[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 29/06 (2006.01)
H04L 12/28 (2006.01)

[12] 发明专利说明书

专利号 ZL 02815510.6

[45] 授权公告日 2009 年 4 月 15 日

[11] 授权公告号 CN 100479451C

[22] 申请日 2002.7.10 [21] 申请号 02815510.6

[30] 优先权

[32] 2001.7.10 [33] US [31] 09/902,770

[86] 国际申请 PCT/IB2002/002720 2002.7.10

[87] 国际公布 WO2003/007524 英 2003.1.23

[85] 进入国家阶段日期 2004.2.6

[73] 专利权人 意大利电信股份公司

地址 意大利米兰

[72] 发明人 亚尼夫·沙菲拉 卓瑞·肖哈特 摩施·泽扎克 尼夫·格尔博

[56] 参考文献

US 6061797A 2000.5.9

US 5473607A 1995.12.5

Security Architecture for the Internet Protocol. S. Knet, BBN Corp, R. Atkinson, 8. 21, Network Working Group Request for Comments: 2401. 1998

审查员 成 谦

[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

代理人 董 莘

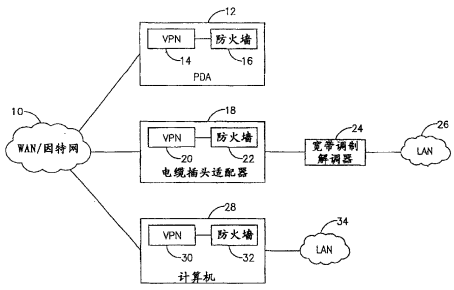
权利要求书 8 页 说明书 31 页 附图 15 页

[54] 发明名称

安全性关联的方法和便携式计算设备

[57] 摘要

一种新颖的有用虚拟专用网络(VPN)机制和相关的安全性关联处理器，用于保持实现诸如加密、解密和鉴权之类安全功能所需的安全参数。 安全性关联数据库(SAD)和相关电路适合于提供实现关于加密/解密和鉴权的 IPSec 组安全规范必需的参数。数据库中的每个安全性关联(SA)包括根据 IPSec 规范接收和传送 VPN 规范所需的全部参数。



1、一种便携式计算设备，包括：

用于连接所述设备和通信网络的通信装置；

包括易失性和非易失性存储器的存储器装置，所述非易失性存储器用于存储程序代码；

与所述存储器装置和所述通信装置耦接、执行所述程序代码的处理器；

包含会话数据库的分组过滤器，所述会话数据库包括用于存储多个通信会话的会话相关数据的会话记录；和

虚拟专用网络电路，所述虚拟专用网络电路包括：

存储多个用于安全性关联的安全数据的安全性关联数据库装置，每个入口包括和唯一套接字对应的安全性关联数据；

多个安全引擎，每个安全引擎用于执行安全处理；

当收到未在所述安全性关联数据库装置中找到的套接字时，打开新的安全性关联的装置；

根据输入分组的套接字，搜索和识别与之相关的安全性关联的装置；

从所述安全性关联数据库装置检索多个安全参数的装置；

将所述多个安全参数转发给至少一个所述安全引擎，以便以此执行安全处理的装置；和

用于利用所述输入分组和所述安全处理的结果，根据特定的安全模式，构建输出分组的分组建立装置；

其中所述会话数据库被配置成使得与会话相关的多个会话记录，对于所述安全性关联数据库装置的同一个入口使用相同的安全性关联指针；

其中所述虚拟专用网络电路还包括：

从所述安全性关联数据库装置中去除未使用的安全性关联的装置，和

在对应的安全性关联被从所述安全性关联数据库装置中去除时，从所述会话数据库中删除与使用相同安全性关联的所有会话相关的会话记录的装置。

2、按照权利要求 1 所述的便携式计算设备，其中所述通信网络包括广域网。

3、按照权利要求 1 所述的便携式计算设备，其中所述通信网络包括局域网。

4、按照权利要求 1 所述的便携式计算设备，其中所述通信网络包括因特网。

5、按照权利要求 1 所述的便携式计算设备，其中所述通信网络包括公共交换电话网。

6、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于进行加密。

7、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于进行解密。

8、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于进行鉴权。

9、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于执行 IPSec 服务。

10、按照权利要求 1 所述的便携式计算设备，其中利用专用集成电

路实现所述虚拟专用网络电路。

11、按照权利要求 1 所述的便携式计算设备，其中利用现场可编程门阵列实现所述虚拟专用网络电路。

12、按照权利要求 1 所述的便携式计算设备，其中利用数字信号处理器实现所述虚拟专用网络电路。

13、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括根据所述安全处理的结果，更新所述安全性关联数据库的内容的装置。

14、按照权利要求 1 所述的便携式计算设备，其中与所述安全性关联相关的参数由位于所述虚拟专用网络电路之外的实体配置，

其中所述虚拟专用网络电路包括将对应于所述安全性关联数据库中的新的安全性关联的安全数据，以及根据与新的安全性关联相关的套接字所计算的散列值，保存在所述安全性关联数据库装置中的装置。

15、按照权利要求 1 所述的便携式计算设备，其中所述安全性关联由位于所述虚拟专用网络电路之外的实体打开，

其中所述虚拟专用网络电路包括将所述新安全性关联的指针插入最近最少使用链接列表中的装置。

16、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括当超过最大超时时，从所述安全性关联数据库装置中去除未使用的安全性关联的装置。

17、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括当超过最大字节计数时，从所述安全性关联数据库装置中

去除未使用的安全性关联的装置。

18、按照权利要求 1 所述的便携式计算设备，其中所述搜索和识别安全性关联的装置包括:

根据与将被识别的安全性关联相关的套接字来计算散列值的装置;

利用散列结果作为索引，在散列表中查寻散列指针的装置;

根据所述散列指针，从所述安全性关联数据库检索数据的装置;

如果所检索的数据和与分组相关的套接字相符，则识别所述安全性关联的装置。

19、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括对入站分组应用防重放机制的装置。

20、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括跟踪入站分组的序列号的装置。

21、按照权利要求 1 所述的便携式计算设备，其中所述虚拟专用网络电路还包括建立并保持具有头部和尾部的最近最少使用双重链接列表的装置，

其中最近最多使用的安全性关联保存在尾部，最近最少使用的安全性关联保存在头部。

22、按照权利要求 21 所述的便携式计算设备，其中在最近最少使用列表充满的情况下，删除位于头部的安全性关联，将新的安全性关联加在尾部。

23、按照权利要求 1 所述的便携式计算设备，其中所述套接字包括安全参数索引、远程 IP 和协议组件。

24、按照权利要求 1 所述的便携式计算设备，其中所述安全性关联数据包括下述值任意之一或者它们的组合：IPSec 模式、加密算法、加密密钥。

25、按照权利要求 1 所述的便携式计算设备，其中所述安全性关联数据包括下述值任意之一或者它们的组合：IPSec 模式、鉴权算法、鉴权密钥。

26、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于实现 IPSec 隧道模式服务。

27、按照权利要求 1 所述的便携式计算设备，其中至少一个所述安全引擎用于实现 IPSec 传送模式服务。

28、一种安全性关联的方法，所述方法包括下述步骤：

建立用于存储多个安全性关联的安全数据的安全性关联数据库，所述安全性关联数据库内的每个入口对应于一个套接字；

当收到未在所述安全性关联数据库中找到的套接字时，打开新的安全性关联；

根据分组的套接字，搜索和识别与之相关的安全性关联；

从所述安全性关联数据库检索多个安全参数；和

将所述多个安全参数转发给虚拟专用网络安全处理器，以便以此执行一个或多个安全处理；

所述方法还包括步骤：

构建会话数据库，所述会话数据库包括用于存储多个通信会话的会话相关数据的会话记录，其中所述会话数据库被配置成使得与会话相关的多个会话记录，对于所述安全性关联数据库装置的同一个入口使用相同的安全性关联指针；

从所述安全性关联数据库中去除未使用的安全性关联，并且在对应

的安全性关联被从所述安全性关联数据库中去除时，从所述会话数据库中删除与使用相同安全性关联的所有会话相关的会话记录。

29、按照权利要求 28 所述的方法，还包括根据所述安全处理的结果，更新所述安全性关联数据库的内容的步骤。

30、按照权利要求 28 所述的方法，其中打开新的安全性关联的所述步骤包括:

将对应于所述新安全性关联的安全数据存储在所述安全性关联数据库中;

根据与所述新的安全性关联相关的套接字来计算散列值；和

将所述散列值存储在散列表中。

31、按照权利要求 28 所述的方法，其中所述打开新的安全性关联的步骤包括将所述新安全性关联的指针插入最近最少使用链接列表中。

32、按照权利要求 28 所述的方法，还包括当超过最大超时时，从所述安全性关联数据库去除未使用的安全性关联的步骤。

33、按照权利要求 28 所述的方法，还包括当超过最大字节计数时，从所述安全性关联数据库去除未使用的安全性关联的步骤。

34、按照权利要求 28 所述的方法，其中所述搜索并识别安全性关联的步骤包括下述步骤:

根据与将被识别的安全性关联相关的套接字来计算散列值;

利用散列结果作为索引，在散列表中查寻散列指针;

根据所述散列指针，从所述安全性关联数据库中检索数据;

如果所检索的数据和与分组相关的套接字相符，则识别所述安全性关联。

35、按照权利要求 28 所述的方法，其中所述虚拟专用网络安全处理器用于进行加密。

36、按照权利要求 28 所述的方法，其中所述虚拟专用网络安全处理器用于进行解密。

37、按照权利要求 28 所述的方法，其中所述虚拟专用网络安全处理器用于进行鉴权。

38、按照权利要求 28 所述的方法，其中所述虚拟专用网络安全处理器用于执行 IPSec 规定服务。

39、按照权利要求 28 所述的方法，还包括对从远程网络接收的分组应用防重放机制的步骤。

40、按照权利要求 28 所述的方法，还包括跟踪从远程网络接收的分组的序列号的步骤。

41、按照权利要求 28 所述的方法，还包括建立并保持具有头部和尾部的最近最少使用双重链接列表的步骤，其中最近最多使用的安全性关联存储在尾部，最近最少使用的安全性关联存储在头部。

42、按照权利要求 41 所述的方法，其中在最近最少使用列表充满的情况下，删除位于头部的安全性关联，将新的安全性关联加在尾部。

43、按照权利要求 28 所述的方法，其中所述套接字包括安全参数索引、远程 IP 和协议组件。

44、按照权利要求 28 所述的方法，其中所述安全性关联数据包括下述值任意之一或者它们的组合：IPSec 模式、加密算法、加密密钥。

45、按照权利要求 28 所述的方法，其中所述安全性关联数据包括下述值任意之一或者它们的组合：IPSec 模式、鉴权算法、鉴权密钥。

46、按照权利要求 28 所述的方法，还包括如果从所述虚拟专用网络安全处理器收到误码，则拒绝所述分组的步骤。

47、按照权利要求 28 所述的方法，其中利用专用集成电路实现所述方法。

48、按照权利要求 28 所述的方法，其中利用现场可编程门阵列实现所述方法。

49、按照权利要求 28 所述的方法，其中利用数字信号处理器实现所述方法。

安全性关联的方法和便携式计算设备

技术领域

本发明涉及数据通信系统，更具体地说涉及实现包括安全性关联数据库和相关处理器的虚拟专用网络（VPN）的机制。

背景技术

近年来，全球目睹了因特网的爆发性增长。每年越来越多的主机加入因特网，同时用户数目似乎无限制地增长。因特网能够实现利用不同技术（包括远程计算机登录、文件传送、万维网（WWW）浏览、电子邮件等）的通信。设计出各种各样的协议，并且这些协议在因特网上用于处理各种通信。例如，文件传送协议（FTP）用于文件传送，超文本置标语言（HTML）用于 web 通信等。通常，在包括 OSI 通信栈各层协议在内的协议的传输控制协议/网际协议（TCP/IP）组的保护下，对和因特网通信相关的协议分组。

因特网的关键特征在于它是几乎具有计算机、电话线和因特网服务提供者（ISP）账户的任何人可访问的公共网络。这种大规模公众可接入性的最终趋势是它便于黑客和意图对因特网上的一个或多个主机采取恶意行动的其它人接近因特网上的一个或多个主机。对于设法侵入远程网络的计算机并成功窃取通信数据的黑客来说，诸如恶意用户窃取保密信息或者删除重要文件之类非法行为是可能的。通过在允许因特网上的安全事务的 IPv6 中包括诸如加密和鉴权之类安全特征，因特网体系结构委员会解决了关于安全性的需要。

为了对抗黑客的威胁和保护专用网络，目前通常在公司或机制的专用网络的入口处设置防火墙。防火墙采用某一形式的分组过滤器，分组过滤器强制实现用户定义的安全策略。防火墙是位于机制的本地网络和全球因特网之间边界处的系统。它实现所有数据通信的过滤，以便防止

信息泄漏到外部网络，或者防止从外部未经授权访问内部网络。防火墙对接收的每个分组进行拒绝/许可判定。

同时，对无线服务（即蜂窝电话机、双向寻呼机、无绳设备等）和诸如膝上型计算机、PDA 之类个人计算设备的需求日益增多。这些个人计算设备中的多数都包含无线通信电路，以便它们能够通过无线网络（例如蜂窝或者其它宽带方案）与诸如因特网之类 WAN 网络通信。从而，越来越多的 PDA 和蜂窝电话机正在连接到因特网上，从而这些设备存在安全风险。这些设备最好采用某一类型的防火墙防范对该设备的未经授权的访问。但是，目前多数防火墙以软件形式实现，需要整个桌上型计算机的计算资源，导致在诸如蜂窝电话机或 PDA 之类便携式计算设备中使用它们的成本很高或者不切实际。

从而，需要一种易于用适于包含在诸如蜂窝电话机和无线连接 PDA 之类小型便携式电子计算设备中的小尺寸硬件实现的防火墙或分组过滤器。

发明内容

本发明提供一种新颖的有用虚拟专用网络（VPN）机制，以便提供实现加密/解密和鉴权必需的安全参数。VPN 机制适合于以较低的成本用硬件实现，从而能够费用低廉地将本发明结合到与因特网或其它广域网连接的便携式电子通信设备中，例如蜂窝电话机、个人数字助手（PDA）、膝上型计算机等。

本发明可和适于连接因特网的便携式计算设备，例如蜂窝电话机和无线连接 PDA 中基于硬件或软件的防火墙一起使用。还可用软件或硬件和软件的组合实现本发明的 VPN 机制。

从而，VPN 机制可被用于实现，例如通过 WAN 的安全分局（branch office）连接性、通过 WAN 的安全远程访问或者通过 WAN 的安全电子商务交易。

VPN 机制包括适于提供实现加密/解密和鉴权的安全性规范的 IPSec 组所需参数的安全性关联数据库（SAD）和相关电路。数据库中

的每个安全性关联（SA）入口包括根据 IPSec 规范接收和传送 VPN 分组所需的所有参数。

本发明对输入分组流进行涉及安全性关联（SA）的处理。注意输入分组流可包括入站分组和出站分组。通常，本发明被置于 WAN（即因特网）和本地 LAN 之间。这种情况下，VPN 机制过滤从 WAN 发送给 LAN 的入站分组和从 LNA 发送给 WAN 的出站分组。

本发明的 VPN 机制保持称为 SAD 的安全参数表，用于保存和单向连接相关的安全参数。对于双向连接，需要产生两个 SA。新的安全性关联被添加到 SA 数据库中，并且一旦被产生，根据关于特定 SA 保存在数据库中的参数，处理与该 SA 相关的后续分组。根据该分组和 SA 参数，分组可被加密、解密、鉴权或丢弃。注意只有当分组符合在 SA 数据库中规定的安全性时，才许可该分组。

虽然本发明特别适合于用硬件实现，不过也可用软件实现本发明。在一个实施例中，包括处理器、存储器等的计算机执行适于实现本发明的 VPN 机制和安全性关联处理的软件。

根据本发明的一个方面，提供了一种便携式计算设备，包括:

用于连接所述设备和通信网络的通信装置;

包括易失性和非易失性存储器的存储器装置，所述非易失性存储器用于存储程序代码;

与所述存储器装置和所述通信装置耦接、执行所述程序代码的处理器;

包含会话数据库的分组过滤器，所述会话数据库包括用于存储多个通信会话的会话相关数据的会话记录; 和

虚拟专用网络电路，所述虚拟专用网络电路包括:

存储多个用于安全性关联的安全数据的安全性关联数据库装置，每个入口包括和唯一套接字对应的安全性关联数据;

多个安全引擎，每个安全引擎用于执行安全处理;

当收到未在所述安全性关联数据库装置中找到的套接字时，打开新的安全性关联的装置;

根据输入分组的套接字，搜索和识别与之相关的安全性关联的装置；

从所述安全性关联数据库装置检索多个安全参数的装置；

将所述多个安全参数转发给至少一个所述安全引擎，以便以此执行安全处理的装置；和

用于利用所述输入分组和所述安全处理的结果，根据特定的安全模式，构建输出分组的分组建立装置；

其中所述会话数据库被配置成使得与会话相关的多个会话记录，对于所述安全性关联数据库装置的同一个入口使用相同的安全性关联指针；

其中所述虚拟专用网络电路还包括：

从所述安全性关联数据库装置中去除未使用的安全性关联，和在对应的安全性关联被从所述安全性关联数据库装置中去除时，从所述会话数据库中删除与使用相同安全性关联的所有会话相关的会话记录的装置。

根据本发明的另一个方面，还提供了一种安全性关联的方法，所述方法包括下述步骤：

建立用于存储多个安全性关联的安全数据的安全性关联数据库，所述安全性关联数据库内的每个入口对应于一个套接字；

当收到未在所述安全性关联数据库中找到的套接字时，打开新的安全性关联；

根据分组的套接字，搜索和识别与之相关的安全性关联；

从所述安全性关联数据库检索多个安全参数；和

将所述多个安全参数转发给虚拟专用网络安全处理器，以便以此执行一个或多个安全处理；

所述方法还包括步骤：

构建会话数据库，所述会话数据库包括用于存储多个通信会话的会话相关数据的会话记录，其中所述会话数据库被配置成使得与会话相关的多个会话记录，对于所述安全性关联数据库装置的同一个入口使用相

同的安全性关联指针;

从所述安全性关联数据库中去除未使用的安全性关联，并且在对应的安全性关联被从所述安全性关联数据库中去除时，从所述会话数据库中删除与使用相同安全性关联的所有会话相关的会话记录。

根据本发明，提供一种安全性关联处理器电路，包括存储多个安全性关联的安全数据的安全性关联数据库，每个入口包括和唯一套接字对应的安全性关联数据，当收到未在安全性关联数据库中找到的套接字时，打开新的安全性关联的装置，根据分组的套接字搜索和识别与之相关的安全性关联的装置，从安全性关联数据库检索多个安全参数的装置，和将多个安全参数转发给虚拟专用网络（VPN）安全处理器，以便以此执行一个或多个安全处理的装置。

根据本发明，还提供一种虚拟专用网络（VPN）电路，包括保存多个安全性关联的安全数据的安全性关联数据库装置，每个入口包括和唯一套接字对应的安全性关联数据，多个安全引擎，每个安全引擎适合于执行安全处理，当收到未在安全性关联数据库装置中找到的套接字时，打开新的安全性关联的装置，根据输入分组的套接字搜索和识别与之相关的安全性关联的装置，从安全性关联数据库装置检索多个安全参数的装置，将多个安全参数转发给至少一个安全引擎，以便以此执行安全处理的装置，和适合于利用输入分组和安全处理的结果，根据特定的安全模式构建输出分组的分组建立装置。

根据本发明，还提供一种便携式计算设备，包括适于连接该设备和通信网络的通信装置，包括易失性和非易失性存储器的存储器装置，非易失性存储器适于保存程序代码，与存储器装置和通信装置耦接，执行程序代码的处理器，和虚拟专用网络（VPN）电路，所述虚拟专用网络（VPN）电路包括保存多个安全性关联的安全数据的安全性关联数据库装置，每个入口包括和唯一套接字对应的安全性关联数据，多个安全引擎，每个安全引擎适合于执行安全处理，当收到未在安全性关联数据库装置中找到的套接字时，打开新的安全性关联的装置，根据输入分组的套接字搜索和识别与之相关的安全性关联的装置，从安全性关联数据库

装置检索多个安全参数的装置，将多个安全参数转发给至少一个安全引擎，以便以此执行安全处理的装置，和适合于利用输入分组和安全处理的结果，根据特定的安全模式构建输出分组的分组建立装置。

根据本发明，还提供一种安全性关联处理器电路，包括保存多个安全性关联的安全数据的安全性关联数据库，每个入口包括和唯一套接字对应的安全性关联数据，当收到未在安全性关联数据库中找到的套接字时，适于打开新的安全性关联的管理单元，适于根据输入分组的套接字，搜索和识别与之相关的安全性关联的识别单元，适于从安全性关联数据库检索多个安全参数，并将它们转发给虚拟专用网络（VPN）安全处理器，以便以此执行一个或多个安全处理的主处理器单元，和包含散列函数和相关散列表，简化保存的安全性关联的搜索的散列单元。

根据本发明，还提供一种安全性关联的方法，所述方法包括下述步骤：建立适于保存多个安全性关联的安全数据的安全性关联数据库，安全性关联数据库内的每个入口对应于一个套接字，当收到未在安全性关联数据库中找到的套接字时，打开新的安全性关联，根据分组的套接字，搜索和识别与之相关的安全性关联，从安全性关联数据库检索多个安全参数，并将所述多个安全参数转发给虚拟专用网络（VPN）安全处理器，以便以此执行一个或多个安全处理。

根据本发明，还提供一种计算机可读存储介质，具有包含于其中的计算机可读程序代码单元，当在计算机上执行这种程序时，使适当编程的计算机实现安全关联性机制，所述计算机可读存储介质包括使计算机建立适于保存多个安全性关联的安全数据的安全性关联数据库的计算机可读程序代码单元，安全性关联数据库内的每个入口包括对应于唯一套接字的安全性关联数据，当收到未在安全性关联数据库中找到的套接字时，使计算机打开新的安全性关联的计算机可读程序代码单元，使计算机根据分组的套接字，搜索和识别与之相关的安全性关联的计算机可读程序代码单元，使计算机从安全性关联数据库检索多个安全参数的计算机可读程序代码单元，和使计算机将所述多个安全参数转发给虚拟专用网络（VPN）安全处理器，以便以此执行一个或多个安全处理的计算机

可读程序代码单元。

附图说明

下面参考附图，举例说明本发明，其中：

图 1 是图解说明在 WAN 或因特网环境中，本发明的虚拟专用网络机制的几种例证应用的方框图；

图 2 是图解说明包括与本发明的虚拟专用网络机制通信的本地和远程 LAN、WAN 和拨号用户的例证网络的方框图；

图 3 是更详细地图解说明本发明的虚拟专用网络机制的方框图；

图 4A 和 4B 是图解说明本发明的主 SA 处理器方法的流程图；

图 5 图解说明了确定与输入分组相关的 SA 的散列技术；

图 6 是图解说明本发明的 SA 识别过程的散列方法的流程图；

图 7 图解说明了在主 SA 处理器进行的检查中执行的防重放窗口机制；

图 8 是图解说明本发明的防重放窗口方法的流程图；

图 9 图解说明了用于跟踪 SA 失效（staleness）的最近最少使用的链路列表结构；

图 10 是图解说明当识别与输入分组相关的 SA 时，更新 LRU 链接列表的方法的流程图；

图 11 是图解说明本发明的 SA 管理模块的处理的流程图；

图 12 是图解说明由本发明的 SA 管理模块执行的打开 SA 的处理的流程图；

图 13 是图解说明由本发明的 SA 管理模块执行的关闭 SA 的处理的流程图；

图 14 图解说明了与每个 SA 记录相关的最后打开的会话指针；

图 15 图解说明了前一和下一会话指针，和匹配与每个会话记录相关的 SA 指针；

图 16 图解说明单一会话和单一 SA 记录之间的例证关系；

图 17 图解说明多个会话和单一 SA 记录之间的例证关系；

图 18 是图解说明适于实现本发明的 VPN 机制和相关的安全关联处理的例证计算机处理系统-平台的方框图。

具体实施方式

全文中使用的符号

本文献中自始至终使用下述符号

术语	定义
ARP	地址解析协议
ARW	防重放窗口
ASIC	专用集成电路
ADSL	非对称数字用户线
AH	鉴权信头
CPU	中央处理器
CRC	循环冗余检验
CBC	密码分组链接
DES	数据加密标准
DAT	数字音频带
DSP	数字信号处理器
DSL	数字用户线
DVD	数字通用光盘
EEPROM	电可擦可编程只读存储器
EEROM	电可擦除只读存储器
ECB	电子密码本
ESP	封装安全载荷
ED	加密/解密
EPROM	可擦可编程只读存储器
FPGA	现场可编程门阵列
FTP	文件传送协议
HMAC	信头消息鉴权代码

HDSL	高位速率数字用户线
HTML	超文本置标语言
IAB	因特网体系结构委员会
ICMP	因特网控制消息协议
IKE	因特网密钥交换
IP	网际协议
ISP	因特网服务提供商
IV	初始向量
IPSec	IP 安全性
LRU	最近最少使用的
LAN	局域网
MTU	最大传输单元
MAN	城域网
MRU	最近最多使用的
NIC	网络接口卡
OSI	开放系统互连
PB	分组建立器
PC	个人计算机
PDA	个人数字助手
PDU	协议数据单元
RAM	随机存取存储器
RIP	远程 IP
ROM	只读存储器
SA	安全关联
SAP	安全关联处理器
SPD	安全策略数据库
SPI	安全参数索引
TCP	传输控制协议
UDP	用户数据报协议

VDSL 超高位速率数字用户线

VPN 虚拟专用网络

WAN 广域网

WWW 万维网

本发明的详细说明

本发明提供一种新的有用虚拟专用网络（VPN）机制，以便提供执行加密/解密和鉴权所需的和安全相关的参数。VPN 机制适合于以低成本硬件实现，从而使得能够费用低廉地将本发明结合到诸如蜂窝电话机、个人数字助手（PDA）、膝上型计算机之类与因特网或其它广域网连接的便携式电子通信设备中。本发明可和诸如蜂窝电话机和无线连接 PDA 之类适于连接到因特网的便携式计算设备中，基于硬件或软件的防火墙一起使用。也可用软件或者硬件和软件的组合，实现本发明的 VPN 机制。

VPN 机制包括适于提供实现关于加密/解密和鉴权的安全规范的 IPSec 组所需参数的安全关联数据库（SAD）和相关电路。数据库中的每个安全关联（SA）入口包括根据 IPsec 规范接收和传送 VPN 分组所需的全部参数。注意最初在因特网密钥交换（IKE）过程中协商 SA。虽然可用软件实现本发明，不过本发明的 VPN 机制特别适合于用能够提供大大加速的安全处理的硬件实现。用硬件实现 SAD 和相关处理使得能够在硬件中进行所有 IPSec 加密/解密和鉴权处理。注意这适用于其中已建立 SA 的 VPN 会话。要指出的是还未建立 SA 的 VPN 会话非常少，从而对系统性能的影响可忽略不计。

通过使系统能够选择所需的安全协议，选择要使用的算法，并将提供服务所需的密钥投入使用，IPSec 提供 IP 层的安全服务。IPSec 中使用两种协议提供安全性，包括表示成鉴权信头（AH）的鉴权协议和表示成封装安全载荷（ESP）的组合加密/解密协议。安全关联或 SA 用在 AH 和 ESP 机制中。SA 被定义成发送者和向其上携带的通信提供安全服务的接收者之间的单向关系。注意对于双向安全交换来说，需要两个 SA。

本发胆对输入分组流执行和安全关联（SA）相关的处理。注意输入分组流既可包括入站分组又可包括出站分组。通常，本发明的 VPN 机制安置在 WAN（即因特网）和本地 LAN 之间。这种情况下，VPN 机制发挥作用，既过滤从 WAN 发送给 LAN 的入站分组，又过滤从 LAN 发送给 WAN 的出站分组。

注意输入流可包括和特定应用相符的任意类型的输入数据，例如帧、分组、字节、PDU 等。只是出于举例说明的目的，输入数据流被看作是一系列的分组。

要指出的是 VPN 机制和举例说明的相关模块只是作为一个例子给出，并不打算限制本发明的范围。在不脱离本发明的范围的情况下，利用这里说明的本发明的原理，本领域的技术人员可用硬件、软件或者硬件和软件的组合构成其它 VPN 模块，完成与安全性相关的处理和 VPN 实现。

本发明的 VPN 机制保持称为安全关联数据库（SAD）的安全参数表，以便保存和单向连接相关的安全参数。对于双向连接，需要产生两个 SA。新的安全关联被加入 SA 数据库中，并且一旦产生，则根据数据库中保存的关于特定 SA 的参数，处理与该 SA 相关的后续分组。根据分组和 SA 参数，分组可能被加密、解密、鉴权或丢弃。注意只有当分组符合在 SA 数据库中规定的安全性时才许可该分组。

本发明的 VPN 机制在许多不同类型的系统中有着广泛应用。图 1 中表示了图解说明在 WAN 或因特网的环境中，本发明的 VPN 机制的几种例证应用的方框图。一般来说，和防火墙一起构成并使用 VPN 机制，以便防止未经许可访问受保护网络，如这里表示的三个例子中每个例子所示。

在第一例子中，VPN 机制 14 和防火墙 16 一起使用，防火墙 16 实现于与网络 10 进行有线或无线通信的个人计算设备 12 中，网络 10 可以是 WAN 或者因特网。个人计算设备可包括任意设备，例如个人数字助手（PDA）（例如手持式掌上机），蜂窝电话机，无线手持机等。本例中，VPN 机制和相关防火墙对在 WAN/因特网和设备之间流动的分组进

行双向安全处理。

在第二例子中，VPN 机制 20 和防火墙 22 一起使用，从而 VPN 机制 20 和防火墙 22 都实现于与 WAN 或因特网 10 进行有线或无线通信的电缆插头适配器 18 中。设备 18 通常位于 WAN/因特网和宽带调制解调器 24 之间，宽带调制解调器 24 连接 LAN 26 和 WAN/因特网。电缆插头适配器中的 VPN 机制对从 WAN/因特网发送给 LAN 的分组，以及从 LAN 发送给 WAN/因特网的分组进行安全处理。宽带调制解调器适合于调制和解调诸如 xDSL（例如 ADSL、HDSL、VDSL 等）、卫星、陆基 RF、微波之类宽带信号。

在第三例子中，VPN 机制 30 和防火墙 32 一起使用，其中 VPN 机制 30 和防火墙 32 都实现于独立的计算机 28 上，例如个人计算机（PC）、膝上型计算机等，计算机 28 与 WAN 或因特网进行有线或无线通信。包含防火墙的计算机位于 WAN/因特网和要保护的 LAN 34 之间。VPN 机制对从 WAN/因特网发送给 LAN 的分组，以及从 LAN 发送给 WAN/因特网的分组进行安全处理。

图 2 中表示了图解说明包括与本发明的虚拟专用网络机制通信的本地和远程 LAN、WAN 和拨号用户的例证网络的方框图。该例证网络 40 包括两个 LAN，一个本地 LAN 42 和一个远程 LAN 52。三个计算机 A、B 和 C 与该网络连接并相互通信。计算机 A、B 和 C 可包括主机、工作站、用户等。VPN 网关 48 和防火墙 46 位于本地 LAN 的入口处，VPN 网关 56 和防火墙 58 同样位于远程 LAN 的入口处。两个 LAN 通过 WAN/因特网 50 连接。

用户 C 也通过适当的装置，例如拨号调制解调器，xDSL 调制解调器接入等，与 WAN/因特网连接，并且可包括 VPN 最终用户，在 VPN 最终用户中，它包含 VPN 网关或者可以是非 VPN 连接。

根据本发明，VPN 机制适于进行安全处理，包括产生、保存和管理在进行加密、解密、鉴权等情况下，一个或多个安全处理器所需的安全参数。VPN 机制既处理从 WAN/因特网接收的指定给 LAN 的入站分组，又处理源于 LAN，并指定给 WAN/因特网的出站分组。

操作中，从 LAN 输出的明文被 VPN 加密，并通过 WAN 传送。类似地，从 WAN 入站的密文被 VPN 解密成明文，并通过 LAN 转发给接收者。根据本发明，VPN 机制适于提供实现 IPSec 规范所需的安全参数。

本发明包括由多个部件构成的 VPN 模块，所述多个部件一起发挥作用，从而实现 VPN 机制。图 3 中表示了更详细地图解说明本发明的 VPN 模块的方框图。VPN 模块 70 包括总线接口 72、缓存器/寄存器 74、安全关联处理器（SAP）75、VPN 安全处理器 88 和分组建立器 92。

安全关联处理器包括 SA 识别模块 76、包括 CPU 接口 79 的主 SA 处理模块 78、SA 管理模块 80、散列表 82、SA 最近最少使用（LRU）电路 84 和 SA 数据库 86，所有这些都通过总线与 VPN 安全处理器中的安全引擎和分组建立器通信。

VPN 安全处理器包括多个安全引擎 90，每个安全引擎 90 被修改和配置成执行特定的涉及安全的操作。例如，VPN 安全处理器包括进行加密、解密、鉴权、DES 算法、信头消息鉴权代码（HMAC）等的独立安全引擎。

VPN 模块通过总线接口，在总线 107 上与主设备/CPU 96、信头解码器 100、静态过滤器 102、动态过滤器 104 和内容搜索单元 106 通信。

注意在本文献中，假定 VPN 模块通常位于 WAN 和 LAN 之间，并且既对入站分组又对出站分组执行安全处理。入站分组指的是从 WAN 接收的去往 LAN 的分组，出站分组指的是从 LAN 接收的去往 WAN 的分组。从而，输入分组流既可包括入站分组，又可包括出站分组。

输入分组流 98 由主设备/CPU 接收，其内容被转发给信头解码器，信头解码器译解（或解析）分组的信头部分。信头解码器抽取由 VPN 模块使用的关心字段。信头解码器抽取的数据通过总线被转发给其它模块，包括静态过滤器，动态过滤器和 VPN 模块。该数据包括两个散列值，两个套接字（分别用于动态过滤器和 SA 处理）和分组的类型，例

如 ICMP、ARP、TCP、UDP 等。用于 SA 和 VPN 处理的套接字包括 SPI、RIP 和协议。用于动态过滤的套接字包括源和目的地 IP 地址，源和目的地端口和协议。注意取决于实现，信头解码器可构成为外部模块，或者与 VPN 模块结合。

操作中，VPN 模块打开新 SA，包括建立与 LRU 的连接和散列链接列表，确定输入分组对应于哪个 SA，在成功处理分组之后更新 SA 的状态，和从连接表中去除未使用的 SA。注意 SA 数据库被实现和安排成以致能够容易并快速地进行 SA 识别和指针管理。

主 SA 处理模块起安全关联处理器的主处理块的作用。它打开新的 SA，处理现有 SA 和跟踪 SA 的状态。它还包括适合于实现防重放机制的电路，所述防重放机制包括更新入站分组的防重放窗口，递增出站分组的序列号/更新入站分组的序列号，检测序列号溢出。

主 SA 处理模块还适合于更新与 SA 相关的状态，包括保持一组标记，序列编号，递减使用期变量，和在处理每个分组之后更新状态寄存器。对于每个接收的分组，主 SA 处理器循环工作。该循环包括从 SA 数据库检索 SA 参数，处理分组，即加密、解密等，和一旦完成处理，就更新 SA 数据库。注意只有当分组的安全处理不会导致错误并且该分组未被拒绝时，才更新 SA 数据库。

主 SA 处理模块还适合于计数按分组在每个 SA 上传送的字节的数目，并且当 SA 的使用期溢出时，产生通知。当检测到溢出时，管理器关闭（即删除）该 SA。

SA 识别模块接收套接字，并在 SA 数据库中搜索和该套接字匹配的 SA。套接字用于识别 SA，并且包括安全参数索引（SPI），远程 IP（RIP）和安全协议标识符。SPI 包括在参考特定 SA 的 IKE 过程之后，由 CPU 产生的 32 位随机数。在 AH 和 ESP 信头中传送 SPI，以使接收站能够选择按其处理接收分组的 SA。远程 IP 是 SA 的目的地端点的 IP 地址，该 SA 可以是最终用户或 VPN 网关或者具有 VPN 能力的任意设备。安全协议标识符指示 SA 是 AH 还是 ESP SA。

SA 管理模块保持 SA 数据库。利用最近最少使用（LRU）双重链

接表管理 SA 数据库，并利用散列表访问 SA 数据库。另外，管理器在入站和出站分组流上将新的 SA 插入数据库。管理器还使与匹配分组相关的 SA the MRU，就请求产生适用于 CPU 的多个未用 SA，在时间方面检查一个或多个 SA 的使用期，当删除 SA 时，更新 LRU 和散列链接表，并通知动态过滤器与该 SA 相关的会话将被删除。

SA 数据库保存套接字和其它 SA 相关数据，包括 SA 状态和当前状态，供 VPN 模块的各个处理模块使用。散列表被用于加速 SA 的识别。下面更详细地说明这些组件。

主 SA 处理

下面更详细地说明主 SA 处理。图 4A 和 4B 中表示了图解说明本发明的主 SA 处理器方法的流程图。构成 VPN 模块和安全关联处理器，以便分三阶段处理分组数据，其中在第一阶段中，从 SA 数据库读取分组，随后在第二阶段中处理数据，并在第三阶段中，将处理结果写回 SA 数据库。在处理阶段中，对分组执行一个或多个安全程序，并根据处理跟踪 SA 状态。

首先检查接收的分组是入站分组还是出站分组（步骤 110）。如果是出站分组，则检查动态过滤器是否找到 SA（步骤 154）。通常，动态过滤器搜索其数据库，寻找具有和接收分组的套接字相符的套接字的会话。对于每个会话，在动态过滤器中的会话数据库中保存有相应的 SA。当找到匹配的会话时，读出对应的 SA 并将其输入 SA 处理器。在美国专利申请序列号 No.09/851768（申请日 2001 年 5 月 9 日）“Dynamic Packet Filter Using Session Tracking”中更详细地说明了动态过滤器的操作，该专利申请同样被转让并作为参考整体包含于此。

如果关于该分组未找到 SA（步骤 154），则该分组被拒绝（步骤 168），因为没有 SA，就不能处理该分组，随后更新状态寄存器（步骤 146）。这种情况下，CPU 打开新的 SA，并将相对于该 SA 的指针保存在动态过滤器中的会话数据库中。随后重新拒绝该分组，但是这次动态过滤器识别出该会话，并将 SA 传递给 SA 处理器。

如果找到 SA，则从 SA 数据库检索一个或多个安全参数，分组最

大传送单元（MTU）和上标记（upper flag）被传送给 VPN 安全处理器中的一个或多个安全引擎（步骤 156）。MTU 代表允许无分段传送的分组的最大大小。

加密/解密（ED）和信头消息鉴权代码（HMAC）是对分组数据执行安全功能的安全引擎的例子。传送给 ED 安全引擎的数据包括 IPSec 模式，特定的加密算法（例如 DES、3DES 等）、CBC 或 ECB、填充流类型（即 0 或递增）、3DES 中每个 DES 的类别、加密初始矢量（IV）（只适用于 CBC 模式）、加密密钥和来自信头解码器的加密/解密指令。

传送给 HMAC 安全引擎的数据包括 IPSec 模式、鉴权算法（例如 SHA1、MD5 等）、鉴权信头大小（即 96 或 160 位）和鉴权密钥。

套接字随后被传送给分组建立器，以便装配成分组（步骤 158）。传送给 PB 的数据包括特定的 IPSec 模式、远程 IP（如果实现隧道模式，则是网关 IP）、SPI、递增 1 的序列号和路径 MTU。

随后从 SA 数据库读取序列号和使用期字段（步骤 160）。随后检查序列号和使用期（步骤 162）。检查序列号是否发生了溢出。序列号溢出可以是硬溢出或软溢出。如果发生硬溢出，则关闭 SA（步骤 164），拒绝该分组（步骤 168），并据此更新状态寄存器（步骤 146）。如果发生软溢出，则通过修改下标记中的软序列（SEQ）位，通知 CPU，处理继续步骤 120。如果没有发生序列号溢出，则控制转到步骤 120。

对于使用期，检查是否发生了使用期溢出。注意主 SA 处理只依据字节检查使用期。SA 管理执行依据字节的使用期检查。从使用期的当前值减去接收分组的大小，结果核实溢出。小于 0 的结果表示溢出。溢出可以是硬溢出或软溢出。如果发生了硬溢出，则关闭 SA（步骤 164），拒绝该分组（步骤 168），并据此更新状态寄存器（步骤 146）。如果发生了软溢出，则通过修改下标记中的软使用期（SLT）位通知 CPU，处理继续执行步骤 S120。如果没有发生使用期溢出，则控制转到步骤 120。

如果接收的分组是入站分组（步骤 110），对该分组执行安全关联识

别（步骤 112），如下更详细所述。通过散列 SPI、远程 IP/网关 IP 和协议识别 SA。识别结果被返回给主 SA 处理器。如果在识别过程中没有发现任何匹配 SA 入口，则拒绝该分组（和 SA 仍然不存在时的情况一样）。如果识别找到匹配 SA，则译解该分组。动态过滤器将该分组识别成第一分组，打开新会话并连接该会话和 SA。

如果 IPSec 模式失败（步骤 114），则拒绝该分组（步骤 152），并更新状态寄存器（步骤 146）。如果 IPSec 模式未失败，则将上标记传递给安全引擎，例如 ED、HMAC，和分组建立器（步骤 116）。分组建立器装配明文形式的分组，以便通过 LAN 转发。

传送给 ED 安全引擎的数据包括 IPSec 模式、加密算法（例如 DES、3DES 等）、CBC 或 ECB、填充流类型、3DES 中每个 DES 的类别、加密密钥和来自信头解码器的加密/解密指令（即是要进行加密还是解密）。

传送给 HMAC 安全引擎的数据包括 IPSec 模式、鉴权算法（例如 SHA1、MD5 等）、鉴权信头和鉴权密钥。

如果任意安全引擎产生错误（例如，如果填充失败，由 ED 产生错误，或者如果鉴权失败，由 HMAC 产生错误，等等），则拒绝该分组（步骤 114）。

随后执行防重放窗口方法（步骤 118），下面更详细地说明。如果接收的序列号超过防重放窗口（ARW），则更新和改变接收的序列号以便调整接收的序列号。

如果 SA 是 AH（步骤 120），则 AH 密钥被传送给鉴权安全引擎（步骤 148），控制转到步骤 140。SA 可以是（1）AH 单一 SA 或者（2）为 ESP 和 AH SA 束（bundle）一部分的 AH SA。在后一情况下，在首次通过该方法的过程中，指针指向所述束的 ESP 部分，在第二次通过该方法的过程中，指针指向所述束的 AH 部分。

如果 SA 不同于 AH SA，则检查分组是否是入站分组（步骤 122）以及是否是 CBC（步骤 124）。如果是，则 IV 被传递给 ED 安全引擎（步骤 126）。如果分组不是入站分组或者不是 CBC，则将 ESP 密钥传

送给 ED 安全引擎（步骤 128）。

如果接收的分组是 ESP/AU 分组（带有鉴权的 ESP）（步骤 130），则将 AH 密钥传送给鉴权安全引擎（步骤 148）。如果否，并且分组不是 ESP（步骤 132），而是入站分组（步骤 136），则关于 ESP 和 AH 类分组指向 AH SA 部分（步骤 137），控制随后转到步骤 118。如果分组是出站分组，则控制转到步骤 162。如果接收分组不是 ESP/AU 分组（步骤 130），而是 ESP 分组（步骤 132），出站分组（步骤 134）和 CBC 分组（步骤 138），则将加密 IV 保存在 SA 数据库中（步骤 139）。

如果分组是入站分组（步骤 134），则将 ARW 保存在 SA 数据库中（步骤 140），将更新的序列号保存在 SA 数据库中（步骤 142），更新下标记（步骤 144），更新状态寄存器（步骤 146）。在状态寄存器更新之后，可出现中断，发信号向 CPU 或者其它主设备通知主 SA 处理的完成。

安全关联识别

现在详细说明 SA 识别过程。图 5 中图解说明确定与输入分组相关的 SA 的散列技术。每个 SA 对应于一个独特的套接字。通过比较接收分组的套接字和与保存在 SA 数据库中的先前打开的 SA 相关的套接字，识别 SA。为了加速 SA 的识别，使用了散列表，所述散列表保存 SA 数据库中的 SA 记录的散列指针，并允许快速查找对应于接收套接字的 SA。

新的 SA 被保存在 SA 数据库中，VPN 模块、SA 处理器或者 CPU 计算关于套接字的散列。散列指针保存在散列表 170（图 5）中散列结果指向的位置。如果一个以上的 SA 被保存在该位置，则将该 SA 添加到链接列表中。注意散列表中的各入口最初被初始化成 NULL。

当收到分组时，套接字 172 被输入散列计算器 176，散列计算器 176 产生并输出散列结果 178。散列结果被用作散列表 170 的索引，散列表 170 包括分别包含散列指针的多个入口 180。散列指针指向 SA 数据库中的 SA 182 的链接列表。记录在数据库中的每个 SA 记录包括前一指针 186 和后一指针 184，从而实现双重链接列表。如果命中套接字，

则必须检查链接列表中的每个 SA，直到找到与接收分组的套接字的匹配为止。

最好，选择散列函数以便横越散列表产生散列结果的尽可能均匀的展形（spread）。散列函数可以是任意适当的函数，例如 XOR 函数或 CRC。和简单的 XOR 散列函数相比，在一个实施例中，通过使用随机矢量 174，根据下述等式计算散列结果，可提高性能。

$$\begin{bmatrix} SOCK_1 \\ SOCK_2 \\ SOCK_3 \\ \vdots \\ SOCK_N \end{bmatrix}_{1\times N} \otimes \begin{bmatrix} RV_1 & RV_2 & RV_3 & \cdots & RV_N \\ RV_2 & RV_3 & RV_4 & \cdots & RV_{N-1} \\ RV_3 & RV_4 & RV_5 & \cdots & RV_{N-2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ RV_R & RV_{R+1} & RV_{R+2} & \cdots & RV_{R+N-1} \end{bmatrix}_{N\times R} = \begin{bmatrix} OUT_1 \\ OUT_2 \\ OUT_3 \\ \vdots \\ OUT_R \end{bmatrix}_{1\times R} \qquad (1)$$

其中运算符⊗被如下定义

$$OUT_i = (RV_{i1}\ AND\ SOCK_1) \oplus (RV_{i2}\ AND\ SOCK_2) \oplus \ldots \oplus (RV_{iN}\ AND\ SOCK_N) \qquad (2)$$

OUT_i 代表输出矩阵的第 i 个字节；

$SOCK_k$ 代表输入矩阵的第 k 个字节；

RV_{ik} 代表随机矢量矩阵的 i，j 字节；

⊕表示 XOR 函数；

从而利用随机矢量和输入套接字数据产生输入套接字数据[2]。

图 6 中表示了图解说明本发明的 SA 识别过程的散列方法的流程图。第一步是从信头解码器获得接收分组的散列指针（通常由信头解码器提供），套接字和序列号（步骤 190）。散列值由散列计算器 176 产生。另一方面，信头解码器或其它实体可计算散列指针。查寻散列表，从散列表读取散列指针（步骤 191）。如果散列指针等于 NULL（步骤 192），则拒绝该分组。

如果散列指针不是 NULL，则使用散列指针读取和散列指针相关的链接列表中的第一 SA 和对应于该 SA 的套接字（步骤 194）。将该 SA 的套接字和接收分组中的套接字进行比较（步骤 196）。如果套接字匹配（步骤 197），则发现 SA 匹配（步骤 202），并向主 SA 过程报告。如果套接字不匹配（步骤 197），从链接列表读取下一散列指针（步骤

198），该方法从步骤 192 开始重复，直到最后的散列指针指向 NULL 或者发现 SA 匹配为止。

注意即使在链接列表中只保存一个 SA 的情况下，也总是对套接字进行完全比较。另外注意本发明的范围不受哪个实体更新和保持链接列表（即或者 SA 处理器或者 CPU 更新和保持链接列表）限制。散列表的深度可以是任意所需值。但是通常根据同时要保持的 SA 的数目设置深度。散列入口的数目最好是 SA 数目的一倍或两倍，因为完全套接字比较费时，副本过多是不可取的。

安全关联数据库

现在更详细地说明 SA 数据库 86（图 3）。如前所述，SA 数据库保存多个安全关联性的涉及安全性的数据。SA 数据库的大小可根据实现和系统要求而变化。下表 1 中列举了构成数据库的每个记录的字段。

表 1：SA 数据库记录字段

字段号	字段说明	长度（位）
1	上标记	16
2	路径 MTU	16
3	远程 IP 地址/网关地址	32
4	安全参数索引（SPI）	32
5	下一 SA 束指针	16
6	前一 SA 束指针	16
7	下一散列指针	16
8	前一散列指针	16
9	下一 LRU 指针	16
10	前一 LRU 指针	16
11	下标记	16
12	最后匹配会话指针	16
13	序列号	32
14	防重放窗口（127：0）	128
15	加密初始值（63：0）	64

16	加密密钥（191：0）	192
17	鉴权密钥/散列密钥输入（159：0）	160
18	散列密钥输出（159：0）	160
19	软/硬使用期	32

字段 1 和 11 分别保存上标记和下标记，上下标记包含后面更详细说明的多个状态位。路径 MTU 保存在字段 2 中，代表关于包括所有信头的分组允许的最大大小。字段 3 中的远程 IP/网关 IP 地址和字段 4 中的 SPI 构成 SA 套接字及安全协议标识符。下一 SA 指针和前一 SA 指针（字段 5 和 6）用于构成均与相同的安全策略数据库（SPD）相关的 SA 的双重链接列表。字段 7 和 8 保存前面说明的散列链接列表内的下一和前一散列指针。在分组的 SA 识别过程中，使用散列链接列表。字段 9 和 10 保存 LRU 链接列表中的下一和前一 LRU 指针，用于按照陈旧性对 SA 排序。下面更详细地说明 LRU 链接列表操作。

相对于具有该 SA 的最后打开的会话的指针保存在字段 12 中。该 SA 的当前序列号保存在字段 13 中。字段 14 保存在 ARW 机制中用于拒绝重放分组的 128 位防重放窗口。字段 15 保存 64 位加密初始值（IV），字段 16 保存 192 位的加密密钥，字段 17 保存 160 位的鉴权密钥/散列密钥输入（HKI）。字段 18 保存 160 位散列密钥输出（HKO）值。字段 19 保存 SA 的软/硬使用期，它被用于确定何时关闭特定 SA。

HKI 和 HKO 是在鉴权过程中产生的中间结果。它们都保存在该表中，以便当收到后续分组时节省处理时间。HKI 和 HKO 都由鉴权密钥和预定填充值的散列计算得到。HKI 值保存在 SA 数据库中鉴权密钥的位置，因为一旦计算得到 HKI 值，就不再需要初始密钥。另一方面，鉴权密钥、HKI 和 HKO 值可单独保存在数据库中。SA 数据库从 HMAC 安全引擎接接收收首次使用 SA 时的 HKI 和 HKO 值。

如上所述，SA 数据库包括上标记和下标记，用于相对于 CPU 传送状态信息。上标记状态包括如下表 2 中所示的多个位。

表 2：上标记位

标记	定义	大小	值	描述

			0000	无 VPN
			0001	传送 AH
			0010	隧道 AH
			0011	传送 ESP
			0100	隧道 ESP
			0101	传送 ESP/AU
IPM	IPSed 模式	4 位	0110	隧道 ESP/AU
			0111	传送 ESP 传送 AH
			1000	隧道 ESP 传送 AH
			1001	传送 ESP 空（仅 AU）
			1010	隧道 ESP 空（仅 AU）
			1011	鉴权产生/检查
			1100	加密/解密
			1101	加密/解密+鉴权
DET	DES 类型	1 位	0	DES
			1	3DES
DEM	DES 模式	1 位	0	ECB
			1	CBC
PAM	PAD 模式	1 位	0	填充 0 的 ESP
			1	填充递增值的 ESP
			000	EEE
			001	EED
			010	EDE
3DE	3DES 类别	3 位	011	EDD
			100	DEE
			101	DED
			110	DDE
			111	DDD
AHT	AH 类型	1 位	0	MD5

		1	SHA1	
KEY	密钥的首次使用	1 位	0	这是同一密钥
			1	该密钥的首次使用-产生 HKI 和 HKO
AHS	AH 大小	1 位	0	AH 大小为 96
			1	AH 大小为 160
AH	AH 或 ESP	1 位	0	该 SA 是 ESP
			1	该 SA 是 AH
DIR	目录	1 位	0	出站分组的 SA
			1	入站分组的 SA
EMP	空	1 位	0	该 SA 被使用（SA 有效）
			1	该 SA 为空（SA 无效）

所有上标记由 CPU 设置并由 SA 处理器读取。EMP 标记也可由 SA 处理器设置。当 SA 为空，即无效时，SA 处理器将该位设置为 1。当 SA 有效时，CPU 将该位设置成 0。IPM 位指示具体的 IPSec 模式，即 ESP、ESP/AU、仅 AU、传送、隧道等。可选的是，最后三种 IPSec 模式可用于实现辅助安全标准，例如安全套接字层（SSL），从而对某一文件加密等。在这些模式中，单独使用 VPN 引擎，而不具有分组建立功能。从而，VPN 引擎起软件加速器的作用，软件加速器实现 DES/3DES 加密/解密引擎模式（IPM=1100），SHA-1/MD-5 鉴权引擎模式（IPM=1011）或者加密和鉴权引擎模式（IPM=1101）。

DET 位指示 DES 的类型或者为 DES 或者为 3DES。DEM 位指示 DES 模式或者为 ECB 或者为 CBC。PAM 位指示填充 0 的 ESP 或填充递增值的 ESP。3DE 位指示特定的 3DES 类别。AHT 位指示 AH 鉴权的类型或者为 MD5 或者为 SHA1。AHS 位指示 AH 的大小或者为 96 位或者为 160 位，AH 位指示 SA 或者为 ESP 或者为 AH，DIR 位指示 SA 是用于出站分组还是用于入站分组。

SA 数据库还包括下标记状态寄存器，用于相对于 CPU 传送状态信息。下标记状态信息包括下表 3 中所示的多个位。

表 3：下标记位

标记	定义	大小	值	描述
			00	禁止防重放
			01	防重放窗口=32
ARW	ARW 大小	2 位	10	防重放窗口=64
			11	防重放窗口=128
HSH	HSAH 中的第一个	1 位	0	这不是本 HASH 入口中的第一 SA
			1	这是本 HASH 入口中的第一 SA
HLD	保持	1 位	0	
			1	不删除本会话
MAN	手动键入	1 位	0	无关于该 SA 的手动键入
			1	关于该 SA 的手动键入
SAL	SA 使用期模式	1 位	0	以秒测量 SA 使用期
			1	以 64 字节为单位测量 SA 使用期
SOH	软或硬使用期	1 位	0	该 SA 中的使用期为软使用期
			1	该 SA 中的使用期为硬使用期
SLT	软使用期	1 位	0	
			1	SA 使用期达到软溢出
SEQ	软序列	1 位	0	
			1	SA 序列达到软溢出

ARW 位位指示防重放窗口的大小（如果存在防重放窗口）。HLD 位指示不删除特定的 SA。该标记使 SA 入口免被 SA 处理器硬件删除。MAN 位指示是否存在关于 SA 的手动键入。如果配置了手动键入，则当达到 0xFFFFFFFF 时，序列翻转，因为手动键入不允许被 SA 处理器删除。SAL 位指示是用时间还是用数据，即用秒还是 64 字节单位测量 SA 使用期。SOH 位指示使用期是软使用期还是硬使用期。SLT 位指示在软使用期内是否发生了软溢出。SEQ 位指示在 SA 序列中发生了软溢出。注意 ARW、HLD、MAN、SAL 和 SOH 标记由 CPU 设置并由 SA 处理器读取。SLT 和 SEQ 标记由 SA 处理器设置并由 CPU 读取。除非设置了 HLD 或 MAN 标记，否则当序列溢出时（0xFFFFFFFF），删除

该 SA。

防重放窗口机制

根据本发明，主 SA 处理适合于实现防重放机制，从而拒绝重放分组。防重放窗口机制跟踪分组中的序列号，并拒绝其序列号小于许可的最小序列号的分组。图 7 中表示了图解说明检查过程中，由主 SA 处理器执行的防重放窗口机制。

x 轴代表 SA 序列号，最小序列号为 0x00000000，最大序列号为 0xFFFFFFFF。产生一个窗口，称为代表许可序列号的防重放窗口（ARW）（段 212）。其序列号低于该窗口的接收分组被拒绝（段 210）。另外，每个序列号只允许一个分组。从而，ARW 包括落入其内的每个序列号的一位。当收到分组时，设置相应的序列号位。其序列号位已被设置的接收分组被拒绝。

图 8 中表示了图解说明本发明的防重放窗口方法的流程图。首先从接收的分组读取序列号（步骤 280）。从 SA 数据库检索防重放窗口的当前位置（步骤 282）。如果序列号在窗口内（步骤 284），则检查是否已收到具有该序列号的分组（步骤 286）。如果是，则拒绝该分组（步骤 290）。如果否，则 ARW 机制允许该分组。

如果接收分组中的序列号在窗口之外（步骤 284），则检查它是低于还是高于 ARW（步骤 288）。如果低于窗口，则拒绝该分组。如果高于窗口，则许可该分组，并将 ARW 向上调节到新的序列号，即向右移动。

1 SA 管理模块

根据本发明，SA 管理模块保持双重链接列表，双重链接列表按照最近最少使用顺序保存 SA。SA 管理模块实现与 SA 数据库的保持和控制相关的几种功能。SA 管理模块根据来自 CPU 的命令，检查 SA 数据库中 SA 的 SA 入口（即使用期）的有效性。当收到检查使用期的命令时，检查 SA 数据库中每个记录的使用期字段。向 CPU 报告软使用期溢出，在硬使用期溢出的情况下，删除相应的 SA。对 SA 进行使用期检查，以便清洗已变得陈旧和由于某一原因未正常关闭的 SA。例如，

如果某一 SA 对应于违反了安全规则的分组，则该 SA 可能变得陈旧。

当 SA 被删除时，SA 管理模块通过保持 LRU 和散列指针关系，保持和更新 SA 束，并通知动态过滤器模块应关闭哪个对应会话，以便确保使用该 SA 的会话将被关闭。

SA 管理模块还应请求向 CPU 提供未使用的（即空的并且可用的）SA。操作中，CPU 请求一个或多个新的 SA，SA 管理模块搜索并将一个或多个索引返回给 CPU。通过断开最近最少使用的 SA 束（即传送方向和接收方向的 SA）和所有相关连接会话，从 LRU 列表获得 SA。类似地，SA 管理模块还断开不再使用的 SA。SA 管理模块还打开新的 SA，其中散列表和 SA 数据库中的 LRU 指针和散列指针被更新。

图 9 中图解说明了用于跟踪 SA 使用的最近最少使用（LRU）链接列表结构。当打开 SA 时，通过 CPU 将安全性关联入口输入 SA 数据库中。CPU 向 SA 管理模块请求 SA 索引。下一 SA 束指针和前一 SA 束指针被用于连接彼此相关的 SA。为了将 SA 插入数据库中，使用前导 SA 的索引将新 SA 插入 LRU 中。另外，CPU 计算新 SA 的套接字的散列值，并将其写入寄存器中标记其散列值。

注意在这里给出的例证实施例中，主机或外部 CPU 在 IKE 过程中配置 SA 的涉及安全性的参数，例如密钥分配、HMAC 的长度、填充等。SA 管理模块依据 CPU 或主机的命令实现 SA 的实际插入。

每次认出（即存取）某一 SA，并且对应分组不被 SA 处理器拒绝时，该 SA 被置于 LRU 链接列表的尾部，代表最近最多使用的 SA。双重链接列表 220 包括多个 SA 226，每个 SA 226 具有下一指针 222 和前一指针 224。tail_LRU_index_reg 寄存器 228 的内容指向位于 LRU 链接列表尾部的 SA。该寄存器指向的 SA 代表最近最多使用的 SA。位于 LRU 链接列表头部的 SA 由 head_LRU_index_reg 寄存器 227 的内容指向。

在 SA 数据库充满的情况下，LRU 链接列表被用于确定当加入新 SA 时，要删除哪个 SA。这种情况下，关闭最近最少使用的 SA，该空间用于保存新 SA。

图 10 中表示了图解说明当认出与输入分组相关的 SA 时，更新 LRU 链接列表的方法的流程图。每次存取匹配 SA 并且许可相应分组时，执行该更新方法。参见图 9，尤其是标记为前一 SA、匹配 SA 和下一 SA 的 SA，匹配 SA 从其在列表中的位置移动到尾部，从而变成最近最多使用的 SA。通过（1）将前一 SA 的下一 LRU 指针设置成匹配 SA 的下一 LRU 指针，和（2）将下一 SA 的前一 LRU 指针设置成匹配 SA 的前一 LRU 指针，删除匹配 SA（步骤 230）。

随后通过（1）将匹配 SA 的下一 LRU 指针设置成 NULL，（2）将匹配 SA 的前一 LRU 指针设置成 tail_LRU_index_reg 寄存器的内容，和（3）将 tail_LRU_index_reg 寄存器设置成匹配 SA 的索引，使匹配 SA 成为最近最多使用的 SA（步骤 232）。

图 11 中表示了图解说明本发明的 SA 管理模块的处理的流程图。SA 管理处理开始于等待步骤 241，并根据事件转移到下一步骤。就来自 CPU 的'获得未使用 SA'指令来说，取出最近最少使用 SA 的状态（步骤 242）。如果该 SA 在使用（步骤 246），则关闭该 SA 和所有相关的连接会话（步骤 244）。如果该 SA 未使用，则将 head_LRU_index_reg 设置成最近最少使用 SA 的下一 LRU 指针，并将下一（前一 LRU 指针）设置成 NULL（步骤 248）。随后将删除的 SA 的下一和前一 LRU 指针设置成 NULL，从而从 LRU 链接列表中删除最近最少使用的 SA（步骤 250）。unused_sa 变量被设置成刚刚断开的 SA，通知 CPU 的标记因此被设置成'1'（步骤 252）。该该过程随后返回等待状态（步骤 254）。

来自 CPU 的'检查使用期'命令使 SA 管理器开支检查所有现用 SA 的使用期。递增 index_counter（步骤 260），取出 SA 的状态（即 index_counter）（步骤 262）。如果该 SA 目前正在使用（步骤 264），从 SA 数据库取出 SA 使用期（步骤 372）。如果使用期已溢出（用时间或字节数测量）（即 SA 已到期）（步骤 272），则关闭该 SA 和所有相关的连接会话（步骤 274）。

如果该 SA 未被使用或者如果该 SA 还未到期，则检查

index_counter 是否小于最后索引（步骤 266）。如果是，则递增 index_counter（步骤 268），该方法继续执行步骤 262。如果否，则通过主 SA 处理模块通知 CPU 使用期检查结束（步骤 258），该方法返回等待状态（步骤 254）。

图 12 中表示了图解说明本发明的 SA 管理模块执行的打开 SA 的处理的流程图。当 SA 被打开时，该 SA 被置于 LRU 链接列表的尾部（步骤 330）。该 SA 还被置于散列链接列表中的适当位置（步骤 332）。该方法随后返回等待状态（步骤 334）。

图 13 中表示了图解说明本发明的 SA 管理模块执行的关闭 SA 的处理的流程图。如果 SA 可关闭或者发生了序列号溢出（步骤 340），则在不修改 LRU 指针的情况下清除该 SA（步骤 342）。随后关闭与该 SA 相关的所有会话（步骤 344）。该方法随后返回等待状态（步骤 359）。

如果该 SA 不能被关闭以及没有发生序列号溢出（步骤 340），并且关闭请求起源于使用期检查（步骤 346），则不关闭该 SA（步骤 358），该方法返回等待状态（步骤 359）。如果关闭请求不是起源于使用期检查，则取出最近最少使用列表指向的下一 SA 状态（步骤 348）。如果该 SA 不能被关闭（步骤 350），该方法继续执行步骤 348，重复该过程，直到找到能够被关闭的 SA 为止。如果找到能够被关闭的 SA，则将该 SA 配置成 LRU（步骤 352）。随后除了 LRU 指针之外，清除该 SA（步骤 354），清除与该 SA 相关的所有会话，该方法返回等待状态（步骤 359）。

如上所述，SA 管理模块还将指向关于该 SA 使用的最后打开会话的指针保持在 SA 数据库中。图 14 中表示了与每个 SA 记录相关的最后打开会话指针。每个 SA 入口可包括指向不同的最后打开会话的指针。每打开一个会话时，更新该指针。注意最后打开会话总是位于相关会话列表的头部。

动态过滤器中的会话数据库适合于保存数个指针，包括 LRU 指针、散列指针、族指针和 SA 指针。SA 指针指向该会话使用的 SA。图 15 中表示了与每个会话记录相关的前一会话指针、下一会话指针和匹配

SA 指针。会话数据库中的会话记录包括将用于该会话的匹配 SA 的指针和使用相同 SA 的会话的前一和下一族指针。当 SA 被关闭时，并且要关闭与该 SA 相关的所有会话时，使用会话入口中的下一和前一指针。SA 入口中的最后打开指针被用于指向使用该 SA 的一个或多个会话的列表。当该 SA 被关闭时，还必须关闭相关会话。注意当 SA 被关闭时，传送和接收 SA 被删除，所有连接会话也被删除，除非它们是某一族的一部分。这种情况下，它们被标记成要删除的候选者（即设置会话数据库标记中的 PSV 位）。

图 16 中表示了单一会话和单一 SA 记录之间的例证关系。本例中，只有一个会话入口使用该 SA。前一和下一指针从而指向 NULL。SA 入口包含该 SA 的最后打开会话的指针。

图 17 中表示了多个会话和单一 SA 记录之间的例证关系。本例中，三个会话使用相同的 SA。从而，所有三个会话的匹配 SA 指针指向相同 SA。每个会话的下一和前一指针被配置成形成包括这三个会话的双重链接列表。链接列表逻辑连接使用相同 SA 的会话。该 SA 入口包括该 SA 的最后打开会话的指针，该指针指向链接列表头部。

计算机实施例

在另一实施例中，计算机执行适于实现本发明的 VPN 机制或其任意部分（例如安全关联处理器）的软件。图 18 中表示了图解说明适于实现本发明的 VPN 机制的例证计算机处理系统-平台的方框图。该系统可包含在诸如 PDA、蜂窝电话机、电缆调制解调器、宽带调制解调器、膝上型计算机、PC、网络传输或交换设备、网络设备之类通信设备，或者任意其它有线或无线通信设备中。可利用硬件和/或软件的任意组合构成该设备。

计算机系统 300 包括处理器 304，处理器 304 可被实现成微控制器、微处理器、微计算机、ASIC 核心、FPGA 核心、中央处理器（CPU）或者数字信号处理器（DSP）。系统还包括均与处理器通信的静态只读存储器（ROM）302 和动态主存储器（例如 RAM）308。处理器还通过总线 326 与也包含在计算机系统中的许多外围设备通信。

该设备通过 WAN 接口 316 与诸如因特网之类 WAN 318 连接。另一方面，网络 318 可包括基于光学以太网的 MAN 或者其它类型的 MAN，取决于位置。接口包括相对于一个或多个 WAN 通信通道的有线和/或无线接口。通信 I/O 处理 314 在 WAN 接口和处理器之间传送数据。计算机系统还通过适合于处理正被使用的特定网络协议，例如铜线或光纤以太网、令牌环等各种协议之一的网络接口卡（NIC）310 连接 LAN 312。操作中，计算机系统如前所述动态过滤从 WAN 到 LAN 的入站分组和从 LAN 到 WAN 的出站分组。

可选的用户接口 320 响应用户输入，并提供反馈和其它状态信息。主机接口 322 连接主计算设备 324 和系统。主机适于配置、控制和维持系统的操作。系统还可包括保存应用程序和数据的磁存储设备 306。系统包括计算机可读存储介质，计算机可读存储介质可包括任意适当的存储装置，包括（但不限于）磁性存储装置、光学存储装置、CD-ROM 驱动器、ZIP 驱动器、DVD 驱动器、DAT 磁带、半导体易失性或非易失性存储器、生物存储装置、或者其它任意存储装置。

实现本发明的 VPN 机制的功能或其任意部分，例如安全性关联处理器的软件适合于驻留在计算机可读介质上，例如磁盘驱动器内的磁盘或者任意其它易失性或非易失性存储器。另一方面，计算机可读介质可包括软盘、快速存储卡、EPROM、EEROM、EEPROM 存储器，磁泡存储器、ROM 存储器等。适于实现本发明的 VPN 机制或其任意部分，例如安全性关联处理器的软件还可整体或部分地驻留于计算机系统的处理器内的静态或动态主存储器或者固件内（即在微控制器、微处理器、微计算机、DSP 等的内部存储器内）。

在备选实施例中，本发明的方法适于用集成电路、现场可编程门阵列（FPGA）、芯片集或专用集成电路（ASIC）、DSP 电路、无线实现和其它通信系统产品实现本发明。

附加的权利要求意图覆盖落入本发明精神和范围内的本发明的所有特征和优点。由于本领域的技术人员易于想到各种修改和变化，因此本发明并不局限于这里描述的有限实施例。因此，可采用落入本发明精神

和范围内的所有适当变化、修改和等同物。

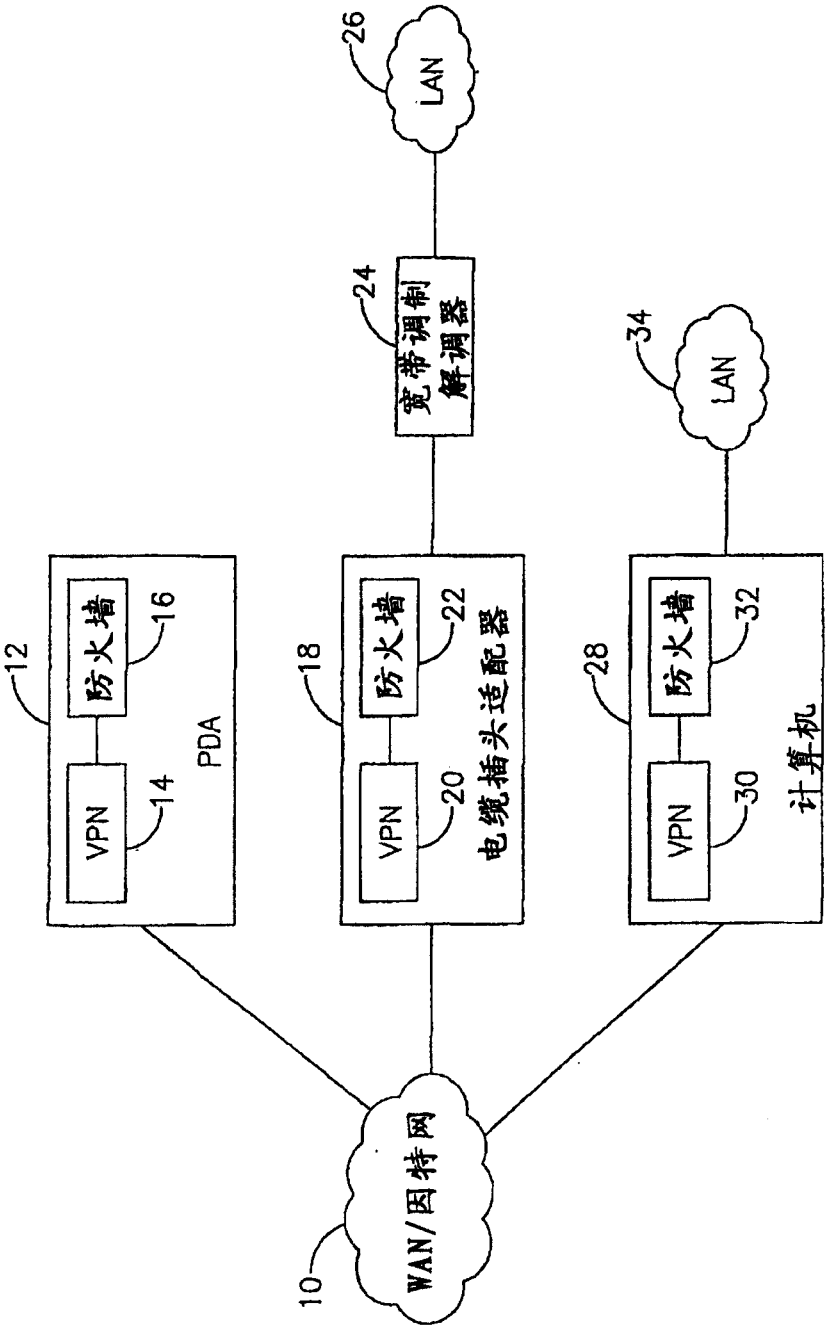

26
LAN
24
宽带调制解调器
34
LAN
12
防火墙
16
VPN
14
PDA
18
防火墙
22
VPN
20
电缆插头适配器
28
防火墙
32
VPN
30
计算机
10
WAN/因特网

图 1

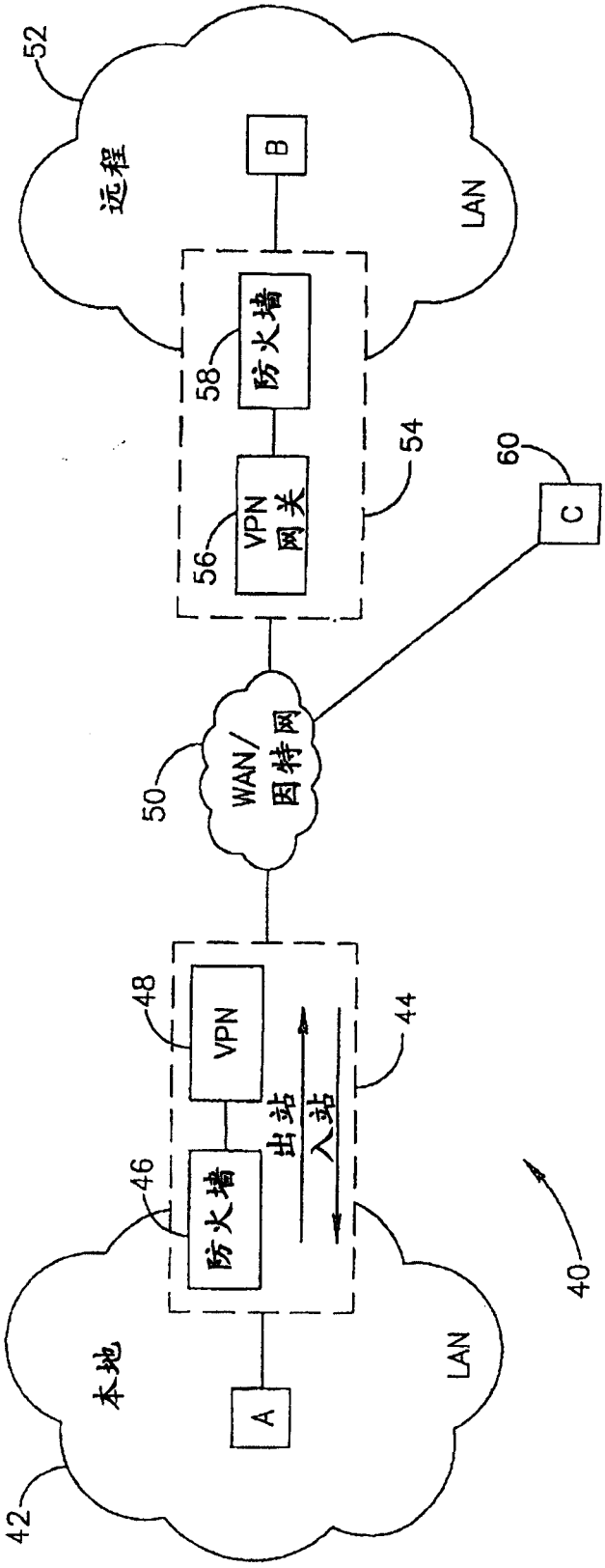

52
远程
LAN
B
58
防火墙
54
60
56
VPN
网关
C
50
WAN/
因特网
48
VPN
44
出站
入站
46
防火墙
本地
LAN
A
42
40

图2

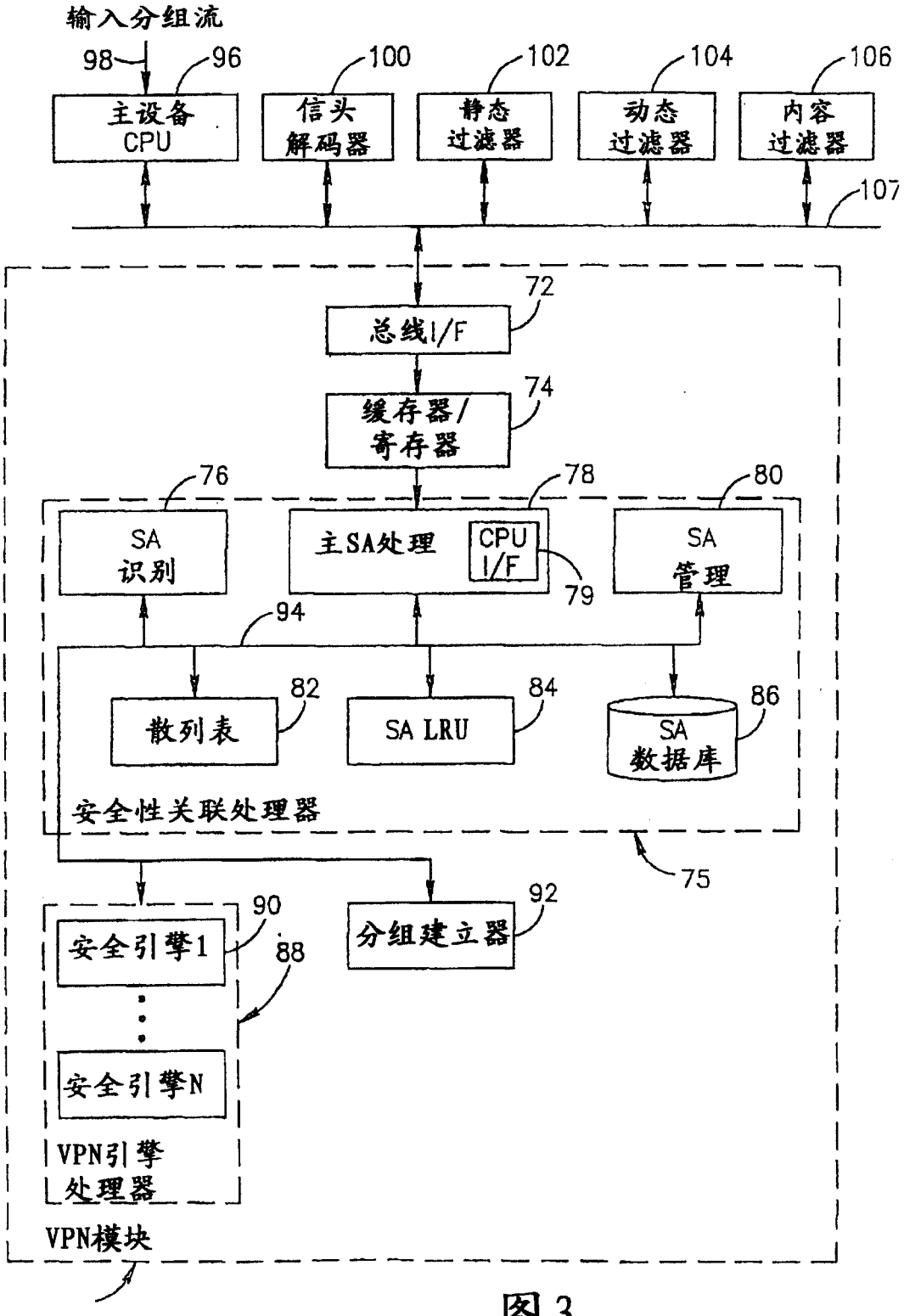

输入分组流
98
96
100
102
104
106
主设备 CPU
信头 解码器
静态 过滤器
动态 过滤器
内容 过滤器
107
72
总线I/F
74
缓存器/ 寄存器
76
78
80
SA 识别
主SA处理
CPU I/F
79
SA 管理
94
82
84
86
散列表
SA LRU
SA 数据库
安全性关联处理器
75
90
92
安全引擎1
88
分组建立器
安全引擎N
VPN引擎 处理器
VPN模块

图3

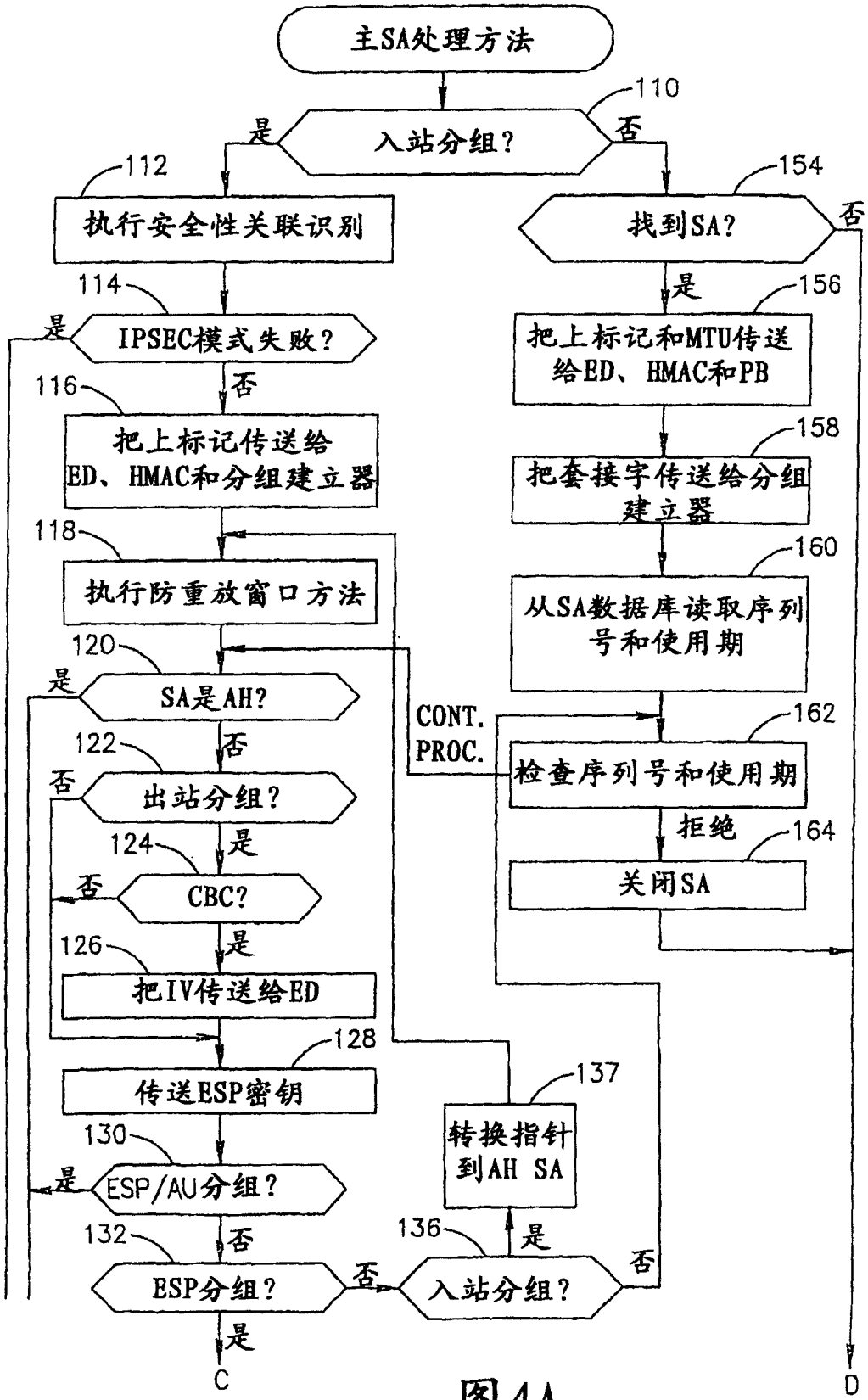

主SA处理方法
110
入站分组?
是
否
112
执行安全性关联识别
154
找到SA?
否
114
是
IPSEC模式失败?
是
156
把上标记和MTU传送给ED、HMAC和PB
否
116
把上标记传送给ED、HMAC和分组建立器
158
把套接字传送给分组建立器
118
执行防重放窗口方法
160
从SA数据库读取序列号和使用期
120
是
SA是AH?
CONT.
PROC.
162
检查序列号和使用期
否
122
否
出站分组?
拒绝
164
关闭SA
是
124
否
CBC?
是
126
把IV传送给ED
128
传送ESP密钥
137
转换指针到AH SA
130
是
ESP/AU分组?
136
是
否
否
132
否
ESP分组?
否
入站分组?
是
C
D

图4A

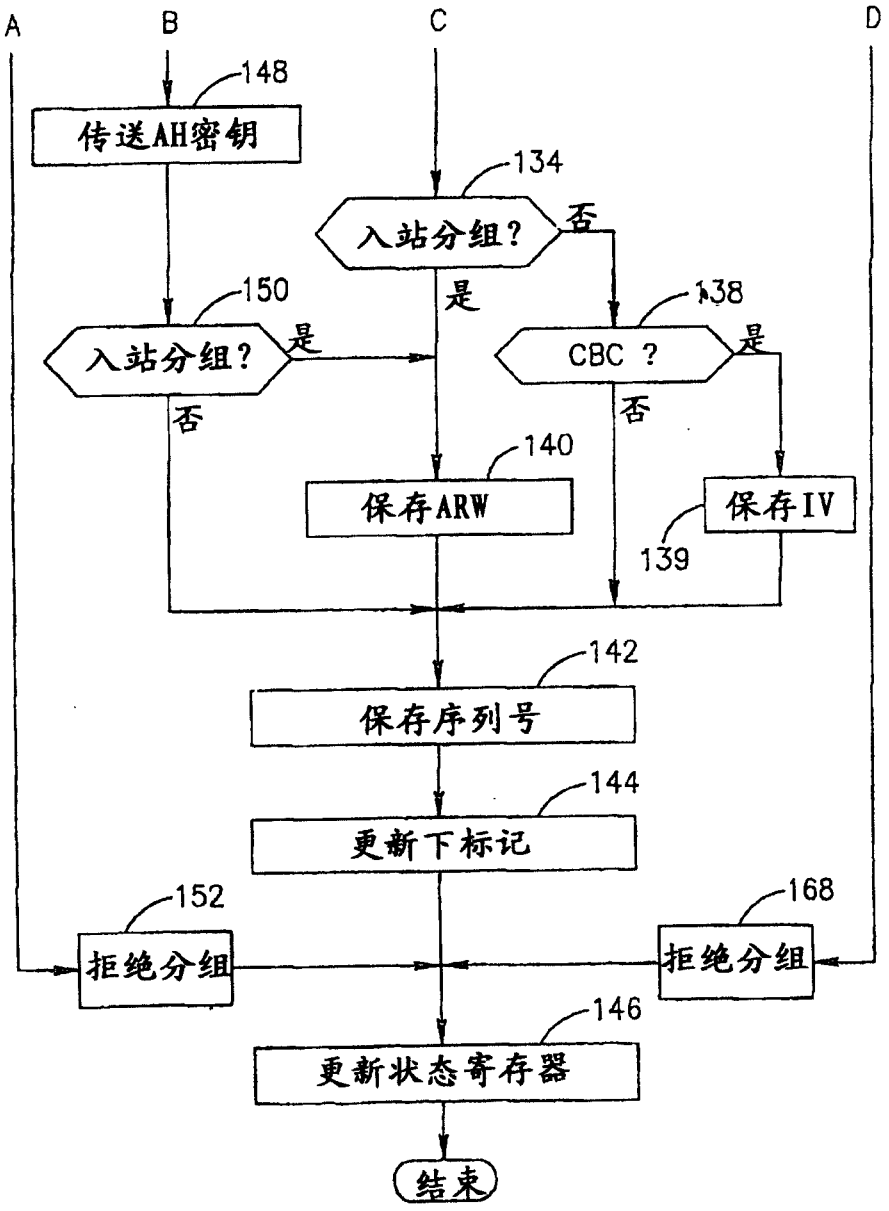

A
B
C
D
148
传送AH密钥
134
入站分组?
否
是
150
入站分组?
是
否
138
CBC ?
是
否
140
保存ARW
保存IV
139
142
保存序列号
144
更新下标记
152
拒绝分组
168
拒绝分组
146
更新状态寄存器
结束

图4B

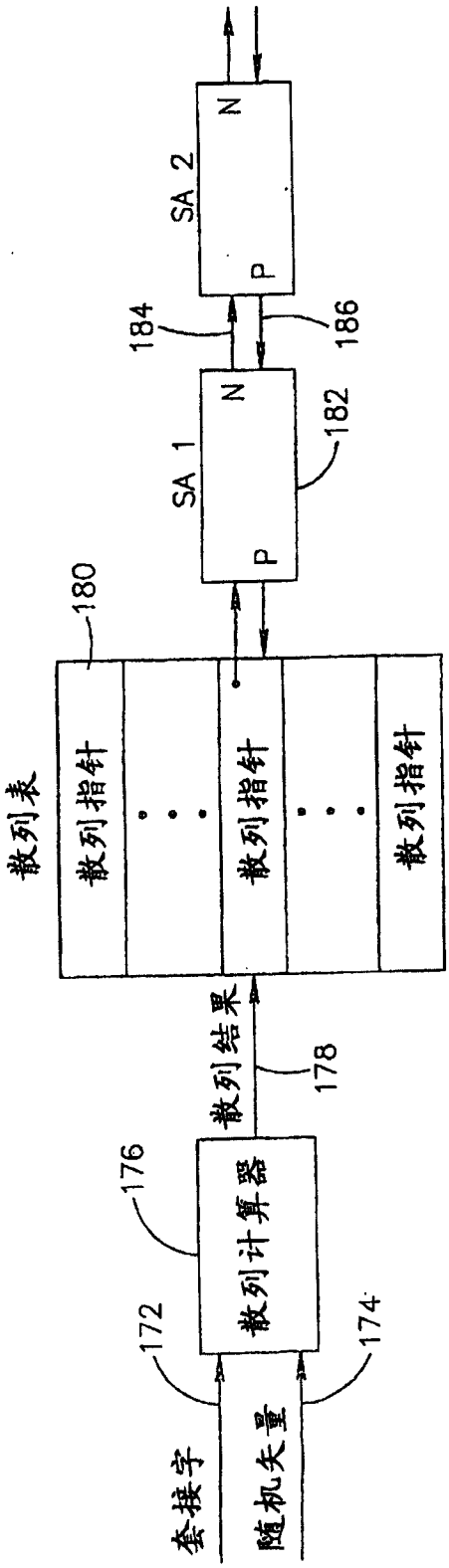

散列表
散列指针
散列指针
散列指针
180
SA 1
N
P
182
SA 2
N
P
184
186
散列结果
178
散列计算器
176
172
174
套接字
随机矢量
170

图 5

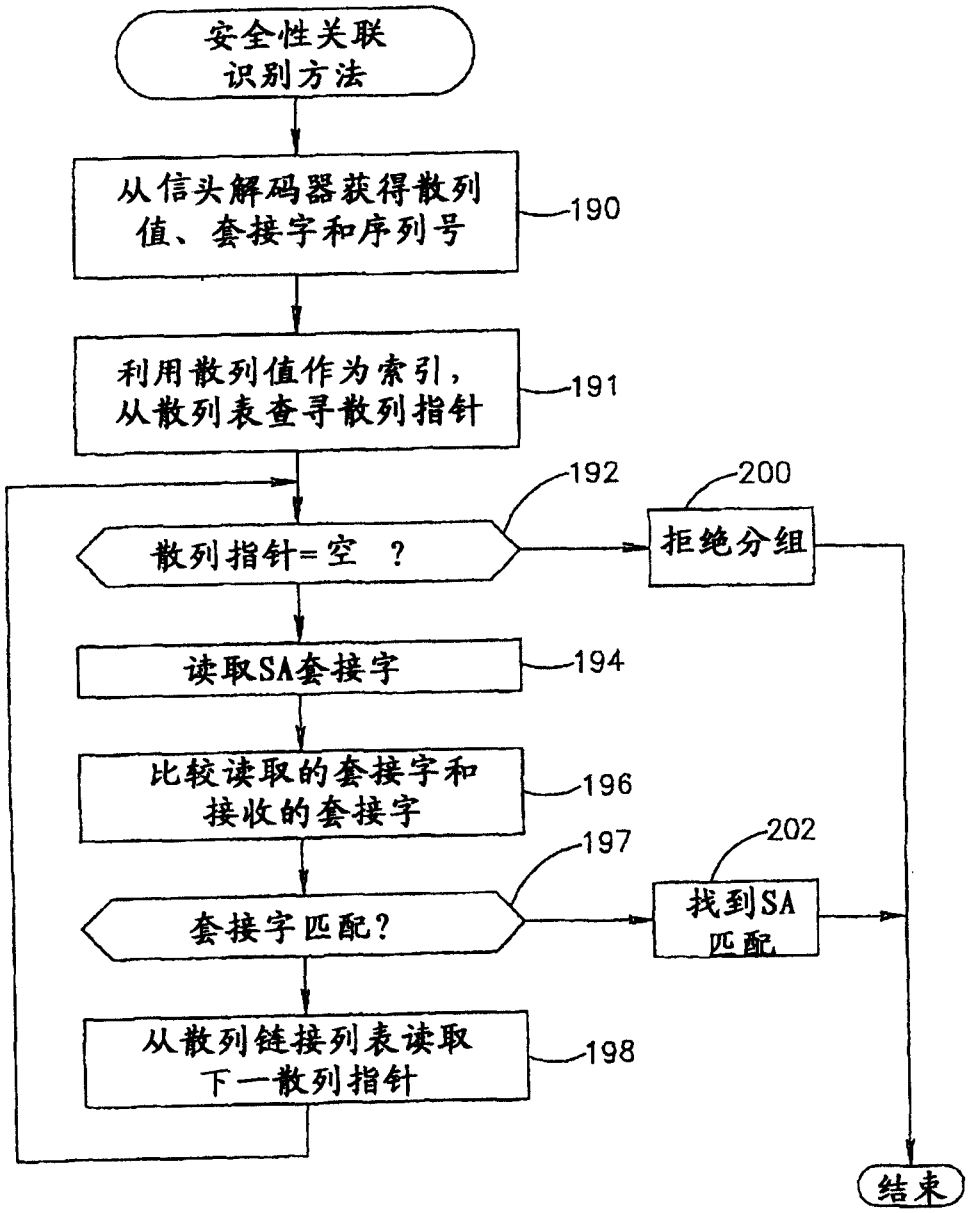

安全性关联
识别方法
从信头解码器获得散列
值、套接字和序列号
190
利用散列值作为索引，
从散列表查寻散列指针
191
192
200
散列指针=空 ？
拒绝分组
读取SA套接字
194
比较读取的套接字和
接收的套接字
196
197
202
套接字匹配？
找到SA
匹配
从散列链接列表读取
下一散列指针
198
结束

图6

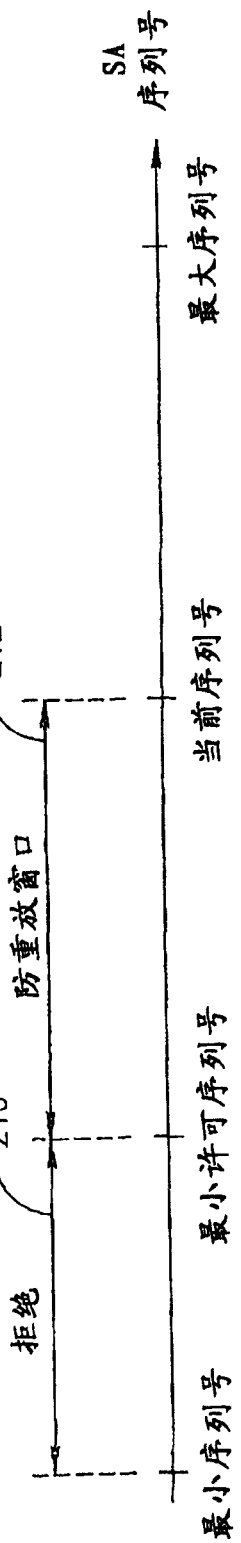


图7

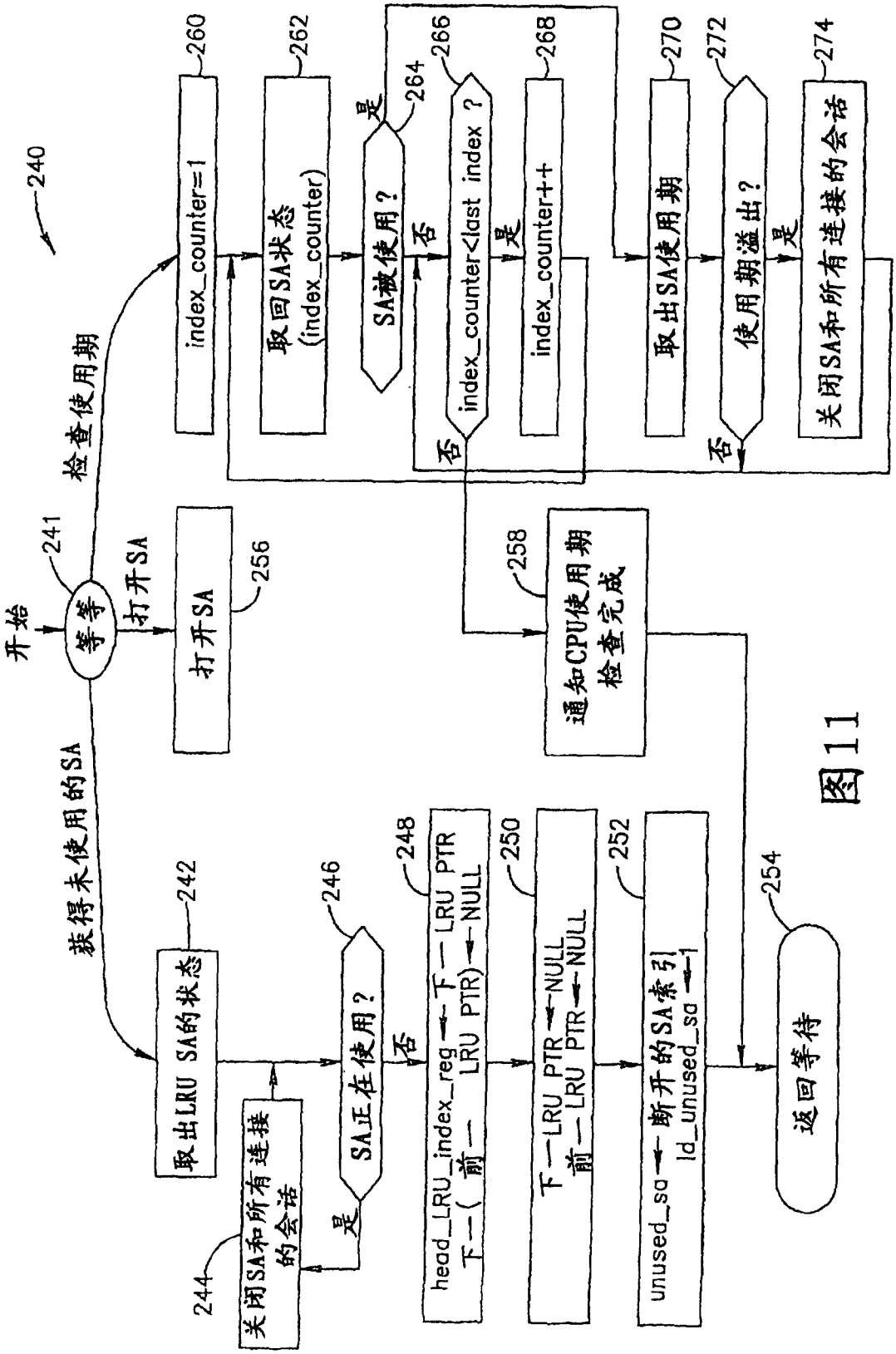

240
开始
等等
241
检查使用期
打开SA
打开SA
256
获得未使用的SA
取出LRU SA的状态
242
关闭SA和所有连接的会话
244
SA正在使用？
246
否
是
head_LRU_index_reg ← 下一 LRU PTR
下一（前一 LRU PTR）← NULL
248
下一LRU PTR ← NULL
前一LRU PTR ← NULL
250
unused_sa ← 断开的SA索引
Id_unused_sa ← 1
252
返回等待
254
通知CPU使用期检查完成
258
index_counter=1
260
取回SA状态
(index_counter)
262
SA被使用？
264
是
否
index_counter<last index ？
266
是
否
index_counter++
268
取出SA使用期
270
使用期溢出？
272
是
否
关闭SA和所有连接的会话
274

图 11

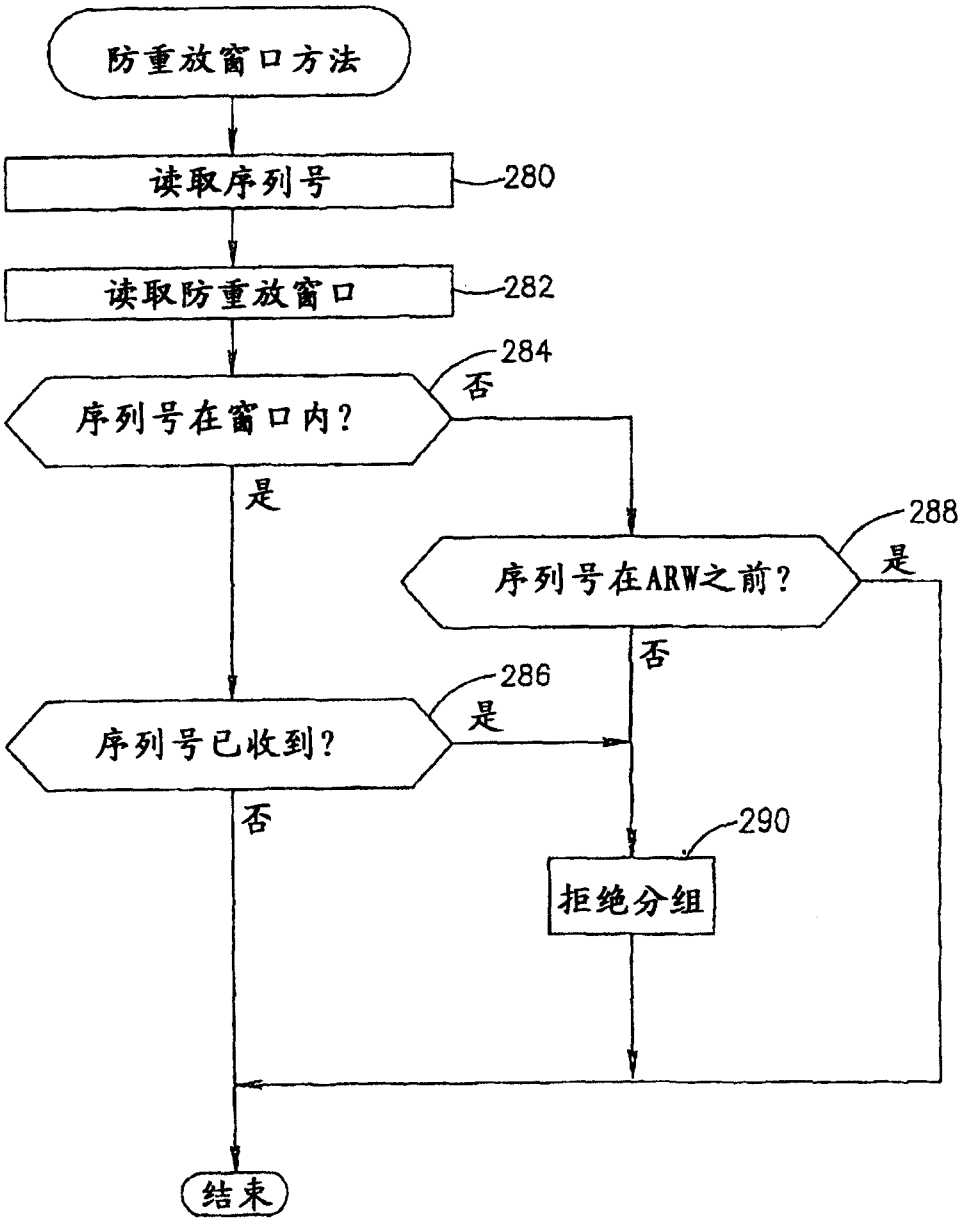

防重放窗口方法
读取序列号
280
读取防重放窗口
282
序列号在窗口内?
284
否
是
序列号在ARW之前?
288
是
否
序列号已收到?
286
是
否
拒绝分组
290
结束

图8

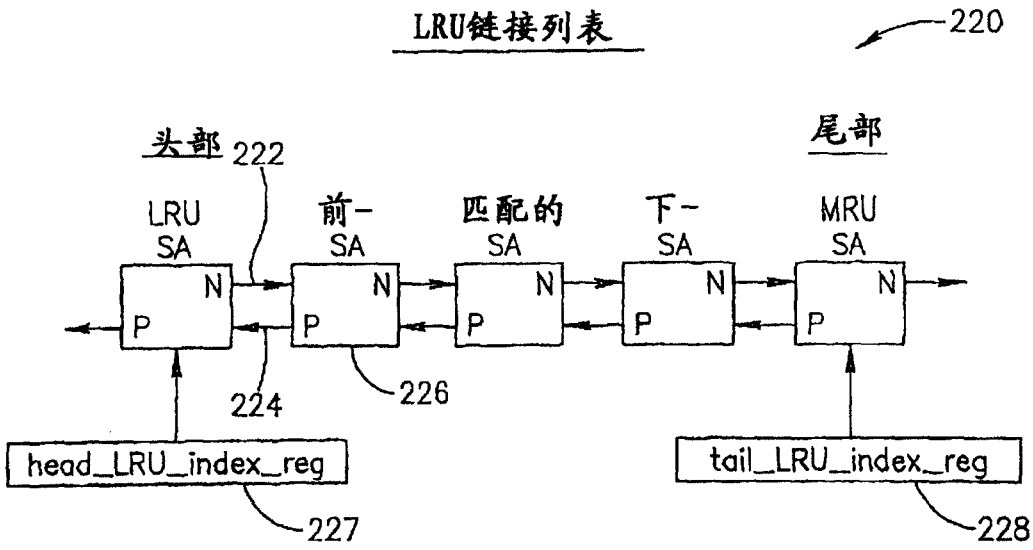

LRU链接列表
220
头部
222
尾部
LRU SA
前- SA
匹配的 SA
下- SA
MRU SA
N
P
224
226
head_LRU_index_reg
227
tail_LRU_index_reg
228

图9

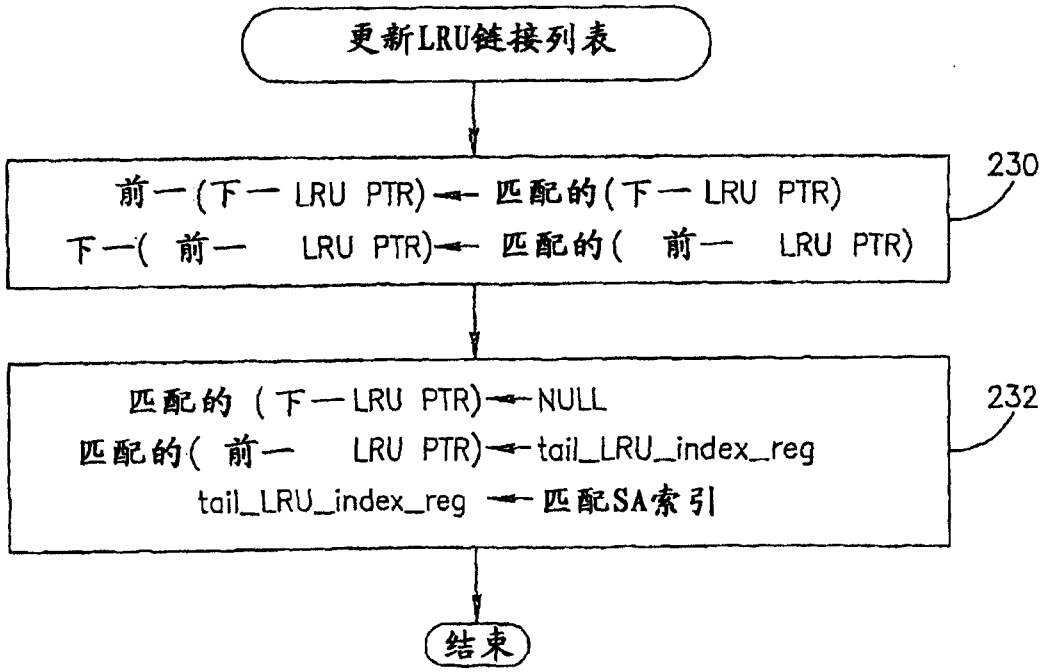

更新LRU链接列表
前一(下一 LRU PTR)← 匹配的(下一 LRU PTR)
下一(前一 LRU PTR)← 匹配的(前一 LRU PTR)
230
匹配的 (下一 LRU PTR)← NULL
匹配的(前一 LRU PTR)← tail_LRU_index_reg
tail_LRU_index_reg ← 匹配SA索引
232
结束

图10

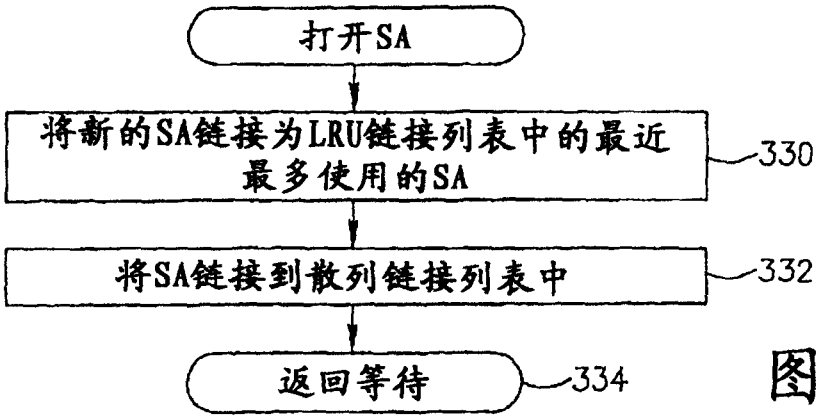

打开SA
将新的SA链接为LRU链接列表中的最近最多使用的SA
330
将SA链接到散列链接列表中
332
返回等待
334

图12

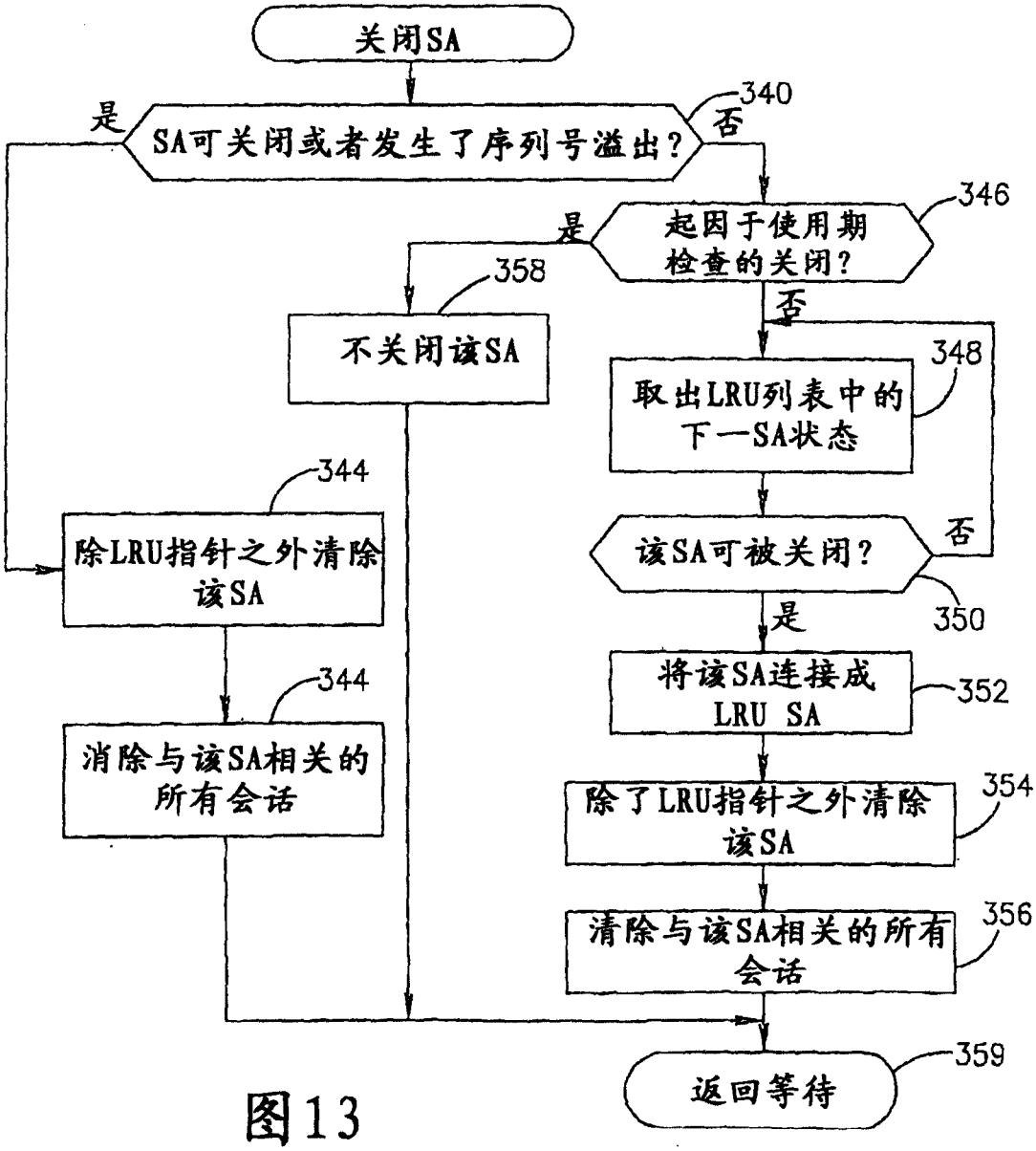

关闭SA
SA可关闭或者发生了序列号溢出?
340
是
否
起因于使用期检查的关闭?
346
是
否
不关闭该SA
358
取出LRU列表中的下一SA状态
348
除LRU指针之外清除该SA
344
该SA可被关闭?
否
350
是
将该SA连接成LRU SA
352
消除与该SA相关的所有会话
344
除了LRU指针之外清除该SA
354
清除与该SA相关的所有会话
356
返回等待
359

图13

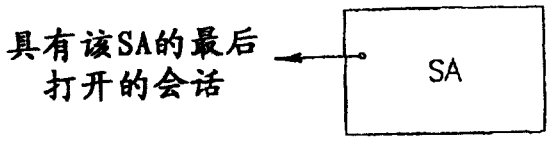

具有该SA的最后
打开的会话
SA

图14

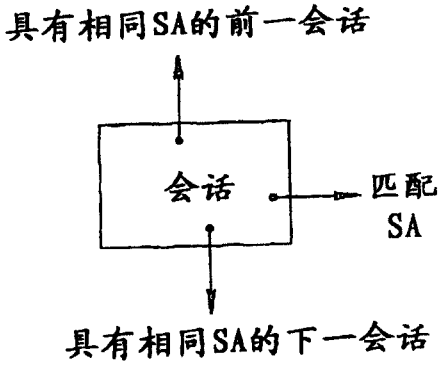

具有相同SA的前一会话
会话
匹配
SA
具有相同SA的下一会话

图15

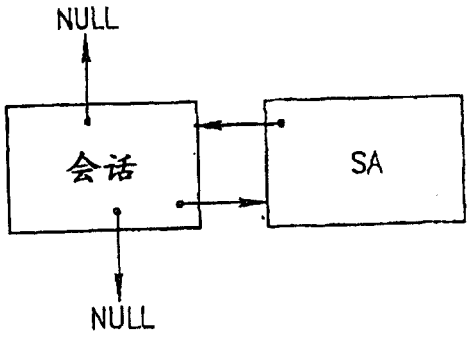

NULL
会话
SA
NULL

图16

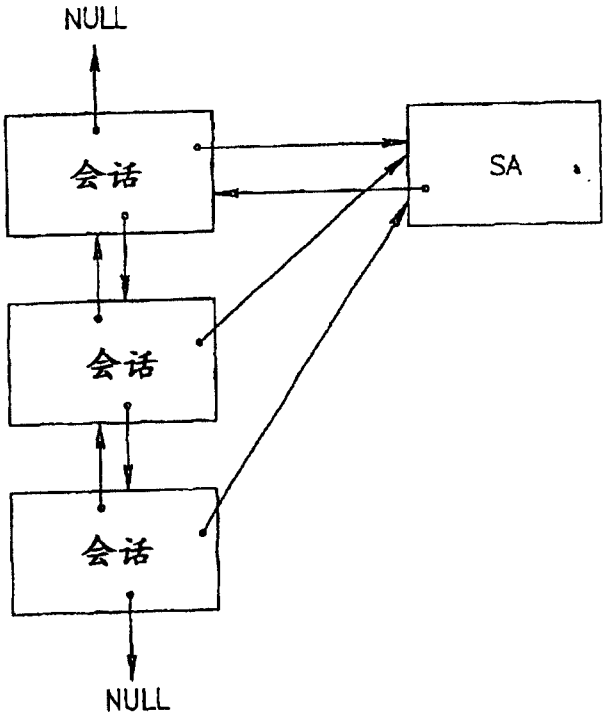


图17

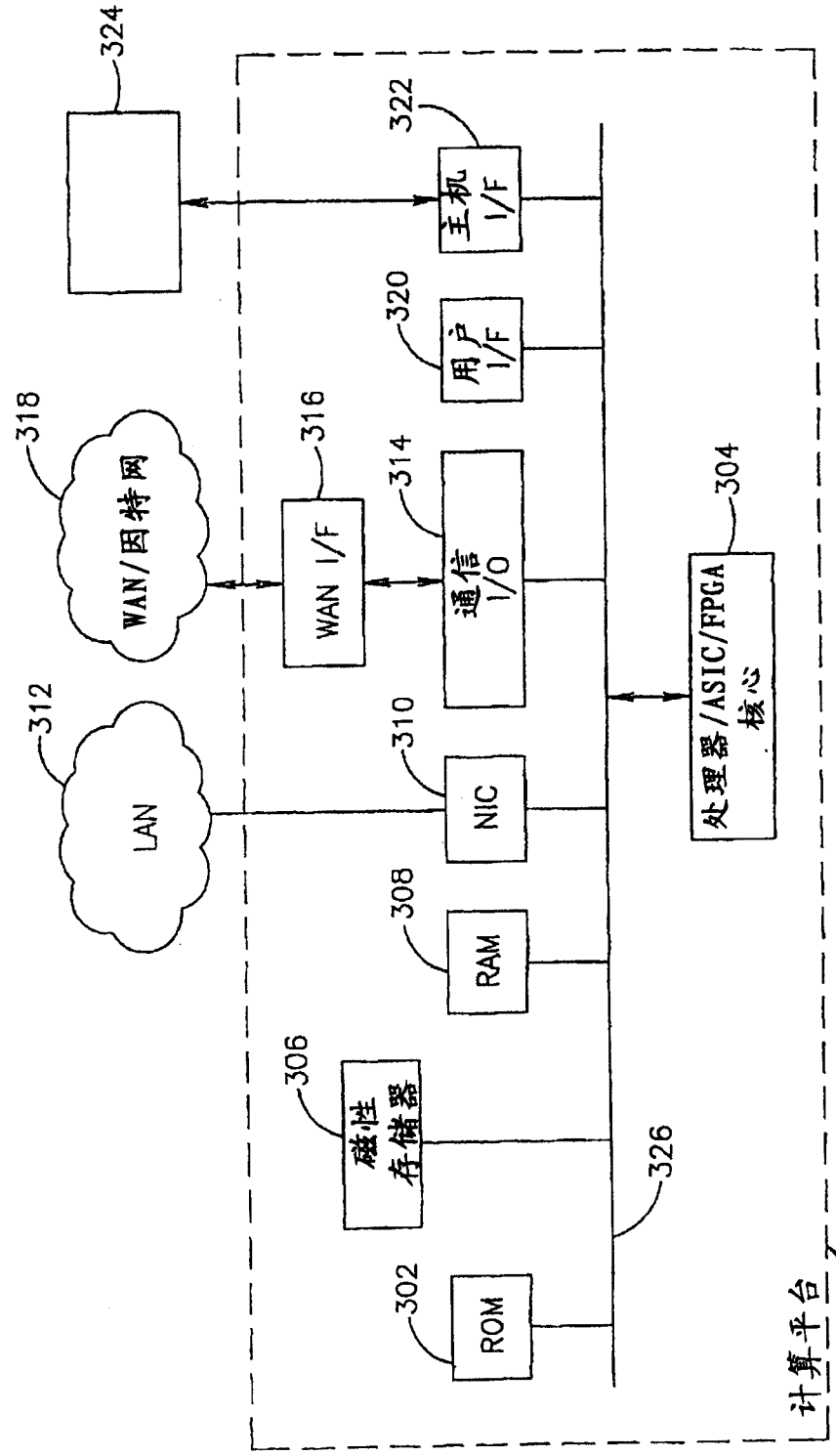

324
322
主机 I/F
320
用户 I/F
318
WAN/因特网
316
WAN I/F
314
通信 I/O
304
处理器/ASIC/FPGA 核心
312
LAN
310
NIC
308
RAM
306
磁性存储器
326
302
ROM
计算平台
300

图18