(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112052201 A
(43)申请公布日 2020.12.08

(21)申请号 202011030556.7

(22)申请日 2020.09.27

(71)申请人 中孚安全技术有限公司
地址 250101 山东省济南市高新区新泺大街1166号奥盛大厦2号楼2530室
申请人 中孚信息股份有限公司
北京中孚泰和科技发展股份有限公司
南京中孚信息技术有限公司

(72)发明人 刘晓萌 崔新安 袁浩

(74)专利代理机构 北京久维律师事务所 11582
代理人 邢江峰

(51)Int.Cl.
G06F 13/10(2006.01)
G06F 13/38(2006.01)
G06F 21/60(2013.01)

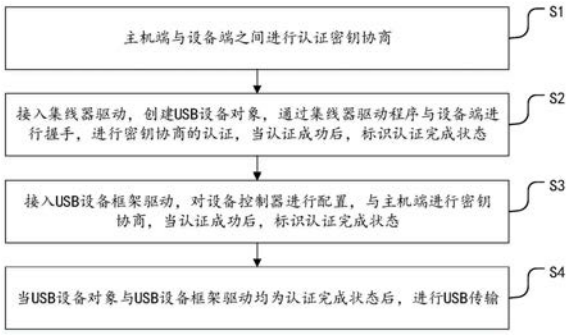
权利要求书2页 说明书5页 附图1页

(54)发明名称

一种基于Linux内核层实现的USB设备管控方法与系统

(57)摘要

本发明提供了一种基于Linux内核层实现的USB设备管控方法与系统，本发明通过在设备枚举过程中增加认证机制，分别在USB总线驱动程序和USB设备框架驱动增加数据加密机制，即接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行秘钥协商的认证以及接入USB设备框架驱动，对设备控制器进行配置，与主机端进行秘钥协商的认证，从而实现USB接口的传输安全，可为信息系统构建安全封闭的USB连接，解决因USB接口引入的安全隐患。



CN 112052201 A

1.一种基于Linux内核层实现的USB设备管控方法,其特征在于,所述方法包括以下操作:

S1、主机端与设备端之间进行认证秘钥协商,

S2、接入集线器驱动,创建USB设备对象,通过集线器驱动程序与设备端进行握手,进行秘钥协商的认证,当认证成功后,标识认证完成状态;

S3、接入USB设备框架驱动,对设备控制器进行配置,与主机端进行秘钥协商,当认证成功后,标识认证完成状态;

S4、当USB设备对象与USB设备框架驱动均为认证完成状态后,进行USB传输。

2.根据权利要求1所述的一种基于Linux内核层实现的USB设备管控方法,其特征在于,所述认证秘钥协商具体包括:向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。

3.根据权利要求1所述的一种基于Linux内核层实现的USB设备管控方法,其特征在于,所述步骤S2具体为:

集线器驱动程序检测到设备端接入集线器下行端口,为之创建USB设备对象,并通过复位、设置设备地址操作使设备进入地址状态;

集线器驱动程序使用标准设备请求与设备进行数次握手,完成认证密钥协商过程;

若认证通过,将会话密钥写入USB设备对象,并设置认证完成状态,集线器驱动程序继续收集设备信息,直至将USB设备对象添加到系统中,USB设备成功接入主机;若认证未通过,主机端集线器驱动程序释放该设备对象及相关资源,并临时禁用设备所连接的集线器端口,USB设备接入失败。

4.根据权利要求1所述的一种基于Linux内核层实现的USB设备管控方法,其特征在于,所述步骤S3具体为:

设备上电后,USB设备框架驱动对设备控制器进行配置,完成芯片初始化工作;

响应部分USB标准设备请求;

使用标准设备请求与主机进行认证密钥协商,若认证通过则记录会话密钥,并设置认证完成状态,此后使能其他请求的响应,USB设备框架驱动继续响应其他设备请求,完成设备配置,向功能单元提供端点访问服务,主机端成功连接USB设备;若认证未通过,禁用USB设备控制器,断开USB连接。

5.一种基于Linux内核层实现的USB设备管控系统,其特征在于,所述系统包括:

秘钥协商模块,用于主机端与设备端之间进行认证秘钥协商,

集线器驱动接入模块,用于接入集线器驱动,创建USB设备对象,通过集线器驱动程序与设备端进行握手,进行秘钥协商的认证,当认证成功后,标识认证完成状态;

框架驱动接入模块,用于接入USB设备框架驱动,对设备控制器进行配置,与主机端进行秘钥协商,当认证成功后,标识认证完成状态;

传输模块,用于当USB设备对象与USB设备框架驱动均为认证完成状态后,进行USB传输。

6.根据权利要求5所述的一种基于Linux内核层实现的USB设备管控系统,其特征在于,所述认证秘钥协商具体包括:向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。

7.根据权利要求5所述的一种基于Linux内核层实现的USB设备管控系统,其特征在于,所述集线器驱动接入模块包括:

设备对象创建单元,用于集线器驱动程序检测到设备端接入集线器下行端口,为之创建USB设备对象,并通过复位、设置设备地址操作使设备进入地址状态;

握手单元,用于集线器驱动程序使用标准设备请求与设备进行数次握手,完成认证密钥协商过程;

设备对象认证单元,用于若认证通过,将会话密钥写入USB设备对象,并设置认证完成状态,集线器驱动程序继续收集设备信息,直至将USB设备对象添加到系统中,USB设备成功接入主机;若认证未通过,主机端集线器驱动程序释放该设备对象及相关资源,并临时禁用设备所连接的集线器端口,USB设备接入失败。

8.根据权利要求5所述的一种基于Linux内核层实现的USB设备管控系统,其特征在于,所述框架驱动接入模块包括:

控制器配置单元,用于设备上电后,USB设备框架驱动对设备控制器进行配置,完成芯片初始化工作;

设备请求单元,用于响应部分USB标准设备请求;

框架驱动认证单元,用于使用标准设备请求与主机进行认证密钥协商,若认证通过则记录会话密钥,并设置认证完成状态,此后使能其他请求的响应,USB设备框架驱动继续响应其他设备请求,完成设备配置,向功能单元提供端点访问服务,主机端成功连接USB设备;若认证未通过,禁用USB设备控制器,断开USB连接。

一种基于Linux内核层实现的USB设备管控方法与系统

技术领域

[0001] 本发明涉及USB设备管控技术领域，特别是涉及一种基于Linux内核层实现的USB设备管控方法与系统。

背景技术

[0002] USB设备是当下最常用的可移动接口设备，其类型多种多样，几乎涵盖所有的可移动类型设备，易用性和多样性的特征使其成为当下最有可能造成数据泄露的外接设备，同时也就成为保密行业重点管控的设备类型。

[0003] USB设备的管控分为两种类型，一种是USB设备插入主机后背管控策略放行或阻断，另一种是正在被使用的USB设备被新管控策略放行或者禁用。通常情况下，USB设备的阻断和禁用才是保密行业重点关注的管控操作，针对Linux平台，亦是如此。而现有技术中，对于USB的管控容易出现数据泄露的问题，不能做到安全的有效防护。

发明内容

[0004] 本发明的目的是提供一种基于Linux内核层实现的USB设备管控方法与系统，旨在解决现有技术中USB设备数据泄露的问题，实现构建安全封闭的USB连接，解决因USB接口引入的安全隐患。

[0005] 为达到上述技术目的，本发明提供了一种基于Linux内核层实现的USB设备管控方法，所述方法包括以下操作：

[0006] S1、主机端与设备端之间进行认证密钥协商；

[0007] S2、接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行密钥协商的认证，当认证成功后，标识认证完成状态；

[0008] S3、接入USB设备框架驱动，对设备控制器进行配置，与主机端进行密钥协商，当认证成功后，标识认证完成状态；

[0009] S4、当USB设备对象与USB设备框架驱动均为认证完成状态后，进行USB传输。

[0010] 优选地，所述认证密钥协商具体包括：向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。

[0011] 优选地，所述步骤S2具体为：

[0012] 集线器驱动程序检测到设备端接入集线器下行端口，为之创建USB设备对象，并通过复位、设置设备地址操作使设备进入地址状态；

[0013] 集线器驱动程序使用标准设备请求与设备进行数次握手，完成认证密钥协商过程；

[0014] 若认证通过，将会话密钥写入USB设备对象，并设置认证完成状态，集线器驱动程序继续收集设备信息，直至将USB设备对象添加到系统中，USB设备成功接入主机；若认证未通过，主机端集线器驱动程序释放该设备对象及相关资源，并临时禁用设备所连接的集线器端口，USB设备接入失败。

[0015] 优选地，所述步骤S3具体为：

[0016] 设备上电后，USB设备框架驱动对设备控制器进行配置，完成芯片初始化工作；

[0017] 响应部分USB标准设备请求；

[0018] 使用标准设备请求与主机进行认证密钥协商，若认证通过则记录会话密钥，并设置认证完成状态，此后使能其他请求的响应，USB设备框架驱动继续响应其他设备请求，完成设备配置，向功能单元提供端点访问服务，主机端成功连接USB设备；若认证未通过，禁用USB设备控制器，断开USB连接。

[0019] 本发明还提供了一种基于Linux内核层实现的USB设备管控系统，所述系统包括：

[0020] 密钥协商模块，用于主机端与设备端之间进行认证密钥协商；

[0021] 集线器驱动接入模块，用于接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行密钥协商的认证，当认证成功后，标识认证完成状态；

[0022] 框架驱动接入模块，用于接入USB设备框架驱动，对设备控制器进行配置，与主机端进行密钥协商，当认证成功后，标识认证完成状态；

[0023] 传输模块，用于当USB设备对象与USB设备框架驱动均为认证完成状态后，进行USB传输。

[0024] 优选地，所述认证密钥协商具体包括：向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。

[0025] 优选地，所述集线器驱动接入模块包括：

[0026] 设备对象创建单元，用于集线器驱动程序检测到设备端接入集线器下行端口，为之创建USB设备对象，并通过复位、设置设备地址操作使设备进入地址状态；

[0027] 握手单元，用于集线器驱动程序使用标准设备请求与设备进行数次握手，完成认证密钥协商过程；

[0028] 设备对象认证单元，用于若认证通过，将会话密钥写入USB设备对象，并设置认证完成状态，集线器驱动程序继续收集设备信息，直至将USB设备对象添加到系统中，USB设备成功接入主机；若认证未通过，主机端集线器驱动程序释放该设备对象及相关资源，并临时禁用设备所连接的集线器端口，USB设备接入失败。

[0029] 优选地，所述框架驱动接入模块包括：

[0030] 控制器配置单元，用于设备上电后，USB设备框架驱动对设备控制器进行配置，完成芯片初始化工作；

[0031] 设备请求单元，用于响应部分USB标准设备请求；

[0032] 框架驱动认证单元，用于使用标准设备请求与主机进行认证密钥协商，若认证通过则记录会话密钥，并设置认证完成状态，此后使能其他请求的响应，USB设备框架驱动继续响应其他设备请求，完成设备配置，向功能单元提供端点访问服务，主机端成功连接USB设备；若认证未通过，禁用USB设备控制器，断开USB连接。

[0033] 发明内容中提供的效果仅仅是实施例的效果，而不是发明所有的全部效果，上述技术方案中的一个技术方案具有如下优点或有益效果：

[0034] 与现有技术相比，本发明通过在设备枚举过程中增加认证机制，分别在USB总线驱动程序和USB设备框架驱动增加数据加密机制，即接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行密钥协商的认证以及接入USB设备框架驱动，对设

备控制器进行配置，与主机端进行密钥协商的认证，从而实现USB接口的传输安全，可为信息系统构建安全封闭的USB连接，解决因USB接口引入的安全隐患。

附图说明

[0035] 图1为本发明实施例中所提供的一种基于Linux内核层实现的USB设备管控方法流程图；

[0036] 图2为本发明实施例中所提供的一种基于Linux内核层实现的USB设备管控系统框图。

具体实施方式

[0037] 为了能清楚说明本方案的技术特点，下面通过具体实施方式，并结合其附图，对本发明进行详细阐述。下文的公开提供了许多不同的实施例或例子用来实现本发明的不同结构。为了简化本发明的公开，下文中对特定例子的部件和设置进行描述。此外，本发明可以在不同例子中重复参考数字和/或字母。这种重复是为了简化和清楚的目的，其本身不指示所讨论各种实施例和/或设置之间的关系。应当注意，在附图中所图示的部件不一定按比例绘制。本发明省略了对公知组件和处理技术及工艺的描述以避免不必要地限制本发明。

[0038] 下面结合附图对本发明实施例所提供的一种基于Linux内核层实现的USB设备管控方法与系统进行详细说明。

[0039] 如图1所示，本发明实施例公开了一种基于Linux内核层实现的USB设备管控方法，所述方法包括以下操作：

[0040] S1、主机端与设备端之间进行认证密钥协商；

[0041] S2、接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行密钥协商的认证，当认证成功后，标识认证完成状态；

[0042] S3、接入USB设备框架驱动，对设备控制器进行配置，与主机端进行密钥协商，当认证成功后，标识认证完成状态；

[0043] S4、当USB设备对象与USB设备框架驱动均为认证完成状态后，进行USB传输。

[0044] USB建立连接即在主机端USB设备被添加到系统中以及在设备端功能单元能够访问设备端点缓冲区，主机端可通过缺省控制管道与设备端通信，若连接的另一端是恶意的，或者连接被监听，则会造成合法主机或合法设备内信息泄露。因此，必须保证只有合法主机与合法设备才能建立USB连接。

[0045] 在设备添加到系统中完成设备配置前，主机与设备间仅缺省控制管道可用，认证密钥协商需要进行传输控制，包括向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。认证密钥协商完成后主机端与设备端之间的数据传输需要加解密操作，并安全接入集线器驱动。

[0046] 集线器驱动程序检测到设备端接入集线器下行端口，为之创建USB设备对象，并通过复位、设置设备地址等操作使设备进入地址状态。集线器驱动程序使用扩展的标准设备请求与设备进行数次握手，完成认证密钥协商过程。若认证通过，将会话密钥写入USB设备对象，并设置认证完成状态，此后所有USB数据传输将以密文形式进行，集线器驱动程序继续收集设备信息，直至将USB设备对象添加到系统中，USB设备成功接入主机；若认证未通

过,主机端集线器驱动程序释放该设备对象及相关资源,并临时禁用设备所连接的集线器端口,USB设备接入失败。

[0047] 进行USB设备框架驱动的安全接入。设备上电后,USB设备框架驱动对设备控制器进行配置,完成芯片初始化工作,主要是控制端的配置。响应部分USB标准设备请求,如获取设备描述符、设置设备地址等。使用扩展的标准设备请求与主机进行认证密钥协商,若认证通过则记录会话密钥,并设置认证完成状态,此后使能其他请求的响应,USB设备框架驱动继续响应其他设备请求,完成设备配置,向功能单元提供端点访问服务,主机端成功连接USB设备;若认证未通过,禁用USB设备控制器,断开USB连接。

[0048] 设备端经认证密钥协商接入主机后,USB设备对象与USB设备框架驱动中均已被设置为认证完成状态,存储本次USB连接传输加密所需会话密钥,通过USB设备框架驱动对输入/输出请求进行过滤加解密,实现USB接口的传输安全。为各端设置统一的读写服务入口,在此服务调用设备控制器驱动读写端点缓冲区数据的过程中增加解密/加密机制。

[0049] 本发明实施例通过在设备枚举过程中增加认证机制,分别在USB总线驱动程序和USB设备框架驱动增加数据加密机制,即接入集线器驱动,创建USB设备对象,通过集线器驱动程序与设备端进行握手,进行密钥协商的认证以及接入USB设备框架驱动,对设备控制器进行配置,与主机端进行密钥协商的认证,从而实现USB接口的传输安全,可为信息系统构建安全封闭的USB连接,解决因USB接口引入的安全隐患。

[0050] 如图2所示,本发明实施例还公开了一种基于Linux内核层实现的USB设备管控系统,所述系统包括:

[0051] 密钥协商模块,用于主机端与设备端之间进行认证密钥协商;

[0052] 集线器驱动接入模块,用于接入集线器驱动,创建USB设备对象,通过集线器驱动程序与设备端进行握手,进行密钥协商的认证,当认证成功后,标识认证完成状态;

[0053] 框架驱动接入模块,用于接入USB设备框架驱动,对设备控制器进行配置,与主机端进行密钥协商,当认证成功后,标识认证完成状态;

[0054] 传输模块,用于当USB设备对象与USB设备框架驱动均为认证完成状态后,进行USB传输。

[0055] 在设备添加到系统中完成设备配置前,主机与设备间仅缺省控制管道可用,认证密钥协商需要进行传输控制,包括向设备端发送控制请求、传输控制请求相关的数据以及向主机端发送控制请求结果。认证密钥协商完成后主机端与设备端之间的数据传输需要加解密操作,并安全接入集线器驱动。

[0056] 所述集线器驱动接入模块包括:设备对象创建单元、握手单元以及设备对象认证单元。

[0057] 集线器驱动程序检测到设备端接入集线器下行端口,为之创建USB设备对象,并通过复位、设置设备地址等操作使设备进入地址状态。集线器驱动程序使用扩展的标准设备请求与设备进行数次握手,完成认证密钥协商过程。若认证通过,将会话密钥写入USB设备对象,并设置认证完成状态,此后所有USB数据传输将以密文形式进行,集线器驱动程序继续收集设备信息,直至将USB设备对象添加到系统中,USB设备成功接入主机;若认证未通过,主机端集线器驱动程序释放该设备对象及相关资源,并临时禁用设备所连接的集线器端口,USB设备接入失败。

[0058] 所述框架驱动接入模块包括:控制器配置单元、设备请求单元以及框架驱动认证单元。

[0059] 进行USB设备框架驱动的安全接入。设备上电后,USB设备框架驱动对设备控制器进行配置,完成芯片初始化工作,主要是控制端的配置。响应部分USB标准设备请求,如获取设备描述符、设置设备地址等。使用扩展的标准设备请求与主机进行认证密钥协商,若认证通过则记录会话密钥,并设置认证完成状态,此后使能其他请求的响应,USB设备框架驱动继续响应其他设备请求,完成设备配置,向功能单元提供端点访问服务,主机端成功连接USB设备;若认证未通过,禁用USB设备控制器,断开USB连接。

[0060] 设备端经认证密钥协商接入主机后,USB设备对象与USB设备框架驱动中均已被设置为认证完成状态,存储本次USB连接传输加密所需会话密钥,通过USB设备框架驱动对输入/输出请求进行过滤加解密,实现USB接口的传输安全。为各端设置统一的读写服务入口,在此服务调用设备控制器驱动读写端点缓冲区数据的过程中增加解密/加密机制。

[0061] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

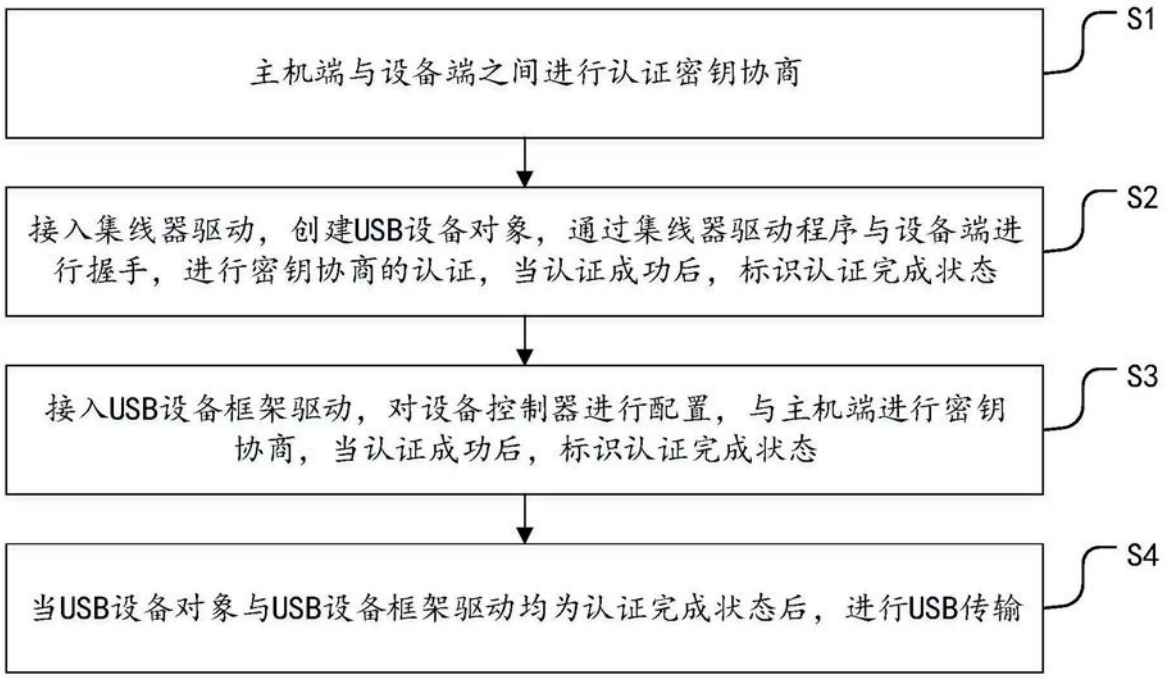

主机端与设备端之间进行认证密钥协商
S1
接入集线器驱动，创建USB设备对象，通过集线器驱动程序与设备端进行握手，进行密钥协商的认证，当认证成功后，标识认证完成状态
S2
接入USB设备框架驱动，对设备控制器进行配置，与主机端进行密钥协商，当认证成功后，标识认证完成状态
S3
当USB设备对象与USB设备框架驱动均为认证完成状态后，进行USB传输
S4

图1


密钥协商模块
集线器驱动接入模块
框架驱动接入模块
传输模块

图2