[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G07F 7/10

[12] 发明专利说明书

专利号 ZL 00810258.9

[45] 授权公告日 2005 年 12 月 14 日

[11] 授权公告号 CN 1231871C

[22] 申请日 2000.5.11 [21] 申请号 00810258.9
[30] 优先权
[32] 1999. 5.11 [33] FR [31] 99/06264
[86] 国际申请 PCT/FR2000/001269 2000.5.11
[87] 国际公布 WO2000/068901 法 2000.11.16
[85] 进入国家阶段日期 2002.1.11
[71] 专利权人 格姆普拉斯公司
地址 法国热姆诺
[72] 发明人 D·纳卡彻
审查员 杨勤之

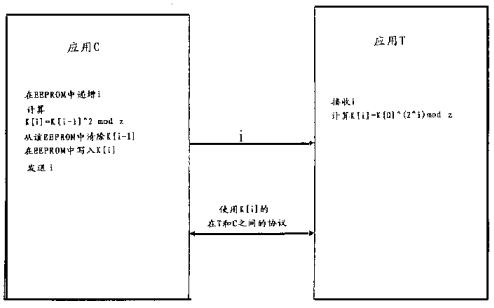
[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 王 勇 张志醒

权利要求书 2 页 说明书 13 页 附图 7 页

[54] 发明名称 在使用一种密钥动态加密算法的电子元件中的对抗方法

[57] 摘要

本发明涉及在一个使用一个尺寸为 k 的密钥 K [0] 加密算法 A，并和一个第二电子元件(T)通信的第一电子元件(C)中的一种对抗方法。该对抗方法的特征在于：通过一个函数 K [i] =f [(k [i-1])、传输 i 到元件(T)、以及允许元件(T)从 K [0] 计算 K [i] 而不用产生在 K [0] 到 K [i] 之间从 k [1] 到 k [i-1] 这 i-1 个密钥的计算简化来系统化和有规则地计算 K [i]。



ISSN1008-4274

1. 一种在第一电子元件（C）和与第一电子元件（C）进行通信的 第二电子元件（T）中对抗方法，其中该第一和第二电子元件使用具有长度 k 的密钥的一种加密算法 A，该密钥最初为 K[0]并且在加密算法 A 的连续使用之间变化到 K[i]，其中 i 是算法 A 的执行次数：

K[i] = f [(k[i-1])，

其中 f 是取一个密钥用作一个输入并返回一个密钥作为一个输出的函数

其特征在于所述第二电子元件（T）从 K[0]开始计算 K[i]而不必在 K[0]和 K[i]之间产生 i-1 个密钥 K[1]、……K[i-1]。

2. 如权利要求 1 所述的对抗方法， 其特征在于：函数 f(x)如下所示，其中 x 为一个变量：

f (x) = x^e mod z，

其中 z 是一个常数从而使由第二电子元件（T）使用的该计算简化为公式 K[i] = K[0]^(e^i) mod z，其中 e 是一个整数，e^i 的计算被理解为模 phi(z)。

3. 如权利要求 1 所述的对抗方法， 其特征在于：函数 f(x)如下所示，其中 x 为一个变量：

f(x) = x*c mod z

其中 z 和 c 是常数，从而使由第二电子元件（T）使用的计算简化为 K[i] = K[0]*c^i mod z。

4. 如权利要求 1、2 或 3 的任何一个所述的对抗方法，其特征在于：i 由所述第一电子元件（C）提供给所述第二电子元件（T）以允许第二电子元件（T）获得基于已供应 i 的密钥 K[i] 。

5. 如权利要求 2 或 3 所述的对抗方法， 其特征在于：z 为小于 2^k 的最大质数。

6. 如权利要求 1 所述的对抗方法， 其中加密算法 A 是 DES、IDEA、AES、FEAL、三元 DES、BlowFish、SAFER、SHA－MAC、RIPEMD、DFC、RC5、RC6 或 SEAL。

7. 一种智能卡，其特征在于：它构成了第一电子元件（C），并实现了权利要求 1 到 6 中的任何一个。

8 . 一种电子终端，其特征在于：它构成了第二电子元件（T），

并实现了权利要求 1 到 6 中的任何一个。

在使用一种密钥动态加密算法的电子元件中的对抗方法

本发明涉及在一个使用一种密钥加密算法的电子元件中的一种对抗方法。这种元件在严格控制对服务或数据的访问的应用中使用。它们具有一个在微处理器和存储器附近形成的、包含一个具有密钥的程序存储器的体系结构。

这些元件特别使用在智能卡中，用于其中的某些应用。举例来说，涉及访问某些数据库的应用、金融应用、或例如用于电视、汽油分配或高速公路通行费的远程支付应用。

这些元件或卡因此使用一种密钥加密算法，其中最公知的一种算法是 DES（在英国和美国文献中，意思是数据加密标准）算法。还存在其它的密钥算法，诸如 RC5 算法或 COMP128 算法。当然这个列表没有穷举。

概括和简洁地来说，这些算法的作用是从用作一主机系统（服务器、银行分配器等等）的一个（到该卡的）输入的一个信息和包含在该卡中的密钥计算一个加密信息，并接下来将这个加密信息提供给该主机系统，例如该加密信息使得该主机系统能够验证该元件或卡、交换数据等等。

然而，很清楚这些元件或卡容易受到包含对电流消耗的一个差值分析的攻击，这使有不良企图的第三方能够发现该密钥。这些攻击被称为 DPA 攻击，是差值能力分析的英语缩写。

这些 DPA 攻击的原理基于以下事实：微处理器执行指令的电流消耗依据正被操作的数据而变化。

特别地，微处理器操作一个数据位的一条指令，取决于该位是等于“1”或“0”而产生两个不同的电流特征。通常，如果该指令正操作一个“0”，则在执行的这个时刻消耗电流的第一数值，而且如果该指令正操作一个“1”，则消耗该电流的不同于第一数值的第二数值。

已知这些加密算法的如下特征：进行的计算、使用的参数。唯一未知的是包含在该程序存储器中的密钥。这不能完全从对用作一个输入的该信息和反过来提供的该加密信息的认识中推导出。

然而，在一个加密算法中，某些计算的数据完全依赖于在该卡的输入端以明码形式应用的信息以及包含在该卡中的该密钥．在该算法中计算的其它数据因此能完全从该加密信息（通常在从该卡到该主机系统的输出端上以明码形式提供）和包含在该卡中的该密钥再计算出来．更确切地说，这些特定数据的每一位能够从输入或输出信息、以及该密钥有限数目的特定位中确定．

因此，一个特定数据项的每一位对应于由该密钥的一组特定位形成的一个子密钥．

能够被推算的这些特定数据位以下简称为目标位．

因此，DPA 攻击的基本思想是：取决于它正操作的是“1”还是“0”、以及从一个已知的输入或输出信息和在相应子密钥上的一个假设通过该算法的指令计算一个目标位的可能性，使用在一条指令的电流消耗特征中的差值．

因此，DPA 攻击的原则是测试一个给定的、应用于大量电流测定曲线的子密钥假设、和一个布尔选择函数，其中每一电流测定曲线都与攻击者的一个已知输入信息有关，且该布尔选择函数是依据该子密钥假设，使用为一个目标位而推算的值为每一曲线定义的．

通过在有关的子密钥上形成一个假设，推算这个目标位将为一个给定的输入或输出信息而使用的值为“0”或“1”在实际上是可能的．

然后，就有可能作为一个布尔选择函数，为所讨论的该子密钥假设应用该目标位预测的值“0”或“1”，以便将这些曲线分类形成以下两个信息包：依据该子密钥假设，第一个信息包包含已经知道在“0”上对该目标位进行操作的曲线，而第二个信息包包含已经知道在“1”上对该目标位进行操作的曲线．通过在每个信息包中采用平均电流消耗，就获得了一个用于第一个信息包的平均消耗曲线 M0(t)以及一个用于第二个信息包的平均消耗曲线 M1(t)．

如果该子密钥假设是正确的，则第一个信息包实际上包含了在 N 个曲线当中已经知道在“0”上对该目标位进行操作的所有曲线，且第二个信息包实际上包含了在这 N 个曲线当中已经知道在“1”上对该目标位进行操作的所有曲线．于是第一个信息包的平均消耗曲线 M0(t)除了在执行关键性指令时之外将处处具有一个平均消耗，和

一个在“0”上对该目标位进行操作的电流消耗分布特性曲线（$profile_0$）. 换句话说，就所有这些曲线来说，除了总有值为“0”的该目标位之外，所有这些操作的位等于“1”的机会与等于“0”的机会相同. 这能够被写为:

$$M0(t) = [(profile_0+profile_1)/2]_t^1{}_{tci} + [profile_0]_{tci}$$

即:

$$M0(t) = [Vm_t]_t^1{}_{tci} = [profile_0]_{tci}$$

其中 tci 表示关键时刻，在此时已经执行了一条关键性指令.

同样地，第二个信息包的平均消耗曲线 M1(t)除了在执行关键性指令时之外将处处对应于一个平均消耗，和一个在“1”上对该目标位进行操作的电流消耗分布特性曲线（$profile_1$）. 它可能被写为:

$$M1(t) = [(profile_0+profile_1)/2]_t^1{}_{tci} + [profile_1]_{tci}$$

即:

$$M1(t) = [Vm_t]_t^1{}_{tci} + [profile_1]_{tci}$$

可以看出两个分布图 $profile_0$ 和 $profile_1$ 不相等. 然后，在曲线 M0(t)和 M1(t)之间的差值给出一个其数值等于在执行这些操作该位的关键性指令的关键时刻 tci 的 $profile_0 - profile_1$ 的信号 DPA(t)，即在图 1 描绘的实例中，在点 tc0 到 tc6，除了这些关键时刻外其数值近似等于零.

如果该子密钥假设是错误的，则该分类未对应于实际. 从统计上来说，在每个信息包中实际上已经知道在“0”上对该目标位进行操作的曲线与已经知道在“1”上对该目标位进行操作的曲线同样多. 由于对每一曲线来说，所有操作的位、包括该目标位其等于“0”的机会和等于“1”的机会相同，因此结果的平均曲线 M0(t)位于由 $(profile_0+profile_1)/2 = Vm$ 给出的一个平均值附近.

相同的原因导致在第二个信息包上的一个平均电流消费曲线 M1(t)，其数值位于由 $(profile_0+profile_1)/2 = Vm$ 给出的一个平均值附近.

在这种情况下由 M0(t)-M1(t)的差值提供的信号 DPA(t)实质上等于零. 在一个错误子密钥假设的情况下信号 DPA(t)如图 2 所示. 因此 DPA 攻击依据被操作位的值，利用在一条指令执行期间电流消耗分布图中的差值，以便依据用于一个给定子密钥假设的一个布尔选择函数完

成对电流消耗曲线的分类．通过在获得的两个信息包曲线之间的平均电流消耗上施加差值分析，以获得一个信息信号 DPA(t)．

然后一个 DPA 攻击的执行总体上包含:

a—生成 N 个随机信息（例如 N 等于 1000);

b—使该卡为这 N 个随机信息中的每一个执行算法，在每个时刻读取电流消耗曲线（在该元件的供应终端上测量);

c—形成一个有关一个子密钥的假设;

d—为这些随机信息中的每一个，由目标位中的一个推算被采用的值，以便获得布尔选择函数，其中的这些目标位的值仅仅依赖于该信息（输入或输出）的这些位和作为一个假设的该子密钥;

e—依据这个布尔选择函数（即依据在该子密钥假设下分别为每一曲线推算的，用于这个目标位的值“1”或“0”）将这些曲线进行分类;

f—在每个信息包中计算产生的平均电流消耗曲线;

g—进行这些平均曲线的差值，以便获得该信号 DPA(t)．

如果有关该子密钥的假设是正确的，则该布尔选择函数是正确的，第一个信息包的曲线实际上相当于那些用作一个输入或输出的信息在该卡中已经给出了一个目标位为 0 的曲线，且第二个信息包的曲线实际上相当于那些用作一个输入或输出的信息在该卡中已经给出了一个目标位为 1 的曲线．

在图 1 中的情况适用于: 信号 DPA(t)在对应于这些关键性指令（那些操作该目标位的指令）执行的 tc0 到 tc6 时刻不为零．在采集期间至少有一个关键性时刻就足够了．

应当注意到: 攻击者不需要精确地知道这些关键性时刻．

如果该子密钥假设不正确，则该分类与实际不符，因此在每个信息包中实际上对应于一个目标位为“0”的曲线与对应于一个目标位为“1”的曲线同样多．该信号 DPA(t)实质上在整个期间为空（在图 2 描绘的情况中）．这就必须返回到步骤 c，并形成一个有关该子密钥的新假设．

如果该假设证明是正确的，则能够转到其它子密钥的评定上，直至已经最大可能地重新构成该密钥为止．举例来说，利用一种 DES 算法，使用了一个 64 位密钥，其中只有 56 位是有用位．就一个 DPA 攻击来说，有可能重新构成该 56 个有用位中的至少 48 位．

本发明的目的是在一个电子元件中使用一种不允许攻击者产生该信号 DPA(t)的对抗方法. 这通过动态地改变该密钥来实现.

本发明假定在相互通信的两个电子元件之间共享一个公共的秘密. 在下文中这些元件将在图 7 和 8 中用标记 C(一个卡)和 T(一个终端)表示, 但是显然对专家来说它们能够采用其它不同的形式. 在这两个元件当中, 假定 C 可能易遭受 DPA. 通过本发明的对抗, 电子元件 C 被保护以防 DPA 攻击.

依据本发明, 该对抗方法允许 T 和 C 以一种同步方式计算不为外界所知的一个会话密钥, 并在一个加密、验证或 MAC 密码协议或其它任何要求一种密钥算法的加密函数中使用这个密钥.

为了使在 C 和 T 之间共享的秘密数据项暴露最小, 这个秘密数据项将仅仅间接地暴露, 并将随时间变化, 在会话 i 使用的密钥是前一会话的密钥(在会话 i-1 中使用的)的一个函数. 每个会话密钥用 K[i]表示, 将最多被操作两次:

- 当它从该密钥 k_{i-1} 创造时;
- 当它在应用中使用时.

在本发明中, 将关心一种 DES 类型的算法. 这个 DES 算法包含十六个同样的计算循环; 在这个算法中, 已经表明了: 能够由一个攻击者推算的数据位于第一个循环和最后一个循环, 而就 DPA 攻击而言那些关键性指令位于最初的三个循环和最后的三个循环中.

因此,本发明的一个目的是通过确保这些数据从该算法的应用到另一应用改变而使由关键性指令操作的数据不可见. 作为特征, 本发明因此涉及在一个使用一个具有长度 k 的密钥 k[0]的加密算法 A 的电子元件 C 中的一种对抗方法, 以便从一个输入信息中计算一个加密信息. 依据本发明, 当以一种非机密方式提供给终端 T 该索引 i 时, 一个会话密钥 k[i]由前一会话密钥 k[i-1]形成. 具有 k[0]和 I, 该终端 T 能够计算 k [i]而不必计算所有中间密钥 k[1]、k[2]、……、K[i-1].

实现该对抗方法的特征在于: 在 A 的连续使用之间在 K[i]方面的变化使用以下规则进行计算, 其中 i 是算法 A 的执行次数:

K[i]=f(K[i-1]),

f 是取一个密钥用作一个输入并返回一个密钥作为一个输出的函数.

在一个实施例中，密钥 k[i]是由密钥 k[i-1]的平方对一个质数求模形成的，其中该质数的长度为该密钥的长度.

函数 f(x)如下所示，其中 x 为一个变量:

f (x) = x^2 mod z,

其中 z 为一个常数以便使由 T 使用的计算简化是通过公式 K[i] = K[0]^(2^i) mod z 计算得到，其中数 2^i 是计算的模 phi(z). 应当注意到: 除了 2 之外的一个整数幂（数字 e）能够被用在本发明的方法中; 举例来说，也可能定义为:

f(x) = x^3 mod z

然后相应的计算简化是 K[i] = K[0]^(3^i) mod z. 其中数 3^i 是计算的模 phi(z).

在另一个实施例中，密钥 k[i]是由密钥 k[i-1]乘以一个常数 c 后对一个质数求模形成的，其中该质数的长度为该密钥的长度.

函数 f(x)如下所示，其中 x 为一个变量:

f (x) = x * c mod z,

其中 z 和 c 是常数,从而使由 T 使用的计算简化为 K[i]=K[0]*c^i mod z.

优先地，z 为小于 2^k 的最大质数.

依据本发明的另一个实施例，后者也涉及在和第二个电子元件（T）进行通信、并使用一个加密算法 B 的第一个电子元件（C）中的一种对抗方法，其中该加密算法 B 使用如上所述的方法，在该方法中密钥包含一连串的字节，本发明的方法将这些字节用一个小质数进行求模，密钥 K[i-1]包含一连串的 L 字节，使用在连续使用 B[i-1]之间的 K[i-1]的对抗方法使用如下规则进行计算，其中 i 为该算法 B 的执行次数:

K[i-1]={B[1, i-1], ……, B[L, i-1]},

上述规则通过对范围从 1 到 L 的 t 进行变换变成 K[i]:

B[t, i]=B[t, i-1] ^2 mod U 其中 U 为一个字节的质数，例如 251,

密钥 K[i]使用对范围从 1 到 L 的 t 计算简化进行计算;

B[t, i]=B[t, i-1]^(2^i) mod U 其中数 2^i 为计算的模 phi(U).

优先地，这个第二实施例涉及的一种对抗方法中，密钥 K[i-1]包

含一连串的字节，本发明的方法将这些字节对一个小质数求模，其特征在于：该密钥K[i-1]包含一连串的L字节：

K[i-1] = {B[1, i-1], …… , B[L, i-1]},

然后通过对范围从1到L的t进行变换变成K[i]：

B[t, i]=B[t, i-1] *c [t] mod U 其中 U 为一个字节的质数，例如251，

然后密钥K[i]使用对范围从1到L的t计算简化进行计算：

B[t, i]=B[t, i-1]*c[t]^i mod U.

就这个特定实施例来说，还有更可取的是该终端和卡在多于密钥加密算法所必需的字节数目上执行会话密钥 K[i]的计算，在该密钥加密算法中应该使用这个密钥，然后将所产生的密钥中断，以便获得为该密钥加密算法所必需的字节数。这是为了补偿在由使用小于数字 256 的模U所引起的平均信息量中的损失。

优先地，该数U为一、二、三、四、五或六个字节。

当然，本发明也涉及用于这个最后特定实施例的一种智能卡和一种电子终端。

本发明的其它特性和优点将在下面作为指示而决不是限制、并结合附图所给出的描述中进行更加详细地描述，其中：

- 图1和2，已经描述过了，说明了能够依据一个DPA攻击，根据在该密钥K的一个子密钥上的一个假设获得的信号DPA（t）；

- 图3和4是表示该DES算法的第一个循环和最后一个循环的流程图；

- 图5是在该DES算法中使用的操作SBOX的方框图；

- 图6显示了在该操作SBOX中使用的一个输入一个输出的一个基本常数表的实例；

- 图7说明了用于执行具有根据本发明的一种对抗方法的DES的一个流程图的第一实例；

- 图8说明了用于执行具有根据本发明的一种对抗方法的DES的一个流程图的第二实例。

尽管以下的描述选择了DES，但是本发明的对抗不特指DES，而且能够同样地应用到其它密钥算法（诸如DES、IDEA、AES、FEAL、三元DES、BlowFish、SAFER、SHA－MAC、RIPEMD、DFC、RC5、RC6或SEAL）

中；在本公开的剩余部分中，通常情况将因此被视为一种算法 A（M，K）（最初易受 DPA 影响的），其中 K 是一个具有长度 k 的密钥，M 为该信息.

该 DES 密钥加密算法（在下文中术语 DES 将更简单地用于表示 DES 算法）包含 16 个计算循环，用 T1 到 T16 表示，如图 3 和 4 中所示.

该 DES 从在该输入信息 M 上的一个初始置换 IP 开始（图 3）. 输入信息 M 是一个 64 位的字 f. 在置换之后，获得一个 64 位的字 e，它被分成两个以便形成第一循环（T1）的输入参数 L0 和 R0. L0 是一个 32 位的字 d，包含该字 e 的最高 32 个有效位. R0 是一个 32 位的字 h，包含该字 e 的最低 32 个有效位.

密钥 K 是一个 64 位的字 q，本身进行一个置换和压缩以便提供一个 56 位的字 r.

第一个循环包含一个在参数 R0 上、包含一个扩充和一个置换的操作 EXP PERM，以便提供一个 48 位的字 1 作为一个输出.

这个字 1 与一个参数 K1 在一个表示 XOR 的异或类型的操作中结合，以便提供一个 48 位的字 b. 这个参数 K1 是一个 48 位的字 m，是从字 r 处通过移位一个位置（在图 3 和 4 中用 SHIFT 表示的操作）、继之以一个置换和压缩（用 COMP PERM 表示的操作）而获得的.

字 b 被用于一个表示为 SBOX 的操作，在其输出端获得一个 32 位的字 a. 这个特定操作将就图 5 和 6 进行更详细地说明.

字 a 进行一个置换 P PERM，产生 32 位的字 c 作为一个输出.

这个字 c 在一个 XOR 表示的异或类型的逻辑运算中与第一个循环 T1 的输入参数 L0 结合，该逻辑运算提供了 32 位的字 g 作为一个输出.

第一个循环的字 h（= R0）提供了随后循环（T2）的输入参数 L1，且第一个循环的字 g 提供了随后循环的输入参数 R1. 第一个循环的字 p 提供了随后循环的输入 r.

除有关根据所讨论的这些循环在一个或两个位置上进行的转移操作 SHIFT 之外，其它循环 T2 到 T16 以相似的方式进行.

这样每一循环 Ti 接受参数 Li-1、Ri-1 和 r 作为一个输入，并提供参数 Li、Ri 和 r 作为一个输出用于随后的循环 Ti+1.

在 DES 算法的末端（图 4），加密信息从由最后一个循环 T16 提供的参数 L16 和 R16 计算出来.

该加密信息C的这个计算实际上包含以下操作:

- 通过将字L16和r16的位置反向、然后连接它们形成一个64位字e';

- 应用与该DES开始置换可逆的置换IP^{-1}，以便获得形成该加密信息C的64位字f'.

在图5和6中详细介绍了操作SBOX. 它包含一个常数表TC_0以便提供一个输出数据项a作为一个输入数据项b的函数.

实际上，这个常数表TCo是基本常数$TC_0$1到$TC_0$8的八个表形式，其中每个仅接收该字b的6位作为一个输入，以便仅仅提供该字a的4位作为一个输出.

因此在图6中描绘的基本常数$TC_0$1的表接收字b的位b1到b6作为一个输入数据项，并提供字a的位a1到a4作为一个输出数据项.

实际上，基本常数$TC_0$1到$TC_0$8的这八个表被保存在该电子元件的程序存储器中.

在第一个循环T1的操作SBOX中，从该常数表TC_0输出的数据项a的一个特定位仅仅取决于用作一个输入数据项b的6位，即仅仅取决于该密钥K的6位和输入信息(M).

在最后一个循环T16的操作SBOX中，从常数表TC_0输出的数据项a的一个特定位能够仅仅从该密钥K的6位和该加密信息(C)中重新计算.

然而，如果该DPA攻击的原则被再次接受时，如果该输出数据项a的一位被选择用作一个目标位，则它足以在该密钥K的6位上形成一个假设，以便推算用于一个给定输入信息(M)或输出信息(C)的一个目标位的值. 换句话说，就DES来说，它足以形成在一个6位子密钥上的一个假设.

在一个DPA攻击中就这样一个用于一个给定目标位的算法来说，因此在64个可能的子密钥当中仅仅需要判定一个子密钥假设.

因此,通过使用仅仅字a的八位作为目标位(基本常数TCO1到TCO8的每个表一个输出位)，就有可能通过在这些目标位中的每一个上进行DPA攻击发现该密钥的高达6×8=48位.

在DES中，因此可在该算法开始和结束时发现在DPA攻击环境内关键性的指令.

在该 DES 算法开始时，能够从一个输入信息 M 和一个子密钥假设中推算的数据是在第一个循环（T1）中计算的数据 a 和 g.

第一个循环 T1（图 3）的数据项 a 是所讨论循环的操作 SBOX 的输出数据项. 数据项 g 是通过置换（P PERM）以及和输入参数 L0 的异或操作从数据项 a 中计算得到的.

事实上，第一个循环的数据项 c 是从第一个循环的数据项 a 导出的一个数据项. 导出的数据项 c 相当于数据项 a 的位的一个简单置换.

第二个循环的数据项 1 是从第一个循环的数据项 g 导出的一个数据项，是由于它相当于 g 的位的一个置换，字 g 的某些位也被复制了.

已知 a 和 g，就也可能已知这些导出的数据了.

该算法开始的关键性指令是操作能够被推算的数据项、诸如第一个循环的数据项 a 或是一个导出的数据项等的那些关键性指令.

操作第一个循环 T1 的数据项 a 或导出数据项 c 的关键性指令因此是操作 SBOX、操作 P PERM 的结尾处以及第一个循环 T1 的操作 XOR 开始处的指令.

操作数据项 g 或导出数据的关键性指令是所有第一个循环 T1 结尾处 XOR 操作结尾的指令直至第二个循环 T2 操作 SBOX 开始处的指令、以及在第三个循环 T3 末端 XOR 操作开始处的指令（L2 = h(T2) = g(T1)）.

在该 DES 算法的结尾，能够从一个加密信息 C 和一个子密钥假设中推算的数据是第 16 个循环 T16 的数据项 a 和等于第 14 个循环 T14 的字 h 的数据项 L15.

操作第 16 个循环的数据项 a 或导出数据的关键性指令是 SBOX 操作、置换操作 P PERM 的结尾处和 XOR 操作开始处的第 16 个循环的指令.

就数据项 L15 来说，操作这个数据项或导出数据的关键性指令是所有在第 14 个循环 T14 的 XOR 操作结尾处指令之后、直至第 15 个循环 T15 的 SBOX 操作开始处的指令，加上用于开始第 16 个循环 T16XOR 操作的指令.

应用于这个 DES 算法的依据本发明的对抗方法在于：对每条关键性指令来说，该关键性指令操作一个数据项和它的反码具有同样多的可能性. 因此，无论可以在其上进行 DPA 攻击的目标位如何，就操作

这个位的关键性指令来说，操作一个“1”或“0”具有同样多的可能性。

实际上，这对每一个可能的目标位来说必须是正确的：换句话说，攻击者在几个可能的攻击之间、即在几个可能的用于完成他的曲线分类的布尔选择函数之间做出选择，就一个给定的子密钥假设来说，依据本发明的对抗方法的实现必须力图确保由每一条关键性指令操作的数据随机地在两次中一次采用一个值或它的反码。关于依据本发明的对抗方法在 DES 算法中的应用，因此需要将对抗施加到 DES 指令的关键性开始和 DES 指令的关键性结尾上，以便被完全地保护。

在 DES 中，所有由关键性指令操作的数据是一个输出数据项或是从操作 SBOX 的一个输出数据项导出的数据。

事实上，在 DES 开始时，能够被推算的数据是第一个循环 T1 的数据 a 和 g。数据项 a 是第一个循环中 SBOX 操作的输出数据项。数据项 g 是从数据项 a 计算得到的，这是由于 g = P PERM(a) XOR L0。因此 g 是从第一个循环中 SBOX 操作的输出数据项 a 导出的一个数据项。因此所有由 DES 指令关键性开始操作的数据直接或间接地起源于第一个循环中 SBOX 操作的输出数据项 a。

就 DES 的结尾来说，能够被推算的数据是第 16 个循环 T16 的数据项 a 和第 14 个循环 T14 的数据项 g，其中 g 等于 L15。

数据项 a 是第 16 个循环 T16 中 SBOX 操作的输出数据项。

就数据项 L15 而论，这是在 DES 算法正常执行中从第 14 个循环 T14 中 SBOX 操作的输出数据项 a 计算得到的：L15 = P PERM(a) XOR L14。

如果这些特定操作 SBOX 的输出数据 a 是不可推算的，则所有导出的数据也是不可推算的：所有由该 DES 算法的关键性指令操作的数据因此也是不可推算的。如果认为这些 SBOX 操作构成了用于从一个输入数据项 E = b 提供一个输出数据项 S = a 的第一装置，则应用于 DES 算法的对抗方法包含了使用其它用于使输出数据项不可推算的装置，从而使由关键性指令操作的这个输出数据项和/或导出的数据都是不可推算的。

依据本发明，形成了由至少最初 3 个循环形成的一组和由至少最后 3 个循环形成的另一组。这些组因此包含了所有包含关键性指令的循环。

在所有循环中使用第一装置的第一序列和在至少某些循环中使用

其它装置的第二序列与这两个组有关.

在其它不在这些组中的循环中，有可能继续使用第一装置.

这样使用这些其它装置以使输出的结果即加密信息保持正确.

这些其它装置能够包含几个不同的装置. 它们是使反码的数据项相应于在第一装置的输入和输出数据当中的一个或其它数据项的那些装置.

因此，考虑到大数量的执行，这些组将平均在两次中一次使用第一序列，一次使用另一个序列，其中第一序列是该算法的正常序列. 对应于某些中间结果，由这些组中的关键性指令操作的数据因此将平均在两次中有一次是反码. 因此就大量曲线从统计上来说一个给定目标位为 1 或 0 有同样多的可能性.

依据图 7，它描绘了用于执行利用依据本发明的对抗方法的 DES 的流程图的第一实例，元件 T（终端）和 C（卡）的通信取决于依据如下所述的步骤 1 到 6 的信号交换:

1. C 在它的非易失性存储器（例如 EEPROM）中递增 i;

2. C 产生会话密钥 K[i] = K[i-1]^2 mod z;

3. C 从它的非易失性存储器（例如 EEPROM）中清除密钥 K[i-1]，并在它的位置输入 K[i];

4. C 传递 i 到 T;

5. T 计算 K[i] = K[0]^(2^i) mod z，其中数 2^i 是计算的模 phi(z);

6. C 和 T 使用 K[i]开始一个加密计算.

或者是，依据图 8，它描绘了用于执行利用依据本发明的一种对抗方法的 DES 的流程图的第二实例，在 C 和 T 之间的通信取决于依据如下所述的步骤 1 到 6 的信号交换:

1. C 在它的非易失性存储器（例如 EEPROM）中递增 i;

2. C 产生会话密钥 K[i] = K[i-1]*c mod z;

3. C 从它的非易失性存储器（例如 EEPROM）中清除密钥 K[i-1]，并在它的位置输入 K[i];

4. C 传递 i 到 T;

5. T 计算 K[i] = K[0]*(c^i) mod z;

6. C 和 T 使用 K[i]开始一个加密计算.

用于数 z 的最优选择是具有长度 k 的最小质数. 尤其是:

k = K 的位长度	z 值
56	2^k —5
64	2^k — 59
80	2^k — 65
96	2^k —17
128	2^k — 159
256	2^k — 189

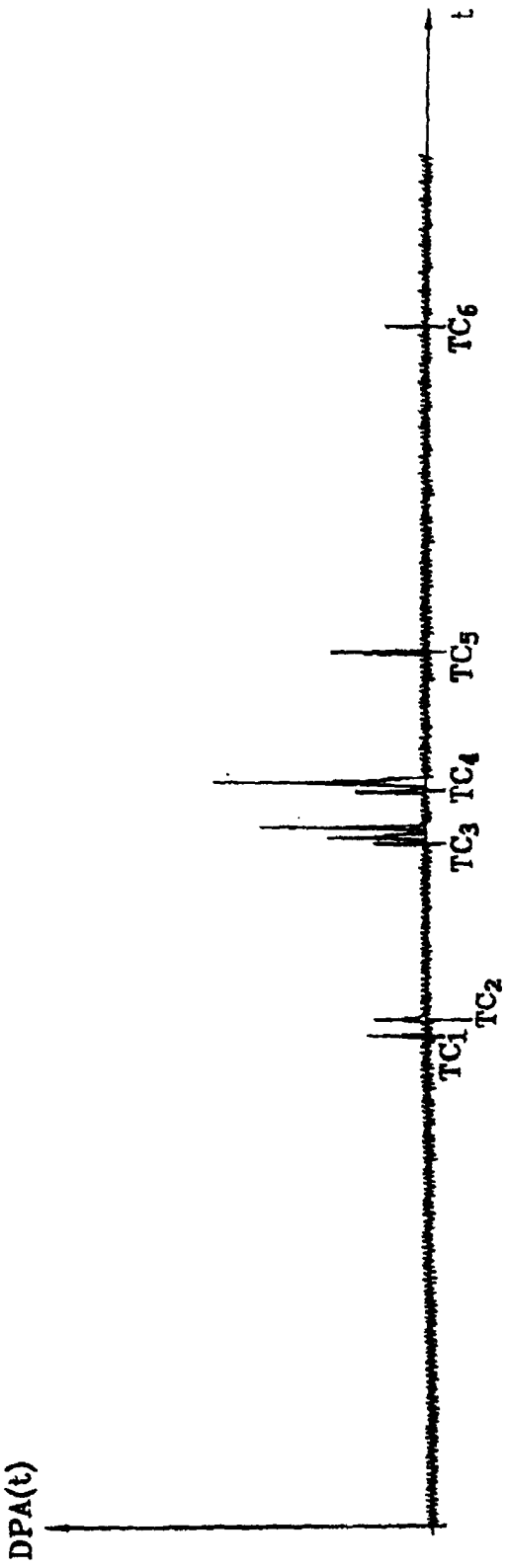


图 1

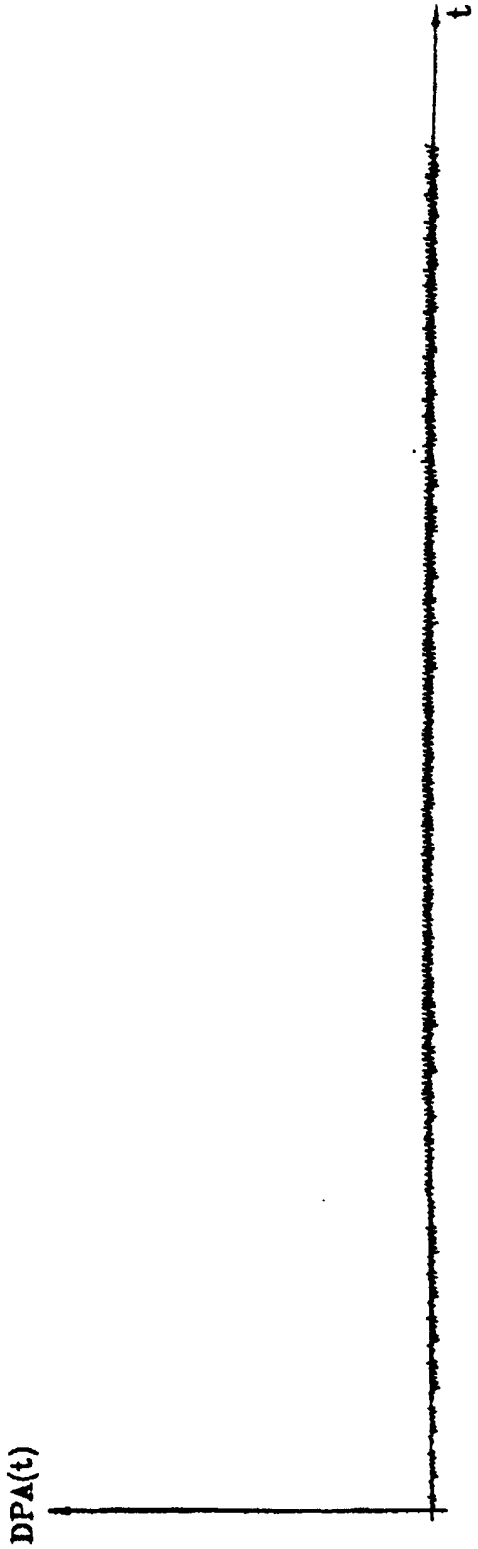


图 2

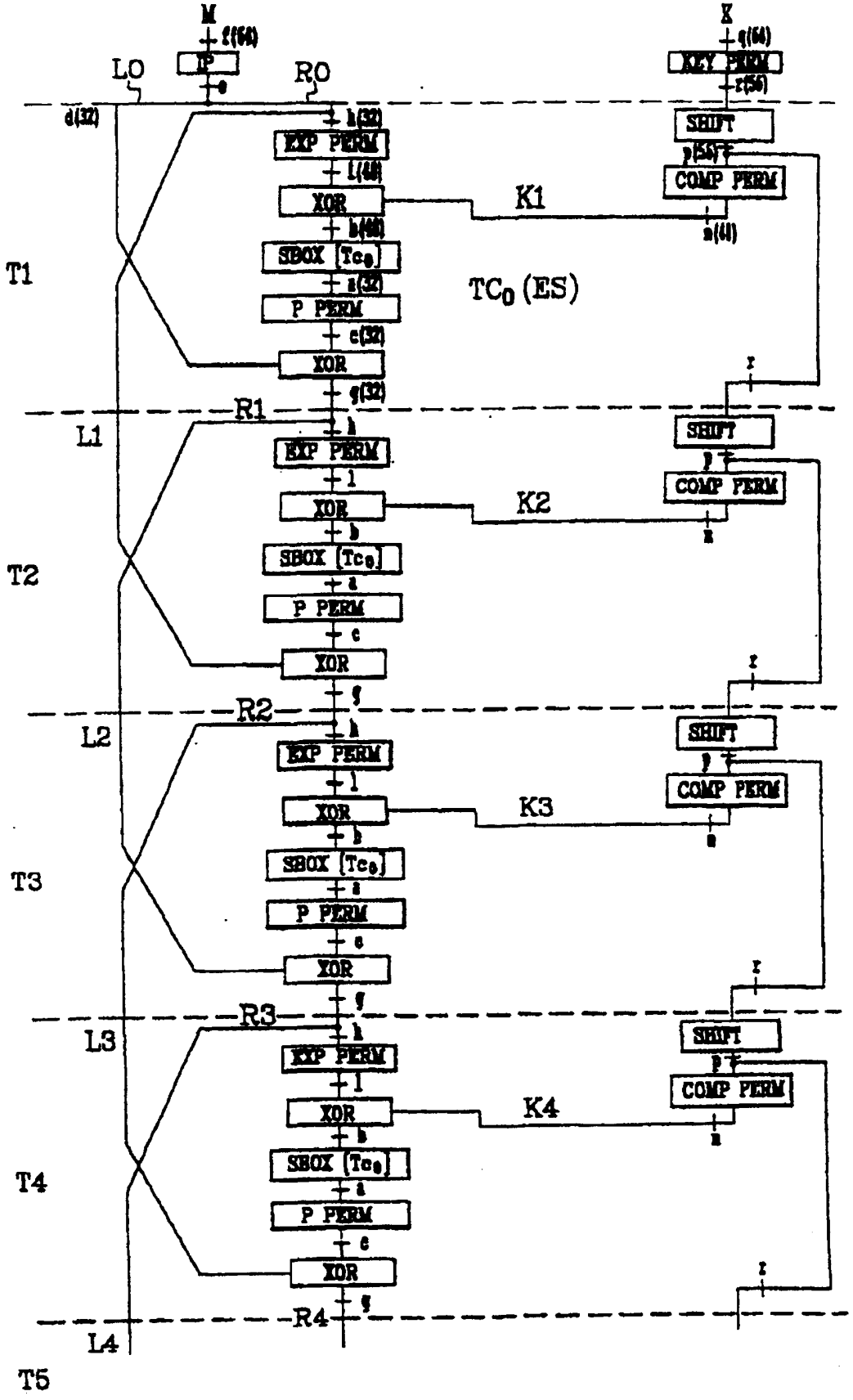

M
IP
LO
RO
K
KEY PERM
SHIFT
COMP PERM
EXP PERM
XOR
SBOX [Tc0]
P PERM
K1
K2
K3
K4
TC0 (ES)
R1
R2
R3
R4
L1
L2
L3
L4
T1
T2
T3
T4
T5

图 3

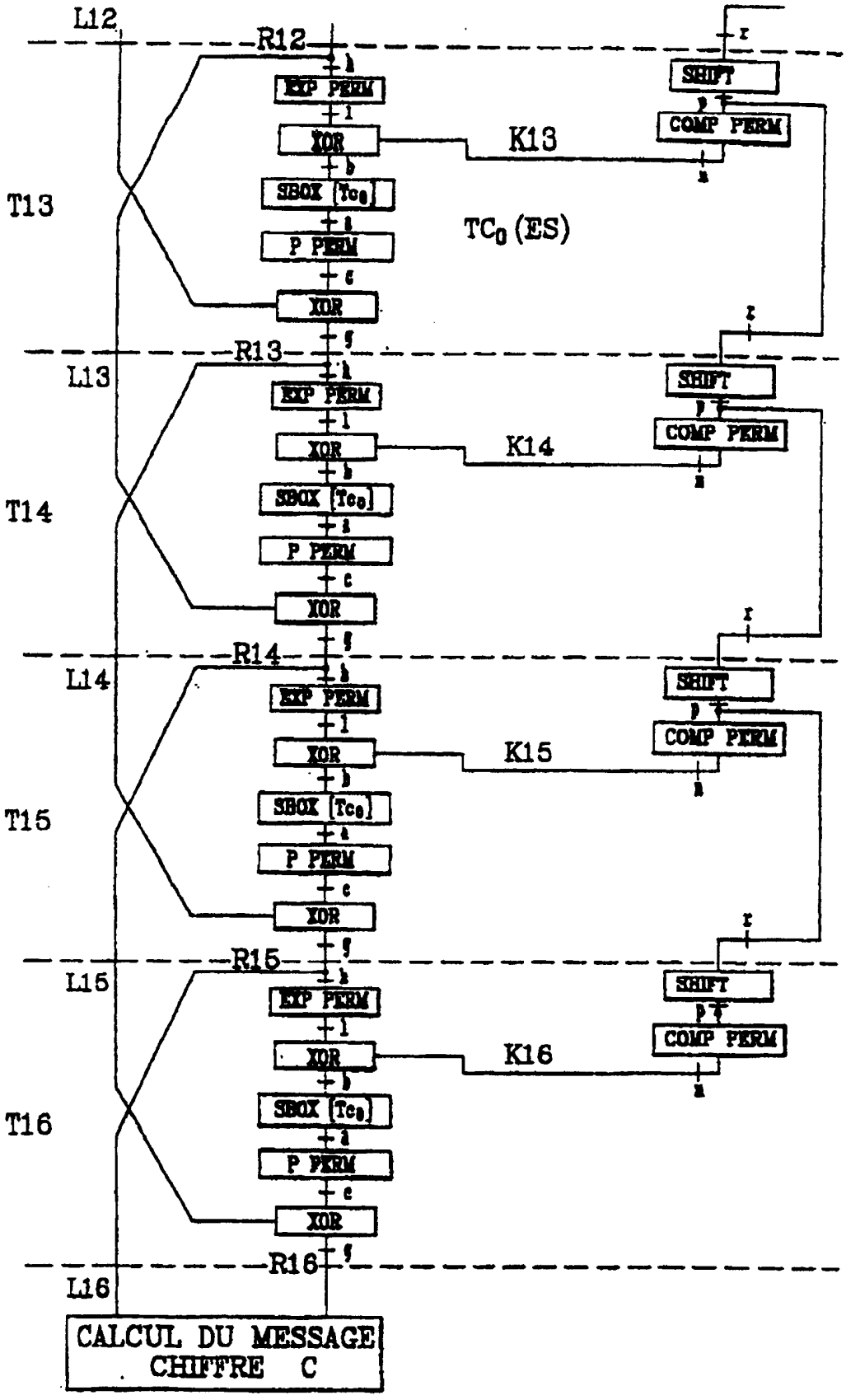

L12
R12
EXP PERM
XOR
K13
SBOX [Tc0]
P PERM
XOR
SHIFT
COMP PERM
T13
TC0 (ES)
L13
R13
EXP PERM
XOR
K14
SBOX [Tc0]
P PERM
XOR
SHIFT
COMP PERM
T14
L14
R14
EXP PERM
XOR
K15
SBOX [Tc0]
P PERM
XOR
SHIFT
COMP PERM
T15
L15
R15
EXP PERM
XOR
K16
SBOX [Tc0]
P PERM
XOR
SHIFT
COMP PERM
T16
L16
R16
CALCUL DU MESSAGE
CHIFFRE C

图 4

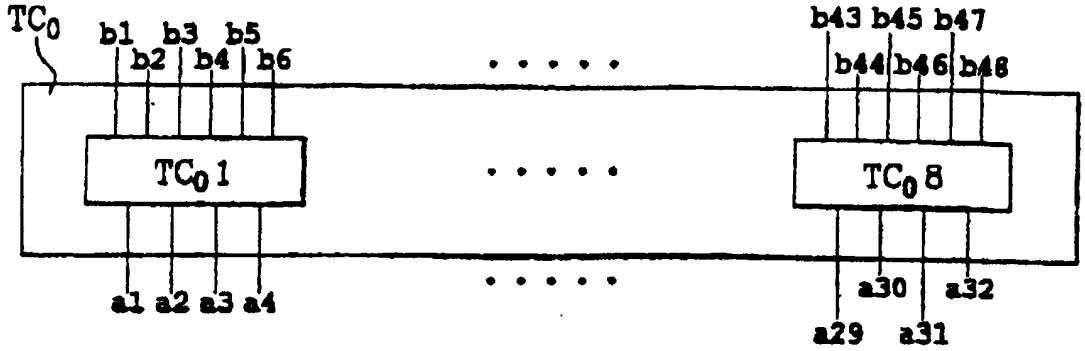


图 5

TC0 1

E=b1b2b3b4b5b6	S=a1a2a3a4
000000	1101
000001	0101
⋮	⋮
111111	1010

图 6

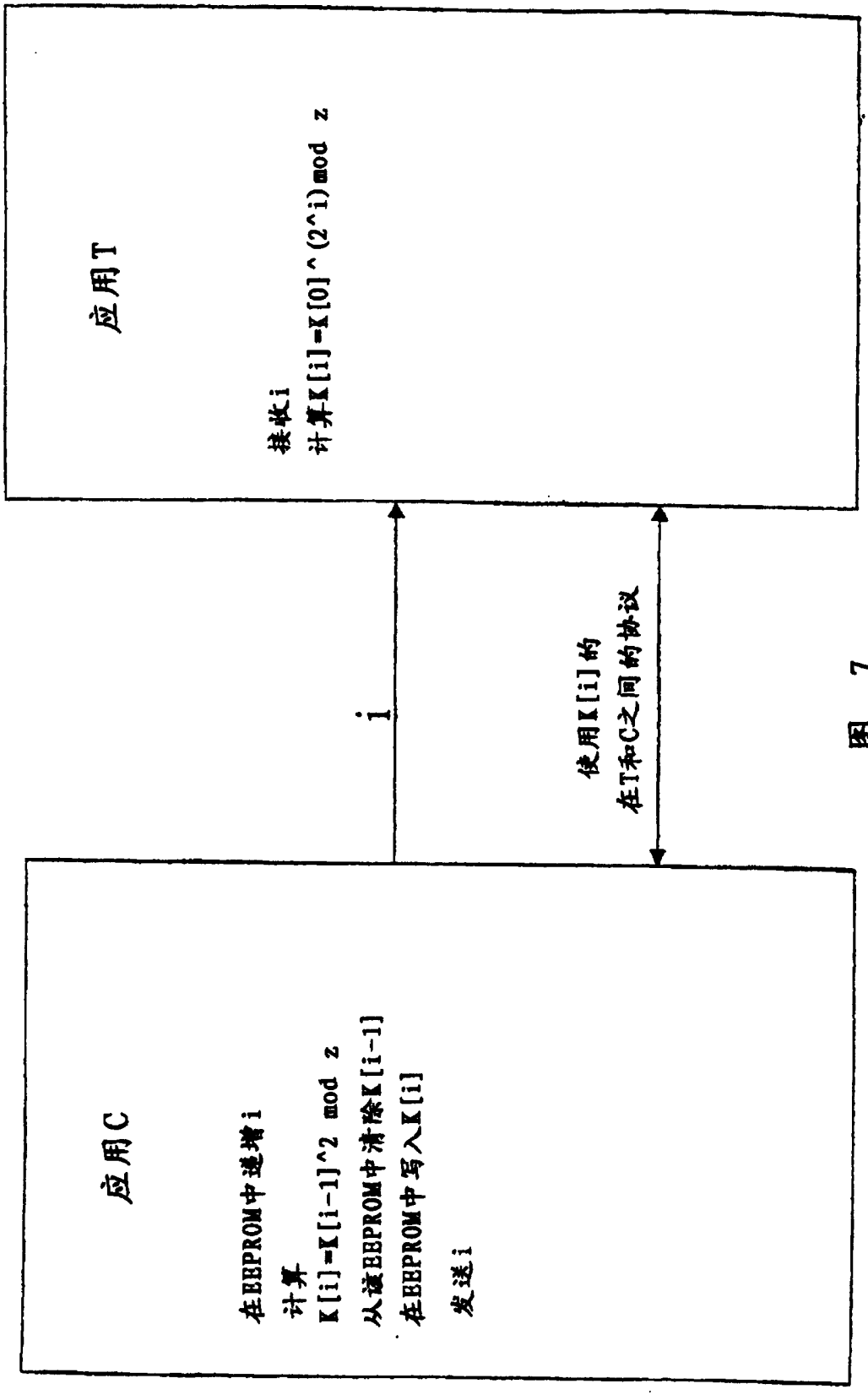

应用T
接收i
计算K[i]=K[0]^(2^i)mod z
i
使用K[i]的
在T和C之间的协议
应用C
在EEPROM中递增i
计算
K[i]=K[i-1]^2 mod z
从该EEPROM中清除K[i-1]
在EEPROM中写入K[i]
发送i

图 7

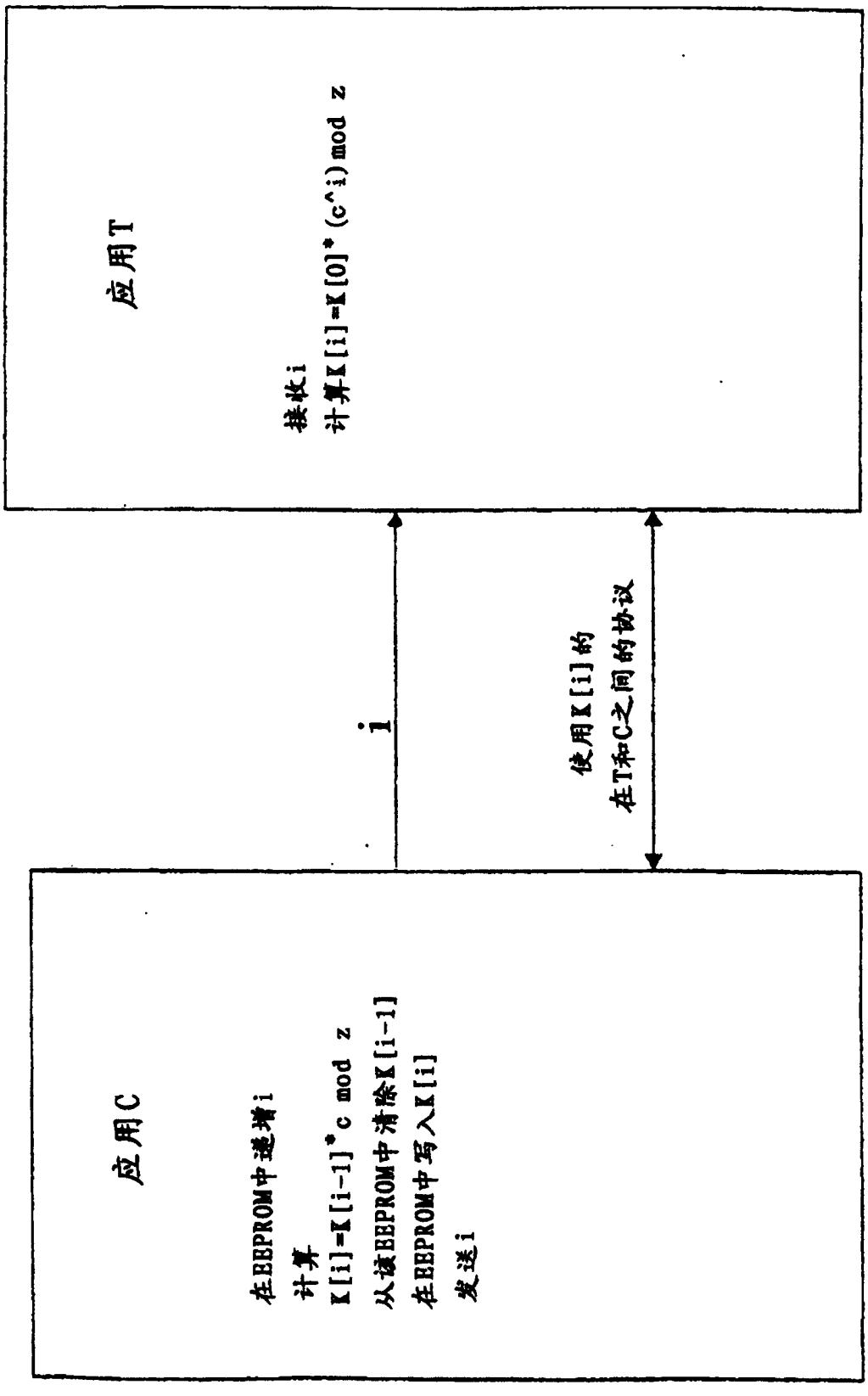

应用T
接收i
计算K[i]=K[0]*(c^i) mod z
i
使用K[i]的
在T和C之间的协议
应用C
在EEPROM中递增i
计算
K[i]=K[i-1]*c mod z
从该EEPROM中清除K[i-1]
在EEPROM中写入K[i]
发送i

图 8