PHYTIUM 飞腾

腾锐 D2000 安全 CPU PSPA 软件编程手册

2021 年 04 月

飞腾信息技术有限公司

www.phytium.com.cn

更新记录

本文档经过如下修订。

更新记录			
日期	版本号	私密性	改变备注
2021.04.15	1.0	NDA	发布的第一个版本

术语与缩略语

缩写	全称	描述
PSPA	Phytium Security Platform Architecture	飞腾安全平台架构
TEE	Trusted Execution Environment	可信执行环境
REE	Rich Execution Environment	丰富执行环境
API	Application Programming Interface	应用编程接口
PBF	Processor Base Firmware	飞腾基础固件
PBR	Processor Boot ROM	飞腾启动 ROM
OTP	One Time Programmable	一次可编程器件
ROTK	Root of trust keys	可信根密钥对
ROTPK	Root of Trust Public Key	可信根公钥

目 录

图目录

表目录

1 概述

腾锐 D2000 安全 CPU 是一款面向安全应用领域的高性能通用 8 核处理器，设计遵循飞腾安全处理器架构规范 PSPA1.0（Phytium Security Platform Architecture）[1,2]。

PSPA 1.0 包含十个方面主要内容，即：密码加速引擎、密钥管理、可信启动、可信执行环境、安全存储、固件管理、量产注入、生命周期管理、抗物理攻击及硬件漏洞免疫。具体如图 1-1 所示。

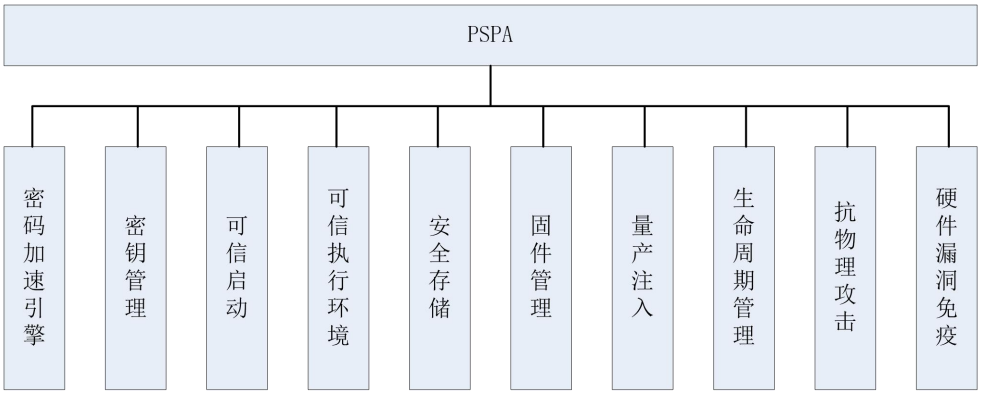

图 1-1 PSPA 总体框架

这十个方面包括密码密钥相关的内容，即密码加速引擎、密钥管理；也包括可信相关的内容，即可信启动、可信执行环境；包括敏感信息和固件的保护相关内容，即安全存储、固件管理；也包括芯片生产及全寿命周期管理的相关内容，即量产注入、生命周期管理；还包括抗物理攻击及硬件漏洞免疫相关内容。

本文档为 D2000 安全 CPU 上遵循 PSPA 1.0 的软件编程手册。

2 PSPA 软件栈

PSPA 软件涉及飞腾引导 ROM（PBR）、飞腾基础固件（PBF）、通用执行环境（REE）和可信执行环境（TEE）等各个执行阶段，如图 2-1 所示。

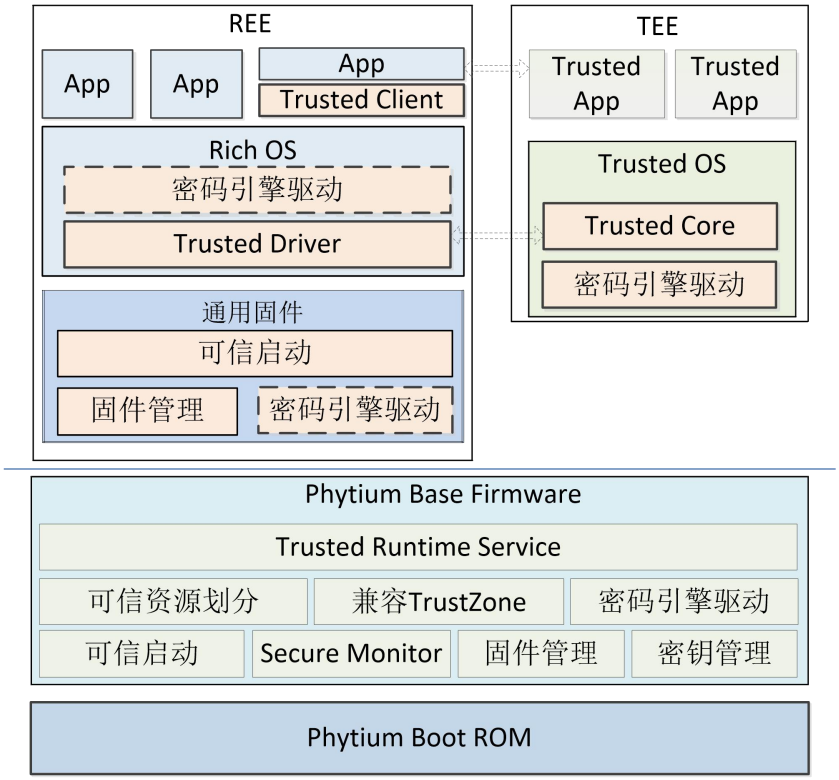

图 2-1 PSPA 软件栈

其中，PBR 是系统的可信根。PBF 对可信资源进行划分，负责可信执行环境与通用执行环境的管理和切换，并提供运行时服务。可信启动流程贯穿 PBR、PBF、通用固件、操作系统等整个执行流程。根据密码引擎的使用模式和使用场景，通用固件、通用操作系统、可信操作系统都可能需要密码引擎的驱动。根据用户需求，同一个密码引擎可以归 REE 或 TEE 专用，但不应在两个执行环境之间共享。

3 密码加速引擎

D2000 内部集成密码加速引擎 SCTO(SM CRYPTO)模块，该模块能够硬件加速 SM2、SM4、SM3、SM9 密码算法，并且能够产生真随机数。

3.1 密码加速引擎寄存器列表

本章中涉及的寄存器属性如下所示：

RO：W: 对存储的数值无影响, R: 对存储的数值无影响

RW：W: 将对应位改为输入数值, R: 对存储的数值无影响

W1C: W: 写 1 将会将对应比特清除，写 0 无影响, R:对存储的数值无影响

W1S: W: 写 1 将会将对应比特置高，写 0 无影响, R:对存储的数值无影响

SCTO 基址：0x28220000。

表 3-1 SCTO 寄存器列表

寄存器名	地址偏移	位宽	R/W	描述
Global				
CR	0x0000	1	W1S	控制寄存器，复位：0x0
CMD	0x0004	3	W1S	指令寄存器
CFG	0x0008	1	RW	配置寄存器，复位：0x0
SR1	0x0010	6	RO	状态寄存器 1，复位：0x0
SR2	0x0014	3	W1C	状态寄存器 2，复位：0x0
VERSION	0x0030	24	RO	版本寄存器，复位：0x00E3 0000
DMA				
CFG	0x0100	3	RW	DMA 配置寄存器，复位：0x0
SR	0x0104	1	W1C	DMA 状态寄存器，复位：0x0
TO_THRES	0x0108	16	RW	DMA 超时阈值寄存器，复位：0x0
SADDR	0x0110~0x0114	32	RW	DMA 源地址寄存器，复位：0x0
DADDR	0x0120~0x0124	32	RW	DMA 目的地址寄存器，复位：0x0
LEN	0x0130	32	RW	DMA 数据长度，复位：0x0
CFG_AW	0x0140	26	RW	AXI AW 通道配置寄存器，复位：

				0x0
CFG_AR	0x0150	27	RW	AXI AR 通道配置寄存器，复位：0x0
SKE_HP				
SKE_CTRL	0x1000	2	W1S	SKE 控制寄存器，复位：0x0
SKE_CFG	0x1004	8	RW	SKE 配置寄存器，复位：0x0
SKE_SR_1	0x1008	2	RO	SKE 状态寄存器 1，复位：0x0
SKE_SR_2	0x100C	1	W1C	SKE 状态寄存器 2，复位：0x0
保留	0x1010~0x106C			
IV	0x1070~0x107C	32	RW	初始向量寄存器，复位：0x0
KEY_WORD	0x1080~0x108C	32	WO	密钥寄存器，复位：0x0
保留	0x1090~0x113C			
SKE_VERSION	0x1140	24	RO	版本寄存器，复位：0x00E3 0000
保留	0x1144~0x11FC			
MDIN	0x1200~0x120C	32	RW	数据输入寄存器
MDOUT	0x1210~0x121C	32	RO	数据输出寄存器，复位：0x0
HASH_HP				
HASH_CTRL	0x2000	3	W1S	哈希控制寄存器，复位：0x0
HASH_CFG	0x2004	2	RW	哈希配置寄存器，复位：0x0000 0000
HASH_SR_1	0x2008	1	RO	哈希状态寄存器 1，复位：0x0
HASH_SR_2	0x200C	1	W1C	哈希状态寄存器 2，复位：0x0
保留	0x2010~0x201C			
PCR_LEN	0x2020~0x2024	32	RW	消息长度寄存器，复位：0x0
保留	0x2028~0x202C			
HASH_OUT	0x2030~0x204C	32	RO	哈希值输出寄存器，复位：0x0

保留	0x2050~0x206C			
HASH_IN	0x2070~0x208C	32	WO	哈希值输入寄存器
保留	0x2090~0x20AC			
VERSION	0x20B0	24	RO	版本寄存器，复位：0x00E3 0000
保留	0x20B4~0x20FC			
M_DIN	0x2100~0x213C	32	RW	消息输入寄存器
TRNG				
TRNG_CR	0x3000	9	RW	TRNG 控制寄存器，复位：0x0103 001F
TRNG_RTCR	0x3004	1	RW	TRNG 运行控制寄存器，复位：0x0
TRNG_SR	0x3008	3	W1C	TRNG 状态寄存器，复位：0x0
TRNG_DR	0x300C	32	RO	TRNG 随机数寄存器，复位：0x0
保留	0x3010~0x301C			
TRNG_FIFO_CR	0x3020	6	RW	FIFO 控制寄存器，复位：0x0007 0007
TRNG_FIFO_SR	0x3024	18	RO	FIFO 状态寄存器，复位：0x0100 0100
保留	0x3028~0x306C			
TRNG_HT_SR	0x3070	6	RO	在线监测状态寄存器，复位：0x0
保留	0x3074~0x307C			
RO_CR	0x3080	32	RW	RO 控制寄存器，复位：0xFFFF FFFF
RO_CR2	0x3084	32	RW	RO 控制寄存器 2，复位：0xFFFF FFFF
RO_CR3	0x3088	2	RW	RO 控制寄存器 3，复位：0x0
SENSOR				
SENSOR_CR	0x4000	9	RW	SENSOR 控制寄存器，复位：0x0

SENSOR_SR	0x4004	4	RO	SENSOR 状态寄存器，复位：0xA
SENSOR_SRST	0x4008	1	W1S	SENSOR 控制寄存器，用于复位 复位：0x0
PKE_HP				
CTRL	0x5000	1	W1S	PKE_HP 控制寄存器，复位：0x0
CONF	0x5004	1	RW	PKE_HP 配置寄存器，复位：0x0
保留	0x5008~0x500C			
MC_PTR	0x5010	12	RW	PKE_HP 过程入口，复位：0x0
保留	0x5014~0x501C			
STAT	0x5020	1	W1C	PKE_HP 状态寄存器，复位：0x0
RT_CODE	0x5024	4	RO	PKE_HP 返回日志寄存器，复位： 0x0
保留	0x5028~0x507C			
VERSION	0x5080	24	RO	PKE_HP 版本寄存器，复位： 0x00E3 0000
保留	0x5084~0x53FC			
MEM_REGION_A	0x5400~0x555C	32	RW	运算操作数 RAM0，复位：0x0
保留	0x5560~0x5FFC			
MEM_REGION_B	0x6000~0x615C	32	RW	运算操作数 RAM1，复位：0x0

3.2 寄存器详细描述

3.2.1 Global 寄存器

3.2.1.1 IP 控制寄存器(CR)

地址偏移：0x0000

描述：IP WRAPPER 控制器，包括开始和停止位

类型：W1S

域	位	读写	复位值	描述
Reserved	31:1	RO	0x0	保留

GO	0	W1S	0x0	启动 SCTO,IP WRAPPER 会根据指令状态寄存器的状态处理所接收到的指令

3.2.1.2 IP 指令寄存器(CMD)

地址偏移：0x0004

描述：指令寄存器，包括 PKE，HASH，SKE 所接受的指令

类型：W1S

域	位	读写	复位值	描述
Reserved	31:3	RO	0x0	保留
P_ST	2	W1S	0x0	PKE START 指令，当对该域进行写操作时，该指令及 DMA 的配置会被保存，等待 IP 接收 GO 指令后处理
H_ST	1	W1S	0x0	HASH START 指令，当对该域进行写操作时，该指令及 DMA 的配置会
S_ST	0	W1S	0x0	SKE START 指令，当对该域进行写操作时，该指令及 DMA 的配置会被保存，等待 IP 接收 GO 指令后处理。

3.2.1.3 IP 配置寄存器(CFG)

地址偏移：0x0008

描述：IP WRAPPER 配置寄存器，包含中断使能

类型：RW

域	位	读写	复位	描述
Reserved	31:1	RO	0	保留
IRQEN	0	RW	0x0	全局中断使能 0：禁止产生中断 1：允许产生中断

3.2.1.4 IP 状态寄存器(SR1)

地址偏移：0x0010

描述：IP WRAPPER 状态寄存器 1，包含忙指示位

类型： RO

域	位	读写	复位	描述
Reserved	31:21	RO	0	保留
BUSY_CODE	20:16	RO	0x0	内部模块状态 [16]: 指示 SKE 状态，1 表示忙 [17]: 指示 HASH 状态，1 表示忙 [18]: 指示 PKE 状态，1 表示忙 [19]: 指示 TRNG 状态，1 表示随机数已就绪 [20]: 指示 SENSOR 状态，1 表示报警
Reserved	15:1	RO	0x0	保留
0	BUSY	RO	0x0	标志 IP 处于忙状态

3.2.1.5 IP 状态寄存器(SR2)

地址偏移：0x0014

描述：IP WRAPPER 状态寄存器 2，包含各 IP 中断指示位。该寄存器可配合中断使用，显示 SMx 的状态。在 CFG.IRQEN=1 时，该寄存器内的域被置位会使得 o_irq(中断信号)拉高。当 o_irq 拉高时，用户应查询该寄存器来确定中断源，并清除该中断，保证后续指令的中断可以正常显示。

域	位	读写	复位	描述
Reserved	31:3	RO	0	保留
PKE_DONE	2	W1C	0x0	标志 PKE 指令已完成
HASH_DONE	1	W1C	0x0	标志 HASH 指令已完成
SKE_DONE	0	W1C	0x0	标志 SKE 指令已完成

3.2.1.6 IP 版本寄存器(VERSION)

地址偏移：0x0030

描述: IP WRAPPER 版本寄存器，用于记录该模块的版本信息

类型：RO

域	位	读写	复位	描述
PROJECT	31:16	RO	0x00E3	项目编号
Reserved	15:8	RO	0	保留
MAR	7:4	RO	0	主版本编号
MIR	3:0	RO	0x9	副版本编号

3.2.2 DMA 寄存器

3.2.2.1 配置寄存器(CFG)

地址偏移：0x0100

描述：DMA 配置寄存器，用于配置突发长度

类型：RW

域	位	读写	复位	描述
Reserved	31:3	RO	0	保留
MBL	2:0	RW	0	DMA 可使用的最大突发长度，该设置会限制 DMA 不使用过长的突发长度进行读写 default：0，即突发长度为 1

3.2.2.2 状态寄存器(SR)

地址偏移：0x0104

描述：DMA 状态寄存器

类型：W1C

域	位	读写	复位	描述
Reserved	31:1	RO	0	保留
TO	0	W1C	0	表示 DMA 是否长时间未得到响应。 0：DMA 未触发 time out 1：DMA 触发 time out

3.2.2.3 超时阈值寄存器(TO_THRES)

地址偏移：0x0108

描述: DMA 超时阈值寄存器

类型：RW

域	位	读写	复位	描述
Reserved	31:16	RO	0	保留
TO_THRESHOLD	15:0	RW	0	DMA 超时阈值 0x0：表示不启用超时报警功能 others：设定的超时阈值，设定 AXI 读写请求发出后最大可等待时钟周期数

3.2.2.4 源地址寄存器(SADDR)

地址偏移：0x0110

描述：DMA 源地址

类型：RW

域	位	读写	复位	描述
SADDR0	31:0	RW	0	所需处理数据所在系统存储器内的源地址，该地址是字地址

地址偏移：0x0114

描述：DMA 源地址

类型：RW

域	位	读写	复位	描述
Reserved	31:12	RO	0	保留
SADDR1	11:0	RW	0	所需处理数据所在系统存储器内的源地址，该地址是字地址

0x0110 对应 SADDR 的低 32 比特。 0x0114 的[11:0]对应 SADDR 的高 12 比特。SADDR 共 44 比特

3.2.2.5 目的地址寄存器(DADDR)

地址偏移：0x0120

描述：DMA 目的地址

类型：RW

域	位	读写	复位	描述
DADDR0	31:0	RW	0	处理后数据所需存放的目的地址，该地址是字地址对于 SKE，加解密后的消息会被 DMA 搬运至目的地址对于 HASH，哈希运算后，结果会被搬运至目的地址

地址偏移：0x0124

描述：DMA 目的地址

类型：RW

域	位	读写	复位	描述
Reserved	31:12	RO	0	保留
DADDR1	11:0	RW	0	处理后数据所需存放的目的地址，该地址是字地址对于 SKE，加解密后的消息会被 DMA 搬运至目的地址对于 HASH，哈希运算后，结果会被搬运至目的地址

0x0120 对应 DADDR 的低 32 比特。 0x0124 的[11:0]对应 DADDR 的高 12 比特。DADDR 共 44 比特。

3.2.2.6 数据长度寄存器(LEN)

地址偏移：0x0130

描述：DMA 数据长度

类型：RW

域	位	读写	复位	描述
LEN	31:0	RW	0	表示所需搬运的数据字长度 对于 SKE，该 LEN 表示所需读写的数据长度 对于 HASH，该 LEN 表示所需读取的数据长度 对于 PKE，该 LEN 的数值应当是 0x18 或 0x30

3.2.2.7 DMA 配置寄存器(CFG_AW)

地址偏移： 0x0140

描述：DMA 配置寄存器，包含 ACE-lite 写地址通道的配置

类型：RW

域	位	读写	复位	描述
Reserved	31:30	RO	0x0	保留
AWBAR	29:28	RW	0x0	用户设置的 AWBAR
Reserved	27:26	RO	0x0	保留
AWDOMAIN	25:24	RW	0x0	用户设置的 AWDOMAIN
Reserved	23	RO	0x0	保留
AWSNOOP	22:20	RW	0x0	用户设置的 AWSNOOP
AWREGION	19:16	RW	0x0	用户设置的 AWREGION
AWQOS	15:12	RW	0x0	用户设置的 AWQOS
Reserved	11	RW	0x0	保留
AWPROT	10:8	RW	0x0	用户设置的 AWPROT
AWCACHE	7:4	RW	0x0	用户设置的 AWCACHE
AWID	3:0	RW	0x0	用户设置的 AWID

3.2.2.8 DMA 配置寄存器(CFG_AR)

地址偏移：0x0150

描述：DMA 配置寄存器，包含 AXI 读地址通道的配置

类型：RW

域	位	读写	复位	描述
Reserved	31:30	RO	0x0	保留
ARBAR	29:28	RW	0x0	用户设置的 ARBAR
Reserved	27:26	RO	0x0	保留
ARDOMAIN	25:24	RW	0x0	用户设置的 ARDOMAIN
ARSNOOP	23:20	RW	0x0	用户设置的 ARSNOOP

ARREGION	19:16	RW	0x0	用户设置的 ARREGION
ARQOS	15:12	RW	0x0	用户设置的 ARQOS
Reserved	11	RW	0x0	保留
ARPROT	10:8	RW	0x0	用户设置的 ARPROT
ARCACHE	7:4	RW	0x0	用户设置的 ARCACHE
ARID	3:0	RW	0x0	用户设置的 ARID

3.2.3 SKE HP 寄存器

Symmetric Key Engine High Performance（SKE_HP）是用于对称密码算法运算的高性能硬件加速模块；支持 SM4 加解密，支持 ECB, CBC, CFB, OFB, CTR 工作模式。

3.2.3.1 SKE 控制寄存器(SKE_CTRL)

地址偏移：0x1000

描述: SKE 控制寄存器

类型：W1S

域	位	读写	复位	描述
Reserved	31:17	RO	0x0	保留位
SKE_SRST	16	W1S	0x0	SKE 软复位 1: 软复位 SKE 0: 无影响
Reserved	15:1	RO	0x0	保留位
SKE_START	0	W1S	0x0	SKE 启动运算 1： SKE 启动运算 0：无影响

3.2.3.2 SKE 配置寄存器 (SKE_CFG)

地址偏移：0x1004

描述：SKE 配置寄存器

类型：RW

注意：配置均会在 SKE_START 为高时被采样。在 SKE 运行期间改变这些设置的值不会影

响 SKE 的运行状态。

域	位	读写	复位	描述
Reserved	31:25	RO	0x0	保留位
IRQEN	24	RW	0x0	中断使能 0： SKE 禁止产生中断 1： SKE 允许产生中断
Reserved	23:19	RO	0x0	保留位
IVU	18	RW	0x0	初始向量更新: 0: SKE 不更新初始向量。第二组及以后的数据无需更新初始向量 1: SKE 更新初始向量。在需要使用初始向量的工作模式下，某块数据的第一组加解密运算必须更新初始向量
SPEN	17	RW	0x0	安全端口使能: 0: SKE 采用 AHB 输入的密钥进行运算 1: SKE 采用安全端口输入的密钥进行运算
Reserved	16:12	RO	0x0	保留位
OP_MODE	11:9	RW	0x0	工作模式选择： 3'b000: ECB 3'b001: CBC 3'b010: CFB 3'b011: OFB 3'b100: CTR others: 保留

DEC	8	RO	0x0	加解密选择： 0: 加密 1: 解密
Reserved	7:0	RO	0x0	保留位

3.2.3.3 SKE 状态寄存器 1 (SKE_SR_1)

地址偏移：0x1008

描述：SKE 状态寄存器

类型：RO

注意：推荐用户在触发 SKE 前先读取该寄存器的值。

域	位	读写	复位	描述
Reserved	31:9	RO	0	保留位
ERR_CFG	8	RO	0x0	错误配置 错误配置标志位不受软复位影响。当用户输入正确配置时，该指示位会自动清 0。该指示位与 SKE_CFG 的值有关。请参阅相关寄存器说明获取更多信息。目前错误配置指示以下错误配置： 1. OP_MODE = others
Reserved	7:1	RO	0x0	保留位
SKE_B	0	RO	0x0	SKE 忙指示位

3.2.3.4 SKE 状态寄存器 2 (SKE_SR_2)

地址偏移：0x100c

描述：SKE 状态寄存器 2

类型：W1C

注意：推荐用户在启动 SKE 前先读取并清除该寄存器的值。

域	位	读写	复位	描述
Reserved	31:17	RO	0	保留
SKE_IRQ	16	W1C	0	SKE 中断指示位

				0: SKE 无中断 1: SKE 已正确结束运算，中断会在 SKE 正确结束后置位。软复位或再次发送 SKE 启动指令会清除该指示位
Reserved	15:0	RO	0x0	保留位

3.2.3.5 初始向量寄存器

地址偏移：0x1070 ~ 0x107C

描述：初始向量寄存器

类型： RW

注意：改寄存器直接由算法模块接收，在算法运行期间请保持改寄存器的值不改变。

域	位	读写	复位	描述
IV	31:0	RW	0x0	初始化向量， SKE 会根据 IVU 的值而决定是否更新初始化向量

IV_0 对应初始向量的最高 32 比特。对于 SM4， IV_0 = IV[127:96]，以此类推。

3.2.3.6 密钥寄存器

地址偏移：0x1080~0x108C

描述：密钥寄存器

类型：WO

注意：改寄存器直接由算法模块接收，在算法运行期间请保持该寄存器的值不改变。

域	位	读写	复位	描述
KW	31:0	RO	0	密钥输入

3.2.3.7 版本寄存器(SKE_VERSION)

域	位	读写	复位	描述
PROJECT	31:16	RO	0x00E3	项目代号
Reserved	15:8	RO	0x0	保留位

MAR	7:4	RO	0x0	主版本号
MIR	3:0	RO	0x0	次版本号

3.2.3.8 数据输入寄存器 (M_DIN)

地址偏移：0x1200~0x120C

描述：数据输入寄存器

类型：RW

注意：当 SKE 接收到启动指令时会缓存该寄存器的数据。

域	位	读写	复位	描述
M_DIN	31:0	RW	0x0	数据输入，系统向 SKE 搬运数据时的入口

M_DIN_0 对应输入消息的最高 32 比特。对于 SM4， M_DIN_0 =MSG[127:96]，以此类推。

3.2.3.9 数据输出寄存器 (M_DOUT)

地址偏移： 0x1210~0x121C

描述：数据输出寄存器

类型：RO

注意：SKE 通过该寄存器输出已处理的数据，在下一次启动 SKE 前，用户需读取该寄存器，防止数据丢失。

域	位	读写	复位	描述
M_DOUT	31:0	RO	0x0	数据输出，外部读取 SKE 处理结果时的数据出口

3.2.4 HASH HP 寄存器

Hash High Performance（HASH HP）是用于密码杂凑算法运算的高性能硬件加速模块，支持 SM3 算法。

3.2.4.1 控制寄存器(HASH_CTRL)

地址偏移：0x2000

描述：HASH 控制寄存器

类型：W1S

域	位	读写	复位	描述
Reserved	31:17	RO	0x0	保留位
SRST	16	W1S	0x0	软复位 1: 软复位 HASH,丢弃当前处理数据 0: 无影响
Reserved	15:2	RO	0x0	保留位
CONT	1	W1S	0x0	继续哈希操作 1: HASH 会使用上一次计算所得哈希值或者用户输入的哈希值进行运算 0: 无影响
START	0	W1S	0x0	启动哈希操作 1: HASH 会使用算法定义的初始哈希值进行运算 0：无影响

3.2.4.2 配置寄存器 (HASH_CFG)

地址偏移：0x2004

描述：HASH 配置寄存器

类型：RW

注意：在 HASH 忙标志位为高期间，HASH 禁止用户改变寄存器的值，即写入无效。

域	位	读写	复位	描述
Reserved	31:25	RO	0x0	保留位
LAST	24	RW	0x0	消息末尾标志： 0: 本次处理消息未包含末尾 1: 本次处理消息包含末尾
Reserved	23:17	RO	0x0	保留位
IRQEN	16	RW	0x0	中断使能：

				0: HASH 禁止产生中断 1: HASH 允许产生中断
Reserved	15:0	RO	0x0	保留位

3.2.4.3 状态寄存器 1 (HASH_SR_1)

地址偏移：0x2008

描述：HASH 状态寄存器 1

类型：RO

注意：在发送 START/CONT 指令前，用户需要先读该寄存器的值。

域	位	读写	复位	描述
Reserved	31:1	RO	0x0	保留位
BUSY	0	RO	0x0	忙指示位 1: HASH 处于忙状态 0: HASH 处于空闲状态

3.2.4.4 状态寄存器 2 (HASH_SR_2)

地址偏移：0x200c

描述：HASH 状态寄存器 2

类型：W1C

注意：推荐用户在发送指令前先清除该寄存器的值。

域	位	读写	复位	描述
Reserved	31:1	RO	0x0	保留位
IRQ	0	W1C	0x0	HASH 中断指示位 0: HASH 无中断产生 1: HASH 已完成操作，中断产生。用户可对该比特位写 1 清除中断。 HASH 在接收到 START/CONT/SRST 指令时也会清除中断

3.2.4.5 运行控制寄存器 (PCR_LEN)

地址偏移：0x2020 ~ 0x2040

描述：HASH 运行控制寄存器

类型：RW

注意：在 HASH 忙标志位为高期间，HASH 禁止用户改变该寄存器的值，即写入无效。

域	位	读写	复位	描述
LEN	31:0	RW	0x0	消息长度 该寄存器表示所需进行哈希运算的消息的字节长度。 HASH 使用 LEN 作为消息长度进行运算。若所需哈希的消息长度为 1024 字节，则 LEN 需设置为 1024

注： 0x20 对应消息长度的最低 32 比特。 如果一条消息的长度需要 64 比特来表示的话，那么最低 32 比特存放于 0x20，次低存放于 0x24。

3.2.4.6 输出寄存器 (HASH_OUT)

地址偏移：0x2030~0x204C

描述：HASH 输出

类型：RO

注意：只有在 HASH 处于空闲状态时，该地址才可读。

域	位	读写	复位	描述
HOUT	31:0	RO	0x0	哈希值输出

3.2.4.7 输入寄存器（HASH_IN）

地址偏移：0x2070 ~ 0x208C

描述：HASH 输入

类型：WO

注意：只有在 HASH 处于空闲状态时，改地址才可写。

域	位	读写	复位	描述
HIN	31:0	WO	0x0	哈希值输入

3.2.4.8 版本寄存器(VERSION)

地址偏移：0x20B0

描述：HASH 版本寄存器

类型：RO

域	位	读写	复位	描述
PROJECT	31:16	RO	0x00E3	项目代号
Reserved	15:8	RO	0x0	保留位
MAR	7:4	RO	0x0	主版本号
MIR	3:0	RO	0x0	次版本号

3.2.4.9 消息输入寄存器 (M_DIN)

地址偏移：0x2100 ~ 0x213C

描述：消息输入寄存器

类型：RW

域	位	读写	复位	描述
DIN	31:0	RW	0x0	消息输入

3.2.5 TRNG 寄存器

True Random Number Generator（TRNG）是用于产生真随机数的随机数发生模块。

3.2.5.1 TRNG 控制寄存器 (TRNG_CR)

地址偏移：0x3000

描述：随机数发生器控制寄存器，包含中断使能，熵源选择及随机数发生器使能

类型：RW

注意：仅在 RBGEN 为低时采样该寄存器的值，当 RBGEN 为高时，改变该寄存器的值不影响设计行为。

域	位	读写	复位	描述
Reserved	31:25	RO	0x0	保留
IRQEN	24	RW	1	中断使能. 16~18 位仅在该使能信号为 1 时

				才有效 0：禁止中断 1: 使能中断
Reserved	24	RO	0x0	保留
ERIEN	18	RW	0x0	读空中断使能 0: 禁止读空中断 1: 使能读空中断
DIEN	17	RW	1	数据中断使能 0: 禁止数据中断 1: 使能数据中断
HTEN	16	RW	1	健康检测使能 0: 禁止健康检测 1: 使能健康检测
Reserved	15；5	RW	0x0	保留
ROSEN	4:1	RW	0xF	RO 熵源使能 [0]： 1 号熵源使能 [1]： 2 号熵源使能 [2]： 3 号熵源使能 [3]： 4 号熵源使能
RBGEN	0	RW	1	随机数模块使能 0: 关闭随机数模块 1: 使能随机数模块

3.2.5.2 TRNG 运行控制寄存器 (TRNG_RTCR)

地址偏移：0x3004

描述：随机数模式选择寄存器，动态选择输出的随机数是否经过后处理

类型：RW

注意：该寄存器可以在 RNG_CR.RBGEN 为 1 期间动态修改。

域	位	读写	复位	描述

Reserved	31:1	RO	0x0	保留位
MSEL	0	RW	0x0	模式选择 0: 无后处理 1: 有后处理

3.2.5.3 TRNG 状态寄存器 (TRNG_SR)

地址偏移：0x3008

描述：随机数状态寄存器，包含读空错误，数据就绪和健康检测失败指示位。中断会根据 3 个指示位而产生。

类型：W1C

注意：推荐先向改寄存器写入 0x7 后再使能随机数模块。

域	位	读写	复位	描述
Reserved	31:3	RO	0x0	保留
ERERR	2	W1C	0x0	0x0
DRDY	1	W1C	0x0	数据就绪
HTF	0	W1C	0x0	健康检测失败

3.2.5.4 TRNG 随机数寄存器 (TRNG_DR)

地址偏移：0x300C

描述：随机数数据寄存器，用户可通过对这个寄存器发起读操作取随机数，当动态切换随机数工作模式时，该寄存器的值会改变。

类型： RO

注意：用户应先检查 RNG_SR.DRDY 再读取本寄存器，推荐先读取 RNG_FIFO_SR 后再根据 FIFO 状态读取本寄存器。

域	位	读写	复位	描述
RDATA	31:0	RO	0x0	随机数

3.2.5.5 TRNG FIFO 控制寄存器 (TRNG_FIFO_CR)

地址偏移：0x3020

描述：FIFO 控制寄存器，包含 FIFO 数据个数阈值

类型： RW

注意：当 RNG_CR.RBGEN 由 0 跳变为 1 时，本寄存器的值会被采样。当 RNG_CR.RBGEN 为 1 时改变本寄存器的值不会影响随机数发生器的行为。

域	位	读写	复位	描述
Reserved	31:19	RO	0x0	保留
TFTV	18:16	RW	0x7	TRNG FIFO 数据个数阈值 当 TRBG FIFO 内数据个数超过该设置时，随机数模块会产生随机数 就绪中断。例如，当阈值为 5 时，FIFO 内数据个数由 5 跳变为 6 时， 随机数模块产生中断。
Reserved	15:3	RO	0x0	保留
DFTV	2:0	RW	0x7	DRNG FIFO 数据个数阈值 当 DRBG FIFO 内数据个数超过该设置时，随机数模块会产生随机数就绪中断。例如，当阈值为 5 时，FIFO 内数据个数由 5 跳变为 6 时，随机数模块产生中断。

3.2.5.6 TRNG FIFO 状态寄存器 (TRNG_FIFO_SR)

地址偏移：0x3024

描述：FIFO 状态寄存器

类型: RO

注意：推荐先读取 FIFO 内数据个数再读 RNG_DR

域	位	读写	复位	描述
Reserved	31:25	RO	0x0	保留

TFE	24	RO	1	TRBG FIFO 空标志
TFCNT	23:16	RO	0x0	TRBG FIFO 数据个数
Reserved	15:9	RO	0x0	TRBG FIFO 数据个数
DFE	8	RO	1	DRBG FIFO 空标志
DFCNT	7:0	RO	0x0	DRBG FIFO 数据个数

3.2.5.7 TRNG 检测状态寄存器 (TRNG_HT_SR)

地址偏移：0x3070

描述：健康检测状态寄存器

类型：RO

域	位	读写	复位	描述
Reserved	31:26	RO	0x0	保留
RCFT	25	RO	0x0	Repetition Count Test 失败
APTF	24	RO	0x0	Adaptive Proportion Test 失败
Reserved	23:11	RO	0x0	保留
RCTBF	10	RO	0x0	Repetition Count Test BIST 失败
APTBBF	9	RO	0x0	Adaptive Proportion Test binary BIST 失败
APTNBBF	8	RO	0x0	Adaptive Proportion Test non-binary BIST 失败
Reserved	7:1	RO	0x0	保留
DBF	0	RO	0x0	DRBG BIST 失败

3.2.5.8 RO 控制寄存器 (RO_CR)

地址偏移：0x3080

描述：RO 控制寄存器 1

类型：RW

域	位	读写	复位	描述

ROEN1	31:16	RW	0xFFFF	熵源 1 内的 RO 使能 每比特对应一个 RO，熵源 1 含有 16 个 RO
ROEN2	15:0	RW	0xFFFF	熵源 2 内的 RO 使能 每比特对应一个 RO，熵源 2 含有 16 个 RO

3.2.5.9 RO 控制寄存器 2 (RO_CR2)

地址偏移：0x3084

描述：RO 控制寄存器 2

类型：RW

域	位	读写	复位	描述
ROEN3	31:16	RW	0xFFFF	熵源 3 内的 RO 使能 每比特对应一个 RO，熵源 3 含有 16 个 RO
ROEN4	15:0	RW	0xFFFF	熵源 4 内的 RO 使能 每比特对应一个 RO，熵源 4 含有 16 个 RO

3.2.5.10 RO 控制寄存器 3 (RO_CR3)

地址偏移：0x3088

描述：RO 控制寄存器 3

类型：RW

域	位	读写	复位	描述
Reserved	31:2	RO	0x0	保留
FSEL	1:0	RW	0x3	00: 采样时钟为源时钟的 4 分频 01: 采样时钟为源时钟的 8 分频 10: 采样时钟为源时钟的 16 分频

				11: 采样时钟为源时钟的 32 分频

3.2.6 SENSOR 寄存器

3.2.6.1 SENSOR 控制寄存器 (SENSOR_CR)

地址偏移：0x4000

描述：SENSOR 控制寄存器

类型：RW

域	位	读写	复位	描述
Reserved	31:13	RO	0x0	保留
CONFIG	12:8	RW	0x0	配置 SENSOR 反相器链长度，数值越大意味着反相器链越长，应当配置反相器链使其延迟大于被保护电路的关键路径长度，且小于时钟 周期。
Reserved	7:4	RO	0x0	保留
EN	3:0	RW	0x0	SENSOR 使能信号 others：使能 SENSOR 1010：关闭 SENSOR

3.2.6.2 SENSOR 状态寄存器 (SENSOR_SR)

地址偏移：0x4004

描述：SENSOR 状态寄存器

类型：RO

域	位	读写	复位	描述
Reserved	31:4	RO	0x0	保留
ALARM	3:0	RO	0xA	SENSOR 输出状态 1010：未受到攻击，SENSOR 正常 others：受到攻击，SENSOR 报警

3.2.7 PKE HP 寄存器

Public Key Engine High Performance（PKE_HP）是专门用来加速公钥密码运算中的大数模运算。PKE HP 的运算通过微码（Microcode）的形式完成，微码存储在程序存储单元中，支持 SM2、SM9 算法。

3.2.7.1 PKE_HP 控制寄存器(CTRL)

地址偏移：0x5000

描述：PKE_HP 控制寄存器，包括开始和停止控制

类型：W1S

域	位	读写	复位	描述
Reserved	31:1	RO	0x0	保留
GO	0	W1S	0x0	开始信号。当向该比特写 1 时， PKE_HP 会在下一个时钟周期开始运行。 PKE_HP 的运行基于向该比特写 1 的那个时钟周期的控制寄存器和数据寄存器的配置。

3.2.7.2 PKE_HP 配置寄存器(CONF)

地址偏移：0x5004

描述：PKE_HP 配置寄存器，用于配置运算位宽和中断等

类型：RW

域	位	读写	复位	描述
Reserved	31:9	RO	0x0	保留
IRQEN	8	RW	0x0	中断使能。当该比特为 1 时， o_irq(中断信号)接口有效。无论该比特是否为 1， STAT 寄存器并不受其影响。
Reserved	7:0	RO	0x0	保留

3.2.7.3 PKE_HP 过程入口寄存器(MC_PTR)

地址偏移：0x5010

描述：PKE_HP 过程入口

类型：RW

域	位	读写	复位	描述
Reserved	31:12	RO	0x0	保留
ADDR	11:0	RW	0x0	该域表明了 PKE_HP 下一条要执行的指令的地址。该寄存器只有在 PKE 不工作时才能被改写，任何在 PKE_HP 工作时的写操作都将会被忽略。 在 PKE_HP 运行过程中，该域也会实时更新。总是指向下一条将被执行的指令地址。因此，该寄存器也可与 CTRL.STOP 配合 debug。 需要注意的是，指令都是字对齐。因此，该域的最低 2 位都是 0。在向该域写指令地址时，被限制在 0x00~0x40 的地址范围内。

3.2.7.4 PKE_HP 状态寄存器 (STAT)

地址偏移：0x5020

描述：PKE_HP 状态寄存器，可以查看 PKE_HP 是否完成操作

类型：W1C

域	位	读写	复位	描述
Reserved	31:1	RO	0x0	保留
DONE	0	W1C	0x0	当该比特为 1 时，表明运算结束。当外部向该比特写 1 时，该比特会被清空。 另外，该比特也作为外部中断的清空位。在 CTRL.IRQEN 有效时，该比特为高时，外部中断信号也会被拉高。向该比特写 1，外部中断也会被清空。

3.2.7.5 PKE_HP 返回日志寄存器(RT_CODE)

地址偏移：0x5024

描述：PKE_HP 返回日志寄存器

类型：RO

域	位	读写	复位	描述
Reserved	31:4	RO	0x0	保留
STOP_LOG	3:0	RO	0x0	该域用来表示 PKE_HP 停止的原因。 如果 PKE_HP 是因为运算完成而停止的，那么该域的值为 0。如果该域的值为非零，则 PKE_HP 的运算未完成，遇到了一些异常，需要外部进行处理，结果不可用。 0: 正常停止 2: 无有效的模逆结果 3: 点不在曲线上 (CTRL.CMD: PVER) 4: 无效的 Microcode 其它: 保留

3.2.7.6 PKE_HP 版本寄存器(VERSION)

地址偏移：0x5080

描述：PKE_HP 版本寄存器，用于记录该模块的版本信息

类型：RO

域	位	读写	复位	描述
PROJECT	31:16	RO	0x00E3	项目代号
Reserved	15:8	RO	0x0	保留位
MAR	7:4	RO	0x0	主版本号
MIR	3:0	RO	0x0	次版本号

3.2.7.7 PKE_HP 运算操作数 RAM0(MEM_REGION_A)

地址偏移：0x5400 ~ 0x555C

描述：运算操作数 RAM0

类型：RW

域	位	读写	复位	描述
DATA_A	31：0	RW	0x0	该域用来存储运算数据

3.2.7.8 PKE_HP 运算操作数 RAM1(MEM_REGION_B)

地址偏移：0x6000 ~ 0x615C

描述：运算操作数 RAM1

类型：RW

域	位	读写	复位	描述
DATA_B	31:0	RW	0x0	该域用来存储运算数据

3.3 软件编程库

为简化用户使用密码加速引擎时的编程复杂性，飞腾对密码加速引擎的寄存器操作接口进行软件封装，提供相应的软件编程库，并提供该库 API 说明手册[7]。

4 密钥管理

腾锐 D2000 八核安全 CPU 集成一个 4Kbites 的片内一次性可烧写非易失性存储器 Electrical Fuse(简称 efuse)，专用于存储密钥和芯片其它的关键数据, efuse 存储器具有只写一次的特性，efuse 的每一个存储位一旦被写入比特值 1，就不能写回为 0，使得存储在 efuse 内部的密钥一旦重写即被破坏且这种破坏是不可逆的。

4.1 密钥存储空间

腾锐 D2000 八核安全 CPU 将密钥分为三个类型：飞腾密钥，OEM 密钥，用户密钥。飞腾密钥和 OEM 密钥存储在 efuse，腾锐 D2000 八核在 efuse 存储器内为用户划分 1 Kbits 的用户密钥存储空间，用户可以使用芯片内部的用户密钥存储空间，也可以由用户程序决定存储在其它位置。密钥存储空间的基地址为 0x2820A000，地址详细介绍如下表所示：

表 4-1 密钥存储空间

密钥名称	存储地址	存储位宽	描述
DFT	0x000~0x07C	1024bits	FT关键数据
HUK	0x080~0x09C	256bits	FT芯片KEY，衍生FT其他密钥，与硬件RTL KEY组成FT芯片的唯一标识。
KPICV	0x0A0~0x0AC	128bits	ICV provisioning master key ，通常用于SM4 算法。要进入CM RMA返厂状态，必须匹配该密钥和KCEICV才能进入。
KCEICV	0x0B0~0x0BC	128bits	ICV code encryption key ，通常用于SM4 算法。要进入CM RMA返厂状态，必须匹配该密钥和KPICV才能进入。
ICVFLAG	0x0C0	32bits	ICV-programmed flags，ICV标识，标记芯片生命周期状态。
HBK	0x0C4~0x0E0	256bits	Root-of-Trust Public Key ，SM3 算法KEY。
KCP	0x0E4~0x0F0	128bits	OEM provisioning master key，OEM厂商KEY，由OEM厂商定义。要进入DM RMA返厂状态，必须匹配该密钥和KCE才能进入。
KCE	0x0F4~0x100	128bits	OEM code encryption key，OEM厂商KEY，由OEM厂商定义。要进入DM RMA返厂状态，必须匹配该密钥和KCP才能进入。
OEMFLAG	0x104	32bits	OEM-programmed flags，OEM标识，标记芯片

			生命周期状态。
anti-rollback counter	0x108~0x118	160bits	anti-rollback counter，防回滚计数器。
USERREV	0x11C-0x17C	800bits	用户关键数据。
USERKEY	0x180~0x1FC	1024bits	用户KEY。
Rtl key	--	128bits	FT硬件RTL KEY，与HUK组成FT芯片的唯一标识。

4.2 密钥读写权限

腾锐 D2000 八核安全 CPU 在其全生命周期中各个不同阶段下，密钥的读写权限是不同的。生命周期(LCS)是指芯片从生产到交付整机厂商，进而由整机厂商将其作为整机的一部分交付最终客户的全生命周期管理过程，腾锐 D2000 八核安全 CPU 定义其生命周期为以下五个阶段：

- CM：芯片出厂阶段，此状态下所有的调试测试端口打开，能够访问芯片内的所有安全信息。
- DM：当芯片进入整机厂商后，芯片处于 DM 状态，DM 状态保留整机产商需要使用的调测试功能，芯片内部的扫描功能关闭，并且不能访问飞腾密钥。
- SE 状态：当芯片到达用户手中，芯片进入 SE 状态，调测测试的功能关闭，不能访问飞腾密钥和 OEM 密钥，必须采用 secure boot 方式启动。
- DM RMA：OEM 厂商返厂状态，密钥不可访问，OEM 调试测试功能打开。
- CM RMA：生成厂商返厂状态，密钥不可访问，CM 调试测试功能打开。

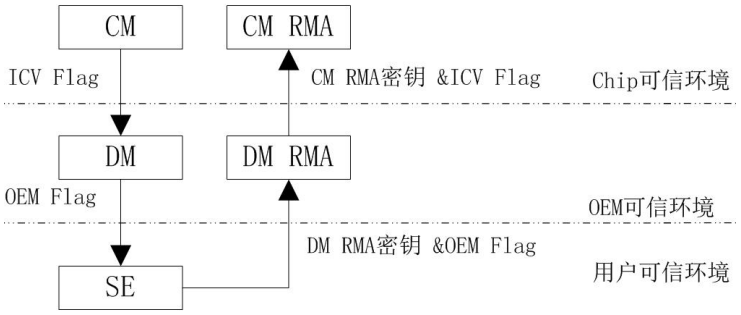

图 4-1 生命周期

腾锐 D2000 八核安全 CPU 的三类密钥：飞腾密钥（即根密钥及其衍生密钥）由飞腾生

产、飞腾固件使用和授权给 OEM 和用户使用；OEM 密钥由整机产商注入并使用，用户密钥由系统用户注入并使用。腾锐 D2000 八核安全 CPU 密钥的读写权限管理是基于芯片的生命周期，权限管理的细节如下表所示：

表 4-2 密钥读写权限

名称	密钥及关键数据生命周期管理		密钥长度(比特)	密钥归属	对应算法或用途
DFT	读	任意	1024	飞腾	FT关键数据。
	写	CM			
HUK	读	CM	256	飞腾	FT芯片KEY，衍生FT其他密钥，与硬件RTL KEY组成FT芯片的唯一标识。
	写	CM			
KPICV	读	CM	128	飞腾	ICVprovisioning master key ，通常用于SM4 算法。要进入CM RMA返厂状态，必须匹配该密钥和KCEICV才能进入。
	写	CM、CM RMA			
KCEICV	读	CM	128	飞腾	ICV code encryption key ，通常用于SM4 算法。要进入CM RMA返厂状态，必须匹配该密钥和KPICV才能进入。
	写	CM、CM RMA			
ICVFLAG	读	CM	32	--	ICV-programmed flags，ICV标识，与OEMFLAG共同标记芯片生命周期状态。
	写	CM、DM RMA			
HBK	读	任意	256	飞腾	Root-of-Trust Public Key ，SM3 算法KEY。
	写	CM			
KCP	读	CM 、DM	128	整机	OEM provisioning master key，OEM厂商KEY，由OEM厂商定义。要进入DM RMA返厂状态，必须匹配该密钥和KCE才能进入。
	写	DM、DM RMA			
KCE	读	CM 、DM	128	整机	OEM code encryption key，OEM厂商KEY，由OEM厂商定义。要进入DM RMA返
	写	DM、			

		DM RMA			厂状态，必须匹配该密钥和KCP才能进入。
OEMFLAG	读	CM 、DM	32	--	OEM-programmed flags ，OEM标识，与ICVFLAG共同标记芯片生命周期状态。
	写	DM、SE			
Anti-Rollback Counter	读	任意	160	--	anti-rollback counter，防回滚计数器。
	写	任意			
USERREV	读	CM、SE	800	用户	用户关键数据。
	写	SE			
USERKEY	读	CM、SE	1024	用户	用户KEY。
	写	SE			
RTL KEY	读	不可读	128	飞腾	FT硬件RTL KEY，与HUK组成FT芯片的唯一标识。
	写	不可写			

5 可信启动

D2000 支持可信启动，PBR 镜像存储在 ROM 中，在芯片生产过程中导入，芯片出厂后就无法修改。以 PBR 为可信根，通过分级验签其他阶段的镜像，确保固件不被篡改。

5.1 可信链构成部件

表格 5-1

部件名称	子部件	说明
ROTPK 的 hash 值		存储在芯片内部
PBR 镜像		存储在 ROM 中，无法被篡改
Keys	ROTK	私钥用来签名 PBF 内容证书和 trusted key 证书； 公钥 ROTPK 的 hash 值存储在芯片内部
	Trusted OS key 证书 key	私钥用来签名 secure world（比如 Trusted OS）镜像对应的 key 证书； 公钥存储在第三方公钥内容证书中
	通用固件 key 证书 key	私钥用来签名 non-secure world 镜像(比如，通用固件)对应的 key 证书； 公钥存储在第三方公钥内容证书中
	第三方 key 组	Trusted OS、通用固件各自的密钥对。这两部分可能由不同的第三方厂商提供，每家厂商有各自的密钥对，构成密钥组。 私钥用来签名对应的内容证书； 公钥存储在对应的 key 证书
Key 证书	第三方公钥内容证书	使用 ROTK 私钥自签名

		存储 Trusted OS key 证书公钥以及通用固件 key 证书公钥
	Trusted OS key 证书	trusted world key 私钥自签名； Trusted OS 可由多个第三方提供，因此该证书需要存储一组 Trusted OS 的公钥
	通用固件 key 证书	Non-trusted 私钥自签名； 通用固件可由不同的第三方提供，该证书需要存储一组公钥
内容证书	PBF 内容证书	包含 PBF 镜像的 hash 值 由 ROTK 的私钥签名
	Trusted OS 内容证书 1 …… Trusted OS 内容证书 n	包含 Trusted OS 镜像的 hash 值，由 Trusted OS 镜像的私钥签名。 Trusted OS 镜像可由不同的第三方提供，因此存在多个内容证书
	通用固件内容证书 1 …… 通用固件内容证书 n	包含通用固件镜像的 hash 值，由通用固件的私钥签名。 通用固件镜像可由不同的第三方提供，因此存在多个内容证书

5.2 可信启动流程

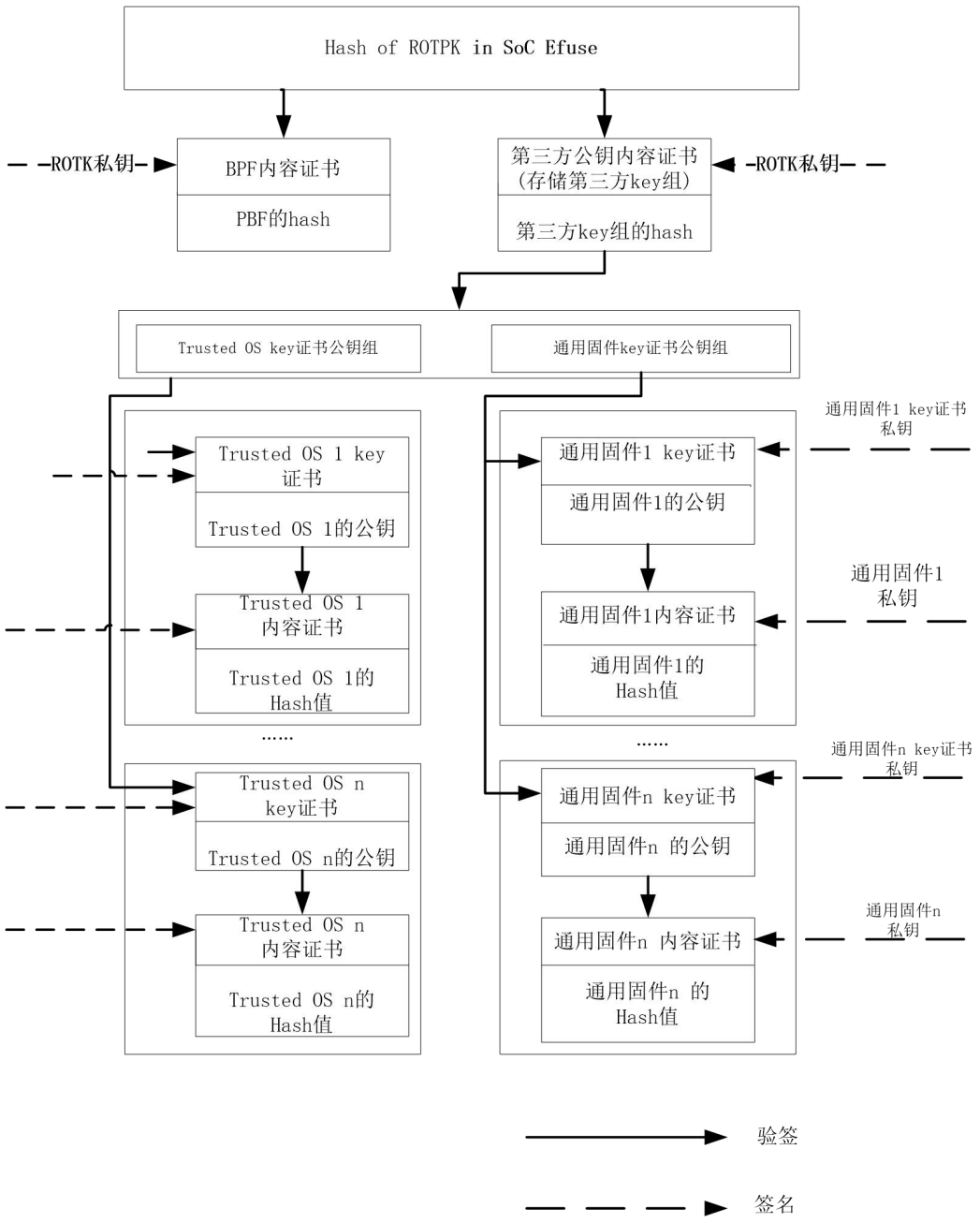
Hash of ROTPK in SoC Efuse
ROTK私钥
BPF内容证书
PBF的hash
第三方公钥内容证书
(存储第三方key组)
第三方key组的hash
ROTK私钥
Trusted OS key证书公钥组
通用固件key证书公钥组
Trusted OS 1 key
证书
Trusted OS 1的公钥
Trusted OS 1
内容证书
Trusted OS 1的
Hash值
.....
Trusted OS n
key证书
Trusted OS n的公钥
Trusted OS n
内容证书
Trusted OS n的
Hash值
通用固件1 key证书
私钥
通用固件1 key证书
通用固件1的公钥
通用固件1
私钥
通用固件1内容证书
通用固件1的
Hash值
.....
通用固件n key证书
私钥
通用固件n key证书
通用固件n 的公钥
通用固件n
私钥
通用固件n 内容证书
通用固件n 的
Hash值
验签
签名

图 5-1 可信启动流程

6 可信执行环境

可信执行环境（Trusted Execution Environment，TEE）是指在处理器内部通过硬件资源隔离等方式建立一个独立的安全区域。与 TEE 对应的为 REE（Rich Execution Environment），REE 与 TEE 在硬件上完全隔离。但是 TEE 具有更高的安全级别，REE 不能访问 TEE 的相关资源，以此保护 TEE 内部代码和敏感信息的安全性。TEE 与 REE 之间有相应的通信机制，REE 可以请求 TEE 为其提供安全服务。同时 TEE 所有出片数据都必须经过加密，不得明文出片。

腾锐 D2000 八核 安全 CPU 内置实现了 TZC（TrustZone Controller）、AES（Advanced Encryption Standard）模块、设备安全等硬件隔离机制，用以构建可信执行环境。

6.1 TZC

TZC（TrustZone Controller）为兼容 TrustZone 协议的存储器安全管理部件，可将地址空间划分为 8 个安全域空间和 1 个普通空间，并对不同空间设置不同的访问权限。TZC 负责对访问 memory 的访问请求进行安全权限检查，屏蔽不合权限的访问操作，以保障系统存储数据的安全。

6.1.1 Secure region

每一个域都有一个安全权限，访问请求只有匹配对应域的安全权限，才可以访问请求的资源。

过滤单元（filter unit）一共包含 8 个安全域（1-8）及一个默认域 0，8 个安全域的起始地址、结束地址及使能信号都可以通过软件进行配置；默认域 0 的起始地址均为 0，结束地址是由实际的物理地址空间决定的。8 个安全域的地址空间之间不可以有重叠部分，对重叠域的访问结果是未知的，但 8 个安全域可以与默认域 0 的地址空间重叠，因为过滤单元首先检测的是 8 个安全域，只有在 8 个安全域的安全权限均不匹配时才使用默认域 0。

TZC 可以通过安全域相应的控制寄存器过滤访问请求：是否允许安全读、是否允许安全写以及非安全请求的 master。

6.1.2 bypass

TZC 支持 bypass 功能，通过配置 BY_PASS 寄存器，可以使对请求通过或绕过过滤单元。需要注意的是：由于在控制寄存器配置期间，TZC 屏蔽的仅仅是 AW 通道与 AR 通道，所以在配置 bypass 寄存器期间，不允许 TZC 模块内还存在未处理的请求或数据，且不能向

TZC 模块发送 AXI 请求，否则会致使系统出现死锁或崩溃。因此建议仅在系统上电时配置一次 bypass 寄存器。

6.1.3 寄存器说明

LMU0TZC 的基址：TZC_BASE = 0x28200200

LMU1TZC 的基址：TZC_BASE = 0x28201200

寄存器地址 = TZC_BASE + 寄存器的偏移地址。

TZC 寄存器列表如表 6-1。

表 6-1 TZC 寄存器列表

Offset	Name	Type	Reset	Width	Description
配置、控制、中断寄存器					
0x000	ACTION	RW	0x00000000	32	访问失败时控制中断和总线的返回信号
0x004	GATE_KEEPER	RW	0x00000000	32	阻止或允许对过滤单元的访问
0x008	BY_PASS	RW	0x00000000	32	控制是否绕过过滤单元
0x00C	INT_STATUS	RO	0x00000000	32	包含中断信号的状态
0x010	INT_CLEAR	WO	0x00000000	32	清中断
错误信息					
0x014	FAIL_ADDRESS_LOW	RO	0x00000000	32	第一次访问失败的低 32 地址[31:0]
0x018	FAIL_ADDRESS_HIGH	RO	0x00000000	32	第一次访问失败的高地址[AXI_ADDRESS_MSB – 1:32]
0x01C	FAIL_CONTROL	RO	0x00000000	32	第一次访问失败的状态控制信息
0x020	FAIL_ID	RO	0x00000000	ACE ID Width	第一次访问失败的 AXI ID，如果 INT_STATUS 中的状态位已被设置，则不会新相应的失败状态组寄存器

域控制寄存器					
0x024	REGION_BASE_0	RO	0x00000000	32	域 0 的底部边界[AXI_ADDRESS_WIDTH-13:0]
0x028	REGION_TOP_0	RO	0xFFFFFFFF	32	域 0 的顶部边界地址[AXI_ADDRESS_WIDTH-13:0]
0x02C	REGION_ATTRIBUTES_0	RW	0xC0000001	32	表明是否允许对该域进行安全写操作
0x030	REGION_ID_ACCESS_0	RW	0xFFFFFFFF	32	NSAID 对应的普通请求读写访问权限
0x030+(0x4*n)	REGION_BASE_<n>	RW	0x00000000	32	域 n 的底部边界[AXI_ADDRESS_WIDTH-13:0]
0x050+(0x4*n)	REGION_TOP_<n>	RW	0x00000000	32	域 n 的顶部边界地址[AXI_ADDRESS_WIDTH-13:0]
0x070+(0x4*n)	REGION_ATTRIBUTES_<n>	RW	0x00000000	32	控制域 1-8 的访问权限及域使能
0x090+(0x4*n)	REGION_ID_ACCESS_<n>	RW	0x00000000	32	控制基于 NSAID 的普通请求的访问权限

注：NSAID 指的是芯片每个主设备的编号

6.1.3.1 ACTION (0x000)

Bits	Name	Access	Reset	description
[31:2]	reserved			
[1:0]	reaction_value	RW	2’b00	在访问失败（权限不足、域重叠）时控制 bresps[1:0]、rresps[1:0]及 TZCINT 信号的值 2’b00 Sets TZCINT LOW and issues an OKAY response. 2’b01 Sets TZCINT LOW and issues a DECERR

				response. 2’b10 Sets TZCINT HIGH and issues an OKAY response. 2’b11 Sets TZCINT HIGH and issues a DECERR response

6.1.3.2 GATE_KEEPER (0x004)

Bits	Name	Access	Reset	Description
[31:17]	reserved			
[16]	open_status	RO	1’b0	过滤单元中的 gate keeper 的当前状态(由 open_request 得到)， 1’b1：允许访问过滤单元 1’b0: 拒绝访问过滤单元
[15:1]	reserved			
[0]	open_request	RW	1’b0	软件控制 gate keeper 功能是否打开,若使 TZC 模块正常工作，在所有需要配置的寄存器配置完成之后，该寄存器必须配置为 1，否则 TZC 将不接受 AW 通道与 AR 通道内的请求。 1’b1: open 1’b0: closed

6.1.3.3 BY_PASS (0x008)

Bits	Name	Access	Reset	Description
[31:1]	reserved			
[0]	by_pass	RW	1’b0	是否绕过过滤单元 1’b1: 绕过过滤单元 1’b0: 通过过滤单元

6.1.3.4 INT_STATUS (0x00C)

Bits	Name	Access	Reset	Description
[31:17]	reserved			
[16]	overlap	RO	1’b0	地址域重叠（不考虑同域 0 之间的重叠），若该位被置为 1，则 status 位也应该被置为 1。即使 ACTION 设置不驱动中断，该位也会被设置。清理 status 位也会清理该位。 1’b1:表示不同地址域之间存在重叠 1’b0: 表示地址域之间均不重叠
[15:9]	reserved			
[8]	overrun	RO	1’b0	距离上次清理中断之后又发生两次以上访问权限不足或者域重叠。即使 ACTION 设置不

				驱动中断，该位也会被设置。清理 status 位也会清理该位。 1’b1:表示存在两次以上访问权限不足或者域重叠 1’b0: 表示访问权限不足或者域重叠的次数为 0 或 1
[7:1]	reserved			
[0]	status	RO	1’b0	中断状态。即使 ACTION 设置不驱动中断，该位也会被设置。 0b1: 发生非法访问或域重叠访问，等待被清理 0b0: 未发生非法访问及域重叠访问

6.1.3.5 INT_CLEAR (0x010)

Bits	Name	Access	Reset	Description
[31:1]	reserved			
[0]	clear	WO	1’b0	中断清理控制位 0b1: 清理 INT_STATUS 寄存器中的 status、overrun、overlap 位 0b0: 不清理中断

6.1.3.6 FAIL_ADDRESS_LOW (0x014)

Bits	Name	Access	Reset	Description
[31:0]	addr_status_low	RO	32’b0	用来存放第一次或清理中断后的第一次访问失败的 AXI 请求地址中的[31:0]位，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该寄存器

6.1.3.7 FAIL_ADDRESS_HIGH (0x018)

Bits	Name	Access	Reset	Description
[31:N+1]	reserved			
[N:0]	addr_status_high	RO	全 0	用来存放第一次或清理中断后的第一次访问失败的 AXI 请求地址中的[N+32:32]位，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该寄存器。

注：N=AXI_ADDRESS_MSB – 33

6.1.3.8 FAIL_CONTROL (0x01C)

Bits	Name	Access	Resetvalue	Description
[31:25]	reserved			
[24]	direction	RO	1'b0	标识非法请求是写请求还是读请求，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该位。 1'b1: Write access 1'b0: Read access
[23:22]	reserved			
[21]	non_secure	RO	1'b0	标识非法请求是安全请求还是普通请求，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该位。 1'b1: Non-secure access 1'b0: Secure access
[20]	privileged	RO	1'b0	标识非法请求是特权访问还是非特权访问，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该位。 1'b1: Privileged access 1'b0: Unprivileged access
[19:0]	reserved			

6.1.3.9 FAIL_ID (0x020)

Bits	Name	Access	Reset	Description
[31:N+1]	reserved			
[N:0]	id	RO	全 0	用来存放第一次或清理中断后的第一次访问失败的 AXI 请求的 ID 值，如果 INT_STATUS 寄存器中的 status 位已被设置，则不更新该寄存器。

注：N = AID_WIDTH – 1

6.1.3.10 REGION_BASE_0 (0x024)

Bits	Name	Access	Reset	Description
[31:N+1]	reserved			
[N:0]	base_address_0	RO	全 0	默认域域 0 底部边界只能是 4KB 对齐的，故该寄存器中存储的是 12 及其更高位,每一位均是 0，且只读。

N = AXI_ADDRESS_WIDTH-13

6.1.3.11 REGION_TOP _0 (0x028)

Bits	Name	Access	Reset	Description
[31:N+1]	reserved			
[N:0]	top_address_0	RO	全 0	默认域域 0 底部边界只能是 4KB 对齐的，故该寄存器中存储的是 12 及其更高位,每一位均是 1，且只读。

N = AXI_ADDRESS_WIDTH-13

6.1.3.12 REGION_ATTRIBUTES_0（0x02C）

Bits	Name	Access	Reset	Description
[31]	s_wr_en_0	RW	1'b1	表明是否允许对 0 域进行安全写操作 1'b1:允许安全写操作 1'b0:不允许允许安全写操作
[30]	s_rd_en_0	RW	1'b1	表明是否允许对 0 域进行安全读操作 1'b1:允许安全读操作 1'b0:不允许安全读操作
[29:1]	reserved			
[0]	region_en_0	RW	1'b1	是否允许使用 0 域 1'b1:可以使用 1'b0:该域不存在

6.1.3.13 REGION_ID_ACCESS_0 (0x030)

NSAID（Non-secure Access IDentity）唯一的标明发出请求的 master 或一组 master 的身份。其默认值为 0。使用默认值是因为某一个 master 没有被分配 NSAID 或者不需要特别的与其他的 master 隔离。

Bits	Name	Access	Reset	Description
[31:16]	nsaid_wr_en_0	RW	全 1	默认域 0 的普通访问写使能，每一位表明了 NSAID 对应的写访问权限 Bit16 NSAIDW=0 Bit17 NSAIDW=1 …… Bit31 NSAIDW=15
[15:0]	nsaid_rd_en_0	RW	全 1	默认域 0 的普通访问读使能，每一位表明了 NSAID 对应的读访问权限 Bit0 NSAIDR=0 Bit1 NSAIDR=1 ……

				Bit15 NSAIDR=15

6.1.3.14 REGION_BASE_<n>(0x030+(0x4*n))

Bits	Name	Access	Reset	Description
[N:0]	base_address_<n>	RW	全 0	安全域 n 底部边界地址只能是 4KB 对齐的，故该寄存器中存储的是边界地址的 12 位及其更高位。即配置这些寄存器时，apb 的 pdata[N:0]对应的是底部边界的[N+12:12]

注：N = AXI_ADDRESS_MSB-13 ， n 为 1-8

6.1.3.15 REGION_TOP_<n>(0x050+(0x4*n))

Bits	Name	Access	Reset	Description
[N:0]	top_address_<n>	RW	全 0	安全域 n 顶部边界地址只能是 4KB 对齐的，故该寄存器中存储的是边界地址的 12 位及其更高位。即配置这些寄存器时，apb 的 pdata[N:0]对应的是顶部边界的[N+12:12]

注：N = AXI_ADDRESS_MSB-13 , n 为 1-8

6.1.3.16 REGION_ATTRIBUTES_<n> (0x070+(0x4*n))

Bits	Name	Access	Reset	Description
[31]	s_wr_en_<n>	RW	1'b0	表明是否允许对该域进行安全写操作 1'b1: 允许安全写操作 1'b0:不允许安全写操作
[30]	s_rd_en_<n>	RW	1'b0	表明是否允许对该域进行安全读操作 1'b1: 允许安全读操作 1'b0:不允许安全读操作
[29:1]	reserved			
[0]	region_en_<n>	RW	1'b0	是否允许使用该域 1'b1:可以使用 1'b0:该域不存在

注：n 为 1-8

6.1.3.17 REGION_ID_ACCESS_<n> (0x090+(0x4*n))

NSAID（Non-secure Access IDentity）唯一的标明发出请求的 master 或一组 master 的身份。其默认值为 0。使用默认值是因为某一个 master 没有被分配 NSAID 或者不需要特别的与其他的 master 隔离。

Bits	Name	Access	Reset	Description
[31:16]	nsaid_wr_en_<n>	RW	全 0	NSAID 写使能，每一位表明了

				NSAID 对应的写访问权限 Bit16 NSAIDW=0 Bit17 NSAIDW=1 …… Bit31 NSAIDW=15
[15:0]	nsaid_rd_en_<n>	RW	全 0	NSAID 读使能，每一位表明了 NSAID 对应的读访问权限 Bit0 NSAIDR=0 Bit1 NSAIDR=1 …… Bit15 NSAIDR=15

注：n 为 1-8

6.2 AES 模块

AES（Advanced Encryption Standard）模块为对存储器访问数据进行加解密的处理部件。AES 模块具有独立的加密和解密通道，通过该模块，可以完成 NOC 到 SDRAM/DIMM 数据的加密、从 SDRAM/DIMM 到 NOC 数据的解密。同时，可以通过 bypass 寄存器控制数据是否需要加解密，不需要加解密的数据将绕过模块内的加解密通道。

6.2.1 bypass

由于 AES 模块会带来一些额外的 cycle，对芯片的性能有一定的影响。对安全性没有要求的用户可以通过配置 bypass 寄存器，关闭 AES 加速引擎，使得 AES 加速引擎对芯片的性能没有影响。

Bypass 一共有三种模式，全加密、安全域加密且普通域不加密、全不加密。前两种方式很类似，如果只需要对安全域内的数据进行防护，此模式下当在安全域的数据远远小于普通域的数据时，由 AES 带来的性能损耗将非常低，而全不加密将不能保证安全域内的数据的安全，此时绕过 AES 的加解密通道，对芯片的性能没有影响。

6.3 设备安全

腾锐 D2000 八核安全 CPU 内置了设备安全机制，可以对设备的安全属性进行配置。当设备的安全属性被设置“安全”（secure）时，只有安全访问请求才可以访问该设备；而当设备的安全属性设置为“非安全”（non-secure）时，安全访问请求和非安全访问请求都可以访问该设备。

安全态系统软件可以在系统运行时动态修改设备的安全属性。

腾锐 D2000 八核安全 CPU 提供一组寄存器来设置设备的安全属性，每个寄存器对应一

组设备，组内的每个设备对应该寄存器的 1 位。如果将控制位设 1，设备的安全属性被配置为“非安全”；控制位设 0，则设备的安全属性被配置为“安全”。配置寄存器的安全属性控制位与设备的对应关系如下表所示。

表 6-2 设备安全属性配置寄存器表

设备	寄存器地址	控制位
pcie	0x29F00028	0
gpio1	0x29F00008	0
uart0		1
uart1		2
uart2		3
uart3		4
i2c0		5
i2c1		6
i2c2		7
i2c3		8
保留		9-10
wdt0		11
wdt1		12
保留		13
spi0		14
gpio0		15
qspi	0x29F0001c	0
qspi_reg		1
lpc	0x29F00018	0
保留		1-4
hds		5
gmac0		6
gmac1		7

spi1		8
保留		9
otp		10
保留		11-14
scto	0x29F00030	0

设备的具体使用说明，参见《腾锐 D2000 八核软件编程手册》[3]。

7 固件管理

固件管理就是指在固件的版本更新过程中，对固件内容、固件版本、固件备份等进行管理。

飞腾平台固件由三部分组成：飞腾可信根 PBR（Phytium Boot Rom）、飞腾基础固件 PBF(Phytium Base Firmware)和第三方固件。其中，PBR 固件在芯片出厂时固定，无法更新。PBF 固件由飞腾负责维护。第三方固件由整机厂商进行维护。

PBF 和第三方固件更新后，执行可信启动流程能够确保固件不被篡改，同时在启动过程中会比较 Efuse 中的防回滚寄存器的计数值与存储在 FLASH 中当前固件版本号，当前固件版本号必须大于 Efuse 中的防回滚寄存器的计数值，才能正常启动。

Efuse 的基址为 0x2820A000，防回滚寄存器如下表：

表 7-1 防回滚寄存器

名称	存储地址	存储位宽	描述
anti-rollback counter	0x108~0x118	160bits	anti-rollback counter，防回滚计数器。

8 执行环境间通信

PSPA 包含两个执行环境：可信执行环境 TEE 和通用执行环境 REE，参见图 2-1。可信执行环境和通用执行环境间可以通过 SMC 指令和中断两种方式进行交互。

8.1 SMC 指令交互

SMC 指令交互主要指 CPU 调用 SMC 指令在可信执行环境和通用执行环境间切换，并通过 cpu 通用寄存器传递参数。例如通用操作系将安全功能参数保存在 CPU 通用寄存器中，然后调用 SMC 指令进到可信操作系统中，并根据安全功能参数执行相关安全服务。功能参数是一个 32bit 的数据类型，保存在 CPU 通用寄存器 R0 中，其定义必须符合 ARM TrustZone 规范[4]。

表 8-1 SMC 函数 ID

位域	屏蔽位	功能描述
31	0x80000000	0：表示标准调用； 1：表示快速调用；执行过程不能被中断
30	0x40000000	O:表示 SMC32 调用； 1:表示 SMC64 调用；
29:24	0x3F000000	0-47：预留； 48-49: 可信应用程序使用的 ID 范围； 50-63：可信操作系统使用的 ID 范围；
23-16	0x00FF0000	当 bit31 为 1 时，该域必须位 0； 当 bit31 为 0 时，该域的值未定义；
15-0	0x0000FFFF	用户自定义，每个值对应一个功能；

8.2 可信中断交互

当两个核分属于不同的执行环境时，可以基于中断方式彼此通信。运行在可信执行环境中的处理器核称为可信核，运行在通用执行环境中的处理器核称为计算核，可信核与计算核之间，可以通过中断和共享内存相结合的方式进行信息交互。共享内存的划分由用户自定义，中断约定采用固定的 SGI 中断号进行交互。可信核向计算核发送 7 号中断，计算核向可信核发送 15 号中断。

8.2.1 中断配置

8.2.1.1 可信核(S-EL1)下的中断配置

- 配置 GICR_NSACR[5]寄存器的 Bit31-30 为 b’10，允许非安全态 CPU 发送 15 号安全中断；
- 配置 DAIF[6]的 I 位为 0，表示不屏蔽 IRQ 中断。

8.2.1.2 计算核 EL2 固件下的中断配置

- 配置 HCR_EL2 寄存器的 IMO 位为 1，目的是让中断路由到 EL2；引导 OS 前，必须将该位清 0。
- 配置 DAIF 的 I 位为 0，表示不屏蔽 IRQ 中断；

8.2.1.3 计算核 EL1 Kernel 的中断配置

- 采用 Kernel 缺省配置即可，无需额外定制。

8.2.2 中断触发

8.2.2.1 可信核向计算核发送中断

- 写 SGI 发送寄存器 ICC_ASGI1R_EL1[5]；

2000/4CPU 共 4 个核，分成 2 个 cluster，每个 cluster 包含 2 个核。即 Core0-1 属于 cluster 0，Core 2-3 属于 cluster 1。如图 8-1 所示，Aff1 为目标 CPU 的 Cluster 号，TargetList 为目标 CPU 在 Cluster 内的 ID，即 0-1。INTID 为 SGI 中断号。

图 8- 1 ICC_ASGI1R_EL1

8.2.2.2 计算核向可信核发送中断

- 写 SGI 发送寄存器 ICC_SGI1R_EL1[5]，寄存器格式参见 ICC_ASGI1R_EL1。

参考文献

[1] Phytium,《Phytium Security Platform Architecture 白皮书》，v1.0.

[2] Phytium,《Phytium Security Platform Architecture 规范》，v1.0.

[3] Phytium, 《腾锐 D2000 软件编程手册》，v1.1

[4] ARM,《SMC CALLING CONVENTION System Software on ARM Platforms》, Issue B.

[5] ARM, 《Generic Interrupt Controller Architecture Specification GIC architecture version 3.0 and version 4.0》, Issue C.

[6] ARM, 《ARM Architecture Reference Manual ARMv8, for ARMv8-A architecture profile》, Issue C.a.

[7] Phytium, 《腾锐 D2000 安全 CPU 密码模块 API 手册》，v1.0