



## (12) 发明专利

(10) 授权公告号 CN 1890914 B

(45) 授权公告日 2011.07.27

(21) 申请号 200480036264.7

(22) 申请日 2004.11.30

(30) 优先权数据

03104643.6 2003.12.11 EP

(85) PCT申请进入国家阶段日

2006.06.06

(86) PCT申请的申请数据

PCT/IB2004/052607 2004.11.30

(87) PCT申请的公布数据

W02005/060147 EN 2005.06.30

(73) 专利权人 耶德托公司

地址 荷兰霍夫多普

(72) 发明人 P·M·H·M·A·戈里森

J·A·特雷斯彻 A·A·M·斯塔林

W·C·马龙 M·A·特雷弗斯

(74) 专利代理机构 中国国际贸易促进委员会专  
利商标事务所 11038

代理人 屠长存

(51) Int. Cl.

H04L 9/06(2006.01)

(56) 对比文件

US 6377687 B1, 2002.04.23, 全文.

US 4751733, 1988.06.14, 全文.

张文涛等. 一类基于混沌函数的分组密码的安全性评估. 软件学报 2003年第3期. 2003, (2003年第3期), 512-517.

审查员 商晓莉

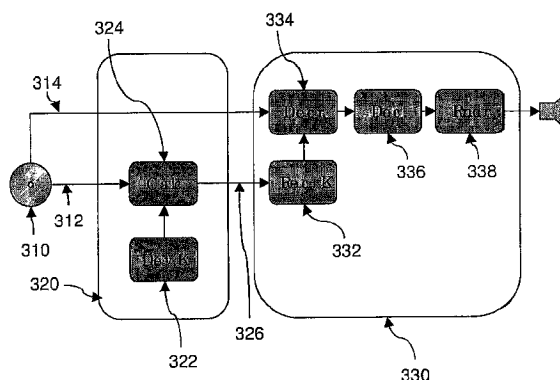
权利要求书 4 页 说明书 9 页 附图 8 页

### (54) 发明名称

块加密系统、利用置换隐藏各加密轮的核心加密函数

### (57) 摘要

在系统(600)中,服务器(610)以模糊形式向执行装置(620)提供数字信号处理函数 $f$ 。函数 $f$ 包括信号处理函数 $f_i$ 的函数级联,对于公式(I), $1 \leq i \leq N$ 。服务器包括处理器(612),用于选择 $2N$ 个可逆置换 $p_i$ 的集合,其中 $1 \leq i \leq 2N$ ;计算 $N$ 个函数 $g_i$ 的集合,其中 $g_i$ 在功能上相当于公式(II),其中 $1 \leq i \leq N$ ;以及计算 $N-1$ 个函数 $h_i$ 的集合,其中 $h_i$ 在功能上相当于公式(III),其中 $2 \leq i \leq N$ 。服务器包括:部件(614),用于为执行装置配备包括公式(IV)的执行装置函数级联,其中 $y_1, \dots, y_N$ 是公式(V)的函数参数;以及部件(616),用于向执行装置提供函数 $g_1, \dots, g_N$ 。执行装置包括:部件(626),用于获得函数 $g_1, \dots, g_N$ ;以及处理器(622),用于加载执行装置函数级联并且对函数 $g_1, \dots, g_N$ 应用加载的执行装置函数级联(例如 $ED_j(g_1, \dots, g_N)$ )。



1. 一种以模糊的形式向执行装置提供数字信号处理函数  $f$  的方法, 所述函数  $f$  包括函数级联, 所述函数级联包括  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ , 其中  $f_i$  是多个信号处理函数,  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出, 所述方法包括:

选择  $2N$  个可逆置换  $p_i$  的集合,  $1 \leq i \leq N$ ;

计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

计算  $N-1$  个函数  $h_i$  的集合, 其中  $h_i$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{2i-2}$ , 其中  $2 \leq i \leq N$ ;

为所述执行装置配备执行装置函数级联, 所述执行装置函数级联包括  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ , 其中  $y_1, \dots, y_N$  是函数参数; 以及

向所述执行装置提供函数  $g_1, \dots, g_N$ , 其中所述函数  $g_1, \dots, g_N$  在所述执行装置中被应用所述执行装置函数级联  $ED_1(g_1, \dots, g_N)$ 。

2. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 还包括在所述执行装置中, 对所述函数  $g_1, \dots, g_N$  应用所述执行装置函数级联  $ED_1(g_1, \dots, g_N)$ 。

3. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 其中所述执行装置函数级联  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  被  $ED_2(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$  替代。

4. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 其中所述函数级联开始于另一个信号处理函数  $f_0$ , 即  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$  被  $FC_2(x) \equiv f_N \circ \dots \circ f_1 \circ f_0(x)$  替代, 并且所述执行装置函数级联  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  被  $ED_3(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$  替代, 其中  $S_1$  在功能上相当于  $p_1^{-1} \circ f_0$ 。

5. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 其中所述执行装置函数级联  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  被  $ED_4(y_1, \dots, y_N) \equiv p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  替代。

6. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 其中所述函数级联结束于另一个信号处理函数  $f_{N+1}$ , 即  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$  被  $FC_3(x) \equiv f_{N+1} \circ f_N \circ \dots \circ f_1(x)$  替代, 并且所述执行装置函数级联  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  被  $ED_5(y_1, \dots, y_N) \equiv S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  替代, 其中  $S_2$  在功能上相当于  $f_{N+1} \circ p_{2N}$ 。

7. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 包括获得所述执行装置和/或所述执行装置的用户唯一身份,  $2N$  个可逆置换  $p_i$  的集合和/或序列对于所获得的身份是唯一的。

8. 如权利要求 1 所述的提供数字信号处理函数  $f$  的方法, 其中为所述执行装置配备所述执行装置函数级联的步骤包括: 提供嵌入在软件程序中的执行装置函数级联, 以便由所述执行装置中的处理器执行。

9. 如权利要求 8 所述的提供数字信号处理函数  $f$  的方法, 其中向所述执行装置提供所述函数  $g_1, \dots, g_N$  的步骤包括: 以用于所述程序的插件程序的形式提供所述函数  $g_1, \dots, g_N$ 。

10. 如权利要求 8 所述的提供数字信号处理函数  $f$  的方法, 其中向所述执行装置提供所述函数  $g_1, \dots, g_N$  的步骤包括: 将所述函数  $g_1, \dots, g_N$  嵌入在所述软件程序中。

11. 一种以模糊的形式向执行装置提供数字信号处理函数  $f$  的设备, 所述函数  $f$  包

括函数级联,所述函数级联包括  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ , 其中  $f_i$  是多个信号处理函数,  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出, 该设备包括:

用于选择  $2N$  个可逆置换  $p_i$  的集合的部件,  $1 \leq i \leq 2N$ ;

用于计算  $N$  个函数  $g_i$  的集合的部件, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ; 以及

用于计算  $N-1$  个函数  $h_i$  的集合的部件, 其中  $h_i$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{2i-2}$ , 其中  $2 \leq i \leq N$ ; 以及

用于为所述执行装置配备执行装置函数级联的部件, 所述执行装置函数级联包括  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ , 其中  $y_1, \dots, y_N$  是函数参数; 以及

用于向所述执行装置提供函数  $g_1, \dots, g_N$  的部件。

12. 一种执行装置的执行方法, 包括:

从如权利要求 11 所述的设备获得所述函数  $g_1, \dots, g_N$ ;

对所述函数  $g_1, \dots, g_N$  应用由如权利要求 11 所述的设备为所述执行装置配备的所述执行装置函数级联  $ED_1(g_1, \dots, g_N)$ ; 以及

将所应用的执行装置函数级联应用于所述数字信号输入  $x$ 。

13. 一种用于应用执行装置函数级联的设备, 包括:

用于对从如权利要求 11 所述的设备获得的所述函数  $g_1, \dots, g_N$  应用由如权利要求 11 所述的设备为所述执行装置配备的所述执行装置函数级联  $ED_1(g_1, \dots, g_N)$  的部件; 以及

用于将所应用的执行装置函数级联应用于所述数字信号输入  $x$  的部件。

14. 一种以模糊的、匿名的形式向多个执行装置提供数字信号处理函数  $f$  的方法, 各个执行装置利用唯一下标  $j$  标识, 所述函数  $f$  包括函数级联, 所述函数级联包括  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ , 其中  $f_i$  是多个信号处理函数,  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出, 所述方法包括:

选择  $2N$  个可逆置换  $p_i$  的集合, 其中  $1 \leq i \leq N$ ;

计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

为各个执行装置  $j$  选择相应的  $2N$  个可逆置换  $p_{j,i}$  的集合和 / 或序列, 其对于所述执行装置和 / 或所述执行装置的用户是唯一的;

为各个执行装置  $j$  计算  $N-1$  个函数  $h_{j,i}$  的相应集合, 其中  $h_{j,i}$  在功能上相当于  $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ , 其中  $2 \leq i \leq N$ ;

为各个执行装置  $j$  配备各自的执行装置函数级联, 所述执行装置函数级联包括  $ED_j(y_1, \dots, y_N) \equiv y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$ ;

为各个执行装置  $j$  配备各自的加载程序函数  $loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$ , 其中  $l_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{2i}$ , 并且  $r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ ; 以及

向所述执行装置提供函数  $g_1, \dots, g_N$ ,

其中, 所述函数  $g_1, \dots, g_N$  在所述执行装置中被应用  $ED_j(loader_j(g_1, \dots, g_N))$ 。

15. 如权利要求 14 所述的提供数字信号处理函数  $f$  的方法, 还包括在所述执行装置中,

执行  $ED_j(loader_j(g_1, \dots, g_N))$ 。

16. 如权利要求 14 所述的提供数字信号处理函数  $f$  的方法, 包括:

通过广播和 / 或在对于各个执行装置具有相同内容的存储介质上的分配来向各个执行装置提供  $g_1, \dots, g_N$ 。

17. 如权利要求 16 所述的提供数字信号处理函数  $f$  的方法, 还包括: 通过广播和 / 或在对于各个执行装置具有相同内容的存储介质上的分配来向各个执行装置提供所述数字信号输入  $x$ 。

18. 如权利要求 14 所述的提供数字信号处理函数  $f$  的方法, 包括:

通过一对一通信信道和 / 或具有装置特定内容的存储介质向执行装置  $j$  提供以下相应函数的集合中的至少一个:  $h_{j,i}$ 、 $l_{j,i}$  或  $r_{j,i}$ 。

19. 如权利要求 1 或 14 所述的提供数字信号处理函数  $f$  的方法, 其中所述函数  $f$  是基于 Feistel 密码网络的解密函数, 并且每一个所述信号处理函数  $f_i$  是各自的 Feistel 解密轮函数。

20. 如权利要求 19 所述的提供数字信号处理函数  $f$  的方法, 其中每一个所述置换  $p_i$  是 Feistel 变换器, 其中如果存在可逆函数  $Q_x$  和  $Q_y$  并且  $Q(\langle x, y \rangle) = \langle Q_x(x), Q_y(y) \rangle$ , 其中  $Q_x(x) \oplus Q_x(y) = Q_x(x \oplus y)$ , 并且  $Q_y(x) \oplus Q_y(y) = Q_y(x \oplus y)$ , 则在连续对  $\langle x, y \rangle$  上操作的函数  $Q$  是 Feistel 变换器。

21. 一种以模糊的、匿名的形式向多个执行装置提供数字信号处理函数  $f$  的设备, 其中各个执行装置利用唯一下标  $j$  标识, 所述函数  $f$  包括函数级联, 所述函数级联包括  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ , 其中  $f_i$  是多个信号处理函数,  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出, 该设备包括:

用于选择  $2N$  个可逆置换  $p_i$  的集合的部件, 其中  $1 \leq i \leq 2N$ ;

用于计算  $N$  个函数  $g_i$  的集合的部件, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

用于为各个执行装置  $j$  选择对于所述执行装置和 / 或所述执行装置的用户是唯一的  $2N$  个可逆置换  $p_{j,i}$  的相应集合和 / 或序列的部件;

用于为各个执行装置  $j$  计算  $N-1$  个函数  $h_{j,i}$  的相应集合的部件, 其中  $h_{j,i}$  在功能上相当于  $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ , 其中  $2 \leq i \leq N$ ;

用于为各个执行装置  $j$  配备各自的执行装置函数级联的部件, 所述执行装置函数级联包括  $ED_j(y_1, \dots, y_N) \equiv y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$ ;

用于为各个执行装置  $j$  配备各自的加载程序函数  $loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$  的部件, 其中  $l_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{j,2i}$ , 并且  $r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ ; 以及

用于向所述执行装置提供函数  $g_1, \dots, g_N$  的部件。

22. 一种执行装置的执行方法, 其中所述执行装置利用唯一下标  $j$  标识, 该方法包括:

从如权利要求 21 所述的设备获得所述函数  $g_1, \dots, g_N$ ;

加载由如权利要求 21 所述的设备为所述执行装置配备的所述执行装置函数级联, 所述执行装置函数级联对于所述执行装置是唯一的并包括  $ED_j(y_1, \dots, y_N) \equiv y_N \circ h_{j,N}$ ,

$_N \circ y_{N-1} \circ h_{j,N-1} \circ \cdots \circ y_1$ , 其中  $y_1, \cdots, y_N$  是函数参数;

加载由如权利要求 21 所述的设备为所述执行装置配备的所述加载程序函数  $\text{loader}_j(x_1, \cdots, x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1}, \cdots, l_{j,N} \circ x_N \circ r_{j,N})$ ;

对函数  $g_1, \cdots, g_N$  的集合应用所述加载程序函数, 产生函数  $g_{j,1}, \cdots, g_{j,N}$  的集合; 以及对所述函数  $g_{j,1}, \cdots, g_{j,N}$  的集合应用所述执行装置函数级联  $\text{ED}_j(g_{j,1}, \cdots, g_{j,N})$ ,

其中:

$g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

$p_i$  是可逆置换, 其中  $1 \leq i \leq N$ ;

$h_{j,i}$  在功能上相当于  $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ , 其中  $2 \leq i \leq N$ ;

$l_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{2i}$ ;

$r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ ; 以及

$p_{j,i}$  是可逆置换, 其中  $1 \leq i \leq 2N$ , 其对于所述装置和 / 或所述装置的用户是唯一的。

23. 一种用于应用执行装置函数级联的设备, 其中所述执行装置利用唯一下标  $j$  标识, 并且包括:

用于加载由如权利要求 21 所述的设备为所述执行装置配备的所述执行装置函数级联的部件, 所述执行装置函数级联对于所述执行装置是唯一的并包括  $\text{ED}_j(y_1, \cdots, y_N) \equiv y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \cdots \circ y_1$ , 其中  $y_1, \cdots, y_N$  是函数参数;

用于加载由如权利要求 21 所述的设备为所述执行装置配备的所述加载程序函数  $\text{loader}_j(x_1, \cdots, x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1}, \cdots, l_{j,N} \circ x_N \circ r_{j,N})$  的部件;

用于对从如权利要求 21 所述的设备获得的函数  $g_1, \cdots, g_N$  的集合应用所述加载程序函数的部件, 产生函数  $g_{j,1}, \cdots, g_{j,N}$  的集合; 以及

用于对所述函数  $g_{j,1}, \cdots, g_{j,N}$  的集合应用所述执行装置函数级联  $\text{ED}_j(g_{j,1}, \cdots, g_{j,N})$  的部件,

其中:

$g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

$p_i$  是可逆置换, 其中  $1 \leq i \leq N$ ;

$h_{j,i}$  在功能上相当于  $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ , 其中  $2 \leq i \leq N$ ;

$l_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{2i}$ ;

$r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ ; 以及

$p_{j,i}$  是可逆置换, 其中  $1 \leq i \leq 2N$ , 其对于所述装置和 / 或所述装置的用户是唯一的。

## 块加密系统、利用置换隐藏各加密轮的核心加密函数

### 技术领域

[0001] 本发明涉及以安全和 / 或个性化方式向执行装置提供级联信号处理函数的方法。本发明还涉及用于以安全和 / 或个性化方式向执行装置提供级联信号处理函数的系统。本发明还涉及一种执行装置,用于执行以安全和 / 或个性化方式提供的级联信号处理函数。

### 背景技术

[0002] 因特网向用户提供了对数字内容的便利且无所不在的访问。由于因特网作为强大的分配渠道的潜能,许多 CE 产品力图与 PC 平台—因特网的主要入口进行互操作。利用因特网作为已取得版权内容的分配介质产生了对保证内容供应商的利益安全的强制挑战。具体地说,需要保证内容供应商的版权和商业模型。控制重放软件是强制实施内容所有者利益的一种方式,包括可用内容下的条款。具体地说,对于 PC 平台,必须假定用户能够完全控制提供对内容和无限量时间和资源的访问的硬件和软件,以攻击和旁路任何内容保护机制。因此,内容供应商必须通过对并非所有用户都能够被信赖的共同体敌对网络传递内容到合法用户。在数字权利管理中用于保护分配给 PC 的内容的一般方法是,对数字内容进行加密(例如用 DES)并在 PC 硬盘上的所谓许可证数据库中存储解密密钥(或“许可证”)。PC 上的数字内容一般利用媒体播放器再现,比如 Microsoft 的媒体播放器、Real 的 RealOne 播放器、苹果的 QuickTime 播放器。这种播放器能够为特定内容格式加载用于执行格式特定解码的相应插件程序。那些内容格式可包括 AVI、DV、运动 JPEG、MPEG-1、MPEG-2、MPEG-4、WMV、音频 CD、MP3、WMA、WAV、AIFF/AIFC、AU 等等。图 1 中图解了播放器和插件程序结构,其中媒体播放器 100 包括核心播放器 100 和几个格式特定的插件程序(显示的是插件程序 120、122 和 124)。核心播放器 100 可以例如提供用于控制播放器的用户界面。各个插件程序包括相应的解码器。它可以直接发送解码内容以再现 HW/SW,比如声卡,或将其传递到核心播放器 100 进行进一步处理。为了安全再现,利用安全插件程序,它不仅解码特定格式下的内容,而且对内容解密。这一点在图 2 中图解了,其中首先通过解密器 230 传递加密内容,然后所解密的内容通过格式特定解码器 220 传递。解密器 230 可从许可证数据库 210 接收解密密钥 / 许可证。

[0003] 依赖于加密的数字权利管理的最大弱点在于密钥分配和处理。为了重放,软件播放器必须从许可证数据库检索解密密钥,然后它必须在存储器的某处存储此解密密钥,以便对加密内容进行解密。这给攻击者留下两个选择来攻击在软件播放器中处理的密钥:第一,对许可证数据库访问函数的逆向工程能够导致黑盒软件(即,攻击者并非必须了解软件函数的内部工作),黑盒软件能够从所有许可证数据库检索资产(asset)密钥。第二,通过观测对在内容解密期间使用的存储器的访问,有可能检索资产密钥。

[0004] 一般地,数字权利管理系统根据块密码使用加密技术,块密码利用称为轮的加密 / 解密步骤序列来处理块中的数据流。第  $i-1$  轮的输出是第  $i$  轮的输入。因此,对于具有  $N$  轮的系统,算法能够被描述为函数级联  $f_N \circ \dots \circ f_1(x)$ ,其中函数  $f_i$  表示  $f$  轮的函数性。大部分的块算法是 Feistel 网络。在这种网络中,均匀长度  $n$  的输入数据块  $x$  被分成长度为

$n/2$  的两个等分部分,通常称为 L 和 R。因此,提供给第一轮输入  $x$  得到  $x = \langle L_0, R_0 \rangle$ 。第  $i$  轮 ( $i > 0$ ) 执行函数  $f_i$ , 其中  $f_i$  定义为:  $f_i(\langle L_{i-1}, R_{i-1} \rangle) = \langle R_{i-1}, (L_{i-1} \oplus F(R_{i-1}, K_i)) \rangle$ ,  $K_i$  是用在第  $i$  轮中的子密钥, 并且  $F$  是任意轮函数。

## 发明内容

[0005] 本发明的目的是提供对诸如 Feistel 网络的级联信号处理函数的更好保护。

[0006] 为了满足本发明的目的,以模糊形式向执行装置提供数字信号处理函数  $f$  的方法,其中函数  $f$  包括一个函数级联,该函数级联包括多个信号处理函数  $f_i, 1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出 (例如,  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ ), 所述方法包括:

[0007] 选择  $2N$  个可逆置换  $p_i$  的集合,  $1 \leq i \leq 2N$ ;

[0008] 计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

[0009] 计算  $N-1$  个函数  $h_i$  的集合, 其中  $h_i$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{2i-2}$ , 其中  $2 \leq i \leq N$ ;

[0010] 为所述执行装置配备执行装置函数级联, 该函数级联包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ , 其中  $y_1, \dots, y_N$  是函数参数 (例如,  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ ),

[0011] 向执行装置提供函数  $g_1, \dots, g_N$ ; 以及

[0012] 在执行装置中, 对函数  $g_1, \dots, g_N$  (例如  $ED_1(g_1, \dots, g_N)$ ) 应用执行装置函数级联。

[0013] 根据本发明, 以封装为  $g_i$  的形式提供组成函数  $f_i$ , 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ 。用于封装的函数  $p_i$  也通过以  $h_i$  的形式提供而隐藏,  $h_i$  是乘法形式  $p_{2i}^{-1} \circ p_{2i-2}$ , 其中  $2 \leq i \leq N$ 。通过在执行装置中以交织方式执行函数  $g_i$  和  $h_i$  (如图 4 中图解的实例), 在  $f_i$  不可直接识别的情况下获得函数级联的函数性。具体地说, 如果  $f_i$  表示 Feistel 密码的轮函数, 则嵌入在轮函数中的轮密钥不可直接识别。 $f_i$  的模糊传递提高了安全性。执行函数装置级联可形成媒体播放器的核心函数性, 其中集合  $g_1, \dots, g_N$  使播放器能够执行包含  $f_1$  直至包含  $f_N$  的函数级联。

[0014] 从属权利要求 2 和 3 给出用于保护函数级联的 (函数) 开始的两个相应备选实施例。在权利要求 2 的实施例中, 执行装置函数级联以  $P_1^{-1}$  开始, 例如  $ED_2(y_1, \dots, y_N) = y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ P_1^{-1}$ 。将它应用到  $g_1, \dots, g_N$ , 作为在装置中执行的函数序列的函数开始给出:  $\dots \circ g_2 \circ h_2 \circ g_1 \circ P_1^{-1} = \dots \circ P_3^{-1} \circ f_2 \circ P_2 \circ P_2^{-1} \circ f_1 \circ P_1 \circ P_1^{-1} = \dots \circ P_3^{-1} \circ f_2 \circ f_1$ , 这样, 执行装置明确地执行  $f_1$ 。在权利要求 3 的实施例中, 通过用帮助隐藏  $P_{1-1}$  的开始函数  $f_0$  扩展函数级联来提高安全性。函数级联例如可以是  $FC_2(x) = f_N \circ \dots \circ f_1 \circ f_0(x)$ 。执行装置函数级联从函数  $S_1$  开始, 例如  $ED_3(y_1, \dots, y_N) = y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$ , 其中  $S_1$  在功能上相当于  $P_1^{-1} \circ f_0$ 。由于  $S_1$  仅表示与  $f_0$  相乘形式的  $P_1^{-1}$ , 因此无法以诸如读取某些存储单元的直接方式从执行装置检索  $P_1^{-1}$ 。最好,  $f_0$  是一个全局秘密。

[0015] 从属权利要求 4 和 5 给出用于以类似于权利要求 2 和 3 的方式保护函数级联的 (函数) 结尾的两个相应的备选实施例。

[0016] 根据从属权利要求 6 的方法, 选择的置换  $p_i$  序列对装置是唯一的。这样, 函数级联不仅以模糊形式而且以个性化形式提供给执行装置。例如, 如果函数级联表示具有嵌入解密密钥的 Feistel 密码, 则密码分析或者强力攻击可能导致获得  $g_1, \dots, g_N$  的黑盒函数性。这个打破的函数性于是只能与相应的执行装置函数级联组合时起作用, 而不是与任何其它

执行装置组合都起作用。这显著地限制了成功攻击的影响。

[0017] 根据从属权利要求 7 的方法,执行装置函数级联嵌入在一个程序中,例如以媒体播放器或媒体播放器的插件程序的形式。执行装置因此配置了安全、个性化的软件。

[0018] 根据从属权利要求 8 的方法,函数  $g_1, \dots, g_N$  构成程序的插件程序。如果程序本身是插件程序,则函数  $g_1, \dots, g_N$  事实上是插件程序的插件程序。作为备选,根据从属权利要求 9 的方法,函数  $g_1, \dots, g_N$  可以与执行装置函数级联嵌入在同一程序中。

[0019] 为了实现本发明的目的,用于使执行装置中的处理器执行数字信号处理函数  $f$  的计算机程序产品包括函数级联,该函数级联包括多个信号处理函数  $f_i$ , 其中  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出 (例如,  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ ), 通过如下方式:

[0020] 加载执行装置函数级联,该函数级联包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ , 其中  $y_1, \dots, y_N$  是函数参数,

[0021] 加载函数  $g_1, \dots, g_N$  的集合;

[0022] 对函数  $g_1, \dots, g_N$  的集合应用执行装置函数级联;其中:

[0023]  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

[0024]  $h_i$  在功能上相当于  $p_{2i}^{-1} \circ p_{2i-2}$  其中  $2 \leq i \leq N$ ; 以及

[0025]  $p_i$  是可逆置换, 其中  $1 \leq i \leq 2N$ 。

[0026] 为了实现本发明的目的,提供了一种以模糊、匿名形式向多个执行装置提供数字信号处理函数  $f$  的方法,各个执行装置由唯一下标  $j$  标识;函数  $f$  包括函数级联,该函数级联包括多个信号处理函数  $f_i$ , 其中  $1 \leq i \leq N$ , 用于处理数字信号输入  $x$  以产生数字信号输出 (例如,  $FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$ ), 所述方法包括:

[0027] 选择  $2N$  个可逆置换  $p_i$  的集合, 其中  $1 \leq i \leq 2N$ ;

[0028] 计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ;

[0029] 为各个装置  $j$  选择相应的  $2N$  个可逆置换  $p_{j,i}$  的集合和 / 或序列, 其对装置和 / 或装置的用户是唯一的;

[0030] 为各个执行装置  $j$  计算  $N-1$  个函数  $h_{j,i}$  的相应集合, 其中  $h_{j,i}$  在功能上相当于  $p_{j,2i-1}^{-1} \circ p_{j,2i-2}$ , 其中  $2 \leq i \leq N$ ;

[0031] 为各个执行装置  $j$  配备各自的执行装置函数级联  $ED_j(y_1, \dots, y_N)$ , 该函数级联包括  $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$ ;

[0032] 为各个执行装置  $j$  配备各自的加载程序函数  $loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$ , 其中  $l_{j,i}$  在功能上相当于  $p_{j,2i-1} \circ p_{2i}$ , 并且  $r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ ;

[0033] 向执行装置提供函数  $g_1, \dots, g_N$ ; 以及

[0034] 在执行装置中, 执行  $ED_j(loader_j(g_1, \dots, g_N))$ 。

[0035] 函数  $f_i$  以与权利要求 1 所述相同的方式以函数  $g_1, \dots, g_N$  的形式模糊化。函数  $g_1, \dots, g_N$  对各个装置一样, 并可看作对应于一个缺省 / 匿名装置。执行装置配备了装置特定的 (“个性化”) 执行装置级联。装置特定的加载程序函数用来将各自的匿名函数  $g_i$  转换为相应的装置特定函数, 该装置特定函数能被馈送到执行装置级联。加载程序函数利用基于未泄漏的置换  $p_{j,i}$  的集合 / 序列的转换函数  $l_{j,i}$  和  $r_{j,i}$ 。

[0036] 根据从属权利要求 12 的方法, 函数  $g_i$  例如可利用广播或者在诸如 CD-ROM 或 DVD 的存储介质上以同一方式提供给所有装置。

[0037] 根据下文所述的实施例, 本发明的这些及其它方面是显而易见的, 并将参考下文所述的实施例进行阐述。

## 附图说明

[0038] 在附图中:

[0039] 图 1 显示基于先有技术插件程序的解码的框图;

[0040] 图 2 显示基于先有技术的解密的框图;

[0041] 图 3 显示先有技术集成解密 / 解码系统的框图;

[0042] 图 4 显示根据本发明的模糊化;

[0043] 图 5 显示模糊的简单实例;

[0044] 图 6 显示根据本发明的系统的框图;

[0045] 图 7 显示根据本发明的系统的另一实施例;

[0046] 图 8 图解根据本发明的匿名模糊; 以及

[0047] 图 9 图解匿名模糊的备选实施例。

## 具体实施方式

[0048] 图 3 显示可采用本发明的先有技术系统的框图。在图 3 的实例中, 内容 (一般为音频和 / 或视频内容) 分布在介质 310 上。介质对于各个播放器可以是一样的。介质可为任何适当的类型, 例如音频 CD、DVD、固态等。介质上的内容是复制保护的, 最好利用诸如 Feistel 密码的加密算法加密。存储介质可包括有关解密密钥的信息。或者, 存储介质可包括信息 312 (比如标识符), 其使播放器能够例如通过将它从因特网中的服务器下载而检索信息。在安全模块 320 中通过利用密钥特定的密钥 322 和信息 312 来计算 324 解密密钥 326, 从而创建解密密钥。解密密钥在第二模块 330 中被接收 332。第二模块 330 解密 334、解码 336 并再现 338 介质 310 的内容 314。

[0049] 图 4 图解根据本发明的方法。将数字信号处理函数  $f$  以模糊形式提供给执行装置。函数  $f$  包括函数级联, 该函数级联包括多个信号处理函数  $f_i, 1 \leq i \leq N$ 。例如, 函数级联的核心可由  $FC_1(x) \equiv f_N \circ \cdots \circ f_1(x)$  形成。应该注意, 这里使用了传统的数学符号:  $g \circ f(x) = g(f(x))$ 。原则上, 函数级联可以是任何数字信号处理函数。在优选实施例中, 函数级联包括密码。例如, 函数  $f_i$  可表示 Feistel 密码的第  $i$  轮 ( $i > 0$ )。在这种情况下,  $f_i$  定义为:

$$[0050] \quad f_i(L_{i-1} \oplus R_{i-1}) = R_{i-1} \oplus (L_{i-1} \oplus F(R_{i-1}, K_i)),$$

[0051] 其中  $K_i$  是用在第  $i$  轮的子密钥, 并且  $F$  是任意轮函数。

[0052] 根据本发明, 选择  $2N$  个可逆置换  $p_i$  的集合,  $1 \leq i \leq 2N$ 。接下来, 计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ 。在此情况下, 在功能上相当于意味着: 对于各个允许的输入值, 如果  $g_i$  应用到同一输入 (例如  $x$ ), 则获得与在将  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$  应用到那个输入相同的结果。合成函数  $p_{2i}^{-1}$ 、 $f_i$  和  $p_{2i-1}$  不是分别可见的,  $g_i$  提供  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$  的黑盒函数性。图 5 图解非常简单的一维函数的这个方法。在此实例

中,

[0053]  $p_4(x) = \sqrt{x}; p_4^{-1}(x) = x^2; p_3(x) = \frac{x}{3}; p_3^{-1}(x) = 3x; f_2(x) = x+3$ 。由此,

[0054]  $g_2(x) = p_4^{-1} \circ f_2 \circ p_3(x) = p_4^{-1} \circ f_2(p_3(x)) = p_4^{-1} \circ f_2(\frac{x}{3}) = p_4^{-1}(\frac{x}{3}+3) = (\frac{x}{3}+3)^2$ 。从计算机

编译程序构造领域众所周知的是,能够如何利用所谓的部分估算而获得  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$  的黑盒函数性。N.D.Jones、C.K.Gomard 和 P.Sestoft 的“部分估算和自动程序生成 (Partial Evaluation and Automatic Program Generation)”第一章描述了部分估算的概念。在此不对其进行更详细描述。应理解,数字信号输入  $x$  是多维参数,例如 64 或 128 位块 / 向量,以能够执行有用的置换。根据本发明,计算  $N-1$  个函数  $h_i$  的集合,其中  $h_i$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{2i-2}$ ,  $2 \leq i \leq N$ 。利用图 5 的简单实例,  $h_2(x) = p_3^{-1} \circ p_2(x) = 3 \bullet p_2(x); h_3(x) = p_5^{-1} \circ p_4(x) = p_5^{-1}(\sqrt{x})$ 。利用这些定义,隐藏  $f_2$  的部分执行装置级联将为:

[0055]

$$\cdots h_3 \circ g_2 \circ h_2 = \cdots (p_5^{-1}(\sqrt{x})) \circ (\frac{x}{3}+3)^2 \circ (3 \bullet p_2(x)) = (p_5^{-1}(\sqrt{x})) \circ (\frac{3 \bullet p_2(x)}{3}+3)^2$$

[0056]  $= (p_5^{-1}(\sqrt{x})) \circ (p_2(x)+3)^2 = p_5^{-1}(\sqrt{(p_2(x)+3)^2}) = p_5^{-1}(p_2(x)+3)$ 。可以看出,这实际上在功能上相当于  $p_5^{-1} \circ f_2 \circ p_2(x)$ 。因此,已经执行这个级联的执行装置执行了  $f_2$ ,而不需要对  $f_2$  具有明确了解。

[0057] 在另一个实例中,  $N = 2$ , 并且  $f_1$  和  $f_2$  分别估算为各自的映射表,如下给出:

[0058]  $f_1: \{0- > 3, 1- > 1, 2- > 6, 3- > 2, 4- > 7, 5- > 5, 6- > 4, 7- > 0, 8- > 8\}$ ,

[0059]  $f_2: \{0- > 4, 1- > 1, 2- > 5, 3- > 7, 4- > 6, 5- > 2, 6- > 0, 7- > 8, 8- > 3\}$ 。

[0060] 在此实例中,  $f_1$  是可逆函数,它将 0 和 8 之间的数转换为 0 和 8 之间的数,例如,值 0 被转换为值 3,值 1 被转换为 1,值 2 被转换为 6 等等。在此实例中使用以下四个相应的置换:

[0061]  $p_1: \{0- > 5, 1- > 3, 2- > 1, 3- > 7, 4- > 0, 5- > 6, 6- > 2, 7- > 8, 8- > 4\}$

[0062]  $p_2: \{0- > 8, 1- > 6, 2- > 7, 3- > 3, 4- > 4, 5- > 2, 6- > 0, 7- > 1, 8- > 5\}$

[0063]  $p_3: \{0- > 3, 1- > 5, 2- > 7, 3- > 1, 4- > 6, 5- > 0, 6- > 2, 7- > 8, 8- > 4\}$

[0064]  $p_4: \{0- > 3, 1- > 0, 2- > 5, 3- > 2, 4- > 7, 5- > 8, 6- > 1, 7- > 4, 8- > 6\}$

[0065] 对于此实例,使用以下三个可逆置换:

[0066]  $f_2^{-1}: \{0- > 6, 1- > 7, 2- > 5, 3- > 3, 4- > 4, 5- > 8, 6- > 1, 7- > 2, 8- > 0\}$

[0067]  $f_3^{-1}: \{0- > 5, 1- > 3, 2- > 6, 3- > 0, 4- > 8, 5- > 1, 6- > 4, 7- > 2, 8- > 7\}$

[0068]  $p_4^{-1}: \{0- > 1, 1- > 6, 2- > 3, 3- > 0, 4- > 7, 5- > 2, 6- > 8, 7- > 4, 8- > 5\}$  给出这些函数,  $h_2(x) = p_3^{-1} \circ p_2(x)$  就给出为:

[0069]  $h_2: \{0- > 7, 1- > 4, 2- > 2, 3- > 0, 4- > 8, 5- > 6, 6- > 5, 7- > 3, 8- > 1\}$ 。例如,  $p_2$  将 0 映射到 8, 并且  $p_3^{-1}$  将 8 映射到 7。由此,  $h_2(0) = p_3^{-1} \circ p_2(0) = 7$ 。类似地,

$g_1(x) = p_2^{-1} \circ f_1 \circ p_1(x)$  由下式给出:

[0070]  $g_1: \{0- > 8, 1- > 5, 2- > 7, 3- > 6, 4- > 3, 5- > 4, 6- > 1, 7- > 0, 8- > 2\}$  并且  $g_2(x) = p_4^{-1} \circ f_2 \circ p_3(x)$  由下式给出:

[0071]  $g_2 : \{0 \rightarrow 4, 1 \rightarrow 3, 2 \rightarrow 5, 3 \rightarrow 6, 4 \rightarrow 1, 5 \rightarrow 7, 6 \rightarrow 2, 7 \rightarrow 0, 8 \rightarrow 8\}$

[0072] 执行装置配置了执行装置函数级联, 该执行装置函数级联包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ , 其中  $y_1, \dots, y_N$  是函数参数。这在图 4 中显示为函数序列  $h_N, h_{N-1}, \dots, h_2$  410。示例性执行装置函数级联是  $ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ 。此外, 将函数  $g_1, \dots, g_N$  提供给执行装置。这在图 4 中显示为函数序列  $g_N, g_{N-1}, \dots, g_1$  420。在执行装置中, 执行装置函数级联应用到函数  $g_1, \dots, g_N$ 。这在执行装置中例如给出总的信号处理函数  $ED_1(g_1, \dots, g_N)$ 。此函数然后能应用到数字信号输入  $x$ 。

[0073] 看一下如  $h_{i+1} \circ g_i \circ h_i$  的链的中间部分, 这给出:

[0074]  $h_{i+1} \circ g_i \circ h_i = p_{2i+1}^{-1} \circ p_{2i} \circ p_{2i}^{-1} \circ f_i \circ p_{2i-1} \circ p_{2i-1}^{-1} \circ p_{2i-2} = p_{2i+1}^{-1} \circ f_i \circ p_{2i-2}$ 。此公式的第一和最小项将被相应的  $g$  项消除。总的结果是, 执行装置执行包括函数级联  $f_N \circ \dots \circ f_1(x)$  的函数, 而不需要访问任何函数  $f_i$ 。这些函数由此被模糊化了。

[0075] 在优选实施例中, 给出用于处理链的开始和结尾的选项。不需要任何其它方法, 执行装置中得到的总的信号处理函数可以为  $ED_1(g_1, \dots, g_N) \equiv p_{2N-1}^{-1} \circ f_N \circ \dots \circ f_1(x) \circ p_1$ 。例如, 项  $p_1$  可以通过利用包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$  的执行装置函数级联而消除。例如  $ED_2(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$ 。最好, 项  $p_1^{-1}$  在执行装置中保持保密。这样做的优选方式是用另一信号处理函数  $f_0$  扩展函数级联 (例如  $FC_2(x) \equiv f_N \circ \dots \circ f_1 \circ f_0(x)$ )。执行装置函数级联则包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$ , 例如  $(ED_3(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1)$ , 其中  $S_1$  在功能上相当于  $p_1^{-1} \circ f_0$ 。用这种方式, 各个项  $p_1^{-1}$  和  $f_0$  不需要暴露, 而是只有复合形式  $p_1^{-1} \circ f_0$  存在。最好,  $f_0$  是一个全局秘密, 即, 对需要了解的各方已知, 但并不作任何进一步地分配。全局秘密本身是已知的, 并且以保密方式传递全局秘密的方式也是已知的, 因此不在这里更进一步论述。

[0076] 以相应的方式, 能够采取一些方法用于处理项  $p_{2N-1}^{-1}$ 。例如, 执行装置函数级联可以包括:  $p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  (例如  $ED_4(y_1, \dots, y_N) \equiv p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ )。为更好地保护  $p_{2N}$ , 函数级联可以另一信号处理函数  $f_{N+1}$  结束 (例如  $FC_3(x) \equiv f_{N+1} \circ f_N \circ \dots \circ f_1(x)$ )。执行装置函数级联由此包括  $S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$  (例如  $ED_5(y_1, \dots, y_N) \equiv S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ ), 其中  $S_2$  在功能上相当于  $f_{N+1} \circ p_{2N}$ 。

[0077] 图 6 图解其中可以采用本发明的系统。系统 600 包括服务器 610 和至少一个执行装置 620。服务器可以实现在传统的计算机平台上, 例如在用作诸如网络服务器或者文件服务器的服务器的平台上。服务器包括处理器 612。处理器 612 在程序的控制下操作。程序可在嵌入式存储器 (如嵌入式 ROM) 中永久地嵌入在处理器中, 但是也可以从诸如硬盘 (未显示) 的后备存储器加载。在程序的控制下, 处理器 612:

[0078] ●选择  $2N$  个可逆置换  $p_i$  的集合, 其中  $1 \leq i \leq 2N$ ;

[0079] ●计算  $N$  个函数  $g_i$  的集合, 其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ , 其中  $1 \leq i \leq N$ ; 以及

[0080] ●计算  $N-1$  个函数  $h_i$  的集合, 其中  $h_i$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{2i-2}$ , 其中  $2 \leq i \leq N$ 。

[0081] 可以选择置换, (例如随机或者伪随机地) 从非常大的置换集合中选择, 置换集合可保存在 (最好保密) 存储器 (未显示) 中。服务器还可利用适当的程序来生成置换。如

何创建可逆置换是众所周知的,因此将不在此更进一步地描述。

[0082] 另外,服务器包括部件 614,用于为执行装置配备执行装置函数级联,该函数级联包括  $y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$ ,其中  $y_1, \dots, y_N$  是函数参数。服务器可以任何适当的形式做这个。例如,在工厂中,项  $h_i$  可在制造执行装置 620 期间保存在执行装置的存储模块中。图 6 给出了该项通过因特网 630 直接下载到执行装置 620。服务器 610 还包括部件 616,用于向执行装置 620 提供函数  $g_1, \dots, g_N$ 。函数  $g_i$  包括了各自的函数  $f_i$ 。函数  $f_i$  可以针对数字信号输入  $x$  具体选择。例如,各个视频标题可以用相应的加密函数加密(例如用同一密码但是具有内容特定的密钥)。为此,服务器 610 还可包括软件,用于控制处理器 612 对内容 640 加密并将加密内容 642 提供给分配介质,例如用于在存储介质上分配,或者通过如因特网的通信介质分配。

[0083] 执行装置 620 包括部件 626,用于从服务器 610 获得函数  $g_1, \dots, g_N$ 。这些部件与服务器的部件 616 配合,因此不再进一步描述。执行装置 620 还包括处理器 622。处理器可以是任何适当的类型,比如区分个人计算机或者嵌入式微控制器的处理器。处理器 622 在程序的控制下操作。程序可以利用嵌入式存储器(如嵌入式 ROM)永久地嵌入在处理器 622 中,但是也可以从诸如硬盘(未显示)的后备存储器加载。在程序的控制下,处理器 622 加载执行装置函数级联,并将加载的执行装置函数级联例如通过执行  $ED_1(g_1, \dots, g_N)$  而应用到函数  $g_1, \dots, g_N$ 。得到的信号处理函数然后可应用到信号输入  $x$ (例如从介质接收的内容)。处理器 622 可以任何适当的形式加载执行装置函数级联。例如,该级联可能已经在制造期间预先存储在存储器中,将加载简化为简单的存储器读访问。在图 6 的实例中,执行装置 620 包括部件 624,用于例如通过因特网 630 或者从介质 650 检索级联(或级联的项)。类似地,执行装置 620 可从介质 650 检索加密内容 652,并利用处理器 622 对其解密。处理器还可对解密的内容进行解码。

[0084] 图 7 显示一个优选实施例,其中将执行装置函数级联提供给嵌入在软件程序 710 中的执行装置 620,以便由处理器 622 执行。图 7 中相同的附图标记是指与图 6 中所用的相同的项。软件程序 710 可以是如媒体播放器的程序的插件程序。因此,图 7 的部件 614 可经因特网提供此插件程序 710(例如图 7 的项 630),或在制造期间将它直接嵌入在执行装置 620 中。

[0085] 在一个实施例,函数  $g_1, \dots, g_N$  以程序 710 的插件程序的形式提供给执行装置 620。在程序 710 已经是插件程序的情况下,函数  $g_1, \dots, g_N$  实际上是插件程序的插件程序。备选地,通过将函数  $g_1, \dots, g_N$  嵌入在软件程序 710 中而将函数  $g_1, \dots, g_N$  提供给执行装置 620,而嵌入又是通过向函数参数  $g_1, \dots, g_N$  应用执行装置函数级联而实现的。这样,程序 710 嵌入了函数  $h_i$  和  $g_i$ 。

[0086] 在一个实施例中,各个执行装置和/或执行装置的用户是唯一的,并且由唯一身份(例如唯一的号  $j$ )标识。在根据本发明的系统和方法中,保证序列  $g_i$  和  $h_i$  对于所涉及的一方是唯一的。这能够通过获得执行装置和/或执行装置用户的唯一身份  $j$  来实现,2N 个可逆置换  $p_i$  的相应集合对于所获得的身份是唯一的。类似地,利用相同的置换集合,可以选择唯一的置换序列。两种技术(选择不同的置换集合或者选择不同的置换序列)可以组合。最好,服务器为每个唯一的身份存储(以保密方式)唯一的集合/序列。这样,能够向个人计算机中的各个软件媒体播放器提供唯一的插件程序,用于对媒体标题解密和/或

解码。媒体本身不必是唯一的。加密内容仅取决于加密函数,而不取决于唯一的置换集合/序列。通过定期地(例如在媒体播放器启动时)检验软件是否对应于身份,并且仅在能够建立匹配的情况下执行软件,就能够保证没有播放器软件能够在它不属于的 PC 上执行。如果无意中黑客设法获得了装置特定的置换,则他们只能在所涉及的 PC 上使用,可能还用于用不同的加密(得到不同的函数  $f_i$ ) 保护的内容,但不能在不同的平台上使用。

[0087] 以上已经描述了一种方法和系统,其中信号处理函数级联以模糊的方式提供给执行装置。对于各个装置,可以利用相同的置换集合/序列,或者可以利用装置特定的集合/序列。在下面的内容中,描述一个优选方法,用于通过以模糊方式分配对各个装置相同的信号函数级联(“密钥”),并利用转换例程(“加载程序”)将公用密钥转换为装置特定密钥,从而获得装置特定的集合/序列。“公用密钥”以和之前描述大致相同的方式产生。公用密钥原则上能够“开启”参考播放器或者匿名播放器,但在此实施例,其并不由任何实际的执行装置执行。如前所述,该方法包括选择  $2N$  个可逆置换  $p_i$  的集合,其中  $1 \leq i \leq 2N$ ,并计算  $N$  个函数  $g_i$  的集合,其中  $g_i$  在功能上相当于  $p_{2i}^{-1} \circ f_i \circ p_{2i-1}$ ,其中  $1 \leq i \leq N$ 。现在作为补充,该方法包括:为分别由唯一下标  $j$  标识的各个执行装置选择相应的  $2N$  个可逆置换  $p_{j,i}$  的集合和/或序列,其对于装置和/或装置的用户是唯一的。这个集合用于为每个装置提供一个唯一“播放器”。这个唯一播放器通过如下方式形成:为各个执行装置  $j$  计算  $N-1$  个函数  $h_{j,i}$  的相应集合,其中  $h_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{j,2i-2}$ ,其中  $2 \leq i \leq N$ ,并且为各个执行装置  $j$  配备各自的执行装置函数级联  $ED_j(y_1, \dots, y_N)$ ,该函数级联包括  $y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$ 。但是,这个装置特定的集合  $h_{j,i}$  与能够“开启”利用集合  $h_i$  的参考播放器的模糊函数级联不匹配。这后一个集合/播放器集合对任何执行装置都不可用。相反,执行装置  $j$  配备了各自的加载程序函数  $loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$ ,其中  $l_{j,i}$  在功能上相当于  $p_{j,2i}^{-1} \circ p_{2i}$ ,并且  $r_{j,i}$  在功能上相当于  $p_{2i-1}^{-1} \circ p_{j,2i-1}$ 。如上所述,各个执行装置配置了相同的函数  $g_1, \dots, g_N$ 。执行装置然后执行  $ED_j(loader_j(g_1, \dots, g_N))$ 。在这个公式中,  $loader_j(g_1, \dots, g_N)$  有效地将匿名密钥  $g_1, \dots, g_N$  转换为装置特定密钥,该装置特定密钥最佳地匹配执行装置函数级联  $ED_j(y_1, \dots, y_N)$ 。利用定义  $loader_j(g_1, \dots, g_N) = (g_{j,1}, g_{j,2}, \dots, g_{j,N})$ ,  $loader_j(g_1, \dots, g_N)$  的第  $i$  个分量是  $g_{j,i} = l_{j,i} \circ g_i \circ r_{j,i}$ ,  $i$ 。利用以上给出的定义,这给出  $g_{j,i} = p_{j,2i}^{-1} \circ p_{2i} \circ p_{2i}^{-1} \circ f_i \circ p_{2i-1} \circ p_{2i-1}^{-1} \circ p_{j,2i-1}$ , 该式能够重写为  $g_{j,i} = p_{j,2i}^{-1} \circ f_i \circ p_{j,2i-1}$ 。这与利用装置特定的置换集合/序列是一样的,其中装置特定的集合  $h_{j,i}$  消除了置换。

[0088] 在图 8 中还图解了使用匿名模糊密钥和装置特定加载程序的概念。匿名播放器 P1-R 810 包括函数  $h_i$ 。匿名播放器 P1-R 能够由相应密钥 K-R 812 开启,密钥 K-R 812 包括以集合  $g_i$  形式的模糊信号处理函数  $f_i$ 。匿名播放器 P1-R 不对任何一方公开。各方改为配置一个唯一的装置特定播放器,所显示的是播放器 P1-1 830 和 P1-2 840。公用密钥 K-R 提供给各方。但是,这个公用密钥不匹配特定播放器。因此,各方也配置了装置特定密钥加载程序 K-L,显示的是 820 和 825。加载程序 820 和 825 用来将匿名密钥 K-R 812 转换为装置特定密钥 K-j。为此,加载程序 K-L<sub>i</sub> 包括函数  $l_{j,i}$  和  $r_{j,i}$ 。如图 8 所示,原则上,利用装置特定加载程序。如图 9 中进一步所示,事实上,加载程序可以相同,但是提供装置特定函数  $l_{j,i}$  和  $r_{j,i}$ 。在图 9 的实例中,在提供  $l_{1,i}$  和  $r_{1,i}$  时,将匿名密钥 K-R 812 转换为装置 1 的装置

特定密钥 832 ;在提供  $l_{2,i}$  和  $r_{2,i}$  时,将匿名密钥 812 转换为装置 2 的密钥 842。装置特定播放器 830、840 然后分别利用装置特定密钥集合  $h_1, i832$  和 842 开启。应该理解,在这些实例中,短语“密钥”和“播放器”是可互换的,因为两个函数链相互锁定。图 4 的实例图解了两个链作为密钥。以类似的方式,它还可以图解为两个联锁播放器。

[0089] 现在应该理解,匿名播放器 810 (包括  $g_N, \dots, g_1$ ) 可有利地通过广播和 / 或在存储介质上的分配向各个执行装置提供对各个执行装置相同的内容,只是因为这个播放器对各个装置是一样的。类似地,要由各个执行装置处理的数字信号输入  $x$  能够通过广播和 / 或在存储介质上的分配而分配对各个执行装置相同的内容。加载程序特定的方面最好通过“一对一的通信信道”和 / 或具有装置特定内容的存储介质向执行装置  $j$  提供至少一个以下相应函数的集合:  $h_{j,i}$ 、 $l_{j,i}$  或  $r_{j,i}$ 。“一对一的通信信道”可以任何适当的方式获得。最好,服务器利用因特网经由保密链路 (例如 SSL) 下载装置特定信息。

[0090] 如上所述,函数  $f$  可以是基于 Feistel 密码网络的解密函数,并且各个信号处理函数  $f_i$  是各自的 Feistel 解密轮函数。在这样的情况下,每一个置换  $p_i$  最好是 Feistel 变换器,其中如果存在可逆函数  $Q_x$  和  $Q_y$  并且  $Q(\langle x, y \rangle) = \langle Q_x(x), Q_y(y) \rangle$ , 其中  $Q_x(x) \oplus Q_x(y) = Q_x(x \oplus y)$  并且  $Q_y(x) \oplus Q_y(y) = Q_y(x \oplus y)$ , 则在连续对  $\langle x, y \rangle$  上操作的函数  $Q$  是 Feistel 变换器。如果这些条件满足,则函数  $f_i$  能最佳地隐藏。实际上,可显示存在许多这种 Feistel 变换器,给出足够的空间用于装置特定的置换选择。Feistel 变换器的定义基于利用以上给出的定义的理解,Feistel 轮  $f_i(\langle L_{i-1}, R_{i-1} \rangle) = \langle R_{i-1}, (L_{i-1} \oplus F(R_{i-1}, K_i)) \rangle$  可看作为  $f_i = \text{swap} \circ \text{involutary}_F$ , 其中定义  $\text{swap}(\langle x, y \rangle) = \langle y, x \rangle$  和  $\text{involutary}_F(\langle x, y \rangle) = \langle x, y \oplus F(x) \rangle$ 。然后,保持  $\text{swap}^{-1} = \text{swap}$  并且  $\text{involutary}_F^{-1} = \text{involutary}_F$ 。

[0091] 应该理解,本发明还扩展到计算机程序,特别是适合于实践本发明的载体上或载体中的计算机程序。程序可以为源代码、目标代码、中间代码源和诸如部分编译形式的目标代码的形式,或者任何其它适用于实现根据本发明方法的形式。载体是能够承载程序的任何实体或装置。例如,载体可包括诸如例如 CD ROM 或半导体 ROM 的 ROM 的存储介质,或者例如软盘或硬盘的磁记录介质。此外,载体可以是可传输的载体,比如可经由电缆或光缆或者通过无线电或其它方式传递的电信号或光信号。在程序在这样的信号中实施时,载体可以由这样的电缆或者其它装置或部件构成。备选地,载体可以是其中嵌入了程序的集成电路,该集成电路适于执行有关方法,或者用于有关方法的执行中。

[0092] 应该注意,以上提到的实施例阐述不是限制本发明,而且本领域技术人员在不背离所附权利要求书范围的情况下将能设计许多备选实施例。在权利要求书中,括号中的任何附图标记不应视作限制权利要求。利用动词“包含”及其动词变化不排除除了权利要求中所述的那些元件或者步骤以外的元件或者步骤的存在。元件前面的冠词“一个”不排除存在多个这样的元件。本发明可以借助于包括几个分离元件的硬件实现,以及借助于适当地程序的计算机实现。在列举几个部件的装置权利要求中,几个这样的部件可由同一个硬件项实施。某些方法在互相不同的从属权利要求中叙述的事实并不表示这些方法的组合不能有利地使用。

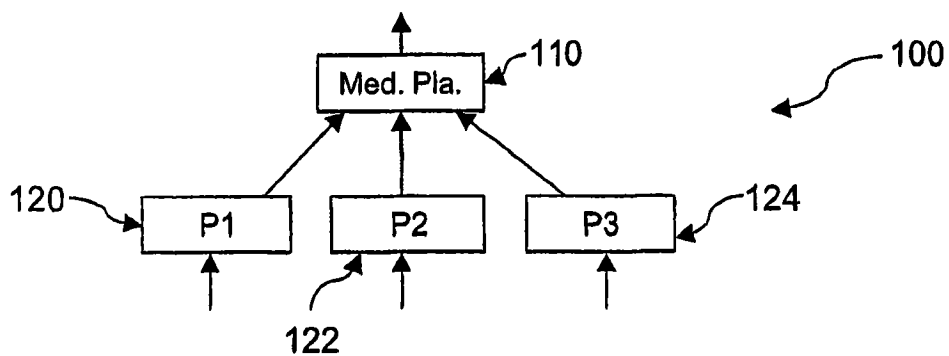


图 1

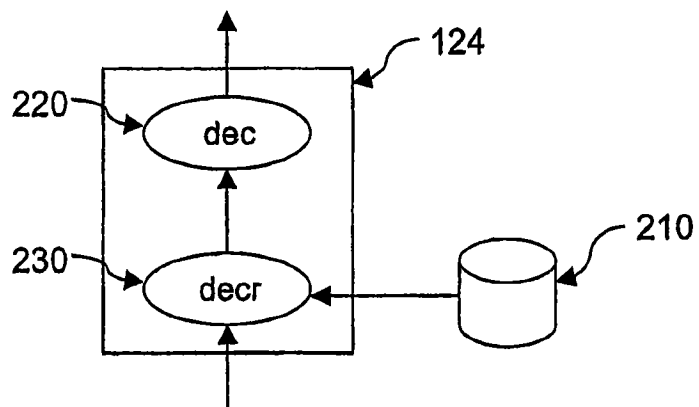


图 2

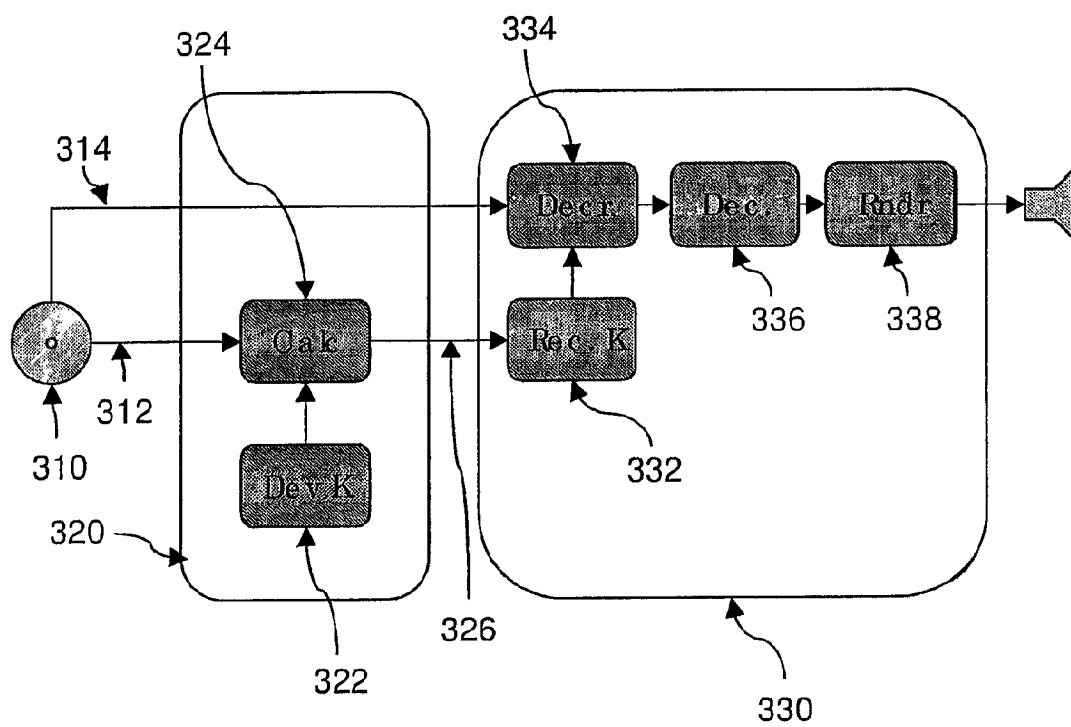


图 3

$$f_N \circ \dots \circ f_1(x)$$
$$g_i = p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$
$$h_i = p_{2i-1}^{-1} \circ p_{2i-2}$$

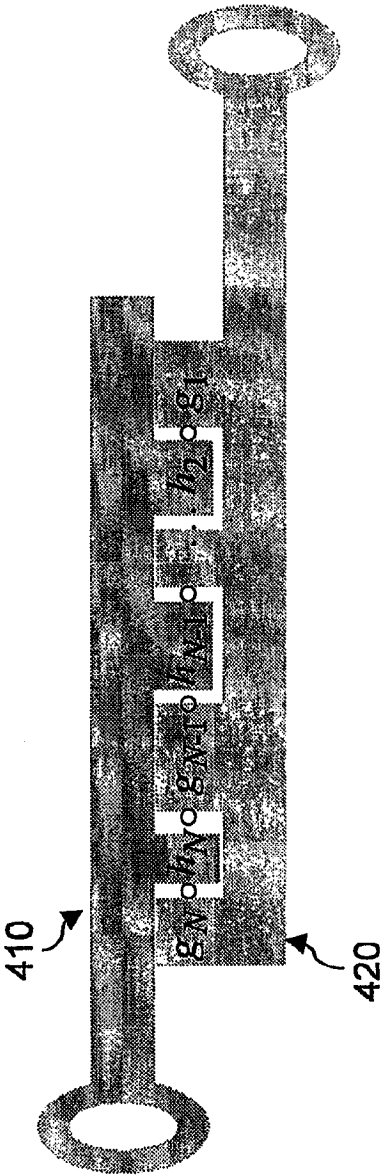


图 4

$$p_4(x) = \sqrt{x}; p_4^{-1}(x) = x^2$$
$$p_3(x) = \frac{x}{3}; p_3^{-1}(x) = 3x$$
$$f_2(x) = x + 3$$

$$g_2(x) = p_4^{-1} \circ f_2 \circ p_3(x) = (\frac{x}{3} + 3)^2$$
$$h_2(x) = p_3^{-1} \circ p_2(x) = 3 \bullet p_2(x)$$
$$h_3(x) = p_5^{-1} \circ p_4(x) = p_5^{-1}(\sqrt{x})$$

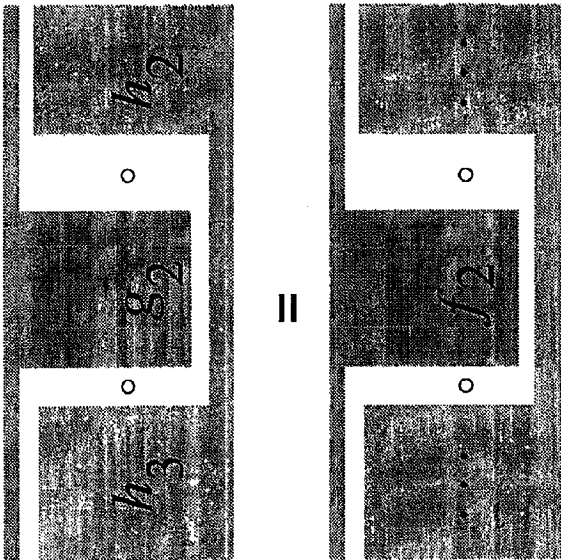


图 5

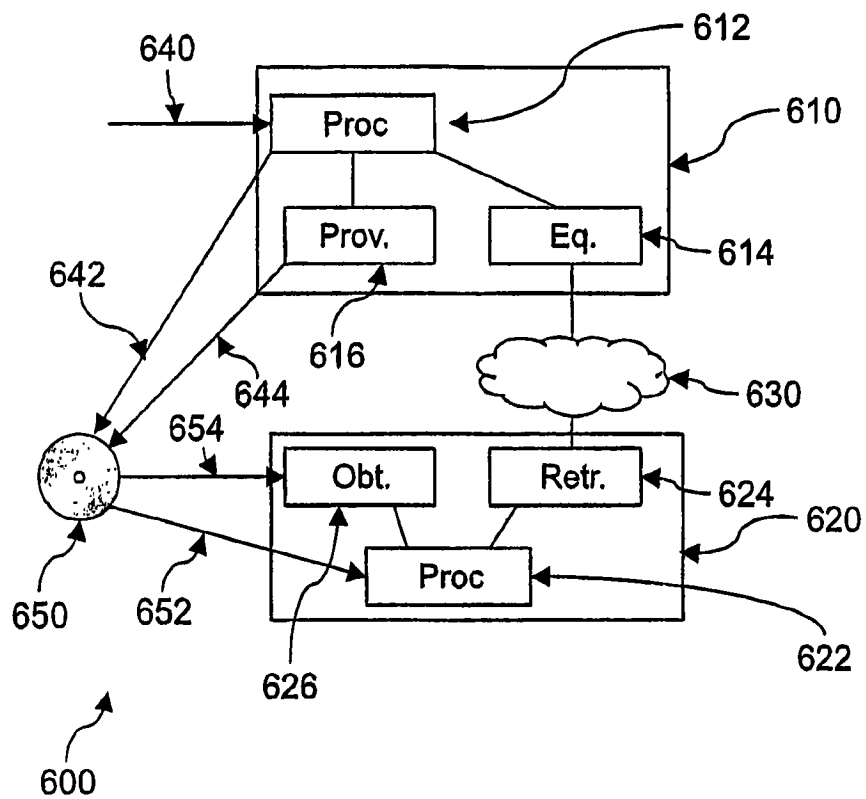


图 6

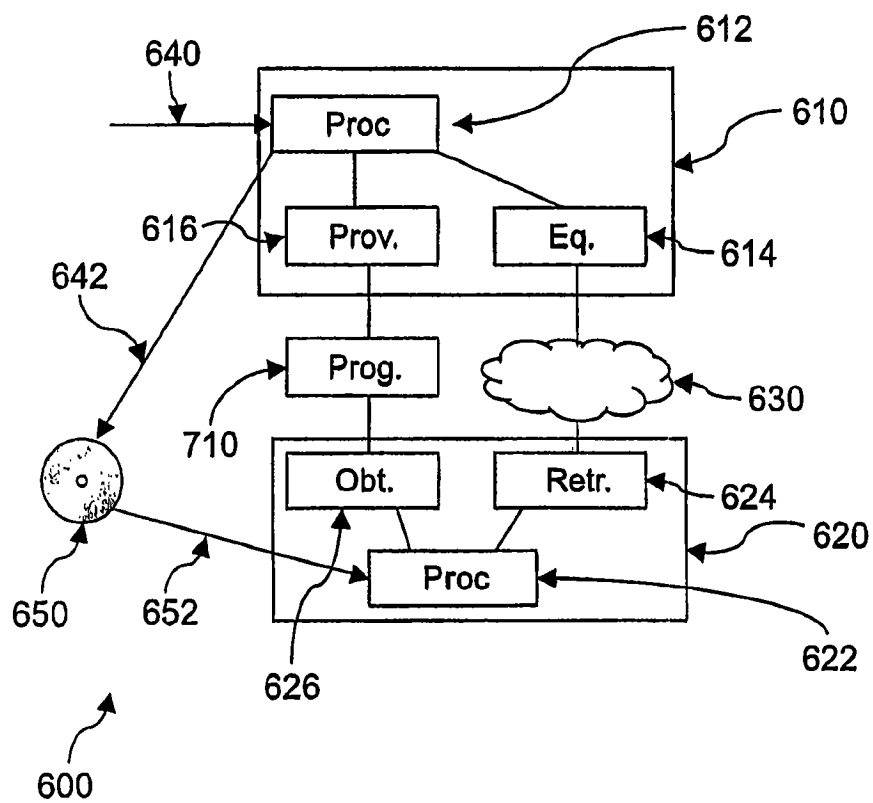


图 7

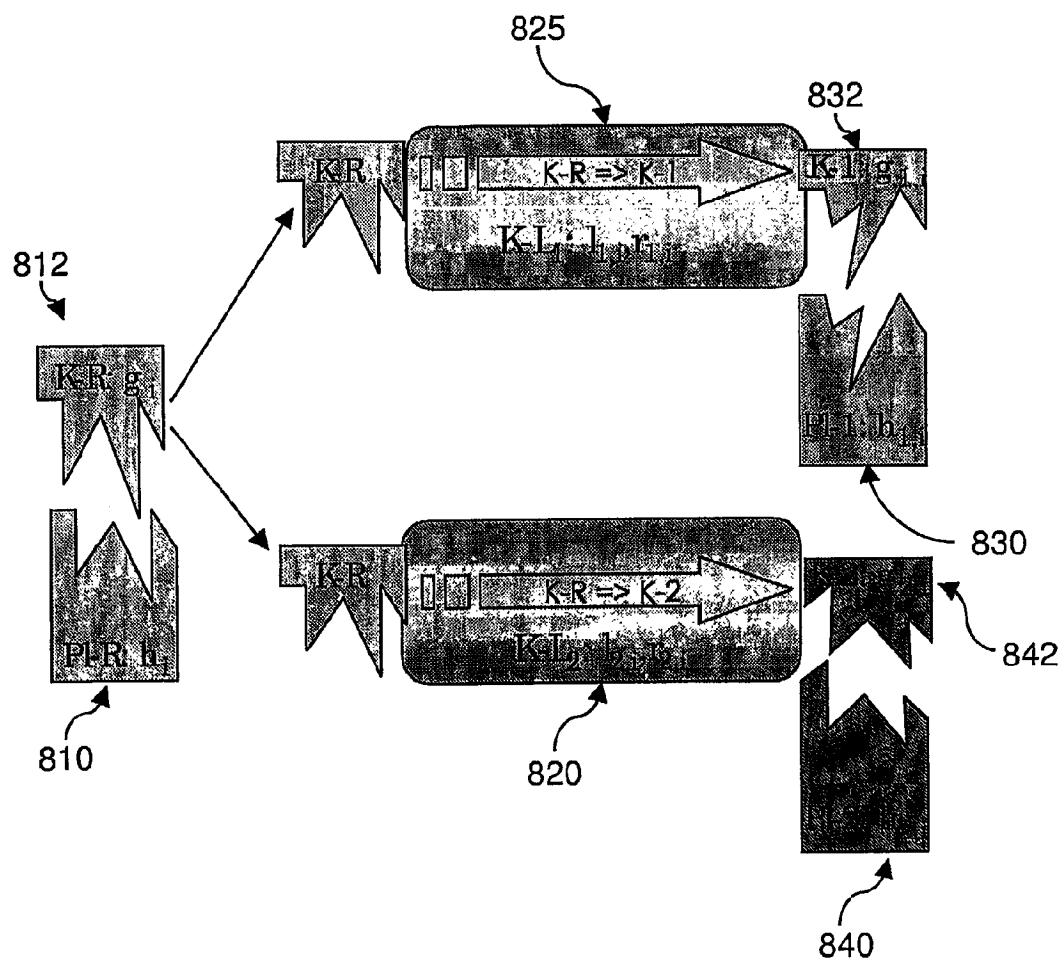


图 8

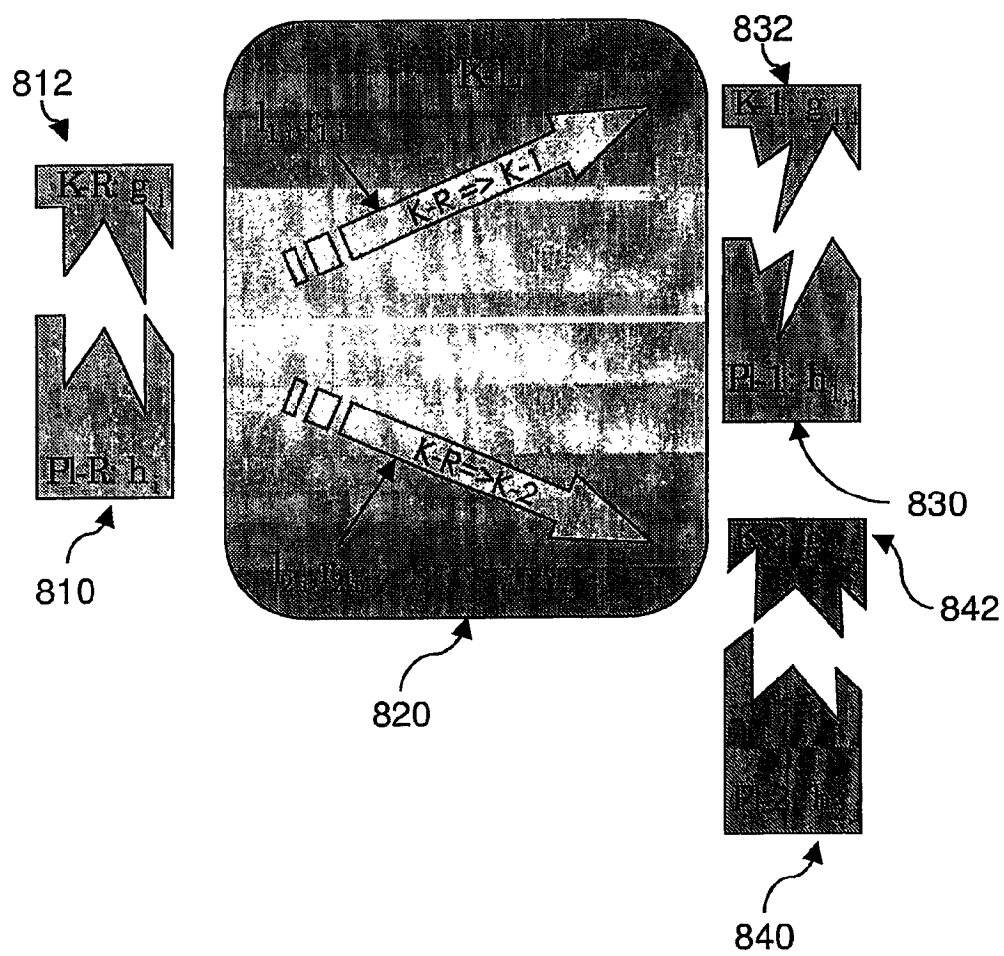


图 9