

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 9/18

G06F 1/00



[12] 发明专利申请公开说明书

[21] 申请号 200380102318.0

[43] 公开日 2005 年 12 月 14 日

[11] 公开号 CN 1708944A

[22] 申请日 2003.10.27

[21] 申请号 200380102318.0

[30] 优先权

[32] 2002.10.28 [33] US [31] 10/282,648

[86] 国际申请 PCT/IB2003/006485 2003.10.27

[87] 国际公布 WO2004/046846 英 2004.6.3

[85] 进入国家阶段日期 2005.4.28

[71] 申请人 松下电器产业株式会社

地址 日本大阪府

[72] 发明人 格雷戈里·M·珀金斯 贺志军

[74] 专利代理机构 永新专利商标代理有限公司

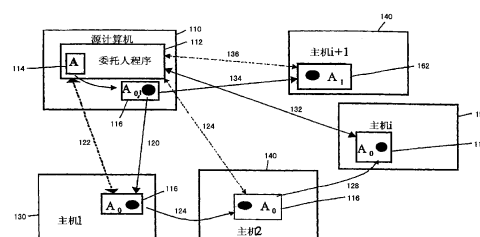
代理人 王 英

权利要求书 7 页 说明书 13 页 附图 3 页

[54] 发明名称 用于可更新防篡改安全系统的自动产生的加密函数

[57] 摘要

一种从一个模板生成的安全加密函数，该模板包含静态程序代码和动态程序代码，该静态程序代码对于所有移动代理都是相同的，该动态程序代码对每个函数都不同。该动态代码执行序列密码加密算法，该加密算法用于将该函数处理的消息进行加密。该动态代码也可生成附加到每条消息的消息摘要。该消息摘要可以是应用于该动态代码以及该消息(可选)的散列函数。通过向每个函数分配一个固定的终止时间或者可发送的消息的最大数量，或者如果该加密函数用于移动代理，则向它分配可访问的主机的最大数量，通过这些方式可以给每个函数分配一个有限的生命期。在其生命期终止后收到的、由加密算法处理的任何消息都被忽略。



1、一种用于创建加密函数的方法，包括以下步骤：

提供一个加密函数模板，该模板具有至少一个静态指令集和用于至少一个动态指令集的指示符；

为所述至少一个动态指令集动态地生成计算机程序指令，所述动态生成的指令对该加密函数是唯一的；以及

用所述动态生成的计算机程序指令取代所述用于至少一个动态指令集的指示符，以形成该唯一加密函数。

2、如权利要求 1 所述的方法，还包括为所述唯一加密函数分配一个唯一标识符的步骤。

3、如权利要求 2 所述的方法，其中，动态地生成计算机程序指令的步骤包括以下步骤：从预定的函数集合中选择多个函数，其中，序列密码算法顺序地应用所选择的函数来加密数据。

4、如权利要求 1 所述的方法，其中，动态地生成计算机程序指令的步骤包括：生成计算机程序指令，所述指令产生一个消息摘要，将该消息摘要与使用所述唯一加密函数处理的至少一条消息一起提供。

5、如权利要求 4 所述的方法，其中，生成产生所述消息摘要的计算机程序指令的步骤包括以下步骤：生成计算机程序指令，所述计算机程序指令将一个散列函数至少应用于所述动态生成的代码上，以产生所述消息摘要。

6、如权利要求 5 所述的方法，其中，所述散列函数是序列密码算法，并且生成将所述散列函数至少应用于所述动态生成的代码上的计算机程序指令的步骤包括以下步骤：从预定的函数集合中选择多个函数，其中，所述序列密码算法顺序地应用所选择的函数，以执行所述散列函数。

7、如权利要求 1 所述的方法，还包括为所述唯一加密函数分配一个有限生命期的步骤。

8、如权利要求 7 所述的方法，其中，为所述唯一加密函数分配一个有限生命期的步骤为代理分配一个预定的终止时间。

9、如权利要求 7 所述的方法，其中，为所述唯一加密函数分配一个有限生命期的步骤分配使用该函数可处理的消息的最大数量。

10、一种用于创建唯一移动代理的方法，包括以下步骤：

提供一个移动代理模板，该模板具有至少一个静态指令集和用于至少一个动态指令集的指示符；

为所述至少一个动态指令集动态地生成计算机程序指令，所述动态生成的指令对该移动代理是唯一的；以及

用所述动态生成的计算机程序指令取代所述用于至少一个动态指令集的指示符，以形成该唯一移动代理。

11、如权利要求 10 所述的方法，还包括为所述唯一移动代理分配一个唯一标识符的步骤。

12、如权利要求 10 所述的方法，其中，所述动态指令集执行一

个加密算法。

13、如权利要求 12 所述的方法，其中，所述加密算法是序列密码算法，并且动态地生成计算机程序指令的步骤包括以下步骤：从预定的函数集合中选择多个函数，其中，所述序列密码算法顺序地应用所选择的函数，以加密数据。

14、如权利要求 10 所述的方法，其中，动态地生成计算机程序指令的步骤包括：生成计算机程序指令，所述指令产生一个消息摘要，将该消息摘要与使用所述唯一加密函数处理的至少一条消息一起提供。

15、如权利要求 14 所述的方法，其中，生成产生所述消息摘要的计算机程序指令的步骤包括以下步骤：生成计算机程序指令，所述计算机程序指令将散列函数至少应用于动态生成的代码，以产生所述消息摘要。

16、如权利要求 15 所述的方法，其中，所述散列函数是序列密码算法，并且生成将所述散列函数至少应用于动态生成的代码的计算机程序指令的步骤包括以下步骤：从预定的函数集合中选择多个函数，其中，所述序列密码算法顺序地应用所选择的函数，以执行所述散列函数。

17、如权利要求 10 所述的方法，还包括为所述唯一移动代理分配一个有限生命期的步骤。

18、如权利要求 17 所述的方法，其中，为所述唯一移动代理分

配一个有限生命期的步骤为该代理分配一个预定的终止时间。

19、如权利要求 17 所述的方法，其中，为所述唯一移动代理分配一个有限生命期的步骤分配所述唯一移动代理可发送的消息的最大数量。

20、如权利要求 17 所述的方法，其中，为所述唯一移动代理分配一个有限生命期的步骤分配所述唯一移动代理可访问的宿主计算机的最大数量。

21、一种用于创建多个唯一移动代理的方法，包括以下步骤：
提供一个移动代理模板，该模板具有至少一个静态指令集和用于至少一个动态指令集的指示符；

对于所述多个移动代理中的各个移动代理，为所述至少一个动态指令集动态地生成多个相互不同的计算机程序指令集；以及

用所述各个动态生成的计算机程序指令取代用于所述至少一个动态指令集的指示符，以生成所述多个唯一移动代理。

22、如权利要求 21 所述的方法，还包括为所述多个唯一移动代理各分配一个相互不同的标识符的步骤。

23、如权利要求 21 所述的方法，其中，所述多个动态指令集各执行一个相互不同的加密算法。

24、如权利要求 23 所述的方法，其中，所述相互不同的加密算法的每一个算法都是序列密码算法，并且动态地生成计算机程序指令的步骤包括以下步骤：从预定的函数集中选择相互不同的多个函数，

其中，所述序列密码算法顺序地应用所选择的函数，以加密数据。

25、如权利要求 24 所述的方法，其中，用于每一个唯一移动代理的所述静态指令集包括将各加密算法应用于所述唯一移动代理发送的至少一条消息的指令。

26、如权利要求 24 所述的方法，其中，用于每一个唯一移动代理的所述静态指令集包括应用各自加密算法以产生一个与由该唯一移动代理发送的至少一条消息一起发送的消息摘要的指令。

27、如权利要求 21 所述的方法，还包括为所述多个唯一移动代理中的每一个移动代理分配一个各自不同的、有限的生命期的步骤。

28、一种用于监控在网络中的消费者宿主计算机间移动的移动代理的方法，以检测可能恶意的宿主计算机，该方法包括以下步骤：

为该移动代理分配一个有限的生命期；

向一个代理监控系统（AMS）注册该移动代理；

在该移动代理与该 AMS 之间建立通信，以确定该移动代理访问的一系列宿主计算机；以及

如果该 AMS 在该移动代理的所述有限生命期内不能与该移动代理通信，则将网络上该代理移到的最后一个宿主计算机标识为可能恶意的。

29、如权利要求 28 所述的方法，其中，为该移动代理分配一个有限生命期的步骤分配该移动代理可向该 AMS 发送的消息的最大数量。

30、如权利要求 28 所述的方法，其中，为该移动代理分配一个有限生命期的步骤分配该移动代理可访问的主机的最大数量。

31、如权利要求 30 所述的方法，其中，为该移动代理分配一个有限生命期的步骤分配一个该移动代理将终止的时间。

32、一种用于监控在网络中的消费者计算机间移动的移动代理的方法，以检测可能恶意的宿主计算机，该方法包括以下步骤：

为该移动代理指定一个唯一加密算法；

向一个代理监控系统（AMS）注册该移动代理；

在该移动代理与该 AMS 之间建立通信，以确定该移动代理访问的一系列宿主计算机；以及

如果该 AMS 从该移动代理收到一个非正确加密的消息，则将网络上该代理移向的最后一个宿主计算机标识为可能恶意的。

33、一种计算机可读载体，包括使计算机执行用于创建加密函数的方法的计算机程序指令，该方法包括以下步骤：

提供一个加密函数模板，该模板具有至少一个静态指令集和用于至少一个动态指令集的指示符；

为所述至少一个动态指令集动态地生成计算机程序指令，所述动态生成的指令对该加密函数是唯一的；以及

用所述动态生成的计算机程序指令取代所述用于至少一个动态指令集的指示符，以形成该唯一加密函数。

34、一种计算机可读载体，包括使计算机执行用于创建多个唯一移动代理的方法的计算机程序指令，该方法包括以下步骤：

提供一个移动代理模板，该模板具有至少一个静态指令集和用于

至少一个动态指令集的指示符；

对于所述多个移动代理中的各个移动代理，为所述至少一个动态指令集动态地生成多个相互不同的计算机程序指令集；以及

用所述各个动态生成的计算机程序指令取代用于所述至少一个动态指令集的指示符，以生成所述多个唯一移动代理。

35、一种计算机可读载体，包括使计算机执行一种用于监控在网络中的宿主计算机间移动的移动代理以检测可能恶意的宿主计算机的方法的计算机程序指令，该方法包括以下步骤：

为该移动代理指定一个唯一加密算法；

向一个代理监控系统（AMS）注册该移动代理；

在该移动代理与该 AMS 之间建立通信，以确定该移动代理访问的一系列宿主计算机；以及

如果该 AMS 从该移动代理收到一个非正确加密的消息，则将该代理移向的网络中最后一个宿主计算机标识为可能恶意的。

用于可更新防篡改安全系统的自动产生的加密函数

发明领域

本发明涉及用于计算机系统的加密函数，尤其涉及用于执行可更新、唯一加密函数的自动产生的程序指令。

技术背景

加密函数广泛用于计算机系统甚至大众消费者应用中。将通过诸如 DVD 这样的媒体发布的程序内容进行加密，以防止未经授权拷贝。加密函数还用于验证计算机系统的用户，将存储在计算机上的诸如日志文件和审计踪迹（audit trails）这样的数据进行编码，以保证所存储数据的完整性。这些函数还可用于移动代理，以便在代理及其委托人（principal）之间建立私有通信信道。只要攻击者可以访问受保护内容并且有充足的时间，就能够破解所有的软件防篡改技术，所以，希望提供一种可更新的加密函数，以增加对受保护内容获得重复访问所需的工作量。虽然结合移动代理对本发明进行了描述，但可以想象得到，本发明可用于上述任何内容和任何系统，以提高加密数据的安全性。

移动代理是在诸如全球网络、广域网或局域网这样的网络中移动并在宿主计算机（而不是其创作者）中执行任务的一种对象。在一个代理的生命期（lifetime）内，有两种类型的计算机服务器：生产者和消费者。代理的生产者，也被认为其委托人，产生并发布该代理。消费者是执行该代理以完成特定任务的服务器。

例如，购物移动代理可用一种诸如 JAVA 这样的、能运行在多种不同类型计算机上的语言写成，然后，用于从若干互联网网页中寻找

一种产品的最低价格。当创建一个代理时，其主人首先要确定他希望该代理完成的任务，如购物。下一步，主人必须指导该代理如何完成该任务，这包括如何在服务器间移动以及如何返回结果。一旦完成这些步骤，该代理就做好被发送到网络上的准备。然后，移动代理在网络上的主机（宿主计算机）间自主地移动，并在主机上执行其程序代码，以达到该移动代理的委托人的目的。

移动代理也可以具有“意识”或伪人工智能，并行动自主于其委托人。例如，它可以经由网络拓扑选择自己的移动路线，以找出完成其使命的新主机。移动代理可与其源计算机系统或其他移动代理通信，从而交换和共享信息。它也可生成其他移动代理或“虫子（bot）”，以辅助其工作。虫子（bot）是驻留在一个计算机上并向其他计算机发送查询请求的程序或指令集。

而移动代理包括在其已旅行和驻留的宿主计算机上执行的命令或指令。

代理的主人对于安全性所关注的是该代理的数据、代码和执行的保密性和完整性。由于各种原因，恶意消费者主机可能试图访问和修改代理的私有数据或代码。对代理做出的大量不同类型的攻击使得恶意主机成为一个很严重的问题。意志坚决的攻击者最终将危及移动代理的安全。唯一的问题是攻击者要花费多长的时间才能成功。

提高移动代理安全性的一种方法是使用防篡改硬件。然而，由于各种资源和花费需求，这种硬件并不是每个人都可用的或在任何地点都可用的。此外，硬件也可以被损害且不容易更新。为了提高移动代理的安全性，软件保护仍是比较理想的，并且，在许多场合也是唯一可用的方法。

另一个安全性担忧是拒绝执行（denial of execution）。当消费者拒绝执行代理的时候，就会出现该问题。例如，购物代理到达一个消费者宿主计算机时，该代理必须首先判断它是否已到达以及是否在正确

的目的主机上执行。在代理确认其已到达正确的消费者之后，该代理将开始执行其任务，该任务通常是从消费者收集和分析数据。一旦任务完成，该代理将确定其下一个目的地。该代理可以移向另一消费者或将结果返回给其委托人然后终止。

在代理执行前和执行期间，恶意消费者可能会阻止该代理的执行。然后，恶意主机可能分析该代理并可能不喜欢该代理获得的结果。例如，一个购物代理被发送到一个恶意主机，该代理的任务是根据其收集的数据来订购商品。由于该代理已发现另一个价钱更优惠的主机，所以该主机可能会终止该代理的执行。一旦执行停止，该恶意主机就可能为自身利益修改该代理的结果，或者该主机可能会假扮该代理。这样，该恶意主机就可以为获得不正当利益而返回虚假信息。

恶意主机也可以尽力分析和危害代理。恶意主机可以仅仅拒绝或延迟代理的执行，因为它希望通过将代理的程序代码进行分析或反向工程而危害该代理。该恶意代理的目的就是随后修改该代理，使得该恶意主机受益，然后重新开始该代理的执行。该恶意主机能够从代理中提取可能包含诸如姓名、信用卡号码和地址等个人数据的敏感信息。

移动代理的开发者和委托人刚开始时忽视了这些特定类型的攻击，因为他们的注意力放在开发有用的移动代理上，并且因为人们认为开发者总能检测出由于委托人停止接收来自代理的信息的拒绝执行。过去，人们曾经错误地认为，代理消费者主机不能从拒绝代理执行中获得任何好处。但是，如上所述，消费者主机有许多拒绝执行的理由，并且，随着移动代理变得更为普遍，该问题也将变得更加严重。

发明内容

本发明体现在一种通过用定制指令集对加密函数动态编程的方式来产生加密函数的方法，其中用于该函数的每一个实例的指令是不

同的。这些定制指令是根据所选择的函数和操作数产生的，然后，将新的动态计算机程序指令集提供为一个唯一加密函数。

根据本发明的一个方面，这些定制指令实现一种加密算法。

根据本发明的另一个方面，该加密算法是序列密码（stream cipher）算法。

根据本发明的再一个方面，为该函数分配一个唯一标识符。

根据本发明的另一个方面，该唯一加密函数实现在一个移动代理中。

根据本发明的再一个方面，一种代理监控系统（AMS）与该唯一移动代理结合使用。该 AMS 监控该唯一移动代理的状态和生命期。该代理向该 AMS 注册，并且在一个数据库中存储唯一移动代理信息。该 AMS 与该代理建立通信协议，从而允许该 AMS 和该代理之间交换信息。该 AMS 也可在当前代理的生命期结束时分配一个新代理来完成该代理的任务，或者在一个时间段上分配和跟踪多个代理，在各同时执行的代理生命期结束时取代它。

应当明白的是，前面的概要描述和下面的详细描述都只是示例性的，而非用于限制本发明。

附图简述

通过下面结合附图的详细描述，可以更好地理解本发明。需要强调的是，根据通常的做法，图中的各种特征没有按比例来画。相反，为清楚起见，将各种特征的尺寸进行了任意的放大或缩小。附图包括：

图 1 是用于描述根据本发明的移动代理的产生和移动的网络框图；

图 2 示出了移动代理组成部分的功能框图；

图 3 是用于描述一个代理在生命周期中旅行于多个主机间时的移动的另一网络框图。

具体实施方式

图 1 是一个示出了穿越于若干宿主消费者计算机的移动代理的生命期及其移动的网络框图。该移动代理由在源计算机 110 上执行的委托人程序 112 产生。虽然该委托人通常是一个人，但其被显示为委托人程序 112，并且由人使用该程序去初始化并且使用该程序与该移动代理通信。

委托人程序 112 使用一个代理模板 114 来产生移动代理。在本示例中，移动代理 A_0 116 是从该代理模板 114 生成的第一个代理。然后，该移动代理 A_0 116 经由网络通信路径 120 被发送到第一个消费者 Host_1 130。到达消费者 Host_1 130 后，移动代理 A_0 116 在 Host_1 上执行，并通过网络路径 122 与委托人 112 通信。在 Host_1 130 上完成其任务之后，移动代理 A_0 116 通过网络路径 124 旅行到 Host_2 140。到达 Host_2 140 后，移动代理 A_0 116 再次执行，并通过网络路径 126 与委托人 112 通信。当代理自己终止时，理想情况是，该代理先将其收集到的任何数据上载到委托人 112，并通知委托人它正在终止，从而使该委托人可以产生一个新代理去完成该任务。

还为移动代理 A_0 116 分配一个有限的生命期。只要代理的生命期没有结束并且其没有被破坏，它将继续其使命。在本发明的示例性实施例中，代理的生命期可由静态程序代码确定，在这种情况下，每个移动代理具有相同的有限生命期。或者，该生命期可由动态程序代码确定，并且可以为每个代理分配一个从预定范围内选择的随机生命期。生命期可以是一个时间量，或者是该代理被允许访问的主机的数量。当代理超过其生命期时，它可以将自己终止或由委托人 112 终止。

在本示例中，在消费者 Host_i 150 处，移动代理 A_0 116 超出了其预定的生命期并自己结束。委托人 112 获知此事并产生一个同样基于

相同代理模板 114 但具有与代理 116 不同动态代码的新移动代理 A_1 162。委托人 112 通过网络路径 134 将新移动代理 A_1 162 发送到 $Host_{i+1}$ 160, 以继续该移动代理的任务。代理 A_1 162 继续执行委托人 112 的任务直至其生命期终止。如果代理 A_1 162 在任务完成前生命期结束, 委托人 112 分别产生其他代理 (A_2 、 A_3 等...) 来完成任务。每个代理具有不同的动态代码。

图 3 示出了移动代理如何在主机间旅行或移动的一个网络框图。移动代理 A_0 116 由委托人 112 创建, 并被发送到第一个 $Host_1$ 130。在这里, 该代理在该主机上执行代码并且可能收集一些数据。可以将该数据存储在移动代理 A_0 116 中。也可以将该数据发送回委托人 112 或者下面描述的代理监控系统 (AMS)。该 AMS 是一个可以使用数据库来监控移动代理状态的软件程序。该 AMS 或委托人 112 可指示移动代理 A_0 116 绕开 $Host_2$ 140 立即到达 $Host_i$ 150, 如虚线箭头 121 所示。或者, 该代理可自主地操作, 并顺序地访问各个主机 130、140、150 和 160。

如果移动代理 A_0 116 工作在自主模式下, 该代理根据已收集数据, 自主决定是移动到 $Host_2$ 140 还是绕开 $Hosts_2$ 140 立即跳到 $Host_i$ 150, 如虚线箭头 121 所示。如果该代理不是处于自主模式下, 它等待来自委托人 110 或 AMS 的指令。

一个运行不佳的网络也可能妨碍代理执行其功能的能力。在该示例性实施例中, 委托人 112 或 AMS 决定是否在网络性能提高而等待一段额外的时间, 还是发动一个额外的移动代理来完成已分配的任务。该 AMS 或委托人 112 可以 “ping” 该移动代理所位于的主机消费者, 并确定是否存在网络传播延迟。“ping” 是一个通过向目的地发送一个或重复的 ICMP (网间控制报文协议) 回应请求并等待答复从而测试目的地的可达性的程序。

图 2 示出了可以构成新的唯一代理的模板的组成部分的模型。该

代理模板有两个部分：(1) 静态代码和数据 212，以及 (2) 动态代码和数据 214。将静态代码和数据 212 不做任何修改地直接放入移动代理 222 的新程序代码中。动态代码和数据 214 首先被传给一个代码生成器 218，以产生新的动态代码 224，新的动态代码 224 也成为新唯一代理 222 的一部分。在本示例性实施例中，该代码生成器 218 也可产生对每个唯一代理不同的唯一标识符 210。

在本发明的示例性实施例中，该唯一标识符与分配给该代理的动态代码相关联。代理将该唯一标识符与发送给委托人的每条消息一起发送。根据此消息，委托人可以知道该代理使用的动态代码，并能将从该代理接收的任何消息正确地解码或验证。

如前所述，全自动代码生成本来是很困难的，在本示例性实施例中，对此问题的解决方法是仅自动产生加密函数。因为自动生成数学函数比生成任意代码容易，所以这是有优势的。由于加密函数能用于将代理的私有数据进行加密和解密，所以其对安全性是很重要的。

根据本发明一个示例性实施例的唯一移动代理使用序列密码 (stream cipher) 作为加密函数。由于将两个或更多个序列密码函数结合起来至少与组合密码分组中最强的密码一样安全，所以序列密码对软件加密来说是可行的。因此，一个现有密码(例如，诸如 DES 这样的分组密码)与另一个密码的结合起来形成序列密码，可以实现至少与 DES 一样强的编码。

序列密码的基本概念是使用一个密钥序列 ($Z=z_1, z_2, \dots$) 将随机代码生成器 216 中的动态代码和数据 214 进行加密。“当前”(第 i 个) 密钥 z_i 是基于前一个密钥 z_{i-1} 和“当前”动态代码和数据 214 而产生的，如公式 (1) 所示：

$$x_{i-1} : z_i = f(z_{i-1}, x_{i-1}) \quad (1)$$

新的动态代码 220 y_i 由“当前”第 i 个密钥 z_i 产生，加密函数如公式 (2) 所示：

$$y_i = e_{z_i}(x_i) \quad (2)$$

计算加密过程中所使用函数的反函数执行解密。然后，通过第 i 个密钥 z_i 和加密函数的反函数来重新生成动态代码和数据 214，如公式 (3) 所示：

$$x_i = d_{z_i}(y_i) \quad (3)$$

为便于自动构造一个多函数 (multi-functioned) 的序列密码，本发明的示例性实施例使用了一个序列密码代码模板作为代码生成引擎 218 的基础。新代码是基于该模板的结构产生的，在本示例性实施例中，该模板是用 Java 写成的，其包括两部分：静态部分和动态部分。静态部分由出现在每个生成实例中出现的 Java 代码组成。动态部分是从静态部分中解析出来的，并通过使用一组标签来对其进行标识。在本示例性实施例中，标签是尖括号 “ $\langle$ ” 和 “ $\rangle$ ”。然后，这些标签被新的、动态生成的代码取代。因此，该模板是我们的代码生成引擎用来自动构建新代码而使用的数据。

作为表 1 中的伪码示出了该序列密码模板的一部分。在此模板内，用于计算内部状态 (S-Box)、加密和解密的函数用标签表示。每个代码实例的函数都是随机产生的。

表 1:

//一元、二元操作定义，如加、平方和、异或、交换。

$\langle$ DEFINITION_OF_NEEDED_OPERATIONS $\rangle$

void calculateInternalStates()

```
{
    if(bPlainTextInvolved)
    {
        int i=(top==0)?(N_PLAIN_TEXT-1):(top-1);
        internalStates[0]=calc(involvedPlainText[i], internalStates[0]);
    }
```

```
else
    internalStates[0]=linearFunction(internalStates[0]);
for(int i=1; i<nInternalStates; i++)
{
    internalStates[i]=calc(internalStates[i-1], internalStates[i]);
}
}
int calc(int x, int y)
{
    <MULTI_OPERATION_ON_X_Y_SBOX>
}
int encrypt(int key, int x)
{
    return <ENCRYPT_OPERATION>;
}
int decrypt(int key, int y)
{
    return <DECRYPT_OPERATION>;
}
```

在一个示例性实施例中，代码生成器 216 负责根据该模板产生新代码。代码生成器 216 由 3 部分组成：模板解析器、标签代码生成器和写入器。

模板解析器通过定位和提取所有标签来解析模板文件。该模板解析器也为所有标签调用相应的标签代码生成器类。然后，标签代码生成器根据标签的分类，动态地生成每个标签的代码。本系统的一个特征在于，可以很容易地替换或修改生成算法。写入器负责用新代码取代标签，然后将结果写进一个新文件中。

在本示例性实施例中，为新动态代码的每个实例分配一个新的、唯一的名字，以将包含此代码的代理与其他代理区分开。该名字用于代理管理和验证。由于该名字只对代理的委托人来说是有意义的，所以，该名字在代理的委托人看来是唯一的就足够了。名字由代码生成器 218 选择和控制。

在本发明的示例性实施例中使用了序列密码算法。该算法首先随机确定在密钥序列例程中使用的操作（函数）的数量，即数值 n 。然后，从函数集合 F 中随机选取 n 个函数。例如，该函数集合可包括诸如交换、异或（XOR）以及平方和这样的函数。为提高效率，根据操作数数量，将这些函数进行分组。为每一个操作（函数）选择操作数。操作数可从内部状态、以前的动态代码数据 214 或以前的密钥中选择。为了获得更高的效率，该内部状态、以前的动态代码和数据 214 以及密钥按固定格式存储在一个数组中。通过为每个动态代码或数据集选择操作和操作数，然后将产生的源代码替换进该模板文件中，从而生成新代码。

为保证随机产生的加密例程是可解密的，用于产生加密代码的函数集合 G 中的每个函数 F 都具有一个在反函数集合 G^{-1} 中的反函数 F^{-1} 。为了生成一个随机产生的序列密码加密解密例程，使用集合 G^{-1} 而不是 G 来执行上述段落中的步骤。增加最终步骤（5），对于随机产生的加密例程，按倒序应用适当的反函数，从而构造解密算法。

在本发明的另一个示例性实施例中，动态代码和数据 220 可通过散列（hash）算法方法得到保护。该方法产生动态代码和数据 220 以及形成消息摘要的消息（可选的）的简明（condensed）表达。然后，例如，可将移动代理的信息内容提供给一个产生或验证该信息签名的签名算法。由于消息摘要通常比消息小很多，所以用摘要给消息签名而不是加密该消息通常能提高处理效率。数字签名的验证者可以像数字签名的创建者一样，使用相同的散列算法。传送（in transit）中对

该消息或用于将消息散列化的算法的任何改变都将产生不同的消息摘要，从而导致签名验证失败。

因为找到与给定消息摘要对应的消息或者找到产生相同消息摘要的两个不同消息在计算上是不可实现的，所以这种类型的认证是安全的。传送中对消息的任何改动将很可能产生不同的消息摘要，从而使得签名验证失败。

不用委托人 112 跟踪移动代理 116 的状态和进度，可以采用代理监控系统 (AMS) 来监控该代理。该 AMS 可以是运行在与委托人 110 相同的计算机上或其他任何能通过网络与该代理通信的计算机上的一个应用程序。

在发布唯一移动代理前，将其注册在 AMS 中。然后，在该移动代理整个生命期内，当其在全球信息网络上从主机移动到主机时，该 AMS 跟踪该移动代理。该唯一移动代理可在某点与该 AMS 建立数据交换。这些数据交换可以下载或上传信息，例如，通知代理有额外的新主机要访问、阻止代理移向已知的恶意主机、报告已访问过的主机或将代理搜索的部分结果发送给该 AMS。

委托人或 AMS 可采用检测技术确定一个主机是否为恶意的。示例性的恶意主机检测算法包括：(1) 指定一个具体的时间范围，在该时间范围内，AMS (或委托人 110) 在唯一移动代理到达一个新主机后期待来自该代理的到达确认信息，如果在此时间范围内没有收到信息则将此主机标注为恶意主机；(2) 检测出该唯一移动代理过早地停止与 AMS 或委托人的通信；以及 (3) AMS 或委托人确定从唯一移动代理收到不正确加密的消息。

可通过使用动态产生的一次性密码 (one-time password)，将动态产生加密代码的函数和有限生命期结合起来。L. Lamport 撰写的题为“Password Authentication with Insecure Communications”一文描述了一次性密码的思想，该文章发表于“*Communications of the ACM*”，

vol.24, No.11, pp 770-772, 1981。该思想被设计成用于对抗基于窃听网络连接来获得登录 id 和密码的攻击。为了使用一次性密码机制，委托人首先选择一个密码并将其存储在验证服务器中。服务器选择一个数值 n （某个相当大的数字）并递归地将该密码做 n 次散列处理（也就是说，用某散列函数计算 hash^n （密码）），然后将该结果与该代理的唯一标识和数值 n 一起存储在该服务器的数据库中。数值 n 代表该代理可使用一次性密码的数量，即该代理在生命期结束前能够发送的消息数量。每次使用时，散列函数少执行一次，分别产生不同的密码。如果代理发送的消息多于 n ，代理生命期结束时，委托人将忽略它们。

用于一次性密码的动态代码生成也可用于产生散列函数，该代理使用所述散列函数来生成密码。在这种情况下，可能希望在选择过程中使用伪随机数生成器从一组已知函数中动态地选择一个散列函数，而不是动态地生成一个新的散列函数。也可使用动态代码产生算法来选择用于产生散列化密码的初始密码。本示例中的密码不一定用作密码，也可用作加密密钥，在将数据发送给委托人 112 或 AMS 前将其加密。收到加密数据后，委托人或 AMS 可根据该唯一代理标识符查找当前密钥，然后，应用一个解密函数，以将该消息解密，该解密函数与用于将该数据加密的函数相对应。

安全移动代理在安全信息收集的很多领域有很大的潜力。它们可用作通过信息网进行搜索的蜘蛛 (spider)，或者，它们具有能通过非安全网络安全地访问数据库的嵌入式 ODBC(开放数据库互连)消息。移动代理也可用于执行旅行计划、互联网网络监控和在企业环境的计算机中查找非法、未授权或未注册的软件拷贝。

上面结合具体的实施例和例子对本发明进行了介绍和描述，但本发明不限于上述给出的细节。相反，在不脱离本发明的前提下，可以在权利要求等价物的范围内做出各种细节性修改。例如，如上所述，加密函数的动态代码生成可用于移动代理通信之外的其他应用。驻留

在个人计算机的任何应用通常都是容易受到反向工程攻击的，反向工程攻击最终将突破这些应用的防篡改特征。如果本发明的可更新防篡改加密函数用于这些应用，攻击者将需要在每次更新该函数时破解它。这使得想要尝试破解代码的人需要做相当多的工作，因此，将鼓励人们使用这些应用，就如同人们对它们预期的那样。

例如，如果一个动态生成的序列密码用于将临时存储然后例如使用诸如 TIVO™ 或 Replay TV™ 系统这样的个人录像机来播放的内容进行加密。在该应用中，可将该动态序列密码进行解码的应用随同内容一起发送，观众可使用它将该内容进行解码和显示。由于下次发送给观众的加密内容将使用动态产生的不同的密码，所以计算机黑客无法从破解该序列密码中获益很多。

此外，如上所述，可更新的生成的加密函数也可用于验证个人计算机上用户的身份。由于加密函数不断地被更新，因此攻击者无法持续访问受保护的材料。可更新加密函数也用于确保日志文件或审计踪迹的完整性，这使得数据库的恶意用户更难隐藏在不同时间或不同地点对数据做出的非法修改。

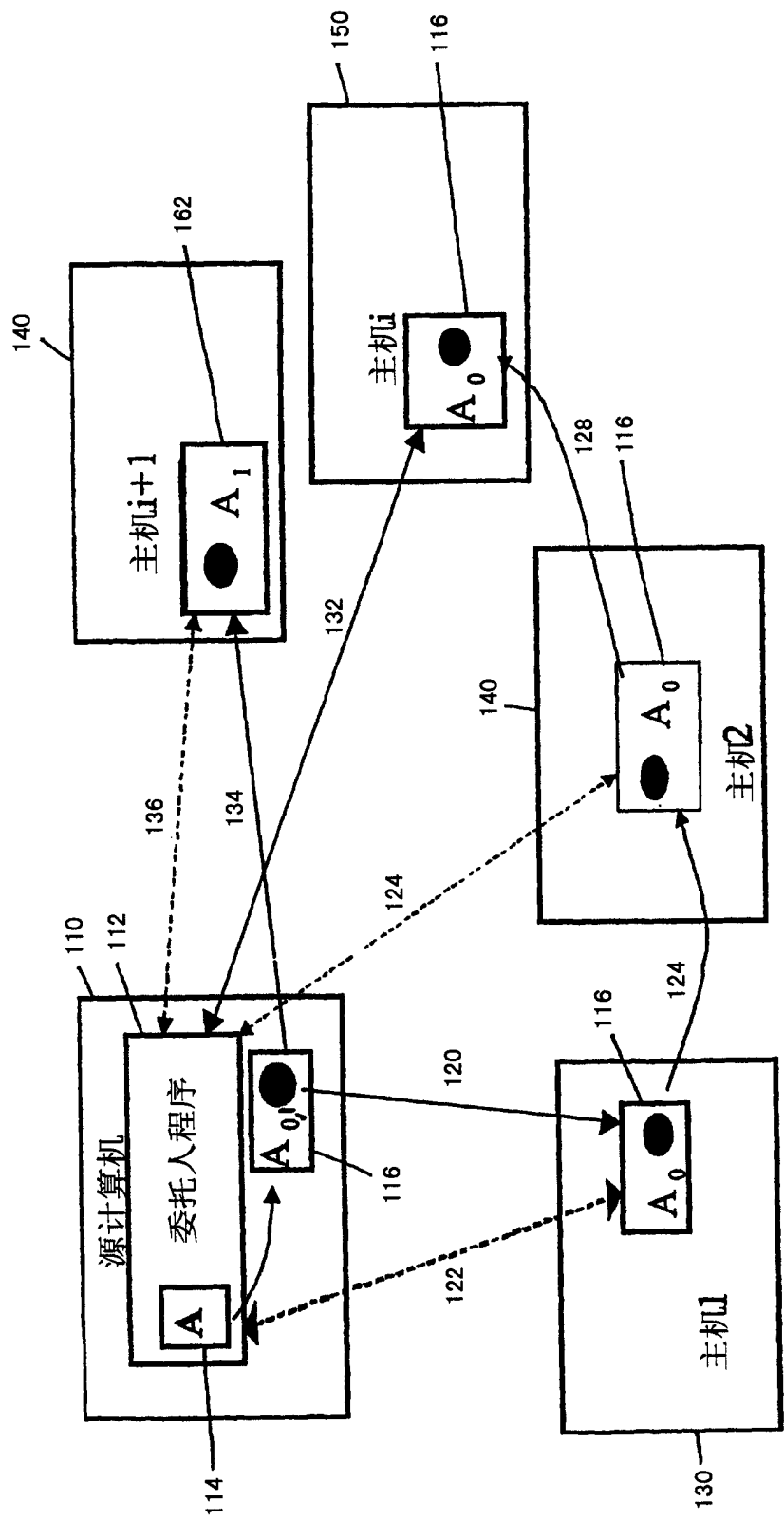


图1

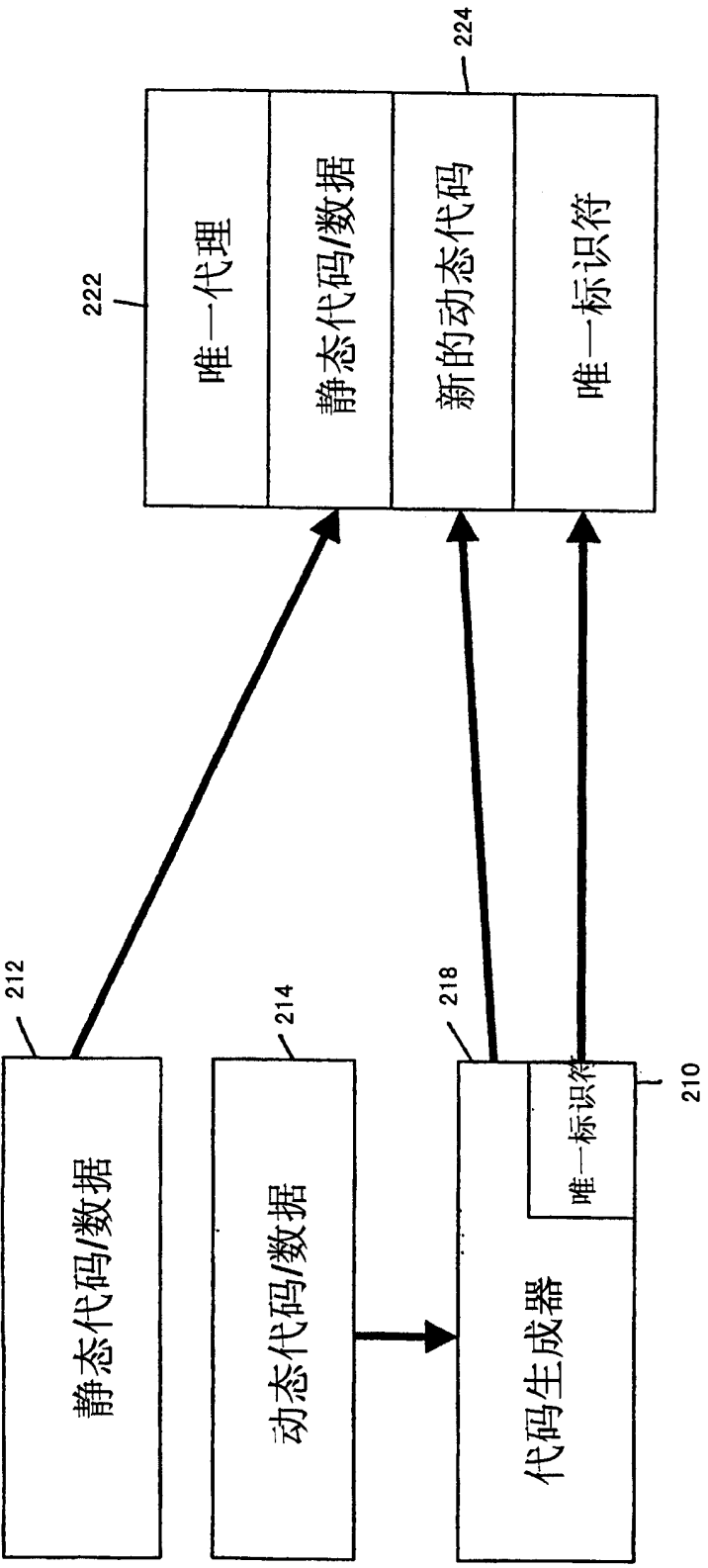


图2

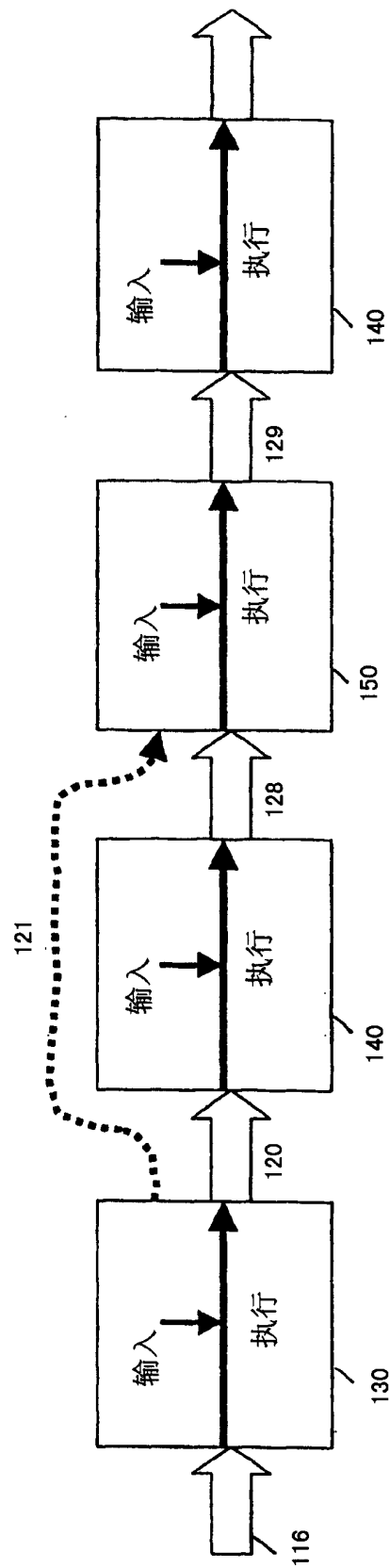


图3