(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107078902 A
(43)申请公布日 2017.08.18

(21)申请号 201580053452.9

(22)申请日 2015.09.01

(30)优先权数据
62/062,306 2014.10.10 US
14/616,110 2015.02.06 US

(85)PCT国际申请进入国家阶段日
2017.03.31

(86)PCT国际申请的申请数据
PCT/US2015/047974 2015.09.01

(87)PCT国际申请的公布数据
WO2016/057134 EN 2016.04.14

(71)申请人 高通股份有限公司
地址 美国加利福尼亚州

(72)发明人 B·布伦利 R·阿万奇

(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 宋献涛

(51)Int.Cl.
H04L 9/06(2006.01)
H04L 9/00(2006.01)

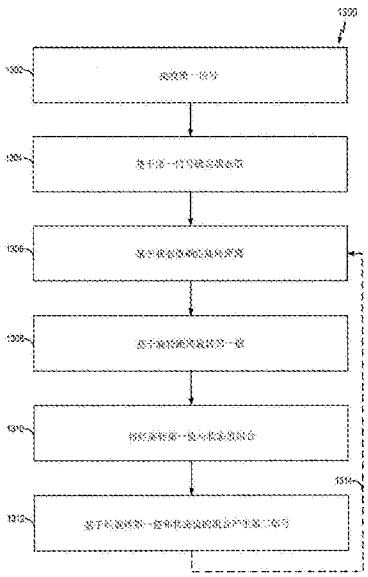
权利要求书3页 说明书23页 附图17页

(54)发明名称

基于旋转的密码术

(57)摘要

一种密码术采用替换框S框值的旋转来提供混淆和扩散两者。在一些方面，对于迭代密码术的每一迭代，扩展状态值的子集以计算用于旋转S框值的旋转距离，借此所述经旋转S框值与所述状态值组合，且针对下一迭代旋转新状态值。有利的是，所述密码术可使用常规指令在软件(或其它代码)中实施，而不需要大S框查找表。



CN 107078902 A

1.一种用于产生经加密信号的方法,其包括:

接收第一信号;

基于所述第一信号确定状态值;

基于所述状态值确定旋转距离;

基于所述旋转距离旋转第一值;

组合所述经旋转第一值和所述状态值;以及

基于所述经旋转第一值和所述状态值的所述组合产生第二信号。

2.根据权利要求1所述的方法,其中所述第一值为替换框值。

3.根据权利要求1所述的方法,其中所述旋转和所述组合对所述状态值施加非线性,且在所述状态值的后续实例内扩散所述组合。

4.根据权利要求1所述的方法,其中所述组合包括:

掩蔽所述经旋转第一值;以及

组合所述经掩蔽的经旋转第一值和所述状态值。

5.根据权利要求1所述的方法,其中所述组合包括:

通过组合所述经旋转第一值和所述状态值产生所述状态值的中间实例。

6.根据权利要求5所述的方法,其中:

所述第二信号的所述产生包括通过旋转所述状态值的所述中间实例产生所述状态值的后续实例;以及

所述状态值的所述中间实例旋转基于所述第一值的位数量的位数量。

7.根据权利要求5所述的方法,其中所述状态值的所述中间实例的所述产生包括将密码密钥与所述经旋转第一值和所述状态值的所述组合的结果进行组合。

8.根据权利要求1所述的方法,其中所述状态值的所述确定包括通过组合密码密钥和所述状态值的先前实例产生所述状态值的当前实例。

9.根据权利要求1所述的方法,其中所述旋转距离的所述确定进一步基于所述状态值的子集。

10.根据权利要求1所述的方法,其中所述旋转距离的所述确定包括:

从所述状态值选择位的子集;

确定对应于所述选定位子集的第一数目;

确定对应于所述第一值的位数量的第二数目;以及

将所述第一数目乘以所述第二数目。

11.根据权利要求1所述的方法,其中所述旋转距离的所述确定包括:

掩蔽所述状态值;以及

移位所述经掩蔽状态值。

12.根据权利要求1所述的方法,其中所述旋转距离的所述确定进一步包括:

基于所述经移位的经掩蔽状态值移位第二值;

掩蔽所述经移位第二值;以及

移位所述经掩蔽的经移位第二值。

13.根据权利要求1所述的方法,其中所述状态值的所述确定包括通过旋转所述状态值的先前实例产生所述状态值的当前实例。

14.根据权利要求1所述的方法，其中所述第二信号的所述产生包括将密码密钥与所述经旋转第一值和所述状态值的所述组合进行组合。

15.根据权利要求1所述的方法，其中所述第一值具有某一值使得所述经旋转第一值和所述状态值的所述组合为对射的。

16.根据权利要求1所述的方法，其中所述第一值为对射的。

17.根据权利要求1所述的方法，其中通过以下操作执行所述旋转距离的所述确定、所述第一值的所述旋转以及所述经旋转第一值和所述状态值的所述组合：

s^＝ROR(sbox,(s&M1)<<S1)&MASK;以及

s＝ROR(s,R1)，

其中^＝为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于所述状态值的实例，M1为第一掩码值，MASK为第二掩码值，S1为移位值，R1为旋转值，且sbox对应于所述第一值的实例。

18.根据权利要求1所述的方法，其中通过以下操作执行所述旋转距离的所述确定、所述第一值的所述旋转以及所述经旋转第一值和所述状态值的所述组合：

s＝ROL(s,R1)；以及

s^＝ROR(sbox,(s&M1)<<S1)&MASK，

其中^＝为XOR运算，ROL为向左旋转运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于所述状态值的实例，M1为第一掩码值，MASK为第二掩码值，R1为旋转值，S1为移位值，且sbox对应于所述第一值的实例。

19.根据权利要求1所述的方法，其中通过以下操作执行所述旋转距离的所述确定、所述第一值的所述旋转以及所述经旋转第一值和所述状态值的所述组合：

s^＝ROR(sbox,(s&M1)<<S1)；以及

s＝ROR(s,R1)，

其中^＝为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于所述状态值的实例，M1为掩码值，S1为移位值，R1为旋转值，且sbox对应于所述第一值的实例。

20.根据权利要求1所述的方法，其中通过以下操作执行所述旋转距离的所述确定、所述第一值的所述旋转以及所述经旋转第一值和所述状态值的所述组合：

s＝ROL(s,R1)；

shift＝(tinv>>((state&M1)<<S1))<<S2;以及

s^＝ROR(sbox,shift)，

其中^＝为XOR运算，ROL为向左旋转运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，>>为位向右移位运算，s对应于所述状态值的实例，M1为掩码值，S1为第一移位值，S2为第二移位值，R1为第一旋转值，shift为第二旋转值，tinv为倒转替换框值，且sbox对应于所述第一值的实例。

21.一种用于产生经加密信号的设备，其包括：

存储器电路；以及

处理电路，其耦合到所述存储器电路且经配置以：

从所述存储器电路接收第一信号；

基于所述第一信号确定状态值；

基于所述状态值确定旋转距离；

基于所述旋转距离旋转第一值；

组合所述经旋转第一值和所述状态值；以及

基于所述经旋转第一值和所述状态值的所述组合产生第二信号。

22.根据权利要求21所述的设备，其中所述第一值为替换框值。

23.根据权利要求21所述的设备，其中所述旋转和所述组合对所述状态值施加非线性，且在所述状态值的后续实例内扩散所述组合。

24.根据权利要求21所述的设备，其中所述旋转距离的所述确定进一步基于所述状态值的子集。

25.根据权利要求21所述的设备，其中，为组合所述经旋转第一值和所述状态值，所述处理电路进一步经配置以：

掩蔽所述经旋转第一值；以及

组合所述经掩蔽的经旋转第一值和所述状态值。

26.根据权利要求21所述的设备，其中：

为组合所述经旋转第一值和所述状态值，所述处理电路进一步经配置以基于所述经旋转第一值和所述状态值产生所述状态值的中间实例；

为产生所述第二信号，所述处理电路进一步经配置以旋转所述状态值的所述中间实例以产生所述状态值的后续实例；以及

所述状态值的所述中间实例旋转基于所述第一值的位数量的位数量。

27.一种用于产生经加密信号的设备，其包括：

用于接收第一信号的装置；

用于基于所述第一信号确定状态值的装置；

用于基于所述状态值确定旋转距离的装置；

用于基于所述旋转距离旋转第一值的装置；

用于组合所述经旋转第一值和所述状态值的装置；以及

用于基于所述经旋转第一值和所述状态值的所述组合产生第二信号的装置。

28.根据权利要求27所述的设备，其中所述旋转和所述组合对所述状态值施加非线性，且在所述状态值的后续实例内扩散所述组合。

29.一种存储计算机可执行代码的非暂时性计算机可读媒体，所述计算机可执行代码包含用以执行以下操作的代码：

接收第一信号；

基于所述第一信号确定状态值；

基于所述状态值确定旋转距离；

基于所述旋转距离旋转第一值；

组合所述经旋转第一值和所述状态值；以及

基于所述经旋转第一值和所述状态值的所述组合产生第二信号。

30.根据权利要求29所述的计算机可读媒体，其中所述旋转和所述组合对所述状态值施加非线性，且在所述状态值的后续实例内扩散所述组合。

基于旋转的密码术

[0001] 相关申请案的交叉参考

[0002] 本申请案主张2014年10月10日在美国专利与商标局申请的第62/062,306号临时申请案以及2015年2月6日在美国专利与商标局申请的第14/616,110号非临时申请案的优先权和权益,这两个申请案的完整内容以引用的方式并入本文中。

技术领域

[0003] 本发明的方面大体上涉及通信,且更确切地说(但非排他地)涉及基于旋转的密码术。

背景技术

[0004] 密码术为用于加密和/或解密的密码算法。在加密这一边,通常被称为明文的信息由密码术运算以产生通常被称为密文的经加密信息。在解密这一边,经加密信息由密码术运算以重新产生原始信息(原始明文)。存在不同类型的密码术,包含所谓的块密码术,例如数据加密标准(DES)密码术和高级加密标准(AES)密码术。

[0005] 良好块密码术围绕由Claude Shannon陈述的两个重要原理而设计:混淆和扩散。混淆意味着密文应以尽可能数学上复杂的方式取决于明文和密码密钥。扩散意味着明文中的小局部改变应影响尽可能多的密文。这些目标通常通过使用实现针对通常放置于称为替换层的宽阵列中的少量位(例如,所谓的替换框(S框))的混淆的简单构建块以及互混替换层的输出的线性层来实现。

[0006] S框通常将n位映射到n位。然而,在某些状况下(例如,在例如BLOWFISH或CAST等设计中),S框将n位映射到m位,其中m大于n。举例来说,可以采用8位到32位S框。在这些情况下,S框还执行扩散的一部分,从而促进密码术的设计。然而,与n位到n位S框相比,此些n位到m位S框不是对射函数,且其可使密码术的分析更困难。并且,这些S框相对较大。举例来说,在基于表的实施方案中,BLOWFISH和CAST S框占据常规8位S框的四倍的空间。

[0007] 此外,这些基于表的实施方案固有地容易发生高速缓冲存储器时序攻击。密码术的内部状态既定为机密的。然而,所述状态的部分用作到S框查找表的指数或偏移。与执行来自表的负载相关联的等待时间取决于处理器的数据高速缓冲存储器中的可用性。通过测量此等待时间,攻击者可采用此相依性来恢复内部密码术状态,并且最终恢复由密码术使用的密码密钥。

发明内容

[0008] 下文呈现本发明的一些方面的简化概述以提供对此些方面的基本理解。此概述并非本发明的所有所预期特征的广泛综述,且既不希望识别本发明的所有方面的关键或至关重要要素,也不希望划定本发明的任何或所有方面的范围。其唯一目的是以简化形式呈现本发明的一些方面的各种概念以作为稍后呈现的更详细描述的序言。

[0009] 本发明的各种方面实现一种密码术,其中值(例如,表、字符串或某一其它值)的可

变旋转被采用作为大S框以实现混淆和扩散两者。在一些方面，对于迭代密码术的每一迭代，扩展状态值的子集来计算此旋转S框的旋转距离，借此其旋转的S框或部分随后与状态值组合，且针对下一迭代旋转新状态值。有利的是，所述密码术可在软件(或其它代码)中使用常规指令实施，而不需要大S框查找表。所述密码术可用于明文的加密(encryption/enciphering)和密文的解密(decryption/deciphering)。

[0010] 在一个方面中，本发明提供一种用于产生经加密信号的方法：包含接收第一信号；基于第一信号确定状态值；基于状态值确定旋转距离；基于旋转距离旋转第一值；组合经旋转第一值和状态值；以及基于经旋转第一值和状态值的组合产生第二信号。

[0011] 本发明的另一方面提供一种经配置用于产生经加密信号的设备，其包含存储器电路和耦合到存储器电路的处理电路。所述处理电路经配置以：从存储器电路接收第一信号；基于第一信号确定状态值；基于状态值确定旋转距离；基于旋转距离旋转第一值；组合经旋转第一值和状态值；以及基于经旋转第一值和状态值的组合产生第二信号。

[0012] 本发明的另一方面提供一种经配置用于产生经加密信号的设备。所述设备包含：用于接收第一信号的装置；用于基于第一信号确定状态值的装置；用于基于状态值确定旋转距离的装置；用于基于旋转距离旋转第一值的装置；用于组合经旋转第一值和状态值的装置；以及用于基于经旋转第一值和状态值的组合产生第二信号的装置。

[0013] 本发明的另一方面提供一种存储计算机可执行代码的非暂时性计算机可读媒体，所述计算机可执行代码包含用以执行以下操作的代码：接收第一信号；基于第一信号确定状态值；基于状态值确定旋转距离；基于旋转距离旋转第一值；组合经旋转第一值和状态值；以及基于经旋转第一值和状态值的组合产生第二信号。

[0014] 除上文以外，根据本发明的额外方面，旋转和组合对状态值施加非线性，且在状态值的后续实例内扩散所述组合。根据本发明的额外方面，第一值为替换框值。根据本发明的额外方面，所述组合包含掩蔽经旋转第一值，以及组合所述经掩蔽的经旋转第一值和状态值。根据本发明的额外方面，所述组合包含通过组合经旋转第一值和状态值产生状态值的中间实例。根据本发明的额外方面，第二信号的产生包含通过旋转状态值的中间实例产生状态值的后续实例。根据本发明的额外方面，将状态值的中间实例旋转基于第一值的位数量的位数量。根据本发明的额外方面，状态值的中间实例的产生包含将密码密钥与经旋转第一值和状态值的所述组合的结果组合。根据本发明的额外方面，状态值的确定包含通过组合密码密钥和状态值的先前实例产生状态值的当前实例。根据本发明的额外方面，旋转距离的确定进一步基于状态值的子集。根据本发明的额外方面，旋转距离的确定包含：从状态值选择位的子集；确定对应于选定的位子集的第一数目；确定对应于第一值的位数量的第二数目；以及将第一数目乘以第二数目。根据本发明的额外方面，旋转距离的确定包含：掩蔽状态值；以及移位经掩蔽状态值。根据本发明的额外方面，旋转距离的确定进一步包含：基于经移位的经掩蔽状态值移位第二值；掩蔽经移位第二值；以及移位经掩蔽的经移位第二值。根据本发明的额外方面，状态值的确定包含通过旋转状态值的先前实例产生状态值的当前实例。根据本发明的额外方面，第二信号的产生包含将密码密钥与经旋转第一值和状态值的所述组合进行组合。根据本发明的额外方面，第一值具有某一值使得经旋转第一值和状态值的组合为对射的。根据本发明的额外方面，第一值为对射的。根据本发明的额外方面，第一信号包括明文，且第二信号包括明文的经加密表示。根据本发明的额外方面，

第一信号包括密文，且第二信号包括密文的经解密表示。根据本发明的额外方面，旋转距离的确定、第一值的旋转以及经旋转第一值和状态值的组合以迭代方式执行以基于所接收的第一信号产生第二信号。

[0015] 根据本发明的额外方面，旋转距离的确定、第一值的旋转以及经旋转第一值和状态值的组合可通过如下运算执行：s^＝ROR(sbox,(s&M1)<<S1)&MASK，以及s＝ROR(s,R1)，其中^＝为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于状态值的实例，M1为第一掩码值，MASK为第二掩码值，S1为移位值，R1为旋转值，且sbox对应于第一值的实例。

[0016] 根据本发明的额外方面，旋转距离的确定、第一值的旋转以及经旋转第一值和状态值的组合可通过如下运算执行：s＝ROL(s,R1)，以及s^＝ROR(sbox,(s&M1)<<S1)&MASK，其中^＝为XOR运算，ROL为向左旋转运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于状态值的实例，M1为第一掩码值，MASK为第二掩码值，R1为旋转值，S1为移位值，且sbox对应于第一值的实例。

[0017] 根据本发明的额外方面，旋转距离的确定、第一值的旋转以及经旋转第一值和状态值的组合可通过如下运算执行：s^＝ROR(sbox,(s&M1)<<S1)，以及s＝ROR(s,R1)，其中^＝为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，s对应于状态值的实例，M1为掩码值，S1为移位值，R1为旋转值，且sbox对应于第一值的实例。

[0018] 根据本发明的额外方面，旋转距离的确定、第一值的旋转以及经旋转第一值和状态值的组合可通过如下运算执行：s＝ROL(s,R1)，shift＝(tinv>>((state&M1)<<S1))<<S2，以及s^＝ROR(sbox,shift)，其中^＝为XOR运算，ROL为向左旋转运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，>>为位向右移位运算，s对应于状态值的实例，M1为掩码值，S1为第一移位值，S2为第二移位值，R1为第一旋转值，shift为第二旋转值，tinv为倒转替换框值，且sbox对应于第一值的实例。所属领域的一般技术人员在结合附图审阅本发明的特定实施方案的以下描述后将了解本发明的其它方面、特征和实施方案。虽然可相对于下文的特定实施方案和图论述本发明的特征，但本发明的所有实施方案可包含本文论述的有利特征中的一或多者。换句话说，虽然可将一或多个实施方案论述为具有特定有利特征，但也可以根据本文中所论述的本发明的各种实施方案来使用此类特征中的一或多者。类似地，虽然特定实施方案可在下文论述为装置、系统或方法实施方案，但应理解，此类实施方案可以各种装置、系统和方法实施。

附图说明

[0019] 图1说明本发明的一或多个方面可应用于的系统的实例。

[0020] 图2说明数据加密标准(DES)块密码术。

[0021] 图3说明用于DES块密码术的逐位线性变换的表。

[0022] 图4说明高级加密标准(AES)块密码术。

[0023] 图5说明根据本发明的一些方面的基于旋转的密码术的实例。

[0024] 图6说明根据本发明的一些方面的基于旋转的密码术的另一实例。

[0025] 图7说明根据本发明的一些方面对明文操作以产生密文的处理电路的实例。

[0026] 图8说明根据本发明的一些方面对密文操作以产生明文的处理电路的实例。

[0027] 图9说明根据本发明的一些方面对明文操作以产生密文的处理电路的另一实例。

[0028] 图10说明根据本发明的一些方面对密文操作以产生明文的处理电路的另一实例。

[0029] 图11说明根据本发明的一些方面的密码术电路的实例。

[0030] 图12说明根据本发明的一些方面用于支持密码安全的电子装置的实例硬件实施方案的框图。

[0031] 图13说明根据本发明的一些方面的密码术过程的实例。

[0032] 图14说明根据本发明的一些方面的密码术过程的另一实例。

[0033] 图15说明根据本发明的一些方面提供图14的密码术过程的更详细实例的密码术过程。

[0034] 图16说明根据本发明的一些方面的密码术过程的另一实例。

[0035] 图17说明根据本发明的一些方面提供图16的密码术过程的更详细实例的密码术过程。

具体实施方式

[0036] 下文结合附图阐述的具体实施方式意图作为对各种配置的描述，且并不意图表示可实践本文中所描述的概念的唯一配置。所述具体实施方式出于提供对各种概念的透彻理解的目的而包含具体细节。然而，所属领域的技术人员将显而易见的是，可在没有这些特定细节的情况下实践这些概念。在一些情况下，以框图形式展示众所周知的结构和组件以免混淆此类概念。

[0037] 图1为采用密码术的系统100的简化实例。系统100包含加密明文104借此产生密文106的加密组件102。举例来说，加密可用于确保明文104在存储或发射到另一组件(未图示)之前为受保护的。系统100还包含解密密文106借此恢复原始明文104的解密组件108。因此，原始明文104的使用可限于已经被给予解密密文106的能力的经授权用户。在实践中，给定系统可包含图1的组件中的一或多者。举例来说，一些系统可仅对信息进行加密，一些系统可仅对信息进行解密，且一些系统可对信息进行加密和解密两者。

[0038] 对于块密码术，通常经由使用非线性技术(通常被称为非线性层)实现混淆。举例来说，可以采用S框，借此给定输入信息块被由S框中的对应一者界定的另一信息块替换。

[0039] 相比而言，用于块密码术的扩散通常经由使用线性技术(通常被称为线性层)实现。举例来说，可以采用置换技术贯穿密码术空间散布信息。出于说明的目的，将参考DES块密码术和AES块密码术的实例描述这些概念。

[0040] 图2说明DES块密码术200的实例。输入信息(明文)202由第一线性层206(扩展置换)、非线性层208(S框)和第二线性层210(置换p)操作以产生输出信息(密文)204。非线性层208涉及用S框信息替换输入到S框中的信息。每一线性层206或210(置换)贯穿密码术空间有效地水平地到处移动位。线性层206和210可被认为是可由例如图3的表300等表表示的逐位线性变换。

[0041] 在实践中，此线性变换在硬件(例如，如图2中所展示)相对容易实施。然而，在软件中有效实施此线性变换较困难。举例来说，非定制处理器通常并不提供本机操作来直接执行这些操作。

[0042] 因此，针对软件实施方案，DES通常被实施为一系列表查找。到查找表中的每一查

找将映射到查找表中的给定条目m。在一个实例中，至多四个位在条目中的任一者中设定。在此情况下，用于相继DES轮次的每一表查找可涉及四个6位查找。跨越密码术空间的输入位的置换因此经由这些表查找实现。

[0043] 图4说明AES块密码术400的实例。输入信息(明文)402由非线性层406(S框)和线性层408(移位行和混合列)操作以产生输出信息(密文)404。在此情况下，线性层408不止涉及简单置换。实际上，位以数学方式混合。举例来说，线性层408可涉及可由向量等式表示的线性变换。

[0044] 类似于DES，用于AES的线性层在软件中相对难以有效实施。因此，针对软件实施方案，AES也通常被实施为一系列表查找。在一个实例中，用于相继AES轮次的每一表查找可涉及移位状态值、掩蔽字节，和使结果穿过查找表。

[0045] 基于旋转的密码术

[0046] 在一些方面中，本发明涉及基于旋转的密码术。在一些方面，此基于旋转的密码术可在软件(以及硬件)中有效实施，且可对抗高速缓冲存储器时序攻击(侧信道抵抗性)。在一些方面，此密码术可以极小代码占用面积实施。举例来说，在一些实施方案中，在Intel® x64处理器上，此密码术可以少于100字节和/或少于25个机器指令实施。基于旋转的密码术的两个实例在下文详细论述。

[0047] 第一实例

[0048] 在第一实例中，通过重复围绕以下运算所设计的单一轮次获得密码术：

[0049] state＝state XOR wide-s-box(state AND 0xF) 等式1

[0050] 后面或前面是轮次密钥混合：

[0051] state＝state XOR key_i 等式2

[0052] 和状态的旋转，举例来说：

[0053] state＝ROR(state,4) 等式3

[0054] 通过串接64位整数或阵列中的小4位替换框S的所有值获得宽s框(因为4位替换框具有16个条目，且16*4＝64)。举例来说，如果称为sbox的值(例如，字符串、表、阵列或某一其它值)被界定为：

[0055] sbox:＝S[15]|S[14]|S[13]|…|S[2]|S[1]|S[0]， 等式4

[0056] 其中S[15]的四个位是sbox的最高有效位，且S[0]的四个位是sbox的四个最低有效位，那么可从值sbox检索S[]的任何值，为：

[0057] S[v]＝(sbox>>(v*4))AND 0xF 等式5

[0058] 在一些方面，此可视为查找。因此，通过移除利用0xF的掩蔽且用旋转替代移位(ROR＝向右旋转)，可将全宽s框构造为值(例如，整数、阵列等)的旋转。因此，可界定将4位映射到64的“宽”、“旋转”S框，如下：

[0059] 等式6

[0060] wide-s-box(v)＝ROR(sbox,v*4) (因此S[v]＝wide-s-box(v)AND 0xF)

[0061] 然而，利用运算：

[0062] state＝state XOR wide-s-box(state AND 0xF) 等式7

[0063] 也变换状态的最低四个位。因此，经变换位无法在逆运算(例如，解密)中有效地使用以确定移位指数(乘以4)和旋转sbox 64位值。为解决此问题，可改为使用以下运算：

[0064] 等式8

[0065] state＝state XOR(wide-s-box(state AND 0xF)AND 0xFFFFFFFFFFFFFFF0)

[0066] 在此情况下，最低四个位未经修改。因此，这些位可用于计算宽s框的值，且因此恢复上一步骤。因此，为了解密，可简单地逆向执行加密步骤。

[0067] 在一些方面，以上密码术为如第三届国际快速软件加密研讨会(the ThirdInternational Workshop on Fast Software Encryption)的会议记录(121-144页，施普林格出版社(Springer-Verlag)，1996)中作者Bruce Schneier和John Kelsey的“不平衡费斯妥网络和块密码术设计(Unbalanced Feistel Networks and Block Cipher Design)”中界定的目标重型费斯妥网络。然而，在此实例密码术中，通过旋转S框计算替换。

[0068] 利用此密码术(例如，加密)的64位块密码术(被命名为“paul_encrypt”)的伪码实例运行如下。

[0069] 等式9

```
Mask = 0xFFFFFFFFFFFFFFF0ULL
void paul_encrypt(uint64_t * ciphertext, uint64_t plaintext, uint64_t key)
{
        int i;
        uint64_t state, boxvalue;
        int shift;

        state = plaintext;

        for (i = 0; i <= ROUNDS - 1; i++)
        {
           state = state XOR key_i          // 密钥混合

[0070]     index = state & 0xF              // 从状态的最低有效 4 位选择 S 框

           boxvalue = sbox >>> (index*4)

           state = state XOR (boxvalue AND Mask)        // 替换和扩散

           if (i <> ROUNDS-1)
              state =   state >>> 4         // 旋转所述状态
        }

        state = state XOR key_ROUNDS        // 输出白化(最后密钥混合)

        * ciphertext = state;
}
```

[0071] 代替于XOR运算，另一运算可用于执行密钥混合和/或经旋转S框的混合。举例来说，可采用整数加法、整数减法或某一其它混合(组合)运算。

[0072] 图5为根据本文中的教示的基于旋转的密码术500的概念表示。对于密码术500的给定轮次，当前状态504的子集502由θ函数506操作以提供扩展输出508。扩展输出508由γ函数510操作，借此γ函数510的输出被掩蔽524，且接着与当前状态504进行XOR运算512。所

得输出随后进行旋转514以提供用于密码术500的下一轮次的状态504。

[0073] 出于说明的目的，将在使用64位状态参数和4位S框的实施方案的上下文中更详细描述密码术500的运算。然而，应理解，本文中的教示不限于此实施方案，且其它状态和S框大小可在其它实施方案中使用。

[0074] 64位状态504由16个4位半字节组成。对于每一轮次，一个半字节输入到θ函数506。θ函数506扩展子集502的4位半字节以提供64位扩展输出508。此运算由图5中的θ描述516展示。此处，0x0的输入a映射到{0x0,0x1,0x2,...,0xE,0xF}，0x1的输入a映射到{0x1,0x2,0x3,...,0xF,0x0}，等等。扩展输出508可表示为(a+15)|(a+14)|...|(a+1)|a；其中|指示4位值的串接。

[0075] 在一些方面，θ函数506为提供扩散函数的线性层。因此，从紧凑型基于表的S框产生相对宽S框。

[0076] γ函数510执行如由图5中的γ描述518指示的字节替换运算。在一些方面，γ函数510为提供混淆函数的非线性层。如522处指示，S框经选择使得S(x)为对射的。

[0077] γ函数510使用64位表S执行扩展输出508的每一半字节的S框替换。图5说明此表520的实例，其中，以逐半字节的方式，针对给定输入x指示4位S框的值S(x)。应理解，本文中的教示不限于此特定表S，且其它表值和/或大小可在其它实施方案中使用。

[0078] 此表查找在图5中由S(x)数学上表示。因此，在γ描述518中，用于扩展输出508的最低有效半字节b_0的S框替换由S(b_0)表示，用于扩展输出508的下一最低有效半字节b_1的S框替换由S(b_1)表示，等等。因此，γ函数510的输出可表示为S(b_{15})|S(b_{14})|...|S(b_1)|S(b_0)；其中|指示4位值的串接。

[0079] 旋转514将其输入向右旋转一个半字节(即，四个位)。所述操作随后在下一轮次中重复直至已经操作状态504的所有半字节为止。在实践中，这些操作针对状态504的全部执行多次以增加最终由密码术500输出的密文的扩散和混淆特性。

[0080] 密码术500可在软件中使用例如旋转、XOR和位移位等标准指令实施。针对一轮次的指令集的实例在等式10中阐述。

[0081] 等式10

[0082] #define ROR64(x,b) (((x)>>(b))|((x)<<(64-(b))))

[0083] #define MASK 0xFFFFFFFFFFFFFFF0

[0084] const uint64_t sbox=0x3F41B87D026C59EA ULL

[0085] state^=key[i];

[0086] state^=ROR(sbox,(state&0xF)<<2)&MASK;

[0087] state=ROR(state,4);

[0088] 其中^=为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，state对应于状态504，sbox为表520，MASK对应于掩码524，且key为密码密钥。

[0089] 以上代码极小：(例如)，64位Intel®中央处理单元(CPU)上约25个指令(总计约100字节)。可见，不需要显式扩散步骤。此外，此类型的设计可利用经典技术进行密码分析，以提供严格的安全界限。与传统表查找相比，本文中所揭示的基于旋转的密码术不必使用机密密码术状态指数用于数据加载指令。因此，所述密码术的数据高速缓冲存储器占用面积不必取决于密码术状态。因此，此设计对高速缓冲存储器时序攻击具有天然抗扰性。

[0090] 第一实例-倒转

[0091] 图5的基于旋转的密码术500的倒转执行与密码术500类似的操作，只是以逆序。举例来说，此倒转密码术可用于解密由密码术500产生的密文。作为另一实例，倒转密码术可用于对明文进行加密，，且密码术500用于解密所得密文。

[0092] 利用此倒转密码术函数的64位块密码术(被命名为“paul_decrypt”)的伪码实例如下。

[0093] 等式11

```
Mask = 0xFFFFFFFFFFFFFFF0ULL
void paul_decrypt(uint64_t ciphertext, uint64_t * plaintext, uint64_t key)
{
        int i;
        uint64_t state, boxvalue;
        int shift;

        state = ciphertext;
[0094]
        state = state XOR key_ROUNDS          // 初始密钥混合


        for (i = ROUNDS-1; I >= 0; i--)
        {
          if (i <> ROUNDS - 1)
              state =   state <<< 4           // 旋转
          index = state & 0xF           // 从状态的最低有效 4 位选择 S 框

          boxvalue = sbox >>> (index*4)

          state = state XOR (boxvalue AND Mask)
[0095]
          state = state XOR key_i           //密钥混合

        }

        * plaintext = state;
}
```

[0096] 如上文所提及，例如整数加法或减法等另一运算可用于执行密钥混合和/或经旋转S框的混合，代替XOR运算。在此情况下，在倒转密码术中使用函数的倒转。举例来说，如果在加密函数中使用加法，那么在解密函数的对应位置使用减法以确保恰当倒转。作为另一实例，如果在加密函数中使用减法，那么在解密函数的对应位置使用加法以确保恰当倒转。

[0097] 倒转密码术可在软件中使用例如旋转、XOR和位移位等标准指令实施。针对一轮次的指令集的实例在等式12中阐述。

[0098] 等式12

[0099] #define ROL64(x,b) (((x)<<(b))|((x)>>(64-(b))))

[0100] #define MASK 0xFFFFFFFFFFFFFFF0

[0101] const uint64_t sbox＝0x3F41B87D026C59EA ULL

[0102] state＝ROL(state,4)；

[0103] state^＝ROR(sbox,(state&0xF)<<2)&MASK；

[0104] state^＝key[i]；

[0105] 其中^＝为XOR运算,ROL为向左旋转运算,ROR为向右旋转运算,&为AND运算,<<为位向左移位运算,state对应于状态504,sbox为表520,MASK对应于掩码524,且key为密码密钥。

[0106] 第二实例

[0107] 在使用值(例如,字符串、表等)的相同类型的旋转以产生宽S框的密码术的第二实例中,当将值混合到状态中时并不采用掩蔽。此密码术不是费斯妥网络。在此实例中,密码术的倒转更棘手,因为最低有效半字节经修改(如上文所论述)：

[0108] new-state[3..0]＝state[3..0]XOR S([state[3..0])等式13

[0109] 假定由T[v]＝v XOR S[v]界定的S框T为可逆的,通过使新状态[3..0]通过T的倒转,可恢复原始状态。因此,原始经移位大Sbox可恢复并进行XOR运算到完整状态。因此,在此实例中,S并不需要为可逆的。然而,新S框T为可逆的。

[0110] 利用此加密函数的64位块密码术(被命名为“paul2_encrypt”)的伪码实例如下。

[0111] 等式14

```
Mask = 0xFFFFFFFFFFFFFFF0ULL
MASQ = 0xFEDCBA9876543210ULL
sbox = (MASQ XOR tbox)
tbox= 0xc19d02e574386bfaULL
tinv = 0x19cf20d473865aebULL

void paul_encrypt(uint64_t * ciphertext, uint64_t plaintext, uint64_t key)
{
        int i;
        uint64_t state, boxvalue;
        int shift;

        state = plaintext;

        for (i = 0; i <= ROUNDS - 1; i++)
        {
[0112]      state = state XOR key_i          // 密钥混合

            index = state & 0xF          // 从状态的最低有效 4 位选择 S 框

            boxvalue = sbox >>> (index*4)

            state = state XOR boxvalue          // 替换和扩散

            if (i <> ROUNDS - 1)
                state =   state >>> 4          // 旋转
        }

        state = state XOR key_i          // 输出白化

        * ciphertext = state;
}
```

[0113] 图6为根据本文中的教示的基于旋转的密码术600的概念表示。对于密码术600的给定轮次，当前状态604的子集602由θ函数606操作以提供扩展输出608。扩展输出608由γ函数610操作，借此γ函数610的输出与当前状态604进行XOR运算612。所得输出随后进行旋转614以提供用于密码术600的下一轮次的状态604。

[0114] 出于说明的目的，将在使用64位状态参数和4位S框的实施方案的上下文中更详细描述密码术600的运算。然而，应理解，本文中的教示不限于此实施方案，且其它状态和S框大小可在其它实施方案中使用。

[0115] 64位状态604由16个4位半字节组成。针对每一轮次，一个半字节输入到θ函数606。θ函数606扩展子集602的4位半字节以提供64位扩展输出608。此运算由图6中的θ描述616展示。此处，0x0的输入a映射到{0x0,0x1,0x2,...,0xE,0xF}，0x1的输入a映射到{0x1,0x2,0x3,...,0xF,0x0}，等等。扩展输出608可表示为(a+15)|(a+14)|...|(a+1)|a；其中|指示4位值的串接。

[0116] 在一些方面，θ函数606为提供扩散函数的线性层。因此，从紧凑型基于表的S框产

生相对宽S框。

[0117] γ函数610执行如由图6中的γ描述618指示的字节替换运算。在一些方面，γ函数610为提供混淆函数的非线性层。如622处指示，S框经选择使得x→x XOR S(x)为对射的。

[0118] γ函数610使用64位表S执行扩展输出608的每一半字节的S框替换。图6说明此表620的实例，其中，以逐半字节的方式，针对给定输入x指示4位S框的值S(x)。应理解，本文中的教示不限于此特定表S，且其它表值和/或大小可在其它实施方案中使用。

[0119] 此表查找在图6中数学上由S(x)表示。因此，在γ描述618中，扩展输出608的最低有效半字节b_0的S框替换由S(b_0)表示，扩展输出608的下一最低有效半字节b_1的S框替换由S(b_1)表示，等等。因此，γ函数610的输出可表示为S(b_{15})|S(b_{14})|...|S(b_1)|S(b_0)；其中|指示4位值的串接。

[0120] 旋转614将其输入向右旋转一个半字节(即，四个位)。所述操作随后在下一轮次中重复直至已经操作状态604的所有半字节为止。在实践中，这些操作针对状态604的全部执行多次以增加最终由密码术600输出的密文的扩散和混淆特性。

[0121] 密码术600可在软件中使用例如旋转、XOR和位移位等标准指令实施。针对一轮次的指令集的实例在等式15中阐述。

[0122] 等式15

[0123] #define ROR64(x,b) (((x)>>(b))|((x)<<(64-(b))))

[0124] const uint64_t sbox＝0x3F41B87D026C59EA ULL

[0125] state^＝key[i];

[0126] state^＝ROR(sbox,(state&0xF)<<2);

[0127] state＝ROR(state,4);

[0128] 其中^＝为XOR运算，ROR为向右旋转运算，&为AND运算，<<为位向左移位运算，state对应于状态604，sbox为表620，且key为密码密钥。

[0129] 再次，以上代码极小：例如，64位Intel®CPU上25个指令(总计100字节)。在一些方面，以上密码术可具有与第一实例密码术相比较快的加密和较好的扩散。此较好扩散可使得能够使用较少轮次。然而，如下文所示，与第一实例密码术中的解密中的轮次函数相比，所述解密中的轮次函数更复杂，且因此通常更慢。

[0130] 第二实例密码术不是费斯妥密码术(用于确定S框值的“分支”也被替换)。此外，所述密码术不是替换-置换网络(SPN)。

[0131] 在以上“paul2_encrypt”密码术中，映射x->x XOR S[X]为对射的，使得可恢复恰当指数。然而，并不严格地必需原始S框为对射(在此情况下，等效于内射)的性质。因此，在一些方面，非内射S框可施加到非费斯妥构造。

[0132] 对于以上“paul2_encrypt”密码术，解密使用提供x+S(x)的第二表。或者，解密和加密的角色可调换。举例来说，此调换可在与其中发生加密的装置相比计算上更有限的装置上发生解密的情况下实施。

[0133] 第二实例-倒转

[0134] 图6的基于旋转的密码术600的倒转执行与密码术600类似的操作，只是以逆序，同时使用如上文所描述的S框T。此倒转密码术可用于例如解密由密码术600产生的密文。作为另一实例，倒转密码术可用于对明文进行加密，且密码术600用于解密所得密文。

[0135] 利用此倒转密码术函数的64位块密码术(被命名为“paul2_decrypt”)的伪码实例如下。

[0136] 等式16

[0137]
```
void paul2_decrypt(uint64_t ciphertext, uint64_t * plaintext, uint64_t key)
{
        int i;
        uint64_t state, boxvalue;
        int shift;

        state = ciphertext;

        state = state XOR key_i

        for (i = ROUNDS - 1; i >= 0; i--)
        {
           if (i <> ROUNDS-1)
              state = state <<< 4                    // 旋转
```

[0138]
```
        index = state & 0xF

        shift = ((tinv >> (index << 2)) & 0xF) << 2

        boxvalue = sbox >>> (shift)

        state = state XOR (boxvalue )

        state = state XOR key_i                     // 密钥混合

        }

        * plaintext = state;
}
```

[0139] 倒转密码术可在软件中使用例如旋转、XOR和位移位等标准指令实施。针对一轮次的指令集的实例在等式1中阐述。

[0140] 等式17

[0141]
```
#define ROL64(x,b) (((x)<<(b))|((x)>>(64-(b))))
```

[0142]
```
const uint64_t tinv=0x19cf20d473865aeb ULL
```

[0143]
```
const uint64_t sbox=0x3F41B87D026C59EA ULL
```

[0144]
```
state=ROL(state,4);
```

[0145]
```
shift=(tinv>>((state&0xF)<<2))<<2;
```

[0146]
```
state^=ROR(sbox,shift);
```

[0147]
```
state^=key[i];
```

[0148] 其中^=为XOR运算,ROL为向左旋转运算,ROR为向右旋转运算,&为AND运算,>>为位向右移位运算,<<为位向左移位运算,state对应于状态604,sbox为表620,且key为密码密钥。

[0149] 额外方面

[0150] 大体来说，本文中的教示适用于“宽”S框，而非仅仅本文中所描述的特定实例。举例来说，掩蔽可用于截断到较小大小。

[0151] 如果起点为n位S框S，那么S框可通过以下映射而“扩展”到m字节宽S框：x->S[x+m-1]|...|S[x+1]|S[x]，其中|为串接。

[0152] 如果m并不太大(例如，小于2^n)且S具有良好算术差分性质，那么密码术可相对安全。此为与上文基本上相同的构造，但发生了“截断”。

[0153] 本文中的教示不限于具有4位输入的S框。举例来说，映射可为从4位到n位。作为一特定实例，其中4<n<64，此映射可通过如下映射实现：

[0154] V→ROT(sbox,4*v)，截断到n位 等式18

[0155] 对于4到16位的截断，可使用以下映射：

[0156] V→ROT(sbox,4*v)&0xFFFF，用作16位字 等式19

[0157] 其它映射将用于n的其它值的截断。

[0158] 举例来说，所揭示基于旋转的密码术的应用包含基于8位S框的密码术。对于具有类似于上文的“paul_encrypt”密码术的机制的128位块密码术，可使用16个连续值。为避免读取表时“包围”，可使用271字节表代替256字节表。

[0159] 8位S框可存储为256字节的阵列T[0..255]。因此，可以Bruce Schneier的Blowfish或Carlisle Adams和Stafford Tavares的CAST(CAST 128或CAST 256)的方式产生8到32位S框，为：

[0160] V→S[v+3]|S[v+2]|S[v+1]|S[v] 等式20

[0161] 此处，指数为(此相当于采取256*8位“sbox”字符串并使其旋转，且接着仅保留几个位)任一经解译模数256，或表自身为实际259字节长，且最后三个字节复制最前三个，使得指数不会是包覆模数256。

[0162] 对于硬件实施方案，相同电路可并行或串行(例如，通过每次递增指数v)而再使用4次。

[0163] 值得注意的是，Blowfish或CAST S框产生为完整表，而非旋转值。因此，在一些实施方案中，Blowfish或CAST S框可比根据本文中的教示采用的S框大很多。

[0164] 与DES的比较

[0165] 如上文所论述，对于4位S框，单一64位寄存器可用于表示16个4位条目。通过寄存器的不同旋转获得不同条目。因此，代替于使用表查找(例如，如DES中)，通过更加高效(从软件角度来看)的旋转操作实现替换。

[0166] 继续使用4位S框的实例。这些S框具有16个条目，且因此含有64位的信息。映射x->S[x]为表查找。通过使用含有全部64位的寄存器Y，可如上文所论述使用以下等式：

[0167] 等式21

[0168] S[x]＝(Y>>>(4*x))&0xF，

[0169] 其中“>>>”表示向右循环旋转。

[0170] DES S框S0可使用4个对射4位S框构建。第0S框由以下等式表示：

[0171] 等式22

[0172] x 0 1 2 3 4 5 6 7 8 9 A B C D E F

[0173] S00(x) E 4 D 1 2 F B 8 3 A 6 C 5 9 0 7

[0174] DES涉及存储器驻留表查找，例如：

[0175] 等式23

[0176] static const unsigned char S00[16]＝{

[0177] 0xE,0x4,0xD,0x1,0x2,0xF,0xB,0x8,0x3,0xA,0x6,0xC,0x5,0x9,0x0,0x7

[0178] }

[0179] DES随后执行替换，例如：y＝S00[x&0xF]。相比而言，根据本文中的教示，实际上使用常数的移位：

[0180] 等式24

[0181] y＝(0x7095C6A38BF21D4E>>((x&0xF)<<2))&0xF

[0182] 此处，密码术围绕以下映射设计：S'[x]＝Y>>>(4*x)。替代实施方案可使用124位寄存器Z，其中Z的64个最低有效位是Y，且Y的60个最低有效位是Z的上部60位中的复本。S'可被实施为Z>>(4x)的64个最低有效位，其中“>>”为简单右移。

[0183] 有利的是，以上操作可相对快速地在软件中执行。大多数商用微处理器表征旋转指令，其中从寄存器取得旋转距离，即，非恒定旋转而是可变旋转。在硬件中，以上操作为简单位选择。

[0184] 用于加密和解密操作的实例处理电路

[0185] 鉴于上文，在一些方面，本发明涉及执行等式10、等式12、等式15、等式17的运算或本文教示的任何其它运算的处理电路。

[0186] 图7说明使用类似于等式10的等式706对明文704操作以产生密文708的处理电路702的实例。因此，图7对应于加密(encipher/encryption)操作。

[0187] 图8说明使用类似于等式12的等式806对密文804操作以产生明文808的处理电路802的实例。因此，图8对应于解密(decipher/decryption)操作。

[0188] 图9说明使用类似于等式15的等式906对明文904操作以产生密文908的处理电路902的实例。因此，图9对应于加密(encipher/encryption)操作。

[0189] 图10说明使用类似于等式17的等式1006对密文1004操作以产生明文1008的处理电路1002的实例。因此，图10对应于解密(decipher/decryption)操作。

[0190] 图11说明可由处理电路702、802、902或1002实施的电路1100的非限制性实例。过程执行电路1102控制存储于存储器装置1104中的指令的执行。因此，取决于当前正执行的指令，可调用各种电路对从存储器装置1104检索的数据进行操作。如所指示，此电路可包含移位电路1106(例如，执行简单移位运算或旋转运算)、掩码电路1108(例如，执行AND运算)，和组合电路1110(例如，执行XOR运算)。

[0191] 实例电子装置

[0192] 图12为根据本发明的一或多个方面的经配置以支持密码术操作的设备1200的说明。设备1200包含通信接口(例如，至少一个收发器)1202、存储媒体1204、用户接口1206、存储器装置1208和处理电路1210。

[0193] 这些组件可经由信令总线或其它合适的组件(通常由图12中的连接线表示)彼此耦合和/或放置成彼此电连通。信令总线可包含任何数目的互连总线和桥，这取决于处理电路1210的特定应用和总体设计约束。信令总线将各种电路链接在一起使得通信接口1202、

存储媒体1204、用户接口1206和存储器装置1208中的每一者耦合到处理电路1210和/或与处理电路1210电连通。信令总线还可链接所属领域中众所周知的各种其它电路(未图示),例如时序源、外围装置、电压调节器和功率管理电路,且因此,将不进一步描述。

[0194] 通信接口1202可适于促进设备1200的无线通信。举例来说,通信接口1202可包含适于促进信息相对于网络中的一或多个通信装置的双向通信的电路和/或代码(例如,指令)。通信接口1202可耦合到一或多个天线1212,用于无线通信系统内的无线通信。通信接口1202可被配置成具有一或多个独立接收器和/或发射器,以及一或多个收发器。在所说明的实例中,通信接口1202包含发射器1214和接收器1216。

[0195] 存储器装置1208可表示一或多个存储器装置。如所指示,存储器装置1208可维持密码术信息1218连同供设备1200使用的其它信息。在一些实施方案中,存储器装置1208和存储媒体1204被实施为共同存储器组件。存储器装置1208还可用于存储由处理电路1210或设备1200的某一其它组件操纵的数据。

[0196] 存储媒体1204可表示一或多个计算机可读、机器可读和/或处理器可读装置,用于存储代码,例如处理器可执行代码或指令(例如,软件、固件)、电子数据、数据库或其它数字信息。存储媒体1204还可用于存储在实行代码时由处理电路1210操纵的数据。存储媒体1204可为可由通用或专用处理器存取的任何可用媒体,包含便携式或固定储存装置、光学存储装置和能够存储、含有或载运代码的各种其它媒体。

[0197] 借助于实例而非限制,存储媒体1204可包含磁性存储装置(例如,硬盘、软盘、磁条)、光盘(例如,压缩光盘(CD)或数字多功能光盘(DVD))、智能卡、快闪存储器装置(例如,卡、棒或键驱动)、随机存取存储器(RAM)、只读存储器(ROM)、可编程ROM(PROM)、可擦除PROM(EPROM)、电可擦除PROM(EEPROM)、寄存器、可装卸式磁盘,和用于存储可由计算机存取和读取的代码的任何其它合适的媒体。存储媒体1204可体现在制品(例如,计算机程序产品)中。以实例说明,计算机程序产品可包含封装材料中的计算机可读媒体。鉴于上文,在一些实施方案中,存储媒体1204可为非暂时性(例如,有形的)存储媒体。

[0198] 存储媒体1204可耦合到处理电路1210使得处理电路1210可从存储媒体1204读取信息和向存储媒体1204写入信息。也就是说,存储媒体1204可耦合到处理电路1210使得存储媒体1204至少可由处理电路1210接入,包含其中至少一个存储媒体与处理电路1210成一体式的实例和/或其中至少一个存储媒体与处理电路1210分离(例如,驻留在设备1200中、在设备1200外部,跨越多个实体分布等)的实例。

[0199] 由存储媒体1204存储的代码在由处理电路1210执行时致使处理电路1210执行本文中所描述的各种功能和/或过程操作中的一或多者。举例来说,存储媒体1204可包含经配置用于调整处理电路1210的一或多个硬件块处的操作以及利用通信接口1202用于利用其相应通信协议的无线通信的操作。

[0200] 处理电路1210通常适于处理,包含存储在存储媒体1204上的此代码的执行。如本文所使用,术语“代码”或“指令”将大致解释为包含但不限于:编程、指令、指令集、数据、代码、码段、程序代码、程序、子程序、软件模块、应用、软件应用、软件包、例程、子例程、对象、可执行文件、执行线程、过程、函数等,不论被称作软件、固件、中间件、微码、硬件描述语言或其它。

[0201] 处理电路1210经布置以获得、处理和/或发送数据,控制数据存取和存储,发出命

令，且控制其它所要操作。在至少一个实例中，处理电路1210可包含经配置以实施由适当媒体提供的所要代码的电路。举例来说，处理电路1210可实施为一或多个处理器、一或多个控制器和/或经配置以执行可执行代码的其它结构。处理电路1210的实例可包含通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它可编程逻辑组件、离散门或晶体管逻辑、离散硬件组件，或其经设计以执行本文所描述功能的任何组合。通用处理器可包含微处理器，以及任何常规处理器、控制器、微控制器或状态机。处理电路1210也可实施为计算组件的组合，例如DSP和微处理器的组合、许多微处理器的组合、一或多个微处理器结合DSP核心、ASIC和微处理器的组合，或任何其它数目的变化配置。处理电路1210的这些实例是用于说明，且本发明范围内的其它合适配置也是预期的。

[0202] 根据本发明的一或多个方面，处理电路1210可适于执行本文中所描述的设备中的任一者或全部的特征、过程、函数、操作和/或例程中的任一者或全部。如本文所使用，相对于处理电路1210的术语“适于”可指代处理电路1210经配置、采用、实施和/或编程以执行根据本文中所描述的各种特征的特定过程、函数、运算和/或例程，这些当中的一或多者。

[0203] 根据设备1200的至少一个实例，处理电路1210可包含以下中的一或多者：用于接收信号的电路/模块1220、用于确定状态值的电路/模块1222、用于确定旋转距离的电路/模块1224、用于旋转替换框值的电路/模块1226、用于组合的电路/模块1228，以及用于产生信号的电路/模块1230。

[0204] 用于接收信号的电路/模块1220可包含适于执行关于(例如)接收明文、密码术文本或呈电信号的形式的其它信息的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于接收信号的代码1232)。初始地，用于接收信号的电路/模块1220获得所接收的信息。举例来说，用于接收信号的电路/模块1220可直接从设备的组件(例如，接收器1216、存储器装置1208或某一其它组件)获得此数据。在一些实施方案中，用于接收信号的电路/模块1220识别存储器装置1208中的值的存储器位置，且调用所述位置的读取。在一些实施方案中，用于接收信号的电路/模块1220处理所接收的信息。用于接收信号的电路/模块1220随后输出所接收的信息(例如，将信息存储在存储器装置1208中或将信息发送到设备1200的另一组件)。

[0205] 用于确定状态值的电路/模块1222可包含适于执行关于(例如)获得待用于迭代密码术的给定实例的状态值的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于确定状态值的代码1234)。在一些实施方案中，初始状态值与密码密钥组合。在获得状态值后，用于确定状态值的电路/模块1222使得能够通过密码术对所述值进行操作(例如，将指示存储在存储器装置1208中或将指示发送到设备1200的另一组件)。

[0206] 用于确定旋转距离的电路/模块1224可包含适于执行关于(例如)依据由用于确定状态值的电路/模块1222确定的状态值计算旋转值(即，数目)的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于确定旋转距离的代码1236)。在一些实施方案中，此涉及获得(例如，掩蔽)状态值的子集、扩展所述子集，以及从扩展值确定至少一个旋转值(例如，旋转值的集合)。在获得状态值后，用于确定旋转距离的电路/模块1224将值传递到另一操作(例如，将指示存储在存储器装置1208中或将指示发送到设备1200的另一组件)。

[0207] 用于旋转替换框值的电路/模块1226可包含适于执行关于(例如)执行旋转运算的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于旋转替换框值的代码

1238)。用于旋转替换框值的电路/模块1226获得旋转距离和替换框值，且将替换框值旋转(例如，移位)所要位数。用于旋转替换框值的电路/模块1226随后输出经旋转值(例如，将值存储在存储器装置1208中或将指示发送到设备1200的另一组件)。

[0208] 用于组合的电路/模块1228可包含适于执行关于(例如)将经旋转替换框值与状态值组合的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于组合的代码1240)。用于组合的电路/模块1228获得经旋转替换框值和状态值，且组合(例如，XOR)所述两个值。用于组合的电路/模块1228随后输出组合值(例如，将值存储在存储器装置1208中或将指示发送到设备1200的另一组件)。

[0209] 用于产生信号的电路/模块1230可包含适于执行关于(例如)产生状态值的后续或最终迭代的若干功能的电路和/或代码(例如，存储在存储媒体1204上的用于产生信号的代码1242)。在一些实施方案中，用于产生信号的电路/模块1230将状态值的迭代与密码密钥组合。用于产生信号的电路/模块1230随后产生对应于所述值的信号且输出所述信号(例如，将值存储在存储器装置1208中或将指示发送到设备1200的另一组件)。

[0210] 如上文所提及，由存储媒体1204存储的代码在由处理电路1210执行时致使处理电路1210执行本文中所描述的各种功能和/或过程操作中的一或多者。举例来说，存储媒体1204可包含：用于接收信号的代码1232、用于确定状态值的代码1234、用于确定旋转距离的代码1236、用于旋转替换框值的代码1238、用于组合的代码1240和用于产生信号的代码1242中的一或多者。

[0211] 实例过程

[0212] 图13说明根据本发明的一些方面的密码术过程1300。过程1300可在处理电路(例如，图7的处理电路702、图8的处理电路802、图9的处理电路902、图10的处理电路1002，或图12的处理电路1210)内发生，所述处理电路可位于电子装置、收发器或某一其它合适的设备中。当然，在本发明的范围内的各种方面中，过程1300可由能够支持密码术操作的任何合适的设备实施。

[0213] 在框1302处，接收第一信号。举例来说，处理电路可从存储器装置检索数据(例如，明文或密文)。

[0214] 在框1304处，基于第一信号确定状态值。

[0215] 在一些实施方案中，在块1306-1312的操作之前，将待加密的信息(例如，明文或密文)与密码密钥组合。举例来说，在一些方面中(例如，对于加密运算)，框1304处状态值的确定涉及通过将密码密钥与状态值的先前实例组合产生状态值的当前实例。如本文所论述，组合可包含XOR运算、加法运算、减法运算、某一其它运算或这些运算中的一或多者的组合。

[0216] 在一些实施方案中，待加密的信息(例如，明文或密文)在块1306-1312的操作之前旋转。举例来说，在一些方面中(例如，对于解密运算)，框1304处状态值的确定涉及通过旋转状态值的先前实例产生状态值的当前实例。

[0217] 在框1306处，基于状态值确定旋转距离。举例来说，框1306的操作可对应于通过图5的γ函数510或通过图66的γ函数610，确定待在等式10的第二指令处、等式12的第二指令处、等式15的第二指令处、等式17的第二和第三指令处执行的向右旋转(ROR)量。

[0218] 框1308的旋转距离的确定可在不同实施方案中以不同方式执行。在一些方面，旋转距离的确定可基于状态值的子集。在一些方面，旋转距离的确定可包含：从状态值选择位

的子集;确定对应于选定位子集的第一数目;确定对应于替换框值的位的数量的第二数目;以及将第一数目乘以第二数目。在一些方面,旋转距离的确定可包含:掩蔽状态值,以及使经掩蔽状态值移位。在一些方面,旋转距离的确定可进一步包含:基于经移位的经掩蔽状态值使第二值移位,掩蔽经移位第二值,以及使经掩蔽的经移位第二值移位。

[0219] 在框1308处,基于框1306处确定的旋转距离旋转第一值(例如,替换框值)。在一些方面,第一值可为替换框值(例如,来自S框的值、从S框导出的值,或用于产生S框的值)。在一些方面,第一值可具有某一值使得经旋转替换框值与状态值的组合为对射的。在一些方面,第一值可为对射的。

[0220] 框1308的旋转可在不同实施方案中以不同方式执行。在一些方面,框1308的旋转可涉及旋转指令。

[0221] 在框1310处,经旋转第一值与状态值组合。在一些方面,框1308的旋转和框1310的组合可对状态值施加非线性,且在状态值的后续实例内扩散所述组合。在一些实施方案中,框1308的旋转和框1310的组合可在单一指令中组合(例如,如等式10、12、15或17中)。

[0222] 框1310的组合可在不同实施方案中以不同方式执行。在一些方面,所述组合可包含掩蔽经旋转第一值,以及将经掩蔽的经旋转第一值与状态值组合。在一些方面,所述组合可包含通过组合经旋转第一值和状态值产生状态值的中间实例。再次,如本文所论述,组合可包含XOR运算、加法运算、减法运算、某一其它运算,或这些操作中的一或多者的组合。

[0223] 在框1312处,基于来自框1310的经旋转第一值和状态值的组合产生第二信号。举例来说,处理电路可将所得状态值存储在存储器装置中。

[0224] 框1312的信号产生可在不同实施方案中以不同方式执行。在一些方面中(例如,对于加密运算),第二信号的产生可包含通过旋转状态值的中间实例产生状态值的后续实例。状态值的中间实例可旋转基于替换框值的位数量的位数量。在一些方面中(例如,对于解密运算),状态值的中间实例的产生可包含将密码密钥与经旋转第一值和状态值的所述组合进行组合。

[0225] 如虚线1314表示,框1306-1312的操作可重复直至对状态值施加所希望水平的扩散和混淆为止。因此,旋转距离的确定、替换框值的旋转以及经旋转替换框值和状态值的组合可以迭代方式执行以基于所接收的第一信号产生第二信号。

[0226] 如上文结合图5所论述,对于第一密码术,旋转距离的确定、第一值的旋转以及经旋转第一值与状态值的组合可通过以下操作执行:s^=ROR(sbox,(s&M1)<<S1)&MASK,以及s=ROR(s,R1),其中^=为XOR运算,ROR为向右旋转运算,&为AND运算,<<为位向左移位运算,s对应于状态值的实例,M1为第一掩码值(例如,0xF),MASK为第二掩码值,S1为移位值(例如,2),R1为旋转值(例如,4),且sbox对应于第一值的实例。

[0227] 同样如上文结合图5所论述,对于第一密码术的倒转,旋转距离的确定、第一值的旋转以及经旋转第一值与状态值的组合可通过以下操作执行:s=ROL(s,R1),以及s^=ROR(sbox,(s&M1)<<S1)&MASK,其中^=为XOR运算,ROL为向左旋转运算,ROR为向右旋转运算,&为AND运算,<<为位向左移位运算,s对应于状态值的实例,M1为第一掩码值(例如,0xF),MASK为第二掩码值,R1为旋转值(例如,4),S1为移位值(例如,2),且sbox对应于第一值的实例。

[0228] 如上文结合图6所论述,对于第二密码术,旋转距离的确定、第一值的旋转以及经

旋转第一值与状态值的组合可通过以下操作执行:s^=ROR(sbox,(s&M1)<<S1),以及s=ROR(s,R1),其中^=为XOR运算,ROR为向右旋转运算,&为AND运算,<<为位向左移位运算,s对应于状态值的实例,M1为掩码值(例如,0xF),S1为移位值(例如,2),R1为旋转值(例如,4),且sbox对应于第一值的实例。

[0229] 同样如上文结合图6所论述,对于第二密码术的倒转,旋转距离的确定、第一值的旋转以及经旋转第一值与状态值的组合可通过以下操作执行:s=ROL(s,R1),shift=(tinv>>((state&M1)<<S1))<<S2,以及s^=ROR(sbox,shift),其中^=为XOR运算,ROL为向左旋转运算,ROR为向右旋转运算,&为AND运算,<<为位向左移位运算,>>为位向右移位运算,s对应于状态值的实例,M1为掩码值(例如,0xF),S1为第一移位值(例如,2),S2为第二移位值(例如,2),R1为第一旋转值(例如,4),shift为第二旋转值,tinv为倒转替换框值,且sbox对应于第一值的实例。

[0230] 图14说明根据本发明的一些方面的密码术过程1400的另一实例。过程1400可在处理电路(例如,图7的处理电路702、图9的处理电路902或图12的处理电路1210)内发生,所述处理电路可位于电子装置、收发器或某一其它合适的设备中。当然,在本发明的范围内的各种方面中,过程1400可由能够支持密码术操作的任何合适的设备实施。

[0231] 在框1402处,接收第一参数的值。举例来说,密码术装置可接收待加密的明文。作为另一实例,密码术装置可接收待解密的密文。此明文或密文可因此充当密码术过程1400中使用的状态参数(第一参数)的第一实例。在一些方面,明文或密文可与密码密钥进行XOR运算以提供给定轮次的状态参数的第一实例(例如,如上文“paul”密码术中)。

[0232] 在框1404处,基于第一参数的当前实例的子集旋转第二参数。举例来说,可将S框向量(第二参数)旋转基于状态参数的半字节的量。在一些方面,第二参数基于第一参数的当前实例的子集的旋转包含:从第一参数的当前实例选择位的子集;以及将选定位子集乘以对应于替换框中的给定一者中的位数量(例如,四个位)的数目(例如,四)。

[0233] 在框1406处,通过将框1404的结果与第一参数的当前实例组合产生第一参数的中间实例。举例来说,可通过组合S框向量和状态参数而产生状态参数的新值。应了解,可例如在等式10的第二指令处或等式15的第二指令处执行框1404-1406的操作。

[0234] 在一些方面,第一参数的中间实例的产生还包含组合密码密钥和状态参数的此新值(例如,参看等式10)。可例如如图7的等式706的第二指令线或图9的等式906的第二指令线处所示执行框1406的操作的方面。

[0235] 在框1408处,通过旋转第一参数的中间实例产生第一参数的后续实例。举例来说,可将在框1406处产生的状态参数的新值旋转所界定位数。在一些方面,将第一参数的中间实例旋转对应于替换框中的给定一者中的位数量(例如,四个位)的位数量。可例如在等式10的第三指令处或等式15的第三指令处执行框1406的操作。

[0236] 图15说明根据本发明的一些方面提供图13的密码术过程1300的更详细实例的密码术过程1500。过程1500可在处理电路(例如,图7的处理电路702、图9的处理电路902,或图12的处理电路1210)内发生,所述处理电路可位于电子装置、收发器或某一其它合适的设备中。当然,在本发明的范围内的各种方面中,过程1500可由能够支持密码术操作的任何合适的设备实施。

[0237] 在框1502处,初始化状态参数。举例来说,框1502的操作可对应于图13的框1302的

操作，借此状态参数的初始值是基于输入值(例如，明文或密文)。

[0238] 在框1504处，掩蔽除状态参数(例如，一个半字节)的子集外的状态参数的所有位。如等式10和12中所展示，此可通过对状态参数和值0xF进行AND运算来实现。

[0239] 在框1506处，框1504的结果乘以所界定值(例如，四)以产生第一旋转值。如等式10和12中所展示，此可通过将AND运算的结果向左移位两个位来实现。

[0240] 在框1508处，S框值(例如，向量)向右旋转由框1506处产生的第一旋转值指定的位数。

[0241] 在框1510处，框1508的结果与状态参数组合(例如，进行XOR运算)以提供状态参数的经更新值。应了解，框1504-1510的操作可例如在等式10或等式15的第二指令线处执行。

[0242] 在框1512处，框1510的结果与密码密钥值组合(例如，进行XOR运算)以提供状态参数的另一经更新值。框1512的操作可例如如图7的等式706的第二指令线或图9的等式906的第二指令线处所示而执行。如上文描述的“paul”密码术中指示，举例来说，状态参数可与密码术的其它部分处的密钥值进行XOR运算。

[0243] 在框1514处，框1512的结果根据第二旋转值向右旋转以提供密码术的下一轮次的状态参数的下一迭代。框1514的操作可例如在等式10或15的第三指令线处执行。

[0244] 框1504到1514的操作以迭代方式执行以基于输入值(明文)产生经加密值(密文)或基于输入值(密文)产生经解密值(明文)。

[0245] 图16说明根据本发明的一些方面的另一密码术过程1600。过程1600为图14的过程1400的倒转。举例来说，过程1400可用于加密，且过程1600用于解密，或反之亦然。

[0246] 过程1600可在处理电路(例如，图8的处理电路802、图10的处理电路1002，或图12的处理电路1210)内发生，所述处理电路可位于电子装置、收发器或某一其它合适的设备中。当然，在本发明的范围内的各种方面中，过程1600可由能够支持密码术操作的任何合适的设备实施。

[0247] 在框1602处，接收第一参数的值。举例来说，密码术装置可接收待解密的密文。作为另一实例，密码术装置可接收待加密的明文。此明文或密文可因此充当密码术过程1600中使用的状态参数(第一参数)的第一实例。在一些方面，明文或密文可与密码密钥进行XOR运算以提供状态参数的第一实例(例如，如上文“paul”密码术中)。

[0248] 在框1604处，通过旋转第一参数的当前实例产生第一参数的中间实例。举例来说，可将状态参数的新当前值旋转所界定位数。在一些方面，将第一参数的中间实例旋转对应于替换框中的给定一者中的位数量(例如，四个位)的位数量。框1606的操作可例如在等式12或17的第一指令线处执行。

[0249] 在一些方面，第一参数的中间实例的产生还包含组合密码密钥和状态参数的此新值(例如，参看等式12)。可例如如图8的等式806的第二指令线或图10的等式1006的第二指令线处所示执行框1604的操作的方面。

[0250] 在框1606处，基于第一参数的中间实例的子集旋转第二参数。举例来说，可将S框向量(第二参数)旋转基于状态参数的半字节的量。在一些方面，第二参数基于第一参数的当前实例的子集的旋转包含：从第一参数的当前实例选择位的子集；以及将选定位子集乘以对应于替换框中的给定一者中的位数量(例如，四个位)的数目(例如，四)。

[0251] 在框1608处，通过将框1606的结果与第一参数的中间实例组合产生第一参数的后

续实例。举例来说，可通过组合S框向量和状态参数而产生状态参数的新值。应了解，可例如在等式12的第二指令线处或等式17的第三指令线处执行框1606-1608的操作。

[0252] 图17说明根据本发明的一些方面提供图16的密码术过程1600的更详细实例的密码术过程1700。过程1700为图15的过程1500的倒转。举例来说，过程1500可用于加密，且过程1700用于解密，或反之亦然。

[0253] 过程1700可在处理电路(例如，图8的处理电路802、图10的处理电路1002，或图12的处理电路1210)内发生，所述处理电路可位于电子装置、收发器或某一其它合适的设备中。当然，在本发明的范围内的各种方面中，过程1700可由能够支持密码术操作的任何合适的设备实施。

[0254] 在框1702处，初始化状态参数。举例来说，框1702的操作可对应于图13的框1302的操作，借此状态参数的初始值是基于输入值(例如，明文或密文)。

[0255] 在框1704处，状态参数根据第一旋转值向左旋转以提供密码术的下一轮次的状态参数的下一迭代。框1704的操作可由等式12或17的第一指令线执行。

[0256] 在框1706处，框1704的结果与密码密钥值组合(例如，进行XOR运算)以提供状态参数的另一经更新值。框1706的操作可例如在图8的等式806或图10的等式1006的第二指令线处执行。如上文描述的“paul”密码术中指示，状态参数可与密码术的其它部分处的密钥值进行XOR运算。

[0257] 在框1708处，掩蔽除状态参数(例如，一个半字节)的子集外的状态参数的所有位。如等式12和17中所展示，此可通过对状态参数和值0xF进行AND运算来实现。

[0258] 在框1710处，框1708的结果乘以四以产生第二旋转值。如等式12和17中所展示，此可通过将AND运算的结果向左移位两个位来实现。

[0259] 在框1712处，S框向量向右旋转由在框1710处产生的第二旋转值指定的位数。

[0260] 在框1714处，框1712的结果与状态参数组合(例如，进行XOR运算)以提供状态参数的经更新值。应了解，框1708-1714的操作可由等式12的第二指令线或等式17的第二和第三指令线执行。

[0261] 框1704到1714的操作以迭代方式执行以基于输入值(密文)产生经解密值(明文)或基于输入值(明文)产生经加密值(密文)。

[0262] 结论

[0263] 图式中所说明的组件、步骤、特征和/或功能中的一或多者可以重新布置及/或组合成单个组件、步骤、特征或功能或体现在若干组件、步骤或功能中。在不脱离本文所揭示的新颖特征的情况下，还可添加额外元件、组件、步骤和/或功能。图中说明的设备、装置和/或组件可经配置以执行本文中所描述的方法、特征或步骤中的一或多者。本文中所描述的新颖算法还可有效地实施于软件中和/或嵌入于硬件中。

[0264] 应理解，所揭示方法中的步骤的特定次序或层级为示范性过程的说明。基于设计偏好，应理解，可以重新布置方法中的步骤的特定次序或层级。随附的方法权利要求项以示例次序呈现各种步骤的要素，且除非本文中特别叙述，否则其不意图限于所呈现的特定次序或层级。在不脱离本发明的情况下，也可添加或不利用额外元件、组件、步骤和/或功能。

[0265] 虽然可能已相对于特定实施方案和图式论述本发明的特征，但本发明的所有实施方案可包含本文论述的有利特征中的一或多者。换句话说，虽然一或多个实施方案可能已

论述为具有特定有利特征，但也可根据本文论述的各种实施方案的任一者使用此类特征中的一或多者。类似地，虽然示范性实施方案可能已在本文论述为装置、系统或方法实施方案，应理解，此些示范性实施方案可在各种装置、系统和方法中实施。

[0266] 而且应注意，至少一些实施方案已经描述为过程，所述过程经描绘为流程图、流图、结构图或框图。尽管流程图可将操作描述为循序过程，但许多操作可并行或同时执行。另外，可以重新布置操作的次序。过程在其操作完成时终止。在一些方面，过程可对应于方法、函数、程序、子例程、子程序等。当过程对应于函数时，其终止对应于函数返回到调用函数或主函数。本文中所描述的各种方法中的一或多者可部分或完全由代码(例如，指令和/或数据)实施，所述代码可存储在机器可读、计算机可读和/或处理器可读存储媒体中，且由一或多个处理器、机器和/或装置执行。

[0267] 所属领域的技术人员将进一步理解，结合本文中所揭示的实施方案描述的各种说明性逻辑块、模块、电路和算法步骤可被实施为硬件、软件、固件、中间件、微码或其任何组合。为清楚地说明此可互换性，上文已大体上关于其功能性而描述了各种说明性组件、块、模块、电路和步骤。此功能性是实施为硬件还是软件取决于特定应用及施加于整个系统的设计约束。

[0268] 在本发明内，词语“示范性”用于表示“充当实例、例子或说明”。本文中描述为“示范性”的任何实施方案或方面未必应解释为比本发明的其它方面优选或有利。同样，术语“方面”不要求本发明的所有方面包含所论述的特征、优点或操作模式。术语“耦合”在本文中用于指两个物体之间的直接或间接耦合。举例来说，如果物体A物理地触摸物体B，且物体B触摸物体C，那么物体A和C可仍被视为彼此耦合，即使它们不直接彼此物理地触摸也如此。举例来说，第一裸片可耦合到封装中的第二裸片，即使第一裸片永不直接物理地接触第二裸片也如此。术语“电路”广义上使用，且希望包含电气装置和导体的硬件实施方案以及信息和指令的软件实施方案两者，所述电气装置和导体当连接和配置时使得能够执行本发明中描述的功能((不限于)关于电子电路的类型)，且所述信息和指令在由处理器执行时使得能够执行本发明中描述的功能。

[0269] 如本文所使用，术语“确定”涵盖各种如本文所使用，术语“确定”涵盖各种各样的动作。举例来说，“确定”可包含推算、计算、处理、导出、调查、查找(例如，在表、数据库或另一数据结构中查找)、断定等等。并且，“确定”可包含接收(例如，接收信息)、存取(例如，存取存储器中的数据)，等等。而且，“确定”可包含解析、选择、挑选、建立等等。

[0270] 提供先前描述以使所属领域的技术人员能够实践本文中所描述的各种方面。对这些方面的各种修改对于所属领域的技术人员来说将容易显而易见，并且本文中定义的一般原理可适用于其它方面。因此，权利要求书不希望限于本文中所展示的方面，而是应符合与权利要求的语言一致的完整范围，其中参考呈单数形式的元件不希望意味着“一个且仅一个”(除非明确地如此陈述)，而是为“一或多个”。除非另外确切地陈述，否则术语“一些”是指一或多个。涉及项目列表中的“至少一者”的短语是指那些项目的任何组合，包含单个成员。作为一实例，“以下各者中的至少一者：a、b或c”希望涵盖：a；b；c；a和b；a和c；b和c；以及a、b和c。所属领域的一般技术人员已知或日后将知晓的贯穿本发明而描述的各种方面的元件的所有结构和功能等效物以引用的方式明确地并入本文中，且既定由权利要求书涵盖。此外，本文揭示的任何内容均不希望奉献给公众，无论权利要求书中是否明确地陈述此公

开。权利要求要素不应依据35U.S.C.§112第六章的条款解释，除非所述要素是明确地使用短语“用于…的装置”来叙述，或者在方法权利要求项的情况下，所述要素是使用短语“用于…的步骤”来叙述。

[0271] 因此，在不脱离本发明的范围的情况下，可在不同实例及实施方案中实施与本文中所描述及附图中所展示的实例相关联的各种特征。因此，尽管已经描述和在附图中展示某些特定构造和布置，但此类实施方案仅为说明性且不限制本发明的范围，因为所属领域的一般技术人员将了解对所描述实施方案的各种其它添加和修改以及自所描述实施方案的删除。因此，本发明的范围仅由所附权利要求书的文字语言及合法等效物来确定。

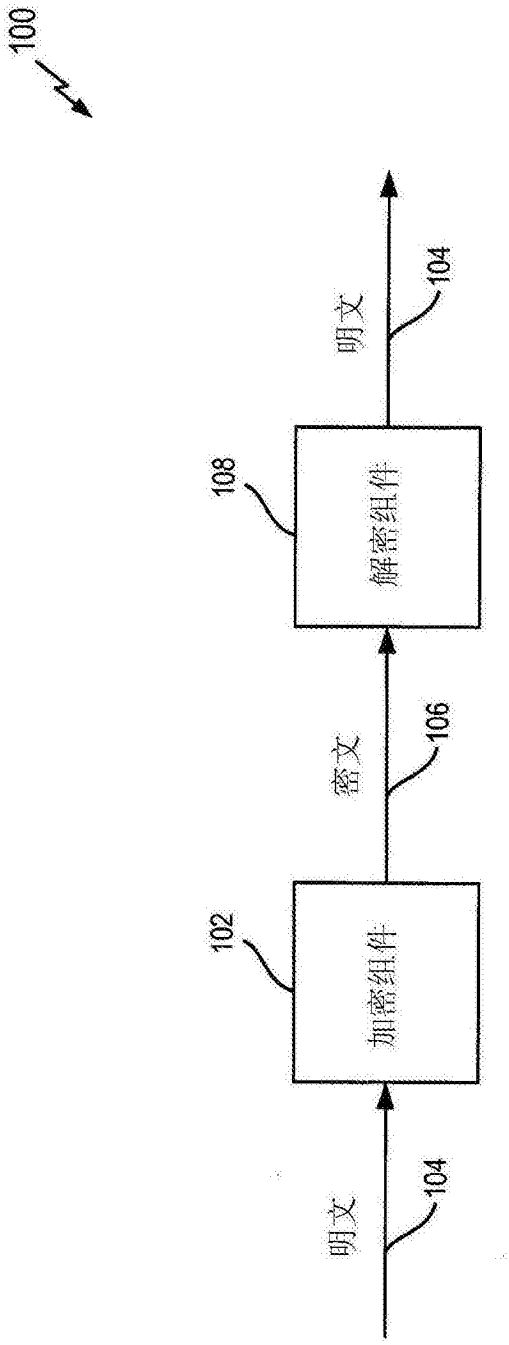


图1

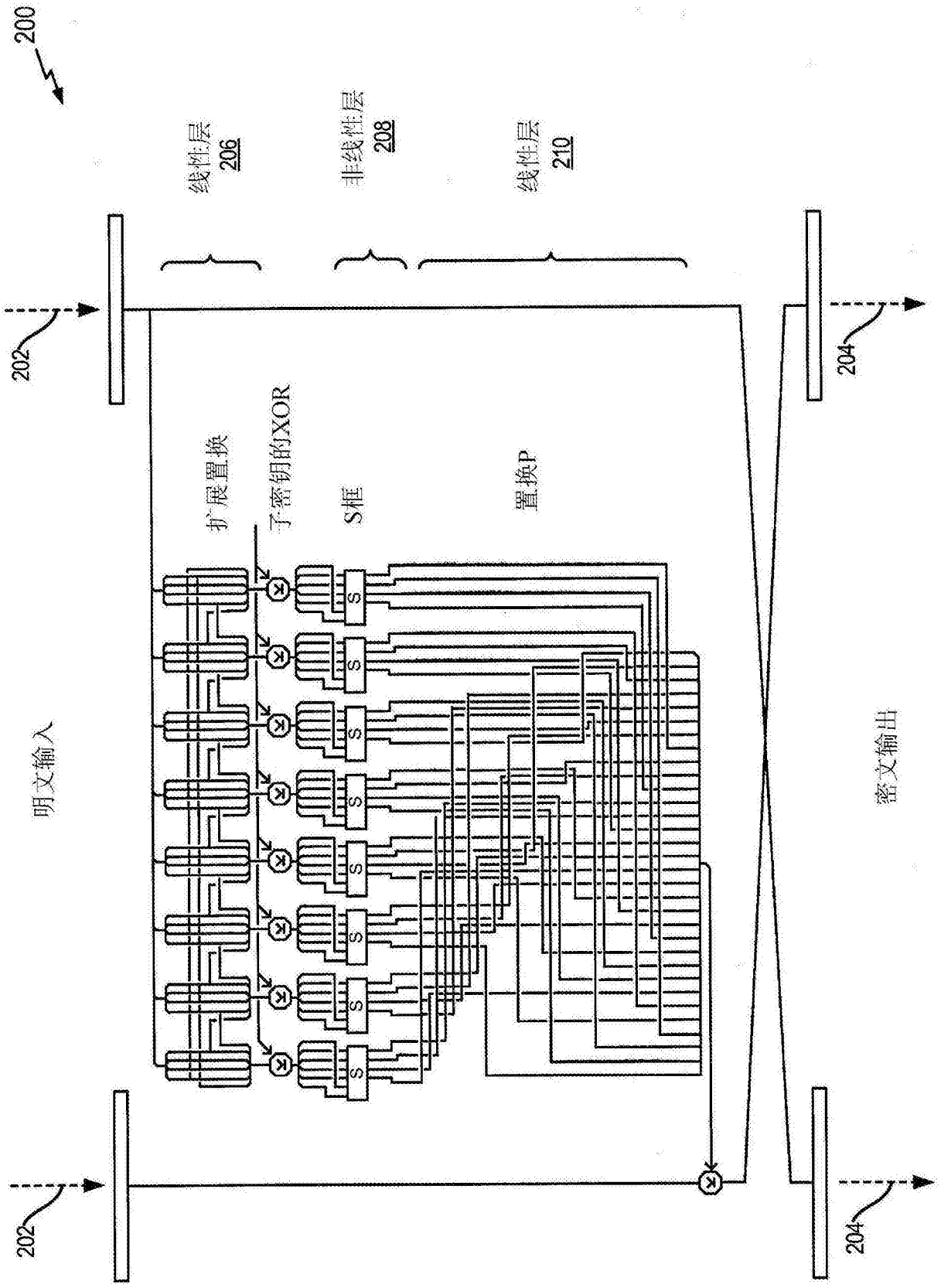

200
线性层
206
非线性层
208
线性层
210
202
204
扩展置换
子密钥的XOR
S框
置换P
明文输入
密文输出
S

图2

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

图3

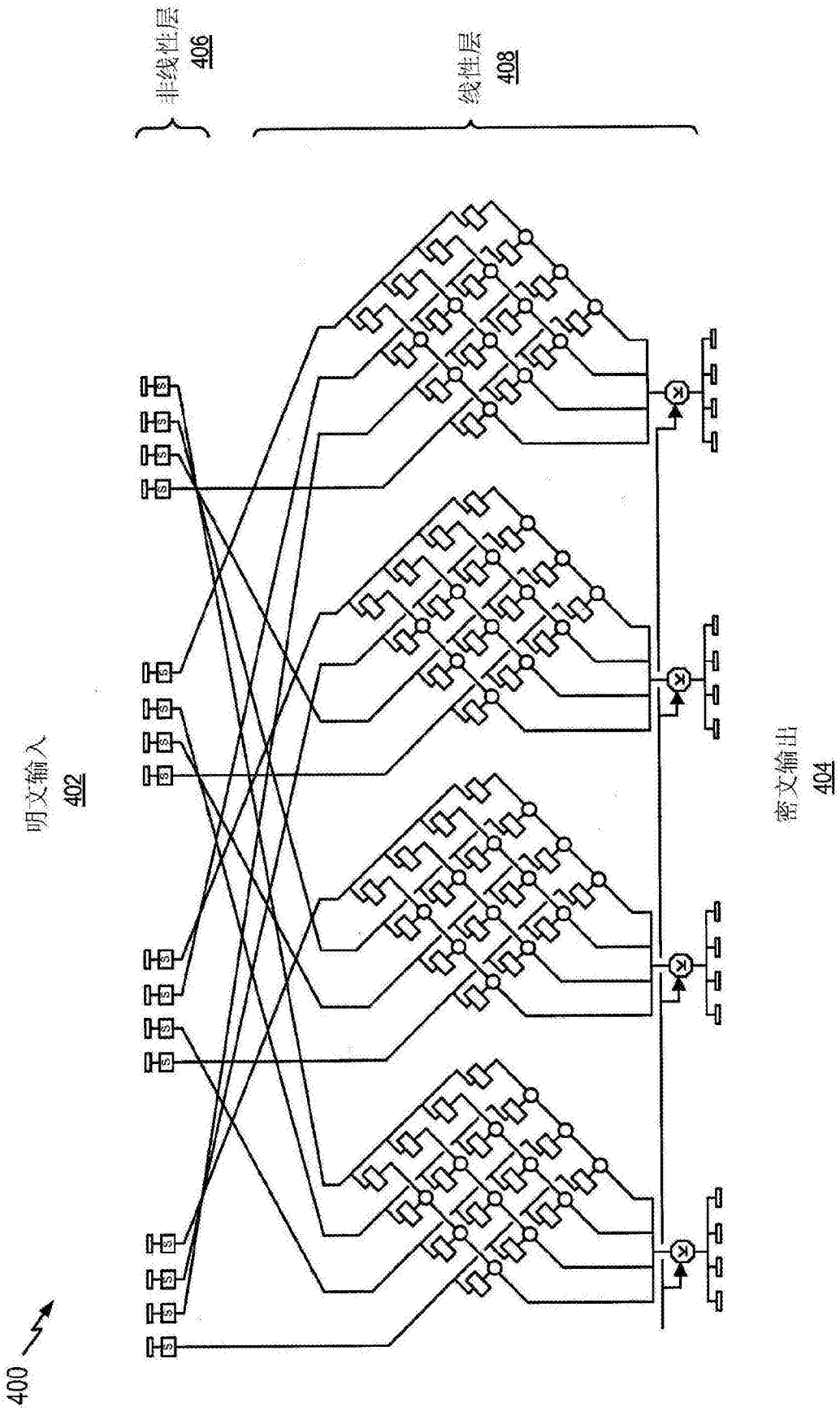


图4

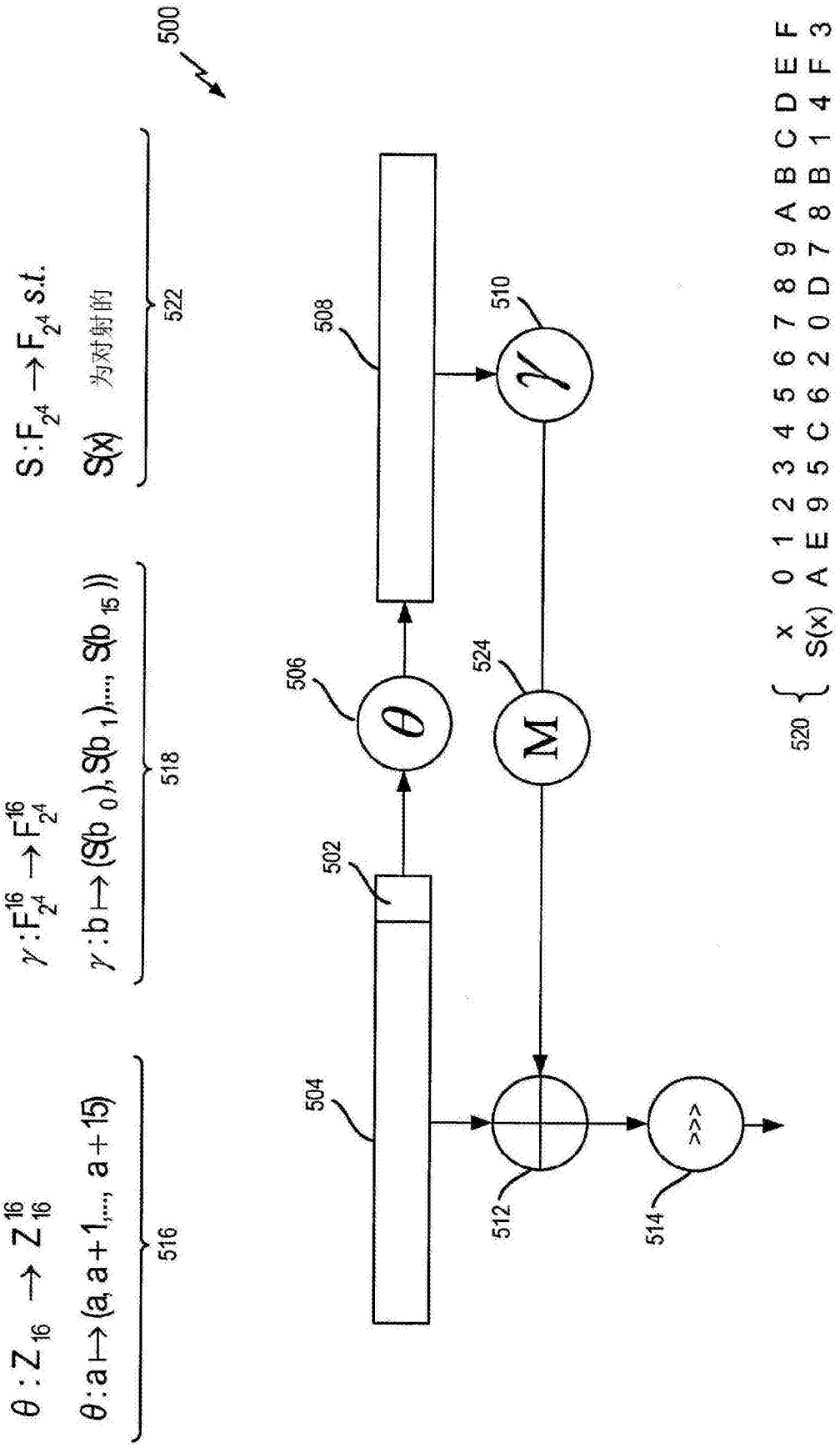

500
$S:F_{2^4} \rightarrow F_{2^4}$ s.t.
S(x) 为对射的
522
$\gamma:F_{2^4}^{16} \rightarrow F_{2^4}^{16}$
$\gamma:b \mapsto (S(b_0),S(b_1),\ldots,S(b_{15}))$
518
$\theta:Z_{16} \rightarrow Z_{16}^{16}$
$\theta:a \mapsto (a,a+1,\ldots,a+15)$
516
502
504
506
508
510
512
514
524
M
>>>
520
x 0 1 2 3 4 5 6 7 8 9 A B C D E F
S(x) A E 9 5 C 6 2 0 D 7 8 B 1 4 F 3

图5

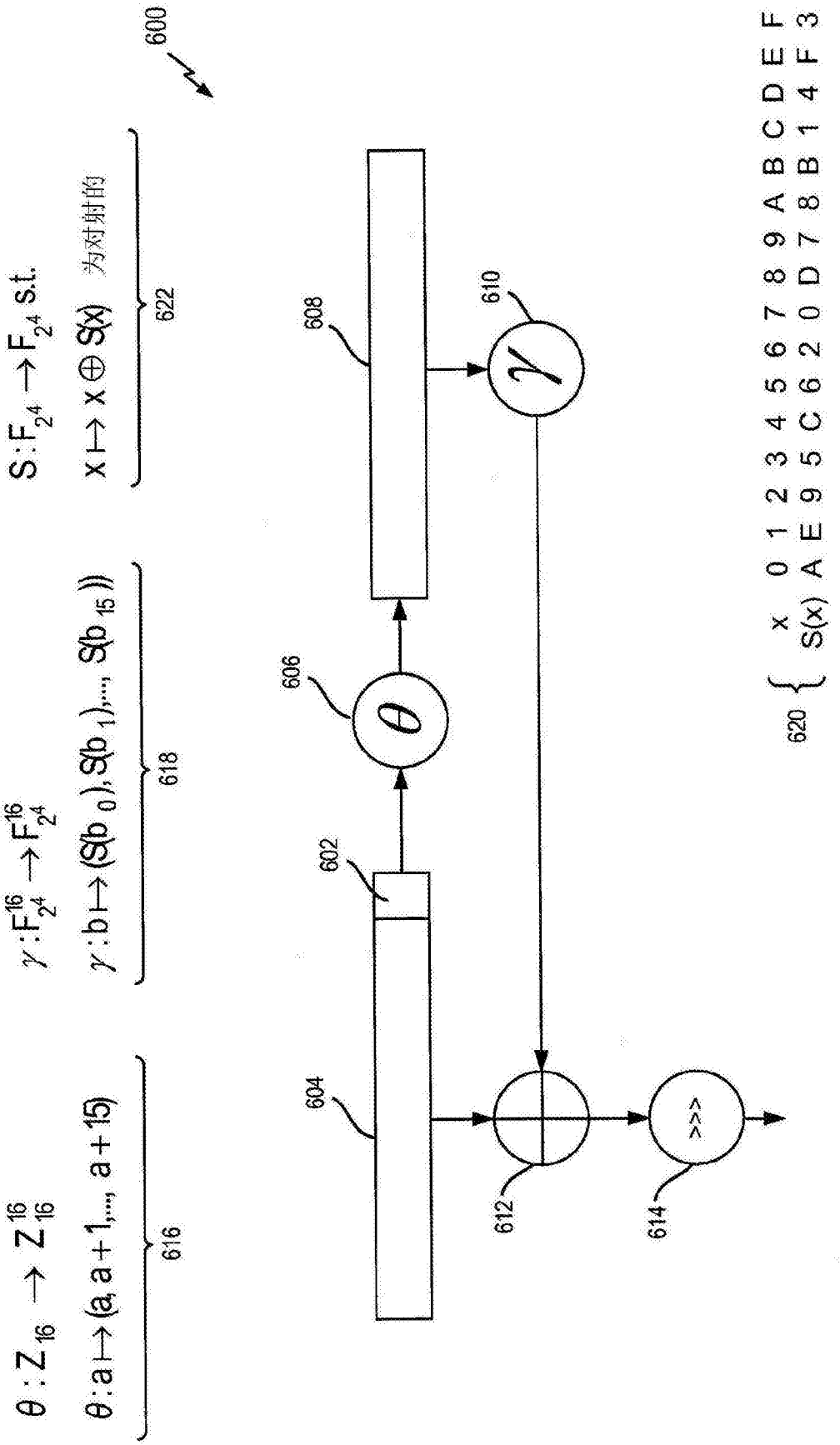

600
θ : Z16 → Z16 16
θ: a ↦ (a, a+1,..., a+15)
616
γ : F16 → F16 24 24
γ : b ↦ (S(b 0), S(b 1),..., S(b 15))
618
S : F24 → F24 s.t.
x ↦ x ⊕ S(x) 为对射的
622
602
604
606
608
610
612
614
<<<
620
x 0 1 2 3 4 5 6 7 8 9 A B C D E F
S(x) A E 9 5 C 6 2 0 D 7 8 B 1 4 F 3

图6

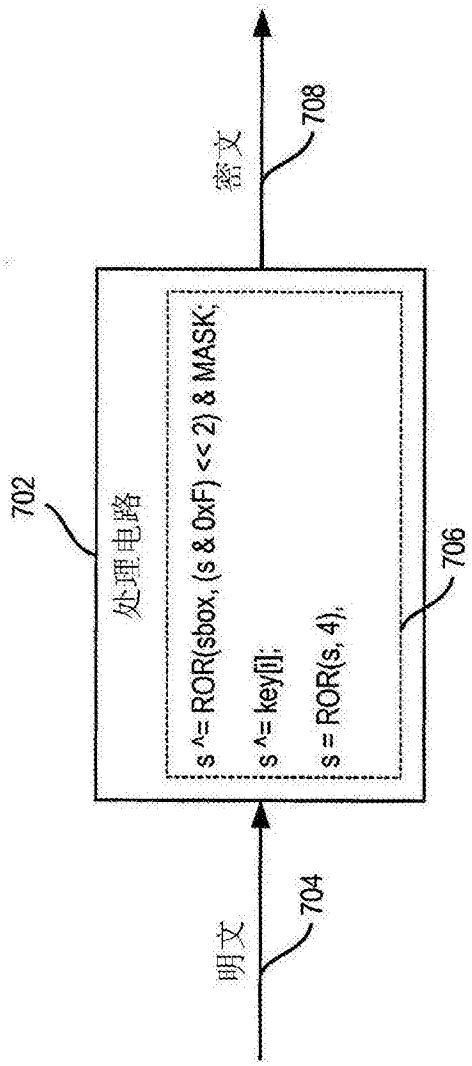


图7

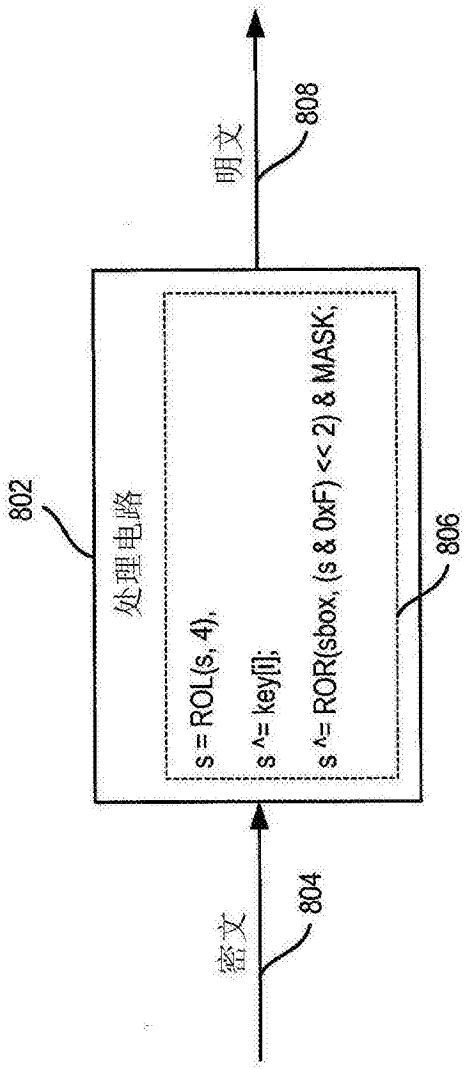


图8

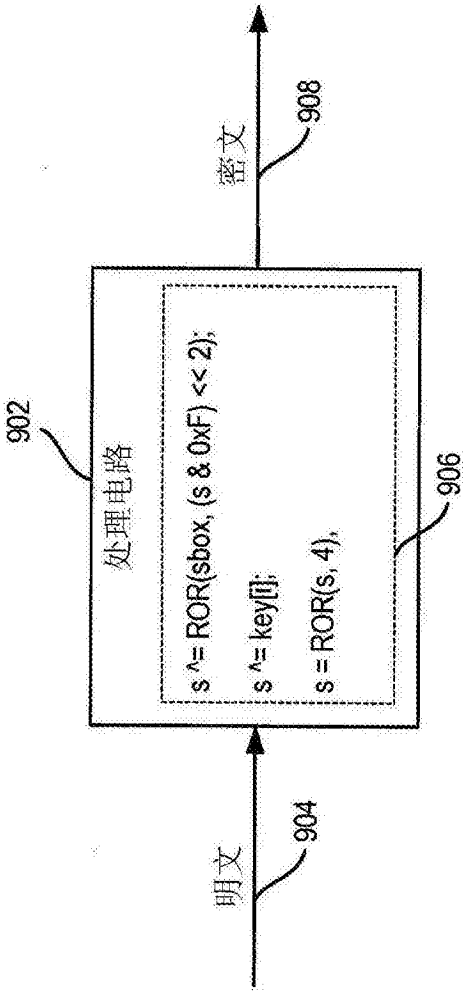


图9

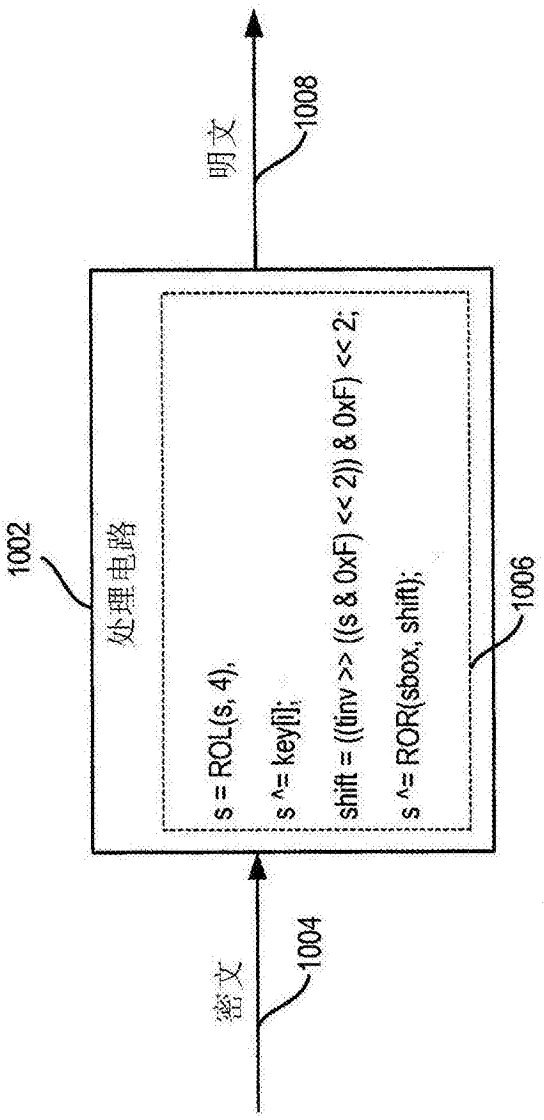

明文
1008
处理电路
1002
s = ROL(s, 4);
s ^= key[i];
shift = ((tinv >> ((s & 0xF) << 2)) & 0xF) << 2;
s ^= ROR(sbox, shift);
1006
密文
1004

图10

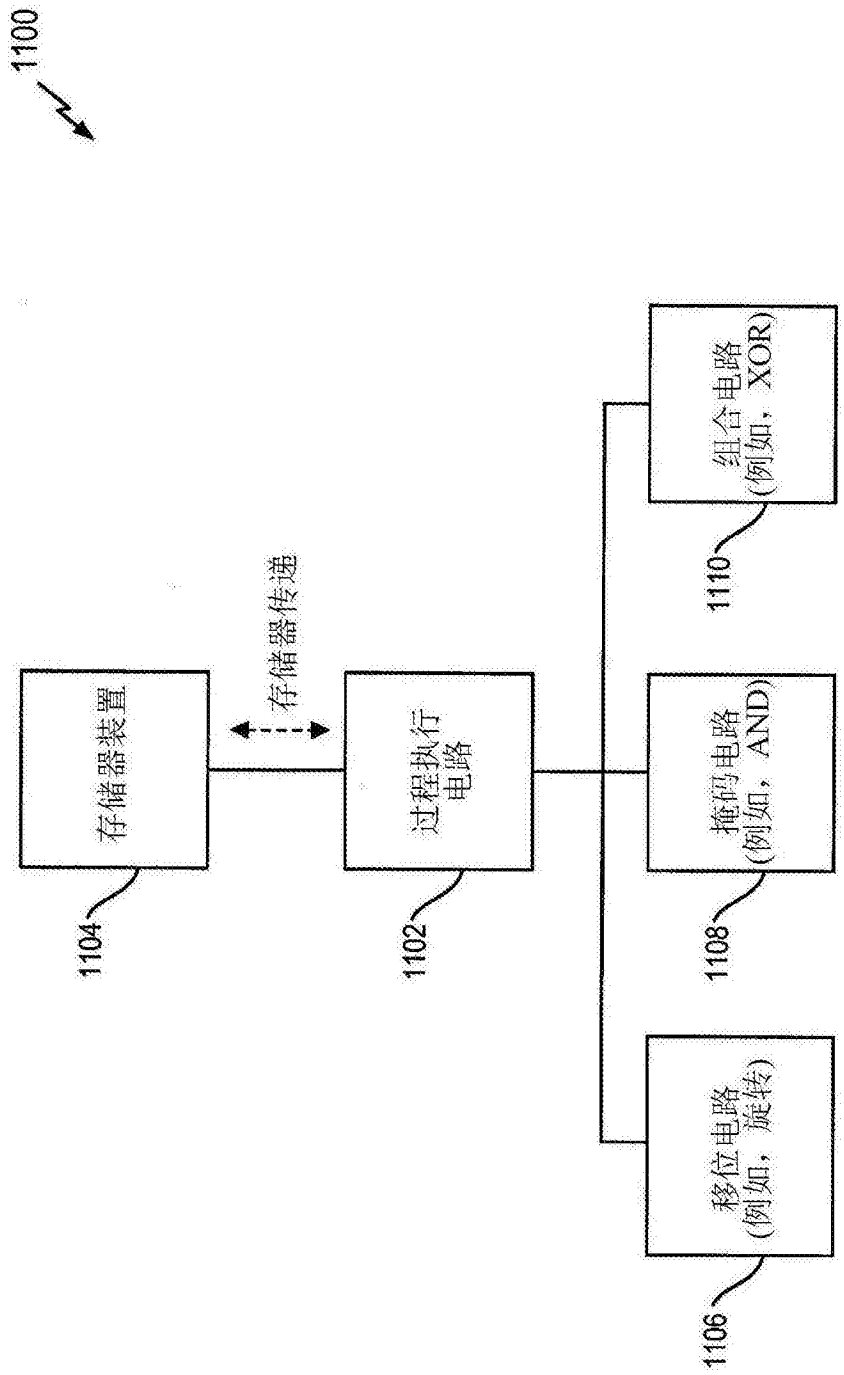

1100
存储器装置
存储器传递
过程执行电路
组合电路(例如，XOR)
掩码电路(例如，AND)
移位电路(例如，旋转)
1104
1102
1110
1108
1106

图11

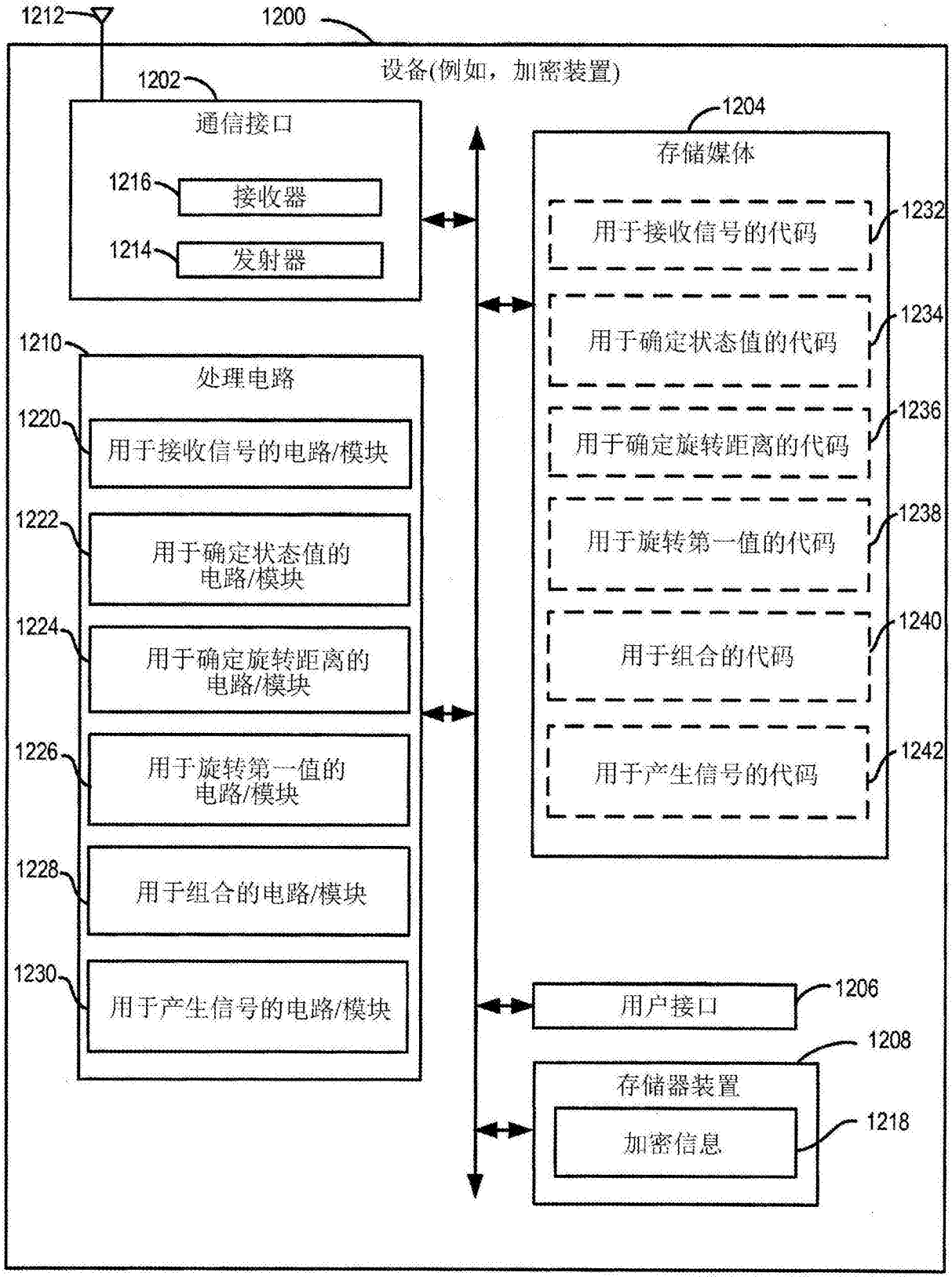

1212
1200
设备(例如，加密装置)
1202
通信接口
1216
接收器
1214
发射器
1204
存储媒体
用于接收信号的代码
1232
用于确定状态值的代码
1234
用于确定旋转距离的代码
1236
用于旋转第一值的代码
1238
用于组合的代码
1240
用于产生信号的代码
1242
1210
处理电路
1220
用于接收信号的电路/模块
1222
用于确定状态值的电路/模块
1224
用于确定旋转距离的电路/模块
1226
用于旋转第一值的电路/模块
1228
用于组合的电路/模块
1230
用于产生信号的电路/模块
用户接口
1206
1208
存储器装置
加密信息
1218

图12

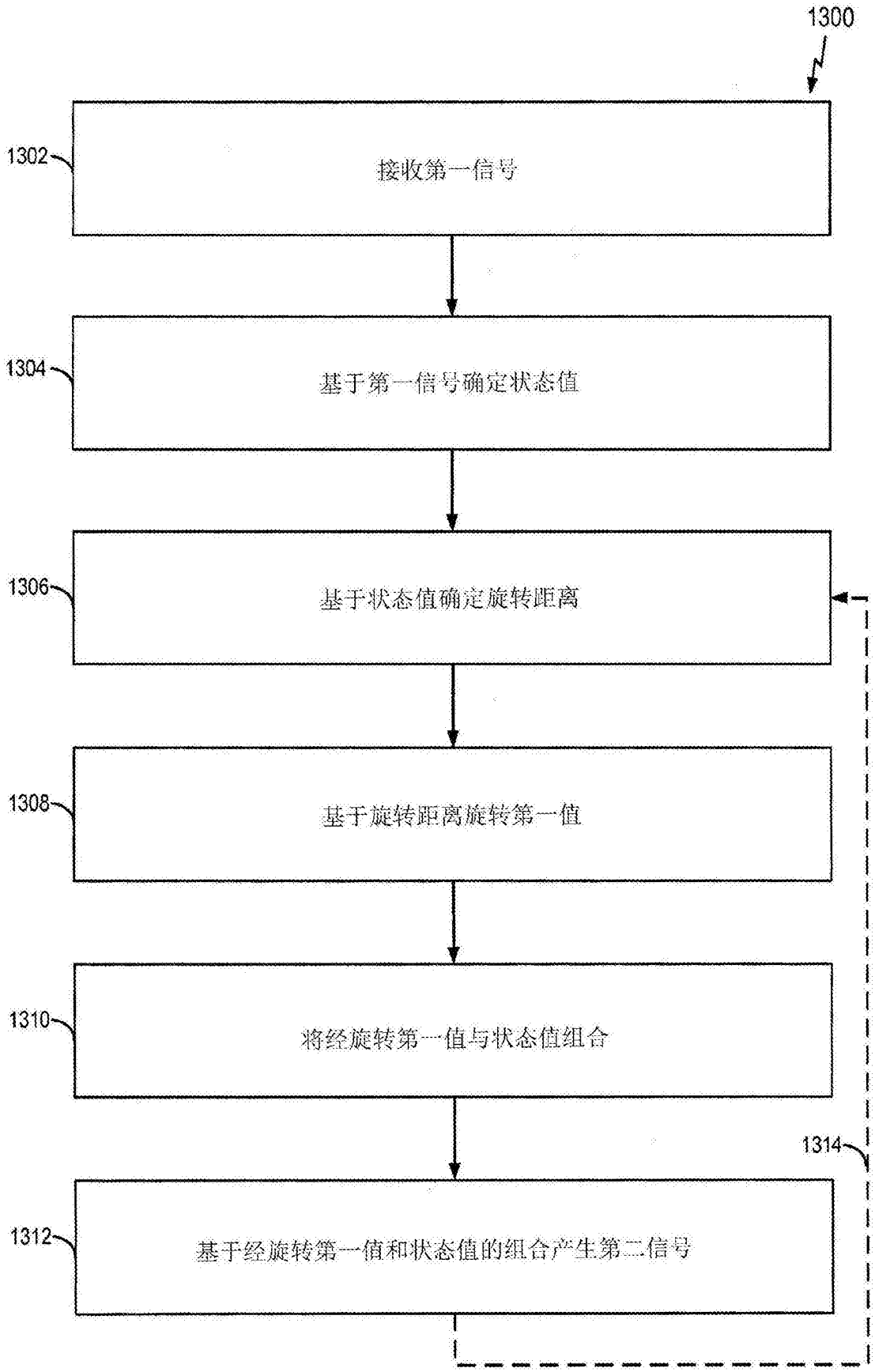

1300
1302
接收第一信号
1304
基于第一信号确定状态值
1306
基于状态值确定旋转距离
1308
基于旋转距离旋转第一值
1310
将经旋转第一值与状态值组合
1312
基于经旋转第一值和状态值的组合产生第二信号
1314

图13

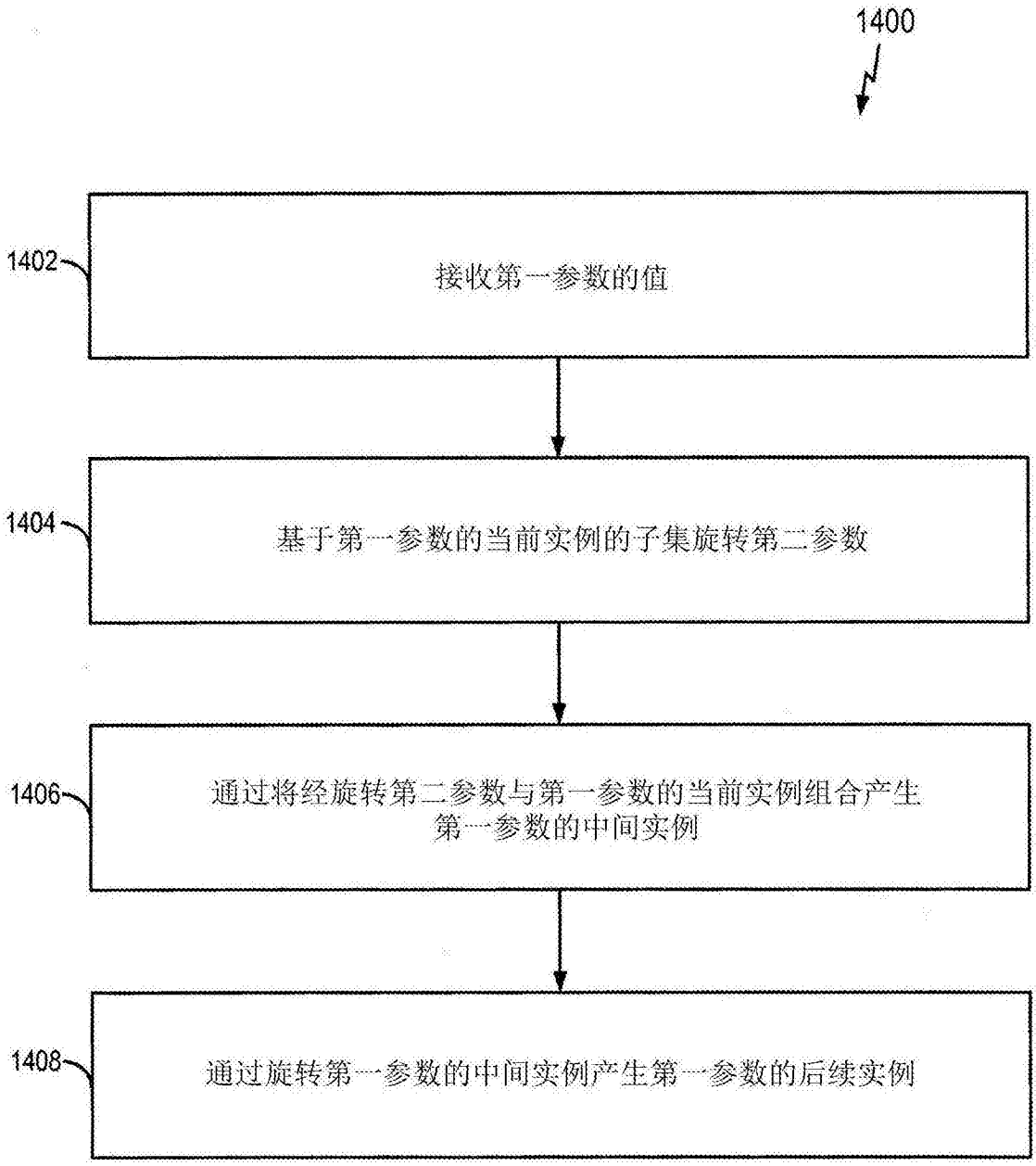

1400
1402
接收第一参数的值
1404
基于第一参数的当前实例的子集旋转第二参数
1406
通过将经旋转第二参数与第一参数的当前实例组合产生第一参数的中间实例
1408
通过旋转第一参数的中间实例产生第一参数的后续实例

图14

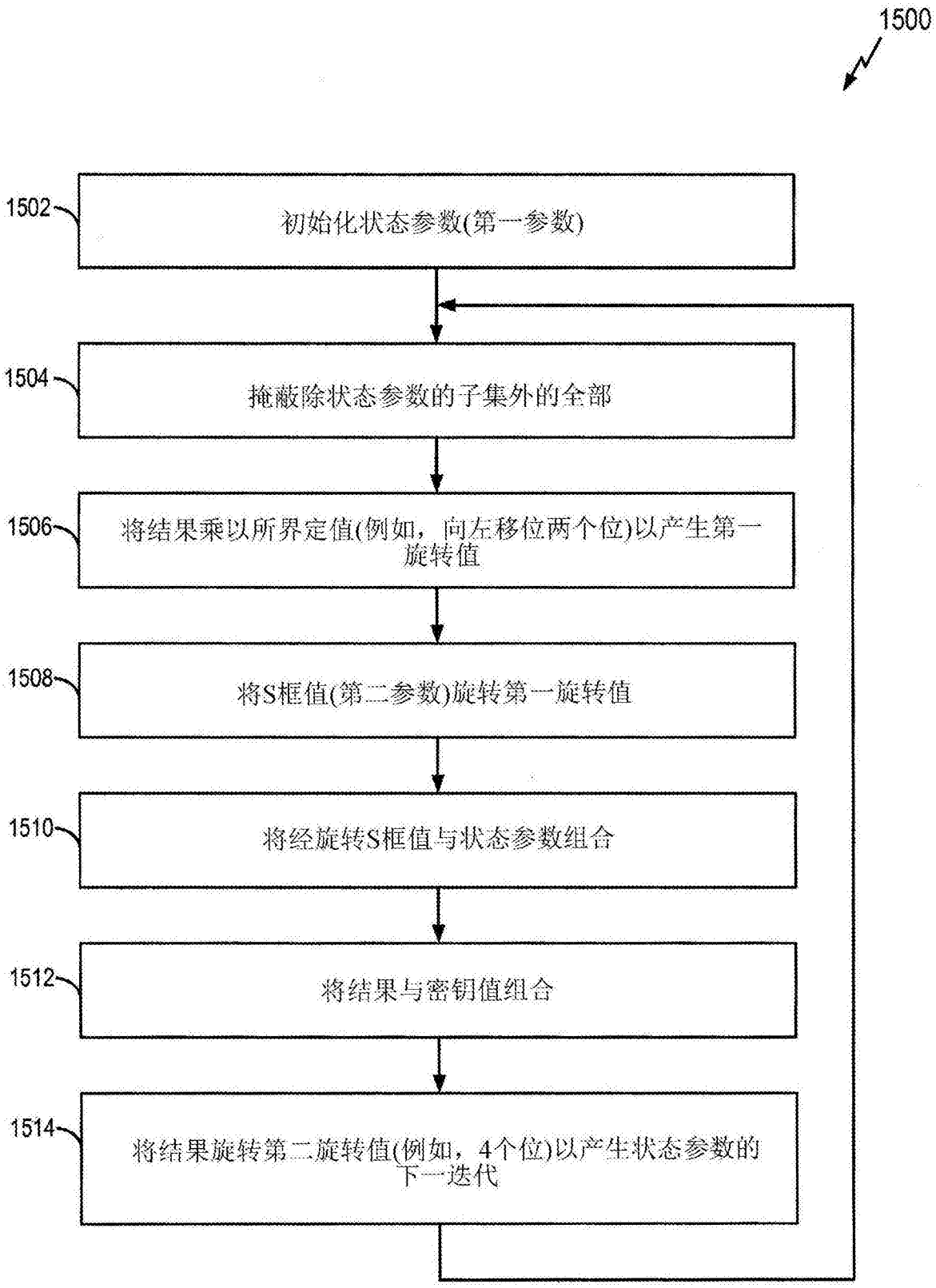

1500
1502
初始化状态参数(第一参数)
1504
掩蔽除状态参数的子集外的全部
1506
将结果乘以所界定值(例如，向左移位两个位)以产生第一旋转值
1508
将S框值(第二参数)旋转第一旋转值
1510
将经旋转S框值与状态参数组合
1512
将结果与密钥值组合
1514
将结果旋转第二旋转值(例如，4个位)以产生状态参数的下一迭代

图15

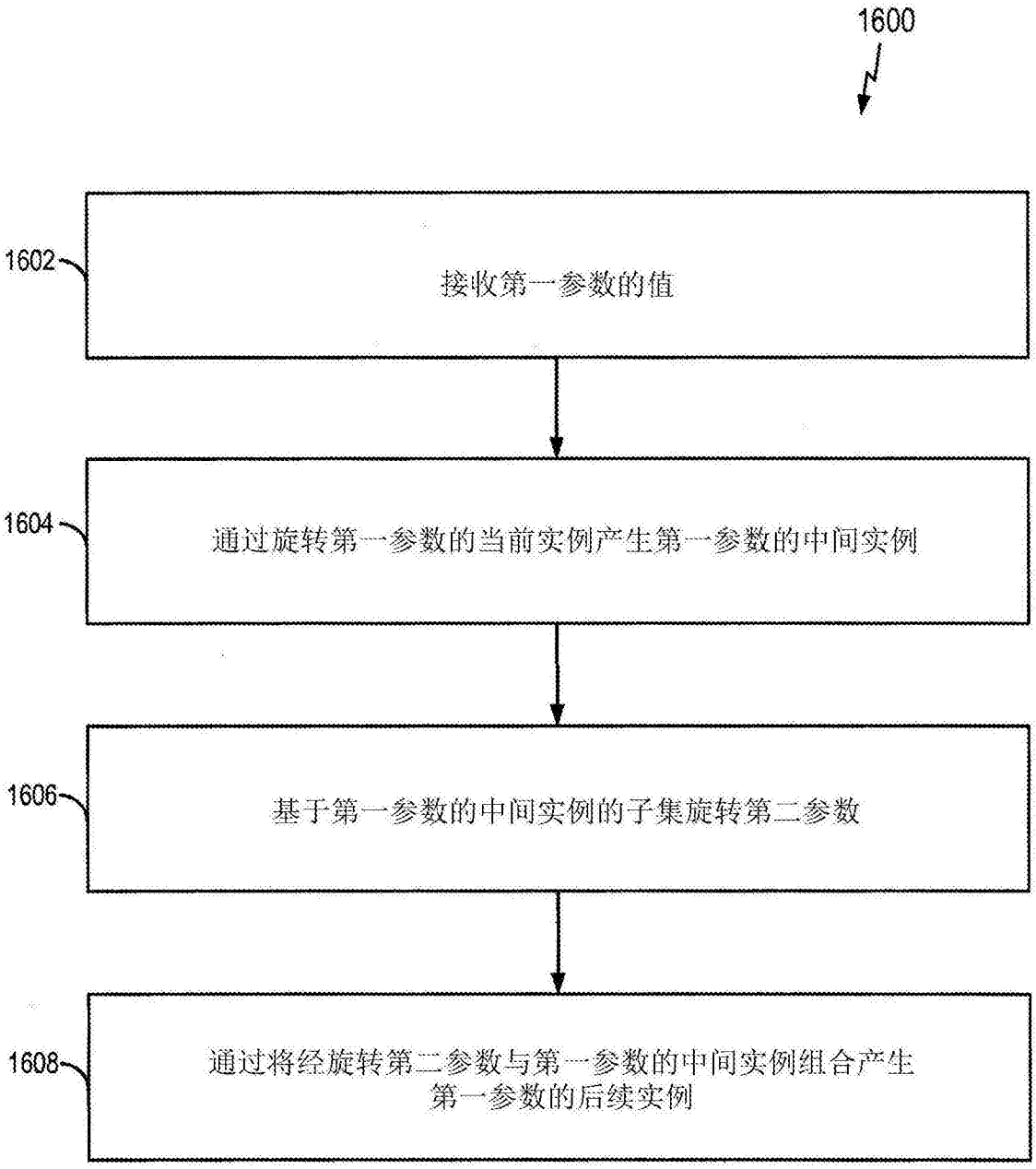

1600
1602
接收第一参数的值
1604
通过旋转第一参数的当前实例产生第一参数的中间实例
1606
基于第一参数的中间实例的子集旋转第二参数
1608
通过将经旋转第二参数与第一参数的中间实例组合产生第一参数的后续实例

图16

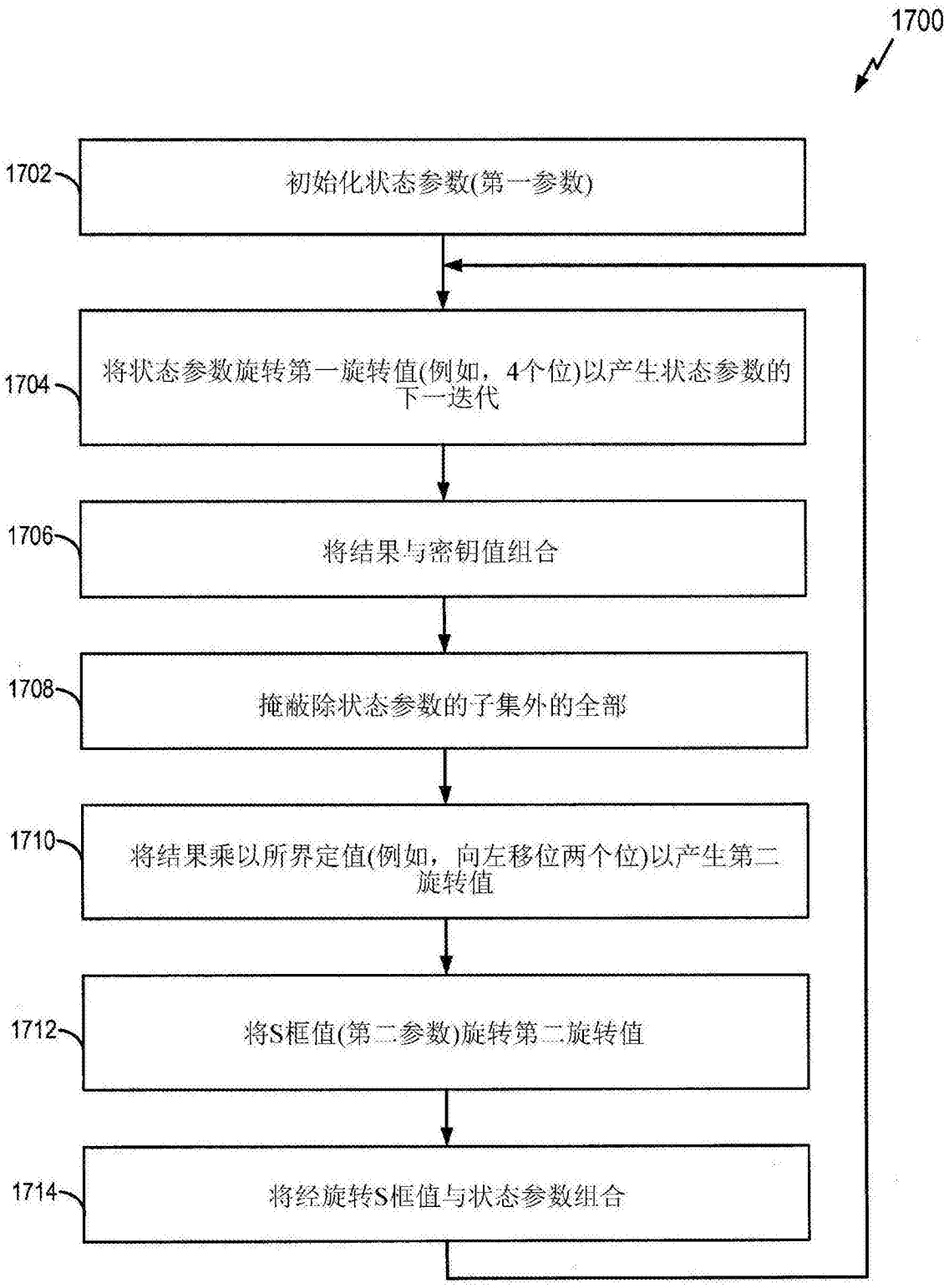

1700
1702
初始化状态参数(第一参数)
1704
将状态参数旋转第一旋转值(例如，4个位)以产生状态参数的下一迭代
1706
将结果与密钥值组合
1708
掩蔽除状态参数的子集外的全部
1710
将结果乘以所界定值(例如，向左移位两个位)以产生第二旋转值
1712
将S框值(第二参数)旋转第二旋转值
1714
将经旋转S框值与状态参数组合

图17