Journal of System Simulation

Volume 28 | Issue 6 Article 21

6-8-2020

Design and Implementation of Secure USB Connection

Songyin Zhao
Information Engineering University, Zhengzhou 450000, China;

Bin Yu
Information Engineering University, Zhengzhou 450000, China;

Design and Implementation of Secure USB Connection

Abstract

Abstract: Aiming at the deficiency of present secure solutions countering USB-based Hardware Trojan Horse, BadUSB, bus wiretapping and other new attack technologies, a secure connection scheme applying to any USB device was proposed. *In scheme, mutual authentication and key exchange were implemented by extending the standard device request in the hub driver and USB device framework driver to prevent the creation of malicious USB connection. Meanwhile, all the data transferred of the created USB connection would be filter encrypted by USB bus driver and USB device framework driver to rebel the monitor of bus.* The experiment results show that *the proposed scheme can be used to build secure sealing USB connection for an information system, solving the secure threats introduced by USB interface*.

Keywords

USB, hardware trojan, BadUSB, wiretapping attack, hub, driver

Recommended Citation

Zhao Songyin, Yu Bin. Design and Implementation of Secure USB Connection[J]. Journal of System Simulation, 2016, 28(6): 1400-1405.

第 28 卷第 6 期 系统仿真学报© Vol. 28 No. 6
2016 年 6 月 Journal of System Simulation Jun., 2016

USB 安全连接方案设计与实现

赵松银，郁滨
(信息工程大学，河南 郑州 450000)

摘要：针对当前安全防护方案在应对 USB 硬件木马、BadUSB、总线窃听等新型攻击技术方面的不足，设计了一种与设备种类无关的 USB 安全连接方案。*方案通过扩展标准设备请求，在 USB 连接建立过程中由集线器驱动程序与 USB 设备框架驱动进行双向认证与密钥协商，在数据传输过程中由 USB 总线驱动与 USB 设备框架驱动对 I/O 请求进行过滤加解密，实现了独立于设备种类的 USB 接入与传输安全。实验结果表明，方案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。*

关键词：USB；硬件木马；BadUSB；窃听攻击；集线器；驱动
中图分类号：TP309.1 文献标识码：A 文章编号：1004-731X (2016) 06-1400-06

Design and Implementation of Secure USB Connection

Zhao Songyin, Yu Bin
(Information Engineering University, Zhengzhou 450000, China)

Abstract: Aiming at the deficiency of present secure solutions countering USB-based Hardware Trojan Horse, BadUSB, bus wiretapping and other new attack technologies, a secure connection scheme applying to any USB device was proposed. *In scheme, mutual authentication and key exchange were implemented by extending the standard device request in the hub driver and USB device framework driver to prevent the creation of malicious USB connection. Meanwhile, all the data transferred of the created USB connection would be filter encrypted by USB bus driver and USB device framework driver to rebel the monitor of bus.* The experiment results show that *the proposed scheme can be used to build secure sealing USB connection for an information system, solving the secure threats introduced by USB interface.*

Keywords: USB; hardware trojan; BadUSB; wiretapping attack; hub; driver

引言

USB(Universal Serial Bus，通用串行总线)的设计初衷主要是为了满足人们对计算机外设接口在方便性、通用性等方面的需求，在安全性方面并未做考虑[1]。按照 USB 设备框架，USB 设备被抽象为一个由配置、接口、端点组成的集合，在同一时刻只能使用一种配置，却可以支持多个接口。当设备接入时，系统便会自动识别、配置设备并为每个接口加载独立的驱动程序。这种设计在方便设备使用、丰富设备功能的同时，也使 USB 成为了攻击、窃密的重要途径。

目前 USB 主机、设备的安全防护机制研究多针对某一类设备设计专用的电路、软件，如大容量存储设备类[2-4]。虽然这种防护思路可以结合设备

收稿日期：2015-04-30 修回日期：2015-07-21；基金项目：信息保障技术重点实验室开放基金(KJ-14-103)；河南省科技攻关项目(132102210003)
作者简介：赵松银(1989-)，男，河南新乡，硕士生，研究方向为 USB、信息安全技术；郁滨(1964-)，男，河南郑州，博士，教授，博导，研究方向为信息安全、无线网络安全技术、视觉密码等。

功能进行深度优化，但实质上仅对 USB 设备的某一种接口进行了安全管控，无法应对基于其他种类 USB 设备设计的攻击技术。John Clark 等[1]便利用当前安全防护系统未关注的 USB 信道(Unintended USB Channel)，USB 键盘、播放器所用的接口，设计了 USB 硬件木马，成功获取计算机的控制权限并窃取数据。更著名的一例是 Karsten Nohl 等[5]于 2014 年 8 月在美国黑帽子大会上所展示的 BadUSB 攻击，利用 USB 支持多种接口的特点，通过固件升级、逆向工程等手段在固件中增加新接口并植入恶意代码，在提供正常功能的同时，成功获取操作系统的控制权限、实现 DNS 的重定向等，隐蔽性极高。USB Meta-Device[6]则基于 USB 测试技术，利用各类设备驱动程序中的代码漏洞，通过编程枚举为相关设备，实现对主机的入侵。不仅如此，部分 USB 设备功能的实现还需要系统中其他模块的支持，如 U 盘在加载 USB 大容量存储类驱动程序的同时还需要磁盘驱动、文件系统驱动、资源管理器等众多程序的配合，才能正常工作，这些代码中的漏洞都是潜在的安全威胁[7-8]。以上攻击、窃密技术能够得逞的根本原因便是，USB 主机在枚举过程中信任设备提供的信息，未进行安全认证便加载驱动并配置相关资源，使之处于工作状态。

此外，USB 协议还存在传输安全问题，An Wang 等[9]不同于以上先攻击再窃取数据的思路，设计实现的 USB 线缆监听器能够以搭接的形式直接获取 USB 总线上所传输的数据。对此，张宇[10]、李翠[11]结合椭圆曲线密码体制(ECC)高安全性分别设计了主机与 USB 设备间的双向认证密钥协商协议，并基于 USB 设备控制器 IP 核(Intellectual Property Core)进行实现。但主机端仍需设计单独的设备驱动，不能实现对所有种类 USB 设备的安全接入与传输。

综上所述，设计一种独立于设备种类的 USB 安全连接方案，在为设备分配总线资源前对其进行安全认证，在数据发送至总线前先进行加密，实现 USB 接口的接入与传输安全，对于解决信息系统因采用 USB 接口而面临的攻击与信息泄漏问题具有重要意义。

1 安全连接模型建立

主机与设备中的 USB 子系统独立于具体的设备功能，负责设备的识别、配置机制并管理 USB 总线的数据传输。本文方案通过在其中的相关驱动中增加认证密钥协商与数据传输加密服务实现设备种类无关的安全接入与安全传输，所设计的 USB 安全连接模型如图 1 所示。

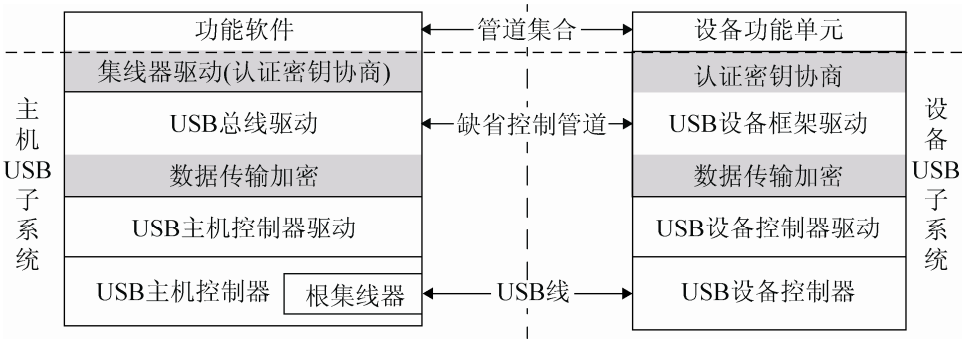

图 1 USB 安全连接模型

模型中，控制器驱动程序负责隐藏控制器的硬件实现细节，其软件接口只有 USB 总线驱动程序(Universal Serial Bus Driver，USBD)和 USB 设备框架驱动(USB Device Framework Driver，UDFD)可以访问。USBD 使用一种与操作系统设备驱动模型相关的数据结构——USB 设备对象(USB Device Object，UDO)对设备的状态与数据传输进行管理，操作系统其他模块则通过访问 UDO 调用

第 28 卷第 6 期 系统仿真学报 Vol. 28 No. 6
2016 年 6 月 Journal of System Simulation Jun., 2016

USBD 对设备进行控制与传输数据。UDFD 则通过操作各端点实现设备在枚举过程中定义的状态转换与功能单元通信的管控。新设备的 UDO 由集线器驱动创建并用来配置设备，被添加到系统中后其他模块方知新设备的接入。

本文中 USB 连接建立即指：在主机端，代表新设备的 UDO 被添加到系统中；在设备端，功能单元能够访问设备端点缓冲区。因为此时主机功能软件已可通过缺省控制管道与设备功能单元通信，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的机密性与完整性破坏。安全连接方案必须保证只有合法主机与合法设备才能建立 USB 连接，且所建立的连接由会话密钥对所传数据进行加密保护。

为此，方案由集线器驱动程序在 UDO 被添加到系统前与 UDFD 进行认证密钥协商，认证通过则在 UDO 与 UDFD 中记录会话密钥、设置认证通过标志为传输加密做准备；否则，销毁 UDO 并断开物理连接。传输安全在主机端通过 USBD 对输入/输出请求包(I/O Request Packet，IRP)进行过滤加密，在设备端通过 UDFD 对设备控制器驱动(Device Controller Driver, DCD)提供的端点读写数据进行加解密实现。两者结合，信息系统内 USB 主机/设备通过 USB 接口，将只能以密文形式与合法的设备/主机进行通信，使系统内信息的机密性与完整性得到保护。

2 USB 接入安全设计

2.1 标准设备请求扩展

在设备 UDO 被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，认证密钥协商需使用控制传输进行。控制传输可包括建立阶段、数据阶段和状态阶段 3 个阶段，其中建立阶段用于向设备发送控制请求，一个 8 字节的 SETUP 数据包；数据阶段负责传输控制请求相关的数据，部分控制请求没有该阶段；状态阶段用于向主机报告前 2 个阶段的传输结果。方案通过建立阶段传输认证密钥协商命令，数据阶段传输相应的握手数据，所扩展的标准设备请求如表 1 所示。

表 1 认证密钥协商请求定义

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10 000 000 B	GET_HANDSHAKE	0	握手次数	握手数据量	握手数据
00 000 000 B	SEND_HANDSHAKE	0	握手次数	握手数据量	握手数据

GET_HANDSHAKE， SEND_HANDSHAKE 数值与 USB 协议中已有标准设备请求不同即可，分别要求设备在第 wIndex 次握手中发送或接收 wLength 字节的握手数据 Data。

2.2 集线器驱动安全接入流程

认证密钥协商完成后主机与设备的数据传输需要进行加解密操作，而完成前则不须要。为便于管理不同设备的认证密钥协商状态与存储会话密钥，方案对 UDO 进行扩展，分别增加认证完成状态字段与会话密钥字段。考虑到部分认证密钥协商用时较长，而所有新设备均需使用缺省设备地址 0，方案将认证密钥协商过程放在地址状态后进行，USB 设备的安全接入流程如图 2 所示。

Step 1 集线器驱动程序检测到新设备接入某集线器下行端口，为之创建 UDO，并通过复位、设置设备地址等操作使设备进入地址状态。

Step 2 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。

Step 3 若认证通过，将会话密钥写入 UDO，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，转向 Step4。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。

Step 4 集线器驱动程序继续收集设备信息，直

至将 UDO 添加到系统中，USB 设备成功接入主机。

此后，由其他系统软件继续对设备进行配置。

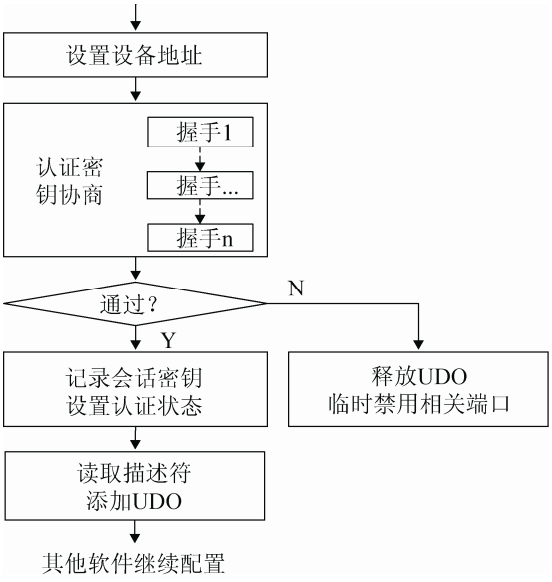

图 2 集线器驱动安全接入流程

2.3 UDFD 安全接入流程

USB 连接建立过程中 UDFD 的工作流程如图 3 所示。

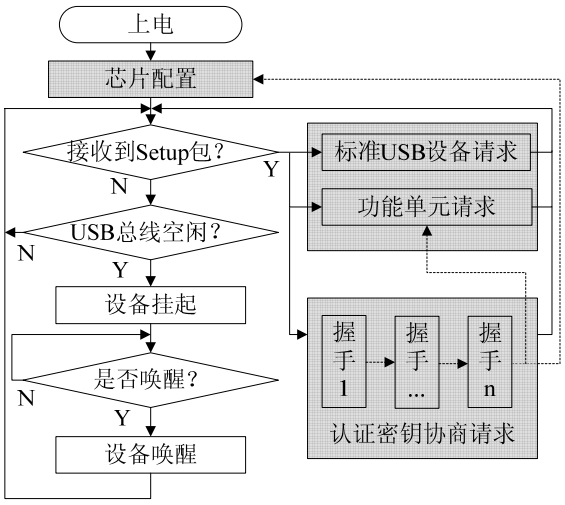

图 3 UDFD 安全接入流程

Step 1 设备上电后，UDFD 对设备控制器进行配置，完成芯片初始化工作，主要是控制端点 0 的配置。

Step 2 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。

Step 3 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，转向 Step4。若认证未通过，禁用 USB 设备控制器，断开 USB 连接。

Step 4 此后，UDFD 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。主机成功连接 USB 设备。

3 USB 传输安全设计

设备经认证密钥协商接入主机后，UDO 与 UDFD 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。USB 安全传输设计即是通过 USBD/UDFD 对输入/输出请求进行过滤加解密，实现 USB 接口的传输安全。

3.1 USBD 传输加解密流程

USBD 使用 IRP 与上层程序和主机控制器驱动程序(Host Controller Driver, HCD)进行通信。本文通过在向 HCD 提交 IRP 的例程中对输出 IRP 中的相关数据字段及缓冲区进行加密实现数据的密文发送，对输入 IRP 则设置完成例程，待设备数据输入完成后解密数据并写入 IRP 所指明的缓冲区中返回给上层程序。具体流程如图 4 所示。

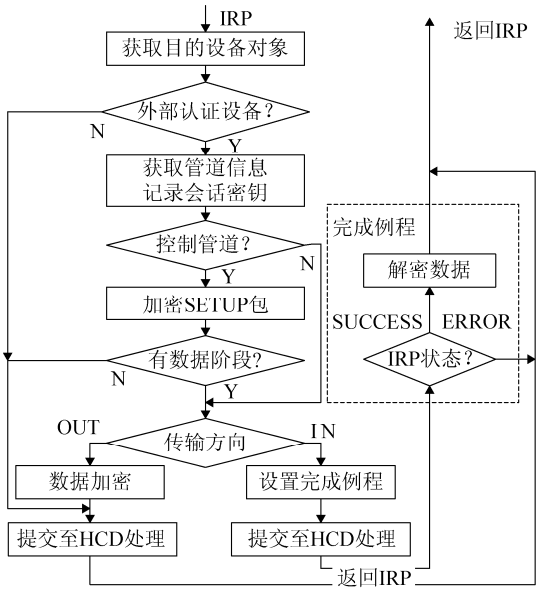

图 4 USBD 传输加解密流程

Step 1 USBD 接收到上层程序的 IRP，获取其目的 UDO。若是通过认证的某外部 USB 设备的 UDO，则须进行数据加解密处理，转向 Step2；否则表明该 IRP 的目的通信设备是根集线器或认证密钥协商尚未完成的新设备，无须进行加解密操作，转向 Step5 交 HCD 处理即可。

Step 2 读取 UDO 中的会话密钥与此 IRP 所用的管道，若是控制管道，转向 Step3；否则转向 Step4。

Step 3 使用控制管道的 IRP 中已存有 SETUP 包，以使设备获悉本次控制传输是否有数据阶段及数据阶段的传输方向。此处加密 SETUP 包并判断有无数据阶段，若无数据阶段则将 IRP 直接交 HCD 处理，转向 Step5；否则转向 Step4。

Step 4 判断此 IRP 在控制传输数据阶段的传输方向，或所用管道的传输方向(除控制管道外，其他管道均只有一个固定的传输方向)。若是 IN，则此时 IRP 所属数据缓冲区中还没有数据，需要设置完成例程，由完成例程待 IRP 返回后再解密；若是 OUT，则加密数据缓冲区中的数据。

Step 5 将 IRP 下传给 HCD，由 HCD 进行实际的数据传输，完成后返回 IRP 执行结果。

完成例程：USBD 在完成例程中重新获得 IRP 的控制权，若 IRP 为成功状态则将 IRP 所指明缓冲区中的数据解密后返回该 IRP，否则直接返回。

3.2 UDFD 传输加解密流程

UDFD 传输加解密的流程相对简单，首先为各端点设置统一的读写服务入口，在此服务调用 DCD 读写端点缓冲区数据的过程中增加解密/加密机制即可，如图 5 所示。由前文设计可知，认证密钥协商未完成时，UDFD 尚未向功能单元提供端点读写服务，仅限于 UDFD 自己读写端点 0 缓冲区，对部分标准设备请求及认证密钥协商请求进行处理的情况。

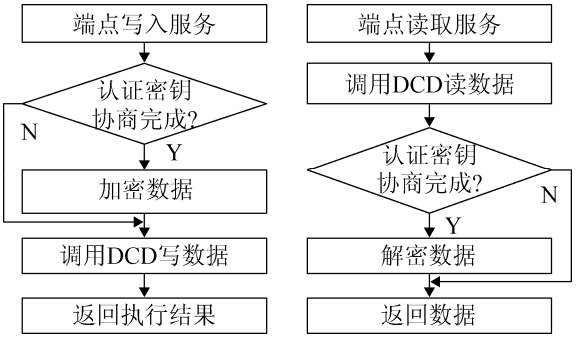

图 5 UDFD 传输加解密流程

4 实验与结果分析

依据设计的 USB 安全连接方案，本文改进了 Linux 2.6.22 内核中的 USB CORE(实现集线器驱动程序、USBD 等功能的内核模块)作为安全 USB 主机，通过在 usb_new_device()中循环使用扩展的标准设备请求调用 usb_control_msg()发起控制传输进行认证密钥协商，协商密钥及认证状态由 usb_device(实现 UDO 的数据结构)记录，在 usb_submit_urb( )，usb_hcd_giveback_urb( )中对 URB (USB Request Block，USB 请求块)的相关数据块进行加解密处理，运行环境为 Fedora 7 操作系统。设备端则采用基于 CY7C68013A USB 设备控制器芯片自主设计的硬件测试平台，编写固件实现 UDFD 等程序，通过下载不同的固件制作安全 USB 设备或普通 USB 设备。测试场景如图 6 所示。

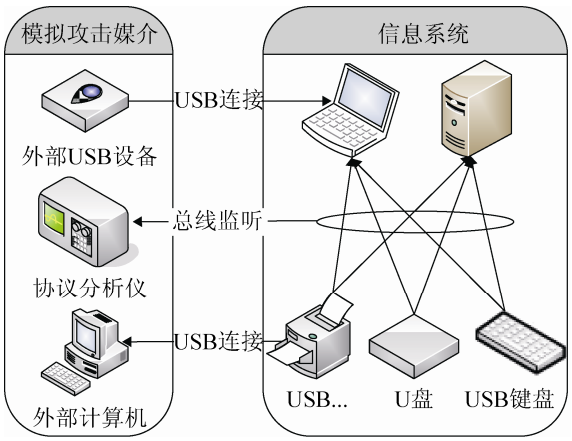

图 6 测试场景

实验选取了分属不同 USB 设备类的键盘(HID 设备类)、U 盘(大容量存储设备类)、USB 集线器(集

线器设备类)进行测试。其中受 USB 设备控制器芯片限制，普通 USB 集线器为成熟产品，未进行安全 USB 集线器测试。各次测试中主机与设备运行的 USB CORE 和固件设置如表 2 所示。

表 2 测试组合设置

测试编号	主机	设备
1	安全 USBCORE	安全 U 盘固件
2	安全 USBCORE	安全键盘固件
3	普通 USBCORE	安全 U 盘固件
4	普通 USBCORE	安全键盘固件
5	安全 USBCORE	普通 U 盘固件
6	安全 USBCORE	普通键盘固件
7	安全 USBCORE	普通集线器

在主机端使用 shell 命令#cat/proc/bus/usb/devices 查看各测试中 USB 设备的接入情况发现仅 1 号与 2 号测试成功建立了 USB 连接，再使用 LeCroy USB 协议分析仪对其总线信号进行监听，发现 USB 线上所传数据包(测试中用到的 DATA0 与 DATA1)中的数据均为密文。

实验结果验证了 USB 安全连接方案的设备种类无关性，可实现对任意 USB 设备的接入与传输安全，与文献[10-11]相比可完全抵抗旁路攻击、监听攻击等，如表 3 所示。

表 3 安全性对比

方案	数据源	旁路攻击	监听攻击
本文	USB 主机	√	√
	USB 设备	√	√
文献[10]	USB 主机	×	×
	USB 设备	√	√
文献[11]	USB 主机	×	×
	USB 设备	√	√

5 结论

本文在深入研究 USB 设备枚举与通信过程的基础上，设计了一种与设备种类无关的 USB 安全连接方案。该方案通过在设备枚举过程中增加认证机制实现了 USB 接口的接入安全，通过在 USBD、UDFD 中增加数据加密机制实现 USB 接口的传输安全。测试结果表明，该方案可有效解决传统安全方案应对不力的 USB 硬件木马、BadUSB、总线窃听等安全问题，为计算机与设备构建安全封闭的 USB 连接，解决信息系统因采用 USB 接口而引入的安全隐患。

参考文献:

[1] Clark J, Leblanc S, Knight S. Compromise through USB-based Hardware Trojan Horse device [J]. Future Generation Computer Systems(S0167-739X), 2011, 27(5): 555-563

[2] Butler K R B, McLaughlin S E, McDaniel P D. Kells: a protection framework for portable data [C]// Proceedings of the 26th Annual Computer Security Applications Conference. USA: ACM, 2010: 231-240.

[3] 夏天河. 基于WDM过滤驱动的USB访问控制系统的研究与实现 [D]. 重庆: 重庆大学, 2012.

[4] Yang F, Wu T, Chiu S, et al. A secure control protocol for USB mass storage devices [J]. IEEE Transactions on Consumer Electronics (S0098-3063), 2010, 56(4): 2239-2343.

[5] S K Karsten Nohl, J Lell. BadUSB-On accessories that turn evil [Z/OL]. (2014-07-15) [2015-04-30]. https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf. 2014.

[6] Barral D Dewey. Plug and root, The USB Key to the Kingdom [Z/OL]. (2005-07-27) [2015-04-30]. http://www.blackhat.com/presentations/bh-usa-05/BH_US_05-Barrall-Dewey.pdf. 2005.

[7] Wang A, Li Z, Yang X, et al. A New Security Problem of USB: Monitoring Cable Attack and Countermeasures [C]// Proceedings of the 2012 International Conference on Information Technology and Software Engineering. Heidelberg,Germany: Springer Berlin, 2012: 129-137.

[8] Jodeit M, Johns M. USB Device Drivers: A Stepping Stone into your Kernel [C]// Computer Network Defense (EC2ND), 2010 European Conference on. Los Alamitos: IEEE, 2010: 46-52.

[9] Schumilo S, Spenneberg R. Don't trust your USB! How to find bugs in USB device drivers [Z/OL]. (2014-09-28) [2015-04-30].https://www.blackhat.com/docs/eu-14/materials/eu-14-Schumilo-Dont-Trust-Your-USB-How-To-Find-Bugs-In-USB-Device-Drivers-wp.pdf. 2014.

[10] 杨先文, 李峥, 王安, 等. 密码安全USB设备控制器IP的系统设计 [J]. 华中科技大学学报, 2010, 38(9): 59-62.

[11] 李翠. 安全USB设备控制器设计与实现 [D]. 郑州: 解放军信息工程大学, 2013.