密 码 卫 士

（PassGuard）

安全输入控件系统技术白皮书

北京微通新成网络科技有限公司

2013 年

目　录

1. 什么是安全控件

简单说，安全控件是一种为了提升账户安全性，防止账户密码等私密信息被木马或病毒窃取的客户端程序，安全控件大多小巧，可以通过 ActiveX 或者手工下载的方式安装。

2. 为什么要使用安全控件

随着 2009 年 CCTV2 对“大小姐”木马案件的系列报道，标志着当前的“木马经济产业链”已经非常成熟，已经成为不可忽视的互联网安全和社会性问题。因此，如何让用户的帐号信息输入更安全，如何防止木马和病毒盗取用户帐号信息成为互联网信息安全的首要问题，在这样的背景下，要求我们必须通过安全控件这种技术手段加强对用户帐号信息的安全保护。

安全控件一般通过以下方式保护用户输入信息的安全：

1） **模拟密码框(防止读取密码框)；**

2） **模拟(屏幕)键盘(防止监听键盘输入)；**

3） **扫描后台程序，检测 Hook 程序。**

关于钩子（Hook）：*钩子（Hook）是盗号密码的常用伎俩，日常生活中，鱼钩是用来钓鱼的，一旦鱼咬了钩，钩子就一直钩住鱼了，任凭鱼在水里怎么游，也逃不出鱼钩的控制。同样的，Windows 的钩子 Hook 也是用来钩东西的，比较抽象的是他是用来钩 Windows 事件或者消息的。最常见的就是鼠标和键盘钩子，用 Hook 钩子钩住鼠标、键盘，当你的鼠标、键盘有任何操作时，通过 Hook 就能知道他们都做了什么了。*

3. 安全控件现状

经过北京微通新成网络科技有限公司的长期跟踪和大量研究，当前各家网银、第三方支付公司等所使用的安全控件存在明显的安全隐患，不能很好的保护用户的帐号信息安全，主要体现在：

1）安全控件面对内核级到应用级的多种钩子无能为力；

2）内存动态捕获的木马更是防不胜防；

3）软键盘保护技术，没有有效的防止后台录、截屏软件工作。

Windows 系统是建立在事件驱动的机制上的，也就是整个 Windows 系统都是通过消息的传递来实现的。而钩子是 Windows 系统中非常重要的系统接口，用它可以截获并处理传递给其他应用程序的消息，来完成普通应用程序难以实现的功能。钩子可以监视系统或进程中的各种事件消息，截获发往目标窗口的消息并进行处理。

因此，可以在系统中安装自定义的钩子，监视系统中特定事件的发生，完成特定的功能，比如截获键盘、鼠标的输入，屏幕取词，日志监视等等。可见，利用钩子可以实现许多特殊而有用的功能，很多黑客也正是利用了钩子技术制作出各式各样的木马，用于窃取用户的帐号信息。

为了避免各种钩子程序获取用户帐号信息，安全控件就必须优先于各种钩子程序处理用户的输入信息以及用户端屏幕显示（屏显）。经北京微通新成网络科技有限公司的长期跟踪研究，可以确认目前的各种钩子程序主要通过两种方式获取用户的帐户信息：第一类是键盘

类 Hook 程序，包括从应用层到核心层多种 Hook 方式；第二类是屏显 Hook 程序。具体分类见下表：

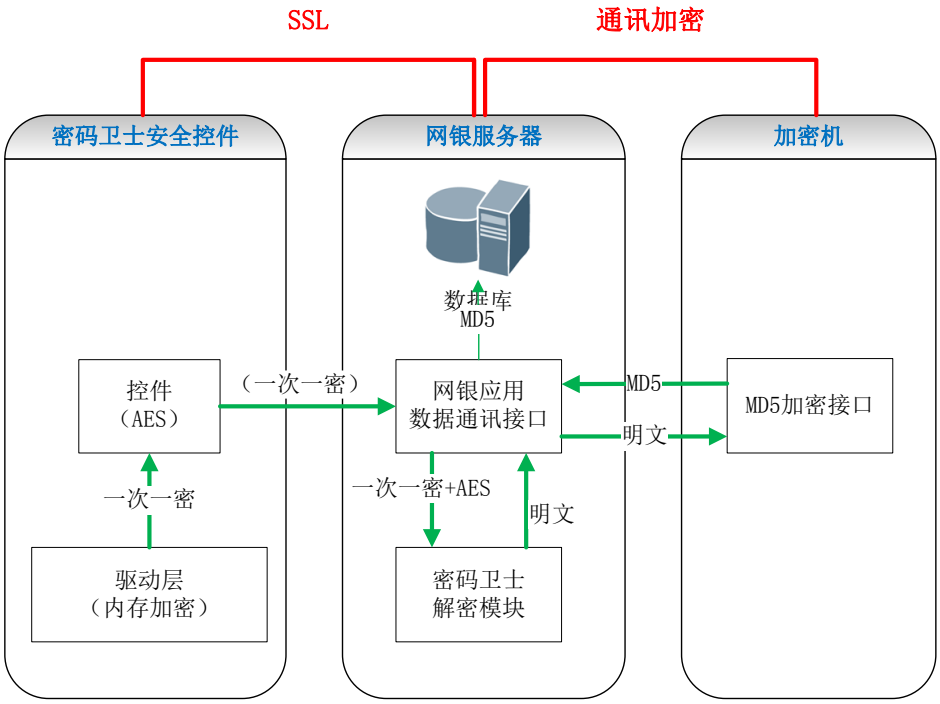
钩子类型	所用对象、API 或函数名称	所处位置
COM 接口调用	IE COM 接口	应用层
API Hook	Windows Procedure (WM_CHAR, WM_KEYDOWN)	应用层
	API 函数（PostMessage）hook	应用层
消息钩子	SendMessage(WM_GETTEXT)	应用层
	SetWindowsHookEX(WH_KEYBOARD)	应用层
	SetWindowsHookEx(WH_KEYBOARD_LL)	应用层
API 调用	GetKeyboardState	应用层
	GetAsyncKeyState	应用层
	GetKeyState	应用层
标准设备过滤驱动	键盘过滤驱动（FDO）	设备过滤驱动层
键盘类驱动	针对 PS/2 键盘类驱动（kbdClass）Dispatch Routine Hook； 针对 USB 键盘类驱动（hidusb、usbhub 等）	类驱动层
Inline Hook	Kbdclass dispatch routine(IRP Hook)	内核驱动层
	KeyboardClassServiceCallback	内核驱动层
	I8042KeyboardInterruptServiceRoutine	内核驱动层
OBJECT Hook	_KINTERRUPT	内核对象

IDT Hook	键盘中断（IRQ1, hook IDT）	IDT
	IOAPIC	IOAPIC 可编程芯片
端口读写	键盘设备（轮询 i8042 芯片）端口	内核驱动层
屏幕显示类	GetFrontBufferData	Direct3D
	DirectDrawSurface 等	DirectDraw
	BitBlt、StretchBlt	GDI

4. 密码卫士安全输入控件系统

密码卫士安全输入控件系统（以下简称密码卫士）是北京微通新成网络科技有限公司集多种顶尖安全技术研发的新一代密码保护产品，具有高度的安全性和广泛的可扩展性，解决了目前大多数密码保护产品保护不了密码的尴尬问题。

4.1. 密码卫士系统架构



如上图所示，密码卫士由密码卫士安全控件（客户端）和密码卫士解密模块（服务器端）组成。

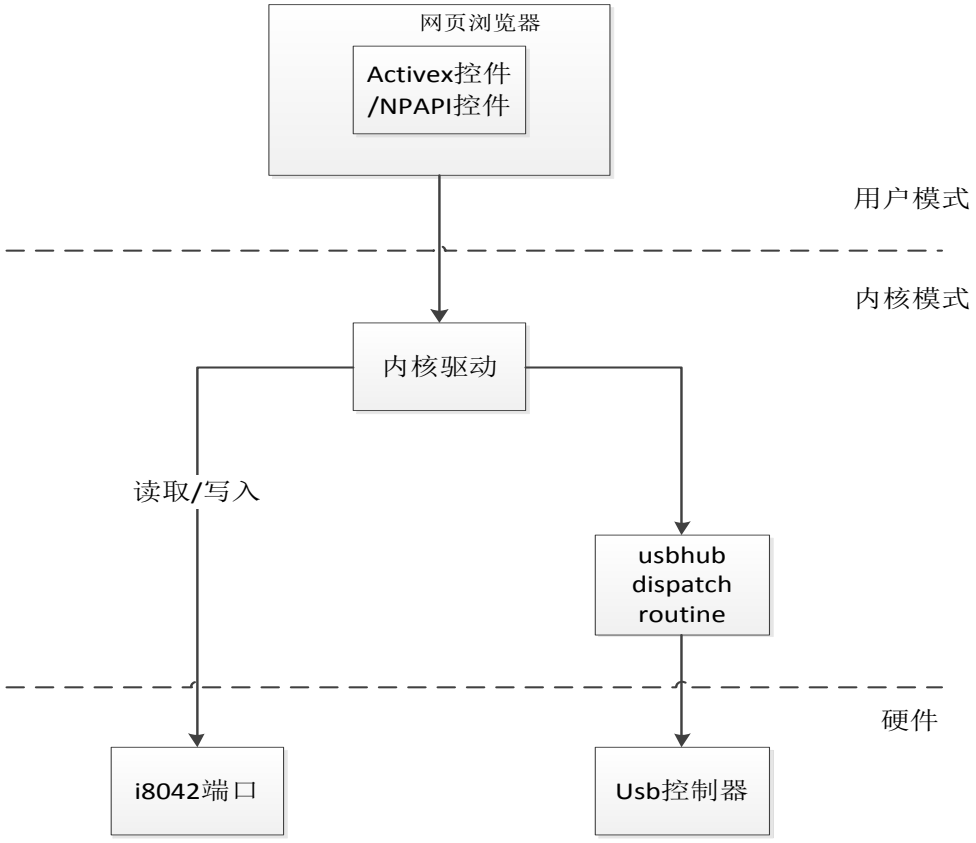
密码卫士安全控件通过控件安装或下载安装的方式安装到网银用户的 PC 机上，密码卫士安全控件在系统底层挂载驱动，在驱动层将按键数据一次一密传送给应用层，再在应用层组合密码并将密码进行 AES 加密，加密后的数据再一次一密通过浏览器 SSL 通道提交到网银服务器。

密码卫士解密模块部署在网银应用服务器上，负责解密由客户端浏览器经过 SSL 通道上传的一次一密的密码数据。解密后的明文数据仍转经网银应用交由加密机进行转加密处理或者直接交由加密机解

密并转加密，最后应用服务器将转加密后的密文存储或进行校验。

密码卫士一次一密的机制支持对称与非对称加密，链式加密，还支持客户端集成硬件加密机的 Lib 库函数对用户密码进行 RSA/SM2 非对称加密后，再进行一次一密的对称加密后，上传至服务端由加密机进行解密与转加密。

客户端分为用户态模块和内核态驱动模块。客户端架构如下图所示。



客户端软件架构

用户态模块，通过向驱动模块请求用户输入数据，来完成用户输入的采集。驱动通过底层保护机制拦截用户输入，并等待用户态模块请求数据时提交。双方通信过程采用私有加密协议处理，完全绕开操作系统的键盘输入处理流程，形成自有的旁路机制，从而保证用户输

入的数据安全。

4.2. 密码卫士安全控件技术原理

密码卫士安全控件主要由以下两个模块组成：

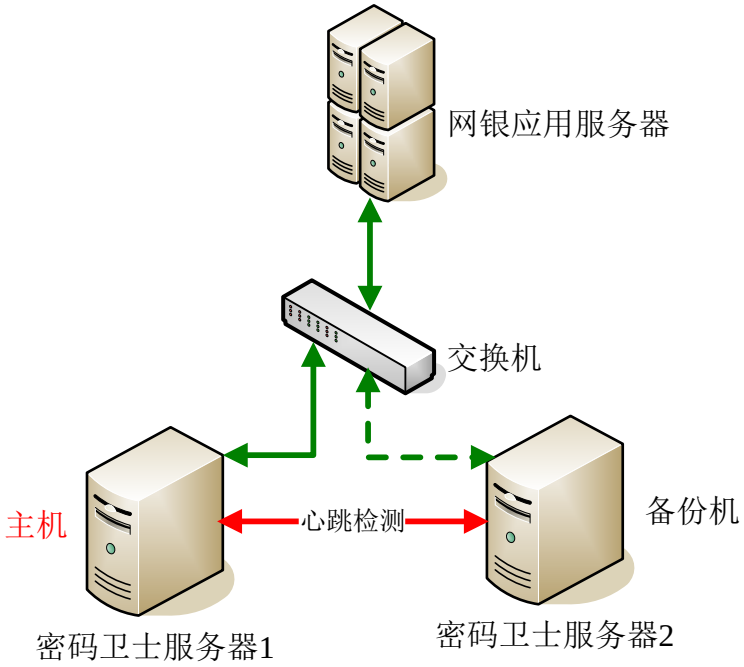
1）键盘输入安全保护模块；

2）屏幕显示安全保护模块。

密码卫士采用了最新的驱动技术和中断技术，优先于其它的键盘 Hook 程序和屏显 Hook 程序处理用户的键盘输入和屏幕显示，因而能够防范各种类型的 Hook 程序。密码卫士在底层对截获的真实数据进行清除，截断操作系统处理逻辑，通过旁路技术直接提交给用户态模块进行处理，安全可靠性高。

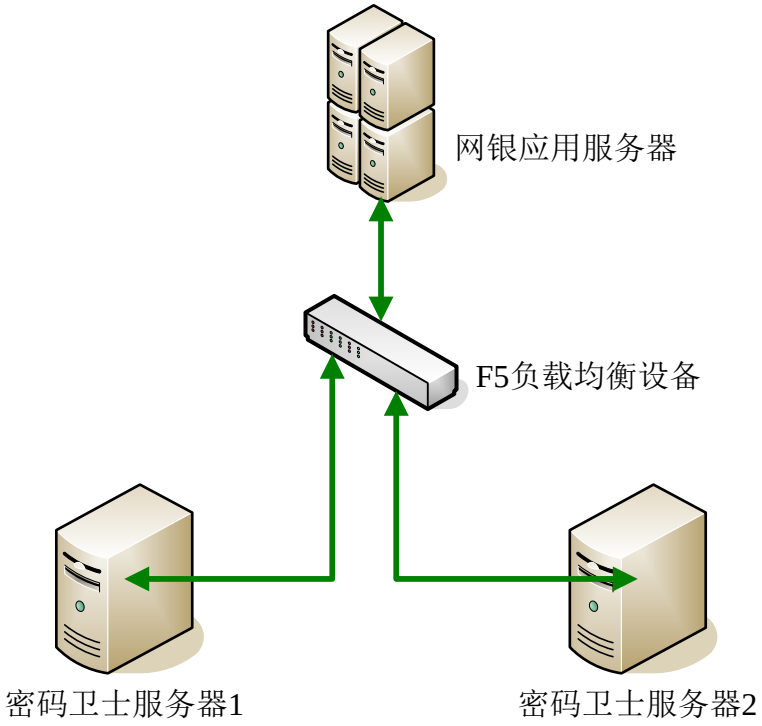
4.3. 密码卫士系统部署

4.3.1 双机部署



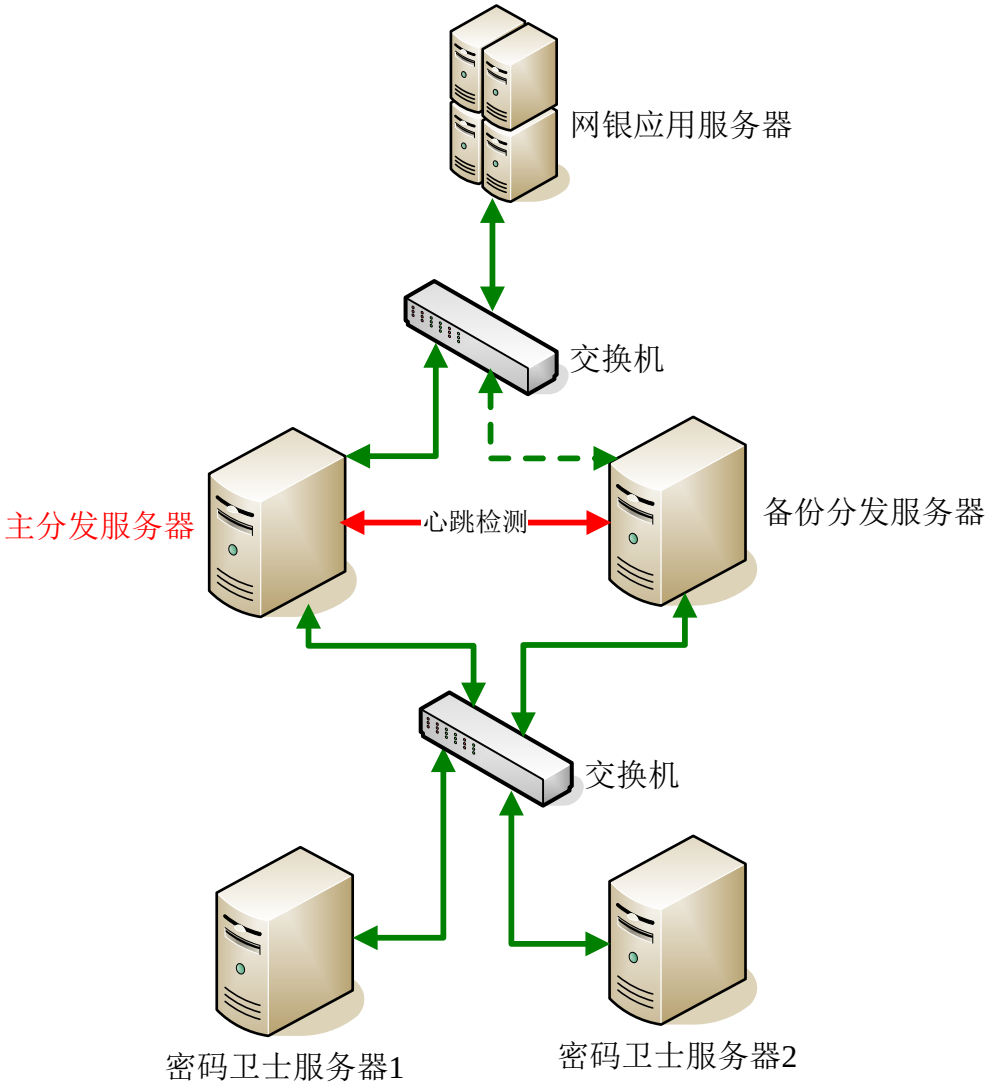
此方式采用 Linux、Heartbeat，实现双机热备，提高了系统稳定性。

4.3.2 集群部署



集群方案 1：

在应用系统所用负载均衡设备端口可用的条件下，直接挂接负载均衡设备，实现硬件负载均衡，提高了系统的并发处理能力和系统稳定性。



集群方案 2：

采用 Linux、LVS、Heartbeat，实现软件负载均衡，提高系统的并发处理能力和系统稳定性，此方案中的密码卫士服务器 1 和密码卫士服务器 2 可复用为主分发服务器和备份分发服务器。

4.4. 密码卫士系统的实施

4.4.1 产品安装

自动安装：当用户访问带有安全控件的网页时，网页自动提示用户在线安装密码卫士安全控件。

手动安装：通过网站下载可执行文件，手动安装。

4.4.2 开发改造与集成

控件集成：控件通过 JavaScript 脚本集成到应用系统中，可以设置控件的样式、输入规则等功能。

服务器端解密模块集成：

1、**集成部署：**服务器端解密模块以库文件形式提供给应用系统程序，应用系统程序直接集成调用解密模块接口；控件解密模块支持的开发语言如：JAVA、.NET、PHP、ASP、C、C++，支持操作系统如：Windows Server、Linux、Aix、Unix。

2、**独立部署：**可以独立部署在单独一组服务器上，通过 API 接口供应用系统程序调用。

3、**硬件加密机集成：**服务器端先用 AES 解密模块解密后，将加密机客户端加密函数加密的密文通过加密机 API 交给加密机，进行解密和转加密等处理。

4.4.3 产品发布与更新

当有新版本发布后，只用修改应用系统程序中控件集成脚本的版

本号，并替换应用系统下的 ActiveX 控件文件和供用户下载的手动安装文件。当用户再次访问带有安全控件的应用时，网页会自动提示用户安装升级新控件。

4.4.4 密码卫士安全控件升级机制

密码卫士账户敏感信息安全解决方案的主要优势在于密码卫士安全控件的高安全防护能力和高强度自身安全性，鉴于账户信息安全攻防形式的日益严峻，有必要定期更新密码卫士安全控件的敏感信息防护机制和自身保护机制，微通新成公司承诺至少每 3 个月对密码卫士安全机制进行一次更新，更新内容包含以下的一项或多项：

实时跟踪国内外最新的盗号木马并分析盗号木马所采用的技术，提前应对可能出现的新型攻击行为；

定期修复密码卫士安全控件可能的自身漏洞；

定期更新密码卫士安全控件信息加密存储方式；

定期更新密码卫士安全控件的动态实时保护程序；

定期更新反加载、反逆向、反跟踪、反调试、断点(软、硬、内存)检测机制；

定期更新构造异常转向、花指令、代码混淆、数据加密和 API 函数的方法；

定期更新加密算法；

定期更新加壳程序。

4.5. 密码卫士的功能和优势

密码卫士功能模块示意图：

密码卫士产品主要功能：

防止木马程序利用 COM 接口调用记录用户键盘输入；

防止木马程序利用 Windows 消息伪造类软件攻击；

防止木马程序利用 Windows 应用层 API 函数（如 PostMessage）hook 方式记录用户键盘输入；

防止木马程序通过 GetAsynKeyState 等 API 调用方式记录用户键盘输入；

防止木马程序通过调用 SetWindwosHookEx 函数 WH_KEYBOARD_LL 等（消息钩子）方式记录用户键盘输入；

防止木马程序利用内核设备过滤技术记录用户键盘输入；

防止木马程序利用键盘内核驱动及内核对象技术记录用户键盘输入；

防止木马程序利用键盘中断（hook IDT）截取技术记录用户键盘输入；

防止木马程序利用键盘设备（轮询 i8042）端口访问技术记录用户键盘输入；

防止木马程序利用 USB 过滤驱动以及 irp hook 技术记录用户 USB 键盘输入；

防止木马程序通过 Direct3D、DirectDraw、GDI 等三类图形接口截取或录制用户屏幕。

通过进程保护与窗口保护机制，避免用户受到黏虫式攻击，规避网络钓鱼风险，保护用户的帐号和密码。

注：黏虫式攻击，通过在用户真实输入用户名、密码页面窗口前端覆着一个伪造的用户不易辨识的假登录框，诱骗用户输入真实的用户名、密码，达到截获用户账户信息的攻击方式。

其他功能：

支持自定义正则表达式验证用户输入（支持用户输入验证和输入结果验证）；

支持密码强度检测（强、中、弱控件返回值）；

支持密码长度获取；

支持自定义字体；

支持自定义前景色、背景色；

支持自定义边框颜色；

支持自定义输入长度限制（最大长度、最小长度）（密码长度、最小密码长度由页面脚本处理、控件返回密码长度值）；

支持回车键、TAB 键、密码强度事件处理、支持 Caps lock 键状

态提示；

禁用编辑指令和功能键的命令，防止用户无意泄密。可防止系统剪切、复制、粘贴命令获取或修改键盘输入；

支持机器指纹获取（客户端的 CPU、硬盘序列号、或者网卡 MAC 地址）

服务器端支持硬件加密机进行解密（目前已支持江南所与卫士通加密机解密）。

密码卫士产品优势：

安全性高

从内核级到应用级的立体保护机制（ring0、ring3），无已知安全隐患；

提供灵活可选的软键盘密码输入方案（全面防截录屏）；

数据采用分块独立加密机制，内存中无明文敏感信息，安全性更高；

通过内存读写保护，避免内存热补丁的程序篡改方式；

自身安全性高，具有高强度抗调试、抗逆向、抗反汇编能力；

通过输入窗口保护与进程保护避免黏虫式攻击（一种网络钓鱼方式）。

兼容性强

兼容国内外主流的主机防病毒软件、防火墙软件、主动防御软件、第三方安全控件、密保类软件、网络游戏客户端安全软件、下载类工具软件、网吧管理软件等。

易用性好

安全控件用户可以在线安装，也可下载安装，安装完成后，无需重启，直接使用；

Windows 7/8 操作系统下无需用户以管理员权限运行 IE 浏览器，即可安装。

操作系统与浏览器支持广泛

支持 Windows 2003/XP/Vista/2008/7/8 操作系统；

支持 Windows Vista/7/8 64 位操作系统；

支持 Mac 操作系统（10.5 以上）；

支持 Linux 操作系统，包括 Ubuntu、Fedora、CentOS 等多种发行版，桌面分别支持 Gnome 和 KDE。

支持 Windows 操作系统上的如下浏览器：

支持 IE6 以上所有版本及所有 IE 内核浏览器（包括遨游、腾讯 TT、世界之窗、360 安全浏览器等，支持搜狗 2.0 等双核浏览器）；

支持 Firefox/Opera/Safari/Chrome 等非 IE 内核浏览器；

支持苹果操作系统上的如下浏览器：

Safari（5.1/5.0 以上）、Camino、Cruz、Firefox、Folck、Google Chrome、Icab、Omniweb、Shiira、Stainless、Sunrise 等浏览器。

也可定制支持专用客户端应用程序。

支持 iOS 与 Android 系统上的 APP 应用程序集成安全控件。

服务端可支持的平台以及编程语言：

可支持 Java 平台、.Net 平台、支持 java、jsp、asp、asp.net、

php、C 及 C#。

可扩展性

控件字体、背景、边框大小、颜色可自定义，贴近用户；

4.6. 密码卫士的跨平台应用

4.6.1 Mac OS 系统

背景分析：

Mac OS 系统存在如下的键盘输入安全隐患：

应用层的 event monitor，这个分为全局事件监听和本地事件监听，全局事件监听可以获取其他 app 的用户输入（不可修改，不能监听本进程），本地事件监听是针对本应用进行输入监听（可修改，不能监听其他应用）。全局和本地事件监听一般结合使用，对于一般的用户输入都可以获取，但是 np 插件和类型为 password 的 textfield 的输入被屏蔽了。

内核驱动，通过编写内核扩展，attach 键盘设备对象，hook 回调函数，类似 Windows 的 keyboardclassservicecallback。更底层的驱动技术，可以研究键盘设备的堆栈，和 Mac 系统的源码。但是，安装这些内核扩展需要管理员权限，即，需要用户输入口令。

其他的方法，如和 Windows 类似的方法，因为 Mac 上也有进程注入，所以，也存在一些 patch 的方式。

还有就是内存数据明文存储的问题。

解决方案：

在 Mac OS 中，对用户的键盘输入实现应用层的保护，通过 NPAPI 技术（Netscape Plugin Application Programming Interface）作为安全控件的用户态接口，避免被事件监听，同时对用户密码在内存中进行加密保护，密码密文传输时通过一次一密的机制避免回放攻击。

由于在 Mac OS 上安装内核驱动需要管理员权限，一般的恶意代码要实现远程安装，实施非常困难，因此应用层保护的安全控件就已经到达了极高的安全性。如在需要时，可通过加入内核驱动来实现更加底层的保护，与盗号木马进行对抗。

客户端集成方式：dmg 安装文件。

4.6.2 iOS 系统

背景分析：

iOS 系统没有正式提供对于 Safari 浏览器的插件开发接口，并且在未越狱的情况下，也没有提供任何其对应的安装方式。而且，iOS 本来也只是提供 Application 这样的应用软件的开发和安装。

考虑到 iOS 中所有应用程序都是用屏幕键盘进行输入的获取，所以关键在于防止系统软键盘的劫持和后台的录、截屏。目前需要防范的还是主要限于已经越狱的 iOS 上正常的客户端的输入截取问题，在未越狱的 iOS 设备要实现键盘记录基本不可能。

解决方案：

为了防止 iKeyGuard 这类输入监听（keylogger）程序，密码卫

士安全控件采用自定义屏幕键盘方式，每次使用的时候重新初始化键盘布局，并且提供“按键无状态模式”。所谓“按键无状态模式”，是当用户按下软键盘中的某个按键，并不绘制按下和释放的图形状态，只是提供声音作为按键反馈，避免屏幕被录、截屏。这样，可以避免对系统键盘的截取或者对屏幕进行截屏和录屏。另外，其控件内部数据采用加密方式存放，密钥随机生成。提交给应用服务器时，采用一次一密加密方式。

客户端集成方式：

采用静态库的方式进行接口集成与调用。

安全控件特点：

1）防止恶意程序拦截 virtual keypad 按键，如 iKeyGuard 等；

2）用户输入数据内存密文存储；

3）无按键状态键盘，防止屏幕截取；

4）支持键盘乱序，随机布局；

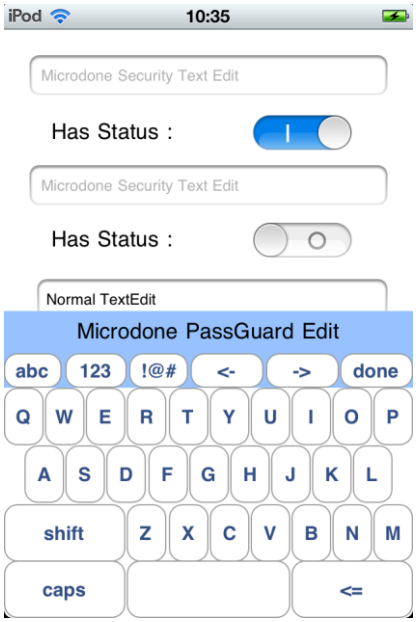
5）通过设置随机数，一次一密提交给服务器，防止回放攻击；

6）密码控件具备反调试、反注入、抗反汇编保护能力；

7）设备自适应，能自适应 iOS 设备尺寸，包括 iPhone，iPod Touch，iPad；

8）支持 Portrait 和 Landscape 键盘布局；

9）键盘 UI 支持客户化开发。



4.6.3 Android 系统

背景分析：

Android 系统与 iOS 类似，也是基于应用程序的开发框架，不提供浏览器的插件开发接口。

除了 iOS 的屏幕键盘获取问题，木马程序还可以通过 dispatchKeyEvent 来监听按键，分析 Android 系统保存键盘事件的 device 文件：mice,mouse0,event0，获取到用户的密码输入。

解决方案：

采用自定义屏幕软键盘，并提供键盘布局随机初始化、“按键无状态模式”以及随机加密存储。提交给应用服务器时，采用一次一密加密方式。

客户端集成方式：

采用 jar 包或 background service 的方式进行接口集成与调用。

功能特点：

1）防止 keylogger 等恶意程序截获用户输入；

2）用户输入数据内存加密；

3）通过无按键状态键盘，防止屏幕截取；通过修改窗口隐私属性，修改 framebuffer 和 getDrawingCache()函数属性实现保护；

4）键盘随机布局，符合监管要求；

5）防止进程注入；

6）防动静态反调试；

7）控件自身完整性保护；

8）移动端防录截屏保护；

9）检测屏幕记录驱动文件的属性，通过取消可读属性，到达保护屏幕按键坐标位置的目的；

10） 保证在不同分辨率设备上正常运作；

11） 键盘 UI 支持客户化开发。

4.7. 密码卫士的扩展应用－－－PIN 码保护

目前广泛使用的一代 UsbKey 在进行交易签名时，存在交易伪造以及交易篡改的安全隐患，主要的问题就是 UsbKey 的 PIN 码输入与传输过程中存在安全隐患，PIN 本身输入时未作防截取保护，PIN 码传输到 key 中验 PIN 时未进行加密保护。

采用密码卫士的跨平台安全控件，可完全解决上述问题。

监管要求：

《网上银行系统信息安全通用规范》（JRT 0068-2012）在安全技术规范中明确要求要加强客户端安全、专用安全设备安全、网络通信安全和网上服务器端安全等。

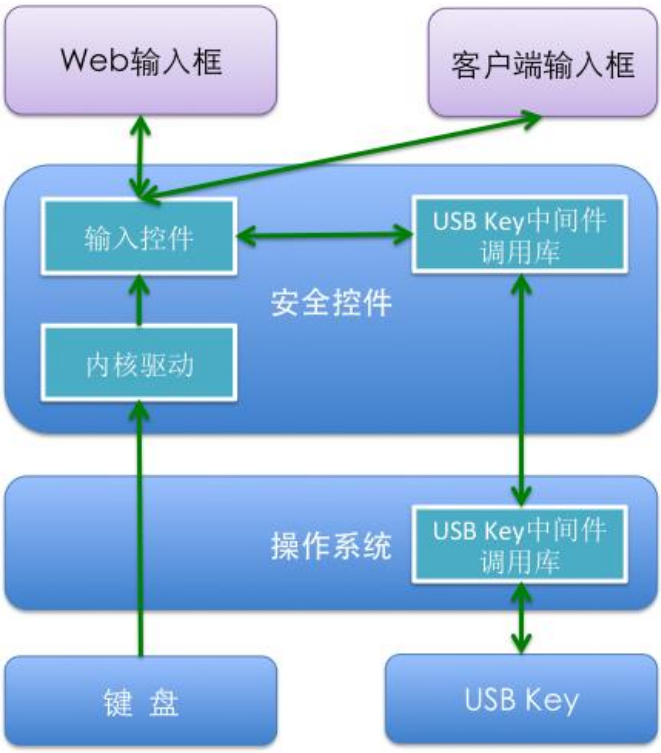
其中，在 6.1.2 专用安全设备安全—6.1.2.1USB Key 基本要求：

j）应保证 PIN 码和密钥的安全

采用安全的方式存储和访问 PIN 码、密钥等敏感信息；

PIN 码和密钥（除公钥外）不能以任何形式输出；

经客户端输入验证的 PIN 码在其传输到 USB Key 的过程中，应加密传输，并保证在传输过程中能够防范重放攻击。



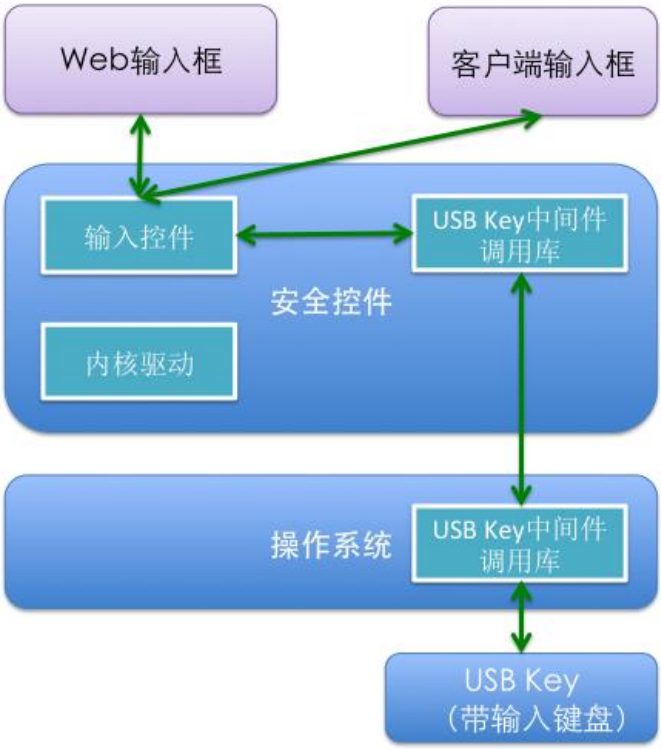
PIN 码保护示意图

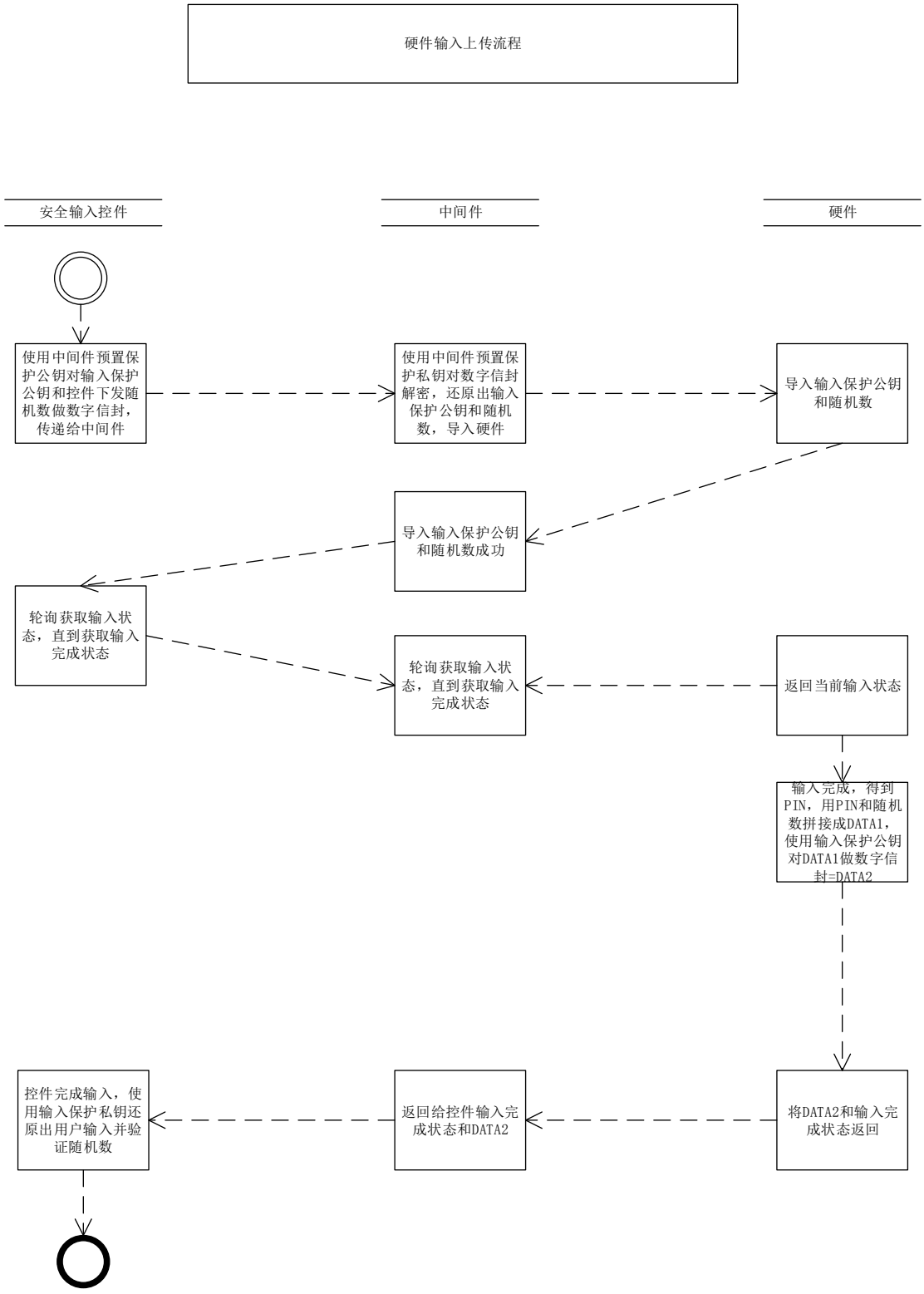
4.8. 密码卫士的扩展应用－－－与三代 key 集成实现输入保护解决方案

利用三代 key 提供的小键盘，输入敏感信息，将敏感信息通过安全加密机制传递给密码输入控件，由应用上传到服务端加密机解密和转加密；

密码等敏感信息输入不在 PC 或者移动终端上完成，在 key 上直接输入，传输过程中全密文。

安全控件判断是否用户当前使用了三代 key，如未使用，正常进行本机键盘输入保护，如使用了，则启用安全通讯传输机制。


Web输入框
客户端输入框
输入控件
USB Key中间件
调用库
安全控件
内核驱动
操作系统
USB Key中间件
调用库
USB Key
（带输入键盘）



需先商定中间件预置保护密钥对，私钥中间件持有，保证输入保护公钥只有中间件能解。此密钥对用途为防止中间人攻击。
中间件和硬件之间目前就是密文通讯，导入密钥采用安全报文方式，可能产生的攻击在于对中间件的实时内存攻击。

此方案可适用三代KEY。

4.9. 密码卫士的适用范围

密码卫士适用于各种网上在线系统，主要包括：

1）网上银行、网上基金、网上证券、网上期货、网上保险；

2）第三方网络支付系统；

3）电子商务领域 B2B、B2C、C2C；

4）网络游戏；

5）即时通讯；

6）SNS 社区网络；

7）电子政务；

以及各种需要加强保护用户帐号信息输入的应用系统。