ICS 35.240
CCS L 70/84

# T/ZISIA

团　　体　　标　　准

T/ZISIA 0103—2026

# 通用操作系统商用密码数字证书体系规范

Specification for commercial cipher digital certificate system of general-purpose operating systems

2026－03－31 发布　　2026－03－31 实施

中关村网络安全与信息化产业联盟　　发 布

# 目　　次

# 前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中关村网络安全与信息化产业联盟提出并归口。

本文件起草单位：中关村网络安全与信息化产业联盟、北京天威诚信电子商务服务有限公司、麒麟软件有限公司、中科方德软件有限公司、兴唐通信科技有限公司、阿里云计算有限公司、长春吉大正元信息技术股份有限公司、北京三博安科技有限公司、蚂蚁科技集团股份有限公司。

本文件主要起草人：王尧、李延昭、齐建新、张大朋、蒋杏松、王玉成、张超、郭亮、胡昆、卢合平、冯建茹、王希、王辉、姚长远、张天佳、李世奇、陈小春、王克、胡金山、罗捷。

# 通用操作系统商用密码数字证书体系规范

## 1 范围

本文件规定了通用操作系统（OS）商用密码数字证书体系的架构，以及启动垫片证书、内核代码证书、驱动模块证书、文件完整性证书、应用软件包证书、可执行程序证书及SSH证书的相关要求，包括证书的作用、证书签发主体和使用主体的职责、应用证书链、证书使用和管理。

本文件为OS商用密码数字证书体系的建设、管理和使用提供指导。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25069 信息安全技术 术语

GB/T 28447-2012 信息安全技术 电子认证服务机构运营管理规范

GB/T 33560 信息安全技术 密码应用标识规范

GM/T 0015 数字证书格式

GM/T 0034-2014 基于SM2密码算法的证书认证系统密码及其相关安全技术规范

GM/Z 4001-2013 密码术语

T/ZISIA 0101-2025 通用操作系统商用密码子系统安全轮廓

T/ZISIA 0105-2026 OS应用软件包签名/验证技术规范

## 3 术语和定义

GB/T 25069、GM/Z 4001-2013 界定的以及下列术语和定义适用于本文件。

3.1

**应用证书 application certificate**

直接由用户、服务器或设备持有，用于标识自身身份，如软件代码签名证书、HTTP网站服务器证书。

注：本文件的“应用证书”对应GM/T 0034-2014的“用户证书”。

3.2

**引导加载器 boot loader**

计算机启动过程中运行的一段程序，亦称引导代码，负责初始化计算机硬件设备、检测可启动的操作系统、将操作系统内核加载到内存并将系统控制权传递给操作系统内核，从而使操作系统能够正常启动和运行。

[来源：T/ZISIA 0101-2025，3.1]

3.3

**启动垫片 boot shim**

一种在操作系统启动过程中的过渡阶段运行的软件，位于BIOS/UEFI和操作系统内核之间，用于增强系统安全性，方便在同一台计算机上安装和启动多个操作系统。

[来源：T/ZISIA 0101-2025，3.2]

3.4 **证书 certificate**

关于实体的一种数据，该数据由认证机构的私钥或秘密密钥签发，并无法伪造。

[来源：GB/T 37092-2018，3.1]

3.5

**证书链 certificate chain**

操作系统用于验证数字证书合法性的机制，由一系列相互关联的数字证书按层级顺序组成的链条，从应用证书（application certificate）开始，到中间证书（intermediate certificate），追溯到根证书（root certificate），形成完整的信任验证路径。

3.6

**证书撤销列表 certificate revocation list（CRL）**

由证书认证机构（CA）签发并发布的被撤销证书的列表。
[来源：GM/Z 4001-2013，2.144]

3.7

**证书信任列表 certificate trust list**

OS（Linux）内核编译时预置的静态可信公钥证书集合，由OS厂商维护，包括根证书和中间证书，用于OS启动和运行阶段，验证内核模块（.ko）、固件文件等组件的签名合法性，这些证书在编译内核时被嵌入到内核镜像（vmlinuz）的特定段（如.rodata.certs），或存储在/etc/keys等内核指定路径，属于内核镜像的一部分，未经OS厂商、OS管理员或OS用户授权无法修改。

3.8

**证书认证机构 certification authority（CA）**

对数字证书进行全生命周期管理的实体。也称为电子认证服务机构。
[来源：GM/Z 4001-2013，2.145]

3.9

**中间 CA intermediate CA**

签发和管理应用证书的证书认证机构。

3.10

**中间证书 intermediate certificate**

位于应用证书与根证书之间的中间节点证书，由根CA签发，用于分担根证书的签发任务，提高证书管理的灵活性和安全性。

3.11

**在线证书状态协议 online certificate status protocol（OCSP）**

一种用于实时查询数字证书状态的网络协议，主要用于验证证书是否有效（如是否被吊销、过期或处于正常状态），可替代证书吊销列表（CRL）的方案。

3.12

**平台密钥 platform key（PK）**

计算机安全启动使用的一组公私钥对，用于对OS厂商及设备厂商的公钥（证书）进行签名，PK公钥存储在非易失性存储中，私钥由PK持有者保存。通常PK由计算机设备制造商拥有，如果一个组织需要完全掌控组织内的安全启动设备，该组织也可以持有PK。PK证书可由受信任的CA签发。

3.13

**根 CA root CA**

生成和管理根证书的证书认证机构。

3.14

**根证书 root certificate**

证书链的最顶层，由受信任的证书认证机构（CA）自签名生成，操作系统提供者根据其制定的安全规则，评估其可信任程度达到了一定水平后，加载成为“受信任的根证书”。OS厂商应根据不同应用的安全级别制定不同用途的根证书库的安全保护措施，并定期地对根证书实施安全审计，将已识别的具有风险的根证书从根证书仓库中移除。

3.15

**安全启动 secure boot**

一种确保计算机只运行可信软件的安全功能，主要目的是防止能够感染操作系统引导加载程序和系统固件的恶意软件。其工作原理是验证每个阶段的引导加载程序和操作系统内核是否具有有效的数字签名。

[来源：T/ZISIA 0101-2025，3.12]

## 4 符号和缩略语

下列缩略语适用于本文件。

BIOS：基本输入输出系统（Basic Input/Output System）

CA：证书认证机构（Certification Authority）

CRL：证书撤销列表（Certificate Revocation List）

IMA：完整性度量架构（Integrity Measurement Architecture）

KeK：密钥交换密钥（Key Exchange Key）

OCSP：在线证书状态协议（Online Certificate Status Protocol）

OS：操作系统（Operating System）

PK：平台密钥（Platform Key）

## 5 OS 数字证书

本文件所述OS数字证书是指OS运行所需要的数字证书，用以保障OS自身安全，如安全启动、代码运行、应用软件包安装及用户登录等过程的安全。应用系统使用的数字证书，如网上银行、电子邮件等系统使用的证书不属于本文件规范范围。

本文件根据上下文，将“证书”等同于“数字证书”。

所有OS证书应由受信任的证书认证机构（CA）的私钥进行签名。

本文件所规定的所有证书规范，仅适用于在实际信息系统中运行的OS，不涵盖处于开发阶段的OS。

## 6 OS 数字证书体系

OS数字证书体系由OS启动证书、内核态证书及用户态证书构成。

——OS 启动证书：对 OS 启动阶段运行的代码进行签名/验证的证书，包括启动垫片证书、内核代码证书。

——内核态证书：对 OS 内核态运行或使用的代码、文件进行签名/验证的证书，包括驱动模块证书和文件完整性证书。驱动模块证书用于对设备启动代码、内核模块（.ko）代码进行签名/验证，文件完整性证书用于对 OS 各类文件，如用户态可执行文件（ELF）、动态链接库（.so）、内核模块（.ko）及脚本（.sh）等文件进行签名/验证。

——用户态证书：在 OS 用户态对应用程序进行签名/验证的证书，包括应用软件包证书、可执行程序（如二进制工具、脚本）证书，以及 OS 与远程用户通信使用的 SSH 证书。

OS数字证书体系层次如图1所示。

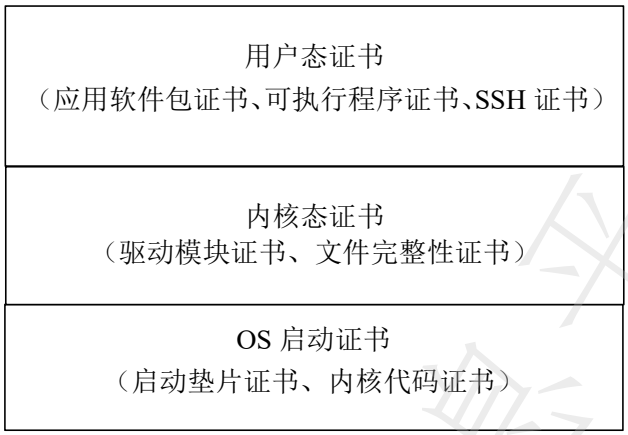



**图 1 OS 数字证书体系层次**

各应用证书使用机制见附录D。

## 7 OS 证书链

### 7.1 概述

OS证书一般以证书链的形式存在，证书链由根证书、中间证书和应用证书构成。

a) 根证书：所有者为根 CA，作为证书链的信任源头，根证书由根 CA 自签发（自签名）。根 CA 使用根证书私钥对中间证书进行数字签名，将根证书的信任度传递给中间证书。

注：根CA的根证书数据应采用电子与纸质双介质方式存储管理。采用硬件安全模块存储电子数据，将根证书数据显性记录在纸质介质上，并加盖该根证书所属组织的物理印章，作为该组织信任度、权威性通过数字证书传递至数字空间的法律凭证，且应执行严格物理管控，防止泄露。

b) 中间证书：所有者为中间 CA，由根 CA 或上一级中间 CA 签发，中间 CA 使用其中间证书对应的私钥对应用证书进行数字签名，将信任度传递给下级证书。中间证书可由多级组成。

c) 应用证书：直接对 OS 数据进行密码计算，以实现数据的完整性、不可抵赖性的安全功能，如启动垫片证书保证 OS 启动代码的完整性、来源可追溯性。应用证书的所有者为 CA 确认的合法组织。

d) OS 证书链验证：从应用证书开始，验证其签名是否由所声明的中间 CA 签发，再验证中间证书的签名是否由所声明的根 CA（或上级中间 CA）签发，最后用根证书自身的公钥验证其自签名；同时查验签发每个证书的 CA 提供的 CRL 或 OCSP，确保证书处于有效状态，若全部验证通过，系统判定该应用证书正确且有效，则证书链通过验证。当系统无法实时获得 CA 提供的 CRL 或访问 OCSP 时，应建立可行机制，定时更新本地存储的 CRL。

e) 除根证书外，OS 商用密码数字证书体系应避免使用自签名证书。

f) OS 应在安全存储区对证书链进行统一存储和全生命周期管理。

OS证书链签发与验证如图2所示。

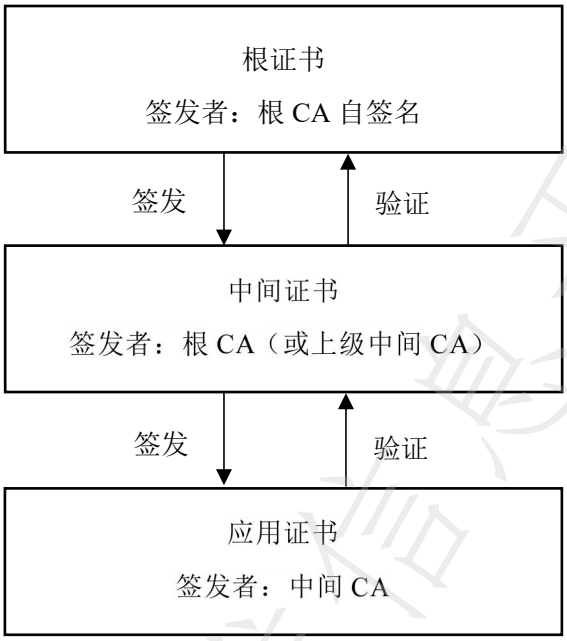



**图 2 OS证书链签发与验证**

OS厂商应建立根证书（含中间证书）的信任列表，并制定管理规则，明确根证书（含中间证书）进入信任列表或被移出信任列表的相关要求。

OS厂商应使用文件完整性证书对根证书（含中间证书）信任列表进行数字签名，保护信任列表的完整性，未经授权禁止修改。

OS部署的根证书宜采用安全芯片进行保护。

## 7.2 根证书规范

**使用主体：**

a) 根 CA 自身：根 CA 使用其根证书私钥向中间 CA 签发中间证书，将其信任度传递给中间 CA；
b) 用户：计算机厂商将根证书预置于计算机启动固件中，或由 OS 厂商将根证书存储在 OS 安全存储区中，在系统验证证书链时使用。

**签发主体：**

a) 国家根 CA：由国家密码主管部门许可并运营的电子认证根 CA；
b) OS 厂商（行业）根 CA：OS 厂商或 OS 行业组织运营的根 CA；
c) 企业根 CA：企业自行运营的根 CA。

**签发主体责任至少包括：**

a) 制定根证书管理策略，包括密钥更新周期和吊销条件；
b) 维护根证书吊销列表（CRL）及在线证书状态协议（OCSP）服务；
c) 确保根证书私钥的安全；
d) 对中间 CA 进行年度审查。

**验证主体：**

根证书由计算机启动固件、OS进行验证。

## 7.3 中间证书规范

**使用主体：**

a) 中间 CA：
- 用于签发下级中间证书。如果证书链中存在两级以上中间证书，上级中间CA使用其证书私钥签发下级中间证书，将信任度传递给下级证书；

- 用于签发应用证书。中间CA直接给应用系统用户签发应用证书；
- 中间CA的层级宜不超过3级，以保证证书链验证效率。

b） 用户：

- 用于证书链存储。将中间证书与应用证书一起存放使用。

c） OS：

- 将中间证书预置于OS信任存储区（证书信任列表）中，在系统验证证书链时使用。

**使用主体责任至少包括：**

a） 中间 CA 确保其证书私钥安全，防止泄露和伪造；
b） 中间 CA 按照规定的标准和流程签发下级证书，确保证书链的连续性和可信度，一旦发现证书存在问题，应及时进行处理，如撤销证书等；
c） 中间 CA 在证书签发和管理过程中，严格遵循国家的相关法律法规和行业政策，确保证书业务的合规性；
d） OS 预置中间证书至信任存储区。

**签发主体：**

根CA（或上级中间CA）。

**签发主体责任至少包括：**

a） 严格按照相关政策法规、标准规范签发中间证书，确保签发的中间证书的真实性和合法性，维护证书体系的可信度；
b） 对中间证书申请主体进行审核，包括身份信息、资质条件等，确保申请主体身份真实有效，防止非法组织或人员获取中间证书。

**验证主体：**

中间证书由计算机启动固件、OS在验证证书链时进行验证。

### 7.4 应用证书

**证书种类：**

OS商用密码应用证书包括：启动垫片证书、内核代码证书、驱动模块证书、文件完整性证书、应用软件包证书、可执行程序证书及SSH证书。各类证书使用主体、使用主体责任、签发主体、签发主体责任及授权等规范在第8章中给出。

**证书管理：**

OS应用证书使用主体除遵照第8章规范外，还应：

a） 建立 OS 证书管理制度及使用策略，并在系统运行中严格执行；
b） 避免同一个或同一类证书作为不同类型的应用证书使用；
c） 使用符合附录 C 的代码签名证书私钥对 OS 代码进行签名，避免使用非代码签名证书私钥对 OS 代码进行签名。

**证书撤销：**

应用证书撤销包括以下事项：

a） 证书撤销触发条件包括但不限于：

- 证书私钥泄露或者疑似泄露；
- 证书信息不准确或者发生重大变更；
- 证书使用主体违反证书策略；
- 证书使用主体停止运营或者失去资质；
- 其他影响证书可信度的情况。

b) 证书撤销主体：

- 证书使用主体；
- 证书签发主体；
- 监管机构。

c） 证书撤销处理步骤：

1） 证书签发主体接收并验证证书撤销申请；
2） 调查撤销原因；

3) 决策是否撤销证书；
4) 更新 CRL 或者 OCSP；
5) 记录撤销日志；
6) 通知相关方。

d) 证书撤销时效：
- 紧急撤销（私钥泄露）：24小时内完成；
- 一般撤销：7日内完成。

# 8 OS 应用证书规范

## 8.1 启动垫片证书

### 8.1.1 证书的作用

OS启动垫片代码责任主体（OS厂商）使用启动垫片证书对OS启动垫片代码进行数字签名，在计算机启动时对垫片签名进行验证，保证代码的完整性及来源可追溯。

### 8.1.2 证书使用主体及责任

**证书使用主体：**

a) OS 启动垫片代码的责任者（OS 厂商）：对垫片进行签名；
b) 计算机启动固件：对垫片签名进行验证。

**证书使用主体责任至少包括：**

a) 对 OS 启动垫片代码兼容性、安全性、合规性负责；
b) 向 CA 申请证书时，应保证向 CA 提交的主体身份、相关信息的真实可信，提交书面的材料，声明遵从 CA 机构所颁发的证书策略以及证书使用用途的限制；
c) 妥善保管启动垫片证书私钥，及时更新过期的证书；
d) 保证证书有效使用，杜绝非法使用和滥用。

### 8.1.3 证书签发主体、责任及授权

**启动垫片证书签发主体包括但不限于以下中间CA：**

——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业开发软件代码的主体签发代码签名证书。在资质要求上，应具备一定的技术实力和安全保障措施，确保证书颁发系统的稳定性和安全性；在监管要求方面，企业内部需建立相应的管理制度，对证书颁发过程进行监督和管理；在能力和技术要求上，要能够满足企业内部代码签名的需求，具备对企业内部人员身份验证和代码管理的能力。

**启动垫片证书签发主体责任至少包括：**

a) 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；
b) 严格审核证书申请者身份、资质、背景等文件，防止高风险的申请人获取 OS 启动垫片证书，并根据证书申请人所提交的申请材料，颁发相应的 OS 启动垫片证书；
c) 对于因审核不严发生不良主体使用证书对 OS 启动垫片签名导致的事故负有直接责任；
d) 掌握启动垫片证书的作用及使用场景，对存在问题的证书进行撤销处理，及时更新和维护 CRL、OCSP，防止其继续被使用；
e) 对于因不了解启动垫片证书的作用及使用场景导致的证书使用事故负有间接责任；
f) CA 签发启动垫片证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；
g) 按规范设置证书有效期，有效期宜为 3～5 年。

**证书签发主体授权：**

企业（行业）CA 由企业（行业）自身授权，负责企业（行业）内部启动垫片证书的签发和管理。

### 8.1.4 启动垫片证书链

启动垫片证书由中间CA签发，中间证书由根CA签发。

当计算机系统启动时，计算机安全启动固件对启动垫片证书链进行验证，验证从启动垫片证书开始，依次验证中间证书，直至验证根证书，同时查验签发每个证书的CA提供的CRL或OCSP，确保证书处于有效状态，若全部通过验证，则判定该启动垫片证书正确且有效，再对启动垫片签名进行验证。

注：计算机启动固件中的KeK、PK可视为启动垫片证书链的中间证书和根证书。

#### 8.1.5 OS 运行时证书使用和管理

——启动垫片证书（链）由计算机厂商预置在计算机安全启动固件中，OS 厂商或授权用户可使用工具对安全启动固件中的启动垫片证书（链）进行配置。

——日常证书管理可遵照 T/ZISIA 0101-2025【要求 6.3-06】实施。

### 8.2 内核代码证书

#### 8.2.1 证书的作用

OS内核代码责任主体（OS厂商）使用内核代码证书对OS引导代码（引导加载器）及内核镜像进行数字签名，在计算机启动时对签名进行验证，保证OS内核代码的完整性及来源可追溯。

#### 8.2.2 证书使用主体及责任

**证书使用主体：**

——OS 引导代码、内核镜像的责任主体（OS 厂商）：使用内核代码证书私钥对 OS 引导代码、内核镜像进行签名；

——OS 启动垫片：使用内核代码证书对 OS 引导代码、内核镜像的签名进行验证。

**证书使用主体责任至少包括：**

a) 对 OS 引导代码（引导加载器）及内核镜像的可靠性、安全性、合规性负责；

b) 向 CA 申请证书时，保证向证书签发 CA 提交的主体身份、相关信息的真实可信；并提交书面的材料，声明遵从 CA 机构所颁发的证书策略，以及证书使用用途的限制；

c) 妥善保管内核代码证书私钥，及时更新过期的证书；

d) 保证证书有效使用，杜绝非法使用和滥用。

#### 8.2.3 证书签发主体、责任及授权

**内核代码证书签发主体包括但不限于以下中间CA：**

——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业开发软件代码的主体签发代码签名证书。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）。

**内核代码证书签发主体责任至少包括：**

a) 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；

b) 严格审核证书申请者身份、资质、背景等文件，防止高风险的申请人获取内核代码证书；并根据证书申请人所提交的申请材料，颁发相应的内核代码证书；

c) 对于因审核不严发生不良主体使用证书对 OS 内核代码签名导致的 OS 引导代码及内核镜像事故负有直接责任；

d) 掌握内核代码证书的作用及使用场景，对存在问题的证书进行撤销处理，及时更新和维护 CRL、OCSP 的可用性，防止其继续被使用；

e) 对于因不了解内核代码证书的作用及使用场景导致的 OS 引导代码及内核镜像事故负有间接责任；

f) 第三方的证书颁发机构签发内核代码证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；

g) 按规范设置证书有效期，有效期宜为 3～5 年。

**证书签发主体授权：**

——企业（行业）CA 由企业（行业）自身授权，负责企业（行业）内部内核代码证书的签发和管理。

#### 8.2.4 内核代码证书链

内核代码证书由中间CA签发，中间证书由根CA签发。

在计算机系统启动过程中，启动垫片对OS引导代码（引导加载器）签名证书链及签名进行验证，OS引导代码（引导加载器）对内核镜像签名证书链及签名进行验证。证书链验证从内核代码证书开始，依次验证中间证书，直至验证根证书，同时，应查验签发该证书的CA提供的CRL或OCSP，确保证书处于有效状态，如果全部通过验证，则判定该内核代码证书正确且有效，再对OS引导代码（引导加载器）签名及内核镜像签名进行验证。

#### 8.2.5 OS 运行时证书使用和管理

——内核代码证书（链）由 OS 厂商预置在启动垫片中。

——OS 厂商或授权用户可对内核代码证书（链）进行修改设置。

——日常证书管理可遵照 T/ZISIA 0101-2025【要求 6.3-06】实施。

### 8.3 驱动模块证书

#### 8.3.1 证书的作用

OS驱动模块（包括设备启动代码及内核模块）开发者、检测者或对OS驱动模块负有责任的组织对OS驱动模块进行签名；OS启动时，由启动固件和内核镜像对签名进行验证，保证OS驱动模块的完整性及来源可追溯。

#### 8.3.2 证书使用主体、责任及授权

**证书使用主体：**

a) OS 驱动模块（设备启动代码及内核模块）开发者、检测者或对 OS 驱动模块负有责任的组织：使用驱动模块证书私钥对 OS 驱动模块进行签名；

b) 计算机启动固件、内核镜像：使用驱动模块证书对签名进行验证。

**证书使用主体责任至少包括：**

a) 对 OS 驱动模块代码的可靠性、安全性、合规性负责；

b) 代码检测主体对检测结论负责；

c) 向 CA 申请证书时，保证向证书签发 CA 提交的主体身份、相关信息真实，并提交书面材料，声明遵从 CA 机构所颁发的证书策略以及证书使用用途的限制；

d) 妥善保管驱动模块证书私钥，及时更新过期的证书；

e) 保证证书有效使用，杜绝非法使用和滥用。

**证书使用主体授权：**

驱动模块证书使用主体向证书签发主体申请证书时，应由OS厂商或OS用户进行备案、审核、许可，并将相关证明材料提供给证书签发主体；必要时，需提交内核模块开发项目许可文件，经审核通过后方可获得驱动模块证书。

#### 8.3.3 证书签发主体、责任及授权

**驱动模块证书签发主体包括但不限于以下中间CA：**

——政务证书签发机构（政务 CA）：取得国家密码主管部门颁发的《电子政务电子认证服务机构资质证书》的机构；

——商业证书签发机构（商业 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构；

——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业开发软件代码的主体签发 OS 软件签名证书。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）。

**驱动模块证书签发主体责任至少包括：**

a) 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；

b) 严格审核证书申请主体的身份、资质、背景、OS 厂商或 OS 用户备案、许可等文件，评估申请流程安全风险，防止高风险的申请人获取证书，根据证书申请人所提交的申请材料，颁发相应的驱动模块证书；

c） 对于因审核不严格导致不良主体使用证书造成的 OS 驱动模块事故负有直接责任；
d） 掌握驱动模块证书的作用及使用场景，对存在问题的证书进行撤销处理，及时更新和维护证书撤销列表（CRL）及在线证书状态协议（OCSP），防止其继续被使用；
e） 对于因不了解驱动模块证书的作用及使用场景导致的驱动模块事故负有间接责任；
f） CA 签发驱动模块证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；
g） 按规范设置证书有效期，有效期宜为 1～3 年。

**证书签发主体授权：**

——政务 CA /商业 CA：签发驱动模块证书应经 OS 厂商、计算机硬件厂商、OS 用户审核、备案，以确保证书的权威性与安全性。
——企业（行业）CA：由企业（行业）自身授权，负责企业内部驱动模块证书的签发和管理。

### 8.3.4 驱动模块证书链

驱动模块证书由中间CA签发，中间证书由根CA签发。

在OS启动过程中，由启动固件和OS内核镜像对驱动模块（.ko文件）证书链进行验证。验证从驱动模块证书开始，依次验证中间证书，直至验证根证书，同时，应通过签发该证书的CA提供的CRL或OCSP查验该证书的有效性，如果全部通过验证，则判定该驱动模块证书正确且有效，再对驱动模块签名进行验证。

注：计算机启动固件中的KeK、PK可视为驱动模块（设备启动代码）证书链的中间证书和根证书。

### 8.3.5 驱动模块证书使用和管理

——计算机厂商将需固件启动的驱动模块证书链预置在计算机安全启动固件中；
——OS 厂商或 OS 系统管理员将非固件启动的驱动模块证书链设置在 OS 系统证书信任列表中；
——OS 厂商或授权用户可修改安全启动固件中的证书链设置；
——设置合理监控策略，定期检查内核日志和安全审计报告；
——日常证书管理可遵照 T/ZISIA 0101-2025【要求 6.3-06】实施。

## 8.4 文件完整性证书

### 8.4.1 证书的作用

对OS运行过程中被保护的文件进行数字签名和验证，保证相关文件在存储、加载和访问过程中的完整性和来源可追溯。

### 8.4.2 证书使用主体、责任及授权

**证书使用主体：**

a） OS 厂商、第三方软件开发者、发行者、企业 IT 安全团队：使用证书私钥进行签名；
b） OS 内核：使用证书进行签名验证。

**证书使用主体责任至少包括：**

a） 对其签名的文件完整性、安全性和合规性负责；
b） 在申请证书时，应向 CA 提交真实、完整、合法的身份信息、资质证明及相关备案材料，并书面承诺遵守证书策略及证书用途限制；
c） 妥善保管证书私钥，采取必要的技术和管理措施防止私钥泄露、滥用或被非法获取；
d） 在证书到期、私钥存在风险或使用主体资格发生变化时，及时更新或撤销证书。

**证书使用主体授权：**

文件完整性证书使用主体向CA申请证书应由OS厂商或OS用户进行备案、审核、许可，并将相关证明材料提供给CA，经审核通过后可获得文件完整性证书。

### 8.4.3 证书签发主体、责任及授权

**文件完整性证书签发主体包括但不限于以下中间CA：**

——电子政务证书签发机构（政务 CA）：取得国家密码主管部门颁发的《电子政务电子认证服务机构资质证书》的机构。

——商业证书签发机构（商业 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构。

——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业 IT 安全设施签发文件完整性证书。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）。

**文件完整性证书签发主体责任至少包括：**

a） 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；

b） 严格审核证书申请主体的身份、资质、背景、OS 厂商或 OS 用户备案、许可等文件，评估待签名文件类型对 OS 的安全风险，并根据证书申请人所提交的申请材料，防止高风险主体获得证书；

c） 对于因审核不严发生不良主体使用证书导致的文件完整性事故负有直接责任；

d） 掌握文件完整性证书的作用及使用场景，对存在问题的证书进行撤销处理，及时更新和维护 CRL、OCSP，防止其继续被使用；

e） 对于因不了解文件完整性证书的作用及使用场景导致的文件完整性事故负有间接责任；

f） CA 签发 OS 代码签名证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；

g） 按规范设置证书有效期，有效期宜为 1～3 年。

**证书签发主体授权：**

——政务 CA/商业 CA：应经 OS 厂商、计算机硬件厂商、OS 用户审核、备案。

——企业 CA 由企业（行业）自身授权。

### 8.4.4 文件完整性证书链

文件完整性证书由中间CA签发，中间证书由根CA签发。

当受保护的文件在加载、执行、访问或使用前，OS对文件完整性证书链进行验证，验证从文件完整性证书开始，依次验证中间证书，直至验证根证书，同时，应通过签发该证书的CA提供的CRL或OCSP查验该证书的有效性，如果全部通过验证，则判定该文件完整性证书正确且有效，再对受保护文件签名进行验证。

### 8.4.5 文件完整性证书使用和管理

——系统管理员将文件完整性证书链添加至 OS 商密子系统中；

——定期检查内核日志和安全审计报告；

——配置签名验证处置策略；

——遵照 T/ZISIA 0101-2025【要求 6.3-06】实施日常管理。

## 8.5 应用软件包证书

### 8.5.1 证书的作用

应用软件包证书用于由应用软件开发者、发行者或经授权的第三方主体，对应用软件包（如*.deb、*.rpm等格式）进行数字签名，保证软件包分发、传输及安装过程中的完整性和来源可追溯。

### 8.5.2 证书使用主体、责任及授权

**证书使用主体：**

a） 应用软件开发者、发行者、第三方检测者以及对软件负有责任的组织：使用应用软件包证书私钥对 OS 应用软件包进行签名；

b） OS：使用应用软件包证书对应用软件包的数字签名进行验证。

**证书使用主体责任至少包括：**

a） 软件开发、发行主体对其签名的应用软件的可靠性、安全性、合规性及功能完整性负责；

b） 软件检测主体对其检测结论负责；

c） 证书使用主体向CA申请应用软件包证书时，应保证所提交的主体身份信息、资质证明、软件来源声明等材料真实、准确、合法，并书面承诺遵守该CA的证书策略及使用限制；

d）使用主体应妥善保管证书私钥，采取必要的技术和管理措施防止私钥泄露、滥用或被非法获取；
e）使用主体应在证书到期、私钥存在风险或使用主体资格发生变化时，及时更新或撤销证书；
f）保证证书有效使用，杜绝非法使用和滥用。

**应用软件包证书使用主体授权：**

应用软件包证书使用主体应向证书签发主体提供由OS厂商和/或OS用户出具的备案、审核或许可证明材料。必要时，提交软件开发项目许可文件，经审核通过后可获得应用软件包证书。

### 8.5.3 证书签发主体、责任及授权

**应用软件包证书签发主体包括但不限于以下中间CA：**

——电子政务证书签发机构（政务 CA）：取得国家密码主管部门颁发的《电子政务电子认证服务机构资质证书》的机构。
——商业证书签发机构（商业 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构。
——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业开发软件代码的主体签发应用软件包签名证书。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）。

**应用软件包证书签发主体责任至少包括：**

a） 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；
b） 对于因审核不严发生不良主体使用证书造成的应用软件供应链安全事故负有直接责任；
c） 掌握应用软件包证书的作用及使用场景，及时更新和维护 CRL、OCSP，对存在问题的证书进行撤销处理，防止其继续被使用；
d） 对于因不了解应用软件包证书的作用及使用场景导致的应用软件事故负有间接责任；
e） CA 签发应用软件包证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；
f） 按规范设置证书有效期，有效期宜为 1～3 年。

**证书签发主体授权：**

——政务 CA /商业 CA 签发应用软件包证书应经 OS 厂商、OS 用户 审核、备案、许可，以确保证书的权威性与安全性。
——企业（行业）CA 由企业（行业）自身授权，负责企业内部应用软件包证书的签发和管理。

### 8.5.4 应用软件包证书链

应用软件包证书由中间CA签发，中间证书由根CA签发。

OS在安装应用软件包前，OS对应用软件包证书链进行验证，验证从应用软件包证书开始，依次验证中间证书，直至验证根证书，同时，应通过签发该证书的CA提供的CRL或OCSP查验该证书的有效性，如果全部通过验证，则判定该应用软件包证书正确且有效，再对应用软件包签名进行验证。

### 8.5.5 OS 运行时证书使用和管理

——系统管理员将应用软件包证书链添加至 OS 商密子系统中；
——设置合理管理策略，定期检查内核日志和安全审计报告；
——配置签名验证处置策略；
——遵照 T/ZISIA 0101-2025【要求 6.3-06】的要求实施日常管理。

## 8.6 可执行程序证书

### 8.6.1 证书的作用

可执行程序证书用于对OS用户态可执行程序（包括但不限于二进制可执行文件、脚本文件及其相关组件）进行数字签名，在程序执行前验证签名，保证可执行程序的完整性及来源可追溯。

### 8.6.2 证书使用主体、责任及授权

**证书使用主体：**

——OS 厂商、第三方软件开发者、企业内部应用开发团队使用可执行程序证书私钥对可执行程序进行签名；

——OS 使用可执行程序证书对可执行程序的签名进行验证。

**证书使用主体责任至少包括：**

a) 对其签名的可执行程序在功能可靠性、安全性和合规性负责；

b) 向 CA 申请证书时，应保证提交真实、完整、合法的主体身份信息、资质材料及相关备案文件，承诺遵守证书策略及证书用途限制；

c) 妥善保管可执行程序证书私钥，采取必要的技术和管理措施防止私钥泄露、滥用或被非法获取；

d) 在证书到期、私钥存在安全风险或主体资格发生变化时，及时更新或申请撤销证书；

e) 保证证书有效使用，杜绝非法使用和滥用。

**可执行程序证书使用主体授权：**

可执行程序证书主体在向证书签发主体申请证书前，应取得OS厂商或OS使用者的备案、审核或许可，并应将相关授权或证明材料提交给证书签发主体。必要时，提交软件开发项目许可文件，经审核通过后可获得可执行程序证书。

### 8.6.3 证书签发主体、责任及授权

**可执行程序证书签发主体包括但不限于以下中间CA：**

——政务证书签发机构（政务 CA）：取得国家密码主管部门颁发的《电子政务电子认证服务机构资质证书》的机构；

——商业证书签发机构（商业 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构；

——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为本企业开发软件代码的主体签发可执行程序证书。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）的要求。

**可执行程序证书签发主体责任至少包括：**

a) 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；

b) 严格审核申请主体身份、资质、备案文件，评估可执行文件的安全风险，防止高风险主体获证；

c) 对审核不严导致的软件供应链安全事故负直接责任；

d) 掌握证书作用及场景，及时撤销问题证书，维护 CRL/OCSP；

e) 对不了解证书使用场景导致的软件供应链事故负间接责任；

f) CA 签发可执行程序证书，应满足相关标准（GM/T AAAA）要求，将证书“扩展密钥用法”字段设置为“代码签名”；

g) 按规范设置证书有效期，有效期宜为 1～3 年。

**证书签发主体授权：**

——政务/商业 CA 签发可执行程序证书，应经 OS 厂商或相关监管主体的审核、备案或许可。

——企业（行业）CA 应由企业（行业）自身授权，负责企业内部可执行程序证书的签发和管理。

### 8.6.4 可执行程序证书链

可执行程序证书应由中间CA签发，中间证书应由根CA签发。

OS在执行可执行程序前，OS对可执行程序证书链进行验证。从可执行程序证书开始，依次验证中间证书，直至验证根证书，同时，应通过签发该证书的CA提供的CRL或OCSP查验该证书的有效性，如果全部通过验证，则判定该可执行程序证书正确且有效，再对可执行程序签名进行验证。

### 8.6.5 OS 运行时证书使用和管理

——系统管理员将可信任的可执行程序证书链添加至 OS 商密子系统中；

——定期检查内核日志和安全审计报告；

——配置签名验证处置策略；

——遵照 T/ZISIA 0101-2025【要求 6.3-06】实施日常管理。

## 8.7 SSH 证书

### 8.7.1 证书的作用

SSH证书用于对SSH用户进行身份认证，建立加密通信通道，保证远程登录、文件传输等操作过程中的身份真实性、数据机密性与完整性。

### 8.7.2 证书使用主体、责任及授权

**证书使用主体：**

网站运营主体、企业内部服务器、云服务提供商的系统管理员或者运维人员。

**证书使用主体责任至少包括：**

a) 对 SSH 通信的服务器的真实性、用户的合法性负责，确保证书使用符合组织安全策略；
b) 申请证书时，应保证向证书签发机构提交的主体身份信息、服务器信息（如域名、IP 地址）、用户身份信息真实、准确、完整，并承诺遵守 CA 的证书策略及使用限制；
c) 应妥善保管 SSH 证书私钥，采取技术与管理措施防止私钥泄露、非法复制或滥用；
d) 在证书到期、私钥存在风险、服务器停用或用户权限变更时，应及时更新或撤销证书；
e) 正确配置服务器软件，不同的 SSH 服务器软件对证书的配置方式有所不同，管理员需熟悉服务器软件的配置方法，确保证书在通信过程中能正常发挥作用。

**证书使用主体授权：**

SSH证书使用主体应向证书签发主体提供服务器的相关证明材料，如域名所有权证明（可通过域名注册机构提供的相关文件来证明）、企业注册信息（包括企业的营业执照、组织机构代码证等）、经办人信息和授权经办人办理SSH证书的证明文件等，经审核通过后获得SSH证书。

### 8.7.3 证书签发主体、责任及授权

**SSH证书签发主体包括但不限于以下中间CA：**

——电子政务证书签发机构（政务 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构；
——商业证书签发机构（商业 CA）：取得国务院信息产业主管部门颁发的《电子认证服务机构资质证书》的机构；
——企业（行业）证书签发机构（企业 CA）：企业自行搭建的 CA 系统，为企业内部服务器签发证书，用于内部网络环境。企业 CA 可根据企业的特定需求和安全策略进行定制化配置，便于管理和控制。例如，企业可对内部服务器证书的有效期、使用范围等进行灵活设置。企业 CA 资质、监管及能力要求同 8.1.3——企业（行业）证书签发机构（企业 CA）。

**SSH 证书签发主体责任至少包括：**

a) 遵照 GM/T 0034-2014 规范及国家相关法规建设、运行和管理证书认证系统；
b) 严格审核申请主体服务器身份信息，防止非法服务器获证；
c) 按行业标准签发证书，确保证书内容准确合规；
d) 及时更新和维护 CRL、OCSP，对存在问题的证书进行撤销处理，如服务器被入侵、证书私钥泄露等情况，及时将证书列入撤销列表，防止其继续被使用；
e) 根据系统安全策略设置证书有效期，一般不超过 1 年。

**证书签发主体授权：**

——政务 CA/商业 CA 签发 SSH 证书，应获得 OS 厂商的许可或备案；
——OS 企业（行业）CA 由 OS 企业或 OS 行业组织自身授权，负责 OS 内部 SSH 证书的签发与管理。

### 8.7.4 SSH 证书链

SSH证书由中间CA签发，中间证书由根CA签发。

在验证SSH证书时，对其证书链进行验证，验证从SSH证书开始，依次验证中间证书，直至根证书，同时，应通过签发该证书的CA提供的CRL或OCSP查验该证书的有效性，如果全部通过验证，则判定该SSH证书正确且有效。

#### 8.7.5 OS 运行时证书使用和管理

——系统管理员应将受信任的 SSH CA 根证书及中间证书预置在 OS 或 SSH 客户端的信任存储中；
——服务器管理员编辑 SSH 服务器软件配置文件，完成证书安装配置；
——定期检查证书有效期并更新，企业环境可通过集中管理工具统一配置；
——日常证书管理可遵照 T/ZISIA 0101-2025【要求 6.3-06】实施。

# 附 录 A
（规范性）
OS 证书链表

表A.1列出OS证书链证书类型、使用主体、签发主体及其责任。

表 A.1 OS 证书链相关主体及责任表

| 主体 | 根证书 | 中间证书 | 应用证书 |
| --- | --- | --- | --- |
| 使用主体 | 1. 根 CA<br>2. OS 厂商及用户 | 1. 中间 CA<br>2. 用户<br>3. OS | 1. 计算机厂商<br>2. OS 厂商<br>3. 设备厂商<br>4. 软件责任主体<br>5. 用户 |
| 使用主体责任 | 1. 根 CA：确保根证书私钥安全，更新密钥、处理吊销，对中间 CA 年度审查<br>2. OS 厂商及使用者：评估根证书信任度，预置可信根证书，定期审计根证书库，移除高风险根证书 | 1. 中间 CA：确保自身证书私钥安全；<br>2. 按标准流程签发下级证书，保障证书链连续性与可信度，及时撤销问题证书；<br>3. 遵循国家法律法规与行业政策，确保业务合规<br>4. OS：妥善预置中间证书至信任存储区，配合证书链验证流程 | 1. 确保应用证书私钥安全，防止泄露与伪造<br>2. 严格按证书使用范围使用，杜绝非法使用与滥用<br>3. 妥善保管证书，防止非法获取<br>4. 及时更新过期证书 |
| 签发主体 | 1. 国家根 CA<br>2. OS 厂商（行业）根 CA<br>3. 企业根 CA | 根 CA（或上级中间 CA） | 中间 CA |
| 签发主体责任 | 1. 制定根证书管理策略，明确密钥更新周期和吊销条件<br>2. 维护根证书吊销列表（CRL）及在线证书状态协议（OCSP）服务<br>3. 确保根证书私钥安全<br>4. 对中间 CA 进行年审 | 1. 严格按照相关政策法规、标准规范签发中间证书，确保证书真实性和合法性<br>2. 审核中间证书申请主体的身份信息、资质，防止非法组织或人员获取证书 | 1. 严格审核应用证书申请主体的身份和资质，确保证明材料真实有效<br>2. 严格划定应用证书使用场景及业务范围，防止滥发、滥用<br>3. 遵循国家相关法律法规和行业政策，确保证书业务合规<br>4. 制定应用证书管理策略，明确密钥更新周期和吊销条件<br>5. 维护 CRL 及 OCSP 服务<br>6. 保障证书链连续性和可信度，及时撤销存在问题的证书 |

# 附 录 B
（规范性）
OS 应用证书体系规范表

表B.1列出OS应用证书及属性规范。

表 B.1 OS 应用证书及属性规范

| 属性 | 启动垫片证书 | 内核代码证书 | 驱动模块证书 | 文件完整性证书 | 应用软件包证书 | 可执行程序证书 | SSH 证书 |
|---|---|---|---|---|---|---|---|
| 作用 | OS 启动垫片代码责任主体使用该证书对启动垫片代码进行数字签名，计算机启动时验证垫片签名，保证代码完整性及来源可追溯 | OS 内核代码责任主体使用该证书对 OS 引导代码（引导加载器）及内核镜像进行数字签名，计算机启动时验证签名，保证 OS 内核代码完整性及来源可追溯 | OS 驱动模块开发者、检测者或责任组织对驱动模块进行签名，OS 启动时由启动固件和内核镜像验证签名，保证驱动模块完整性及来源可追溯 | 对 OS 运行过程中被保护的文件进行数字签名和验证，保证文件在存储、加载和访问过程中的完整性和来源可追溯 | 由应用软件开发者、发行者或经授权的第三方主体，对应用软件包进行数字签名，保证软件包分发、传输及安装过程中的完整性和来源可追溯 | 对 OS 用户态可执行程序（包括二进制可执行文件、脚本文件及其相关组件）进行数字签名，在程序执行前验证签名，保证可执行程序的完整性及来源可追溯 | 对 SSH 用户进行身份认证，建立加密通信通道，保证远程登录、文件传输等操作过程中的身份真实性、数据机密性与完整性 |
| 使用主体 | 1. OS 启动垫片代码的责任者（OS 厂商）；2. 计算机启动固件 | 1. OS 引导代码、内核镜像的责任主体（OS 厂商）；2. OS 启动垫片 | 1. OS 驱动模块（设备启动代码及内核模块）开发者、检测者或责任组织；2. 计算机启动固件、内核镜像 | 1. OS 厂商、第三方软件开发者、发行者、企业 IT 安全团队；2. OS 内核 | 1. 应用软件开发者、发行者、第三方检测者以及对软件负有责任的组织；2. OS | 1. OS 厂商、第三方软件开发者、企业内部应用开发团队；2. OS | 网站运营主体、企业内部服务器管理员、云服务提供商等 |
| 使用主体责任 | 1. 对OS启动垫片代码兼容性、安全性、合规性负责；2. 向CA申请证书时，保证提交的主体身份及相关信息真实可信，提交书面材料声明遵从 CA 证书策略及使用用途限制；3. 妥善保管私钥，及时更新过期证书；4. 保证证书有效使用，杜绝非法使用和滥用 | 1. 对OS引导代码及内核镜像的可靠性、安全性、合规性负责；2. 向 CA 申请证书时，保证提交的主体身份及相关信息真实可信，提交书面材料声明遵从CA证书策略及使用用途限制；3. 妥善保管私钥，及时更新过期证书；4. 保证证书有效使用，杜绝非法使用和滥用 | 1. 对OS驱动模块代码的可靠性、安全性、合规性负责；2. 代码检测主体对检测结论负责；3. 向 CA 申请证书时，保证提交的主体身份及相关信息真实，提交书面材料声明遵从 CA 证书策略及使用用途限制；4. 妥善保管私钥，及时更新过期证书；5. 保证证书有效使用，杜绝非法 | 1. 对其签名的文件在完整性、安全性和合规性负责；2. 申请证书时，提交真实、完整、合法的身份信息、资质证明及相关备案材料，并书面承诺遵守证书策略及用途限制；3. 妥善保管私钥，采取技术和管理措施防止私钥泄露、滥用或被非法获取；4. 证书到期、私钥存在风险或 | 1. 软件开发、发行主体对其签名的应用软件的可靠性、安全性、合规性及功能完整性负责；2. 软件检测主体对其检测结论负责；3. 申请证书时，保证提交的主体身份信息、资质证明、软件来源声明等材料真实、准确、合法，并书面承诺遵守 CA 的证书策略及使用限制；4. 妥善 | 1. 对其签名的可执行程序在功能可靠性、安全性和合规性负责；2. 申请证书时，提交真实、完整、合法的主体身份信息、资质材料及相关备案文件，承诺遵守证书策略及用途限制；3. 妥善保管私钥，采取技术和管理措施防止私钥泄露、滥用或被非法获取；4. 证书到期、私钥存 | 1. 对 SSH 通信的服务器真实性、用户合法性负责，确保证书使用符合组织安全策略；2. 申请证书时，保证提交的主体身份信息、服务器信息（如域名、IP 地址）、用户身份信息真实、准确、完整，并承诺遵守 CA 的证书策略及使用限制；3. 妥善保管私钥，采取技术与管理措施防 |

| 属性 | 启动垫片证书 | 内核代码证书 | 驱动模块证书 | 文件完整性证书 | 应用软件包证书 | 可执行程序证书 | SSH 证书 |
|---|---|---|---|---|---|---|---|
| | | | 使用和滥用 | 主体资格变化时，及时更新或撤销证书 | 保管私钥，采取技术和管理措施防止私钥泄露、滥用或被非法获取；5. 证书到期、私钥存在风险或主体资格变化时，及时更新或撤销证书；6. 保证证书有效使用，杜绝非法使用和滥用 | 在安全风险或主体资格变化时，及时更新或申请撤销证书；5. 保证证书有效使用，杜绝非法使用和滥用 | 止私钥泄露、非法复制或滥用；4. 证书到期、私钥存在风险、服务器停用或用户权限变更时，及时更新或撤销证书；5. 正确配置服务器软件，确保证书在通信过程中正常发挥作用 |
| 使用主体授权 | 无额外授权要求，责任主体直接申请使用 | 无额外授权要求，责任主体直接申请使用 | 向证书签发主体申请证书时，应由OS厂商或OS用户进行备案、审核、许可，并提供相关证明材料，必要时提交内核模块开发项目许可文件，审核通过后方可获得证书 | 向CA申请证书应由OS厂商或OS用户进行备案、审核、许可，并提供相关证明材料，审核通过后可获得证书 | 向证书签发主体提供由OS厂商和/或OS用户出具的备案、审核或许可证明材料，必要时提交软件开发项目许可文件，审核通过后可获得证书 | 向证书签发主体申请证书前，应取得OS厂商或OS使用者的备案、审核或许可，并提交相关授权或证明材料，必要时提交软件开发项目许可文件，审核通过后可获得证书 | 向证书签发主体提供服务器相关证明材料（如域名所有权证明、企业注册信息、经办人信息及授权证明文件等），审核通过后获得证书 |
| 签发主体 | 企业（行业）证书签发机构（企业CA） | 企业（行业）证书签发机构（企业CA） | 1. 政务证书签发机构（政务CA）；2. 商业证书签发机构（商业CA）；3. 企业（行业）证书签发机构（企业CA） | 1. 电子政务证书签发机构（政务CA）；2. 商业证书签发机构（商业CA）；3. 企业（行业）证书签发机构（企业CA） | 1. 电子政务证书签发机构（政务CA）；2. 商业证书签发机构（商业CA）；3. 企业（行业）证书签发机构（企业CA） | 1. 政务证书签发机构（政务CA）；2. 商业证书签发机构（商业CA）；3. 企业（行业）证书签发机构（企业CA） | 1. 电子政务证书签发机构（政务CA）；2. 商业证书签发机构（商业CA）；3. 企业（行业）证书签发机构（企业CA） |
| 签发主体责任 | 1. 遵照GM/T 0034-2014规范及国家相关法规建设、运行和管理证书认证系统；2. 严格审核申请者身份、资质、背景等文件，防止高风险申 | 1. 遵照GM/T 0034-2014规范及国家相关法规建设、运行和管理证书认证系统；2. 严格审核申请者身份、资质、背景等文件，防止高风险申 | 1. 遵照GM/T 0034-2014规范及国家相关法规建设、运行和管理证书认证系统；2. 严格审核申请主体身份、资质、背景、备案许可等文件，评 | 1. 遵照GM/T 0034-2014规范及国家相关法规建设、运行和管理证书认证系统；2. 严格审核申请主体身份、资质、背景、备案许可等文件，评 | 1. 遵照GM/T 0034-2014规范及国家相关法规建设、运行和管理证书认证系统；2. 对因审核不严导致的应用软件供应链安全事故负直接 | 1. 遵照GM/T 0034-2014及国家法规建设、运行和管理认证系统；2. 严格审核申请主体身份、资质、备案文件，评估可执行文件的安全风险， | 1. 遵照GM/T 0034-2014及国家法规建设、运行和管理认证系统；2. 严格审核申请主体服务器身份信息，防止非法服务器获证；3. 按行业标 |

| 属性 | 启动垫片证书 | 内核代码证书 | 驱动模块证书 | 文件完整性证书 | 应用软件包证书 | 可执行程序证书 | SSH 证书 |
|---|---|---|---|---|---|---|---|
| | 请人获取证书；3. 对因审核不严导致的事故负直接责任；4. 掌握证书作用及使用场景，及时撤销问题证书，维护CRL、OCSP；5. 对因不了解证书使用场景导致的事故负间接责任；6. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；7.设置证书有效期，3~5 年 | 请人获取证书；3. 对因审核不严导致的OS 引导代码及内核镜像事故负直接责任；4. 掌握证书作用及使用场景，及时撤销问题证书，维护CRL、OCSP 可用性；5. 对因不了解证书使用场景导致的事故负间接责任；6. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；7.设置证书有效期，3~5 年 | 估申请流程安全风险，防止高风险申请人获取证书；3. 对因审核不严导致的驱动模块事故负直接责任；4. 掌握证书作用及使用场景，及时撤销问题证书，维护CRL、OCSP；5. 对因不了解证书使用场景导致的事故负间接责任；6. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；7.设置证书有效期，1~3 年 | 估待签名文件类型对OS 的安全风险，防止高风险主体获得证书；3. 对因审核不严导致的文件完整性事故负直接责任；4. 掌握证书作用及使用场景，及时撤销问题证书，维护CRL、OCSP；5. 对因不了解证书使用场景导致的事故负间接责任；6. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；7.设置证书有效期，1~3 年 | 责任；3. 掌握证书作用及使用场景，及时更新和维护CRL、OCSP，撤销问题证书；4. 对因不了解证书使用场景导致的应用软件事故负间接责任；5. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；6.设置证书有效期，1~3 年 | 防止高风险主体获证；3. 对审核不严导致的软件供应链安全事故负直接责任；4. 掌握证书作用及场景，及时撤销问题证书，维护CRL/OCSP；5. 对不了解证书使用场景导致的软件供应链事故负间接责任；6. 满足相关标准要求，将证书“扩展密钥用法”字段设为“代码签名”；7.设置证书有效期，1~3 年 | 准签发证书，确保证书内容准确合规；4. 及时更新和维护CRL、OCSP，撤销问题证书（如服务器被入侵、证书私钥泄露等情况）；5.设置证书有效期，1 年 |
| 签发主体授权 | 企业（行业）CA 由企业（行业）自身授权，负责内部启动垫片证书的签发和管理 | 企业（行业）CA 由企业（行业）自身授权，负责内部内核代码证书的签发和管理 | 1. 政务 CA / 商业 CA：签发证书应经 OS 厂商、计算机硬件厂商、OS 用户审核、备案；2. 企业（行业）CA：由企业（行业）自身授权，负责内部驱动模块证书的签发和管理 | 1. 政务 CA / 商业 CA：应经 OS 厂商、计算机硬件厂商、OS 用户审核、备案；2. 企业 CA：由企业（行业）自身授权 | 1. 政务 CA / 商业 CA：签发证书应经 OS 厂商、OS 用户审核、备案、许可；2. 企业（行业）CA：由企业（行业）自身授权，负责内部应用软件包证书的签发和管理 | 1. 政务 / 商业 CA：签发证书应经 OS 厂商或相关监管主体的审核、备案或许可；2. 企业（行业）CA：由企业（行业）自身授权，负责内部可执行程序证书的签发和管理 | 1. 政务 CA / 商业 CA：签发 SSH 证书应获得 OS 厂商的许可或备案；2. OS 企业（行业）CA：由 OS 企业或 OS 行业组织自身授权，负责 OS 内部 SSH 证书的签发与管理 |
| 证书链 | 启动垫片证书由中间 CA 签发，中间证书由根 CA 签发<br>计算机启动固件中的 KeK、PK 可视为启动垫 | 内核代码证书由中间 CA 签发，中间证书由根 CA 签发<br>计算机启动固件中的 KeK、PK 可视为驱动模 | 驱动模块证书由中间 CA 签发，中间证书由根 CA 签发 | 文件完整性证书由中间 CA 签发，中间证书由根 CA 签发 | 应用软件包证书由中间 CA 签发，中间证书由根 CA 签发 | 可执行程序证书由中间 CA 签发，中间证书由根 CA 签发 | SSH 证书由中间 CA 签发，中间证书由根 CA 签发 |

| 属性 | 启动垫片证书 | 内核代码证书 | 驱动模块证书 | 文件完整性证书 | 应用软件包证书 | 可执行程序证书 | SSH 证书 |
|---|---|---|---|---|---|---|---|
| | 片证书链的中间证书、根证书 | 块（设备启动代码）证书链的中间证书和根证书 | | | | | |
| 运行时管理 | 1. 由计算机厂商预置在计算机安全启动固件中，OS 厂商或授权用户可使用工具配置；2. 日常管理遵照 T/ZISIA 0101-2025【要求 6.3-06】实施 | 1. 由 OS 厂商预置在启动垫片中；2. OS 厂商或授权用户可对证书链进行修改设置；3. 日常管理遵照 T/ZISIA 0101-2025【要求 6.3-06】实施 | 1. 计算机厂商将需固件启动的驱动模块证书链预置在计算机安全启动固件中；2. OS 厂商或系统管理员将非固件启动的驱动模块证书链设置在 OS 系统证书信任列表中；3. OS 厂商、用户可修改安全启动固件中的证书链设置；4. 设置合理监控策略，定期检查内核日志和安全审计报告；5. 设置合理有效期（如 3 年）；6. 日常管理遵照 T/ZISIA 0101-2025【要求 6.3-06】实施 | 1. 系统管理员将文件完整性证书链添加至 OS 商密子系统中；2. 定期检查内核日志和安全审计报告；3. 配置签名验证处置策略；4. 遵照 T/ZISIA 0101-2025【要求 6.3-06】实施日常管理 | 1. 系统管理员将应用软件包证书链添加至 OS 商密子系统中；2. 设置合理管理策略，定期检查内核日志和安全审计报告；3. 配置签名验证处置策略；4. 遵照 T/ZISIA 0101-2025【要求 6.3-06】实施日常管理 | 1. 系统管理员将可信任的可执行程序证书链添加至 OS 商密子系统中；2. 定期检查内核日志和安全审计报告；3. 配置签名验证处置策略；4. 遵照 T/ZISIA 0101-2025【要求 6.3-06】实施日常管理 | 1. 系统管理员将受信任的 SSH CA 根证书及中间证书预置在 OS 或 SSH 客户端的信任存储中；2. 服务器管理员编辑服务器软件配置文件，完成证书安装配置；3. 定期检查证书有效期并更新，企业环境可通过集中管理工具统一配置；4. 日常管理遵照 T/ZISIA 0101-2025【要求 6.3-06】实施 |

# 附 录 C
（规范性）
OS 签名证书格式

遵照GM/T 0015证书格式规范，将密钥用法（Key Usage）字段设置成digitalSignature、nonRepudiation，证书扩展密钥用途（Extended Key Usage）字段设置成id-kp-codeSigning，表示证书公钥用于代码签名；密钥用法（Key Usage）设置成keyEncipherment，证书扩展密钥用途（Extended Key Usage）字段设置成id-kp-serverAuth、id-kp-clientAut，表示证书公钥用于SSH通信。遵照GB/T 33560，将证书策略（Certificate Policies）字段分别设置为OS启动垫片证书、内核代码证书、驱动模块证书、文件完整性证书、可执行程序证书、应用软件包证书及SSH证书的策略OID。OS签名证书格式如下：

| 字段 | 内容 |
|---|---|
| Version 版本 | Version 3 |
| Serial Number 序列号 | 最大20字节的正整数 |
| Signature Algorithm 签名算法 | 1.2.156.10197.1.501<br>SM3WithSM2Encryption |
| Issuer DN 颁发者辨别名 | 由CPS的定义决定 |
| Validity Period 有效期 | 由CPS的定义决定 |
| Subject DN 主体辨别名 | 由CPS的定义决定 |
| Subject Public Key 主体公钥 | 1.2.156.10197.1.301 SM2密码算法 |
| Key Usage 密钥用法 | critical,<br>keyUsage: digitalSignature, nonRepudiation,<br>keyEncipherment |
| Extended Key Usage 扩展密钥用法 | 1.3.6.1.5.5.7.3.3 codeSigning<br>1.3.6.1.5.5.7.3.3 serverAuth<br>1.3.6.1.5.5.7.3.3 clientAuth |
| Basic Constraints 基本限制 | critical,CA:FALSE |
| Subject Key Identifier 主体密钥标识符 | 可选 |
| Authority Key Identifier 颁发机构密钥标识符 | 可选 |
| Certificate Policies 证书策略 | GB/T 33560给出的策略OID：<br>1.2.156.10197.6.4.1.3.1 启动垫片证书<br>1.2.156.10197.6.4.1.3.2 内核代码证书<br>1.2.156.10197.6.4.1.3.3 驱动模块证书<br>1.2.156.10197.6.4.1.3.4 文件完整性证书<br>1.2.156.10197.6.4.1.3.5 可执行程序证书<br>1.2.156.10197.6.4.1.3.6 应用软件包证书<br>1.2.156.10197.6.4.1.3.7 SSH证书 |

# 附 录 D
（规范性）
OS 应用证书使用机制

## D.1 启动垫片证书、OS 内核代码证书及驱动模块证书使用机制

### D.1.1 签名及证书预置

**签名：**

——计算机厂商或设备所有者使用平台密钥（PK）私钥对密钥交换密钥（KeK）证书进行签名，建立计算机系统安全信任根。

——OS 厂商使用 KeK 证书私钥对启动垫片证书及驱动模块证书进行签名。

——OS 厂商使用启动垫片证书私钥对启动垫片代码（shim）进行签名，使用 OS 内核代码证书私钥对 OS 引导代码（GRUB）及内核镜像进行签名。

——设备厂商应使用驱动模块证书私钥，对计算机启动固件启动的设备启动代码，以及非计算机启动固件启动的内核模块（.ko）进行签名。

注：计算机启动固件启动的设备启动代码包括网卡、显卡等硬件设备中用于对设备进行初始化的程序。在计算机上电阶段，这些设备启动代码由计算机启动固件使用允许列表（db）中的证书进行验证和加载；而这些设备的驱动程序（.ko文件）则由内核镜像启动后进行验证、加载。

**证书预置：**

——PK 由计算机厂商或设备所有者生成，并设置在计算机启动固件中。

——KeK 证书由 OS 厂商提供，由相关机构经授权的 CA 统一签发，签发应遵循第 8 章规范。

——启动垫片证书及驱动模块证书由 OS 厂商、设备厂商提供。

——计算机出厂时，应依据产品需求，在计算机启动固件中配置计算机平台密钥（PK）、密钥交换密钥（KeK）证书、启动垫片证书及驱动模块（设备启动代码）证书。

——OS 厂商将内核代码证书设置在启动垫片中。

——OS 厂商或 OS 系统管理员将非计算机固件启动的内核模块（.ko）的签名证书链设置在 OS 系统证书信任列表中。

注：PK、KeK证书的运行管理超出本文件范围，由其他标准进行规范。

### D.1.2 安全启动证书验证

计算机安全启动过程分为硬件启动和OS启动，在硬件启动过程之后进行OS启动，本附录规范OS启动的证书验证。

注：硬件启动证书验证：验证KeK是否由PK签名，启动垫片证书及驱动模块证书是否由KeK签名。

OS启动证书验证至少包括图D-1所示的步骤及事项。

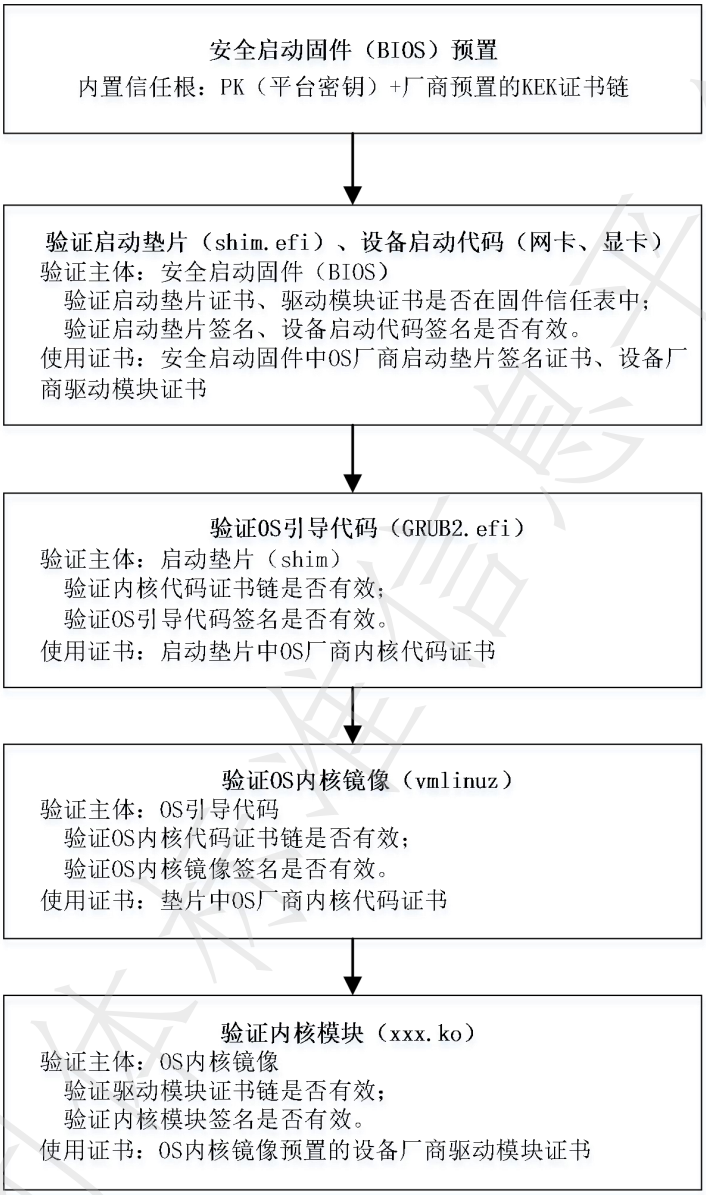
验证步骤：

**验证1**（固件验证启动垫片和设备启动代码）：计算机安全启动固件使用自身信任的证书（PK、KeK证书、预置的启动垫片证书）验证启动垫片的签名，只有通过验证的启动垫片才能被加载。同时，安全启动固件使用设备厂商提供的驱动模块证书，对装在设备（如网卡、显卡）中的设备启动代码的签名进行验证，只有通过验证的设备启动代码才能被运行。

**验证2**（启动垫片验证引导代码）：启动垫片使用自身内置的OS厂商内核代码证书，验证OS引导代码（GRUB）的签名，只有通过签名验证的OS引导代码才能被加载。

**验证3**（OS引导代码验证OS内核镜像）：OS引导代码使用启动垫片内置的OS厂商内核代码证书，验证OS内核镜像（vmlinuz）的签名，只有通过签名验证的OS内核镜像才能被加载。

**验证4**（OS内核镜像验证内核模块）：OS内核镜像使用预置的设备厂商驱动模块证书，对内核模块（.ko）的签名进行验证，未签名或签名无效的内核模块被拒绝加载。




图D-1 OS安全启动过程证书验证步骤

## D.2 文件完整性证书使用机制

文件完整性证书是OS完整性度量架构（Integrity Measurement Architecture，IMA）机制使用的证书，其核心作用是验证OS关键文件/数据的完整性、来源可信性。OS关键文件/数据包括：

——用户态可执行文件（ELF 格式，如/bin/bash、/usr/bin/nginx）；
——动态链接库（.so，如/lib64/libc.so.6）；
——内核模块（.ko，如驱动 nvidia.ko、功能模块 ntfs3.ko）；
——脚本文件（如/bin/sh 执行的.sh 脚本、Python/Perl 脚本）；
——固件文件（如硬件固件/lib/firmware/rtl8192cu.bin）；
——关键配置文件（如/etc/passwd、/etc/sudoers、/etc/fstab）；
——系统服务文件（如/usr/lib/systemd/system/*.service）；
——内核配置文件（如/sys/module/*/parameters/*）；
——磁盘镜像/块设备（如/dev/sda1 挂载前的完整性校验）；
——网络传输的可信文件（结合 IMA 网络验证扩展）；
——容器镜像/ OCI 运行时文件（结合 CRI-O/containerd 的 IMA 集成）。

当以上文件被加载、执行、访问时，OS进行签名验证，并根据验证策略执行相应的处理。验证策略由系统管理员配置，用于定义IMA度量与评估规则。

文件完整性证书的使用、签发、管理应遵照第8章的规范要求。

注：文件完整性证书签名机制，包括对第三方设备驱动（.ko）签名，与内核模块签名有重叠，但签名/验证方法不同，文件完整性证书签名对文件整体签名，内核模块签名只对.ko中的驱动代码签名，两种签名值存放位置不同。两种机制可以同时使用。

### D.3 OS 应用软件包签名证书使用机制

签名与验证逻辑：

应用软件发行方（开发者）使用私钥对应用软件包进行签名，OS用相应的公钥进行验签。

a） 发行方签名

1） 计算软件包（.deb/.rpm）的哈希值；

2） 使用私钥对该杂凑值进行签名，生成数字签名文件（后缀为.sea）或直接将其内嵌在软件包中；

3） 将发行方的公钥封装在数字证书中，存放在数字签名文件中。

OS应用软件包签名格式遵循T/ZISIA 0105-2026。

b） OS 验签

1） 用户下载软件包及其对应的签名文件；

2） OS 获取发行方应用软件包签名证书，对软件包的签名进行验证；

- 若验证通过，允许安装；
- 若验证不通过，拒绝安装（或根据验证策略处置）。

证书使用、签发、管理遵照第8章规范要求。

### D.4 可执行程序签名证书使用机制

可执行程序签名证书用于验证用户态可执行程序（如二进制工具、脚本）的完整性及来源。

验证全程在用户态下进行，内核不参与验证，是否对可执行程序签名进行验证由用户/企业策略决定。

证书使用、签发、管理遵照第8章规范要求。

### D.5 SSH 证书使用机制

SSH证书主要用于远程登录工具SSH，采用SSH CA（证书颁发机构）机制，由企业统一的SSH CA签发用户证书，服务器只需信任CA公钥，即可验证所有用户证书，管理员用证书登录服务器，客户端无需上传公钥，直接用“私钥+用户证书”登录。

SSH证书使用、签发、管理遵照第8章规范要求。

# 参 考 文 献

[1] 商用密码管理条例
[2] 电子政务电子认证服务管理办法
[3] 电子认证服务使用密码管理办法（征求意见稿）
[4] GM/T AAAA 公共可信证书管理 代码签名证书运营和管理要求（征求意见稿）
[5] UEFI Specification：https://uefi.org/sites/default/files/resources/UEFI_Spec_Final_2.11.pdf