全国信息安全标准化技术委员会
NATIONAL INFORMATION SECURITY STANDARDIZATION TECHNICAL COMMITTEE

TC260-PG-20205A

网络安全标准实践指南

—移动互联网应用程序（App）使用软件开发工具包（SDK）安全指引

（v1.0-202011）

全国信息安全标准化技术委员会秘书处

2020 年 11 月

本文档可从以下网址获得：
www.tc260.org.cn/

全国信息安全标准化技术委员会
NATIONAL INFORMATION SECURITY STANDARDIZATION TECHNICAL COMMITTEE

前　言

《网络安全标准实践指南》（以下简称《实践指南》）是全国信息安全标准化技术委员会（以下简称“信安标委”）秘书处组织制定和发布的标准相关技术文件，旨在围绕网络安全法律法规政策、标准、网络安全热点和事件等主题，宣传网络安全相关标准及知识，提供标准化实践指引。

声　明

技术支持单位

本《实践指南》得到中国电子技术标准化研究院、深圳市腾讯计算机系统有限公司、三六零科技集团有限公司、北京百度网讯科技有限公司、小米科技有限责任公司、阿里巴巴（北京）软件服务有限公司、浙江蚂蚁小微金融服务集团股份有限公司、北京小桔科技有限公司、中国移动通信集团有限公司、每日互动股份有限公司、华为技术有限公司、北京字节跳动科技有限公司、京东数字科技控股有限公司、北京三快科技有限公司、友盟同欣（北京）科技有限公司、上海钧正网络科技有限公司、北京京东尚科信息技术有限公司、高德软件有限公司、OPPO 广东移动通信有限公司等单位的技术支持。

摘　要

当前，软件开发工具包（SDK）被广泛应用于各类移动互联网应用程序（App）的开发中，为提升App兼容性和灵活性、节约开发成本带来了便利，但由SDK带来的安全风险也引起多方关注。2018年“寄生推”推送SDK通过云端控制的方式对目标用户下发包含恶意功能的代码包，殃及300多款应用，潜在可影响近2000万用户。2020年央视“3.15”晚会曝光了SDK违法违规收集用户个人信息的问题，在社会上引起了强烈反响。

本实践指南依据法律法规和政策标准要求，针对当前App使用SDK过程中可能面临的SDK安全漏洞、恶意行为、违法违规收集使用个人信息等问题，参考当前SDK安全最佳实践，给出了App使用SDK的安全实践指引，旨在减少因SDK造成的App安全与个人信息保护问题。

目　录

1 适用范围

本实践指南给出了 App 使用 SDK 的相关方责任和常见安全问题，并针对常见问题给出了 App 提供者和 SDK 提供者的安全原则和安全措施。

本实践指南适用于 App 提供者使用 SDK 时防范 SDK 安全和合规风险，也适用于 SDK 提供者保障 SDK 安全和用户个人信息时参考。

2 术语定义

2.1 移动互联网应用程序

通过预装、下载等方式获取并运行在移动智能终端上、向用户提供服务的应用软件，简称 App。

2.2 移动互联网应用程序提供者

移动互联网应用程序的开发者、运营者或所有者，简称 App 提供者。

2.3 软件开发工具包

协助软件开发的相关二进制文件、文档、范例和工具的集合，简称 SDK。本指南中的 SDK，是指对实现 App 特定功能的代码进行封装，向外提供简捷的调用接口的二进制文件。

2.4 软件开发工具包提供者

软件开发工具包的开发者、运营者或所有者，简称 SDK 提供者。

2.5 热更新

在不重新下载和安装 App 的情况下，通过动态加载的方式更新 App 中的代码或资源文件，实现 App 功能的更新。

3 概述

3.1 App 使用 SDK 的相关方和责任

App使用SDK通常涉及App用户、App提供者[1]和SDK提供者[2]三方角色，如图1所示。SDK提供者将实现特定功能的代码进行封装，并提供简单的调用接口。App提供者将SDK嵌入App代码中，调用SDK提供的接口实现相应的功能，如果某些SDK具有独立交互界面，App交互页面也会将其嵌入。用户通过App交互界面使用SDK功能时，通常对SDK提供者没有感知，但如果通过页面跳转等方式访问SDK提供者的独立交互页面时，可能感知到SDK提供者。SDK这种方式使得App提供者不必关心所需功能的具体代码实现便能使用相关功能，提高了App开发和运营的效率。

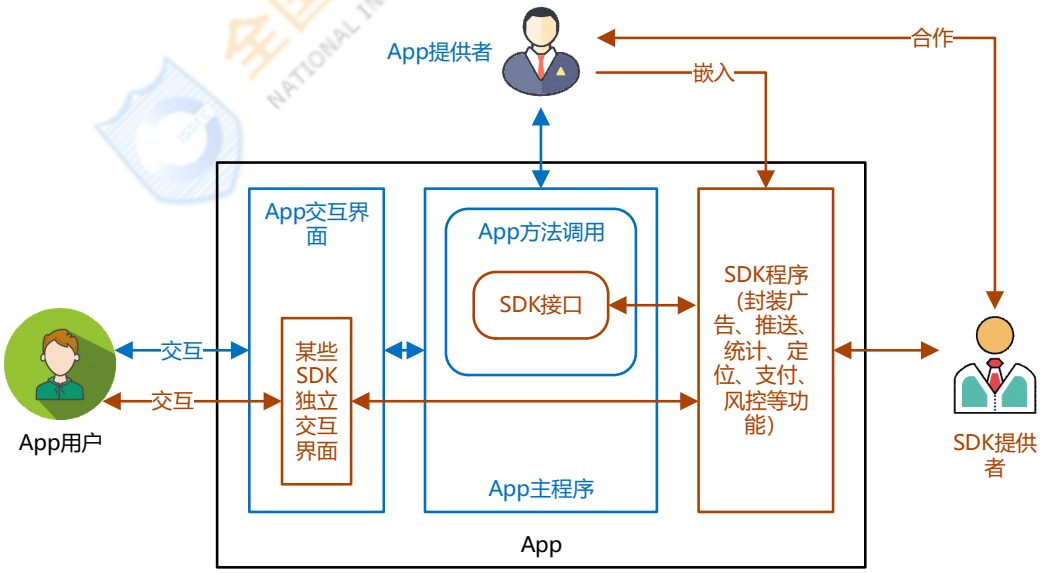

图 1 SDK使用示意图

[1] 也存在 App 提供者和 SDK 提供者同为一方的情况，即 App 提供者使用自身开发的 SDK。
[2] 如果 App 提供者和 SDK 提供者不是同一方，通常将 SDK 称为第三方 SDK。

从App个人信息安全的角度来看，原则上App提供者是App个人信息控制者及保护用户个人信息安全的首要责任人，SDK提供者按照App使用SDK的不同方式承担相应的个人信息安全责任。具体而言：

a）当App嵌入开源SDK，或App提供方与SDK提供方是同一方时，由App提供方承担保护个人信息安全责任；

b）当App提供者委托SDK提供者处理个人信息，或SDK提供者按照App提供者的指导或要求，代表App提供者处理个人信息的，App提供者承担个人信息安全责任、SDK提供者需配合App提供者履行相关责任；

c）如果App提供者和SDK提供者均是以单独身份向App用户提供服务，且均自行决定处理数据的目的与方式时，SDK提供者承担个人信息控制者责任，App提供者承担个人信息控制者和接入第三方管理的责任。

d）如果App提供者和SDK提供者共同决定数据的处理目的与方式时，App提供者和SDK提供者是个人信息共同控制者，需通过合同等形式约定各自承担的责任。如存在侵害个人信息权益，应承担连带责任。

3.2 常见 SDK 类型

当前，SDK被广泛应用于各类App开发中。按SDK功能划分，常见的SDK有框架类、广告类、推送类、统计类、地图类等，详见表1。

表 1 常见 SDK 类型列表

序号	SDK 分类	功能描述
1	框架类	提供开发某一类 App 或跨平台 App 所需的整体框架。

序号	SDK 分类	功能描述
2	广告类	提供广告展示功能，通过使用广告 SDK，App 提供者可以在 App 中展示广告商投放的广告，进而根据用户的点击赚取收益。
3	推送类	提供消息推送功能。
4	统计类	提供收集用户与 App 之间的交互行为的功能。根据用户使用 App 的情况，开发者可以有针对性地改进 App。
5	地图类	提供地图和定位功能。
6	第三方登录类	提供通过其他账号体系（如微博、微信、QQ）等第三方账号登录 App 的功能。
7	社交类	提供社交功能，如消息、分享、排行等功能。
8	支付类	提供移动支付功能。
9	客服类	提供客服对话窗口、客服机器人等客服功能。
10	测试类	提供线上测试功能，如 AB 测试。
11	安全风控类	提供移动业务安全风控功能。
12	Crash 监控类	提供 App 崩溃、App 无响应、卡顿的数据收集与分析。
13	人脸识别类	提供人脸识别、活体检测等功能。
14	语音识别类	提供语音转文字等功能
15	短信验证类	提供短信验证功能。
16	基础功能类	提供 App 的基础功能，如网络访问、图片缓存、多媒体操作等。

4 常见安全问题

SDK向App屏蔽了特定功能的实现细节，简化了App开发和运营，但也正因为如此，SDK自身的行为具有较强的隐蔽性，其所造成的安全问题不易被察觉。此外，一款SDK可能会被多款App集成，因此一旦该SDK出现安全问题，就会影响多款App及其用户。App使用SDK可能面临以下三类常见安全问题：

4.1 SDK 自身安全漏洞

SDK 在开发时聚焦于功能实现而忽视了安全性，可能导致 SDK 本身存在安全漏洞。表 2 列出了 SDK 常见安全漏洞，这些漏洞可被恶意攻击者利用，对嵌入该 SDK 的大量 App 及其终端用户的数据及隐私安全造成严重威胁。

表 2 常见 SDK 安全漏洞

类型	名称
源文件安全	Java 代码未混淆风险
	私有函数调用风险
	AES 弱加密漏洞
	RSA 算法不安全使用漏洞
	随机数不安全使用
	敏感函数调用风险
内部数据交互安全	低保护级别的自定义权限
	PengdingIntent 不安全使用
	携带敏感信息的隐式 Intent 调用
	动态注册广播
	FFmpeg 文件读取
	Intent Scheme URLs 攻击
	Provider 文件目录遍历
	Fragment 注入
	Webview 未移除隐藏接口
	Webview 明文保存密码
	Activity 绑定 browserable 与自定义协议
	存在剪切板读或写操作漏洞检测
通信数据传输安全	SSL 通信服务端检测信任任意证书
	SSL 通信客户端检测信任任意证书

	HTTPS 关闭主机名验证
	Webview 存在本地 Java 接口
	Webview 忽略 SSL 证书错误
	开放 socket 端口
	Webview 启用访问文件数据
本地数据存储安全	getdir 读写权限配置错误
	全局文件读写权限配置错误
	配置文件读写权限配置错误
	AES/DES 硬编码密钥
	打开或创建数据库文件权限配置错误
防御检测	DEX 文件动态加载
	外部加载 so 文件漏洞
	未使用编译器堆栈保护技术
	未使用地址空间随机化技术
	unzip 解压缩（ZipperDown）
	动态链接库中包含执行命令函数
	libupnp 栈溢出漏洞
	Webview 组件远程代码执行（调用 getClassLoader）
	保存明文数字证书风险
	篡改/二次打包风险
	资源文件泄露风险
	so 文件破解风险

其中，ZipperDown 漏洞[3]是由于 App 使用第三方 Zip 库解压 Zip 文件的过程中没有对 Zip 内文件名做校验导致，如果文件名中含有“../”文件路径则可实现目录的上一级跳转，进而实现 App 内任意目

[3] ZipperDown 漏洞，炒作还是一触即发？ https://www.freebuf.com/articles/terminal/172627.html

录的跳转和文件覆盖，攻击者便可对应用资源、代码进行任意篡改、替换，从而实现远程代码劫持等高危操作，危害业务应用场景。

4.2 SDK 恶意行为

SDK恶意行为是指嵌入App中的SDK自身产生的恶意行为，这种恶意行为将破坏使用SDK的App的安全性，对用户权益、数据等方面造成严重威胁。某些SDK在嵌入App的初期可能并不具有恶意行为，但可后续通过热更新动态加载恶意代码实施恶意行为，如图 2 所示。以“寄生推”事件[4]为例，SDK开发者通过云端控制的方式对目标用户下发包含恶意功能的代码包，进行Root提权，静默应用安装等隐蔽操作，进而通过恶意广告行为和应用推广牟取灰色收益。

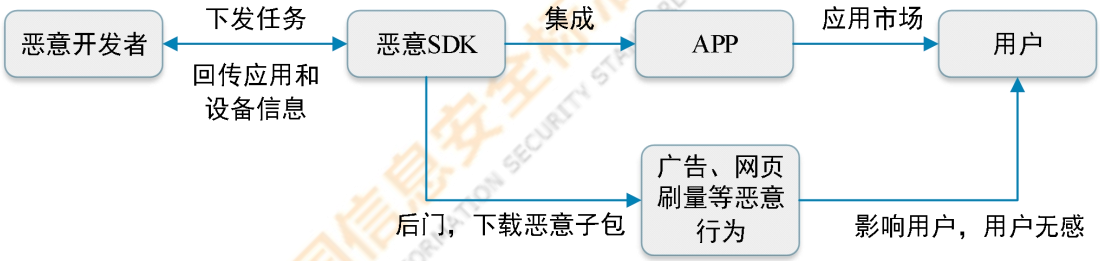

图 2 SDK 热更新后隐蔽执行恶意行为

典型的恶意行为如流量劫持、资费消耗、隐私窃取等，详见表3。

表 3 SDK 恶意行为分类

序号	行为名称	注释
1	流量劫持	SDK信息拉取、上报和展示目标与App提供者设定的目标不同，恶意劫持App流量，可能对App造成损害。
2	资费消耗	SDK可通过消耗用户网络套餐资费、恶意发送收费短信，订阅收费服务等行为，造成用户的资金损失。
3	隐私窃取	SDK在用户不知情或误导用户的情况下，隐蔽窃取用户的通讯录、短信息等个人敏感信息，隐蔽进行拍照、录音等敏感行为，并发送给恶意开发者。

[4] “寄生推”SDK 云控作恶，300 多款应用不幸躺枪. https://www.freebuf.com/articles/terminal/168984.html

序号	行为名称	注释
4	静默下载安装	SDK在后台静默下载、安装其它恶意软件或病毒木马。
5	广告刷量	SDK在用户不知情的情况下，在后台模拟人工点击广告链接进行牟利。
6	恶意广告	SDK向用户推送包含欺诈内容、病毒木马的广告链接。推送过量广告，进而长期占用系统通知栏、屏幕界面，干扰用户正常使用App。
7	勒索	SDK恶意加密用户手机中的文件，干扰用户对手机的正常使用，并以恢复正常使用为由向用户勒索钱财。
8	挖矿	SDK在用户不知情的情况下利用其手机的计算能力来为攻击者获取电子加密货币，对用户设备硬件造成性能损耗。
9	远程控制	SDK在手机端启动本地后台服务器，接收远程控制端发来的控制指令，隐蔽进行上述其他恶意行为。
10	剪切板劫持	SDK对系统剪切板进行监听，获取剪切板中的敏感信息，或者根据剪贴板内容的变化触发悬浮窗，干扰系统功能，欺骗用户，或者影响其他应用正常使用。

4.3 SDK 收集使用个人信息问题

如果SDK收集使用个人信息方面存在安全问题，App使用SDK时将对App用户个人信息构成安全风险，主要体现在:

a）SDK超范围收集个人信息。主要表现在SDK收集与提供服务无关的个人信息，强制申请非必要的权限，自动收集个人信息的频度和时机不合理等。例如，一些SDK会收集设备信息、网络环境信息，读取用户设备已安装应用程序列表，甚至超范围收集用户通讯录、通话记录、地理位置等敏感信息。

b）未说明App嵌入的SDK收集使用个人信息的目的、类型、方式。SDK通常无法独立展示前台页面，其告知行为往往需要借助宿主App透出给用户，但由于一些SDK未向App告知或完整告知自身所收集的个人信息，或者SDK公开了收集使用规则但App担心影响用户体验未向用户明示等原因，造成用户对SDK收集使用个人信息无感知。

c）SDK未经用户同意收集、使用或对外提供个人信息。例如App嵌入的SDK未经用户同意，私自调用权限隐蔽收集个人信息，私自通过自启动、关联启动等方式收集个人信息，SDK实际收集的用户个人信息超出公开文档所声明的系统权限和个人信息，SDK提供者超出用户授权范围使用个人信息，私自向其他应用或服务器发送、共享用户个人信息等。

d）App对嵌入SDK的安全管理监督不足。由于许多App与SDK通过开放平台在线签署开发者服务协议来约定权利义务，而开发者服务协议很少有专门约束数据安全的规定，以及App对SDK收集使用个人信息进行技术检测存在难度等原因，App对嵌入的SDK收集使用个人信息情况不够了解，容易引入SDK违法违规收集使用个人信息的合规风险，需要App对共享给SDK的个人信息，双方权利义务及第三方SDK收集使用个人信息情况等加强管理监督。

5 基本原则和安全措施

5.1 App 使用 SDK 安全原则

App 使用 SDK 处理个人信息时，App 提供者、SDK 提供者应满足 GB/T 35273-2020《信息安全技术 个人信息安全规范》相关角色要求，并均应遵循《个人信息安全规范》的七项基本原则，具体包括：

a) 权责一致原则：App 提供者、SDK 提供者均应对其个人信息处理活动对用户合法权益造成的损害承担相应责任。

b) 目的明确原则：App 选用处理个人信息的 SDK 时，应保证 SDK 具有明确、清晰、具体、合理的个人信息处理目的，且与 App 业务功能直接相关。

c) 选择同意原则：向用户明示嵌入的 SDK 收集使用个人信息的目的、类型、方式，及 App 共享给 SDK 提供者的个人信息类型，并征求用户授权同意。

d) 最小必要原则：App 仅嵌入满足业务功能需要的最少够用的 SDK。SDK 不申请与 App 业务功能无关的权限，只收集满足所提供服务所需的最少个人信息类型、数量和频度。

d) 公开透明原则：SDK 以明确、易懂和合理的方式向 App 公开 SDK 处理个人信息的范围、目的、规则等，SDK 收集使用个人信息实际行为应与公开文档声明保持一致。App 提供者对嵌入的 SDK 收集使用个人信息和安全风险进行管理监督。

e) 确保安全原则：App 提供者、SDK 提供者采取足够的管理措施和技术手段，保护数据的保密性、完整性、可用性，SDK 提供者采取措施保障 SDK 开发安全和隐私设计。

f) 主体参与原则：SDK 提供者建立能够查询、更正、删除其个人信息，及撤回授权同意、投诉等渠道，并协助 App 提供者展示相应渠道，以响应用户个人信息权利请求。

5.2 App 提供者安全措施

App 提供者应基于 5.1 中安全原则，采取充分的安全措施保证使用 SDK 时不引入安全风险，包括但不限于：

a) 遵循合法、正当、必要的原则选择使用SDK。

b) 集成SDK前对SDK进行安全性评估，根据实际情况可选择如下评估内容：

1) 来源安全性评估，包括但不限于：SDK提供者的基本信息；SDK提供者的沟通反馈渠道；SDK隐私政策链接地址；SDK提供者的安全能力；SDK的基本功能；SDK的版本号等。

2) 代码安全性评估，包括但不限于：是否存在已知的恶意代码；是否存在已知的安全漏洞；是否申请敏感权限[5]；是否嵌入了其他SDK等。

3) 行为安全性评估，包括但不限于：调用的敏感权限、目的和频率；收集的个人信息类型、目的和频率；个人信息回传服务器域名、IP地址、所在地域；是否存在热更新行为及热更新是否可主动关闭；传输数据是否加密；是否存在单独收集用户个人信息的界面；是否存在后台自启动和关联启动后收集个人信息的行为等。

[5] 本文件所指的敏感权限，是指可访问用户个人信息（如短信、通信录、设备唯一标识符等）和可调用敏感操作能力（如摄像头、麦克风、精确地理位置等）的系统权限。

c) 使用提供者基本信息明确、沟通反馈渠道有效的SDK。

d) 对于使用的具有热更新功能的SDK，对SDK的热更新内容进行内容校验，对于非官方的热更新内容进行阻断，对于发现问题的热更新内容应及时停用。

e) 对集成后的SDK进行持续动态监测或定期进行安全评估。对于已经发现的SDK安全漏洞，及时修复，或者采用其它替代方案，并从SDK官方渠道及时更新最新版本SDK。对于已经发现存在恶意行为的SDK，及时停止使用。

f) 通过接口调用SDK功能的，对接口增加鉴权机制。

g) 向用户告知所接入的涉及个人信息收集的SDK的名称，SDK收集的个人信息类型、目的和方式，申请的敏感权限、申请目的等，并征得用户同意。若SDK需向用户单独告知收集使用个人信息的行为，App需为其中无单独页面的SDK提供向用户告知的便捷渠道。

h) 与SDK提供者签订合作协议或进一步完善与SDK提供者的合作协议，明确SDK收集的个人信息类型、申请的敏感权限、个人信息的收集目的、保存期限、超期处理方式等，明确双方在个人信息保护方面分别应采取的措施、承担的责任和义务等。当双方合作存在重大变更时[6]，应重新达成合作协议。

i) 停用某SDK后，及时从App中移除该SDK的代码和调用该SDK的代码，存在通过本App共享或收集个人信息的，应敦促SDK

[6] 重大变更所包含内容可参考 GB/T 35273-2020 附录 D。

提供者按照合作协议约定，删除从本App共享或收集的个人信息或做匿名化处理。

5.3 SDK 提供者安全措施

App 提供者在选用 SDK 时，应选用安全性、可靠性高的 SDK，建议 SDK 提供者基于 5.1 中安全原则采取以下安全措施，并向 App 提供者提供有关安全能力的证明。可采取的安全措施包括但不限于:

a) 收集使用个人信息和申请敏感权限应遵循合理、最小、必要原则。

b) 对功能独立的模块，宜进行拆分或提供单独的开启关闭选项，允许App提供者按需进行选择使用或开启关闭，不应强制捆绑无关功能并以此为由申请无关权限或收集无关的个人信息。

c) 收集个人信息的频率应是实现自身业务功能所必需的最低频率。在App用户或App提供者未使用SDK相关业务功能时，不应强制申请权限或通过自启动、关联启动等方式开始收集个人信息。

d) 通过代码审计、代码混淆等方式，增强自身安全性。在发布上线前，进行安全评估，评估内容包括但不限于：完整性校验、恶意代码检测、安全漏洞检测、权限申请和调用频率检测、收集个人信息类型和频率检测、后台自启动和关联启动并收集个人信息的行为检测。发布上线后，持续进行安全监测或定期进行安全检测，发现新的安全漏洞时应及时进行修复并告知App提供者。

e) 通过接口调用提供自身功能的SDK，对接口增加鉴权机制，并对不同App调用接口的上下文环境进行隔离。

f) 数据传输使用HTTPS安全信道、双向证书校验、证书绑定等安全机制，避免因中间人攻击导致传输数据泄露或被篡改。传输用户个人敏感信息的，在传输前，对个人敏感信息内容进行加密。

g) 采用热更新技术的SDK，需建立完善的热更新安全保障机制，包括但不限于：

1) 向App提供者明示自身SDK存在热更新机制；

2) 在热更新推送前至少5个工作日向App提供者说明本次热更新包更新的时间节点、热更新的具体内容、更新后可能造成的影响、热更新包的有效校验方式等；如果热更新内容涉及个人信息收集使用的目的、方式和范围的变更，安全性变更或重大的功能变更，进一步通过邮件、短信等逐一触达的方式告知App提供者；

3) 提供单独控制热更新功能开启关闭的选项，说明关闭热更新功能带来的影响，并保留App提供者在不接受热更新功能的情况下仍可正常使用SDK其他功能的权利。

h) 向App提供者告知SDK的相关信息，告知的信息应完整、准确、及时，不存在故意隐瞒、欺骗等行为。告知内容包括但不限于：SDK提供者的基本信息、沟通反馈渠道、安全能力；SDK的基本功能、版本号、隐私政策链接地址；申请的敏感权限和申请

目的；收集的个人信息类型和收集目的[7]；个人信息回传服务器所在地域；热更新机制及其开启关闭方式；是否存在单独收集用户个人信息的界面；嵌入的其他可收集个人信息的SDK；是否向其他应用或服务器发送、共享收集的用户个人信息；是否存在后台自启动和关联启动并收集个人信息的行为等。

i) 作为个人信息共同控制者或独立控制者收集使用用户个人信息的SDK，单独向用户告知收集使用个人信息的行为并征得用户同意。

j) 在保障安全的前提下，优先在本地的App私有存储空间内存储和处理个人信息。在本地存储和处理个人敏感信息，对个人敏感信息内容进行加密。

k)不留存不可变更的设备唯一标识符，收集可变更的标识符或采用技术手段将原始的不可变更标识符转化为可变更状态。

l) 建立响应个人信息主体请求和投诉等机制，并在接入App前及时告知App提供者相应的请求和投诉渠道，以供个人信息主体查询、使用。

m) 宜建立“Opt-out”选择退出机制，当个人信息主体不希望使用SDK提供的服务时，个人信息主体可通过“Opt-out”机制行使退出权利[8]。在官网或个人信息保护政策中透出“Opt-out”的链接，以便个人信息主体行使权利。

[7] 对于必须要申请的敏感权限或必须要收集的个人信息，应进一步说明其必要性。
[8] 例如，友盟+，终端设备 Opt-out。
https://outdip.umeng.com/opt_out.html?spm=a213m0.13887608.0.0.3cb275ef0jDEVu

n) 完善与App提供者的合作协议，明确SDK收集的个人信息类型、申请的敏感权限、个人信息的使用目的、保存期限、超期处理方式等，明确双方在个人信息保护方面分别应采取的措施、承担的责任和义务等。当双方合作存在重大变更时，应重新达成合作协议。

o) 当某App停止接入SDK后，若SDK提供者存在从该App共享或收集个人信息的，应按照合作协议约定，删除从该App共享或收集的个人信息或做匿名化处理。