

超级狗 - V2.4

软件保护和许可指南

免责声明

我们尽最大努力使本文档完善、准确且有用，但我们无法保证其完美无缺。一旦我们发现了错误或疏漏，或者它们引起了我们的注意，我们将会尽力在产品的后续版本中将其更正。对于因为本文档内容的不准确或疏漏造成的任何直接或间接损害或业务损失，赛孚耐（北京）信息技术有限公司不负责任。

2016 年 10 月

P/N 007-013515-001 修订 A

许可协议

感谢您购买并使用赛孚耐（北京）信息技术有限公司的软件保护产品。在使用本产品前，请仔细阅读下面的许可协议。

同意本许可协议的所有条款及此处包含的任何补充或特殊的许可条款是获得本产品许可的必要条件。如果您不同意此协议的所有条款，请在三天内将产品退还赛孚耐（北京）信息技术有限公司或其代理商。您对本软件的使用将表明您同意接受本协议中条款的约束。

授予使用许可权

您可以为了备份的目的而复制光盘中的软件，可以为了保护您的软件的目的而根据本产品的文档说明将我们提供的软件合并进您的程序中。

禁止使用

除按上述“授予使用许可权”条款被授权的行为外，不可以复制、修改、逆向工程、分解或重组该产品的全部或部分，不可向他人销售、租借、许可、转让、分发全部或部分本产品或本协议授予的权利。

有限担保

保证在自产品交给您之日起的 12 个月内，在正常使用情况下，产品不会出现实质性的质量问题和生产设计上的缺陷。自您购买产品之日起的 12 个月内，赛孚耐（北京）信息技术有限公司的全部责任和您能获得的全部补救措施为：可选择退货、更换或修理；购买产品之日起的 12 个月以后，您能获得的全部补救措施为：更换或修理。

除了上述对本产品的原始购买者所提供的有限保证之外，不向任何人作任何其它的保证。对赛孚耐（北京）信息技术有限公司的产品、性能或服务亦没有明示的或暗示的或其它任何形式的保证，包括但不限于商品的适销性和对特定用途的适用性。

任何情况下，无论如何引起及依据何种责任理论，均不承担任何因使用或不能使用本产品造成的损失责任，包括：由于使用本产品或本产品的功能失效导致丢失数据、损失利润及其它特别的、偶然的、附随的、继发的或间接的损失。

所有权

所有的产品，包括开发狗、超级狗、超级狗试用件、软件、文档、与本产品一并附送的其它材料及您制作的备份的所有权与版权均属于赛孚耐（北京）信息技术有限公司。

协议的终止

违反上述条款时，本协议的授权将自动终止。

“狗”是北京金天地软件发展有限公司的注册商标，并已授权赛孚耐（北京）信息技术有限公司使用。本文所涉及的其它产品和公司名称可能是各自相应所有者的商标。

赛孚耐（北京）信息技术有限公司

北京市朝阳区机场路丽都广场丽都商业大厦4层408室(100004)

总机：+86-10-59373000

免费销售咨询热线：800-810-0804

中文官网：<http://www.gemalto.com/china>

本地活动中心：<http://china.safenet-inc.com>

销售咨询

电话：+86-10-59373000

邮箱：marketing-china@gemalto.com

技术支持

电话：+86-10-59373000

邮箱：support@safenet-inc.com

版权所有 © 2016 SafeNet, Inc. 保留所有权利。

目录

免责声明.....	3
许可协议.....	4
授予使用许可权	4
禁止使用	4
有限担保	4
所有权.....	4
协议的终止	5
前言	11
产品清单.....	12
产品功能.....	12
本指南各部分	13
获得支持.....	14
培训	14
技术支持	14
1 第 1 部分 入门	15
2 超级狗介绍	16
超级狗简介	17
一次保护、多种分发.....	17
超级狗的软件保护	17
软件保护	17
知识产权保护	18
超级狗的业务流程	18
软件试用	18
软件租赁	18
购买产品部分功能.....	18
超级狗的组件	18
超级狗的硬件设备.....	18
超级狗的软件工具.....	19
超级狗 API 函数库.....	21
开发商代码和开发号.....	21
开发号	21
开发商代码	21
演示的开发号和开发商代码	22

3	第 2 部分 保护	23
4	超级狗可提供的软件保护	24
	超级狗的工作原理	25
	超级狗的基本要素	25
	保护程序	25
	AES 加密和解密	25
	安全通道	26
	V-Clock 虚拟时钟	26
	利用超级狗存储区	26
	阻止逆向工程	27
	超级狗的保护方法	27
	超级狗 Licensing API 保护	27
	外壳加密保护	28
5	超级狗 Licensing API 保护	29
	超级狗 Licensing API 概述	30
	实施 Licensing API 的必备条件	30
	导入开发商代码	30
	超级狗 Licensing API 下载目录	31
	超级狗 Licensing API 库文件命名	32
	学习超级狗 Licensing API 示例	32
	Licensing API 示例	32
	超级狗 Licensing API 的函数	32
	函数组	33
	实施超级狗 Licensing API 保护	34
	需求规划	34
	工作流程	35
	登录函数	35
	使用超级狗 API 保护安卓平台应用程序	36
	测试版本	36
	使用需求	36
	如何编译/使用示例程序	36
	在安卓平台的工程中使用 SuperDog API	37
	如何编译 Java JNI 库	38
	使用超级狗加密助手	39
	准备步骤	39
	加密关键数据	39
	验证程序是否已授权	39

6 超级狗外壳加密保护	41
外壳保护功能	42
基本保护流程	42
基本保护设置	43
特征	43
如何使用外壳加密工具	44
必备条件	44
运行图形界面外壳加密工具	44
运行命令行外壳加密工具	44
关于保护 .NET 程序集	45
.NET 考虑事项	45
方法级别的保护	46
关于保护安卓平台应用程序	47
测试版本	47
使用需求	47
操作步骤	48
如何加密数据文件	48
使用外壳加密工具	48
使用超级狗 DRM 打包工具	49
播放加密保护的 Flash 文件和 MP4 文件	50
操作步骤	50
使用代码植入功能保护应用程序	51
7 超级狗软件保护策略	52
软件保护策略概述	53
总的保护策略	53
实施超级狗 Licensing API 保护后再进行外壳保护	53
使用最新的软件保护工具	53
不断重新评估保护策略	53
防止可预测性，使用不断发展的策略	53
在查出有破解尝试时，变化行为	54
常见攻击类型及应对措施	54
修补可执行文件与动态链接库	54
修改保护狗的存储区	54
模拟保护狗	54
使用终端服务器与终端服务解决方案	55
克隆保护狗	55
时钟篡改	55
通用的保护策略	57
简单技巧	57

中级技巧	58
高级技巧	60
8 第 3 部分 许可	61
9 超级狗许可的实施和远程升级.....	62
准备产品计划	63
确定功能组件（许可特征）	63
定义软件产品	63
指定许可类型	64
与许可相关的文件	65
许可定义文件	65
升级定义文件	66
升级文件	66
请求文件	67
超级狗授权管理工具.....	67
许可定义	68
升级操作	69
生成升级文件	69
远程升级工具	70
工作流程	70
使用说明	71
远程升级界面定制工具	72
使用说明	72
10 第 4 部分 分发	73
11 超级狗分发	74
超级狗编程工具	75
安装说明	75
使用说明	75
日志查看工具	76
使用说明	76
分发超级狗软件	76
Windows	77
Linux	79
12 第 5 部分 附录	81
13 疑难解答	82
检查事项	83
问题与解决方法	83

14 超级狗 Licensing API 参考	84
函数概述.....	85
详细说明.....	86
dog_login()	86
dog_login_scope().....	88
dog_logout()	90
dog_encrypt()	91
dog_decrypt()	93
dog_read().....	95
dog_write()	97
dog_get_info()	99
dog_get_size().....	101
dog_get_time()	103
dog_datetime_to_dogtime()	104
dog_dogtime_to_datetime()	106
dog_get_sessioninfo()	108
dog_get_version()	110
状态代码.....	114
15 术语表	117
16 索引	119

前言

前言将简要介绍本产品和本指南各章节的内容，并提供相关支持信息。SafeNet 建议您先阅读这些信息。以下将分别介绍：

- 产品清单
- 产品功能
- 本指南各部分
- 获得支持

产品清单

产品组件	超级狗试用套件	超级狗开发套件
超级狗 Drive 试用件	√	
开发狗		√
超级狗		√
安装光盘		√
快速入门	√	√

产品功能

产品组件	说明
安装光盘	<p>安装光盘包含演示开发商代码（位于 VendorCodes 目录下），所有超级狗软件工具（位于 Tools 目录下），超级狗 API（位于 API 目录下），超级狗 API 示例应用程序（位于 Samples 目录下），开发狗运行环境（位于 MasterRuntime 目录下），开发商指南（位于 Docs 目录下）以及帮助文档。</p> <p>默认的超级狗安装目录为：... \Program Files\Gemalto\SuperDog\[version]</p>
超级狗试用件	超级狗试用件用于演示和评估本产品，其外壳上印有演示开发号 DEMOMA。
开发狗	开发狗由开发商使用，内含开发商代码，允许使用本产品中的各种软件工具。
超级狗	超级狗由最终用户使用，用于保护软件并提供软件的使用许可。
超级狗开发商向导	超级狗开发商向导可以将开发狗中的开发商代码导入 Windows 系统，并允许开发商从 SafeNet 下载定制的 Licensing API 函数库。
外壳加密工具	开发商使用外壳加密工具可以实现快捷、安全的自动化软件保护，可阻止软件被施行逆向工程，且无需调整源代码。
超级狗 DRM 打包工具	使用该工具可快速地对数据文件进行加密。在加密数据文件时，可选择使用不同的特征 ID 进行加密，或者选择起始数据不加密的百分比（例如：视频文件预留 10% 的起始数据用于免费用户的预览），或者将数据文件与特定的超级狗绑定。（例如：将数据文件与超级狗 ID 绑定后，只有对应 ID 的超级狗才能正常打开数据文件）。
授权管理工具	主要用于许可设计，以及许可升级设计，还可以编程超级狗、查看超级狗，以及生成超级狗软许可。
远程升级工具	远程升级工具根据需要定制后，由最终用户使用，可以对已部署的超级狗进行远程升级。

产品组件	说明
远程升级界面定制工具	用于定制远程升级工具。
超级狗编程工具	供开发商的生产部门批量生产，可以将许可定义文件中的许可特征以及数据文件写入超级狗。
日志查看工具	用于查看编程超级狗的历史记录。
超级狗加密助手	主要用于数据加密，以及生成随机加密数组。使用该工具可以简化加密应用程序的工作流程。
超级狗 Licensing API	实施超级狗 Licensing API 保护可将保护机制集成到软件的源代码中。
超级狗 License Generation API	使用超级狗 License Generation API 可以将编程超级狗的功能、生成许可升级文件 (V2C 文件) 的功能和生成超级狗软许可的功能集成到开发商自己的后端系统。
快速入门	快速入门说明了使用外壳加密工具保护 Windows 应用程序的步骤。
开发商指南	开发商指南是为开发商提供的本产品使用说明书。
帮助文档	超级狗软件工具的帮助系统，可在软件工具的使用过程中提供详细的说明。启动超级狗的软件工具后，按 F1 键，该软件工具的帮助文档将显示。

本指南各部分

本指南提供相关背景知识和其他信息，说明如何能最好地使用本产品并全面发挥其作用。本书分为以下五个部分：

- **第 1 部分 入门**

介绍主要概念、详细介绍本产品的组件，并关注超级狗如何针对您的要求提供全面的解决方案。

- **第 2 部分 保护**

这部分的主题是如何使用本产品实施软件保护，详细讲解了本产品的两种软件保护方法、以及软件保护策略。

- **第 3 部分 许可**

关注软件使用许可的实施以及远程升级。

- **第 4 部分 分发**

关注在分发阶段中需要分发给最终用户的软件。

- **第 5 部分 附录**

提供以下信息：

- 疑难解答，解释可能遇到的各种问题，并提供解决方案。

- Licensing API 函数库参考，提供各函数及其参数、结构声明和详细信息、以及返回代码的摘要和说明。
- 全面的术语表，简明扼要地解释超级狗的术语。

获得支持

SafeNet 拥有许多经销商，可随时随地为您提供相关支持。要查找离您最近的办事处或经销商的名称，请拨打销售咨询电话。

销售咨询电话：800-810-0804

培训

有关其他信息以及相关的培训，请与我公司顾问团队联系。我们的顾问可以在以下几个方面为您量身定制培训内容：

- 将超级狗集成到您的产品中
- 针对您的软件产品分析最佳的保护策略
- 帮助您实现保护和使用许可模式

技术支持

您可以拨打以下电话，与我们的技术支持团队联系。

技术支持电话：86-10-59373000

第 1 部分 入门

本部分包含如下内容：

- 第 1 章 [超级狗介绍](#) 第 16 页

详细介绍本产品，讲解主要概念，并关注如何针对您的要求提供全面的解决方案。

1

超级狗介绍

本章将详细介绍本产品，讲解主要概念，并关注如何针对市场要求提供全面的解决方案。本章包含如下内容：

- 超级狗简介
- 一次保护、多种分发
- 超级狗的软件保护
- 超级狗的业务流程
- 超级狗的组件
- 开发商代码和开发号

超级狗简介

超级狗是集软件保护、内容保护和身份认证于一身的加密狗。

本文档重点介绍超级狗的软件保护功能。关于内容保护功能，请参阅第 7 章 [超级狗文档保护插件](#)、[超级狗 Media Player 插件](#)、[超级狗 DRM 浏览器插件](#)，关于身份认证功能，请参阅安装目录下 AuthenticationSample 文件夹中的《[超级狗身份认证使用指南](#)》。

一次保护、多种分发

超级狗的核心概念是“一次保护、多种分发”，即软件保护过程和软件产品的业务流程完全分开、相互独立。实施一次软件保护过程后，可执行多次业务流程、分发多种产品。

软件工程师使用超级狗实施的软件保护过程，首先在软件源代码中嵌入超级狗 Licensing API 函数，完成代码的编译后，再进行超级狗外壳加密保护。

产品或业务经理执行的软件产品业务流程，具体来说，包括计划产品、定义产品、确定产品的功能组件（特征）、对功能组件指定使用的许可类型，超级狗支持四种许可类型：永久、执行次数、过期日期以及执行天数，最后进行软件分发和销售。

实施一次软件保护过程后，可以按照需要执行多次软件产品业务流程。超级狗优势、独特性和灵活性就是基于上述这种“一次保护、多种分发”的原则。

超级狗的软件保护

软件与知识产权保护是站在开发商的角度，通过实施自动化和定制的安全策略，保障软件与知识产权的安全。超级狗提供自动化及自定义的软件保护方法，可以阻止逆向工程、避免机密信息的泄露、全面实现软件保护及知识产权。以下的部分将分别阐述超级狗的软件和知识产权保护。

软件保护

软件保护是指通过加密的手段对软件实施特定的安全策略。在软件开发过程中，软件工程师可以在源代码中实施各种各样的安全措施。实施的安全措施越多、复杂度越高，软件保护的安全程度也就越高。

使用超级狗的软件保护，软件工程师首先实施 Licensing API 到要保护的软件源代码，完成编译后对软件实施进一步的外壳加密保护。

实施超级狗的软件保护后，被保护的软件在运行时与超级狗通讯，只有在收到超级狗发出的响应，并且验证该响应正确无误后，被保护软件才会正常运行。

知识产权保护

知识产权是软件开发的基础，然而知识产权的盗窃却极易发生，据有关报道，因知识产权盗窃导致的损失每年就高达几十亿美元。因此，软件开发过程中所用到的算法、专业知识和机密信息必须得到保护，只有这样才能确保开发商拥有独一无二和有竞争力的软件产品。

超级狗使用 LicenseOnChip 技术确保许可免受篡改，支持基于 AES 算法的加密和解密功能，可以对软件用到的重要数据进行保护，并使用 ECC 密钥交换技术，使得重要数据在传输过程中也得到完善的保护。另外，超级狗硬件还使用了专用芯片，可以更好地防御硬件攻击及硬件克隆。

超级狗的业务流程

产品或业务经理面临的挑战，首先，为迅速适应市场变化以及业务需求，必须实施灵活的产品策略，其次，必须使用安全的方法。超级狗能够协助产品或业务经理应对此类挑战，一方面，超级狗提供各种方法，保证软件安全，确保只有合法用户才能使用软件，另一方面，在软件产品仍处于开发过程中，产品或业务经理就可以对功能组件（特征）指定许可类型，在软件分发后，还可以对功能组件（特征）重新指定许可类型，按照需要进行多种分发。

超级狗的业务流程包括计划产品、确定功能组件（特征）、指定许可类型、最后进行分发和销售。这样的业务流程有助于开发商拥有灵活的产品策略，创造更多的销售来源。以下列举超级狗可以提供的产品策略之一二：

软件试用

确保在试用期或有限次数内，用户可以试用软件，过了试用期，用户无法继续试用。

软件租赁

让用户能够按照使用次数进行付费或者按照年度订购并使用软件。

购买产品部分功能

在发布复杂且昂贵的软件产品时，无法承担全部费用的用户能够以较低的价格购买软件产品的部分功能。

超级狗的组件

前言部分简要介绍了本产品的硬件设备、软件工具和 API 函数库，以下将详细介绍这些组件。

超级狗的硬件设备

超级狗的硬件设备包括：超级狗试用件、超级狗、和开发狗。它们都是 USB 接口硬件。

超级狗试用件

超级狗试用件用来演示本产品的主要功能。将超级狗试用件连接至计算机，允许运行下列软件工具：

- 外壳加密工具
- 超级狗 DRM 打包工具
- 超级狗授权管理工具
- 远程升级界面定制工具
- 远程升级工具
- Licensing API 示例
- 超级狗加密助手



注意：超级狗试用件仅用于演示及评估。请不要发布使用超级狗试用件保护的软件。

开发狗

开发狗包含开发商代码。开发商代码是本产品的重要概念，关于开发商代码的详细信息，请参阅本章第 21 页的“[开发商代码](#)”。下载为开发商定制的 API 库、编程超级狗以及对超级狗进行升级需要开发狗。请您妥善管理开发狗。



注意：

1. 您需要安装开发狗运行环境才能使用开发狗。
 2. 您仅需要一个开发狗就可以使用超级狗的所有软件工具。
-

超级狗

超级狗采用了最新的专为 SafeNet 定制的高性能、高安全性芯片，其他厂商不能采购该专用芯片。超级狗存储软件的使用许可以及数据文件。在使用被保护软件时，最终用户必须将特定的超级狗连接至计算机，被保护软件才能正常运行。超级狗可以从一台计算机移动至另一台计算机上使用，因此，被保护的软件可以安装在多台计算机上，但只有经过超级狗的连接和验证之后才能运行。

超级狗的软件工具

下面详细介绍超级狗开发套件中的软件工具。

超级狗开发商向导

该软件工具可以将开发狗中的开发商代码导入 Windows 系统并从 SafeNet 下载特定的超级狗 Licensing API 函数库。这保证了每个开发商下载的超级狗 Licensing API 函数库都是不同的。

外壳加密工具

外壳加密工具用于实现自动化软件保护，无需更改源代码，只需简单的步骤，被保护软件和超级狗之间就可以建立联系，从而实现下列保护：

- 阻止对被保护软件施行逆向工程。
- 只有当特定的超级狗连接至计算机时，被保护软件才能运行。

超级狗 DRM 打包工具

使用该工具可快速地对数据文件进行加密。在加密数据文件时，可选择使用不同的特征 ID 进行加密，或者选择起始数据不加密的百分比（例如：视频文件预留 10% 的起始数据用于免费用户的预览），或者将数据文件与特定的超级狗绑定。（例如：将数据文件与超级狗 ID 绑定后，只有对应 ID 的超级狗才能正常打开数据文件）。

超级狗授权管理工具

使用该工具可以定义软件产品、指定许可类型、定义数据文件、生成许可定义文件、进行许可升级设计、编程超级狗、生成许可升级文件、查看超级狗以及进行其他操作。

远程升级工具

远程升级工具需要经过定制后才能分发给最终用户使用。该软件工具用于收集已部署的超级狗中的许可文件和状态信息并生成请求文件。该工具还用于对已部署的超级狗应用许可升级文件，完成软件使用许可的升级。

远程升级界面定制工具

远程升级工具需要经过定制才能分发给最终用户使用。该软件工具用于定制远程升级工具。进行定制有两个目的：一是将开发商代码关联到远程升级工具，二是自定义远程升级工具的用户界面。

超级狗编程工具

供开发商的生产部门将许可定义文件中的产品、特征以及数据文件写入超级狗。

日志查看工具

使用日志查看工具可以查看编程超级狗的历史记录。

超级狗加密助手

该工具可以帮助用户更加方便快捷地保护应用程序。超级狗加密助手工具可以针对重要数据生成加密源文件，以及生成随机加密数组。

超级狗 API 函数库

超级狗 API 函数库包括 Licensing API 和 License Generation API。

超级狗 Licensing API

软件工程师可以在软件源代码中调用 Licensing API 函数，将保护机制集成至源代码中，使被保护程序在运行时查找超级狗，并且使用超级狗对重要数据进行加密或者解密。

不同开发商的 Licensing API 库是不同的，开发商需要从 SafeNet 下载经过定制的超级狗 Licensing API 库。

超级狗 License Generation API

软件工程师可以使用 License Generation API 提供的功能，把超级狗编程工具的功能以及授权管理工具中的生成超级狗软许可的功能和生成许可升级文件的功能集成到开发商自己的 ERP 或 CRM 系统。

开发商代码和开发号

在开发商购买超级狗开发套件后，SafeNet 将向开发商分配一个开发号。开发号存储在开发狗中。

开发号

开发号由多个字符组成，有数据类型的开发号，例如 37517，有字符类型的开发号，例如 W3FLY。为便于识别超级狗硬件设备所属批次，开发号将被印在开发狗及超级狗的外壳上。

开发商代码

开发商代码用于关联被保护程序和超级狗，开发商代码存储在开发狗中。

开发商代码文件夹

使用超级狗开发商向导可以将开发狗中的开发商代码导入开发商代码文件夹 (VendorCodes)。该文件夹位于以下目录：... \Program Files\Gemalto\SuperDog\[version]\VendorCodes

开发商代码文件的命名

开发商代码文件保存开发商代码，其命名格式为[开发号].hvc。例如，如果开发号为 W3FLY，则开发商代码文件名将是 **W3FLY.hvc**。

演示的开发号和开发商代码

开发商可以使用超级狗试用件对本产品进行评估。超级狗试用件对应的开发号为演示的开发号 (DEMOMA)，对应的开发商代码保存于 DEMOMA.hvc 文件，该文件位于超级狗安装光盘 Windows 目录下的开发商代码文件夹 (VendorCodes) 内。评估本产品时，演示开发商代码将保存于系统的开发商代码文件夹 (VendorCodes) 中。

第 2 部分 保护

本部分包含如下内容：

- 第 2 章 [超级狗可提供的软件保护](#) 第 24 页

本章概述基于超级狗的软件保护，包括基本要素、工作原理以及保护方法。

- 第 3 章 [超级狗 Licensing API 保护](#) 第 29 页

本章讲解 Licensing API 软件保护，包括该 API 概述、实施前提、如何实施，以及函数说明。

- 第 4 章 [超级狗外壳加密保护](#) 第 41 页

介绍超级狗外壳保护、外壳保护的前提条件、保护参数的设置、如何保护 Windows 程序、.NET 程序集以及安卓平台应用程序。

- 第 5 章 [超级狗软件保护策略](#) 第 52 页

介绍超级狗的软件保护策略，针对常见破坏软件保护措施的方法，阐述超级狗的应对措施。

2

超级狗可提供的软件保护

前面章节概述了基于超级狗的软件保护，本章将进一步介绍超级狗软件保护的基本要素、工作原理以及超级狗提供的两种软件保护方法。本章包含如下内容：

- 超级狗的工作原理
- 超级狗的基本要素
- 超级狗的保护方法

超级狗的工作原理

在软件源代码中调用超级狗 Licensing API 函数，使被保护软件在运行中访问超级狗内的特征或数据文件，将超级狗与被保护的软件绑定。只有当被保护软件能够访问超级狗，超级狗发出的响应被软件验证正确无误时，被保护软件才能正常工作。当被保护软件发现超级狗不存在或返回的信息不正确时，被保护软件应停止运行。

超级狗的基本要素

以下讲解基于超级狗的软件保护基本要素。

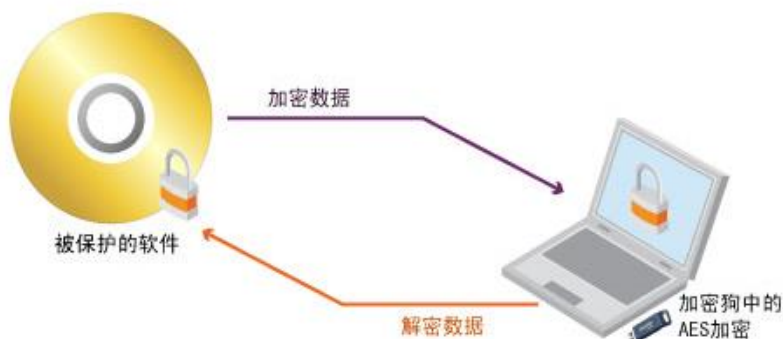
保护程序

超级狗可以保护多种应用程序，可被超级狗保护的内容如下：

- 从软件功能到整个程序
- 编译过的 Windows 可执行文件、DLL 和 .NET 程序集
- 软件用到的数据

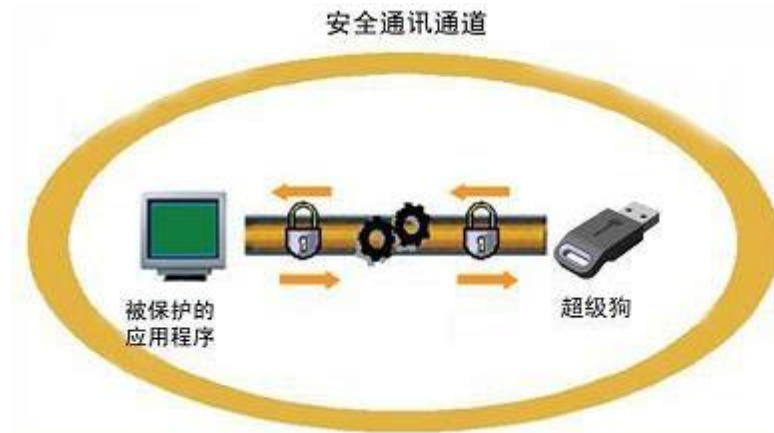
AES 加密和解密

超级狗支持基于 AES 算法的加密和解密功能。您的软件的保护方案不应只停留于是否能够登录超级狗，还应当使用超级狗对数据进行加密或解密，以验证超级狗。具体来说，可以先加密数据，将加密数据存储在程序中，在软件运行过程中，将加密的数据发送至超级狗进行解密，如果数据正确，才可视为超级狗“存在”。下图说明超级狗加密、解密的过程。



安全通道

被保护软件在运行中会与超级狗建立安全通讯通道，被保护软件和超级狗之间的通讯被加密，如下图所示。



V-Clock 虚拟时钟

超级狗支持虚拟时钟(V-Clock)可以检测时钟篡改，使得开发商可以安全可靠地分发具有时间限制的软件。有关超级狗虚拟时钟的更多内容，请参阅第 55 页的[时钟篡改](#)。

利用超级狗存储区

超级狗具有 1032 字节大小的数据存储区，可以用于存储特征和数据文件。每个超级狗都包含一个默认特征(特征 ID 为 0)和一个默认数据文件(数据文件 ID 为 65524)。开发商可以创建多个特征和数据文件。要想访问超级狗中的数据文件，必须登录到特征。“特征”是本产品的重要概念，关于特征的详细信息，请参阅第 43 页的[特征](#)。

超级狗中的数据文件有三种类型：只读类型，读写类型和写一次类型。使用 Licensing API 提供的函数 `dog_read()` 可以读取超级狗中的数据文件；使用 `dog_get_size()` 可以获取超级狗中数据文件的大小。

读写类型数据文件

使用 Licensing API 提供的 `dog_write()` 函数可以向超级狗中的读写或写一次类型数据文件写入数据。

只读类型数据文件

只读类型数据文件的数据在程序运行过程中不能被更改，除非通过超级狗的远程升级才能修改。

写一次类型数据文件

使用 `dog_write()` 函数对该类型数据文件进行一次写操作后，文件类型将变为只读。

许可类型

超级狗提供以下四种许可类型，有关更多信息，请参阅第 64 页的[许可类型](#)。

- 永久
- 执行次数
- 过期日期
- 执行天数

不同类型的许可文件占用的空间不同，永久类型的许可占用的空间最少。

阻止逆向工程

逆向工程是指试图解开软件保护方案并加以破坏，通常的行为是跟踪编译过的应用程序、修改其代码。使用超级狗外壳加密工具保护的程序可以阻止此类攻击。

超级狗的保护方法

下表描述超级狗提供的两种软件保护方法的各自特点。

外壳加密保护	超级狗 Licensing API 保护
<ul style="list-style-type: none"> • 快速、自动的保护为您的软件保驾护航 • 可以为程序定义特定的保护参数 • 不需要源代码 • 阻止逆向工程 	<ul style="list-style-type: none"> • 可以实施手动调用 Licensing API • 可以控制的提供最佳安全性的保护 • 保护力度与实施 Licensing API 的程度成正比 • 源代码必须可用 • 最佳的灵活性

超级狗 Licensing API 保护

在该保护方法中，软件工程师实施 Licensing API 到软件源代码，各种保护措施经过周密评估后被集成在源代码中。软件工程师控制整个保护过程，决定在源代码中哪些点调用 Licensing API 函数、被保护程序何时查询超级狗、以及在不同的情况下应当采取怎样的行动措施。

关于该保护方法更详细的内容，请参阅第 3 章 [超级狗 Licensing API 保护](#)。

外壳加密保护

外壳加密保护提供一种快捷、安全的自动化软件保护方法。外壳加密工具能够直接对可执行文件、DLL、.NET 程序集进行保护，不需要更改软件的源代码，并且可以阻止软件被施行逆向工程。

关于该外壳加密保护更详细的内容，请参阅第 4 章 [超级狗外壳加密保护](#)。

3

超级狗 Licensing API 保护

本章讲解超级狗 Licensing API 保护方法，包括该 API 概述、实施前提、如何实施、以及函数说明。本章包含如下内容：

- 超级狗 Licensing API 概述
- 实施 Licensing API 的必备条件
- 学习超级狗 Licensing API 示例
- 超级狗 Licensing API 的函数
- 实施超级狗 Licensing API 保护
- 使用超级狗 API 保护安卓平台应用程序
- 使用超级狗加密助手

超级狗 Licensing API 概述

软件工程师在实施超级狗的 Licensing API 保护时，可以在被保护软件源代码中插入对超级狗的调用。当被保护程序运行时，就可以检查程序是否有超级狗的保护。软件工程师可以对这些检查指定响应，例如，如果被保护程序未找到特定的超级狗，被保护程序就被指定自行挂起或终止。

在源代码中实施超级狗 Licensing API 保护是一个强健软件保护方法，然而要最大程度发挥其保护能力，取决于实施过程。调用该函数库的程度决定了被保护软件的整体安全水平。

软件工程师在实施超级狗 Licensing API 软件保护工程之前，最重要的是熟知 Licensing API 的函数。关于超级狗 Licensing API 的各函数的详细描述，请参阅第 84 页的附录 B [超级狗 Licensing API 参考或帮助文档](#)。

要了解超级狗 Licensing API 保护的工作方式和能力范围，软件工程师还应学习超级狗 Licensing API 示例。有关更多的信息，请参阅本章第 32 页的[学习超级狗 Licensing API 示例](#)。

实施 Licensing API 的必备条件

实施超级狗 Licensing API 的必备条件是：首先把开发商代码从开发狗导入 Windows 的开发商代码文件夹，其次是从 SafeNet 下载 Licensing API 函数库。有关开发商代码的信息，请参阅第 21 页的[开发商代码和开发号](#)。

导入开发商代码

使用超级狗开发商向导可以把开发狗中的开发商代码导入 Windows 与 Android 的开发商代码文件夹。以下是超级狗开发商向导的使用说明。

使用超级狗开发商向导

1. 将开发狗连接至计算机。
2. 运行超级狗开发商向导（**开始 > 程序 > 超级狗开发套件 V (版本) > 超级狗工具包**，选择：**软件保护 > 超级狗开发商向导**）。超级狗开发商向导随即启动，检测并列出的连接的开发狗。



注意：如果首次将开发狗连接到计算机，并且已启动超级狗应用程序，例如外壳加密工具，超级狗开发商向导会自动启动。

3. 超级狗开发商向导将提示导入 Licensing API 函数库。您有两个选项，可以选择“**下载**”或“**解压缩**”，默认的设置是“**下载**”。如果下载的速度慢，您可以选择解压缩的方式，那么您需要联系 SafeNet 技术支持部门，取得一份以开发号为文件名，扩展名为 mwp 的压缩文件，然后对其解压缩。
4. 在超级狗开发商向导的“**开发狗信息**”窗口中，选择路径并保存开发商代码。开发商代码应被保存在开发商代码文件夹(VendorCodes)中。

默认的开发商代码文件名为**[开发号名称].hvc**。开发商代码文件夹(VendorCodes)位于以下目录：

```
... \Program Files\Gemalto\SuperDog\[version]\VendorCodes
```

建议您使用开发商代码文件夹(VendorCodes), 因为在默认情况下, 超级狗软件工具将搜索该文件夹以获取开发商代码。

5. 在超级狗开发商向导的“**指定 API 设置**”窗口中, 选择下载您指定的 Licensing API 库。关于不同语言的函数库, 请参见本章[超级狗 Licensing API 下载目录](#)。
6. 如果出现升级 API 库的提示, 请进行升级。

超级狗 Licensing API 下载目录

下表列出 Licensing API 的下载目录:

目录	下载目录
C	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\API\Licensing\C
.NET	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\API\Licensing\DotNet
COM	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\API\Licensing\COM
Java	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\API\Licensing\Java
C	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\C
Java	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\Java
C#	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\csharp
VB.NET	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\vb.net
ASP.NET	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\asp.net
Delphi	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\Delphi
Javascript	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\Javascript
VB	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\vb
Authentication	...\Documents and Settings\[user_name]\My Documents\ Gemalto\SuperDog [version]\Samples\Licensing\Authentication

目录	下载目录
Android	...\\Documents and Settings\\[user_name]\\My Documents\\ Gemalto\\SuperDog [version]\\Android\\API\\Licensing\\C\\armeabi ...\\Documents and Settings\\[user_name]\\My Documents\\ Gemalto\\SuperDog [version]\\Android\\API\\Licensing\\C\\x86

超级狗 Licensing API 库文件命名

以下是超级狗 Licensing API 库的名称格式：

`dog_windows_[开发号].[库扩展名]`

例如，`dog_windows_demo.dll` 是一个与超级狗试用件关联的动态链接库，`dog_windows_37517.dll` 是一个与开发号为 37517 的超级狗关联的动态链接库。

学习超级狗 Licensing API 示例

通过学习超级狗 Licensing API 示例，软件工程师可以了解 Licensing API 软件保护的工作方式及其能力范围。

Licensing API 示例

超级狗提供针对主流编程语言的 Licensing API 示例。示例应用程序用于演示在源代码中实施超级狗 Licensing API 的软件保护。

超级狗安装目录下的 Samples 文件夹包含 Licensing API 和 Encryption 示例应用程序。Encryption 目录中包含了使用超级狗 API 进行数据保护的示例。

在演示这些示例应用程序时，必须将标有 DEMOMA 的超级狗试用件连接至计算机。



注意： 请参阅 [SafeNet/SuperDog 网站](#) 和超级狗安装光盘，获取有关特定编程语言的示例的相关信息。

超级狗 Licensing API 的函数

以下介绍超级狗 Licensing API 的几组函数，更详细的说明，请参阅附录 B [超级狗 Licensing API 参考](#)。

如何优化实施超级狗 Licensing API 保护，请参阅第 57 页的[通用的保护策略](#)。有关超级狗 Licensing API 的示例应用程序，请参见 [Licensing API 示例](#)。

函数组

以下分别介绍基于常见功能的五组超级狗 Licensing API 的函数。

- 会话函数
- 加密函数和解密函数
- 存储区函数
- 时间函数
- 管理函数

会话函数

调用 `dog_login()` 或 `dog_login_scope()` 对超级狗中特定一个特征进行一次成功的登录后，一个会话就创建了。在结束会话时，调用注销函数 `dog_logout()`，关闭该会话。

有关会话函数的更多信息，请参阅本章第 35 页的[登录选项](#)。

加密函数和解密函数

超级狗含有基于 AES 的加密引擎，可对数据进行加密或解密。

有关加密函数和解密函数的更多信息，请参阅：

- `dog_encrypt()`
- `dog_decrypt()`

读写函数

对重要信息的访问控制是保护计划中的一部分，利用超级狗存储区可以存储软件在运行时所需数据。

超级狗 Licensing API 提供以下读写函数。

- 使用 `dog_read()` 可以读取超级狗存储区中的数据文件。
- 使用 `dog_write()` 可以将数据写入超级狗存储区中的读写类型或写一次类型数据文件。

时间函数

时间函数组包括读取超级狗 V-Clock 时间、以及两个时间转换函数。

更多信息，请参阅：

- 使用 `dog_get_time()` 可以读取超级狗 V-Clock 时间
- 使用 `dog_datetime_to_dogtime()` 将日期时间值转换为超级狗时间
- 使用 `dog_dogtime_to_datetime()` 将超级狗时间值转换为日期时间值 (UTC)

管理函数

使用管理函数可以获取超级狗信息、获取会话信息、对超级狗进行升级。

更多信息，请参阅：

- 使用 `dog_get_info()` 获取超级狗信息
- 使用 `dog_get_sessioninfo()` 获取会话信息
- 使用 `dog_update()` 对超级狗进行升级

实施超级狗 Licensing API 保护

以下说明实施超级狗 Licensing API 保护前应该考虑的问题，讲解实施 Licensing API 的工作流程。

需求规划

在考虑实施超级狗 Licensing API 时，软件工程师应当考虑以下几个初步问题。

要保护什么？

这看似显而易见，在确定在源代码的哪些点调用 Licensing API 函数时，它却是关键。通常，在启动被保护软件时，您将要验证超级狗是否存在。您还可以使用不同的特征来保护软件的多个功能组件，这需要先确定要保护哪些功能组件。



注意：通过特征 ID 参数可以实现对不同特征(功能组件)的登录。关于更多信息，请参阅 `dog_login()`、`dog_login_scope()`。

实施计划中能否含有加密数据？

实施计划中可以含有加密数据。如果您打算在软件运行的时候使用加密数据，那么您可以先使用超级狗 Licensing API 的加密函数对数据进行加密，然后在软件中调用解密函数的地方插入已加密的数据。在软件运行时，加密的数据将由超级狗解密。

您可以使用超级狗加密助手快捷地完成数据加密的操作，然后参考该工具的示例工程来使用加密数据。

（工程默认路径为：`... \Program Files\Gemalto\SuperDog\[version]\Samples\Encryption\[language type]`）。

数据能否被存储在超级狗的存储区中？

超级狗数据存储区中的只读类型，读写类型和写一次类型数据文件都可以存储数据。您可以将应用程序的重要数据存储于只读类型的数据文件。在软件运行时，调用超级狗 Licensing API 提供的 `dog_read()` 函数读出存储的数据。

如何使用写一次类型数据文件？

您可以将某些只需要修改一次的信息存储于写一次类型数据文件中。例如：软件注册的用户名，与硬件绑定的相关信息等。在软件运行时，调用超级狗 Licensing API 提供的 `dog_read()` 函数读出存储的数据。

工作流程

以下是推荐的实施超级狗 Licensing API 保护的工作流程：

1. 学习与您开发环境相应的示例应用程序的代码。
2. 在您的应用程序源代码中，插入登录函数 `dog_login()` 或 `dog_login_scope()`，登录成功即建立会话。登录会话拥有其唯一的句柄标识符。



注意：会话的句柄标识符是自动生成的，应用于单个登录会话。如需更多信息，请参阅 `dog_login()`、`dog_login_scope()`。

3. 成功建立一个登录会话后，您可以调用超级狗 Licensing API 提供的其他函数，例如，使用解密函数 `dog_decrypt()` 对重要数据进行解密，也可以使用函数 `dog_read()` 读取储存在超级狗存储区中的数据等。
4. 使用第 3 步中生成的输出结果，检查潜在的不匹，针对不匹配对最终用户做出相应通知。
5. 对不同的功能组件重复第 2 至 4 步，完成对不同软件组件的保护。
6. 编译源代码。



注意：完成源代码的编译后，使用超级狗的外壳加密工具为软件再添加一层保护，可以防止破解者对被保护的代码执行逆向工程。

登录函数

调用登录函数 `dog_login()` 和 `dog_login_scope()` 是通往实施超级狗 Licensing API 保护的大门。要搜索超级狗并与超级狗通讯，就必须先建立登录会话。建立登录会话时，需要特征 ID 和有效的开发商代码。如登录未搜索到超级狗，系统就会显示错误消息，如发现声明的开发商代码与被搜索到的超级狗不匹配，也会显示错误信息。

登录选项

特征 ID 用于标识“特征”。为声明要登录到超级狗中的特定特征，调用登录函数时必须指定特征 ID 参数。

- 特征 ID

“特征”是本产品的重要概念，关于特征的详细信息，请参阅第 43 页的[特征](#)。

当登录到特定特征时，被保护软件不仅检查超级狗是否存在，还会检查超级狗中对该特征的许可，如果许可有效，才能使用该软件。当登录默认特征(特征 ID 为 0)时，API 仅搜索超级狗而忽略检查许可。



注意：每个登录会话都必须调用相应的注销函数来终止。更多信息，请参阅 `dog_login()`、`dog_login_scope()`、以及 `dog_logout()`。

使用超级狗 API 保护安卓平台应用程序

使用超级狗 Java API，可以保护安卓平台(ARM 架构或 x86 架构)开发的应用程序。想要知道如何保护安卓平台应用程序的更多信息，请参阅《*超级狗[版本]安卓开发商指南*》。

测试版本

已测试过安卓系统版本：

- Android 4.1.1 (ARM 架构)
- Android 4.2.2 (ARM 架构和 x86 架构)
- Android 4.3 (ARM 架构)
- Android 4.4.2 (ARM 架构和 x86 架构)
- Android 5.1.2 (ARM 架构)

使用需求

使用本软件包之前需要安装：

- Android 开发环境 ADT bundle
- Android NDK (如果您需要重新编译 Java JNI 库)

如何编译/使用示例程序

请按照如下步骤使用示例目录：

1. 打开 Android 开发环境 ADT 里的 Eclipse；
2. 导入工程 SuperDogDemo；
3. 编译工程；
4. 将生成的 apk 上传到安卓设备并安装应用。

如果您使用超级狗硬件：

5. 请使用 OTG 连接线将超级狗连接至安卓设备；
6. 打开应用程序 SuperDogDemo；

7. 安卓设备显示提示信息，“允许应用 SuperDogDemo 访问该 USB 设备吗？”，点击“确定”按钮；
8. 点击“Demo”按钮运行调用 API 的示例，将显示调用 API 的信息。

如果您使用超级狗软许可：

5. 打开应用程序 SuperDogDemo，点击“GetFP”按钮将获取安卓设备指纹并存储到 SD 卡根目录：**fingerprint.c2v**；
6. 复制 fingerprint.c2v 到 Windows 系统；
7. 运行授权管理工具，添加特征，请不要添加数据文件。点击“生成 SL”按钮：
 - a. 指定指纹文件 fingerprint.c2v
 - b. 指定 SL 存放目录
 - c. 按“确定”将生成超级狗 SL: **update.v2c**
8. 将 update.v2c 拷贝到安卓设备的 SD 卡上；
9. 在安卓设备上再次运行 SuperDogDemo 应用，点击“Update”按钮，从 SD 卡上读取 update.v2c 并安装 SL；
10. 安装 SL 成功后，点击“Demo”将演示对超级狗 SL 的操作。



注意：

- 超级狗 Java API 的包名称为 **com.SuperDog**。
- 当登录默认特征(特征 ID 为 0)时，API 仅搜索超级狗而忽略检查许可。

在安卓平台的工程中使用 SuperDog API

请将下列文件加入到您的工程：

1. SuperDogDemo\src\com\SuperDog\Dog.java
2. SuperDogDemo\src\com\SuperDog\DogApiVersion.java
3. SuperDogDemo\src\com\SuperDog\DogTime.java
4. SuperDogDemo\src\com\SuperDog\DogStatus.java
5. SuperDogDemo\src\com\SuperDog\DogUsbHandler.java
6. SuperDogDemo\libs\armeabi\libDogJava.so
7. SuperDogDemo\libs\armeabi\libdog_android_[开发号].so

如果您想同时支持 x86 架构，还需要将下列文件加入到您的工程：

8. SuperDogDemo\libs\x86\libDogJava.so
9. SuperDogDemo\libs\x86\libdog_android_x86_[开发号].so



注意：

- 使用前需要先用超级狗开发商向导下载您的 Android API，从目录...\Documents and Settings\[user_name]\My

Documents\Gemalto\SuperDog [version]\Android\API\Licensing\C\armeabi 中将 libdog_android_[开发号].so 拷贝至目录 SuperDogDemo/libs/armeabi 中。

- 从目录...\Documents and Settings\[user_name]\My Documents\Gemalto\SuperDog [version]\Android\API\Licensing\C\x86 中将 libdog_android_x86_[开发号].so 拷贝至目录 SuperDogDemo/libs/x86 中。
- 关于如何调用超级狗 Java API，请参考文件 SuperDogDemo\src\com\SuperDogDemo\SuperDogDemo.java。
- 请在您的应用中添加下列语句加载超级狗 Java JNI 库：
`System.loadLibrary("DogJava")`。
- 如果您使用超级狗硬件，请在您的应用中添加下列语句加载您的 Android API：
`System.loadLibrary("dog_android_[开发号]")(ARM 架构)`，
`System.loadLibrary("dog_android_x86_[开发号]")(x86 架构)`。
- 如果您使用超级狗硬件，在您的应用的运行过程中想要获取操作当前连接的超级狗的权限，可以调用 DogUsbHandler 中的 getPermission 方法。
- 在您的应用的 AndroidManifest.xml 文件中添加下列语句：`<uses-sdk android:minSdkVersion="14" />`。
- 如果您使用超级狗硬件，在您的应用的 AndroidManifest.xml 文件中添加下列语句：`<uses-feature android:name="android.hardware.usb.host" />`。

如何编译 Java JNI 库

编译 Java JNI 库之前需要安装并配置 Android NDK。

使用超级狗开发商向导下载您的 Android 的 API 库，从目录...\Documents and Settings\[user_name]\My Documents\Gemalto\SuperDog [version]\Android\API\Licensing\C\armeabi 中将 libdog_android_[开发号].a 拷贝至目录 SuperDogDemo/jni 中，从目录...\Documents and Settings\[user_name]\My Documents\Gemalto\SuperDog [version]\Android\API\Licensing\C\x86 中将 libdog_android_x86_[开发号].a 拷贝至目录 SuperDogDemo/jni 中。

修改文件：SuperDogDemo/jni/Android.mk 其中 API 库名称指向您的安卓 API 库。

返回上级目录 SuperDogDemo，运行 NDK 的命令 `ndk-build`。

编译好的 Java JNI 库在目录 SuperDogDemo\libs\armeabi 和 SuperDogDemo\libs\x86 中。

使用超级狗加密助手

超级狗加密助手用于简化超级狗 Licensing API 保护的步骤。开发商仅需将工具生成的源文件包含到自己的代码工程中，添加简单的调用代码就可实现超级狗 Licensing API 保护。

以下说明使用超级狗加密助手的步骤。如需更多的信息，请参阅超级狗加密助手的帮助文档。

准备步骤

1. 将开发狗和超级狗连接至计算机。
2. 运行超级狗加密助手，在菜单栏中选择 **工具 > 设置**，然后选择“使用正式开发号”。



注意：如果首次使用超级狗加密助手，需要先运行超级狗开发商向导以下载最新的 API 库。

加密关键数据

1. 在超级狗加密助手主界面选择**数据加密**选项卡。
2. 选择**编程语言**，**特征 ID**，以及加密的**数据类型**。
3. 如选择加密字符串或原始数据，请在主界面上边空白编辑框中输入字符，也可点击右边“**Hex 输入**”按钮切换到 Hex 输入模式。
4. 指定**数据长度**，并输入数据。
5. 要使用一个已存在的数据文件，单击“**装载**”按钮可以将其导入。
6. 文本输入模式下的文本将以 **UTF-8** 编码的形式进行加密，解密后的字符串需要从 **UTF-8** 编码格式转换为需要的编码格式，可参考示例目录（默认路径为：`...\Program Files\Gemalto\SuperDog\[version]\Samples\Encryption`）下的示例工程。
7. 点击“**加密数据**”按钮，生成包含加密数据的源代码。
8. 点击“**保存**”按钮，可保存生成的源代码为指定文件。
9. 将生成的源文件包含到自己的工程中，在源代码中使用超级狗 Licensing API 解密，然后使用解密后的数据进行各种运算。可点击打开示例文件夹链接参考示例工程（默认路径为：`...\Program Files\Gemalto\SuperDog\[version]\Samples\Encryption\[language type]`）。
10. 要加密数字类型数据，请选择加密类型为：**常量**。选择**编程语言**，**特征 ID**，**数据类型**，输入常量数据。然后重复步骤 7—9。

验证程序是否已授权

1. 在超级狗加密助手主界面选择**生成加密数组**选项卡。
2. 选择**编程语言**，**特征 ID**，**数据长度**和**数组长度**。
3. 点击“**生成加密数组**”按钮，随机生成包含明文与密文的数组。

4. 点击“保存”按钮，可保存生成的源代码为指定文件。
5. 将生成的源文件包含到自己的工程中，可在程序中定时（例如 2 分钟）随机的访问数组中一行数据的明文，然后使用超级狗 Licensing API 进行加密。将加密后的数据与数组中对应的密文数据进行比较，如果不相等，则判断用户使用了未授权的软件。可点击打开示例文件夹链接参考示例工程（默认路径为：...\Program Files\Gemalto\SuperDog\[version]\Samples\Encryption\[language type]）。



注意：超级狗出厂时内置默认的特征（特征 ID 为 0）。如果您选择了其他特征，需要确保该特征在超级狗中存在。请使用授权管理工具进行许可设计，并使用超级狗编程工具对超级狗编程。

4

超级狗外壳加密保护

本章介绍外壳保护功能、使用外壳保护的前提条件、保护参数的设置、如何外壳保护 Windows 程序、如何外壳保护 .NET 程序集以及如何外壳保护安卓平台应用程序。本章包含如下内容：

- 外壳保护功能
- 基本保护流程
- 基本保护设置
- 如何使用外壳加密工具
- 关于保护 .NET 程序集
- 关于保护安卓平台应用程序
- 如何加密数据文件
- 播放加密保护的 Flash 文件
- 使用代码植入功能保护应用程序

外壳保护功能

超级狗的外壳加密工具用于实施快速、简单的软件保护，无需像使用 Licensing API 函数那样对源代码进行改动，就可以自动给可执行应用程序加一个保护层。

外壳保护后，被保护程序在开始运行和运行当中会自动地访问超级狗是否存在。如果超级狗不存在，应用程序将不能运行。一旦被保护的软件无法访问超级狗，被保护的软件将提示错误信息。

外壳加密工具可以保护 Win32、Windows x64、以及 .NET 可执行文件 (EXE) 与动态库 (DLL)。外壳加密工具可以防止逆向工程和其他调试措施。



注意：本章将使用“程序”一词来泛指可用外壳加密工具保护的各种文件类型，无论这些文件是可执行文件(executables)、二进制文件(binaries)、程序集(assemblies)、还是库(libraries)。

超级狗的外壳加密工具有图形用户界面和命令行两种形式。使用图形用户界面的外壳加密工具，您可以实现以下操作：

- 保护 Win32、Windows x64 及 .NET 可执行文件与动态库
- 通过设置方法级别的保护增强对 .NET 程序集的保护
- 自定义被保护软件向最终用户显示的错误信息

使用支持命令行的外壳加密工具，您可以实现以下操作：

- 保护 Win32、Windows x64 及 .NET 可执行文件与动态库
- 轻松应用在图形用户界面中定义的保护参数。这可以简化在开发过程中对程序重复应用外壳加密的过程。

基本保护流程

本节介绍使用外壳加密工具保护程序的流程，以下说明使用外壳加密工具保护程序的步骤。如需更多的信息，请参阅外壳保护的帮助文档。

1. 启动图形用户界面的外壳加密工具。
2. 向项目添加要保护的 executable 文件、动态库。
3. 为被保护程序设置保护参数。
4. 保护程序。
5. 将被保护软件连同超级狗一起分发给最终用户。



注意：虽然外壳保护不影响被保护的 file，但是我们强烈建议为被保护的程序指定单独的输出文件夹，以便区分未保护的源文件和已被保护的输出文件。

外壳保护不改变软件源文件或被保护程序实际的工作方式。实施外壳保护后唯一的改变是：最终用户访问被保护程序的权限取决于所需特定的超级狗是否存在。如果该超级狗存在，那么被保护的软件就可以运行。

下图说明了外壳保护的逻辑关系。请注意，原始文件可以是：Win32、Windows x64、.NET可执行文件或动态库。



注意：为了确保软件的最高安全级别，对 Win32 程序的外壳保护将从正在保护的程序中删除调试数据。

基本保护设置

基本保护设置是使用外壳加密工具时必须指定的信息。

- 输入文件：要保护的程序文件。输入文件路径的默认位置是您向项目添加程序的位置。
- 输出文件：被保护的程序文件。输出文件路径的默认位置是：...\\Documents and Settings\\[user_name]\\My Documents\\Gemalto\\SuperDog [version]\\Tools\\Protected
- 特征 ID：当保护 Windows 程序或 .NET 程序集时，给程序指定的特征 ID。该特征 ID 的默认值为 0。每个超级狗都包含一个默认特征(特征 ID 为 0)。如果您指定的特征 ID 不为 0，请使用授权管理工具进行许可设计，确保许可定义包含该特征，并使用超级狗编程工具对超级狗编程。

特征

“特征”是指程序中可被识别的功能组件，可以是一个可执行文件、软件模块、.NET 方法或者打印、保存、绘制等特定功能。特征被分配一个唯一的标识号码，称作特征 ID。本节描述特征 ID 适用于所有被保护的 executable 文件和动态库。有关特征的更多信息，请参阅第 6 章 [超级狗许可的实施和远程升级的确定功能组件（许可特征）](#)。

使用特征保护 Windows 程序或 .NET 程序集

当外壳保护 Windows 程序 (Win32、Windows x64 可执行程序 and DLL)、ARX、安卓 APK 或 .NET 程序集 (.NET 可执行文件和 DLL) 时，您可以给整个程序指定一个特征 ID。如果您要为各组件指定特征，那么

必须使用超级狗 Licensing API。有关信息，请参阅第 3 章 [超级狗 Licensing API 保护中的需求规划](#) 和 [工作流程](#)。

如何使用外壳加密工具

本节介绍如何使用外壳加密工具。

必备条件

使用外壳加密工具之前，Windows 系统必须有以下组件：

- 待保护程序：Win32、Windows x64 或 .NET 可执行文件或动态库，ARX 或安卓 APK。
- 如果要保护 .NET 程序集，需要 .NET Framework 2.0 或更高版本。
- 超级狗试用件或开发狗已被连接至计算机。

运行图形界面外壳加密工具

在开始菜单中，选择 [程序](#) > [超级狗开发套件 V \(版本\)](#) > [超级狗工具包](#)，在超级狗工具包中选择：[外壳加密工具](#)。

启动外壳加密工具后，超级狗开发商向导会自动启动。有关超级狗开发商向导的更多信息，请参阅第 30 页的 [导入开发商代码](#)。

保护参数

将要保护的程序添加至保护项目之后，您完全可以使用默认的保护参数执行外壳保护，您也可以使用自定义的保护参数，这些保护参数将影响被保护程序的行为。

“[保护设置](#)”窗口中包括基本保护设置和保护选项卡。基本保护设置包括输入文件、输出文件、和特征 ID。保护选项卡用于查看以及修改保护参数。“[保护设置](#)”窗口有以下选项卡：

- “[常规](#)”选项卡(仅对于保护 .NET 程序集)
- “[保护设置](#)”选项卡
- “[高级](#)”选项卡
- “[数据文件保护](#)”选项卡

所有保护参数在外壳保护帮助文档中均有详细介绍。

运行命令行外壳加密工具

从 Windows 命令行启动外壳加密工具，这在运行不需要图形界面的自动化进程时非常有用。



注意：外壳加密保护的命令行版本主要用于自动化进程。运行命令行工具之前，使用 `envelope.exe` 创建并保存保护项目及其配置文件。

要访问外壳保护的命令行版本，请转至... \Program Files\Gemalto\SuperDog\[version]\Tools\envelope.com

在命令行中键入 `envelope.com` 即可启动命令行外壳加密工具。

命令行选项

以下参数用于外壳加密工具的命令行版本：

命令	说明
<code>-h /--help</code>	显示命令行参数列表。
<code>-p /--protect <project></code>	命令行工具使用所指定的项目作为封装进程的输入数据，该项目包括的所有文件都将被保护。
<code><project></code>	命令行工具将启动外壳加密工具(图形用户界面)并运行所指定的项目。
<code>-n /--no-warnings</code>	命令行不显示警告信息。
<code>-s /--searchPath <path></code>	设置搜索.NET 程序集的路径。

关于保护.NET 程序集

在保护.NET 程序集时，您可以对该程序集指定一个特征。要对单个方法设置保护参数，请参阅本章第 46 页的[方法级别的保护](#)。要对单个方法应用代码与符号混淆技术，请参阅本章第 46 页的[代码与符号混淆技术](#)。

.NET 考虑事项

保护.NET 程序集时，请考虑以下问题：

- 您必须在开发环境下保护.NET 程序集。外壳加密工具需要的库并不是.NET Framework 的组成部分，却是包含在开发环境中的。
- 外壳加密工具要求访问所有程序集及其依赖项。
- 外壳保护会破坏签名程序集的强名称签名，您可以选择重新对外壳保护的程序集进行强名称签名。
- 在保护.NET Framework 1.x 程序集时，由于外壳保护的输出文件基于.NET Framework 2.0，从而要求最终用户计算机上安装.NET Framework 2.0。
- 被保护的.NET 程序集在运行时需要使用两个超级狗 DLL 文件，请参阅第 77 页的[超级狗 DLL](#)。

- 在使用方法级别加壳保护.NET 程序时，如果选择一个许可类型为执行次数的特征 ID，程序每次启动时将消耗一次执行次数。另外，多个被保护的方法第一次调用时，每个方法将分别消耗一次执行次数。在程序未关闭前，被加壳保护的方法再次被调用时，不再消耗执行次数。
- 要想使用数据文件保护功能，请选择保护类型为：方法级别与 Windows 加密方式或仅 Windows 加密方式。

方法级别的保护

取决于被保护的.NET 程序集是可执行文件还是动态库，外壳加密工具自动设置最佳的保护类型。保护类型被自动设置后，程序中所有可保护的方法均被选中，相关保护参数均被设置。







注意：建议您不要更改被外壳加密工具自动设置的保护类型。

选择.NET 方法

“保护设置”窗口中的“可用于保护的方法”列表中列出要保护的.NET程序集，以树形目录结构显示类构造函数和方法，以代表方法类型的图标、类或方法的名称表示各个条目。当光标驻留于方法条目时，该方法签名将显示。

下表列出代表各方法类型的图标：

图标	方法类型	说明
	类构造函数	类构造函数图标代表该类的所有.ctor 方法。类中包括的实例、静态和入口点方法都嵌套在类构造函数下。 注： <ul style="list-style-type: none"> • 基础.ctor 方法未作为单独的方法显示在列表中。 • 如果某个类不包括.ctor 方法，则将显示类构造函数名称，但无法选择该项。这不会影响嵌套方法，其仍受到单独的保护。 • 为类构造函数定义的保护或设置不影响嵌套方法
	实例方法	不包括 static 修饰符的方法
	入口点	.NET 可执行文件的入口点
	静态方法	包括 static 修饰符的方法



注意：

- 勾选或清除更高级别条目的复选框，不会影响嵌套条目。例如，如果清除类构

造函数的复选框，那么其下嵌套的方法仍然保持选中。

- 当方法名称以灰色显示时，无法被选中用于保护。
 - 如果保护类型为“仅 Windows 加密方式”，则无法保护该.NET 程序集中单个的方法。
 - 如果列表中所有条目的复选框都已清除，则无法保护该程序集。
-

代码与符号混淆技术

使用混淆技术可以将有意义的字符串转变为随机字符串或数字。使用外壳加密工具时，您可以将应用混淆技术作为一种阻止逆向工程的安全措施。

因为实施代码混淆可能降低程序的执行速度，所以外壳加密工具的默认设置为不使用混淆技术。

如果使用混淆技术，默认的情况将是打乱被保护.NET 程序集中的所有的符号名称，包括程序集、类、方法的名称。此外，您可以选择打乱指定方法的全部代码。当选中一个方法后，不管有没有勾选其左侧的复选框，您都可以对它使用混淆技术。

关于保护安卓平台应用程序

使用安卓外壳加密工具可以对安卓平台 (ARM 架构或 x86 架构) 开发的 Java 应用程序进行外壳加密保护。该工具对 apk 文件进行保护并生成新的 apk 文件，新的 apk 文件包含加密过的代码以及一些其他的外壳加入的文件。

测试版本

已测试过安卓系统版本

- Android 4.1.1 (ARM 架构)
- Android 4.2.2 (ARM 架构和 x86 架构)
- Android 4.3 (ARM 架构)
- Android 4.4.2 (ARM 架构和 x86 架构)
- Android 5.1.2 (ARM 架构)

使用需求

请在 Windows XP/7 上使用此工具，使用本软件包之前需要安装：

- Android SDK
- JDK 1.6 或 JDK 1.7

操作步骤

1. 打开超级狗外壳加密工具，在菜单栏中选择：**工具 > 设置 > 安卓设置**，指定安装好的 JDK 所在目录和 Android aapt 工具所在目录，点击“确定”。
2. 在程序列表窗格中，双击一个 apk 程序文件，**保护设置**窗口随即显示。
3. 在保护设置窗口检查默认的保护设置或修改设置。

您可以在**安卓设置**选项卡中设置**保护类型**。如果选择 All，会加密保护 apk 程序中的 JAVA 文件和 SO 文件；Java Only，只加密保护其中的 JAVA 文件；Native Only，只加密保护其中的 SO 文件。

您还可以设置 apk 文件的**密钥文件**、**别名**、**口令**等信息，用于对外壳加密保护后的 apk 文件签名。您也可以自己对外壳加密保护后的 apk 文件进行签名，具体请参考：

(<http://developer.android.com/tools/publishing/app-signing.html>)

在安卓保护设置的高级选项卡中，LOCKING_TYPE（锁定类型）可以指定为：SuperDog，SL，SuperDog or SL。

WHOLE_DEX_PROTECTION 选项设置为 True 时，整个 DEX 文件将被加密，安全性将进一步提升。

4. 单击“**保护**”按钮，外壳加密工具将保护所选的程序。

如需更多的信息，请参阅《*超级狗外壳加密工具帮助*》文档中的“保护 Android 程序”与“安卓数据文件保护选项卡”与《*超级狗【版本】安卓开发商指南*》。



注意：

- 外壳加密工具的安卓平台应用程序保护设置中输出路径不支持中文。
- 安卓外壳加密保护不支持对安卓应用程序运行过程中产生的数据文件进行加密保护。
- 如果您保护了安卓应用程序中的 SO 文件，最终用户需要安装超级狗 SL 来运行加密保护的安卓应用程序。

如何加密数据文件

外壳加密工具支持加密数据文件。以下是加密数据文件的基本步骤，如需更多的信息，请参阅外壳加密工具的帮助文档。

使用外壳加密工具

操作步骤

1. 启动图形用户界面的**外壳加密工具**。
2. 向项目添加要保护的 executable 文件、动态库，为被保护程序设置保护参数。
3. 选择“**数据文件保护**”选项卡，勾选启用“**数据文件加密**”选项。必须勾选该选项才能使用数据文件保护功能。

4. 点击“**添加文件**”按钮或“**添加目录**”按钮，添加需要加密的数据文件或文件夹，并指定保护数据文件需要使用的**特征 ID**。
5. 点击“**高级设置**”按钮，将显示设置文件过滤器窗格。
在**设置文件过滤器**窗格中，可设置文件过滤条件。设置的过滤条件只针对受保护的应用程序在运行过程中产生的数据文件有效，当前选中的数据文件无论设置与否都将全部加密。您可以使用文件扩展名或者自定义的通配符作为文件过滤条件，便于对批量的同类数据文件进行加密或忽略。
6. 点击“**加密选择的文件**”按钮或“**加密全部文件**”按钮，将使用指定特征 ID 对主窗格中选定的数据文件进行加密。
7. 点击“**保护**”按钮，将对加载程序进行加密保护。

**注意：**

- 超级狗外壳加密工具 2.0 版本与更高版本保护的应用程序不支持打开超级狗外壳加密工具 1.1 版本加密的数据文件。您可以使用超级狗外壳加密工具 2.0 版本与更高版本重新加密您的数据文件，您还可以继续使用超级狗外壳加密工具 1.1 版本加密您的应用程序。
- 文件过滤器中设置的过滤条件只针对受保护的应用程序在运行过程中产生的数据文件有效，当前选中的数据文件无论设置与否都将全部加密。
- 想要设置更多数据文件加密的相关参数，请使用超级狗 DRM 打包工具。例如：对视频类数据文件，预留从文件头开始 10% 的数据不加密以使用户预览；将加密的数据文件绑定到特定的超级狗以提升保密级别等等。

使用超级狗 DRM 打包工具

操作步骤

1. 启动图形用户界面的**超级狗 DRM 打包工具**。
2. 点击“**添加文件**”按钮或“**添加目录**”按钮，添加需要加密的数据文件或文件夹，并设置相关选项和加密参数。

如果勾选了“**对 Media Player 插件支持的文件类型更改文件扩展名**”选项，在对支持类型的媒体文件进行加密保护后，被保护媒体文件的扩展名将添加"drm"后缀（例如：*.mp3→*.mp3drm）。此类媒体文件可以在安装超级狗 Media Player 插件后使用 Windows Media Player（支持版本：11, 12）查看。

超级狗 Media Player 插件安装包位于光盘路径：

Windows\Installed\DRM\Redistribute 中。

该插件支持下列文件类型：mpg, dat, avi, wav 和 mp3。

其他参数：

参数	说明
指定保护数据文件的特征	特征 ID 范围：0~65471。指定加密文件或加密目录中的文件时使用的特征 ID。

参数	说明
起始数据不加密比例	指定从文件头开始多少比例的数据内容不加密。
允许打印被保护的文档	指定被保护文档是否支持打印功能。
绑定特定的超级狗	如果选择了绑定特定的超级狗，运行过程中将检测特定的超级狗是否存在。超级狗 ID 为 0 表示不绑定。

3. 点击“加密选择的文件”或“加密全部文件”按钮，对数据文件进行加密保护。

要想查看加密后的数据文件，需要使用超级狗外壳加密工具加密的对应的加载程序。例如：要想查看使用超级狗 DRM 打包工具加密的 txt 文档，需要使用超级狗外壳加密工具来加密一个文本阅读器（例如：记事本）。详细操作步骤请查看超级狗外壳加密工具帮助文档。



注意：加密数据文件的加载程序时，

- 必须勾选启动数据文件加密选项（该选项位于外壳加密工具的数据文件保护选项卡中）。
- 加密加载程序与加密数据文件的特征 ID 必须保持一致。

播放加密保护的 Flash 文件和 MP4 文件

安装超级狗浏览器插件后，可以使用 IE 浏览器播放在线或本地的外壳加密保护后的 flv、swf 和 MP4 文件。如需更多的信息，请查看安装目录下的 DRM 目录中的《超级狗 Flash 文件和 MP4 文件保护方案说明》。

操作步骤

播放在线的 web 站点的加密 Flash 文件和 MP4 文件

1. 编辑您的网页用于播放 flv、swf 或 MP4 文件（播放 flv 文件需要准备一个 flv 网页播放器（通常是一个 swf 文件），可参考示例目录：Samples/flv/on-line；播放 swf 文件，可参考示例目录：Samples/swf/on-line；播放 MP4 文件，可参考示例目录：Samples/mp4/on-line）。
2. 将您的网页放到您的 Web 站点上。
3. 将超级狗和超级狗 DRM 浏览器插件安装包发给您的用户；
4. 您的用户在安装超级狗 DRM 浏览器插件并将超级狗连接到机器上后，他就可以用 IE 来浏览您的网站上的网页并播放外壳加密保护后的 flash 或 MP4 文件了。

播放本地的加密 Flash 文件和 MP4 文件

1. 编辑您的网页用于播放 flv、swf 或 MP4 文件（播放 flv 文件需要准备一个 flv 网页播放器（通常是一个 swf 文件），可参考示例目录：Samples/flv/local、Samples/swf/local 或 Samples/mp4/local。

2. 将超级狗，超级狗 DRM 浏览器插件安装包，加密的 Flash 文件或 MP4 文件以及相关的网页文件发给您的用户；
3. 您的用户在安装超级狗 DRM 浏览器插件并将超级狗连接到机器上后，就可以用 IE 直接打开、播放外壳加密保护的 swf 文件、flv 文件或 MP4 文件。



注意：在 IE 浏览器"保护模式"和"增强保护模式"均启用的情况下，不能播放被保护的 Flash 文件。

使用代码植入功能保护应用程序

使用代码植入功能，外壳工具可以将您选中的应用程序的函数代码放在超级狗中执行。被保护应用程序在运行过程中需要调用某个被保护的函数方法时，该函数被加密的代码片段将上传到超级狗中。在超级狗中，函数代码片段将会被解密并加载到虚拟机中执行。执行结果将会回传到应用程序中。函数方法在整个运行过程中没有在 PC 端执行，破解者将无法通过反汇编等手段来破解数据。从而显著地提升了使用超级狗保护的应用程序的安全性。

目前，代码植入功能有如下限制：

1. 仅支持 Windows 的 32 位的 exe 与 dll 文件
2. 不支持基于 .NET 程序集的应用程序

在保护过程中，外壳工具将使用代码转换引擎去分析被保护应用程序的代码。外壳工具使用应用程序对应的 map 文件分析应用程序中的函数，并检查哪些函数适合上传到超级狗中执行。这些合适的函数将在代码植入选项卡的程序列表窗格中列出。

除了可通过 map 文件生成可保护函数列表，如果应用程序包含导出函数，也可以使用导出函数表标识可保护的函数。您可以在勾选启用代码植入选项时，点击“搜索导出表”按钮来生成可保护函数的列表。

您需要查看列出的函数信息，然后选择合适的函数使用代码植入的方式进行加密保护。

当外壳工具保护应用程序时，外壳工具将把您选择的函数的代码片段提取出来，并使用一个占位符代替代码片段在应用程序中的位置。提取的代码片段将被加密并使用特定的开发商代码密钥进行签名，并被保存到分隔的二进制区域。

代码植入模块的处理过程是自动运行的，不需要您对您的工程源代码进行任何改动以适应这个处理过程。

被保护应用程序在运行过程中需要调用某个被保护的函数方法时，该函数被加密的代码片段将上传到超级狗中。在超级狗中，函数代码片段将会被解密并加载到一个虚拟机中执行。执行结果将会传递到代码片段在应用程序中对应的占位符（在被保护过程中写入）位置处。

通过以上处理，被保护函数方法在任何时候都始终没有暴露在外，杜绝了通过反汇编等手段来破解分析被保护的函数的可能性。

想要了解更多关于代码植入功能的信息，请参考超级狗外壳加密工具的帮助文档。

5

超级狗软件保护策略

前面的几个章节已详述超级狗可提供的软件保护，然而实施最优化的软件保护将取决于强大而完善的保护策略。本章将描述超级狗的软件保护策略、介绍软件破解者识别和破坏软件保护使用的方法、并推荐增强软件安全性的措施。本章包含如下内容：

- 软件保护策略概述
- 总的保护策略
- 常见攻击类型及应对措施
- 通用的保护策略

软件保护策略概述

随着越来越先进的软件安全措施不断地被开发出来，软件破解者也在开发更为复杂的破解方法，对软件代码执行逆向工程，复制分发非法软件、窃取知识产权。

为了限制非法访问软件的行为，阻止软件破解并维护开发商权益，开发商需要对破解策略保持高度警觉并机智地实施最先进和最强大的软件保护技术。虽然不存在无法破解的软件保护，但是，如果能够持之以恒地使用最强大的软件保护方法、实施最新的保护策略，就能显著减少易被攻击的安全漏洞。

对软件保护的投资程度取决于以下几种因素：

- 软件的价值
- 软件被破解的历史记录
- 分发软件的地理区域
- 软件的目标市场（例如，销售目标是个人消费用户、小型(或家庭)办公用户、还是企业用户）

SafeNet 多年以来坚持不懈地调查在软件安全领域中存在的潜在和实际的威胁，并在威胁构成危害之前，就不断对自身产品进行升级以应对此类威胁。除了本书介绍的信息以外，SafeNet 顾问团队还提供加强软件安全与保护方面的个性化帮助。如需有关的咨询服务信息，请联系您当地的 SafeNet 代表。

总的保护策略

无论实施何种的软件保护，您都应遵循以下总的保护策略。

实施超级狗 Licensing API 保护后再进行外壳保护

实施超级狗 Licensing API 保护并进行外壳保护可以实现最佳的安全性。

使用最新的软件保护工具

软件的升级通常包含对于最新威胁的改进措施。请选择并使用市面上最新版本的超级狗软件工具。您可以拨打我们的技术支持电话以便得到最新版本。

不断重新评估保护策略

经常考虑可以升级或增强的保护策略，以便为软件提供更强的安全性。

防止可预测性，使用不断发展的策略

在软件发行期间，不断变化所实施的策略。如果软件破解者能够检测到保护策略的模式，就可以解决或避开这些保护策略。

在查出有破解尝试时，变化行为

在破解尝试被查出后（例如本章稍后介绍的通过使用“校验和”查出破解尝试），您可以延迟软件的反应行为，从而可以破坏“原因”与“结果”之间的逻辑。延迟的反应可以使破解尝试和软件产生的负面反应之间的关系模糊化，借此可以迷惑软件破解者。处理破解尝试时，采取削弱程序功能之类的行为极其有效，这类行为包括使程序崩溃、覆盖数据文件，故意使程序出错，以及使程序变得不可靠等。

常见攻击类型及应对措施

“了解敌人”对于应对软件攻击非常重要。当开发商熟知破解者可能采取的攻击类型时，就可以轻松设计并实施防止攻击的最佳策略。以下是软件破解者经常使用的一些攻击类型以及超级狗的防御策略。

修补可执行文件与动态链接库

通过分解并调试 EXE 或 DLL 文件，软件破解者找到被保护代码，修补这些文件，改变运行流程或除掉代码中的调用，最后发布一个小的可执行文件补丁。

如果保护的文件越多，软件破解者解除保护所用的时间也就越长。使用超级狗的外壳加密工具可以安全、快捷地保护多个 EXE 文件和 DLL 文件。

修改保护狗的存储区

软件使用许可通常存储在保护狗的存储区中。软件破解者往往得以访问该存储区并修改条款，例如，基于执行次数的使用许可被更改为永久使用许可，或者无需付款就可能激活某项功能。

超级狗中的许可文件存储于安全存储区。检查许可是否有效在超级狗内部进行。要改写许可文件，只能通过超级狗的远程升级。超级狗远程升级有安全可靠的措施，所以软件破解者没有可乘之机。

模拟保护狗

软件破解者可以记录软件的保护狗调用，然后创建一个应用程序来重播被记录的保护狗调用，如同真正的保护狗在返回调用一样。功能有限的模拟器只能记录和重播调用。功能齐全的模拟器可以模拟保护狗的加密和解密操作，但软件破解者需要加密密钥才能创建功能齐全的模拟器。

被超级狗保护的软件在运行时，被保护的应用程序和超级狗之间的数据通讯在安全通道中进行。安全通道是基于 ASE 的加密技术。每次通讯均使用随机密钥，即使相同的调用也将产生不同的通讯数据。因此破解者即使记录了软件的超级狗调用也很难利用它进行重播。

超级狗的安全通道在建立时，使用了白盒密钥交换技术在软件和超级狗之间传递安全通道使用的基础密钥。白盒密钥交换技术是一种安全性很高的技术。使用白盒密钥交换技术，首先，基础密钥不直接传递给超级狗，而是传递密钥生成所需要的参数，超级狗根据这些参数使用特定的算法在超级狗内部生成安全密钥；其次，基础密钥参数使用了 ECC 密钥交换技术进行加密，只有拥有对应的 ECC 私钥才能解密获

得这些参数，而该私钥是在超级狗生产时写入超级狗并永远不会暴露到超级狗外。因此，破解者没有办法通过记录的超级狗调用来获得安全通道使用的基础密钥。

另外，超级狗的加密和解密操作均是在超级狗内部进行，破解者也没有办法获得加密密钥。因此，模拟超级狗是极其困难的。

使用终端服务器与终端服务解决方案

使用一些操作系统的终端服务器并将保护狗连接至终端服务器时，可能在多个同时运行的终端上使用同一软件。

超级狗软件保护对上述行为进行了控制。开发商使用超级狗授权管理工具时，当设置“允许远程桌面访问”时，允许上述行为，反之，不允许上述行为。

克隆保护狗

软件破解者先对保护狗实施逆向工程，然后“克隆”制造保护狗的副本。这样的做法，对于软件破解者而言，代价非常高昂，因为这既需要有逆向工程工具，也需要有专业技术。克隆保护狗的代价是巨大的。

超级狗采用了最新的专为 SafeNet 定制的高性能、高安全性芯片，其他厂商不能采购该专用芯片，可以更好地防御硬件攻击及硬件克隆。

时钟篡改

系统时钟篡改或回滚系统时钟是软件侵权的最常使用的方法之一。软件破解者通过回调计算机的系统时钟以达到延期使用的目的。

超级狗具有虚拟时钟(V-Clock)，无需实时时钟就可以检测时钟篡改。一旦计算机的系统时钟被回滚到24小时以前，被保护软件访问基于时间许可的特征将会出错(有关 API 函数调用将返回错误)，只有当计算机的系统时钟被纠正后，相关软件功能才能恢复正常的功能。

您无需担心由于夏令时引起的系统时钟调整造成程序不正常运行，超级狗已考虑到这种情况。如果发生夏令时时钟调整，程序仍然可以正常运行。

下表列出几种常见情景和行为供参考：

情景	行为
正常使用	<p>超级狗出厂时对虚拟时钟进行过初始化，时间被设置为超级狗的生产或编程时间。</p> <p>被保护软件运行时，Licensing API 会以系统时间为参考，不断更新超级狗虚拟时钟的时间。</p> <p>超级狗虚拟时钟的时间只能增加，或者通过对超级狗升级进行修改。</p> <p>如果您想改变超级狗虚拟时钟的时间，请生成许可升级文件，包含</p>

情景	行为
	<p>设置时间的升级项，对超级狗进行升级。</p> <p>您还可以在编程超级狗的时候根据当前系统时间重新设置超级狗虚拟时钟的时间。</p>
<p>如果计算机的系统时钟被回滚的范围在 24 小时以内</p> <p>（例如：超级狗虚拟时钟的时间为 2013 年 3 月 10 日 15 点 30 分，系统时钟被回滚到 2013 年 3 月 10 日 5 点 30 分）</p>	<p>正在运行的被保护软件可以继续正常运行；重新启动被保护软件时，该软件也可以正常运行。</p> <p>但是，超级狗虚拟时钟的时间不会被回滚到 2013 年 3 月 10 日 5 点 30 分，而是在 2013 年 3 月 10 日 15 点 30 分的基础上继续增加。</p>
<p>如果计算机的系统时钟被回滚到 24 小时以前</p> <p>（例如：超级狗虚拟时钟的时间为 2013 年 3 月 10 日 15 点 30 分，系统时钟被回滚到 2013 年 3 月 9 日 5 点 30 分）</p>	<p>正在运行的被保护软件可以继续正常运行。但是，超级狗虚拟时钟的时间不会被回滚到 2013 年 3 月 9 日 5 点 30 分，而是在 2013 年 3 月 10 日 15 点 30 分的基础上继续增加。</p> <p>重新启动被保护软件时，最多 5 分钟后，该软件访问基于时间许可的特征将会返回错误 45 (DOG_TIME_ERR)，包括如下 API：</p> <ul style="list-style-type: none"> • dog_login • dog_login_scope • dog_get_time <p>这时，该软件还可以访问不是基于时间许可的特征，但是 dog_get_time 会返回错误 45 (DOG_TIME_ERR)，并同时返回超级狗虚拟时钟的时间。</p> <p>此时如果将系统时间纠正到与超级狗虚拟时钟的时间的差距在 24 小时以内，则可以正常使用被保护的软件。</p> <p>如果您想改变超级狗虚拟时钟的时间，请生成许可升级文件，包含设置时间的升级项，对超级狗进行升级。</p>
<p>例如：超级狗虚拟时钟的时间和当前系统时间为 2013 年 3 月 10 日，系统时间被调整为 2013 年 4 月 10 日，被保护软件调用 dog_login 或 dog_login_scope 登录了基于时间许可的特征，或者调用 dog_get_time 读取了超级狗虚拟时钟的时间，超级狗虚拟时钟的时间将被更新为 2013 年 4 月 10 日，这时将系统时间调整到 2013 年 3 月 10 日。</p>	<p>重新启动被保护软件时，最多 5 分钟后，该软件访问基于时间许可的特征将会返回错误 45 (DOG_TIME_ERR)，包括如下 API：</p> <ul style="list-style-type: none"> • dog_login • dog_login_scope • dog_get_time <p>在这种情况下，如果您想改变超级狗虚拟时钟的时间，请生成许可升级文件，包含设置时间的升级项，对超级狗进行升级。</p>
<p>例如：超级狗虚拟时钟的时间为 2013 年 3 月 10 日，超级狗中某个特征的过期日期为 2014 年 3 月 10 日，最终用户收到超级狗硬件的时间是 2013 年 9 月 10 日，如果最终用户在首次使用之前将系统时间调整为 2013 年 3 月 11 日，最终用户则可以一直使用到 2014 年 3 月 10 日。</p>	<p>对于这种情况，建议您使用基于执行天数的许可类型。</p> <p>如果您一定要使用基于过期日期的许可类型，建议您在编程超级狗的时候根据当前系统时间重新设置超级狗虚拟时钟的时间。</p>

情景	行为
如果计算机主板电池没电了，可能导致系统时间不正确，与超级狗虚拟时钟的时间的差距在 24 小时以上。	<p>重新启动被保护软件时，最多 5 分钟后，该软件访问基于时间许可的特征将会返回错误 45 (DOG_TIME_ERR)，包括如下 API：</p> <ul style="list-style-type: none"> • dog_login • dog_login_scope • dog_get_time <p>在更换主板电池以后，被保护软件可以正常运行。</p>



注意：如果被保护软件使用的是超级狗 1.0 Licensing API，如果计算机的系统时钟被回滚到 24 小时以前，只有在系统重新启动或者超级狗运行环境重新启动后才能检测到时钟篡改。检测到时钟篡改后，超级狗虚拟时钟的时间不会被回滚，而是在原有时间的基础上继续增加。被保护软件调用 dog_login 或 dog_login_scope 登录基于时间许可的特征，或者调用 dog_get_time 读取超级狗虚拟时钟的时间，将会返回错误 45 (DOG_TIME_ERR)。

通用的保护策略

简单技巧

1. 赋值和判断分开

参考强度值：1

在赋值和对值的判断之间，可以插入很多其他的操作，让解密者难以猜到二者之间的联系。对于全局变量，可以在其他函数块中做判断。

2. 多次判断值

参考强度值：2

对于一个值，可以在不同的地方做多次判断，增加跟踪软件保护的难度，而只要有一个地方判断结果不对就可以认为整个结果是不对的。

3. 多次赋值

参考强度值：3

对于一个返回值，可以对它多次赋值，并使用复杂循环，让解密者无法跟踪该值的变化。

4. 赋多个值

参考强度值：3

返回的结果可以赋给很多的值，让解密者难以跟踪所有的这些返回值。而任意一个返回值不正确，都可以认为结果不正确。

5. 使用返回值 参考强度值：4

其实，对返回值做判断是一种不好的方法，容易被识破和屏蔽。直接使用返回值可以更好地让应用软件和加密软件结合。

6. 使用多个变换结果 参考强度值：10-20

变换的结果可以看成是一种可预知的返回值，随着输入的不同，输出也会不同，进行多次变换，并且使用变换的结果。

7. 关键加密点不要使用系统的比较函数进行比较 参考强度值：10-20

不要简单地使用字符串和硬件狗中的数据进行比较，更不要使用系统的比较函数，最好自己编写 `strncmp`、`memcmp`、`radom`、`getsystemtime`、`messagebox` 等函数，因为黑客很容易使用静态反编译工具识别出此类函数。

8. 尽量在加密点附近不要调用一些敏感的系统函数 参考强度值：10-20

例如：`random`、`getsystemtime`、`messagebox`、`exitprocess` 等函数。

9. 在超级狗中存放敏感数据 参考强度值：20

将敏感数据放在超级狗中，要使用时从超级狗中读出。您还可以调用加密函数和解密函数，使被保护程序使用的数据被加密后存放在超级狗中。能否解密取决于超级狗以及特定的特征是否同时存在，这样，被保护程序和超级狗之间的链接不会被轻易除掉。

中级技巧

10. 随机执行 参考强度值：10

解密者往往是通过大工作量地跟踪程序以发现加密点的，这需要不断地运行程序，以发现规律。增加程序执行的随机性可以迷惑解密者。实现随机性的方式可以是随机对返回值进行判断，随机地赋值，或程序随机地执行。

一些解密的高手可能会注意到程序执行的随机性，从而屏蔽 `Random` 函数。下面几条技巧也用到随机性，如果可能，请使用自己编写的随机数发生器。随机数发生器的初始值可以取系统时间等随机值。

11. 随机读 参考强度值：15

可以随机地从狗中读取一些内容，但是除非需要，这些内容可以不使用。地址、长度都可以是随机的。

12. 随机写

参考强度值：20

可以随机地向狗中写入一些随机内容，地址、长度都可以是随机的。但是要注意不要覆盖狗中有用的内容。较好的办法是在狗中保留一个特定数据文件区域用于随机写，并保证随机写不会越界。

13. 随机写然后读出

参考强度值：35

这其实是上面两个技巧的配对使用。要注意的是读和写要从程序的运行逻辑上分离远一些，中间可以加入其他对狗的操作，这样解密者极难分析两者的联系。

14. 随机变换

参考强度值：35

随机变换是指做变换的内容和长度可以是随机的。而返回的变换结果可以不使用。

15. 随机变换并使用结果

参考强度值：20—100

使用方法是这样的：假如我们要使用常数 1，先产生一个随机内容 A，把 A 保存起来，按照技巧 14 用 A 做一次随机变换得到变换结果 B，再把 B 同我们要得到的常数做一次运算，这里假设是加法，得到结果 C，把 C 保存起来，此时 $C = B + 1$ 。在程序的另一处，我们用 A 再做一次变换，变换的结果与上次的 B 相同，同保存的 C 做一次反运算，这里是减法，会得到结果 $D = C - B = 1$ 。这时，D 就是我们要用的常量。要注意，这个技巧基于这样的原理：即使变换的结果是不可预知的，但是两次相同变换的结果肯定是一样的。

16. 使用狗传递全局变量

参考强度值：40

可以把一些变量放在狗中，使用狗传递变量。这些变量是全局的，甚至下一次运行程序时还在。

17. 随机加解密

参考强度值：40

可以使用超级狗提供的加解密函数对随机的数据进行加解密和比较。

18. 定时查狗

参考强度值：40

可以在被加密的程序中根据日期来判断是否需要查询狗。比如说，在 2003-12-26 调用某个加密接口函数，然后在 2004-1-26 以后又调用某个加密接口函数，或者每天的某个时间调用加密接口函数。

19. 加密点分散原则

参考强度值：40

如果开发商的软件包括很多的模块，比如说 exe、dll、ocx 等子程序，建议不要仅仅在主程序中使用加密函数或者外壳工具，最好其它的模块也使用超级狗进行加密。

高级技巧

20. 使用“校验和”确认可执行文件的完整性

参考强度值：20—150

将被保护软件可执行文件的校验和与存储于超级狗存储区中的校验和进行比较，如果发现两个值不相等，有可能已有修改这些文件的尝试。

21. 多线程

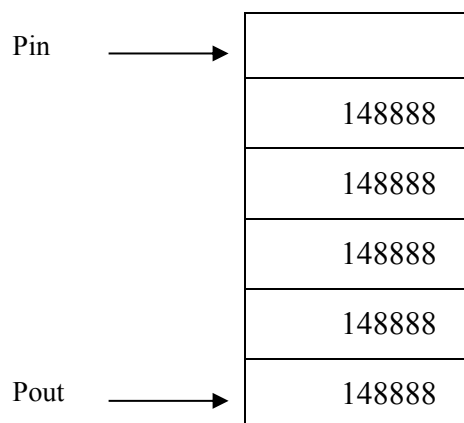
参考强度值：20—150

利用多线程技术可以很好地迷惑解密者和解密工具。基本思路是这样的：在一个线程中对狗操作，在另一个线程中判断和使用。

22. 队列

参考强度值：40—60

可以建造一个队列以管理返回值和返回结果，管理的方法可以复杂一些，而程序的执行是依赖于这个队列的。当然不用队列而使用其他的数据结构也是一样的，甚至效果更好。一个简单的使用队列进行返回值管理的例子如下图所示：



在上图中，有一个初始化了的队列，共有 6 个单元，初始数据为 148888，当发生的返回值不为 0 时，将 205429 赋给 Pin，并使队列移动一个单元。在 Pout 处检测被推出的数据，当为 148888 时不做动作，当为 205429 时即认为非法。

在程序中综合使用以上各种加密方法可以很好地保护您的软件。参考强度值仅供参考，与使用者的具体使用方法有很大关系。事实上也并不一定要追求高强度值的技巧，低强度参考值的技巧只要使用巧妙，而且多次使用，也可以达到很好的效果。高级的技巧一般难于实现，使用和调试都比较困难。开发商同时要注意，对超级狗操作是要耗时的，延时可能在 50 - 200 毫秒之间，与所读写字节的长度有关。

第 3 部分 许可

本部分包含如下内容：

- 第 6 章 超级狗许可的实施和远程升级 第 62 页

本章关注软件使用许可的实施，以及远程升级，介绍与许可相关的文件，以及超级狗授权管理工具、远程升级工具、远程升级界面定制工具的使用。

6

超级狗许可的实施和远程升级

本章关注软件使用许可的实施以及对许可的远程升级，首先介绍实施许可的前期准备工作——准备产品计划，然后引入与许可相关的文件，最后介绍如何使用超级狗授权管理工具、远程升级工具、远程升级界面定制工具。本章包含如下内容：

- 准备产品计划
- 与许可相关的文件
- 超级狗授权管理工具
- 远程升级工具
- 远程升级界面定制工具

准备产品计划

本指南在第一章中提到的超级狗的业务流程，产品或业务经理需要准备产品计划。产品计划的准备包括以下方面：

1. 确定功能组件
分析所有应用程序，确定可被指定许可类型的功能组件。
2. 定义软件产品
把各功能组件(许可特征)组合成可销售的软件产品。
3. 指定许可类型
分别对各功能组件(许可特征)指定许可类型。

以下对上述三个步骤分别举例说明。

确定功能组件（许可特征）

大多数的应用程序都可以被分割成多个独特的功能组件。这些功能组件可以被单独指定使用许可（许可类型）。通过分析现有的应用程序就可以确定这些功能组件。

在超级狗用“许可特征”代表这些功能组件。在产品范围内，为每个许可特征指定一个唯一的特征 ID。“许可特征 ID”可以关联整个应用程序、某个模块、或者诸如打印、保存之类的特定的功能组件。

示例

场景：质高软件公司的经营范围是提供建筑行业使用的建筑设计软件。质高公司的产品或业务经理分析该公司现有的应用软件后，确定了以下各功能组件/许可特征。

功能组件	许可特征
绘制设计	绘图
查看设计	查看
保存项目	保存
打印设计	打印设计
打印预定义的报表	打印报表

定义软件产品

列出所有许可特征后，按照业务需要，产品或业务经理可以将各许可特征组合成可销售的产品。

示例

场景：质高软件公司的产品或业务经理确定要销售以下产品：

- 面向小公司客户的高Q Design Lite 产品:包括绘图、查看、保存、以及打印设计这些特征。
- 面向大客户的HQ Design Pro 产品:包括所有许可特征。

下表说明这两个产品及其包含的许可特征：

产品：	HQ Design Lite	HQ Design Pro
许可特征		
绘图	有	有
查看	有	有
保存	有	有
打印设计	有	有
打印报表		有

指定许可类型

通过给许可特征指定许可类型可以控制软件的使用。即使不同产品中的相同许可特征也可以被指定不同的使用许可类型。

许可类型

超级狗提供了以下四种许可类型。

- 永久：可以永久使用，没有时间和次数的限制。
- 执行次数：限制可以使用的最多次数。
- 过期日期：在过期日期到达之前可以使用。
- 执行天数：从首次使用开始的一定天数内可以使用。

示例

场景：质高软件公司决定对上述两个产品包含的特征指定以下许可类型：

- HQ Design Lite 产品：低价年租适用于绘图和保存，无限制使用打印设计和查看。
- HQ Design Pro 产品：以较高价格可无限制使用全部功能。

下表说明这两个产品及其包含的许可特征：

产品:	HQ Design Lite	HQ Design Pro
许可特征	许可类型	许可类型
绘图	有效期为一年	永久
查看	永久	永久
保存	有效期为一年	永久
打印设计	永久	永久
打印报表		永久

至此，质高软件公司准备好他们的产品计划，下一步，他们将为上述产品准备许可定义文件。

与许可相关的文件

以下列出与“许可”相关的文件：

- 许可定义文件
- 升级定义文件
- 升级文件(V2C 文件)
- 请求文件(C2V 文件)

许可定义文件和升级定义文件均以 XML 格式描述，文件的扩展名为 xml。

升级文件的扩展名为 v2c，请求文件的扩展名为 c2v。这两种文件为加密的文件，无法被直接识别，以便被用于安全远程传输。

请求文件由最终用户生成，其他三种文件均由开发商生成。

许可定义文件

许可定义文件用于初次实施软件的使用许可。一份许可定义文件可以包括一个或多个许可特征。许可定义文件中的许可特征以及数据文件被写入超级狗，超级狗和被保护软件一同被分发至最终用户。

许可定义包括特征名称、特征 ID、各特征的许可类型，以及数据文件。许可定义(license definition)以 XML 格式描述。

生成方法

使用以下方法可以生成许可定义文件：

- 使用超级狗授权管理工具

“许可定义”选项卡用于生成许可定义文件。关于详细的使用说明步骤，请参阅第 68 页的[许可定义](#)，或者帮助文档。

应用方法

将许可定义文件应用于超级狗，或者说将许可定义文件中的许可特征以及数据文件写入超级狗，称为“编程超级狗”。

使用超级狗授权管理工具或超级狗编程工具都可以编程超级狗。超级狗编程工具适用于批量编程超级狗。关于超级狗编程工具的使用，请参阅第 71 页的[使用说明](#)，或者帮助文档中的《[超级狗授权管理工具帮助](#)》。

升级定义文件

升级定义文件以 XML 格式描述，文件扩展名为 xml。升级定义文件包括对超级狗及其内部信息的修改，例如，修改超级狗 V-Clock 时间、添加或删除许可特征、修改许可类型等。

升级定义文件和请求文件 (C2V 文件) 一起用于生成升级文件 (V2C 文件)。

生成方法

使用以下方法可以生成升级定义文件：

- 使用超级狗授权管理工具

“升级操作”选项卡被用于生成升级定义文件。关于详细的使用说明步骤，请参阅第 69 页的[升级操作](#)，或者帮助文档。

升级文件

升级文件是加密文件，无法被直接识别，其文件扩展名为 v2c，表示文件由开发商发送到最终用户，因此又称为 V2C 文件。升级文件包含加密的许可升级信息，将被应用于已部署的超级狗。开发商使用升级定义文件和请求文件可以生成 V2C 文件。

生成方法

生成 V2C 文件有两个必备条件：一是具有请求文件，二是具有升级定义文件或许可定义文件。

生成 V2C 文件有以下两种方法：

- 方法一：使用超级狗授权管理工具

“生成升级文件”选项卡被用于生成 V2C 文件。关于详细的使用说明步骤，请参阅第 69 页的[生成升级文件](#)，或者帮助文档。

- 方法二：使用 License Generation API

软件工程师可以使用超级狗 License Generation API 函数编程生成 V2C 文件。

有关 License Generation API 函数的详细说明，请参阅超级狗 License Generation API 帮助文档。

有关 License Generation API 示例应用程序，请参阅超级狗安装目录下的 Samples\Licgen 示例文件夹。

应用方法

最终用户使用经过定制的远程升级工具，可以把升级文件 (V2C 文件) 应用到已部署的超级狗。关于详细的说明步骤，请参阅第 72 页的[应用升级文件](#)。

请将定制的远程升级工具软件和超级狗一同分发给最终用户使用。

请求文件

请求文件包含在最终用户端已部署的超级狗状态信息。请求文件是加密文件，无法被直接识别，文件的扩展名为 c2v，表示由最终用户发送到开发商，因此又称为 C2V 文件。

生成方法

最终用户使用经过定制的远程升级工具，生成请求文件 (C2V 文件)。有关详细说明，请参阅第 71 页的[收集超级狗信息](#)。请将定制的远程升级工具软件和超级狗一同分发给最终用户使用。

超级狗授权管理工具

超级狗授权管理工具主要用于设计和生成软件使用许可、进行升级设计、导入请求文件、生成升级文件。

下表简要介绍超级狗授权管理工具的图形用户界面中主要的三个选项卡，稍后分别详细介绍。

选项卡	生成的文件
许可定义	许可定义文件和 SuperDog SL (超级狗软许可)
升级操作	升级定义文件
生成升级文件	升级文件

除了生成这些文件，超级狗授权管理工具还可以完成一些其他任务，例如，编程超级狗、查看超级狗、查看请求文件 (C2V 文件) 等。关于更多信息，请参阅帮助文档。

关于准备产品计划，请参阅本章第 63 页的[准备产品计划](#)。

以下分别详细说明超级狗授权管理工具中的三个选项卡。

许可定义

“许可定义”选项卡用于完成初次的许可定义和 SuperDog SL（超级狗软许可）。许可定义以 XML 格式描述。如果选择加密许可定义，您可以在 **工具 > 设置** 选项中勾选“**以加密形式存储许可定义文件**”选项。


“许可定义”选项卡生成的 SuperDog SL 则应用于安卓平台应用程序，无需超级狗硬件。


以下说明生成一份许可定义文件的详细步骤。如果您已有许可定义文件，也可以直接导入该文件然后再进行编辑。

生成许可定义文件步骤：

1. 将超级狗试用件或开发狗连接至计算机。
2. 在开始菜单中，选择**程序 > 超级狗开发套件 V (版本) > 超级狗工具包**，在超级狗工具包中选择：**授权管理工具**，启动超级狗授权管理工具。
3. 在“**许可定义**”选项卡中添加一个许可特征，单击“**许可授权**”窗格中的“**+许可特征**”按钮，“**添加特征**”窗口将显示。
4. 在“**特征&许可信息**”窗格中，填写**特征 ID**和**特征名称**，指定该特征的**许可类型**，并指定是否允许远程桌面访问该特征，然后单击“**确定**”。
 - 关于“许可类型”选项，请参阅第 64 页的**许可类型**。
 - 关于“允许远程桌面访问”选项，请参阅第 55 页的**使用终端服务器与终端服务解决方案**。

要添加更多的特征，回到**步骤 3**。

要编辑某个许可特征，选定该条目后单击 。

要删除某个许可特征，选定该条目后单击 。

5. 要添加一份数据文件，单击“**数据文件**”窗格中的“**+数据文件**”按钮，“**添加数据文件**”窗口将显示。
6. 在“**添加数据文件**”窗口中，指定该数据文件的**类型**、**文件 ID**、**文件长度**，并填入数据，最后单击“**确定**”按钮。

要使用一个已存在的数据文件（文本文件），单击“**装载**”按钮可以将其导入。

要添加更多数据文件，**步骤 5**。

7. 单击文件菜单栏下的“**保存为许可定义文件**”菜单项，许可定义文件将被保存至设定路径下。



注意：

- 授权管理工具生成超级狗 SL 不支持添加数据文件，只支持使用默认数据文件。
- 超级狗 SL 没有许可特征个数限制。

升级操作

使用“升级操作”选项卡，您可以设计升级操作，这将用于生成升级定义文件。关于更多信息，请参阅帮助文档。

生成升级定义文件的步骤：

1. 将超级狗试用件或开发狗连接至计算机。
2. 在开始菜单中，选择**程序 > 超级狗开发套件 V (版本) > 超级狗工具包**，在超级狗工具包中选择：**授权管理工具**，启动超级狗授权管理工具。
3. 选中“**升级操作**”选项卡，您可以参照已有的许可定义填写升级信息。

从以下来源可以导入许可定义文件：

- 从许可定义页面导入
- 从许可定义文件导入
- 从请求文件（C2V）导入
- 从连接的超级狗导入

导入的许可定义中的许可特征将显示在下方。

您可以修改许可特征、数据文件以及超级狗的设置。修改操作将被记录并显示于下方的升级列表。



注意：超级狗的虚拟时钟只能设置一次。

4. 单击文件菜单栏下的“**保存为升级定义文件**”菜单项，将保存为升级定义文件。

生成升级文件

生成升级文件 (V2C 文件) 有两个必要条件：一是收到从最终用户发来请求文件 (C2V 文件)，二是具有升级定义文件，或者许可定义文件。

设置好所需路径和选项后，您可以一次生成单个或多个升级文件。从以下来源可以导入升级列表：

- 从升级操作页面导入
- 从升级定义文件导入

设置路径：

- 请求文件
- 存储升级文件的文件夹

设置选项：

升级类型：指升级对超级狗存储区的操作方式。

- 升级：直接升级。
- 格式化后升级：清除超级狗中的许可特征和数据文件，并清空默认数据文件的内容。
- 清空特征后升级：清除超级狗中的许可特征但保留数据文件，也不会清空默认数据文件的内容。

需要回执：当勾选此选项时，最终用户完成升级后将生成“回执 C2V 文件”并发送给开发商。

可读升级文件：该选项默认值为勾选，生成的升级文件（V2C 文件）中将包含一段可被识别的关于许可升级的信息，但许可升级信息仍然为加密，不可被直接识别。

生成升级文件(V2C 文件)步骤：

1. 将超级狗试用件或开发狗连接至计算机。
2. 在开始菜单中，选择程序 > 超级狗开发套件 V (版本) > 超级狗工具包，在超级狗工具包中选择：授权管理工具，启动超级狗授权管理工具。
3. 选中“生成升级文件”选项卡。
4. 导入升级列表并设置上述路径和选项。
5. 单击“生成升级文件”按钮。

远程升级工具

在软件和超级狗被分发至最终用户后，开发商可以对已部署的超级狗进行远程升级。开发商无需召回超级狗，只需将升级文件（V2C 文件）发送给最终用户。远程升级是本产品核心概念的一部分。有关本产品的核心概念，请参阅第 17 页的[超级狗简介](#)。

超级狗简介

超级狗是集软件保护、内容保护和身份认证于一身的加密狗。

本文档重点介绍超级狗的软件保护功能。关于内容保护功能，请参阅第 7 章 [超级狗文档保护插件](#)、[超级狗 Media Player 插件](#)、[超级狗 DRM 浏览器插件](#)，关于身份认证功能，请参阅安装目录下 AuthenticationSample 文件夹中的《[超级狗身份认证使用指南](#)》。

一次保护、多种分发超级狗的远程升级工具 (Remote Update System) 是一个可执行文件 (该软件工具的文件名为 rus.exe)。远程升级工具必须经过定制才能分发给最终用户，否则最终用户将无法使用。使用远程升级界面定制工具 (rusbranding.exe) 可以使开发号和远程升级工具 (rus.exe) 相关联，并且还可以设置面向最终用户的界面显示信息。关于定制的更多信息，请参阅第 72 页的[远程升级界面定制工具](#)。



注意：使用超级狗 Licensing API 中的 **dog_update()** 函数可以对已部署的超级狗进行升级。如需更多信息，请参阅超级狗安装目录下的帮助文件：
...\\Program Files\\Gemalto\\SuperDog\\[version]\\API\\Licensing\\licensing_api_zh.chm

工作流程

在对最终用户分发软件时，开发商可以在软件的安装包中附带一个经过定制的远程升级工具(可执行文件)及其使用说明。

当最终用户需要升级超级狗时，可以使用定制的远程升级工具收集超级狗的信息，生成加密的请求文件(C2V 文件)。最终用户再发送请求文件给开发商。

请求文件的内容包括被保护的软件的当前许可，以及超级狗的状态信息，但不包含最终用户的个人信息。

开发商收到请求文件后，结合许可升级定义文件，生成加密的升级文件(V2C 文件)，发送给最终用户。

最终用户收到 V2C 文件后，再次使用定制的远程升级工具，将 V2C 文件应用于超级狗，从而完成远程升级。

示例

某设计公司购买了质高软件公司 20 套 HQ Design Lite 软件产品。该公司要把其中 5 套升级为 HQ Design Pro。

该设计公司使用定制的远程升级工具生成 5 份请求文件(C2V 文件)，通过电子邮件发送给质高软件公司。

收到这 5 份 C2V 文件后，质高软件公司使用超级狗授权管理工具，结合许可升级定义文件，生成 5 份 V2C 文件，最后通过电子邮件将这些 V2C 文件发送给该设计公司。

该设计公司收到这 5 份 V2C 文件后，再次使用定制的远程升级工具，将 5 份 V2C 文件应用于那 5 只超级狗，从而完成远程升级。

使用说明

远程升级工具经过定制以后，最终用户可以使用它收集超级狗信息并应用升级文件到超级狗。最终用户在使用定制的远程升级工具时，必须将超级狗连接至计算机。

收集超级狗信息

以下是最终用户收集超级狗信息的操作步骤。收集到的超级狗信息将被存储于请求文件(C2V 文件)中。

1. 将超级狗连接至计算机。
2. 启动定制的**远程升级工具**。
3. 单击“**生成升级请求文件**”选项卡。



注意：在连接多个超级狗的情况下，“选择超级狗”界面将显示，此界面会列出与计算机连接的所有超级狗，由最终用户选择要生成 C2V 文件的超级狗。

4. 单击“生成升级请求文件”按钮。
5. 指定保存文件的路径，输入文件名，单击“保存”按钮，将生成请求文件(C2V 文件)。

应用升级文件

收到开发商发来的升级文件(V2C 文件)后，最终用户使用定制的远程升级工具，将升级文件(V2C 文件)应用于超级狗，从而完成远程升级，以下是操作步骤。

1. 将要被升级的超级狗连接至计算机。
2. 启动定制的远程升级工具。
3. 选中“应用许可升级文件”选项卡。
4. 浏览并导入升级文件(V2C 文件)到“许可升级文件”字段。
5. 单击“应用许可升级文件”按钮。

远程升级界面定制工具

使用远程升级界面定制工具可以定制给最终用户使用的远程升级工具。开发商必须定制远程升级工具，否则最终用户将无法使用。

“定制”有以下两个目的：

- 使远程升级工具与开发号相关联

如果不关联开发号，最终用户将不能使用远程升级工具。

- 设置面向最终用户的界面显示信息

您可以显示例如公司名称、软件名称等信息。

关于远程升级界面定制工具的更多信息，请参阅帮助文档。

使用说明

使用远程升级界面定制工具的步骤如下：

1. 将开发狗连接至计算机。
2. 在开始菜单中，选择程序 > 超级狗开发套件 V(版本) > 超级狗工具包，在超级狗工具包中选择：软件保护 > 远程升级界面定制工具，远程升级界面定制工具将启动。
3. 在 HTML 文本框填写信息，点击格式按钮，编辑成所希望样式。
4. 单击“生成 RUS”按钮，为定制的远程升级工具可执行文件命名、选择路径并保存。

第 4 部分 分发

本部分包含如下内容：

- 第 7 章 [超级狗分发](#) 第 74 页

介绍在分发阶段中使用的超级狗编程工具、日志查看工具、以及开发商需要分发给最终用户的超级狗软件。

7

超级狗分发

本章介绍分发阶段使用的超级狗编程工具、日志查看工具、以及开发商需要分发给最终用户的超级狗软件。

本章包含如下内容：

- 超级狗编程工具
- 日志查看工具
- 分发超级狗软件

超级狗编程工具

您的生产部门进行批量生产时，使用超级狗编程工具把许可定义文件中的产品、特征以及数据文件写入多个超级狗。被编程的超级狗将和被保护的软件一同分发给最终用户。编程超级狗的操作被记录到日志文件。

安装说明

以下说明在您的生产部门部署超级狗编程工具的步骤：

1. 在您的开发环境中安装超级狗开发套件
2. 使用超级狗开发商向导下载针对您的开发号的 API
3. 将下列文件发送给您的生产部门：

超级狗编程工具所在的目录下的所有文件，位于... \Program Files\Gemalto\SuperDog\[version]\Tools\makekeys\
--

开发狗运行环境安装包，位于... \Program Files\Gemalto\SuperDog\[version]\MasterRuntime\

包含许可设计的许可定义文件

4. 在您的生产部门安装开发狗运行环境

使用说明

以下是使用超级狗编程工具的步骤：

1. 将开发狗连接至计算机。
2. 在开始菜单中，选择**程序 > 超级狗开发套件 V(版本) > 超级狗工具包**，在超级狗工具包中选择：**软件保护 > 超级狗编程工具**，超级狗编程工具将启动。
3. 单击“**装载许可定义文件**”按钮，浏览并导入许可定义文件。
4. 连接一个或多个超级狗至计算机，单击“**开始**”按钮，将开始编程。请不要在编程过程中移除或连接超级狗。编程的完成程度将显示于下方的进度条中。如果编程失败，超级狗的 LED 会一直闪烁，以引起您的注意。并且询问您是否继续编程其它超级狗。

编程超级狗操作完成后，许可定义文件中的产品、特征以及数据文件被写入连接的超级狗中，编程总次数、编程成功次数、编程失败次数、编程已用时间的统计信息显示于右侧的统计窗格第一列中，编程硬件总数、编程硬件成功次数、编程硬件失败次数的统计信息显示于右侧的统计窗格第二列中。单击“**重置统计信息**”按钮，统计数据显示复零。



注意：

- 多次编程同一个超级狗，编程硬件次数只记为一次。如果多次编程同一个

超级狗最后一次失败，编程硬件成功次数减一，编程硬件失败次数加一。

- 系统时间将会编程到连接的超级狗中作为初始时间。请确保编程时当前系统时间基本准确。
-

日志查看工具

日志文件是加密的文本文件。日志文件中的记录有三种：使用超级狗编程工具编程超级狗的记录、使用超级狗授权管理工具编程超级狗的记录，以及生成许可升级文件（V2C 文件）的记录。

使用日志查看工具可以查看日志文件。查看日志文件时需要将开发狗连接至计算机，以便对日志文件进行解密。

为便于管理和使用，当日志文件达到 3M 大小时，系统将自动创建一份新的日志文件。

关于此软件工具的更多信息，请参阅帮助文档。

使用说明

以下是使用日志查看工具的步骤：

1. 将开发狗连接至计算机。
2. 在开始菜单中，**选择程序 > 超级狗开发套件 V(版本) > 超级狗工具包**，在超级狗工具包中选择：**软件保护 > 日志查看工具**，日志查看工具将启动。
3. 打开一份日志文件，日志文件中的编程记录将列表显示。
4. 双击一条记录将弹出窗口显示该记录的“**详细信息**”，编程超级狗时用到的许可定义将显示于“**许可定义**”字段。

分发超级狗软件

开发商需要分发给最终用户以下的超级狗软件：

Windows：

- 定制的远程升级工具
- 超级狗 DLL(适用于外壳保护.NET 程序集)
- 超级狗文档保护插件
- 超级狗 Media Player 插件
- 超级狗 DRM 浏览器插件

Linux：

- 超级狗运行环境

Windows

定制的远程升级工具

最终用户使用定制的远程升级工具，可以收集超级狗中的信息，生成请求文件，还可以将许可升级文件应用于超级狗，完成对许可的远程升级。

有关远程升级工具的更多信息，请参阅第 70 页的[远程升级工具](#)。

关于如何定制远程升级工具，请参阅第 72 页的[远程升级界面定制工具](#)。

超级狗 DLL

使用外壳工具的代码植入功能保护应用程序时，开发商必须将下表中的超级狗 DLL 与被保护的应用程序一起分发给最终用户。最终用户只有在安装超级狗 DLL 后，被保护的应用程序才能与超级狗通讯。

被保护应用程序的类型	最终用户操作系统	超级狗 DLL 文件
应用程序	32 位 Windows	snvm.dll dog_windows_[开发号].dll
	64 位 Windows	snvm.dll dog_windows_[开发号].dll

使用外壳保护 .NET 程序集时，开发商必须将下表中的超级狗 DLL 与被保护的 .NET 程序集一起分发给最终用户。最终用户只有在安装超级狗 DLL 后，被保护的 .NET 程序集才能与超级狗通讯。

被保护应用程序的类型	最终用户操作系统	超级狗 DLL 文件
.NET 程序集	32 位 Windows	dogdnert.dll dog_windows_[开发号].dll
	64 位 Windows	dogdnert.dll dogdnert_x64.dll dog_windows_[开发号].dll dog_windows_x64_[开发号].dll

超级狗文档保护插件

超级狗文档保护插件安装包 (DogDFPWin.msi) 位于光盘路径: Windows\Installed\DRM\Redistribute 中。

该插件支持以下文件类型:

- Office 文档包括 docx, doc, pptx, xlsx, xls
- PDF 文档

该保护方案支持以下平台：

- Windows 7（32 位和 64 位）
- Windows 8（32 位和 64 位）
- Windows 8.1（32 位和 64 位）
- Windows 10（32 位和 64 位）

最终用户在安装超级狗文档保护插件并将超级狗连接到机器上后，使用超级狗 DRM 打包工具加密保护的 Office 文档和 PDF 文档可以双击打开。

超级狗 Media Player 插件

超级狗 Media Player 插件安装包（DogWMPPluginSetup.exe）位于光盘路径：Windows\Installed\DRM\Redistribute\EXE 中。

使用超级狗 DRM 打包工具加密保护的媒体文件，最终用户在安装超级狗 Media Player 插件后，只需双击被保护视频文件就可以启动 Windows Media Player（支持版本：11, 12）来播放。

该插件支持以下文件类型：

mpg, dat, avi, wav 和 mp3。



注意：在您使用 Windows Media Player 播放被保护的 AVI 文件的过程中，如果出现 AVI 文件无法播放或者播放过程中有声音、无图像的情况，请您查看 AVI 文件的视频编码格式，下载并安装相应的解码器后，再次播放被保护的 AVI 文件。

下表列出了几种常见的 AVI 文件视频编码格式及相应的解码器：

视频编码格式	解码器
MPEG4 (DivX)	DivX Codec
MPEG4 (Xvid)	Xvid Codec
AVC (H264)	CoreAVC Codec

超级狗 DRM 浏览器插件

超级狗 DRM 浏览器插件安装包（DogBrowserPlugin.msi）位于光盘路径：Windows\Installed\DRM\Redistribute 中。

使用超级狗 DRM 打包工具加密的 Flash 媒体文件，最终用户在安装超级狗 DRM 浏览器插件后，可以使用 IE 浏览器来播放。

该插件支持如下平台和 IE 版本（只支持 32 位 IE）：

- Windows XP: IE6, IE7, IE8;
- Windows 7 (32 位和 64 位): IE8, IE9, IE10, IE11;
- Windows 8 (32 位和 64 位): IE10;
- Windows 8.1 (32 位和 64 位): IE11;
- Windows 10 (32 位和 64 位): IE11。

测试过如下 Adobe Flash Player 版本:

Adobe Flash Player 21.0

使用插件播放加密保护的 Flash 文件和 MP4 文件

安装超级狗 DRM 浏览器插件后, 可以使用 IE 浏览器播放在线或本地的外壳加密保护后的 flv 和 swf 文件。如需更多的信息, 请查看安装目录下的 DRM 目录中的《*超级狗 Flash 文件和 MP4 文件保护方案说明*》。

操作步骤

播放在线的 web 站点的加密 Flash 文件和 MP4 文件

1. 编辑您的网页用于播放 flv、swf 或 MP4 文件 (播放 flv 文件需要准备一个 flv 网页播放器 (通常是一个 swf 文件), 可参考示例目录: **Samples/flv/on-line**; 播放 swf 文件, 可参考示例目录: **Samples/swf/on-line**; 播放 MP4 文件, 可参考示例目录: **Samples/mp4/on-line**)。
2. 将您的网页放到您的 Web 站点上。
3. 将超级狗和超级狗 DRM 浏览器插件安装包发给您的用户;
4. 您的用户在安装超级狗 DRM 浏览器插件并将超级狗连接到机器上后, 他就可以用 IE 来浏览您的网站上的网页并播放外壳加密保护后的 Flash 文件或 MP4 文件了。

播放本地的加密 Flash 文件和 MP4 文件

1. 编辑您的网页用于播放 flv、swf 或 MP4 文件 (播放 flv 文件需要准备一个 flv 网页播放器 (通常是一个 swf 文件)), 可参考示例目录: **Samples/flv/local**、**Samples/swf/local** 或 **Samples/mp4/local**。
2. 将超级狗, 超级狗 DRM 浏览器插件安装包, 加密的 Flash 文件或 MP4 文件以及相关的网页文件发给您的用户;
3. 您的用户在安装超级狗 DRM 浏览器插件并将超级狗连接到机器上后, 他可以用 IE 直接打开、播放外壳加密保护的 swf 文件、flv 文件或 MP4 文件。

Linux

超级狗运行环境

超级狗运行环境是参与被保护软件与超级狗之间通讯的组件。在最终用户安装超级狗运行环境后, 被保护软件才能与超级狗通讯。

使用 RPM 或 DEB 命令安装超级狗运行环境

1. 将已经插上计算机的超级狗都拔出；
2. 将超级狗安装光盘放入光盘驱动器。找到光盘路径：`Linux/Runtime`，将里面的文件拷贝到本地文件夹中；
3. 作为一个 `root` 用户，在终端窗口中进入包含 `Runtime` 的本地路径中，然后输入以下命令：

- RedHat 或 SUSE:

```
rpm -i superdog-1.0-1.i386.rpm
```

- Ubuntu 或 Debian 32 位操作系统:

```
dpkg -i superdog_1.0-1_i386.deb
```

- Ubuntu 或 Debian 64 位操作系统:

```
dpkg -i --force-architecture superdog_1.0-1_i386.deb
```

所有安装操作都需要 `root` 权限。在 Ubuntu 或 Debian 系统中，需要在以上命令前加入前缀 `sudo`，对于其他操作系统需要先通过 `su` 命令切换当前用户为 `root` 用户。

4. 重新将超级狗连接到计算机上。

使用 RPM 或 DEB 命令卸载超级狗运行环境

作为一个 `root` 用户，在终端窗口输入以下命令：

- RedHat 或 SUSE:

```
rpm -e superdog
```

- Ubuntu 或 Debian 32 位操作系统:

```
dpkg -r superdog
```

- Ubuntu 或 Debian 64 位操作系统:

```
dpkg -r superdog:i386
```

所有卸载操作都需要 `root` 权限。在 Ubuntu 或 Debian 系统中，需要在以上命令前加入前缀 `sudo`，对于其他操作系统需要先通过 `su` 命令切换当前用户为 `root` 用户。

第 5 部分 附录

本部分包含如下内容：

- 附录 A 疑难解答 第 82 页

提供对于常见问题的检查事项，以及可能遇到特定问题的列表及解决方法。

- 附录 B 超级狗 Licensing API 参考 第 84 页

提供 Licensing API 的函数概述、各函数的结构声明和详细信息、以及所有返回代码的摘要和说明。

- 附录 C 术语表 第 117 页

全面的术语表，简明扼要地解释超级狗的术语。

附录 A 疑难解答

超级狗符合最高质量保证标准，但和其他任何的计算机外围设备一样，有可能由于设备缺陷或不正确的安装而无法在某些计算机配置中运行。本附录可以帮助开发商应对此类情形。

除本附录中提供的信息之外，您还可以拨打技术支持电话获得解决方案。

本附录包含如下内容：

- 检查事项
- 问题与解决方法

检查事项

针对最终用户可能遇到的问题，请检查以下内容：

- 返回的错误代码或消息的内容。
- 超级狗是否正确连接至计算机 USB 端口。
- 最终用户的硬件或操作系统是否出现技术故障，例如：设备管理器冲突、系统事件、启动日志失败等。
- 被保护软件运行在相同型号的其他计算机上时是否出现该问题。

问题与解决方法

问题	被保护的应用程序找不到超级狗
解决方法	<p>超级狗 LED 是否亮起？如果未亮起，可能是以下原因之一所致：</p> <ul style="list-style-type: none"> • 超级狗未正确连接到 USB 端口。断开连接，几秒钟之后重新连接。如果 LED 亮起，则应用程序应该可以访问。 • 检查 USB 端口是否正常运行。断开其他所有的 USB 端口连接。将超级狗连接到不同的 USB 端口。尝试在无法访问超级狗的端口使用不同的 USB 设备。 <p>检查超级狗上的开发号是否与被保护应用程序的开发号相匹配。</p>
问题	加密保护后的应用程序被杀毒软件误报
解决方法	<p>当使用外壳工具加密保护后的应用程序可能会被一些杀毒软件误报为病毒, 此时请尝试使用以下方法：</p> <ul style="list-style-type: none"> • 使用权威的证书授权中心（例如：VeriSign 公司）签发的数字证书对加密保护后的应用程序进行数字签名，杀毒软件误报的情况将大大减少。 • 将加密保护后的应用程序提交给相关的杀毒软件提供商，将此应用程序添加到杀毒软件的白名单中。

附录 B

超级狗 Licensing API 参考

本附录提供超级狗 Licensing API 各函数的详细说明。关于如何实施超级狗 Licensing API 软件保护，请参阅第 3 章 [超级狗 Licensing API 保护](#)。如需超级狗 Licensing API 的全套文档，请参阅超级狗安装目录下的帮助文件：

...\Program Files\Gemalto\SuperDog\[version]\API\Licensing\licensing_api_zh.chm

针对主流编程语言的超级狗 Licensing API 示例位于超级狗安装目录下的 Samples 文件夹：

...\Program Files\Gemalto\SuperDog\[version]\Samples

本附录包含如下内容：

- [函数概述](#)
- [详细说明](#)
- [状态代码](#)

函数概述

本参考中的 Licensing API 函数适用于 C 编程语言接口。每个函数的信息包含以下内容：

- 名称：在 C 语言接口中所指定的该函数名称。
- 说明：该函数主要用途的简要说明。
- 语法：C 编程语言中明确的函数声明。
- 参数：该函数的相关参数。
- 返回值：执行该函数后所有可能的返回结果。
- 使用说明：关于怎样使用该函数的详细信息。

函数	说明
<code>dog_login()</code>	登录特定特征并建立会话环境
<code>dog_login_scope()</code>	根据自定义的搜索参数登录特定的特征并建立会话环境
<code>dog_logout()</code>	从会话环境中注销
<code>dog_encrypt()</code>	使用 AES 加密算法加密数据
<code>dog_decrypt()</code>	使用 AES 加密算法解密数据
<code>dog_read()</code>	从超级狗中的数据文件读取数据
<code>dog_write()</code>	将数据写入超级狗中的读写或写一次类型的数据文件
<code>dog_get_size()</code>	检索数据文件的字节大小
<code>dog_get_time()</code>	从超级狗读取当前时间
<code>dog_datetime_to_dogtime()</code>	将日期时间值转换为超级狗时间值
<code>dog_dogtime_to_datetime()</code>	将超级狗时间值转换为日期时间值
<code>dog_get_info()</code>	按自定义参数搜索并按自定义格式显示信息
<code>dog_get_sessioninfo()</code>	检索有关会话环境的信息
<code>dog_get_version()</code>	检索库的版本和编译号
<code>dog_free()</code>	释放被分配的内存资源
<code>dog_update()</code>	把升级的许可写入超级狗

详细说明

按上述函数概述中提到各函数的信息，本节将提供各 Licensing API 各函数的详细说明。

dog_login()

描述

登录到某个特征从而建立会话环境。

语法

```
dog_status_t DOG_CALLCONV dog_login(
    dog_feature_t      feature_id,
    dog_vendor_code_t vendor_code,
    dog_handle_t *     handle
)
```

参数

feature_id	[输入]特征 ID，储存于超级狗中特征的唯一标识符。
vendor_code	[输入]指向开发商代码的指针
handle	[输出]指向产生的会话句柄的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_NOT_FOUND	未找到所需的超级狗
DOG_FEATURE_EXPIRED	特征已失效
DOG_FEATURE_NOT_FOUND	特征不可用
DOG_INSUF_MEM	系统内存不足
DOG_UNKNOWN_VCODE	API 不识别开发商代码
DOG_INV_VCODE	输入的开发商代码无效

DOG_TIME_ERR	系统时钟已被篡改
DOG_TMOF	打开的登录会话数目过多
DOG_TS_DETECTED	程序在终端服务器上运行
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境 (License Manager) 通讯错误
DOG_OLD_LM	超级狗的运行环境版本太旧

使用说明

该函数登录到超级狗中的一个特征 ID，并建立会话环境。

当登录到默认特征 (特征 ID 为 0) 时，API 仅搜索超级狗，而忽略超级狗中指定的许可信息。

开发商代码存储于您系统中的开发商代码文件夹 (VendorCodes) 中。没有正确的开发商代码，将无法成功调用该函数。

相关主题

- [dog_logout\(\)](#)
- [dog_login_scope\(\)](#)

dog_login_scope()

描述

按自定义的搜索参数登录到一个特征。

语法

```
dog_status_t DOG_CALLCONV dog_login_scope (
    dog_feature_t      feature_id,
    const char *       scope,
    dog_vendor_code_t  vendor_code,
    dog_handle_t *     handle
)
```

参数

feature_id	[输入]特征 ID，储存于超级狗中特征的唯一标识符。
scope	[输入]要搜索的数据的 XML 格式定义。更多信息，参阅帮助文档中《 <i>超级狗 Licensing API 帮助</i> 》的“范围输入 XML 标记”一节。
vendor_code	[输入]指向开发商代码的指针
handle	[输出]指向产生的会话句柄的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_NOT_FOUND	未找到所需的超级狗
DOG_FEATURE_EXPIRED	特征已失效
DOG_FEATURE_NOT_FOUND	特征不可用
DOG_INSUF_MEM	系统内存不足
DOG_INV_SPEC	无效的 XML 格式
DOG_INV_SCOPE	无效的 XML 范围
DOG_INV_VCODE	输入的开发商代码无效

DOG_INVALID_PARAMETER	无效的函数参数
DOG_SCOPE_RESULTS_EMPTY	不能找到与范围匹配的特征
DOG_TIME_ERR	系统时钟已被篡改
DOG_TMOF	打开的登录会话数目过多
DOG_TS_DETECTED	程序在终端服务器上运行
DOG_UNKNOWN_VCODE	API 不识别开发商代码
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境 (License Manager) 通讯错误
DOG_OLD_LM	超级狗的运行环境版本太旧

使用说明

使用该函数可以指定条件用以指定到哪里搜索登录信息。

开发商代码存储在您系统中的开发商代码文件夹 (VendorCodes) 中。没有正确的开发商代码，无法成功调用该函数。

相关主题

- [dog_get_info\(\)](#)
- [dog_get_sessioninfo\(\)](#)
- [dog_login\(\)](#)
- [dog_logout\(\)](#)

dog_logout()

描述

从当前的一个会话环境中注销。

语法

```
dog_status_t DOG_CALLCONV dog_logout (
    dog_handle_t    handle
)
```

参数

handle	[输入]会话句柄
--------	----------

返回值

DOG_STATUS_OK	请求已成功完成
DOG_INV_HND	输入函数的句柄无效
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境 (License Manager) 通讯错误

使用说明

使用该函数终止与 API 对象的连接。一旦从会话注销，所有分配给该会话的内存资源都被释放。

相关主题

- [dog_login\(\)](#)
- [dog_login_scope\(\)](#)

dog_encrypt()

描述

使用 AES 加密算法加密指定的数据缓冲区。

语法

```
dog_status_t DOG_CALLCONV dog_encrypt(
    dog_handle_t      handle,
    void *            buffer,
    dog_size_t        length
)
```

参数

handle	[输入]会话句柄
buffer	[输入]要被加密的数据
length	[输入]将被加密的数据缓冲区的字节大小，要求最少有 16 个字节。

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_HND	输入函数的句柄无效
DOG_SCHAN_ERR	安全通道中发生了通信错误
DOG_TIME_ERR	系统时钟已被篡改
DOG_TOO_SHORT	加密或解密的数据长度太短
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager)通讯错误

使用说明

该函数使用超级狗内的加密引擎，所指定的会话句柄决定特定的超级狗对数据缓冲区执行加密。密钥驻留在超级狗内。如果加密操作失败，数据缓冲区不被改变。

要对数据缓冲区执行解密，使用 `dog_decrypt()`。

相关主题

- `dog_decrypt()`

dog_decrypt()

描述

与 dog_encrypt() 函数相反，该函数将把加密的数据返回为未加密状态。

语法

```
dog_status_t DOG_CALLCONV dog_decrypt(
    dog_handle_t      handle,
    void *            buffer,
    dog_size_t        length
)
```

参数

handle	[输入]会话句柄
buffer	[输入]要被解密的数据
length	[输入]将被解密的数据缓冲区的字节大小，要求最少有 16 个字节。

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_HND	输入函数的句柄无效
DOG_SCHAN_ERR	安全通道中发生了通信错误
DOG_TIME_ERR	系统时钟已被篡改
DOG_TOO_SHORT	加密或解密的数据长度太短
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager) 通讯错误

使用说明

该函数使用超级狗中的加密引擎，所指定的会话句柄决定特定的超级狗对数据缓冲区执行解密，密钥驻留在超级狗内。如果解密操作失败，数据将不被改变。

要加密数据缓冲区，使用 `dog_encrypt()`。

相关主题

- `dog_encrypt()`

dog_read()

描述

从超级狗中的数据文件读取数据。

语法

```
dog_status_t DOG_CALLCONV dog_read (
    dog_handle_t      handle,
    dog_fileid_t      fileid,
    dog_size_t        offset,
    dog_size_t        length,
    void *            buffer
)
```

参数

handle	[输入]会话句柄
fileid	[输入]将被读取的数据文件的标识符
offset	[输入]数据文件的字节偏移量
length	[输入]从数据文件读取的字节数
buffer	[输出]指向被检索数据的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯中出现 USB 通信错误
DOG_INV_FILEID	无法识别文件标识符
DOG_INV_HND	输入函数的句柄无效
DOG_MEM_RANGE	请求超出数据文件的范围
DOG_SCHAN_ERR	安全通道中发生了通信错误

DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager) 通讯错误
--------------------	------------------------------------

使用说明

读取数据文件前，使用 `dog_get_size()` 函数确定将要读取数据文件的字节大小。

相关主题

- `dog_get_size()`
- `dog_write()`

dog_write()

描述

将数据写入超级狗中的读写或写一次类型的数据文件。

语法

```
dog_status_t DOG_CALLCONV dog_write(
    dog_handle_t      handle,
    dog_fileid_t     fileid,
    dog_size_t        offset,
    dog_size_t        length,
    const void *      buffer
)
```

参数

handle	[输入]会话句柄
fileid	[输入]将被写入的数据文件的标识符
offset	[输入]数据文件的字节偏移量
length	[输入]将写入数据字节数
buffer	[输入]指向要写入数据的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_FILEID	无法识别的文件标识符
DOG_INV_HND	输入函数的句柄无效
DOG_MEM_RANGE	请求超出数据文件的范围
DOG_SCHAN_ERR	安全通道中发生了通信错误
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager)通讯错误

DOG_ACCESS_DENIED	访问被拒绝
-------------------	-------

使用说明

写入数据前，使用 `dog_get_size()` 函数确定要写入的数据字节大小。

相关主题

- `dog_get_size()`
- `dog_read()`

dog_get_info()

描述

按自定义参数检索信息，并按自定义格式显示信息。

语法

```
dog_status_t DOG_CALLCONV dog_get_info(
    const char *      scope,
    const char *      format,
    dog_vendor_code_t vendor_code,
    char **           info
)
```

参数

scope	[输入]以 XML 格式定义的要搜索数据范围。有关更多信息，参阅帮助文档中《 <i>超级狗 Licensing API 帮助</i> 》的“范围输入 XML 标记”一节。
format	[输入]用于显示被搜索到数据的 XML 格式定义。有关更多信息，参阅帮助文档中《 <i>超级狗 Licensing API 帮助</i> 》的“格式输入 XML 标记”一节。
Vendor_code	[输入]指向开发商代码的指针
info	[输出]指向检索到的以 XML 格式显示的信息的指针。关于 XML 标记，参阅帮助文档中《 <i>超级狗 Licensing API 帮助</i> 》的“输出 XML 标记”一节。

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INSUF_MEM	系统内存不足
DOG_INV_FORMAT	无效 XML 格式
DOG_INV_SCOPE	无效 XML 范围
DOG_INV_VCODE	输入的开发商代码无效

DOG_INVALID_PARAMETER	无效的函数参数
DOG_SCOPE_RESULTS_EMPTY	不能找到与范围匹配的特征
DOG_UNKNOWN_VCODE	API 不识别的开发商代码
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager) 通讯错误

使用说明

无需登录特征就可调用该函数。

如果检索到的信息有合适的格式，可以作为模板用于 `dog_login_scope()` 函数。

开发商代码储存于系统中的开发商代码文件夹(VendorCodes)中。如果开发商代码无效，无法成功调用该函数。

该函数为搜索到的信息分配内存。使用 `dog_free()` 可以释放被分配的内存资源。

相关主题

- `dog_get_sessioninfo()`
- `dog_free()`
- `dog_login_scope()`

dog_get_size()

描述

检索数据文件的字节大小。

语法

```
dog_status_t DOG_CALLCONV dog_get_size(
    dog_handle_t      handle,
    dog_fileid_t     fileid,
    dog_size_t *     size
)
```

参数

handle	[输入]会话句柄
fileid	[输入]将被查询数据文件的标识符
size	[输出]指向文件大小结果的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_FILEID	无法识别的文件标识符
DOG_INV_HND	输入函数的句柄无效
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager)通讯错误

使用说明

此函数用于查询一份数据文件的字节大小。在读写超级狗的数据文件时，该信息很有用。数据文件中首位字节的索引为 0。

相关主题

- `dog_read()`
- `dog_write()`

dog_get_time()

描述

从超级狗读取当前时间。

语法

```
dog_status_t DOG_CALLCONV dog_get_time (
    dog_handle_t      handle,
    dog_time_t *     time
)
```

参数

handle	[输入]会话句柄
time	[输出]指向当前虚拟时间的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_TIME_ERR	系统时钟已被篡改
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_HND	输入函数的句柄无效
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager)通讯错误

使用说明

该函数返回超级狗虚拟时钟的时间，即从 1970 年 1 月 1 日 0:00 起所经过的秒数。使用 `dog_dogtime_to_datetime()` 函数可以转换该秒数值到日期时间值（UTC）。当函数返回值为 `DOG_TIME_ERR` 时，仍然可以取到超级狗虚拟时钟的时间。

相关主题

- [dog_dogtime_to_datetime\(\)](#)

dog_datetime_to_dogtime()

描述

将日期时间值转换为超级狗时间。

语法

```
dog_status_t DOG_CALLCONV dog_datetime_to_dogtime (
    unsigned int    day,
    unsigned int    month,
    unsigned int    year,
    unsigned int    hour,
    unsigned int    minute,
    unsigned int    second,
    dog_time_t *    time
)
```

参数

day	[输入]日值（范围 1-31）
month	[输入]月份值（范围 1-12）
year	[输入]年份值（自 1970 年起）
hour	[输入]小时值（范围 0-23）
minute	[输入]分钟值（范围 0-59）
second	[输入]秒值（范围 0-59）
time	[输出]指向产生的时间值的指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_INV_TIME	输入的时间值超出被支持的数值范围

使用说明

转换出的超级狗时间值是自 1970 年 1 月 1 日 0:00 (UTC) 起经过的秒数。

相关主题

- `dog_get_time()`
- `dog_dogtime_to_datetime()`

dog_dogtime_to_datetime()

描述

将超级狗时间值转换为日期时间值。

语法

```
dog_status_t DOG_CALLCONV dog_dogtime_to_datetime (
    dog_time_t          time,
    unsigned int *     day,
    unsigned int *     month,
    unsigned int *     year,
    unsigned int *     hour,
    unsigned int *     minute,
    unsigned int *     second
)
```

参数

time	[输入]时间值
day	[输出]日期值指针
month	[输出]月份值指针
year	[输出]年份值指针
hour	[输出]小时值指针
minute	[输出]分钟值指针
second	[输出]秒值指针

返回值

DOG_STATUS_OK	请求已成功完成
DOG_INV_TIME	输入的时间值超出被支持的数值范围

使用说明

超级狗时间值是自 1970 年 1 月 1 日 0:00 所经过的秒数。函数所有转换出的日期时间值是基于世界标准时间 (UTC)。

该函数可以与检索超级狗时间值的函数以及检索许可的时间值的函数结合使用。

相关主题

- `dog_get_time()`
- `dog_datetime_to_dogtime()`
- `dog_get_info()`
- `dog_get_sessioninfo()`

dog_get_sessioninfo()

描述

检索会话环境信息。

语法

```
dog_status_t DOG_CALLCONV dog_get_sessioninfo (
    dog_handle_t      handle,
    const char *      format,
    char **           info
)
```

参数

handle	[输入]会话句柄
format	<p>[输入]以 XML 格式表示的数据输出结构类型定义。以下是已有的预定义格式：</p> <ul style="list-style-type: none"> • DOG_KEYINFO: 用于检索关于超级狗的信息。 • DOG_SESSIONINFO: 用于检索关于会话的信息。 • DOG_UPDATEINFO: 用于检索通常包含于 C2V 文件中关于许可升级的信息。检索到已部署超级狗的信息包括许可以及数据文件。 <p>您也可以自定义格式。更多有关自定义格式或使用预定义格式，参阅帮助文档中的《<i>超级狗 Licensing API 帮助</i>》的“使用 XML 标记定义格式”一节。</p>
info	[输出]指向被检索到信息的指针，该信息以 XML 格式表示。更多关于使用 XML 标记格式，请参阅帮助文档中的《 <i>超级狗 Licensing API 帮助</i> 》的“输出 XML 标记”一节。

返回值

DOG_STATUS_OK	请求已成功完成
DOG_BROKEN_SESSION	会话被中断
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INV_FORMAT	无效的 XML 格式

DOG_INV_HND	输入函数的句柄无效
DOG_INVALID_PARAMETER	无效的函数参数
DOG_TIME_ERR	系统时钟已被篡改
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager) 通讯错误

使用说明

该函数检索到的信息应用于：

- 已部署的超级狗
- 当前或特定的登录会话
- 各种升级，包括许可升级、固件升级、数据文件升级等

该函数为其检索到的信息分配内存。使用 `dog_free()` 函数可以释放被分配的内存资源。

使用 `dog_dogtime_to_datetime()` 函数可以将返回的时间值转换为当前的日期与时间值。

相关主题

- `dog_free()`
- `dog_dogtime_to_datetime()`
- `dog_get_info()`

dog_get_version()

描述

检索库的版本和编译号。

语法

```
dog_status_t DOG_CALLCONV dog_get_version (
    unsigned int *    major_version,
    unsigned int *    minor_version,
    unsigned int *    build_server,
    unsigned int *    build_number,
    dog_vendor_code_t vendor_code
)
```

参数

major_version	[输出]指向主版本号的指针
minor_version	[输出]指向副版本号的指针
build_server	[输出]指向编译服务器 ID 的指针
build_number	[输出]指向编译版本号的指针
vendor_code	[输入]指向开发商代码的指针

返回值

DOG_STATUS_OK	请求已成功完成
---------------	---------

使用说明

除指向开发商代码的指针以外，如不需信息，其他指针可以为 NULL。

dog_update()

描述

把升级的许可写入超级狗。

语法

```
dog_status_t DOG_CALLCONV dog_update (
    const char *      update_data,
    char **           ack_data
)
```

参数

update_data	[输入]指向完整的升级数据的指针
ack_data	[输出]指向储存回执数据的缓冲区的指针。关于信息的 XML 标记格式，请参阅帮助文档中《 <i>超级狗 Licensing API 帮助</i> 》的“DOG_UPDATE XML 标记”。

返回值

DOG_STATUS_OK	更新成功完成
DOG_DEVICE_ERR	与超级狗通讯时出现 USB 通信错误
DOG_INSUF_MEM	系统内存不足
DOG_INV_SIG	签名验证失败
DOG_INV_UPDATE_CNTR	更新计数器设置不正确
DOG_INV_UPDATE_DATA	未找到所需的 XML 标记，或者二进制数据内容丢失或无效。
DOG_INV_UPDATE_NOTSUPP	该超级狗不支持升级请求
DOG_KEYID_NOT_FOUND	未找到待升级的超级狗
DOG_NO_ACK_SPACE	升级要求回执数据，但输入参数 ack_data 为 NULL。
DOG_TOO_MANY_KEYS	当前连接的超级狗过多

DOG_UNKNOWN_ALG	V2C 文件中使用了未知算法
DOG_UPDATE_TOO_NEW	文件中的升级计数器的数值大于超级狗中的升级计数器的数值，不允许安装 V2C 文件。
DOG_UPDATE_TOO_OLD	文件中的升级计数器的数值小于超级狗中的升级计数器的数值，不允许安装 V2C 文件。
DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager) 通讯错误

使用说明

要在已部署超级狗上执行升级，必须有以下所有信息：

1. 将被操作(写入)的超级狗
2. 访问超级狗所需的开发商代码
3. 实际升级信息(V2C 文件)

无需登录特征就可调用该函数。

取决于升级数据中的要求，该函数可以返回一个回执代码。回执代码由本升级签名并加密。该回执代码是超级狗内的软件使用许可已被升级的凭据。

分配给回执数据的内存资源，必须使用 `dog_free()` 函数释放。

相关主题

- [dog_free\(\)](#)

dog_free()

描述

释放被分配的内存资源。

语法

```
void DOG_CALLCONV dog_free(  
  
    char * info  
)
```

参数

info	[输入]指向要被释放的内存资源的指针
------	--------------------

使用说明

使用 `dog_get_info()`、`dog_get_sessioninfo()` 和 `dog_update()` 函数时，函数检索到的信息占用内存资源，`dog_free()` 函数用于释放被分配的内存资源。

`dog_free()` 函数没有返回值。`dog_free()` 函数仅用于 C 代码。

相关主题

- `dog_get_info()`
- `dog_get_sessioninfo()`
- `dog_update()`

状态代码

以下是运行超级狗 Licensing API 的函数可能返回的状态代码列表。

编号	状态代码	说明
0	DOG_STATUS_OK	请求成功完成
1	DOG_MEM_RANGE	请求超出数据文件的范围
3	DOG_INSUF_MEM	系统内存不足
4	DOG_TMOF	打开的登录会话数目过多
5	DOG_ACCESS_DENIED	访问被拒绝
7	DOG_NOT_FOUND	未找到所需的超级狗
8	DOG_TOO_SHORT	加密/解密的数据长度太短
9	DOG_INV_HND	输入函数的句柄无效
10	DOG_INV_FILEID	无法识别文件标识符
15	DOG_INV_FORMAT	无效的 XML 格式
16	DOG_REQ_NOT_SUPP	当前会话环境不支持的功能
17	DOG_INV_UPDATE_OBJ	无效的升级内容
18	DOG_KEYID_NOT_FOUND	未找到待升级的超级狗
19	DOG_INV_UPDATE_DATA	未找到所需的 XML 标记，或者二进制数据内容丢失或无效。
20	DOG_INV_UPDATE_NOTSUPP	该超级狗不支持升级请求
21	DOG_INV_UPDATE_CNTR	升级计数器设置不正确
22	DOG_INV_VCODE	输入的开发商代码无效
24	DOG_INV_TIME	输入的时间值超出被支持的数值范围
26	DOG_NO_ACK_SPACE	升级要求回执数据，但输入参数 <code>ack_data</code> 为 NULL。
27	DOG_TS_DETECTED	程序在终端服务器上运行
29	DOG_UNKNOWN_ALG	V2C 文件中使用了未知算法

编号	状态代码	说明
30	DOG_INV_SIG	签名验证失败
31	DOG_FEATURE_NOT_FOUND	特征不可用
33	DOG_LOCAL_COMM_ERR	API 和超级狗运行环境(License Manager)通讯错误
34	DOG_UNKNOWN_VCODE	API 不识别开发商代码
35	DOG_INV_SPEC	无效的 XML 格式
36	DOG_INV_SCOPE	无效的 XML 范围
37	DOG_TOO_MANY_KEYS	当前连接的超级狗数目过多
39	DOG_BROKEN_SESSION	会话被中断
41	DOG_FEATURE_EXPIRED	特征已失效
42	DOG_OLD_LM	超级狗的运行环境版本太旧
43	DOG_DEVICE_ERR	与超级狗通讯中出现 USB 通信错误
45	DOG_TIME_ERR	系统时钟已被篡改
46	DOG_SCHAN_ERR	安全通道中发生了通信错误
47	DOG_STORAGE_CORRUPT	超级狗软许可数据被破坏
50	DOG_SCOPE_RESULTS_EMPTY	不能找到与范围匹配的特征
52	DOG_HARDWARE_MODIFIED	超级狗软许可与设备不匹配
54	DOG_UPDATE_TOO_OLD	文件中的升级计数器的数值小于超级狗中的升级计数器的数值，不允许安装 V2C 文件。
55	DOG_UPDATE_TOO_NEW	文件中的升级计数器的数值大于超级狗中的升级计数器的数值，不允许安装 V2C 文件。
64	DOG_CLONE_DETECTED	发现克隆的超级狗软许可
65	DOG_UPDATE_ALREADY_ADDED	已经使用过当前的超级狗升级文件
78	DOG_SECURE_STORE_ID_MISMATCH	超级狗安全存储 ID 与当前应用程序不匹配
400	DOG_NO_API_DYLIB	未找到 API 的动态库
401	DOG_INV_API_DYLIB	API 的动态库无效

编号	状态代码	说明
500	DOG_INVALID_OBJECT	对象的初始化不正确
501	DOG_INVALID_PARAMETER	无效的函数参数
502	DOG_ALREADY_LOGGED_IN	两次登录到同一对象
503	DOG_ALREADY_LOGGED_OUT	从同一对象注销两次
525	DOG_OPERATION_FAILED	系统或平台的使用不正确
698	DOG_NOT_IMPL	未实施要求的功能
699	DOG_INT_ERR	API 中内部错误

附录 C

术语表

AES	Advanced Encryption Standard（高级加密标准）的缩写，是一种符合行业标准的对称密钥加密算法，也是超级狗执行加密和解密操作的基础。
API	Application Program Interface（应用程序接口）的缩写。本产品的 API 是您的应用程序用来与超级狗进行通讯的一组客户端接口。
DEMOMA	用于代表演示开发商代码的开发号。
ECC	Elliptic Curve Cryptography（椭圆曲线加密算法）的简称。ECC 主要用于创建数字签名 - 使用私钥签名，使用公钥验证。通过 ECC 密钥交换，被保护软件和超级狗获取加密密钥，得以进行通讯。
UTC	Coordinated Universal Time（协调世界时）的简称，又称世界统一时间，世界标准时间。
SuperDog SL	超级狗软许可。用于安卓平台的应用保护。每个使用超级狗的安卓应用将生成自己的安全存储 ID，超级狗软许可除了绑定安卓设备，还将绑定到这个安全存储 ID。
编程超级狗	将许可定义文件应用于超级狗，或者说将许可定义文件中的产品、特征以及数据文件写入超级狗，称为“编程超级狗”。
逆向工程	试图通过追溯已编译程序的源代码来破解算法及目标程序执行流的软件攻击。
后台检查	指被保护的程序随机检查计算机是否连接到所需的超级狗。
加密	将明文数据（原始格式）转换为非目标接收者无法阅读的形式。
加密引擎	超级狗的加密引擎基于 AES 算法，没有正确的加密引擎将无法执行解密。
解密	将加密数据转换为明文数据（原始格式），使之能够被目标接收者/进程译解。
句柄	句柄是应用程序访问超级狗建立会话时产生的唯一标识符。
开发号	是代表开发商代码的唯一字符串，由 5 个字符组成。
开发商代码	每个被保护软件被分配唯一的机密开发商代码。开发商代码允许被保护软件访问超级狗及其资源。

开发商	使用超级狗开发套件对自己的软件产品进行保护的开发商。
密钥	指在密码学中广泛使用的二进制字符串，可以加密和解密数据，还可以执行其他数学运算。
许可定义文件	包括基于特征（功能组件）的使用许可和所需数据，该文件以 XML 格式描述。
升级定义文件	基于许可定义文件的升级定义文件。
升级文件	包含许可升级数据，由开发商发送至最终用户，又称 V2C 文件。
请求文件	包含已部署的使用许可以及超级狗状态信息，由最终用户发送至开发商，又称 C2V 文件。
一次保护、多种分发	指软件保护过程与软件产品的业务流程完全分开、相互独立。实施一次软件保护过程后，可多次对软件功能组件（特征）指定许可类型，组合不同特征形成新产品，并进行分发。
业务流程	产品或业务经理开发产品策略、计划产品、确定产品、准备并实施产品计划的过程。
许可特征	特征指可被独立赋予许可类型的软件功能组件。许可特征可以指整个应用程序、某个模块，或者诸如“打印”、“保存”、“绘图”之类的特定功能。
特征 ID	许可特征的唯一标识符。
产品	表示某个可营销的软件产品。一个产品可以包括一个或几个许可特征，每个许可特征分别被指定各自的许可类型。
产品 ID	产品的唯一标识符。
状态代码	由超级狗 API 返回的错误代码。
最终用户	使用被超级狗授权和保护的软件产品的用户。

索引

- AES 加密和解密, 25
- 安全通道, 26
- 保护策略
 - 常见的软件攻击类型, 54
 - 超级狗通用的保护策略, 57
 - 总的软件保护策略, 53
- 保护方法, 27
- 保护内容, 25
- 产品功能, 12
- 产品清单, 12
- 超级狗 Licensing API, 30
 - 登录函数, 35
 - 函数概述, 82
 - 函数组, 33
 - 库文件命名, 32
 - 下载目录, 31
- 超级狗 Licensing API 保护, 29
 - 工作流程, 35
 - 实施前提, 30
 - 示例, 32
 - 需求规划, 34
- 超级狗安装目录, 12
- 超级狗的 V-Clock 虚拟时钟, 26
- 超级狗的工作原理, 25
- 超级狗的软件工具, 19
 - 超级狗 DRM 打包工具, 49
 - 超级狗 Media Player 插件, 75
 - 超级狗编程工具, 73
 - 超级狗加密助手, 39
 - 超级狗开发商向导, 30
 - 超级狗授权管理工具, 65
 - 日志查看工具, 74
 - 外壳加密工具, 44
 - 远程升级工具, 68
 - 远程升级界面定制工具, 70
- 超级狗的业务流程, 18
 - 分发超级狗软件, 74
 - 准备产品计划, 61
- 超级狗数据存储区, 26
- 超级狗运行环境, 77
- 获得支持, 14
 - 技术支持, 14
 - 培训, 14
- 开发商代码和开发号, 21
 - 开发商代码文件的命名, 21
 - 开发商代码文件夹, 21
- 外壳保护.NET 程序集
 - 代码与符号混淆技术, 47
 - 代码植入, 51
 - 方法级别的保护, 46
 - 考虑事项, 45
- 外壳保护功能, 42
 - 基本保护流程, 42
 - 命令行版本, 45
- 一次保护、多种分发, 17
- 与许可相关的文件
 - 请求文件, 65
 - 升级文件, 64
 - 许可定义文件, 63
 - 许可升级定义文件, 64
- 重要概念—特征, 43
- 阻止逆向工程, 27