3)^CD4) \bmod n$ 、

[0260] $S = S_1 * (((M^CD6)*(M^CD7))^CD5) \bmod n$

[0261] 生成与 $S = M^d \bmod n$ 相同的结果。

[0262] 更具体地说,如 $(CD1*CD2)$ 和 $(CD3*CD4)$ 和 $CD5*(CD6+CD7)$ 这样用“+”算子分隔上式 G,利用分隔后的各个式子,对消息 M 进行乘方运算,通过乘以各自的乘方运算结果,制作签名 S。

[0263] 以下转移到流程的详细说明,但在以下的例子中,假设签名密钥生成恒等式 G121 是(式 4)。此外,假设签名生成单元 150 利用使用逆波兰表示法表现了(式 4)的方法生成签名。在此,所述逆波兰表示法是在运算对象的后面描述算子的表示法,是用每次出现算子

时都对该算子的 2 个前面的值和 1 个前面的值, 计算适用于该算子的值的规则来表示的表示法。

[0264] 例如, 用逆波兰表示法表示(式 4), 若示出加入到数组中的例子, 就成为如下的数组 P。

[0265] $P[13] = \{CD1, CD2, *, CD3, CD4, *, +, CD5, CD6, CD7, +, *, +\}$

[0266] 以下, 使用图 12 和图 13, 关于签名生成流程详细地进行说明。

[0267] 首先, 如上所述地用逆波兰表示法表现签名密钥生成恒等式 G, 将其表现为数组 P(步骤 S1201)。

[0268] 接着, 设定 i 和 j 为 1, 设定 Chk 和 Flag 为 0, 设定 N 为数组 P 的尺寸, 将 P[0] 推入到栈中(步骤 S1202)。在此, i 是当前着眼的数组的索引, j 表示当前堆积在栈中的分割结合密钥的个数。此外, Chk 表示是否确认了在数组中直到哪个索引其算子是否为“+”。在此, 在使用上述的 j, 从当前访问的数组 P[i] 中判定 j-1 个前面的算子是否是“+”, 判定是否是“+”算子的有效区间内时, 利用 Chk。此外, 利用 Flag 作为表示是否是“+”算子的有效区间内的标志。更具体地说, 利用在签名密钥生成恒等式 G 的分隔点的判定中。

[0269] 接着, 判别 P[i] 是否是结合分割密钥(步骤 S1203)。

[0270] 若步骤 S1203 的判别结果是“是”, 即是结合分割密钥, 就将 P[i] 的值推入到栈中, 为了移动 1 个数组 P 中的访问而增加 i, 此外, 由于栈中堆积的分割结合密钥的个数增加, 因此增加 j(步骤 S1204)。

[0271] 反之, 若判别结果是“否”, 即 P[i] 是算子, 则使堆栈中堆积着的高位 2 个结合分割密钥出栈, 分别设为 Val1、Val2(步骤 S1205)。

[0272] 接着, 判定 $i+j-1$ 是否大于 Chk 的值, 并且 $i+j-1$ 是否在 N 以下(步骤 S1206)。

[0273] 若步骤 S1206 的判定结果是“是”, 就将 Chk 设定为 $i+j-1$ 的值。即, 直到数组 P 的第 $i+j-1$ 个, 是否是“+”的判定已结束, 因此更新 Chk 的值。反之若是“否”, 就向步骤 S1208 转移处理。再有, 在步骤 S1206 也确认了 $i+j-1$ 是否在 N 以下, 是为了防止 Chk 表示超过了数组尺寸的位置。

[0274] 接着, 判定是否 $j > 0$ 并且 $Flag = 1$, 并且 $P[Chk]$ 是否是“+”算子(步骤 S1208)。

[0275] 若步骤 S1208 的判定结果是“是”, 就将 Flag 设定为 0(步骤 S1209)。

[0276] 反之, 若判定的结果是否, 就向处理“A”转移处理。

[0277] 下面, 使用图 13, 关于图 12 的流程的继续进行说明。

[0278] 首先, 判定 Val1 和 Val2 的类型(步骤 S1301)。

[0279] 步骤 S1301 的判别结果, 在 Val1 和 Val2 两者是结合分割密钥的情况下, 向步骤 S1310 转移处理。

[0280] 反之, 在步骤 S1301 的判别结果, Val1 和 Val2 的某一个是结合分割密钥, 另一个是签名乘方中间值的情况下, 向步骤 S1320 转移处理。此处的所述签名乘方中间值, 示出用至少一个以上的结合分割密钥将消息 M10 乘方后的值, 成为在计算签名 S30 的过程中的中间值。具体地说, 后述的 S0 和 S1 等相当于它。

[0281] 反之, 在步骤 S1301 的判别结果是 Val1 和 Val2 两者都是签名乘方中间值的情况下, 向步骤 S1330 转移处理。

[0282] 以下, 关于在 S1301 处理后分别分支为 S1310、S1320、S1330 时的各处理以后的个

别处理进行说明。

[0283] 以下,关于处理转移到步骤 S1310 的情况进行说明。

[0284] 首先,判别 Flag(步骤 S1310)。

[0285] 在步骤 S1310 的判别结果是 $Falg = 0$,即是“+”算子的有效区间内的情况下,向步骤 S1311 转移处理。

[0286] 接着,判别 $P[i]$ 的算子(步骤 S1311)。

[0287] 若步骤 S1311 的判别结果是 $P[i]$ 是“+”算子,则向步骤 S1312 转移处理。

[0288] 反之,若步骤 S1311 的判别结果, $P[i]$ 是“*”算子,则向步骤 S1313 转移处理。

[0289] 在步骤 S1312 中,将 $(M^{\wedge}Val)*(M^{\wedge}Val2) \bmod n$ 的值推入栈中(步骤 S1312)。

[0290] 在步骤 S1313 中,将运算了 $((M^{\wedge}Val1)^{\wedge}Val2) \bmod n$ 后的值推入到栈中(步骤 S1313)。

[0291] 在步骤 S1312、S1313 的处理之后,设 $j = j-2$,向步骤 S1341 转移处理(步骤 S1314)。在此,设 $j = j-2$ 是因为,在一系列计算中使用结合分割密钥(相当于 Val1 和 Val2),因此要减少堆积在栈中的结合分割密钥的数量。再有,在步骤 S1312 和 S1313 中,将计算结果推入到栈中,但由于该值是上述的签名乘方中间值而不是结合分割密钥,因此将 j 减 2。

[0292] 在步骤 S1341 中,由于对于该区间的运算已结束,因此,重新将 Flag 设定为 1(步骤 S1341)。即,将 Flag 设定为 0 的期间是从判定为用“+”算子分隔数组 $P[Chk]$ 时开始,直到执行 S1312 和 S1313 的处理而更新 j 的值为止的期间,除此以外的处理是在 $Flag = 1$ 的状态下处理各步骤。换言之,从上一次判定为 $P[Chk]$ 的用“+”算子的分隔,到判定为下一个 $P[Chk]$ 的用“+”算子分隔为止,持续 $Flag = 1$ 的状态。

[0293] 反之,在步骤 S1310 的判别结果是 $Flag = 1$ 的情况下,即、是利用算子“+”分隔的区间外的情况下,向步骤 S1315 转移处理。

[0294] 接着,在步骤 S1315 中,判别 $P[i]$ 的算子(步骤 S1315)。

[0295] 若步骤 S1315 的判别结果为 $P[i]$ 是“+”算子,就向步骤 S1316 转移处理。

[0296] 反之,若步骤 S1315 的判别结果为 $P[i]$ 是“*”算子,就向步骤 S1317 转移处理。

[0297] 在步骤 S1316 中,将运算了 $Val1+Val2$ 后的结果推入栈中(步骤 S1316)。

[0298] 在步骤 S1317 中,将运算了 $Val1*Val2$ 后的结果推入到栈中(步骤 S1317)。再有,在此,在步骤 S1316 和步骤 S1317 中推入的值也看作是分割结合密钥。即,将作为乘方中间值以外而推入的值看作是分割结合密钥。

[0299] 在步骤 S1316 和 S1317 处理之后,向步骤 S1340 转移处理。以后叙述步骤 S1340 以后的处理。

[0300] 到此为止是关于 S1310 以后的个别处理的说明。

[0301] 以下,关于向步骤 S1320 转移了处理的情况,即、是 Val1 和 Val2 的组、结合分割密钥和签名乘方中间值的组的情况进行说明。

[0302] 首先,判别 $P[i]$ 的算子(步骤 S1320)。

[0303] 若步骤 S1320 的判别结果为 $P[i]$ 是“+”算子,则向步骤 S1321 转移处理。

[0304] 反之,若步骤 S1320 的判别结果为 $P[i]$ 是“*”算子,就向步骤 S1322 转移处理。

[0305] 以下,首先关于向步骤 S1321 分支时的流程进行说明。

[0306] 在步骤 S1321 中,假设结合分割密钥的值为 Val1,乘方中间值为 Val2,向栈推入运算了 $(M^{\wedge} Val1) * Val2 \bmod n$ 后的值(步骤 S1321)。

[0307] 在步骤 S1322 中,假设结合分割密钥的值为 Val2,乘方中间值为 Val1,向栈推入运算了 $Val1^{\wedge} Val2 \bmod n$ 后的值(步骤 S1322)。

[0308] 步骤 S1321 和 S1322 处理之后,向步骤 S1340 转移处理(步骤 S1340)。再有,后面叙述步骤 S1340 以后的处理。

[0309] 到此为止是关于 S1310 以后的个别处理的说明。

[0310] (S1330 以后的个别处理)

[0311] 以下关于向步骤 S1330 转移了处理的情况,即、Val1 和 Val2 两者是签名乘方中间值的情况进行说明。

[0312] 在步骤 S1330 中,利用乘方中间值即 Val1 和 Val2,将运算了 $Val1 * Val2 \bmod n$ 后的值推入到栈中,并向步骤 S1341 转移处理(步骤 S1330)。

[0313] 到此为止是关于 S1330 以后的个别处理的说明。

[0314] 以上,结束关于 S1310、S1320、S1330 以后的个别流程的说明。

[0315] (S1310、S1320、S1330 以后的共通处理)

[0316] 以下,关于在 S1310、S1320、S1330 以后成为共通的流程的部分进行说明。

[0317] 在步骤 S1316 或 S1317,或者 S1321 或 S1322 处理之后,减小 j,否则不进行任何处理而向步骤 S1341 转移处理(步骤 S1340)。在此,减小 j 是因为,在步骤 S1316 或 S1317 的情况下,使用结合分割密钥即 Val1 和 Val2,仅推入被看作结合分割密钥的值,因此,减 1 的结合分割密钥减少。此外还因为,在步骤 S1321 或 S1322 的情况下,使用乘方中间值和结合分割密钥生成了新的乘方中间值,因此,在该生成中使用的结合分割密钥从栈中消失。

[0318] 在步骤 S1314 或者 S1340 处理之后,置 Flag 为 1,向步骤 S1342 转移处理(步骤 S1341)。

[0319] 在步骤 S1341 的处理之后,判别是否为 $i < N$ (步骤 S1342)。

[0320] 若步骤 S1342 中的判别结果是“否”,就关于数组 P 的下一个要素继续进行同样的处理,因此,在步骤 S1343 中增加 i,并向 B 转移处理。

[0321] 反之,若步骤 S1342 中的判别结果是“是”,则签名的生成已结束,所以将栈的值设为签名 S30,输出签名 S30(步骤 S1344),结束签名生成。

[0322] 利用以上步骤,使用签名生成恒等式 G' 和结合分割密钥 141 生成了对消息 M10 的签名 S30。

[0323] 在图 13 中,关于签名生成流程图进行了说明。在此,使用图 14 示出利用了图 13 的流程时的栈的样子。

[0324] 图 14 是示出了在图 12 和图 13 中说明的签名生成流程中的栈的样子的图。

[0325] 对于用逆波兰表示法表现了签名生成恒等式 G121 的 P,依次说明签名生成。

[0326] 首先,根据 $P[13] = \{CD1, CD2, *, CD3, CD4, *, +, CD5, CD6, CD7, +, *, +\}$,向栈中推入 P[0] 的结合分割密钥 CD1 和 P[1] 的结合分割密钥 CD2(步骤 S1401)。这时, i 和 j 都是 2, $i+j-1$ 是 3,因此,设定 Chk 的值为 3。

[0327] 接着,由于 P[2] 是算子“*”,因此使栈的高位 2 个的值出栈,由于 Flag = 0 并且是算子“*”,因此,将 $(M^{\wedge} CD1)^{\wedge} CD2 \bmod n$ 作为 S0,向栈中推入 S0(步骤 S1402)。在该处理结

束了的阶段, $i = 2, j = 0, \text{Chk} = 3$ 。此外, 利用步骤 S1341 的工作, 将 Flag 的值返回到 1。
[0328] 接着, 向栈中推入 $P[3]$ 的结合分割密钥 CD3 和 $P[4]$ 的结合分割密钥 CD4(步骤 S1403)。这时, $i = 4, j = 2, \text{Chk}$ 的值被更新为 5。再有, 由于 $P[\text{Chk}]$ 不是算子“+”, 因此 Flag 的值仍是 1。

[0329] 接着, 由于 $P[5]$ 是算子“*”, 因此使栈的高位 2 个的值出栈, 此外, 由于 $i = 5, j = 2$, 因此, Chk 被更新为 6。在此, $P[\text{Chk}]$ 的算子是“+”, 因此, Flag 的值被更新为 0。由于 $\text{Flag} = 0$ 并且 $P[5]$ 的算子是“*”, 因此, 将 $(M^{\wedge} CD3)^{\wedge} CD4 \bmod n$ 作为 S1, 向栈中推入 S1, 将 Flag 更新为 1(步骤 S1404)。在该处理结束了的阶段, $i = 5, j = 0, \text{Chk} = 6, \text{Flag} = 1$ 。

[0330] 接着, 由于 $P[6]$ 是算子“+”, $\text{Flag} = 1$, 因此使栈的高位 2 个的值 S0 和 S1 出栈, 将 $S0 * S1 \bmod n$ 的运算结果作为 S2, 向栈中推入 S2。

[0331] 接着, 依次向栈中推入相当于 $P[7], P[8], P[9]$ 的结合分割密钥 CD5、CD6、CD7(步骤 S1405)。

[0332] 接着, $P[10]$ 是算子“+”, 这时的 i 的值是 10, j 的值是 3, $\text{Flag} = 1$ 并且 $P[12] = “+”$, 因此设定 Flag 为 0。之后, 使栈的高位 2 个的值 CD6 和 CD7 出栈, 由于 $\text{Flag} = 0$ 并且是算子“+”, 因此, 将运算了 $(M^{\wedge} CD6) * (M^{\wedge} CD7) \bmod n$ 后的值作为 S3, 向栈中推入 S3(步骤 S1406)。

[0333] 接着, 由于 $P[11]$ 是算子“*”, 因此使栈的高位 2 个的 S3 和 CD5 出栈。这时 S3 是签名乘方中间值, CD5 是结合分割密钥, 是算子“*”, 因此, 将运算了 $S3^{\wedge} CD5 \bmod n$ 后的值作为 S4, 推入到栈中(步骤 S1407)。

[0334] 接着, 由于 $P[12]$ 是算子“+”, 因此使栈的高位 2 个的 S2 和 S4 出栈。这时, S2 和 S4 都是签名乘方中间值, 因此, 将运算了 $S2 * S4 \bmod n$ 后的结果推入到结果栈中(步骤 S1408)。

[0335] 利用以上步骤, 就将运算了 $S2 * S4 \bmod n$ 后的结果作为签名 S30, 输出签名 S30, 且签名生成结束。

[0336] 这样地, 在本实施方式中, 一切都不利用签名密钥 D21 的值, 而能生成签名 S30。另外, 由于在生成签名时动态地生成结合分割密钥 141, 因此能使用每次不同的结合分割密钥来生成签名。另外, 不仅签名密钥 D, 连 RSA 签名中的秘密信息 (p、q) 的值也都不利用就能生成签名。此外, 由于在每次执行签名生成处理时签名生成恒等式 G 也每次不同, 因此, 即使取随机数据的收集结果的差分, 也难以确定成为用于计算签名密钥 d 的密钥候补的结合分割密钥。另外, 即使特定了结合分割密钥, 为了根据结合分割密钥求签名密钥 D, 也要确定签名密钥生成式 G, 必须要分析能取得签名密钥生成恒等式 G 的所有类型, 签名密钥 D 的分析非常困难。另外, 由于签名生成流程也每次不同, 生成签名时的功率和签名生成处理时间也变化, 因此, 有对于功率差分攻击和定时攻击也很安全的效果。

[0337] 再有, 在本实施方式 1 中, 随机洗牌和随机分组结合分割密钥 141, 然后使用矩阵表现生成了签名密钥生成恒等式 G, 但这些只不过是一个实施例, 只要使用交换法则、结合法则和分配法则等, 从签名密钥生成式 F21 生成签名密钥生成恒等式 G 就可以, 但也可以是本实施方式中说明的方法以外的方法。

[0338] 再有, 在本实施方式 1 中, 通过彼此结合运算已预先存储在分割密钥存储单元中的分割密钥来生成了结合分割密钥, 但用结合分割密钥生成单元生成的结合分割密钥也可

以是将结合分割密钥彼此结合运算的值。若这样做,由于能取结合分割密钥的值的变动增加,因此就具有更难确定签名密钥 D 的效果。

[0339] 再有,在本实施方式 1 中,通过将已预先存储在分割密钥存储单元中的分割密钥彼此进行结合运算来生成了结合分割密钥,但只要生成与作为使用了签名密钥 D 的签名生成结果的签名 S 相同的结果就可以,也可以使用对生成相同结果的处理没有影响的冗余密钥进行处理。此外,冗余密钥也可以设定为产生了随机数的值。在利用以随机数值为代表的冗余密钥的情况下,也可以设定为用签名密钥生成恒等式生成的式子也包含该值。通过利用这样的冗余密钥,利用在原来的处理中不需要的信息,具有使非法分析者的分析更困难的效果。

[0340] (实施方式 2)

[0341] 以下,关于本发明的实施方式 2 进行说明。

[0342] 在实施方式 1 中说明了 RSA 签名生成中的本发明的实施例,但在本实施方式 2 中关于应用于 ECDSA (Elliptic Curve Digital Signature Algorithm 即,椭圆曲线数字签名算法) 的例子进行说明。ECDSA 是基于椭圆曲线上离散对数问题考虑的签名方法。由于 ECDSA 自身是公知技术,因此省略详细的说明,以下说明对 ECDSA 适用于本发明的情况下的实施例。再有,在下述的图 15 的说明中,除了使用分割随机数信息和结合随机数信息等进行不在存储器上出现秘密信息地计算的处理以外的处理流程,都相当于通常的 ECDSA 签名的生成流程。

[0343] 图 15 是关于适用于本发明时的 ECDSA 的流程概要进行说明的图。

[0344] 再有,适用于本发明时的签名生成单元 100 的结构、签名密钥生成恒等式 G121 和结合分割密钥 141 的制作方法与实施方式 1 相同,故省略说明。

[0345] 在本实施方式 2 中,关于适用在 ECDSA 中时与本实施方式 1 不同的部分进行说明。

[0346] ECDSA 中的签名生成式如下式所示。

[0347] $S = (h+r*d)/k \bmod q$

[0348] 在此, S 是签名, d 是签名密钥, r 是基点 P 的 k 倍点的 x 坐标, q 是基点 P 的位数。

[0349] 此外, h 是 M 的散列值,表示为 $h = \text{Hash}(M)$, M 是签名生成对象消息, $\text{Hash}(M)$ 示出计算签名生成对象消息 M 的散列值。

[0350] 在 ECDSA 中,必须要保密的信息是在步骤 S1502 中利用的随机数 k 和在步骤 S1504 中利用的签名密钥 d 的值。在此,必须要保密随机数 k 的理由是,在知道了签名对象消息 M 和签名 (r, S) 的情况下,能够利用下式计算签名密钥 d。

[0351] $h = \text{Hash}(M)$

[0352] $d = (k*S-h)/r \bmod q$

[0353] 以后,使用图 15,关于保密了签名密钥 d 和随机数 k 这 2 个信息的情况下 ECDSA 的签名生成流程进行说明。再有,未在图 1 的签名生成装置 100 示出签名生成单元装置 100,但具有保持着椭圆曲线的系统参数 ($y^2 = x^3 + a*x + b$ 、定义体 GF(p)、基点 P、基点 P 的位数 q),生成随机数的随机数生成单元。

[0354] 首先,对于签名生成对象消息 M,计算散列值 $\text{Hash}(M)$,将该散列值设为 h (步骤 S1501)。

[0355] 接着,对椭圆曲线的基点 P 产生随机数 k,计算 P 的 k 倍点,将 k 倍后的点设为 R (步

骤 S1502)。这时,由于随机数 k 是秘密信息,因此,生成多个分割随机数信息,使得不直接在存储器中出现随机数 k 的值,仅使用生成的分割随机数信息计算 P 的 k 倍点。以后,使用图 16,关于步骤 S1502 中的该处理进行详细说明。

[0356] 接着,将 R 的 x 坐值设为 r(步骤 S1503)。

[0357] 接着,进行与使用了随机数 k 和签名密钥 d 的式子 $S = (h+r*d)/k \bmod q$ 恒等的运算,不直接在存储器上保持随机数 k 和签名密钥 d 的值,取而代之,在存储器上保持分割随机数信息和已预先分割了签名密钥 d 的分割密钥,使用它们生成签名 S(步骤 S1504)。

[0358] 以后,使用图 17,关于步骤 S1504 中的该处理进行详细说明。

[0359] 最后,根据步骤 S1503 和 S1504 的结果,输出 (r, S) 的对作为签名,并结束 ECDSA 的签名生成。

[0360] 下面,使用图 16 中示出的流程图,关于步骤 S1502 的详细内容进行说明。

[0361] 首先,生成 m 个随机式 $R_1 \dots R_m$,制作运算其积的随机数生成式 T。在此, $R_1 \dots R_m$ 的积相当于随机数 k(步骤 S1601)。

[0362] $T = R_1 * R_2 * \dots * R_m$

[0363] 另外,进一步将 $R_1 \dots R_m$ 分别分割为更细小值(分割随机数信息)的和。

[0364] $T = \sum t(1, h) * \sum t(2, i) * \dots * \sum t(m, j) \dots$ (式 T)

[0365] 在此

[0366] $\sum t(1, h) = t(1, 1) + t(1, 2) + \dots + t(1, h)$

[0367] $\sum t(2, i) = t(2, 1) + t(2, 2) + \dots + t(2, i)$

[0368] $\sum t(m, j) = t(m, 1) + t(m, 2) + \dots + t(m, j)$

[0369] $t(1, 1)、t(1, 2)、\dots、t(1, h)$ 分别是分割随机数信息,式 $\sum t(1, h)$ 表示 h 个分割随机数信息的和。

[0370] 此外, $h, i, j \dots$ 是任意的数。

[0371] 接着,对于随机数生成式 T 的项即 $\sum t(1, h)、\sum t(2, i)、\dots、\sum t(m, j)$ 中使用着的分割随机数信息 $t(x, y)$,即、 $t(1, 1)、t(1, 2)、\dots、t(1, h)、t(2, 1)、\dots、t(2, i)、t(m, 1)、\dots、t(m, j)$,分别生成并设置随机数(步骤 S1602)。

[0372] 接着,判定 $\sum t(1, h)、\sum t(2, i)、\dots、\sum t(m, j)$ 的值是否分别与 q 互为质数(步骤 S1603)。

[0373] 在此,所谓的“与 q 互为质数”的条件是为了随机数 k 具有逆元而必须的条件。

[0374] 若步骤 S1603 的判定结果是“否”,就向步骤 S1602 返回处理,再次进行上述的随机数的生成和设置。

[0375] 反之,若步骤 S1603 的判定结果是“是”,就制作与随机数生成式 T 恒等的随机数生成恒等式 U(步骤 S1604)。

[0376] 由于能够用与实施方式 1 中的从签名密钥生成式 F21 生成签名密钥生成恒等式 G121 的方法同样的方法,来进行步骤 S1604 中的从随机数生成式 T 制作随机数生成恒等式 U,故省略说明。

[0377] 接着,基于随机数生成恒等式 U 制作结合随机数信息(步骤 S1605)。在此,结合随机数信息相当于实施方式 1 中的结合分割密钥。从而,步骤 S1605 的结合随机数信息的制作方法与实施方式 1 中的结合分割密钥的制作相同,故省略说明。

[0378] 接着,使用随机数生成恒等式 U 和在步骤 S1605 中制作的结合随机数信息,计算基点 P 的 U 倍点 U*P,将该计算结果设为 R(步骤 S1606)。

[0379] 这时,在随机数生成恒等式 U 的式子中代入结合随机数信息的值来计算出的结果,对应于在 ECDSA 的本来算法中利用的随机数 k。即,不直接在存储器上出现与 S1502 的随机数 k 相对应的值,而仅利用将多个分割随机数进行结合运算后的值即结合随机数信息,就能够计算出 P 的 U 倍点。

[0380] 步骤 S1606 进行椭圆曲线上的点彼此之间的加法和点的 n 倍点运算。在此,与实施方式 1 对应地说明该计算,在实施方式 1 中, RSA 的签名生成处理的乘方运算中的“乘方运算”对应于椭圆曲线上的“n 倍点运算”, RSA 的签名生成处理的“乘法”对应于椭圆曲线上的加法。从而,在 RSA 的签名生成流程中,利用变更了适用的运算种类后的流程,就能够进行椭圆曲线上的点彼此之间的加法和 n 倍点运算。

[0381] 在此,为了容易理解,关于步骤 S1601 ~ S1606 举具体例进行说明。

[0382] 首先,在步骤 S1601 中,假设用上述方法生成的随机数生成式 T 是以下的式子。

[0383] $T = \sum t(1,4) * \sum t(2,4) \dots$ (式 T)

[0384] $\sum t(1,4) = t(1,1) + t(1,2) + t(1,3) + t(1,4)$

[0385] $\sum t(2,4) = t(2,1) + t(2,2) + t(2,3) + t(2,4)$

[0386] 接着,对 $t(1,1)、t(1,2)、\dots、t(1,4)、t(2,1)、t(2,2)、\dots、t(2,4)$ 分别设定随机数值(步骤 S1602)。

[0387] 这时,重新进行随机数值的设定,直到 $\sum t(1,4)$ 与 $\sum t(2,4)$ 互为质数。(步骤 S1603)

[0388] 接着,根据

[0389] 随机数生成式 $T = (t(1,1) + t(1,2) + t(1,3) + t(1,4))$

[0390] $* (t(2,1) + t(2,2) + t(2,3) + t(2,4))$,

[0391] 利用交换法则、结合法则、分配法则,例如

[0392] $T = (((t(1,1) + t(1,3)) + ((t(1,2) + t(1,4)))$

[0393] $* ((t(2,1) + t(2,3))$

[0394] $+ (((t(1,1) + t(1,4)) + ((t(1,2) + t(1,3))))$

[0395] $* ((t(2,2) + t(2,4)))$

[0396] 这样地将随机数生成恒等式 U 进行恒等变换,将自变量 $u_1 \sim u_6$ 设定为

[0397] $u_1 = t(1,1) + t(1,3)$

[0398] $u_2 = t(1,2) + t(1,4)$

[0399] $u_3 = t(2,1) + t(2,3)$

[0400] $u_4 = t(1,1) + t(1,4)$

[0401] $u_5 = t(1,2) + t(1,3)$

[0402] $u_6 = t(2,2) + t(2,4)$ (步骤 S1605)。

[0403] 这样地,随机数生成恒等式 U 就成为

[0404] $U(u_1, u_2, u_3, u_4, u_5, u_6)$

[0405] $= (u_1 + u_2) * u_3 + (u_4 + u_5) * u_6$ 。

[0406] 在自变量 u_1 中代入结合随机数信息 U_1 ,在自变量 u_2 中代入结合随机数信息 U_2 ,

同样地,在自变量 u3 ~ u6 中代入结合随机数信息 U3 ~ U6。

[0407] $U = (U_1+U_2)*U_3+(U_4+U_5)*U_6$

[0408] 此外,能用

[0409] $U*P = U_3*(U_1*P+U_2*P)+U_6*(U_4*P+U_5*P)$

[0410] 计算基点 P 的 U 倍点 U*P。

[0411] 在此,由于所述随机数生成式 T 和随机数生成恒等式 U 恒等,因此, U*P 与 T*P 成为相同结果。另外,由于 T 是计算随机数 k 的随机数生成式,因此,不在存储器上装载随机数 k 和在其生成中使用的随机数生成式 T,就能计算 P 的 k 倍点。此外,由于随机数生成恒等式 U 和对应的结合随机数信息等每次都随机生成,因此,能够使随机数 k 的值的分析变得困难。这样,实施方式 2 的签名生成单元就能够使 ECDSA 签名生成流程的分析变得困难。

[0412] 接着,使用图 17 说明图 15 的步骤 S1504 的详细流程。

[0413] 首先,生成 m 个随机值,分别作为分割冗余密钥 du1、du2、...、dum,将相加了各分割冗余密钥后的 du1+du2+...+dum 作为分割冗余密钥生成式 Du(步骤 S1701)。

[0414] $Du = du_1+du_2+\dots+du_m \dots$ (式 Du)

[0415] 这是用于不在存储器上直接出现式 $S = (h+r*d)/k \bmod q$ 中的 $1/k$ 而进行计算的。

[0416] 即,由于

[0417] $1/k = Du/(Du*k)$

[0418] $= du_1/(Du*k)+du_2/(Du*k)+\dots+du_m/(Du*k)$

[0419] 并且,在 S1502 中生成的 U 与 k 等效,因此,只要取代 $1/k$,计算

[0420] $du_1/(Du*U)+du_2/(Du*U)+\dots+du_m/(Du*U)$ 就可以。

[0421] 另外,若假设上式的 Du*U 为式 A,用与 Du*U 恒等的式子 B1、B2...Bm 置换 Du*U,则成为

[0422] $1/k = (du_1/B_1)+(du_2/B_2)+\dots+(du_m/B_m),$

[0423] 若将该 (du_1/B_1) 置换为 C1,同样地将 (du_2/B_2) 置换为 C2, ..., 将 (du_m/B_m) 置换为 Cm,则成为

[0424] $1/k = C_1+C_2+\dots+C_m.$

[0425] 结果,为了计算签名 S,只要计算

[0426] $S = (h/k+r*d/k) \bmod n$

[0427] $= (h*(C_1+C_2+\dots+C_m))$

[0428] $+r*F*(C_1+C_2+\dots+C_m)) \bmod n$

[0429] $= (h*(C_1+C_2+\dots+C_m))$

[0430] $+r*H) \bmod n$

[0431] 就可以。

[0432] 在此,上式中的 F 示出了在实施方式 1 中示出的下式

[0433] $F(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8)$

[0434] $= (d_1+d_2+d_3+d_4)*(d_5+d_6+d_7+d_8)$

[0435] 此外,H 是与式子 $F*(C_1+C_2+\dots+C_m)$ 恒等的式子,F 是上述的签名密钥生成式。

[0436] 以下,返回到步骤 S1702 以后的流程图的说明。

[0437] 下面,判定分割冗余密钥生成式 Du 的值是否与 q 互为质数(步骤 S1702)。

[0438] 若步骤 S1702 的判定结果是“否”，为了进一步生成别的 $du_1 \dots du_m$ ，向步骤 S1701 返回处理。

[0439] 反之，若步骤 S1702 的判定结果是“是”，就将在步骤 S1502 中求得的随机数生成恒等式 U 和分割冗余密钥生成式 Du 相乘后的式子 $U*Du$ 作为 A，生成 m 个与 A 恒等的式子，作为 B_1, B_2, \dots, B_m （步骤 S1703）。

[0440] 基于在步骤 S1703 中生成的恒等式 B_1, B_2, \dots, B_m ，生成结合分割密钥 b_{ij} ($i, j = 1, 2, \dots$)（步骤 S1704）。

[0441] 步骤 S1703 和 S1704 将 A 看作实施方式 1 中的签名密钥生成式 F21，将构成了 A 的式子的分割冗余密钥 du_1, du_2, \dots, du_m 和在步骤 S1504 中制作的结合随机数信息 U_1, U_2, \dots, U_i 看作分割密钥 111，通过进行与实施方式 1 同样的处理，能够生成恒等式 B_1, B_2, \dots, B_m 和与各个恒等式相对应的结合分割密钥 b_{ij} ($i, j = 1, 2, \dots$)。

[0442] 再有，在实施方式 1 中，仅制作了 1 个签名密钥生成恒等式 G121，但在此生成 m 个相当于签名密钥生成恒等式 G121 的恒等式 B_1 等，分别制作 m 组结合分割密钥的组。

[0443] 接着，使用在步骤 S1701 中生成的 m 个分割冗余密钥 du_1, du_2, \dots, du_m 和 m 个恒等式 B_1, B_2, \dots, B_m ，制作如下的 m 个 C_1, C_2, \dots, C_m （步骤 S1705）。

[0444] $C_1 = du_1/B_1$

[0445] $C_2 = du_2/B_2$

[0446] $C_m = du_m/B_m$

[0447] 这时，能够利用 B_1, B_2, B_m 还同与 A 恒等的情况和 $Du = du_1+du_2+\dots+du_m$ 的性质，将 $C_1+C_2+\dots+C_m$ 如下地进行式变形。

[0448] $C_1+C_2+\dots+C_m$

[0449] $= (du_1/B_1)+(du_2/B_2)+\dots+(du_m/B_m)$

[0450] $= (du_1+du_2+\dots+du_m)/A$

[0451] $= (du_1+du_2+\dots+du_m)/U*Du$

[0452] $= 1/U$

[0453] 由于 U 相当于步骤 S1502 的随机数 k，因此， $1/U \bmod q$ 的值就相当于 $k^{-1} \bmod q$ 。这样，就能够不在存储器上装载 k 的值，而计算 $k^{-1} \bmod q$ 的值。

[0454] 此外，由于随机地生成恒等式 $B_1 \dots B_m$ 和分割冗余密钥 du_1, du_2, \dots, du_m 等，因此，能够每次进行签名生成时都使 $k^{-1} \bmod q$ 的计算过程动态地变化。这样，实施方式 2 的签名生成装置就能够使 ECDSA 签名生成流程的分析变得困难。

[0455] 接着，使用签名密钥生成式 F 和在步骤 S1705 中生成的 m 个 C_i ($i = 1, 2, \dots, m$) 式，生成（式 E）（步骤 S1706）。

[0456] $F*(C_1+C_2+\dots+C_m) \dots$ (式 E)

[0457] 接着，生成与（式 E）恒等的（式 H'）（步骤 S1707）。

[0458] 接着，基于（式 H'），使用实施方式 1 中说明过的流程，对签名密钥生成式 F 的要素即分割密钥和 C_1, C_2, \dots, C_m 的要素值进行结合运算，将结合运算后的值作为结合分割密钥（步骤 S1708）。再有，由于通过将实施方式 1 中的签名密钥生成式 F21 看作（式 E），将 C_1, \dots, C_m 的要素值和签名密钥生成式 F21 的要素值 di 看作实施方式 1 中的签名密钥生成式 F21 的要素值 di ，步骤 S1807 和步骤 S1708 的处理就成为与实施例 1 同样的处理，故

省略详细的说明。这样做，恒等式 H' 就能够表示作使用了结合分割密钥的 H 。

[0459] 这样，由于每次随机地生成对应于恒等式 H 的结合分割密钥 h_i ，因此，能够使 ECDSA 签名生成流程的分析变得困难。

[0460] 接着，使用在步骤 S1501 中计算出的散列值 h 、在步骤 S1705 中生成的式子 C_1 、 C_2 、 \dots 、 C_m 、在步骤 S1707 中生成的恒等式 H 、在步骤 S1503 中计算出的点 R 的 X 坐标的值 r ，用下式

$$[0461] S = h*(C_1+C_2+\dots+C_m)+r*H \bmod q$$

[0462] 计算与

$$[0463] S = (h+r*d)/k \bmod q$$

[0464] 恒等的运算，将计算值作为 S （步骤 S1709）。

[0465] 以上，实施方式 2 的签名生成装置能够不直接在存储器上出现随机数值 k 的值和签名密钥 d 的值而生成签名 S 。

[0466] 再有，在步骤 S1709 的运算中，期望按照 $h*(C_1+C_2+\dots+C_m)$ 的运算结果值不直接出现在存储器上的顺序进行运算。

[0467] 这是因为能从 $h*(C_1+C_2+\dots+C_m)$ 的值复原随机数信息即 k 。

[0468] 关于运算顺序以后叙述具体例，故省略在此的说明。

[0469] 接着，举具体例说明图 17 的流程。

[0470] 首先，生成 3 个随机数，分别作为分割冗余密钥 du_1 、 du_2 、 du_3 （步骤 S1701）。

$$[0471] Du = du_1+du_2+du_3$$

[0472] 在此，在 Du 与 q 不是互为质数的情况下，反复进行分割冗余密钥的生成，直到 Du 与 q 互为质数（步骤 S1702）。

[0473] 接着，转移到使用了随机数生成恒等式 U 的计算的说明，但在以下的说明中，假设随机数生成恒等式 U 为在图 16 的说明中生成的以下的（式 U ）。

$$[0474] U = (U_1+U_2)*U_3+(U_4+U_5)*U_6\dots \text{ (式 } U\text{)}$$

[0475] 这时，（式 A）例如变为如下。

$$[0476] ((U_1+U_2)*U_3+(U_4+U_5)*U_6)$$

$$[0477] *(du_1+du_2+du_3) \dots \text{ (式 } A\text{)}$$

[0478] 接着，生成与（式 A）恒等的（式 B1）、（式 B2）、（式 B3）。

[0479] 作为恒等式 B1 的计算过程，首先生成如下的式 A 的恒等式 1，

$$[0480] (((U_1+U_2)*U_3)+((U_4+U_5)*U_6))*(du_1+du_2)$$

$$[0481] +((U_1+U_2)*U_3+(U_4+U_5)*U_6)*du_3$$

[0482] 基于生成的恒等式 1，使结合分割密钥分别如下。

$$[0483] b_{11} = (U_1+U_2)*U_3$$

$$[0484] b_{12} = (U_4+U_5)*U_6$$

$$[0485] b_{13} = du_1+du_2$$

$$[0486] b_{14} = (U_1+U_2)*U_3+(U_4+U_5)*U_6$$

$$[0487] b_{15} = du_3$$

[0488] 于是，使用式 A 的恒等式 1 和上述结合分割密钥，恒等式 B1 变为以下的（式 B1）。

$$[0489] (b_{11}+b_{12})*b_{13}+b_{14}*b_{15}\dots \text{ (式 } B1\text{)}$$

- [0490] 同样地,作为恒等式 B2 的计算过程,首先生成如下的式 A 的恒等式 2,
- [0491] $(U1*U3+U2*U3+U4*U6+U5*U6)$
- [0492] $*((du1)+(du2+du3))$
- [0493] 关于生成的恒等式 B2,使结合分割密钥分别如下。
- [0494] $b21 = U1*U3$
- [0495] $b22 = U2*U3$
- [0496] $b23 = U4*U6$
- [0497] $b24 = U5*U6$
- [0498] $b25 = du1$
- [0499] $b26 = du2+du3$
- [0500] 于是,使用式 A 的恒等式 2 和上述结合分割密钥,恒等式 B2 变为以下的(式 B2)。
- [0501] $B2 = (b21+b22+b23+b24)*(b25+b26)\dots$ (式 B2)
- [0502] 另外,同样地,作为恒等式 B3 的计算过程,首先生成如下的式 A 的恒等式 3,
- [0503] $((U1*U3+U4*U6)+(U2*U3+U5*U6))$
- [0504] $*((du1+du2+du3))$
- [0505] 按照生成的恒等式 3,使结合分割密钥分别如下。
- [0506] $b31 = U1*U3+U4*U6$
- [0507] $b32 = U2*U3+U5*U6$
- [0508] $b33 = du1+du2+du3$
- [0509] 于是,使用式 A 的恒等式 3 和上述结合分割密钥,恒等式 B3 变为以下的(式 B3)。
- [0510] $(b31+b32)*b33\dots$ (式 B3)
- [0511] 接着,使用生成的(式 B1)、(式 B2)、(式 B3),在步骤 S1705 中生成以下的 3 个式子 C1、C2、C3。
- [0512] $C1 = du1/B1 = du1/((b11+b12)*b13+b14*b15)\dots$ (式 C1)
- [0513] $C2 = du2/B2 = du2/((b21+b22+b23+b24)*(b25+b26))\dots$ (式 C2)
- [0514] $C3 = du3/B3 = du3/((b31+b32)*b33)\dots$ (式 C3)
- [0515] 这时,如上所述地 $C1+C2+C3 = k^{\wedge}(-1) \bmod q$ 成立。
- [0516] 在此,将 ECDSA 的签名密钥 d 分割为 8 个分割密钥,设签名密钥生成式 F 为如下的(式 F)。
- [0517] $F = (d1+d2+d3+d4)*(d5+d6+d7+d8)\dots$ (式 F)
- [0518] 这时,接着在步骤 S1706 中生成的(式 E)成为
- [0519] $(d1+d2+d3+d4)*(d5+d6+d7+d8)*(C1+C2+C3)$ (步骤 S1706)。接着,从(式 E)生成与(式 E)恒等的(式 H)(步骤 S1707)。在此,设(式 E)的恒等变换例为
- [0520] $H = ((d1+d3)+(d2+d4))*((d6+d7)+(d8+d5))*C1$
- [0521] $+((d2+d3)+(d1+d4))*((d6+d8)+(d7+d5))*(C2+C3),$
- [0522] 设结合分割密钥分别为
- [0523] $H1 = d1+d3$
- [0524] $H2 = d2+d4$
- [0525] $H3 = d6+d7$

[0526] $H_4 = d_8 + d_5$

[0527] $H_5 = C_1$

[0528] $H_6 = d_2 + d_3$

[0529] $H_7 = d_1 + d_4$

[0530] $H_8 = d_6 + d_8$

[0531] $H_9 = d_7 + d_5$

[0532] $H_{10} = C_2 + C_3$

[0533] 该情况下,利用上述的结合信息,恒等式 H 成为如下的(式 H)。

[0534] $((H_1 + H_2) * (H_3 * H_4)) * H_5$

[0535] $+ ((H_6 + H_7) * (H_8 * H_9)) * H_{10} \dots$ (式 H)

[0536] 接着,使用交换法则、结合法则、分配法则,计算

[0537] $h * (C_1 + C_2 + C_3) + r * H$

[0538] $= h * (C_1 + C_2 + C_3) + r * \{ ((H_1 + H_2) * (H_3 * H_4)) * H_5 +$

[0539] $((H_6 + H_7) * (H_8 * H_9)) * H_{10} \}$,

[0540] 使得 $h * (C_1 + C_2 + C_3)$ 的值不直接出现在存储器上,将计算结果作为签名 S。作为使得 $h * (C_1 + C_2 + C_3)$ 的值不直接出现在存储器上的计算顺序的一例,变形为下式进行计算。

[0541] $h * (C_1 + C_2 + C_3) + r * \{ ((H_1 + H_2) * (H_3 * H_4)) * H_5 +$

[0542] $((H_6 + H_7) * (H_8 * H_9)) * H_{10} \}$

[0543] $= h * (C_1 + C_2) + r * \{ ((H_1 + H_2) * (H_3 * H_4)) * H_5 \}$

[0544] $+ h * C_3 + r * \{ ((H_6 + H_7) * (H_8 * H_9)) * H_{10} \}$

[0545] 即,在上述的计算式中,将 $h * (C_1 + C_2 + C_3)$ 分为 $h * (C_1 + C_2)$ 和 $h * C_3$ 进行计算,并且,通过在该计算之间夹入剩余的计算,使 $h * (C_1 + C_2 + C_3)$ 的值就不直接出现在存储器上。

[0546] 通过如上所述的过程,就能不在存储器上直接出现相当于步骤 S1502 的随机数 k 的值和签名密钥 d 的值,并且,能在每次执行签名生成时,使在每次执行签名 S 的生成中使用的随机数生成恒等式 U 和签名生成式 H 改变,因此,具有利用非法分析确定签名密钥 d 变得非常困难的效果。

[0547] 再有,在本实施方式 1 和 2 中,关于 RSA 签名和 ECDSA 进行了说明,但不限定于此,也可以利用在使用了 RSA 密码、椭圆密码、ElGamal 密码等计算运算的公开密钥密码中。此外,由本实施方式 1 和 2 例示的过程的应用目的不必限于在密码和签名中利用的密钥信息,当然也可以对于不同于它们的需要保护的秘密信息进行适用。

[0548] (实施方式 3)

[0549] 以下,关于本发明的实施方式 3 进行说明。

[0550] 在本实施方式 3 中,将签名密钥分割为分割密钥,使用图 18 说明直到将分割密钥嵌入到终端中的工序。

[0551] 密钥发行机关是发行用于进行签名生成的签名密钥的机关。

[0552] 在密钥发行机关中,向 n 台(信息终端 1(1810)、信息终端 2(1820)、...、信息终端 n(1830))终端发行各自不同的签名密钥 1、签名密钥 2、...、签名密钥 n。

[0553] 分割密钥生成装置 1802 是将签名密钥生成式作为输入,基于签名密钥生成式,将从密钥发行机关发行的签名密钥进行分割的装置,输出分割密钥和签名密钥生成式。

[0554] 分割密钥写入装置 1805 是向进行签名生成的信息终端写入在分割密钥生成装置 1802 中生成的分割密钥和由签名密钥生成式构成的分割密钥信息 1804 的装置。例如，在信息终端是内装了便携式电话机等的机器的情况下，使用 ROM 记录器作为分割密钥写入装置。

[0555] 在此，信息终端 1(1810)、信息终端 2(1820)、…、信息终端 n(1830) 是进行签名生成的信息终端。由于这些信息终端的结构是与实施方式 1 中说明的签名生成单元 100 相同的结构，因此仅图示了本实施方式 3 的说明中必要的结构。

[0556] 制造终端的制造厂对分割密钥生成装置 1802 输入从密钥发行机关发行的签名密钥 1801 和签名密钥生成式 1803，生成分割密钥信息 1804，所述分割密钥信息 1804 包括分割了签名密钥 1801 的分割密钥和用于从分割密钥生成签名密钥的信息即签名密钥生成式 1803。

[0557] 分割密钥信息 1804 包括签名密钥 1 的分割密钥信息 1811、签名密钥 2 的分割密钥信息 1821、签名密钥 n 的分割密钥信息 1831，示出了签名密钥 1 的分割密钥信息 1811 是面向信息终端 1810 的分割密钥信息，签名密钥 2 的分割密钥信息 1821 是面向信息终端 2(1820) 的分割密钥信息，签名密钥 n 的分割密钥信息 1831 是面向信息终端 n(1830) 的分割密钥信息。分割密钥信息 (1811、1821、1831) 也可以作为实施方式 1 中示出的分割密钥识别信息表 200 和分割密钥信息表 400，此外，只要是能够识别分割密钥和签名密钥生成式 F 的关系以便能够计算出与签名密钥 d 相同的值的信息，就也可以是其他信息。

[0558] 接着，制造终端的制造厂使用分割密钥写入装置 1805，向信息终端 1810、1820、1830 的各分割密钥存储单元中写入由分割密钥生成装置 1802 生成的分割密钥信息 1804。

[0559] 在此说明直到写入为止的更具体的一例，由分割密钥生成装置生成的分割密钥信息 (1811、1821、1831)，是表现了分割密钥识别信息表 200 和分割密钥信息表 400 的数据文件，将该数据文件编译来作为二进制数据，使用 ROM 记录器等分割密钥写入装置 1805，将该二进制数据写入到分割密钥存储单元中。

[0560] 以上，就在信息终端 1(1810) 的分割密钥存储单元中写入签名密钥 1 的分割密钥信息 1811，在信息终端 2(1820) 的分割密钥存储单元中写入签名密钥 2 的分割密钥信息 1821，在信息终端 n(1830) 的分割密钥存储单元中写入签名密钥 n 的分割密钥信息 1831。

[0561] 再有，在本实施方式 3 中，信息终端 (1810、1820、1830) 的签名密钥生成式设定为各自不同的签名密钥生成式 F1、F2、…Fn，但也可以在全部信息终端中设定相同的签名密钥生成式。该情况下，为了使各信息终端进行各不相同的处理，最好由分割密钥生成装置 1802 对每个信息终端生成不同的分割密钥，将生成的按每个信息终端各不相同的分割密钥写入到各信息终端 (1810、1820、1830) 中。

[0562] 再有，在本实施方式 3 中，分割密钥生成装置 1802 从外部输入签名密钥生成式 1803，基于输入的签名密钥生成式分割了签名密钥，但输入的信息不限于此，也可以是用于从签名密钥生成分割密钥的任意的参数信息。例如，也可以向分割密钥生成装置 1802 输入分割签名密钥的分割个数的参数，按照在分割密钥生成装置 1802 内输入的分割个数，分割签名密钥。另外，也可以按照在分割密钥生成装置 1802 内输入的分割个数生成签名密钥生成式，并将其输出。在此，为了生成分割密钥而输入的参数也可以不是分割个数，而是表示安全等级和 CPU 性能的参数信息，该情况下，也可以按照输入的参数，在分割密钥生成装置

1802 内决定分割个数。具体地说，也可以在安全等级低或 CPU 性能高的情况下，生成更多的分割密钥。通过这样做，就能调整签名密钥 d 的分割数量，因此能够灵活地设定安全强度。

[0563] 此外，也可以在由密钥发行机关加密后的状态下发行签名密钥 1801。该情况下，制造终端的制造厂在对已加密的签名密钥 1801 进行解密之后，将解密后的签名密钥输入到分割密钥生成装置 1802 中。此外，若此处的加密方式利用公开密钥加密方式，则密钥发行机关也可以在签名密钥 1801 中添加签名密钥 1801 的签名后进行发送，制造厂通过利用密钥发行机关的公开密钥验证签名密钥 1801 的签名，就能够验证合法性。通过这样做，就能够保护从密钥发行机关交给制造厂的签名密钥 1801 不在递交途中被盗听和窜改。

[0564] 再有，已对终端制造制造厂利用分割密钥生成装置 1802 和分割密钥写入装置 1805 进行了说明，但现实中，在制造工序中有分割密钥生成装置 1802 和分割密钥写入装置 1805 的利用场合不同的情况。作为具体例，有在终端的制造工厂中实施分割密钥写入装置 1805 的情况等。该情况下，也可以将在分割密钥生成装置 1802 中生成的分割密钥信息 1804 加密，将加密后的分割密钥信息分发给终端的制造工厂，在终端的制造工厂中对分发的加密信息进行解密，将解密了的分割密钥信息 1804 输入到分割密钥写入装置 1805 中。通过这样做，能进行安全的分割密钥的分发。

[0565] 再有，在上述中示出了签名密钥 1801 和分割密钥信息 1804 的加密方法，但关于该情况下的加密方法不应该特殊限定，只要通过使用 AES 等共通密钥密码或者 RSA 和椭圆密码等公开密钥密码来实现就可以。

[0566] (实施方式 4)

[0567] 在实施方式 4 中，使用图 19 和图 20，关于利用签名密钥生成恒等式生成单元 120，通过网络更新计算签名密钥生成恒等式的签名密钥生成恒等式生成程序的方法进行说明。

[0568] 图 19 是签名生成单元 100 从位于网络上的更新服务器 1901 下载更新用的签名密钥生成恒等式生成程序 X1902，更新签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序的概要图。与实施方式 1 的不同点在于，实施方式 4 能够从网络上的更新服务器下载新的签名密钥生成恒等式生成程序，将签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序更新为下载后的签名密钥生成恒等式生成程序 X1902。

[0569] 在此，签名生成单元 100 和更新服务器 1901 通过因特网等网络 1900 进行连接。

[0570] 更新服务器 1901 保持着与签名生成单元 100 的签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序不同的签名密钥生成恒等式生成程序 X1902。在此，签名密钥生成恒等式生成程序 X1902 具有与签名生成程序同等的功能，即、生成与签名密钥生成式 F21 恒等的式子即签名密钥生成恒等式 G121 的功能。但是，假设签名密钥生成恒等式生成程序 X1902 制作签名密钥生成恒等式 G121 的流程不同于签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序。

[0571] 此外，更新服务器 1901 也保持着为了验证更新用的程序即签名密钥生成恒等式生成程序 X1902 的合法性而利用的签名密钥生成恒等式生成程序 X 的窜改检测值 1903。再有，作为窜改检测值 1903 的具体例，有签名密钥生成恒等式生成程序 X 的散列值等。

[0572] 签名生成单元 100 除了具备在实施方式 1 的图 1 中说明的结构要素之外，在本实施方式 4 中还具有收发单元 1910 和签名密钥生成恒等式生成程序更新单元 1920。

[0573] 收发单元 1910 是通过网络与更新服务器 1901 进行数据的收发的装置。

[0574] 签名密钥生成恒等式生成程序更新单元 1920 利用收发单元 1910, 向存在于网络上的更新服务器 1901 发送签名密钥生成恒等式生成程序的更新请求消息。另外, 通过收发单元 1910, 从更新服务器 1901 接收签名密钥生成恒等式生成程序 X1902, 用接收到的签名密钥生成恒等式生成程序 X1902 更新签名密钥生成恒等式生成单元 120 的程序。

[0575] 下面, 使用图 20, 关于签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序的更新流程进行说明。

[0576] 首先, 签名密钥生成恒等式生成程序更新单元 1920 向收发单元 1910 请求签名密钥生成恒等式生成程序 X1902 的下载 (步骤 S2001)。

[0577] 接着, 收到了下载请求的收发单元 1910, 通过网络 1900 向更新服务器 1901 发送签名密钥生成恒等式生成程序 X1902 的下载请求消息 (步骤 S2002)。

[0578] 接着, 收到了下载请求消息的更新服务器 1901, 向收发单元 1910 发送在更新服务器 1901 内保持着的签名密钥生成恒等式生成程序 X1902 和签名密钥生成恒等式生成程序 X1902 的散列值 1903 (步骤 S2003)。

[0579] 接着, 收发单元 1910 在从更新服务器 1901 接收完了签名密钥生成恒等式生成程序 X1902 和签名密钥生成恒等式生成程序 X1902 的散列值 1903 的时刻, 将下载完了通知通知给签名密钥生成恒等式生成程序更新单元 1920 (步骤 S2004)。

[0580] 接着, 签名密钥生成恒等式生成程序更新单元 1920 利用签名密钥生成恒等式生成程序 X1902 的散列值 1903, 判定是否窜改了签名密钥生成恒等式生成程序 X1902。在该判定结果是“是”, 即判定为已窜改的情况下, 结束更新流程, 在该判定结果是“否”, 即判定为未窜改的情况下, 继续更新处理 (步骤 S2005)。

[0581] 接着, 签名密钥生成恒等式生成程序更新单元 1920 通过用下载的签名密钥生成恒等式生成程序 X1902 进行写入, 来更新签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序 (步骤 S2005)。

[0582] 写入一完了, 签名密钥生成恒等式生成程序更新单元 1920 就通知写入完了通知, 保持着结束更新流程 (步骤 S2006)。

[0583] 如上所述地, 签名生成单元 100 利用网络 1900, 从更新服务器 1901 下载新的签名密钥生成恒等式生成程序 X1902 后, 进行签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式程序的更新。

[0584] 通过具有这样的更新功能, 通过在非法分析者窜改了签名密钥生成恒等式程序的情况和签名密钥生成恒等式生成程序不妥的情况下, 更新签名密钥生成恒等式程序, 就能够确保签名生成装置 100 的安全性。

[0585] 再有, 在实施方式 4 中, 关于发送图 20 的步骤 S2001 的下载请求的定时没有示出, 但也可以例如在签名生成处理开始的请求之前, 在签名生成装置 100 内验证签名密钥生成恒等式生成单元 120 的程序 (相当于签名密钥生成恒等式生成程序 X) 的合法性, 在验证结果判定为“不合法”的情况下, 进行步骤 S2001 的下载请求。作为验证签名密钥生成恒等式生成单元 120 的程序的合法性的更具体的方法, 有在签名生成单元 100 内的安全的存储器中保持签名密钥生成恒等式生成单元 120 的程序的散列值, 在开始签名生成处理之前, 计算签名密钥生成恒等式生成单元 120 的签名密钥生成恒等式生成程序的散列值, 比较计算出的散列值和存储器内保持着的散列值的方法等。

[0586] 此外,在实施方式4中,也可以在安全通信线路(SAC)中共有会话密钥,将签名生成单元100与更新服务器1901间的数据通信作为用会话密钥加密后的数据的收发。关于SAC,由于是在Secure Sockets Layer(SSL)等中利用着的已知的技术,故省略说明。

[0587] 此外,在实施方式4中,通过网络,从更新服务器1901下载了更新用的签名密钥生成恒等式生成程序X1902,但也可以不是网络,而是通过使用记录了签名密钥生成恒等式生成程序X1902的DVD和CD等的记录介质,来更新签名密钥生成恒等式生成单元120的程序。通过使图20中的收发单元1910具有操作与记录介质的数据的读写的驱动器功能,能够实现进行该处理的签名生成单元100。即,通过在图20中,将收发单元1910作为记录介质驱动器,将更新服务器1901作为记录媒介,将下载请求置换为来自媒体的读入请求,就能够实现如上所述的签名生成单元100,因此省略详细的说明。

[0588] 此外,在实施方式4中更新了签名密钥生成恒等式生成单元120的程序,但也可以将进行更新的对象设为分割密钥存储单元中存储的分割密钥111。该情况下,不仅更新分割密钥111,也同时更新可知分割密钥111和签名密钥生成式F21的关联信息的分割密钥识别信息表200和分割密钥信息表400。此外,也可以将进行更新的对象设为结合分割密钥生成单元的程序。

[0589] 通过这样做,更新服务器就能调整生成的分割密钥的个数和结合分割密钥的个数,能灵活地设定安全强度。

[0590] <重写>

[0591] 再有,基于上述实施方式说明了本发明,但本发明不限定于上述的实施方式。如下的情况也包含在本发明中。

[0592] (1) 上述的装置具体地说是由微处理器、ROM、RAM、硬盘单元、显示单元、键盘、鼠标等构成的计算机系统。在上述RAM或上述硬盘单元中存储有计算机程序。通过上述微处理器按照上述计算机程序进行工作,上述装置就达到其功能。在此,为了达到规定的功能,组合多个示出对计算机的指令的命令代码来构成了计算机程序。

[0593] (2) 构成上述的各装置的结构要素的一部分或全部也可以由1个系统LSI(Large Scale Integration:大规模集成电路)构成。系统LSI是在1个芯片上集成多个结构部来制造的超多功能LSI,具体地说是由包含微处理器、ROM、RAM等而构成的计算机系统。在上述RAM中存储有计算机程序。通过上述微处理器按照上述计算机程序进行工作,系统LSI就达到其功能。可以将它们个别地集成为1个芯片,也可以包含其中一部分或全部集成为1个芯片。

[0594] 此外,在此设定为系统LSI,但有时也根据集成度的不同,称作IC、系统LSI、超级LSI、极级(ultra)LSI。

[0595] 此外,电路集成化的方法不限于LSI,也可以用专用电路或通用处理器来实现。也可以在LSI制造后,利用可编程的FPGA(Field Programmable GateArray:现场可编程门阵列)和可以再组合LSI内部的电路单元的连接和设定的可重构处理器。

[0596] 另外,若出现了利用半导体技术的进步或派生的其他技术置换为LSI的集成电路化的技术,当然也可以使用该技术进行功能块的集成化。也能有适应生物技术等的可能性。

[0597] (3) 在本实施方式1中,分割密钥生成装置22写入分割密钥识别信息表200和分割密钥信息表400,但当然也可以在签名生成装置100内部生成这些表。

[0598] 再有,由于具有利用非法分析从分割密钥信息表 400 的内容确定签名密钥生成式 F21 的危险,因此,作为更安全的安装方法,也可以在已加密的状态下保持分割密钥信息表 400,仅在生成签名时进行解密。这样,由于利用静态解析的签名密钥生成式 F21 的确定变得困难,因此,安全性提高。此外,若不确定签名密钥生成式 F21,签名密钥 d 的确定就也困难。

[0599] (4) 在实施方式中,关于首先生成签名密钥生成恒等式,接着生成符合上述签名生成恒等式的结合分割密钥的情况进行了说明,但不限于此,只要遵从签名密钥生成恒等式和结合分割密钥所能生成的过程就可以。

[0600] 例如,也可以首先随机地生成结合分割密钥,接着生成包含相当于该结合分割密钥的部分的签名生成恒等式。

[0601] (5) 构成上述装置的结构要素的一部分或全部也可以由可拆装在装置上的 IC 卡或单体模块构成。上述 IC 卡或上述模块是由微处理机、ROM、RAM 等构成的计算机系统。上述 IC 卡或上述模块也可以包括上述超多功能 LSI。通过微处理机按照计算机程序进行工作,上述 IC 卡或上述模块就达到其功能。该 IC 卡或该模块也可以具有抗窜改性。

[0602] (6) 本发明也可以是上述所示的方法。此外,也可以是利用计算机实现这些方法的计算机程序,也可以是由上述计算机程序构成的数字信号。

[0603] 此外,本发明也可以将上述计算机程序或上述数字信号记录在计算机可读取的记录介质中,例如软磁盘、硬盘、CD-ROM、MO、DVD、DVD-ROM、DVD-RAM、BD(Blu-ray Disc)、半导体存储器等中。此外,也可以是这些记录介质中记录着的上述数字信号。

[0604] 此外,本发明也可以经由电气通信线路、无线或有线通信线路、以因特网为代表的网络、数据广播等,来传送上述计算机程序或上述数字信号。

[0605] 此外,本发明也可以是具有微处理机和存储器的计算机系统,上述存储器存储有上述计算机程序,上述微处理机按照上述计算机程序进行工作。

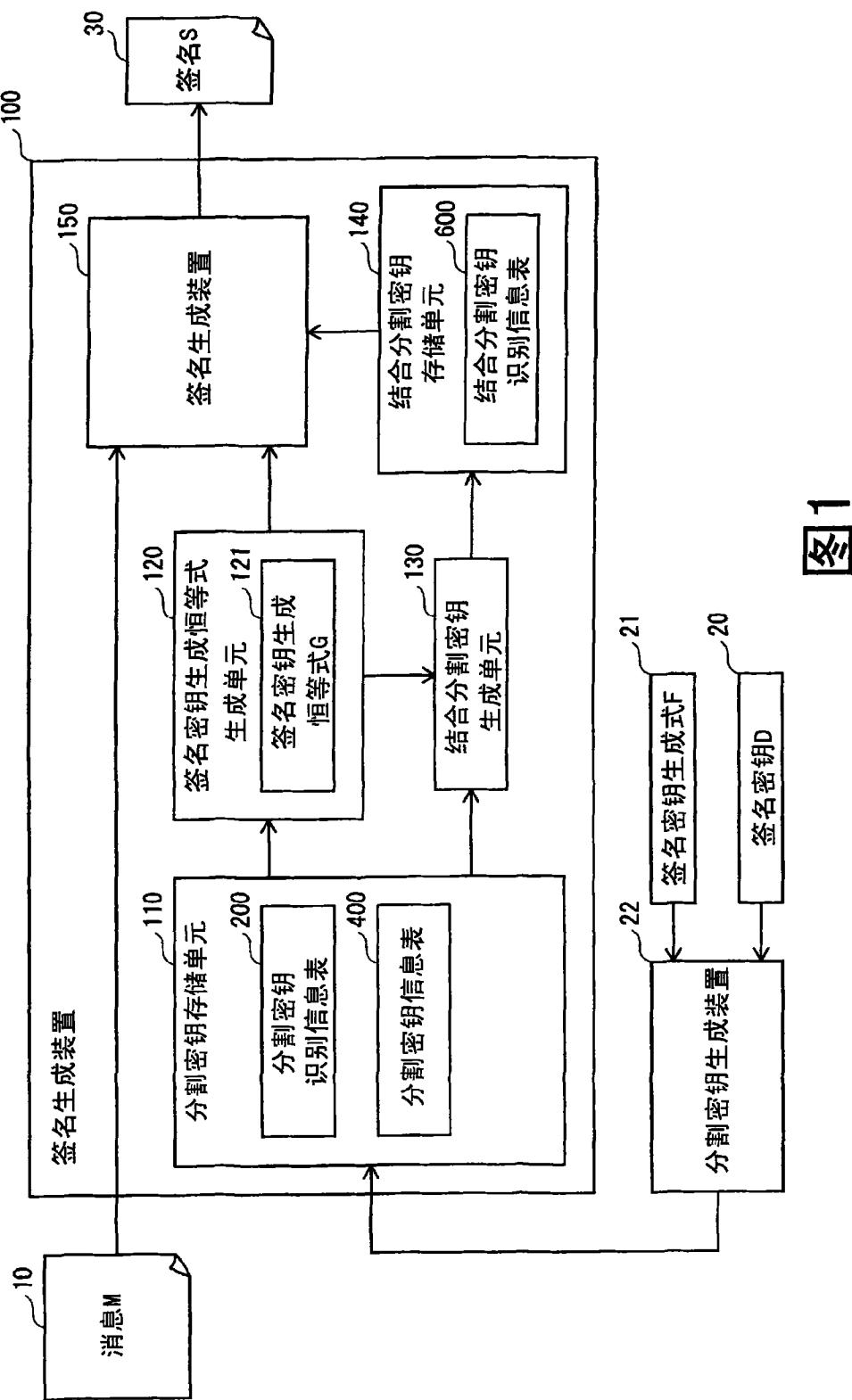
[0606] 此外,也可以通过记录在上述记录介质中来移送上述程序或上述数字信号,或者通过经由上述网络等移送上述程序或者上述数字信号,利用独立的其他计算机系统来实施。

[0607] (7) 也可以分别组合上述实施方式和上述变形例。

[0608] (8) 说明书中的有关用语“交换法则”、“分配法则”、“结合法则”、“逆波兰表示法”的描述,只不过是为便于描述而描述了公知的用语,不是对这些用语给予新的定义。

[0609] 工业上的实用性

[0610] 本发明涉及的安全处理装置和方法通过从运算结果仅导出分割后的秘密信息,具有不在程序的执行中出现秘密信息而能够执行安全处理的效果。此外,还具有每次在执行利用秘密信息的安全处理时,动态地生成每次不同的分割秘密信息,通过使安全处理的执行流程动态地变更,使非法分析者的动态分析变得困难的效果。因此,在进行若泄露给非法分析者就导致利益损失的、使用了秘密信息的处理的软件等的领域中有用。



200 分割密钥识别信息表

201 111

| 分割密钥标识符 | 分割密钥 |
|---------|------|
| ID001 | D1 |
| ID002 | D2 |
| ID003 | D3 |
| ID004 | D4 |
| ID005 | D5 |
| ID006 | D6 |
| ID007 | D7 |
| ID008 | D8 |

图 2

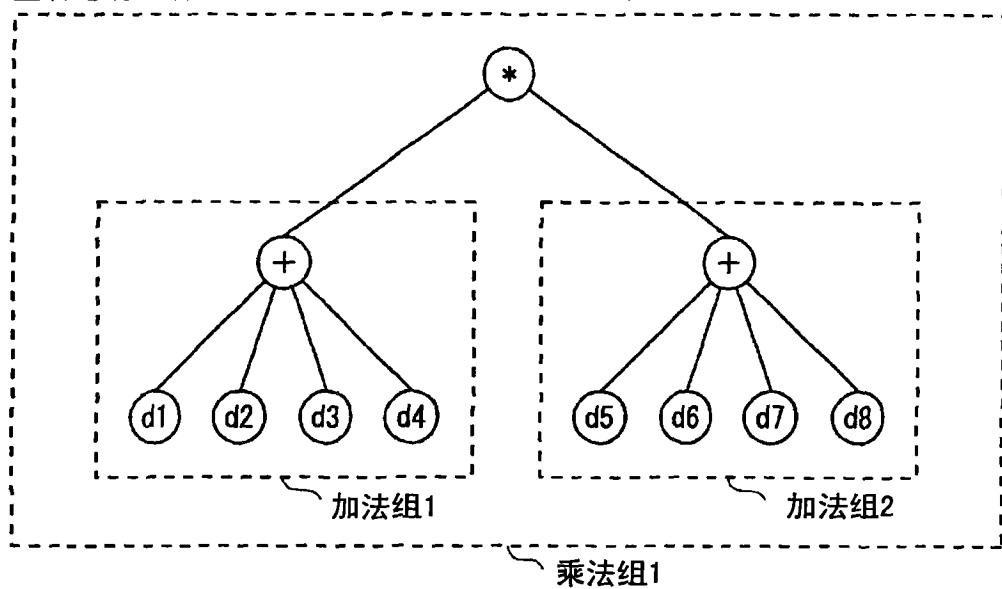
签名密钥生成式 F : $(d_1+d_2+d_3+d_4) * (d_5+d_6+d_7+d_8)$ 

图 3

400 分割密钥信息表

| 分割密钥组标识符 | 分割密钥组成员标识符 |
|-------------|----------------------------|
| 乘法组1： MG001 | AG001, AG002 |
| 加法组1： AG001 | id001, id002, id003, id004 |
| 加法组2： AG002 | id005, id006, id007, id008 |

图 4

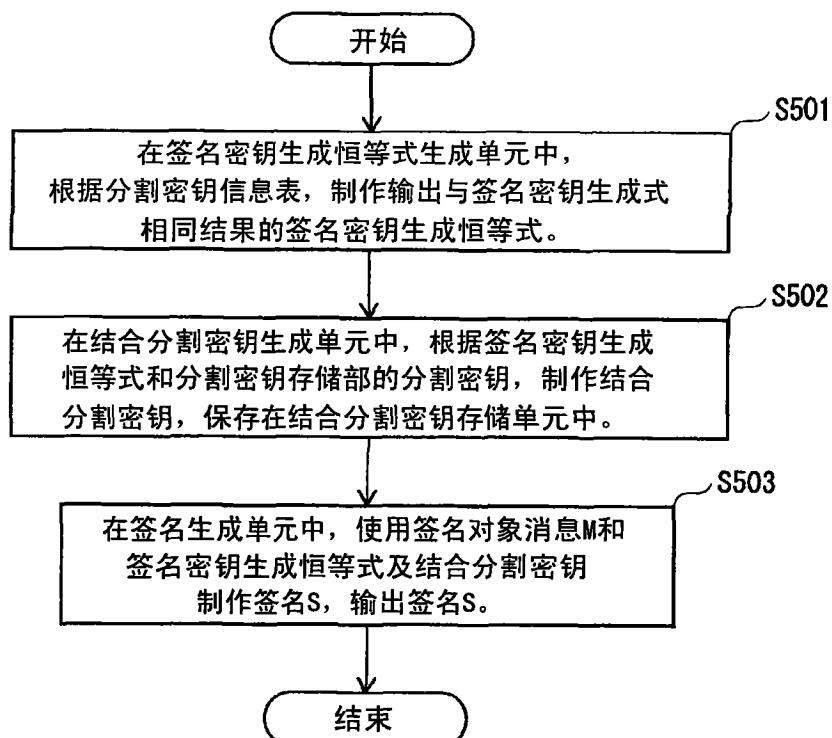


图 5

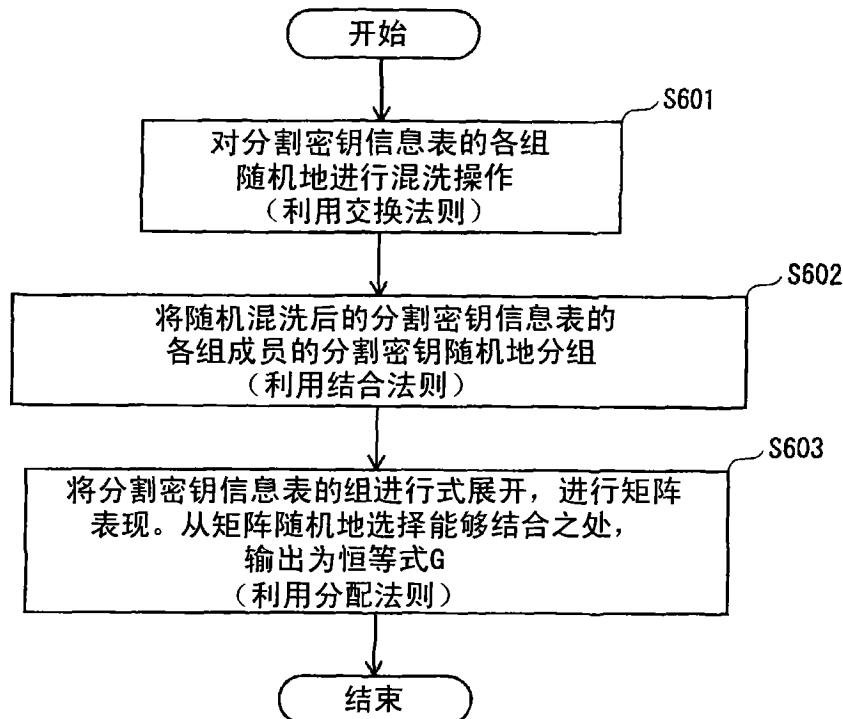


图 6

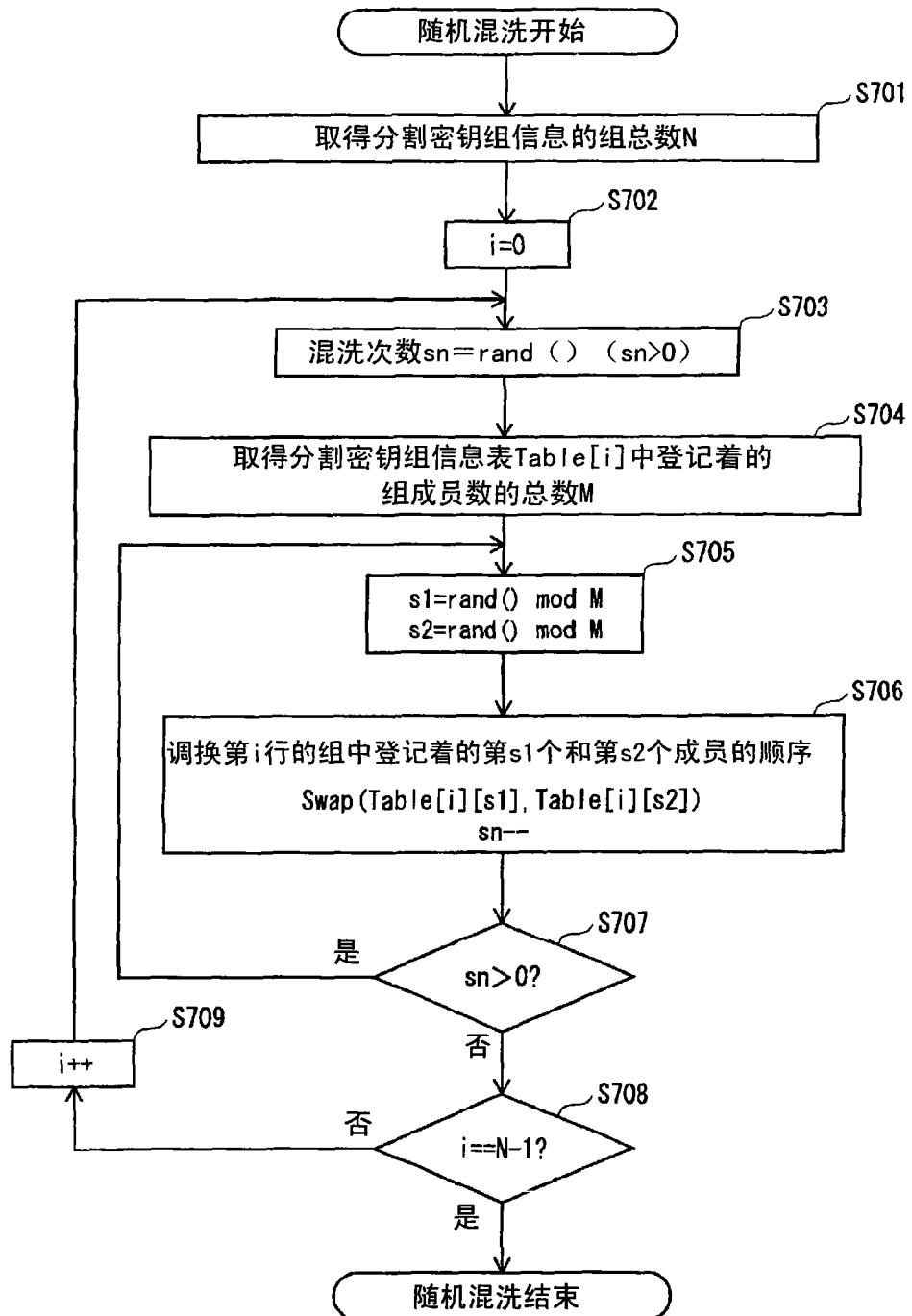


图 7

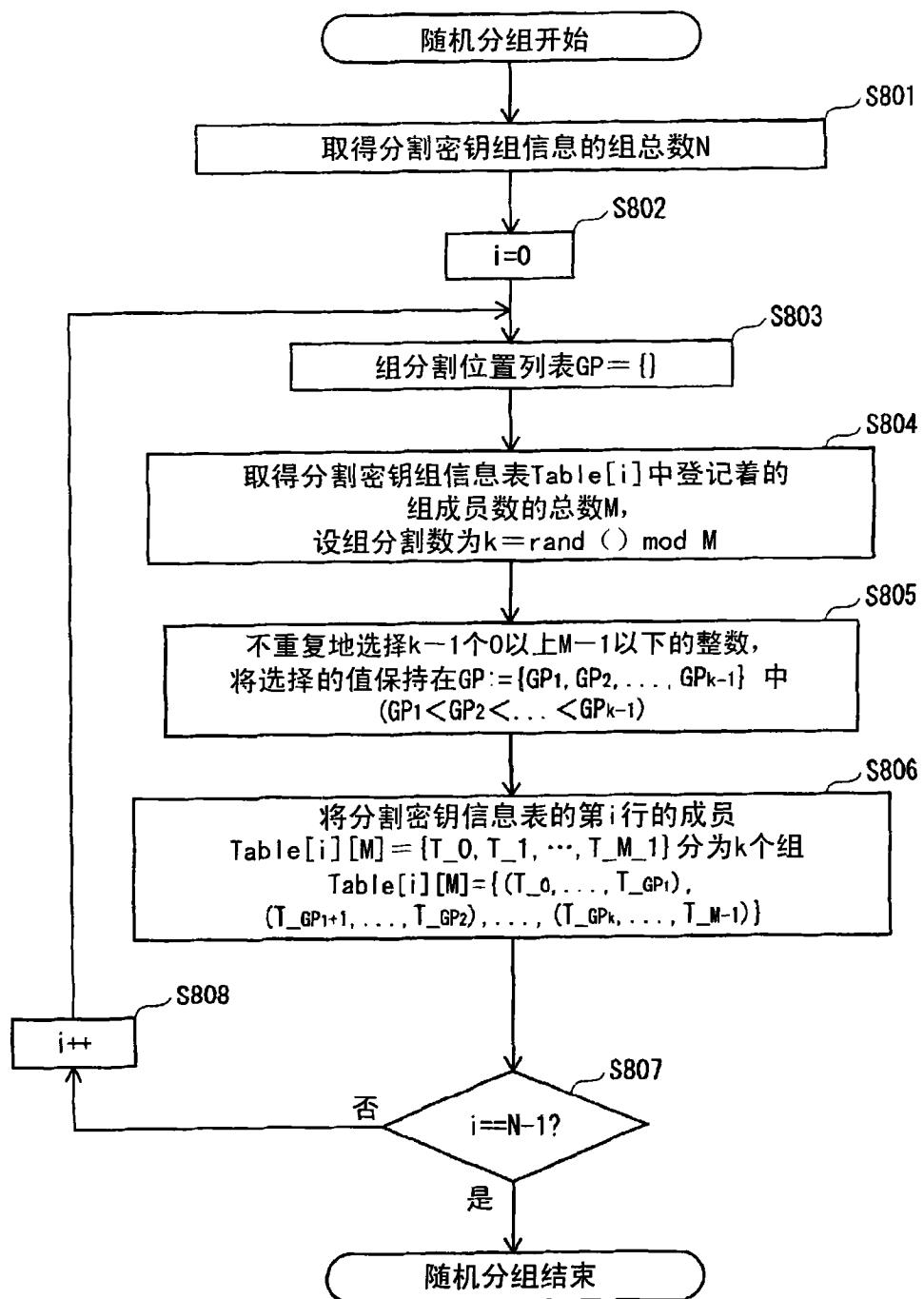


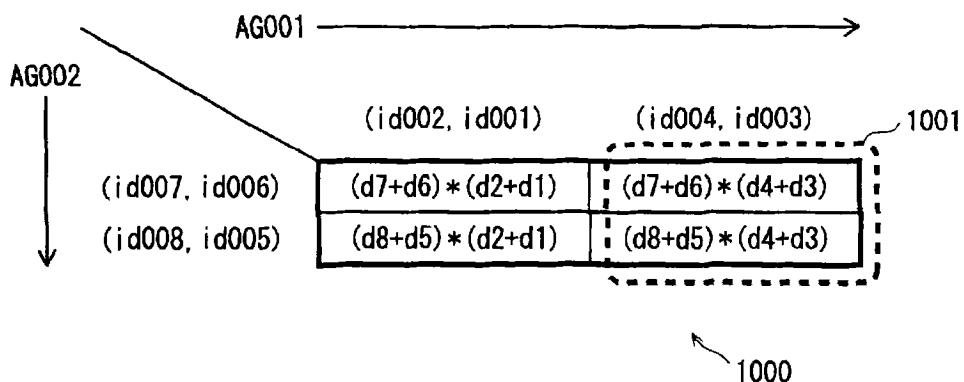
图 8

| | |
|-------|--------------------------------|
| MG001 | {AG002}, {AG001} |
| AG001 | {id007, id006}, {id008, id005} |
| AG002 | {id002, id001}, {id004, id003} |

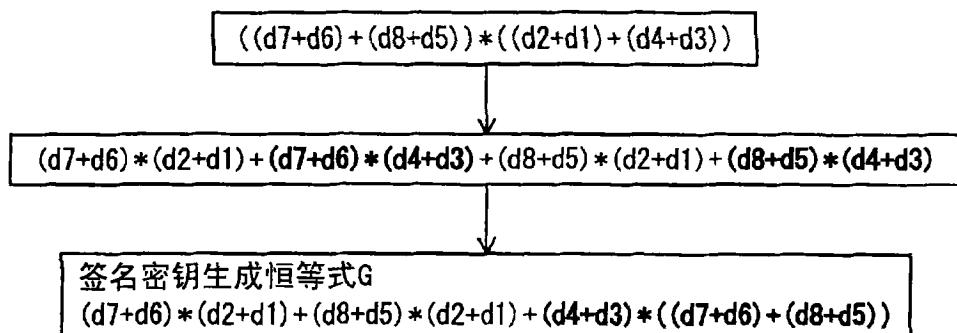
签名密钥生成恒等式G' : $((d7+d6)+(d8+d5)) * ((d2+d1)+(d4+d3))$

图 9

(a)



(b)



冬 10

600 结合分割密钥识别信息表

| 结合分割密钥标识符 | 结合分割密钥值 |
|-----------|--------------|
| ID001 | CD1 (=D7+D6) |
| ID002 | CD2 (=D2+D1) |
| ID003 | CD3 (=D8+D5) |
| ID004 | CD4 (=D2+D1) |
| ID005 | CD5 (=D4+D3) |
| ID006 | CD6 (=D7+D6) |
| ID007 | CD7 (=D8+D5) |

$$\begin{aligned}
 G'(&d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8) \\
 &= (d_7+d_6) * (d_2+d_1) + (d_8+d_5) * (d_2+d_1) + (d_4+d_3) * ((d_7+d_6) + (d_8+d_5)) \\
 &= (cd1*cd2) + (cd3*cd4) + cd5* (cd6+cd7) \\
 &= G(cd1, cd2, cd3, cd4, cd5, cd6, cd7)
 \end{aligned}$$

图 11

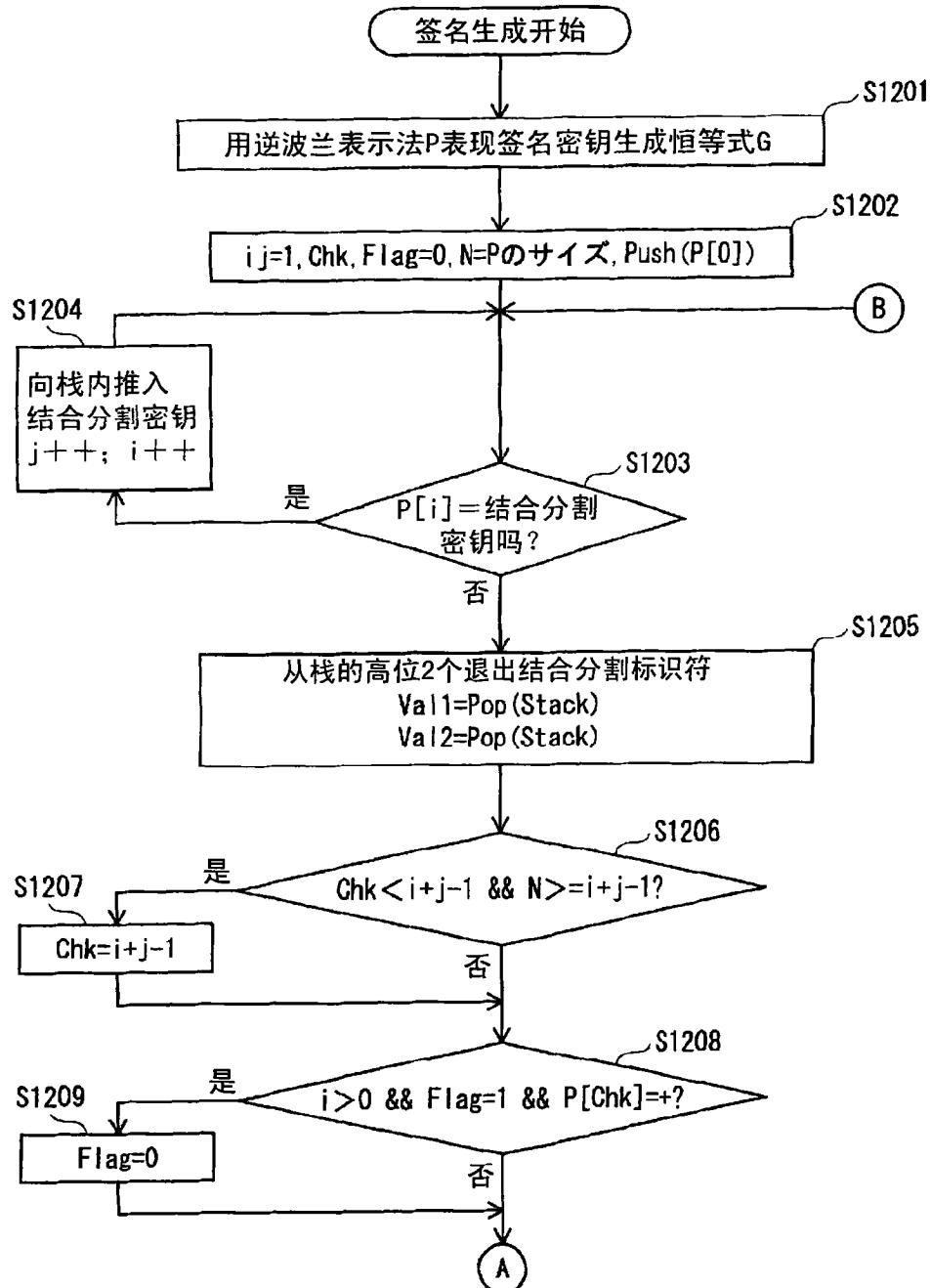


图 12

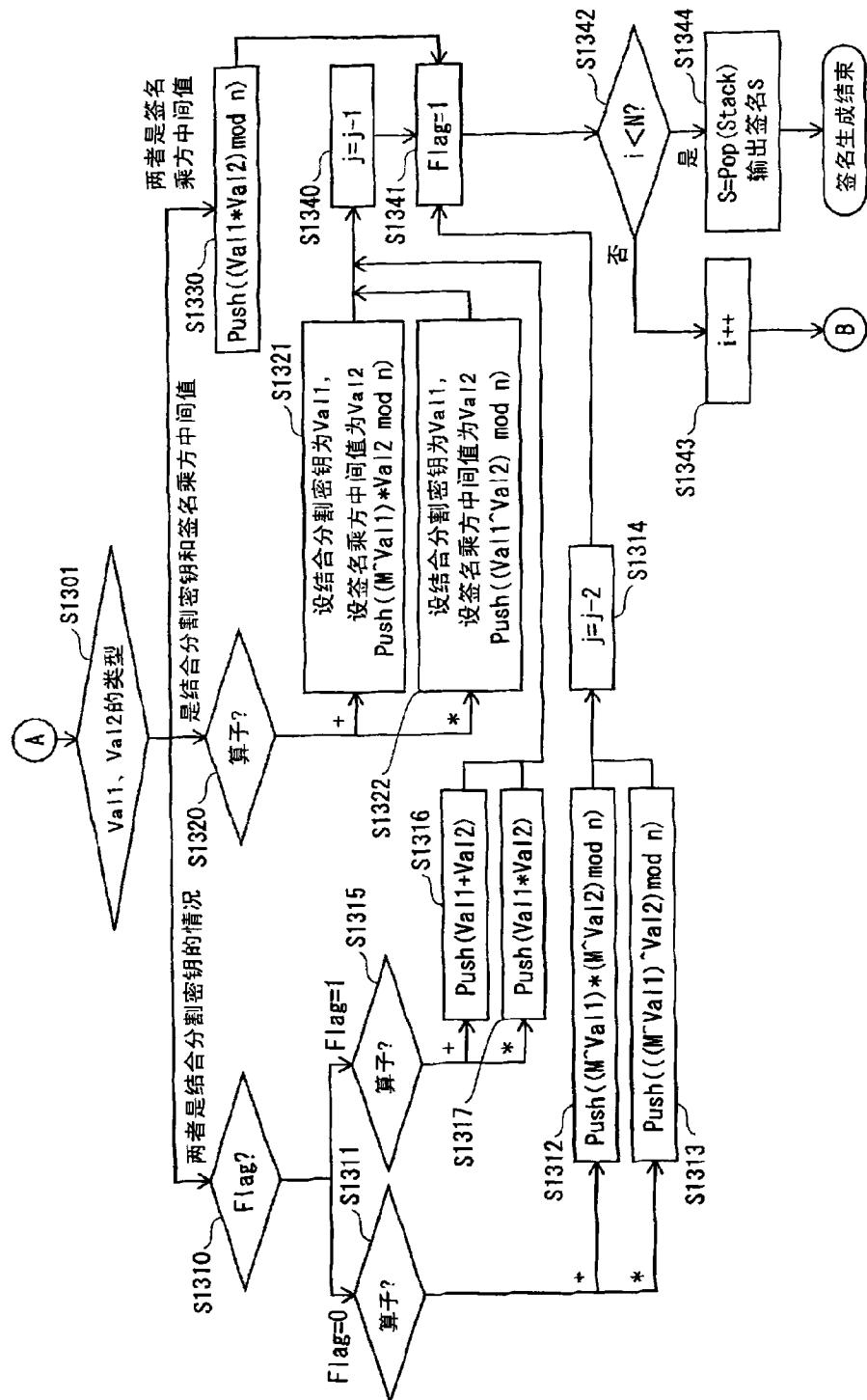
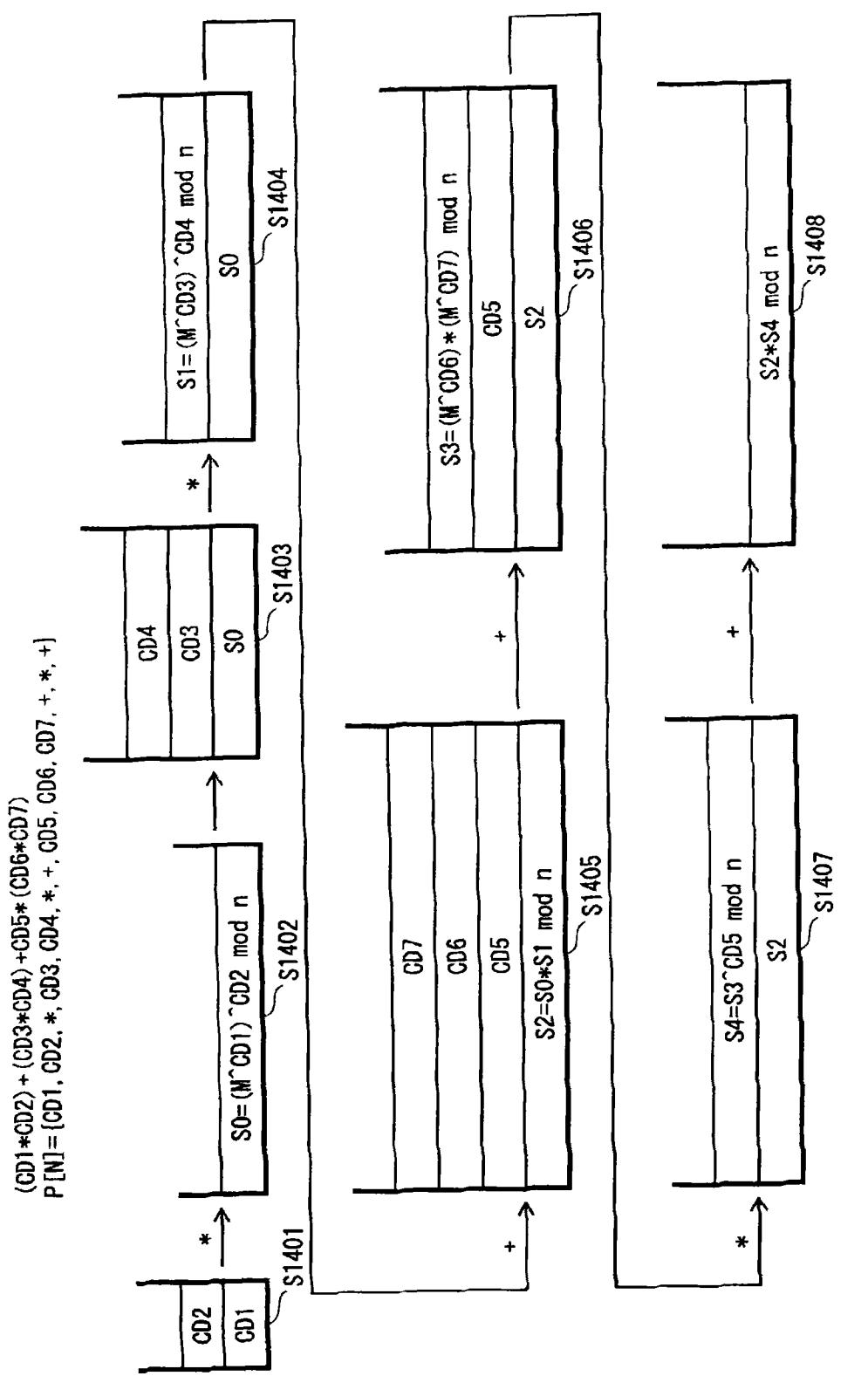


图 13



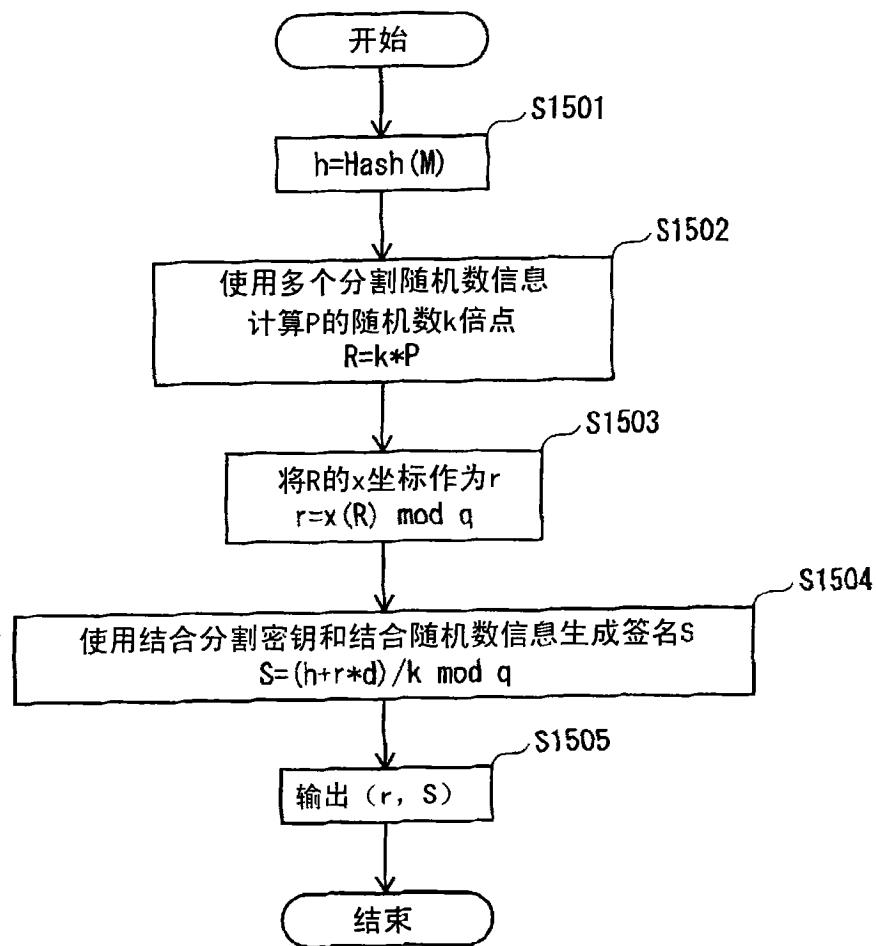


图 15

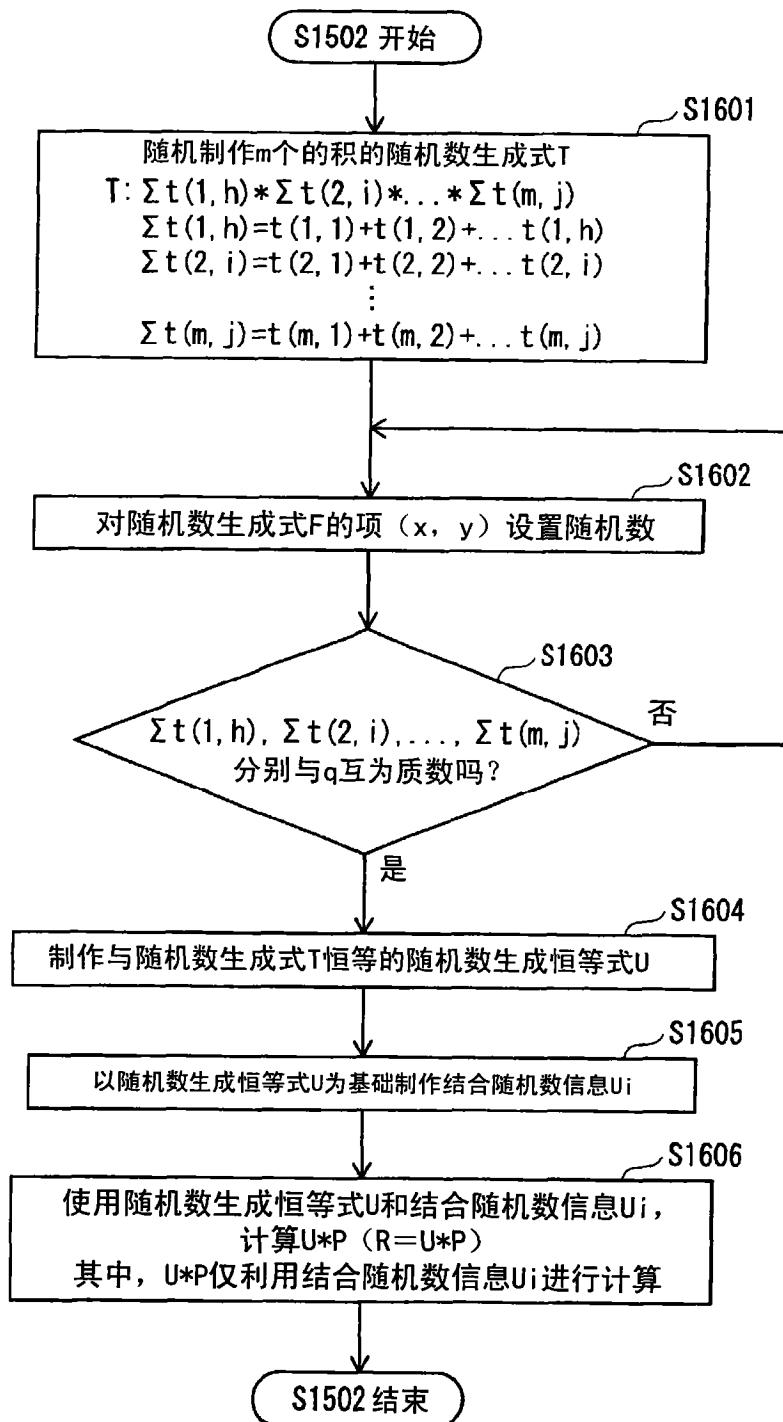


图 16

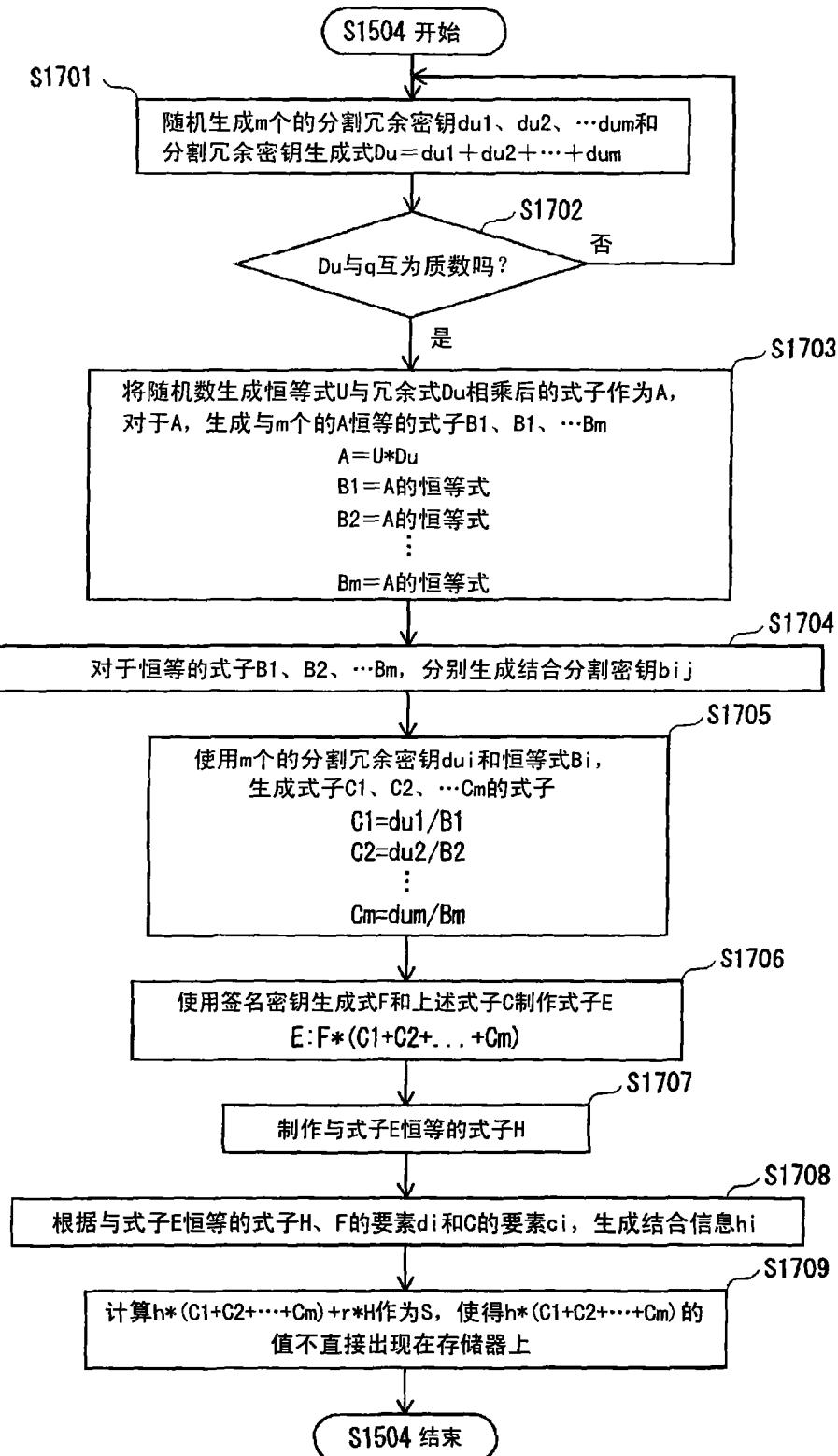


图 17

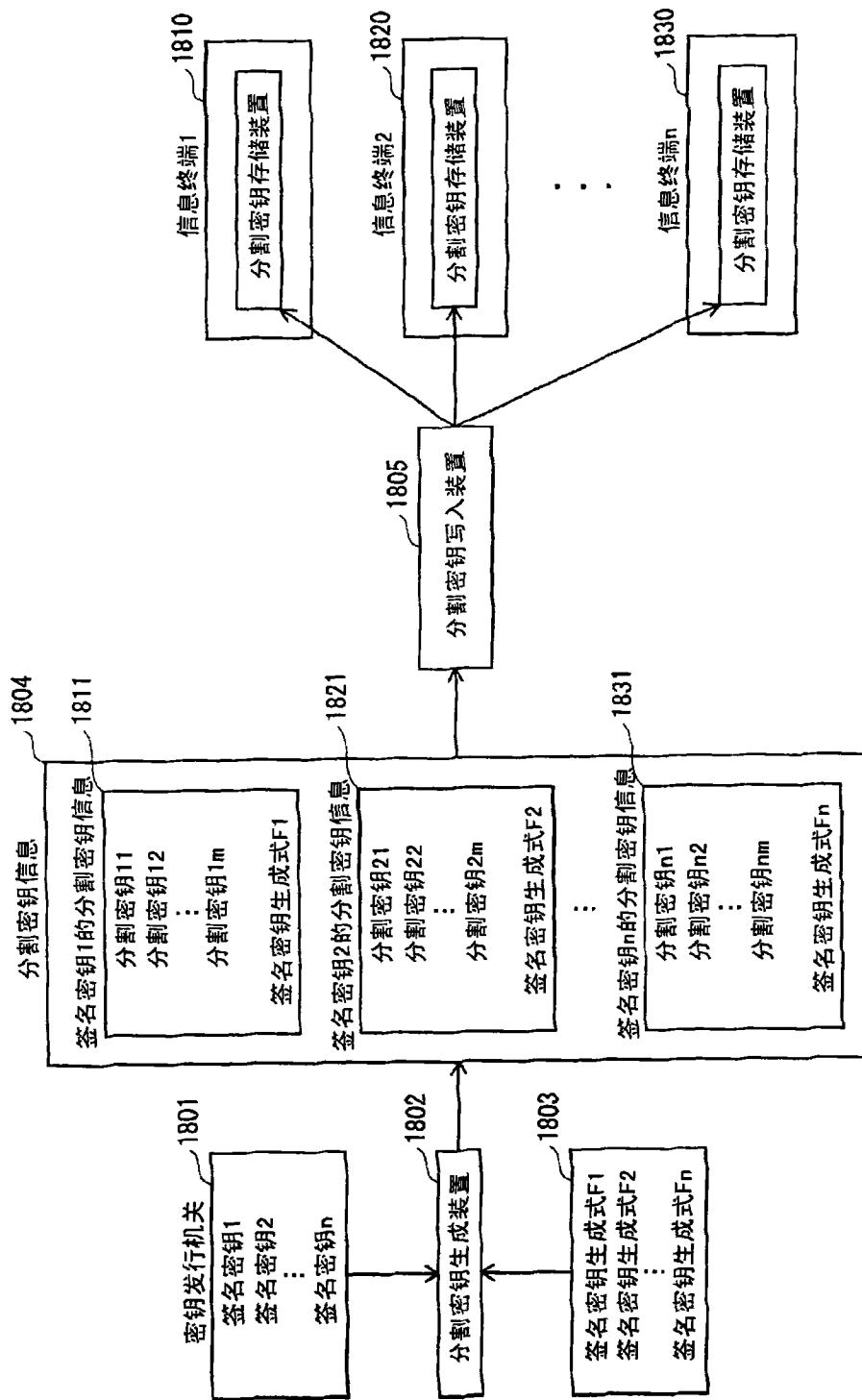
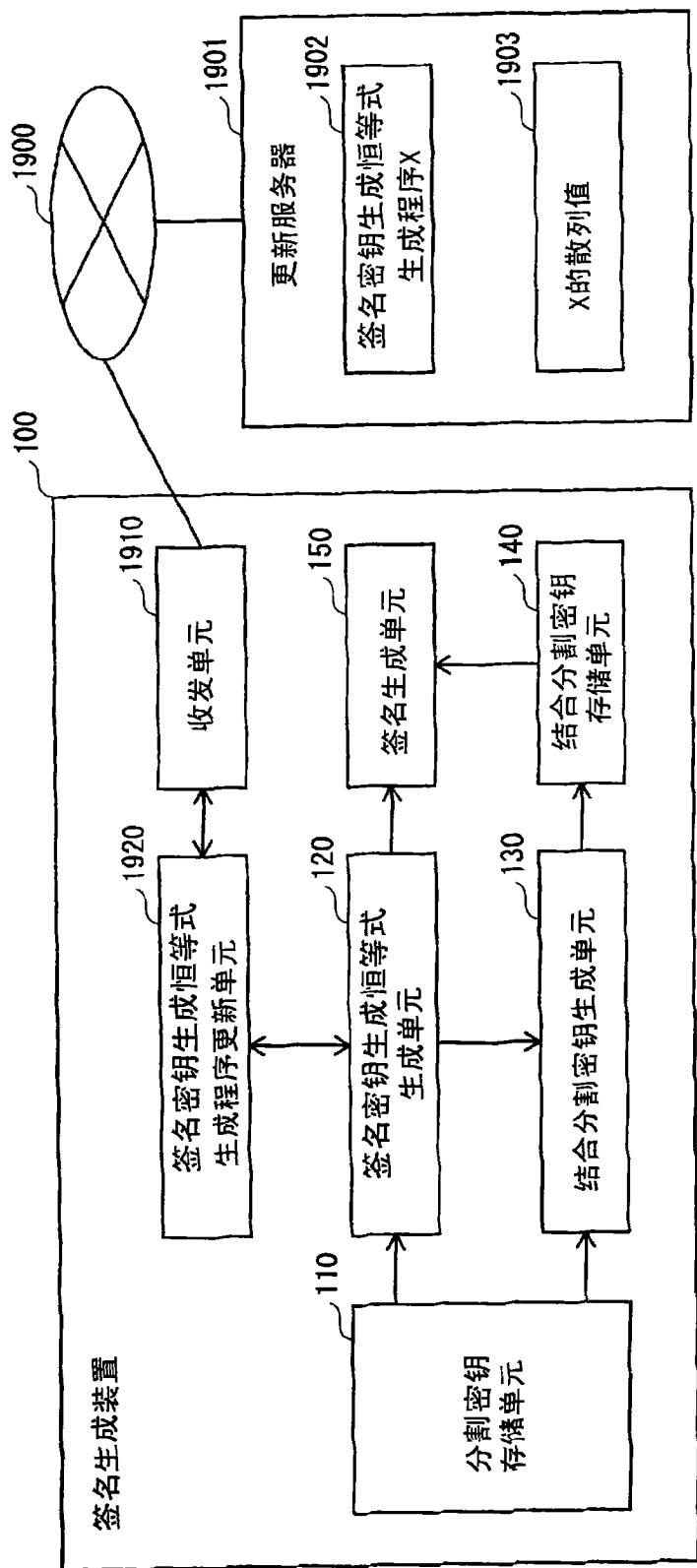


图 18



19

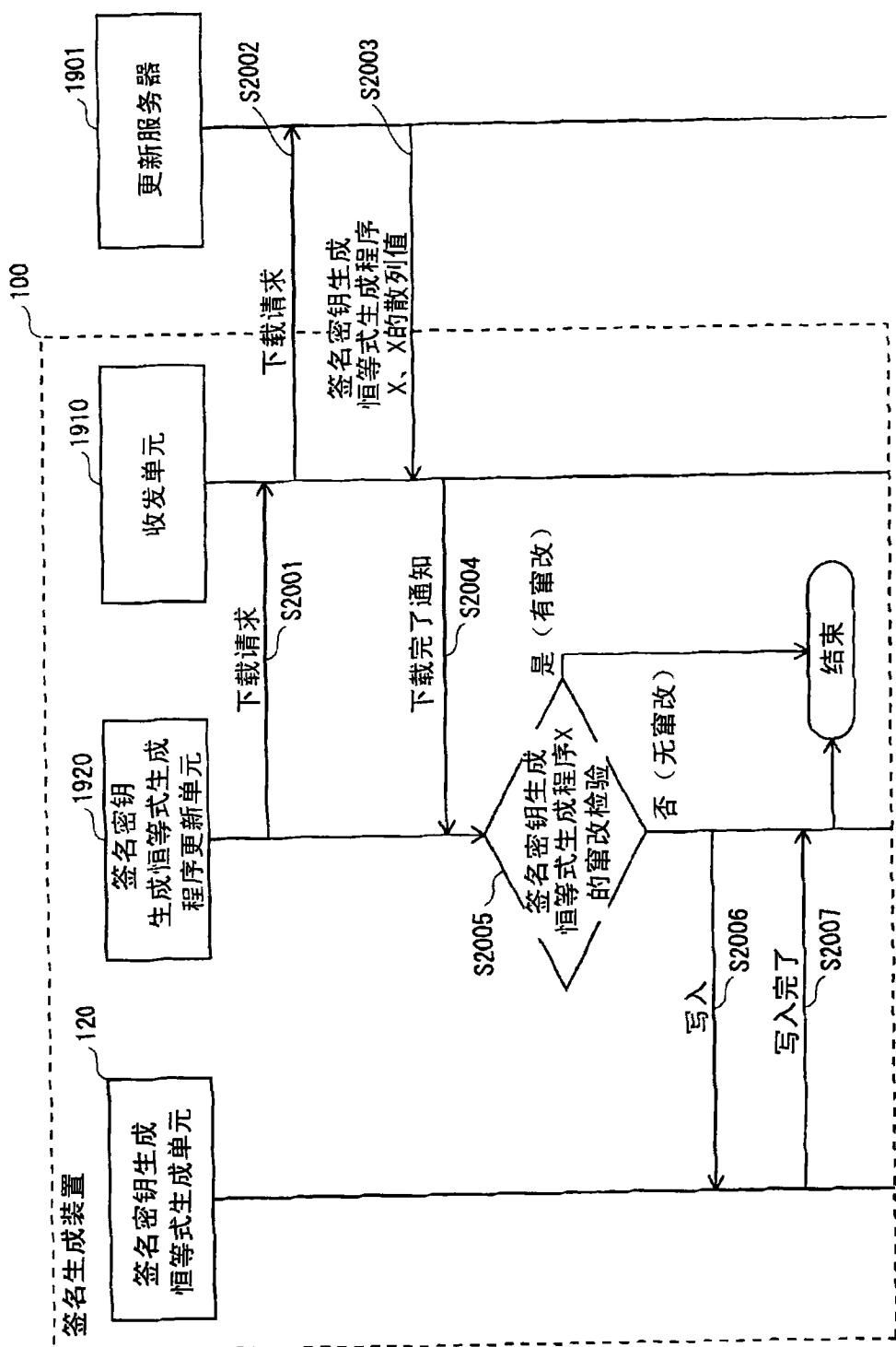


图 20