用户手册

安全密钥管理工具

用户手册

此等资料中所含的包括产品和产品规范在内的所有信息均为资料发布时的产品信息，信息将不时由 Renesas Electronics 公司更改，恕不另行通知。请通过包括 Renesas Electronics 公司网站 (http://www.renesas.com) 在内的各种方式了解 Renesas Electronics 公司发布的最新信息。

Renesas Electronics
www.renesas.com

版本 1.09 2025 年 7 月

注意

1. 本文档中电路、软件和其他相关信息的描述仅用于说明半导体产品的操作和应用示例。用户应对产品或系统设计中电路、软件和信息的纳入或任何其他用途承担全部责任。对于您或第三方因使用这些电路、软件或信息而引起的任何损失和损害，Renesas Electronics 不承担任何责任。
2. Renesas Electronics 特此声明，对于因使用本文档中所述的 Renesas Electronics 产品或技术信息（包括但不限于产品数据、图纸、图表、程序、算法和应用示例）而引起的侵权或与第三方有关的专利、版权或其他知识产权的任何其他索赔，概不承担任何责任和赔偿。
3. 对Renesas Electronics 或其他公司的任何专利、版权或其他知识产权均不授予任何明示、暗示或其他形式的许可。
4. 您应负责确定需要从任何第三方获得哪些许可，并在需要时为合法进口、出口、制造、销售、使用、分销或以其他方式处置包含 Renesas Electronics 产品的任何产品获得此类许可。
5. 不得对 Renesas Electronics 产品的全部或部分进行更改、修改、复制或逆向工程。对于因更改、修改、复制或逆向工程而导致您或第三方蒙受的任何损失或损害，Renesas Electronics 不承担任何责任。
6. Renesas Electronics 产品根据以下两个质量等级进行分类：“标准”和“优质”。Renesas Electronics 每种产品的预期应用取决于产品的质量等级，具体如下所示。
   “标准”：计算机、办公设备、通信设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备、工业机器人等
   “优质”：运输设备（汽车、火车、轮船等）；交通管制（交通信号灯）；大型通信设备；关键金融终端系统；安全控制设备等
   除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中明确指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 产品不适合或不授权用于可能对人类生命构成直接威胁或造成人身伤害（人造生命支持设备或系统；手术植入物等），或者可能造成严重的财产损失（空间系统、海底中继器、核动力控制系统、飞机控制系统、关键设备系统、军事装备等）的产品或系统。对于因使用任何与 Renesas Electronics 数据手册、用户手册或其他 Renesas Electronics 文档不一致的 Renesas Electronics 产品而引起的您或任何第三方所造成的任何损坏或损失，Renesas Electronics 不承担任何责任。
7. 没有任何半导体产品是绝对安全的。尽管 Renesas Electronics 的硬件或软件产品中可能实施了任何安全措施或功能， Renesas Electronics 对因任何漏洞或侵骇（包括但不限于以任何未经授权的方式访问或使用 Renesas Electronics 产品或使用 Renesas Electronics 产品的系统）而产生的任何后果概不负责。RENESAS ELECTRONICS 不担保或保证 RENESAS ELECTRONICS 产品或使用 RENESAS ELECTRONICS 产品创建的任何系统不会被破坏，或者可免于数据损坏、攻击、病毒、干扰、黑客攻击、数据丢失或失窃或其他安全入侵（“漏洞问题”）。RENESAS ELECTRONICS 不承担由任何漏洞问题引起的或与之相关的任何和所有责任或义务。此外，在适用法律允许的范围内，RENESAS ELECTRONICS 不对本文件和任何相关或附带的软件或硬件提供任何和所有明示或暗示的保证，包括但不限于对适销性或对特定用途的适用性的暗示保证。
8. 使用 Renesas Electronics 产品时，请参见最新的产品信息（数据手册、用户手册、应用笔记、可靠性手册中的“处理和使用半导体器件的一般说明”等），并确保使用条件符合 Renesas Electronics 在最大额定值、工作电源电压范围、散热特性和安装等方面的规定。对于因在超出上述规定范围的条件下使用 Renesas Electronics 产品而引起的任何失常、故障或事故，Renesas Electronics 不承担任何责任。
9. 尽管 Renesas Electronics 致力于提高 Renesas Electronics 产品的质量和可靠性，但半导体产品具有特定的特性，例如在特定速率下发生故障以及在某些使用条件下出现故障。除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 的产品将不受抗辐射设计的约束。用户应负责采取安全措施，以防止人身伤害、火灾造成的伤害，和/或因 Renesas Electronics 产品发生故障或失常而对公众造成的危险，例如硬件和设备的安全设计，包括但不限于冗余、火控和故障预防、针对老化退化的适当处理或任何其他适当的措施。由于对微型计算机软件进行评估非常困难且无实操性，因此用户有责任评估自己生产的最终产品或系统的安全性。
10. 请联系 Renesas Electronics 销售办事处，以获取有关环境事宜的详细信息，例如每个 Renesas Electronics 产品的环境相容性。用户有责任认真、充分地研究有关纳入或使用受控物质的适用法律和法规（包括但不限于欧盟 RoHS 指令），并按照所有适用法律和法规使用 Renesas Electronics 产品。对于因您未遵守适用的法律和法规而造成的损坏或损失，Renesas Electronics 不承担任何责任。
11. Renesas Electronics 产品和技术不得被用于或纳入为任何适用的本国或外国法律、法规所禁止制造、使用或销售的产品或系统范围内。用户应遵守由对当事方或交易拥有管辖权的任何国家/地区的政府颁布和管理的任何可适用的出口控制法律和法规。
12. 应由Renesas Electronics 产品的购买方或分销商，或者对产品进行分发、处置或以其他方式出售或转让给第三方的任何其他当事方，负责将本文档中阐明的内容和条件提前通知前述第三方。
13. 未经 Renesas Electronics 事先书面同意，不得以任何形式全部或部分重印、再现或复制本文档。
14. 如果对本文档中包含的信息或 Renesas Electronics 产品有任何疑问，请联系 Renesas Electronics 销售办事处。

（注 1）本文档中的“Renesas Electronics”是指 Renesas Electronics Corporation，也包括其直接或间接控制的子公司。
（注 2）“Renesas Electronics 产品”是指 Renesas Electronics 开发或制造的任意产品。

（版本 5.0 2020 年 10 月 1 日）

公司总部

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

联系信息

有关产品、技术、文档最新版本或离您最近的销售办事处的更多信息，请访问：www.renesas.com/contact/。

商标

Renesas 和 Renesas 徽标是 Renesas Electronics Corporation 的商标。所有商标和注册商标都是各自所有者的财产。

© 2025 Renesas Electronics Corporation. All rights reserved.

有关微处理器和微控制器产品处理的一般预防措施

下方的使用说明适用于 Renesas 的所有微处理器和微控制器产品。有关本文档所述产品的详细使用说明，请参见本文档的相关章节以及针对相应产品发布的各项技术更新。

1. 防静电 (ESD) 措施

   当暴露于 CMOS 器件时，强电场会导致栅极氧化物损坏，并最终使器件的工作性能下降。必须采取措施来尽可能地阻止静电的产生，并在发生静电时将其迅速消除。必须充分进行环境控制。如果环境干燥，应使用加湿器。建议避免使用容易产生静电的绝缘体。半导体器件必须在防静电容器、静电屏蔽袋或导电材料中存储和运输。所有测试和测量工具（包括工作台和地板）都必须接地。操作员还必须使用腕带接地。禁止徒手触摸半导体器件。对于装有半导体器件的印刷电路板，必须采取类似的预防措施。

2. 上电时的处理

   产品在上电时仍处于未定义状态。在上电时，LSI 内部电路状态未定，寄存器设置和引脚的状态都未定义。对于复位信号已施加到外部复位引脚的成品，从上电开始到复位过程完成之前，引脚的状态并不能确定。同样地，对于以片上上电复位功能复位的产品，从上电开始到达到指定的复位电位之前，都无法确保引脚的状态。

3. 断电状态下的信号输入

   器件断电时，请勿输入信号或 I/O 上拉电源。输入此类信号或 I/O 上拉电源导致的电流注入可能导致故障，此时流经器件的异常电流可能导致内部元件性能下降。请遵循产品文档中所述的电源关闭状态下输入信号的准则。

4. 未用引脚的处理

   未使用的引脚应根据本手册“未用引脚的处理”部分给出的说明处理。CMOS 产品的输入引脚通常为高阻抗状态。在与开路状态下的未用引脚配合使用时，将在 LSI 周围产生额外的电磁噪声，内部将产生相关的直通电流，从而可能因为错误地将引脚状态识别为输入信号而产生故障。

5. 时钟信号

   在应用复位后，只有在作业时钟信号稳定后再释放复位线。当在程序执行过程中切换时钟信号时，等待目标时钟信号稳定。如在复位时外部谐振器（或外部振荡器）生成时钟信号，确保在时钟信号完全稳定后再释放复位线。此外，如在程序执行过程中切换到外部谐振器（或外部振荡器）生成的时钟信号，也需要等待目标时钟信号稳定。

6. 输入引脚上的电压施加波形

   输入噪声或反射波引起的波形失真可能会导致故障。例如，如果由于噪声而使 CMOS 器件的输入处于 V_{IL}（最大值）和 V_{IH}（最小值）之间，则器件可能会发生故障。当输入电平固定时，以及在输入电平通过 V_{IL}（最大值）和 V_{IH}（最小值）之间时的过渡期间，请注意防止颤动噪声进入器件。

7. 禁止访问保留地址

   禁止访问保留地址。保留地址用于未来可能的功能扩展。请勿访问此类地址，否则将无法保证 LSI 能正确运行。

8. 产品差异

   在从一个产品切换到另一个前（例如，切换到具有不同零件号的产品），确认切换不会引起问题。对于同在一个系列但零件号不同的微处理器单元或单片机产品，其内部存储能力、布局图案和其他特性因素可能有所差异，进而可能会影响特性值、运行裕量、抗干扰性和噪声辐射量等电气特征的范围。在更换为部件编号不同的产品时，应针对给定产品执行系统评估测试。

如何使用本手册

1. 目的和目标读者

本手册旨在帮助用户了解安全密钥管理工具的功能。目标受众是使用安全加密引擎、可信安全引擎和 RSIP-Exxx（瑞萨微控制器内置的安全引擎）的系统设计人员和开发人员。要使用本手册，您需要具备微控制器以及 Windows、Linux 和 macOS 的基本知识。

在使用本软件之前，务必充分确认所用单片机的手册中的内容。

2. 约定

注：文本中标有“注”的项目的脚注。

数字表示：二进制 ... xxxx 或 xxxxB

十进制 ... xxxx

十六进制 ...0xXXXX 或 xxxxH

“　　”：屏幕上可以选择或输入的任何字符或项目

[　　]：屏幕上命令、对话框、选项卡页、选项或区域的名称

3. 术语

术语	含义
CLI	命令行接口
DLM 密钥	产品生命周期管理用密钥，部分 MCU/MPU 支持 DLM Key
DOTF	读取和执行外部闪存 ROM 等中写入的加密映像时，可直接解密的功能。 有些 MCU/MPU 称之为 "Decryption On-The-Fly（DOTF）"或 "On-The-Fly Decryption（OTFD）"。在本文中，它被称为 DOTF。
加密 KUK	密钥更新 由 UFPK 封装的密钥
加密的用户密钥	使用 UFPK 或 KUK 封装的用户密钥
FSBL	第一阶段引导加载程序 验证 OEM 引导加载程序的合法性所需的程序
GUI	图形用户界面
HRK	硬件根密钥 每个 MCU 产品家族特有的密钥数据
HUK	硬件唯一密钥 每个 MCU 唯一的密钥数据
KUK	密钥升级用密钥 用于升级用户密钥的密钥
OEM 引导加载程序	客户创建的设备启动后最先执行的程序
PEM	Base64 编码的 X509 ASN.1 密钥文件 该工具支持读取 Openssl genrsa 命令或 ecparam -genkey 命令生成的密钥
瑞萨密钥文件	瑞萨电子密钥文件 RFP 使用的密钥格式。格式请参见附录。
瑞萨密钥封装服务	使用器件固有密钥封装 UFPK 并生成 W-UFPK 的服务 https://dlm.renesas.com/keywrap/
RFP	瑞萨闪存编程器 https://www.renesas.com/rfp
RSIP	瑞萨安全 IP 在 RA 系列、RX 系列和 RZ 系列等中实现的安全 IP。
RSU	瑞萨安全更新 RX 产品家族 Firmware Update FIT(R01AN5824)中定义的要更新的程序映像文件格式。
SCE	安全加密引擎 RA 产品家族和 Synergy 平台产品内置的安全加密 IP
SFP	安全工厂编程 解密瑞萨电子 MCU 的启动固件的加密映像并同时写入器件的功能

术语	含义
TSIP	可信安全引擎 RX 和 RZ 产品家族内置的安全加密 IP
UFPK	用户工厂烧录用密钥 用于封装用户密钥和 DLM 密钥的密钥
用户密钥	用户应用中使用的加密密钥 AES 密钥、RSA 公钥、RSA 私钥等
W-UFPK	使用瑞萨密钥封装服务封装的 UFPK
封装的用户密钥	通过 SCE 或 TSIP 重新封装的用户密钥，可用于 SCE 保护模式或 TSIP

目录

1. 瑞萨密钥管理系统 10
1.1 信任根简介 10
1.2 瑞萨安全引擎和关联密钥简介 10
1.3 瑞萨安全密钥安装的优势 12
1.3.1 密钥封装与密钥加密相比的优势 12
1.3.2 使用 HUK 的密钥封装的优势 13
1.4 封装密钥安装程序概述 13
1.4.1 安全密钥注入的一般步骤 13
1.4.2 安全密钥升级的一般步骤 16
1.5 瑞萨安全功能 18
1.5.1 第一阶段引导加载程序 18
1.5.2 实时解密 21
1.5.3 安全工厂编程 22

2. 概述 23
2.1 特性 23
2.1.1 注入/更新安全密钥 23
2.1.2 安全密钥管理工具支持的安全功能 25
2.2 运行环境 26
2.2.1 硬件环境 26
2.2.2 软件环境 26

3. GUI 功能说明 27
3.1 主窗口 27
3.2 菜单栏 29
3.2.1 [文件] 菜单 29
3.2.2 [视图] 菜单 29
3.2.3 [帮助] 菜单 29
3.3 [概要] 选项卡 30
3.4 [生成 UFPK] 选项卡 31
3.5 [生成 KUK] 选项卡 32
3.6 [封装密钥] 选项卡 33
3.6.1 [密钥类型] 选项卡 34
3.6.2 [密钥数据文件] 选项卡 37
3.6.2.1 在 [密钥类型] 选项卡中选择了 DLM/KUK/AES/TDES/ARC4/ChaCha20-Poly1305/ECC 私钥时 37
3.6.2.2 在 [密钥类型] 选项卡中选择了 RSA 公钥时 38
3.6.2.3 在 [密钥类型] 选项卡中选择了 RSA 私钥时 39

3.6.2.4 在 [密钥类型] 选项卡中选择了 ECC 公钥时 40
3.6.2.5 在 [密钥类型] 选项卡中选择了 OEM Root public 41
3.6.3 封装密钥 42
3.6.4 IV 43
3.6.5 输出 44
3.7 [TSIP Update]选项卡 46
3.7.1 输出映像 47
3.7.2 固件映像/安全引导映像 47
3.7.3 [RSU 标头]选项卡 48
3.7.4 [加密地址范围]选项卡 49
3.7.5 [Image Encryption Key]选项卡 50
3.7.6 [IV]选项卡 51
3.7.7 输出 51
3.8 [FSBL]选项卡 52
3.8.1 编程验证方法 54
3.8.2 [证书]选项卡 55
3.8.3 [OEM 根密钥]选项卡 56
3.8.4 [OEM 引导加载程序密钥]选项卡 57
3.9 [DOTF/OTFD]选项卡 59
3.9.1 加密范围 60
3.9.2 目标地址 61
3.9.3 映像加密密钥 62
3.9.4 初始向量 IV 62
3.9.5 输出映像地址和内容 63
3.10 [SFP]选项卡 64
3.10.1 [固件映像]选项卡 67
3.10.2 [映像加密密钥]选项卡 68
3.10.3 [Nonce]选项卡 69
3.10.4 [AL2/SECDBG_KEY]选项卡 70
3.10.5 [AL1/NONSECDBG_KEY]选项卡 71
3.10.6 [Boundary]选项卡 72
3.10.7 [外部闪存区]选项卡 73

4. CLI 函数说明 74
4.1 命令行语法 74
4.2 命令 75
4.3 **genufpk** 命令选项 76
4.4 **genkuk** 命令选项 76
4.5 **genkey** 命令选项 77
4.5.1 **mcu** 选项 79
4.5.2 **keytype** 选项 80

4.5.3 **key** 选项 82
4.5.3.1 十六进制数据直接输入 82
4.5.3.2 文件输入 88
4.5.3.3 省略 **key** 选项 90
4.5.4 **filetype** 选项 91
4.5.4.1 **filetype** 的 **rfp** 选项 91
4.5.4.2 **filetype** 的 **csource** 选项 91
4.5.4.3 **filetype** 的 **bin** 选项 93
4.5.4.4 **filetype** 的 **mot** 选项 94
4.5.5 **fileadd** 选项 97
4.5.6 **bswap** 选项 97
4.5.7 **keyfileoutput** 选项 98
4.6 **enctsip**命令选项 99
4.6.1 **ver** 选项 101
4.6.1.1 指定 **ver 选项** 1 时 101
4.6.1.2 指定 **ver 选项** 2 时 103
4.6.2 **mode** 选项 105
4.6.3 **imgflg** 选项 108
4.6.4 **filetype** 选项 108
4.6.5 **df_ena** 选项 108
4.7 **gencert**命令选项 109
4.7.1 **mode** 选项 112
4.7.2 OEM 引导加载程序的签名或 CRC 运算对象区域 113
4.7.2.1 指定**oembl_size**选项时 113
4.7.2.2 仅指定**cfsize**选项时 114
4.8 **encdotf**命令选项 115
4.8.1 **keytype** 选项 117
4.8.2 **enckey** 选项 117
4.9 **encsfp**命令选项 118
4.9.1 **mcu** 选项 122
4.9.2 **trn** 选项 122
4.9.3 **prg** 选项 123
4.9.4 **boundary** 选项 123
4.9.4.1 对于参数输入 123
4.9.4.2 文件输入 123
4.9.5 extarea0/extarea1 选项 124
4.10 **calcresponse**命令 选项 125

5. 操作程序 126
5.1 独立 126
5.1.1 Windows 版本 126
5.1.1.1 GUI 版本 126

5.1.1.2 CLI 版本 127
5.1.2 Linux 版本 128
5.1.2.1 GUI 版本 128
5.1.2.2 CLI 版本 129
5.1.3 macOS 版本 130
5.1.3.1 GUI 版本 130
5.1.3.2 CLI 版本 131
5.2 e²studio 插件 132
5.2.1 安装 e²studio 插件版本 132
5.2.2 卸载 e2studio 插件版本 137

6. 使用示例 139
6.1 示例 1 – 在具有 TSIP 的 RX 产品家族 MCU 上安装 AES128 密钥 139
6.1.1 使用 GUI 版本 139
6.1.2 使用 CLI 版本 143
6.2 示例 2 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上安装升级密钥 144
6.2.1 使用 GUI 版本 144
6.2.2 使用 CLI 版本 148
6.3 示例 3 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上升级 RSA 2048 公钥 150
6.3.1 使用 GUI 版本 150
6.3.2 使用 CLI 版本 153
6.4 示例 4 – RX产品家族TSIP Secure Update时的使用方法 154
6.4.1 使用 GUI 版本的 Security Key Management Tool 时 154
6.4.2 使用 CLI 版本的 Security Key Management Tool 时 157
6.5 示例 5 - RA产品家族FSBL密钥证书/代码证书生成时的使用方法 158
6.5.1 使用 GUI 版本的 Security Key Management Tool 时 158
6.5.2 使用 CLI 版本的 Security Key Management Tool 时 162
6.6 示例6 – RA产品家族 使用 DOTF 时的程序加密方法 163
6.6.1 使用 GUI 版本的 Security Key Management Tool 时 163
6.6.2 使用 CLI 版本的 Security Key Management Tool 时 166
6.7 示例7 – RA产品家族 使用安全工厂编程功能时的程序加密方法 167
6.7.1 使用 GUI 版本的 Security Key Management Tool 时 168
6.7.2 使用 CLI 版本的 Security Key Management Tool 时 173

7. 注意事项 174
7.1 使用 Windows 环境时的显示设置 174
7.2 在Linux环境下使用e²studio插件版本的注意事项 175
7.3 在macOS环境下使用e²studio插件版本的注意事项 175
7.4 macOS 版本的限制 175
7.5 安全工厂编程功能的局限性 176
7.5.1 安全密钥管理工具 User data and write address/size 作成方法 176

7.6 单机版和插件版 在 [概要]选项卡中更改 MCU/MPU 和加密引擎时的注意事项。 177

8. 附录.....179

8.1 License 179

8.2 瑞萨密钥文件（Key File）的格式.....181

8.2.1 文本格式.....181

8.2.2 文件结构.....181

8.2.3 文件扩展名 181

8.2.4 密钥数据.....182

8.2.4.1 结构 182

8.2.5 加密密钥计算公式 183

8.3 安全工厂编程文件格式 184

8.3.1 文件格式.....184

8.3.1.1 Pre-Data Field 186

8.3.1.2 TLV Length Field 190

8.3.1.3 TLV Field 190

8.3.2 文件扩展名 191

8.4 密钥证书/代码证书.....192

8.4.1 密钥证书文件格式 192

8.4.2 在 **mode** “**signature**”下生成的代码证书文件格式.....193

8.4.3 在 **mode** “**crc**”下生成的代码证书文件格式.....194

8.4.4 CRC32 的计算公式 194

8.5 TSIP Update.....195

8.5.1 用户程序的加密公式.....195

附图目录

图 1-1 安全加密引擎概览 10
图 1-2 RX 可信安全引擎概览 11
图 1-3 密钥封装与密钥加密 12
图 1-4 使用 HUK 的密钥封装 13
图 1-5 使用 DLM 服务器封装 UFPK 14
图 1-6 使用 UFPK 加密用户密钥 14
图 1-7 通过串行编程接口注入用户密钥 15
图 1-8 使用 KUK 加密用户密钥 16
图 1-9 通过串行编程接口注入 KUK 16
图 1-10 使用 KUK 加密新用户密钥 17
图 1-11 升级用户密钥 17
图 1-12 安全系统的Chain of Trust 18
图 1-13 OEM引导加载程序工厂烧录时的Authenticity Check 19
图 1-14 设备启动时的FSBL动作 20
图 1-15 DOTF的MCU/MPU的内部系统总线示例 21
图 1-16 具有安全工厂编程功能MCU/MPU的系统示例 22
图 3-1 独立版本的主窗口 27
图 3-2 e²studio 插件版本的主窗口 28
图 3-3 [概要] 选项卡 30
图 3-4 [生成 UFPK] 选项卡 31
图 3-5 [生成 KUK] 选项卡 32
图 3-6 [封装密钥] 选项卡 33
图 3-7 [密钥类型] 选项卡 34
图 3-8 选择了 DLM/KUK/AES/TDES/ARC4/ChaCha20-Poly1305/ECC 私钥时的 [密钥数据文件] 选项卡 37
图 3-9 选择了 RSA 公钥时的 [密钥数据文件] 选项卡 38
图 3-10 选择了 RSA 私钥时的 [密钥数据文件] 选项卡 39
图 3-11 选择了 ECC 公钥时的 [密钥数据文件] 选项卡 40
图 3-12 选择了 ECC 公钥时的 [密钥数据文件] 选项卡 41
图 3-13 封装密钥选项 42
图 3-14 IV 选项 43
图 3-15 输出文件选项 44
图 3-16 [TSIP Update]选项卡 46
图 3-17 输出映像 47
图 3-18 固件映像/安全引导映像 47
图 3-19 [RSU标头]选项卡 48
图 3-20 [加密地址范围]选项卡 49
图 3-21 [Image Encryption Key]选项卡 50
图 3-22 [IV]选项卡 51
图 3-23 输出 51
图 3-24 [FSBL]选项卡 52
图 3-25 编程验证方法 54
图 3-26 [证书]选项卡 55
图 3-27 [OEM根密钥]选项卡 56

图 3-28 [OEM引导加载程序密钥]选项卡.....57
图 3-29 [DOTF/OTFD]选项卡.....59
图 3-30 加密范围.....60
图 3-31 目标地址.....61
图 3-32 映像加密密钥.....62
图 3-33 IV.....62
图 3-34 输出映像地址和内容.....63
图 3-35 [SFP]选项卡.....64
图 3-36 [固件映像]选项卡.....67
图 3-37 [映像加密密钥]选项卡.....68
图 3-38 [Nonce]选项卡.....69
图 3-39 [AL2/SECDBG_KEY]选项卡.....70
图 3-40 [AL1/NONSECDBG_KEY]选项卡.....71
图 3-41 [Boundary]选项卡.....72
图 3-42 [外部闪存区]选项卡.....73
图 4-1 手动创建 AES 128 位密钥文件的示例.....89
图 4-2 手动创建 RSA 2048 公钥文件的示例.....90
图 4-3 指定 ver 选项 1 时的加密流程.....101
图 4-4 指定 ver 选项 2 时的加密流程.....103
图 4-5 mode选项 指定factory时的文件生成示意图.....106
图 4-6 mode选项指定update且 filetype选项指定rsu时的文件生成示意图.....107
图 4-7 mode选项指定update且 filetype选项指定mot时的文件生成示意图.....107
图 4-8 指定oembl_size选项时的签名、CRC运算对象.....113
图 4-9 仅指定cfsize选项时的签名、CRC运算对象.....114
图 5-1 安全密钥管理工具 – 从 Windows 启动时的 GUI 对话框.....126
图 5-2 安全密钥管理工具 - 从命令提示符执行 CLI 示例.....127
图 5-3 安全密钥管理工具 – 从 Linux 启动时的 GUI 对话框.....128
图 5-4 安全密钥管理工具 - 从 Linux 终端软件执行 CLI 示例.....129
图 5-5 安全密钥管理工具 – 从 macOS 启动时的 GUI 对话框.....130
图 5-6 安全密钥管理工具 - 从 macOS 终端软件执行 CLI 示例.....131
图 5-7 e²studio“帮助(H)”–“安装新软件...”.....132
图 5-8“安装”对话框.....133
图 5-9“添加资源库”对话框.....133
图 5-10“安装”对话框 – 选择“Security Key Management Tool”.....134
图 5-11“安装”对话框 – 安装详细信息.....134
图 5-12“安装”对话框 – 查看许可协议.....135
图 5-13 “信任”对话框.....135
图 5-14 项目“属性”对话框.....136
图 5-15“关于 e²studio”对话框.....137
图 5-16“e² studio 安装细节”对话框.....137
图 5-17“卸载”对话框.....138
图 6-1 [概要] 选项卡，使用 RX 产品家族 TSIP 安装 AES128 密钥.....139
图 6-2 [生成 UFPK] 选项卡，使用指定的值生成 UFPK 的示例.....140
图 6-3 使用指定的值生成 UFPK 的执行结果示例.....140
图 6-4 [封装密钥] - [密钥类型] 选项卡，以 Motorola 十六进制格式创建 AES128 密钥文件的示例.....141
图 6-5 [封装密钥] - [密钥数据文件] 选项卡，以 Motorola 十六进制格式创建 AES128 密钥文件的示例...141

图 6-6 以 Motorola 十六进制格式创建 AES128 密钥文件的执行结果示例.....142
图 6-7 CLI genufpk 命令的执行结果.....143
图 6-8 以 Motorola 十六进制格式创建 AES128 密钥文件的 CLI 示例.....143
图 6-9 [概要] 选项卡，在 RA 产品家族 SCE9 保护模式下安装 KUK.....144
图 6-10 [生成 UFPK] 选项卡，使用指定的值生成 UFPK 的示例.....145
图 6-11 使用指定的值生成 UFPK 的执行结果示例.....145
图 6-12 [生成 KUK] 选项卡，创建 KUK 密钥文件的示例.....146
图 6-13 创建 KUK 文件的执行结果示例.....146
图 6-14 [封装密钥] - [密钥类型] 选项卡，以 RFP 文件格式创建 KUK 文件的示例.....147
图 6-15 [封装密钥] – [密钥数据文件] 选项卡，以 RFP 文件格式创建 KUK 文件的示例.....147
图 6-16 以 RFP 文件格式创建 KUK 文件的执行结果示例.....148
图 6-17 CLI genufpk 命令的执行结果.....148
图 6-18 CLI genkuk 命令的执行结果.....148
图 6-19 以 RFP 文件格式创建 AES128 密钥文件的 CLI 示例.....149
图 6-20 [概要] 选项卡，在具有 SCE9 保护模式的 RA 产品家族上升级 RSA 2048 公钥.....150
图 6-21 [封装密钥] - [密钥类型] 选项卡，以 C 源文件格式创建 RSA 2048 公钥文件的示例.....151
图 6-22 [封装密钥] – [密钥数据文件] 选项卡，以 C 源文件格式创建 RSA 2048 公钥文件的示例.....151
图 6-23 以 C 源文件格式创建 RSA 2048 公钥文件的执行结果示例.....152
图 6-24 以 C 源文件格式创建 RSA 2048 公钥文件的 CLI 示例.....153
图 6-25 [概要]选项卡.....154
图 6-26 [TSIP Update] – [RSU标头]选项卡设置示例.....155
图 6-27 [TSIP Update] – [加密地址范围]选项卡等的设置示例.....155
图 6-28 [TSIP Update] – [Image Encryption Key]选项卡设置示例.....156
图 6-29 [TSIP Update] – [IV]选项卡设置示例.....156
图 6-30 [TSIP Update]选项卡 执行结果.....156
图 6-31 enctsip命令执行示例.....157
图 6-32 [概要]选项卡.....158
图 6-33 [FSBL] – [证书]选项卡等的设置示例.....159
图 6-34 [FSBL] – [OEM根密钥]选项卡设置示例.....160
图 6-35 [FSBL] – [OEM引导加载程序密钥]选项卡设置示例.....160
图 6-36 [FSBL]选项卡 执行结果.....161
图 6-37 gencert命令执行示例.....162
图 6-38 [概要]选项卡.....163
图 6-39 [DOTF/OTFD]选项卡 – 明文映像、加密范围、执行地址的设置示例.....164
图 6-40 [DOTF/OTFD]选项卡 – 映像加密密钥、IV设置示例.....164
图 6-41 [DOTF/OTFD]选项卡 – 加密映像 输出映像地址和内容的设置示例.....165
图 6-42 [DOTF/OTFD]选项卡 执行结果.....165
图 6-43 encdotf命令执行示例.....166
图 6-44 [概要]选项卡.....168
图 6-45 [SFP] – [固件映像]选项卡等的设置示例.....169
图 6-46 [SFP] – [映像加密密钥]选项卡设置示例.....170
图 6-47 [SFP] – [Nonce]选项卡设置示例.....171
图 6-48 [SFP] – [AL2/SECDBG_KEY]选项卡设置示例.....171
图 6-49 [SFP]选项卡 执行结果.....172
图 6-50 encsfp命令执行示例.....173
图 7-1 SecurityKeyManagementTool.exe 属性“高 DPI 缩放覆盖”设置.....174

图 7-2 [概要]选项卡 MCU/MPU 和加密引擎 选择 RX Family，TSIP.....177
图 7-3 [密钥封装]选项卡 [密钥类型]选项卡 选择 RSA 2048 位，公开 177
图 7-4 [概要]选项卡 MCU/MPU 和加密引擎 选择 RX Family，TSIP Lite 177
图 7-5 [密钥封装]选项卡 [密钥类型]选项卡 AES 128bit [密钥数据]选项卡显示 RSA public输入.....178
图 8-1 V1格式图像图.....184
图 8-2 V2格式图像图.....185
图 8-3 V2 格式 Header Field 187
图 8-4 V2 格式 Parameter Field.....188
图 8-5 V2 格式 DLM_AL Key Field 189
图 8-6 V2 格式 Nonce and MAC for encrypted user data Field 189
图 8-7 TLV格式 190

1. 瑞萨密钥管理系统

1.1 信任根简介

信任根是高度可靠的硬件、固件和软件组件，用于执行特定关键安全功能 (https://csrc.nist.gov/projects/hardware-roots-of-trust)。在物联网系统中，信任根通常由器件硬件中固有的标识和加密密钥组成。它建立唯一不变的防克隆标识，以授权器件存在于物联网网络中。

在许多安全系统中，安全引导是信任根所提供服务的一部分。应用程序的身份验证使用公钥加密。关联密钥是系统信任根的一部分。器件标识由器件私钥和器件证书组成，是许多物联网设备信任根的一部分。

基于上述信任根的探讨，我们可以了解到加密密钥泄露会导致安全系统处于危险状态。信任根的保护包括将密钥可访问性仅限制在加密边界内，同时使用安全存储密钥且最好是防克隆密钥。信任根应锁定，以防止未授权方对其进行读写访问。

瑞萨用户密钥管理系统可以提供上述全部所需的保护。

1.2 瑞萨安全引擎和关联密钥简介

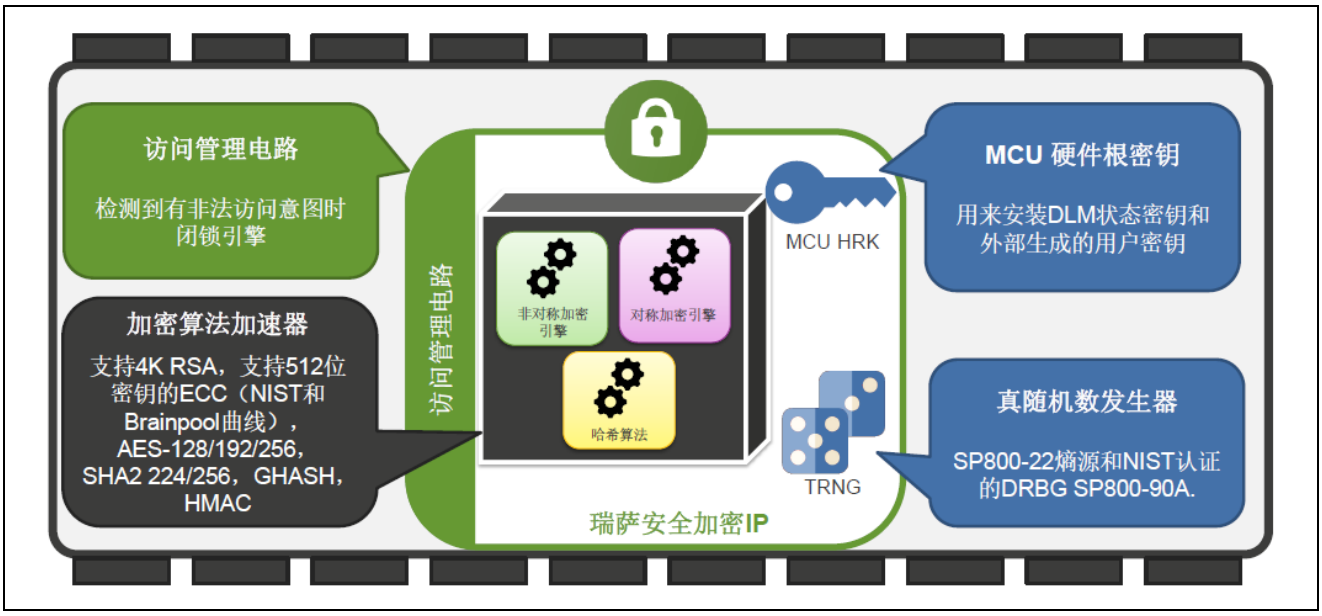

图 1-1 安全加密引擎概览

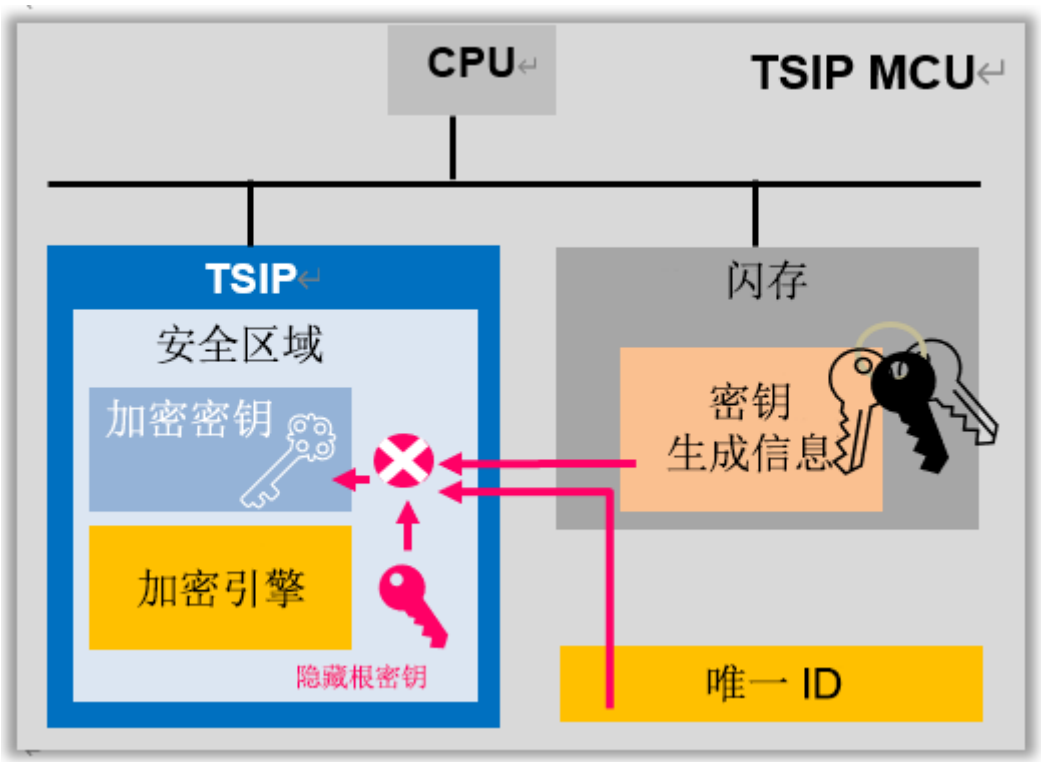

图 1-2 RX 可信安全引擎概览

瑞萨安全引擎包括安全加密引擎 (SCE) 和可信安全引擎 (TSIP) 和 RSIP-E 型安全引擎 (RSIP) 是 MCU 内部的孤立子系统。加密引擎包括用于对称和非对称加密算法的硬件加速器，以及各种哈希和消息验证代码。其中还包括真随机数发生器 (TRNG)，用于为加密操作提供熵源。加密引擎受访问管理电路保护，该电路可以在有非法外部访问意图时闭锁加密引擎。

根据特定 MCU/MPU，支持特定算法、安全密钥注入和安全密钥升级。有关更多信息，请参见设备的硬件用户手册及以下网页：

表 1-1 MCU/MPU 相关信息

MCU/MPU	类别	URL
RA 产品家族	MCU 驱动程序	https://www.renesas.com/eu/en/software-tool/flexible-software-package-fsp
	应用程序项目	https://www.renesas.com/eu/en/document/apn/installing-and-updating-secure-keys-ra-family
RX 产品家族	MCU 驱动程序和示例项目	https://www.renesas.com/software-tool/trusted-secure-ip-driver
RZ/T2M, RZ/T2ME, RZ/T2L, RZ/N2L	安全软件包	https://www.renesas.com/software-tool/rz-mpu-security-package-rtos-bare-metal
Synergy 平台	MCU 驱动程序	https://www.renesas.com/eu/en/products/microcontrollers-microprocessors/renesas-synergy-platform-mcus/renesas-synergy-software-package

1.3 瑞萨安全密钥安装的优势

安全密钥安装和升级，结合加密引擎对封装密钥的支持，解决了与使用明文密钥相关的许多漏洞：

- 明文密钥永远不会存储在代码闪存中。在程序存储器被攻破的情况下，可以保护敏感的密钥数据。
- 明文密钥永远不会存储在 RAM 中。在系统上可执行恶意代码的情况下，仍会保护敏感的密钥数据。
- 密钥可以安全地存储在代码闪存、数据闪存中，甚至可以复制到外部存储器中，从而实现无数量限制的安全密钥存储。

此外，瑞萨密钥封装技术可防止器件被克隆，如下文所述。

1.3.1 密钥封装与密钥加密相比的优势

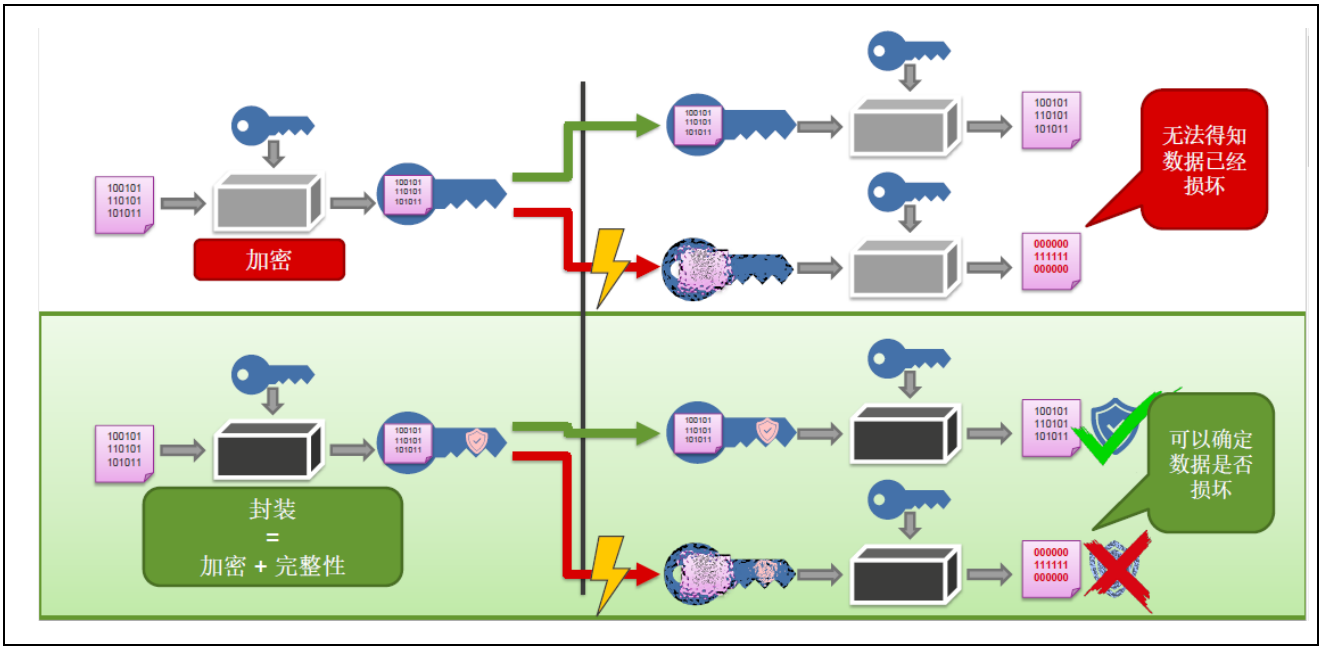

图 1-3 密钥封装与密钥加密

务必要了解安全资产存储时使用封装和加密之间的区别。当数据经过加密并发送到另一接收方时，如果该接收方具有相同的密钥，则可以对该数据进行解密。结果即是秘密地交换了信息。但是，如果加密数据的传输出现问题会如何？如果接收方在不知情的情况下接收到已损坏的信息，则解密算法将生成垃圾数据，并且无法得知原始数据已损坏。

利用封装技术，可在加密输出上附加用于完整性检查的消息验证数据，从而解决了此问题。

1.3.2 使用 HUK 的密钥封装的优势

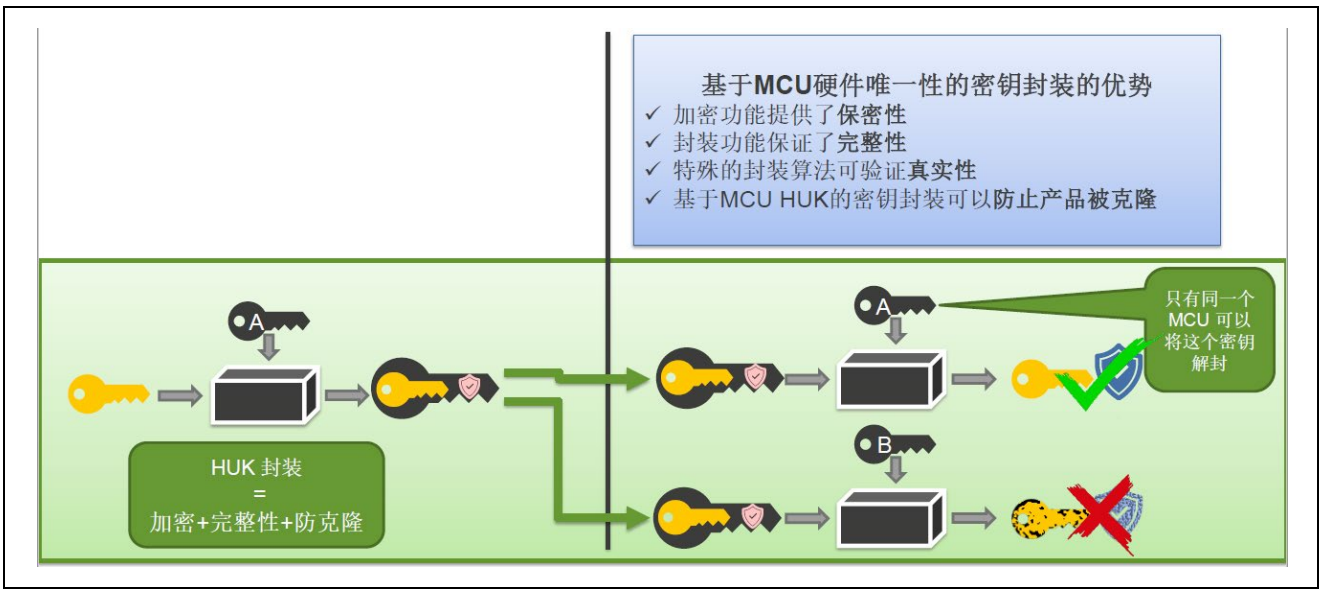

图 1-4 使用 HUK 的密钥封装

使用硬件唯一密钥 (HUK) 封装存储的密钥可提供另一项保护功能 – 防克隆保护。如果封装的密钥传输或复制到其他 MCU/MPU，则该 MCU/MPU 将无法解封也无法使用复制的密钥。即使将整个 MCU 的内容复制到其他器件上，也无法使用或显示密钥。

1.4 封装密钥安装程序概述

本节介绍了封装密钥的注入程序，该程序需要安全密钥管理工具提供的功能。不同 MCU/MPU 支持的密钥类型不同。有关每个器件的应用笔记和驱动程序，参见*表 1-1 MCU/MPU 相关信息*。

1.4.1 安全密钥注入的一般步骤

安全密钥注入是以 MCU 唯一封装的方式存储应用程序密钥的过程，该过程采用在配置过程中启用不显示明文密钥的机制实现。该过程的具体机制取决于具体的 MCU/MPU。一些器件支持通过编程接口实现安全密钥注入；其他器件支持使用器件上运行的固件进行安全密钥安装。请注意，使用安全密钥管理工具的密钥准备步骤（其中密钥材料以明文显示）必须在安全环境中执行。

密钥注入分三个大的步骤。

1. 安全密钥注入过程的第一步是使用瑞萨的产品生命周期管理 (DLM) 密钥封装服务来封装任意指定的用户工厂烧录密钥 (UFPK)，此步骤使用瑞萨硬件根密钥 (HRK)，生成封装的 UFPK (W-UFPK)。UFPK 是用户选择的 256 位值。可以使用相同的 UFPK 注入任意数量的密钥。安全密钥管理工具可以创建随机的 UFPK，并将其作为适合发送到密钥封装服务的密钥文件。该工具还可以接受指定的 UFPK，并创建适合发送到密钥封装服务的密钥文件。

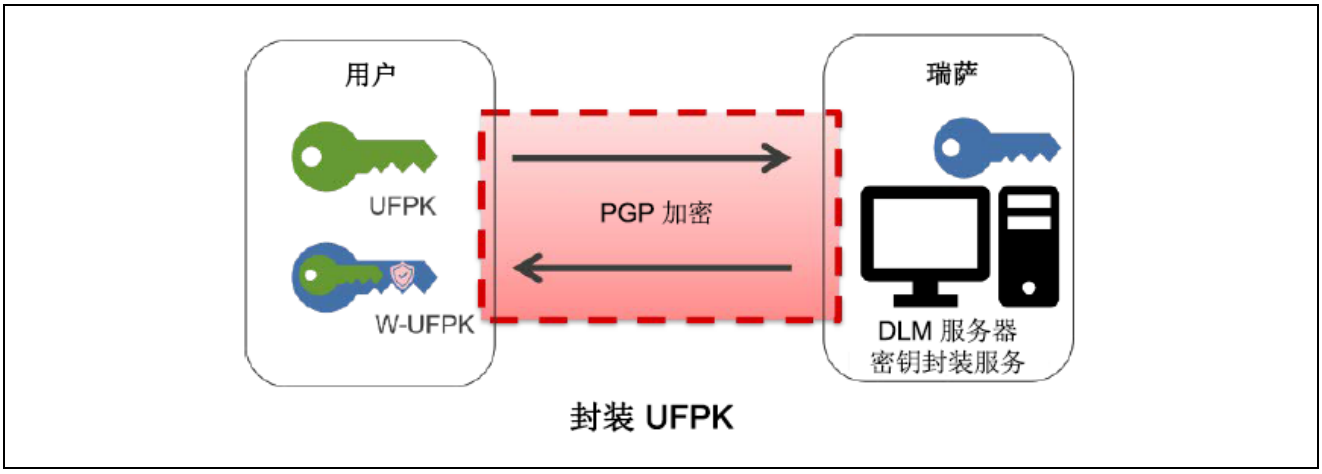

图 1-5 使用 DLM 服务器封装 UFPK

2. 接下来，必须使用 UFPK 来封装用户密钥。安全密钥管理工具可以封装明文密钥数据形式或二进制密钥文件形式的用户密钥，并生成可用于所选器件进行安全密钥安装的文件。请注意，并非所有器件产品家族都支持所有输出文件类型。例如，瑞萨 RA 产品家族仅支持通过编程接口实现安全密钥注入；因此，必须生成用于瑞萨闪存编程器（RFP）的安全密钥文件。

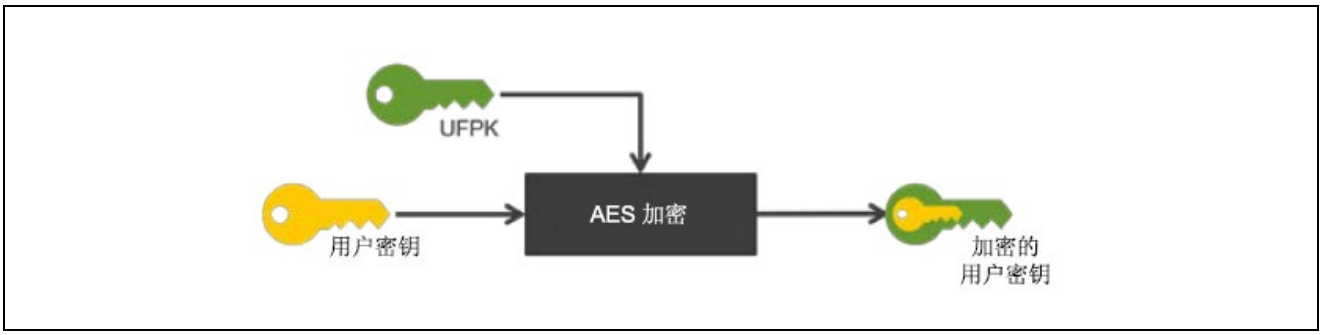

图 1-6 使用 UFPK 加密用户密钥

3. 最后，通过串行编程接口或器件上运行的固件注入用户密钥，具体取决于所选器件支持的密钥注入过程。该过程的输入包括前面步骤中准备的封装 UFPK (W-UFPK) 和封装用户密钥。

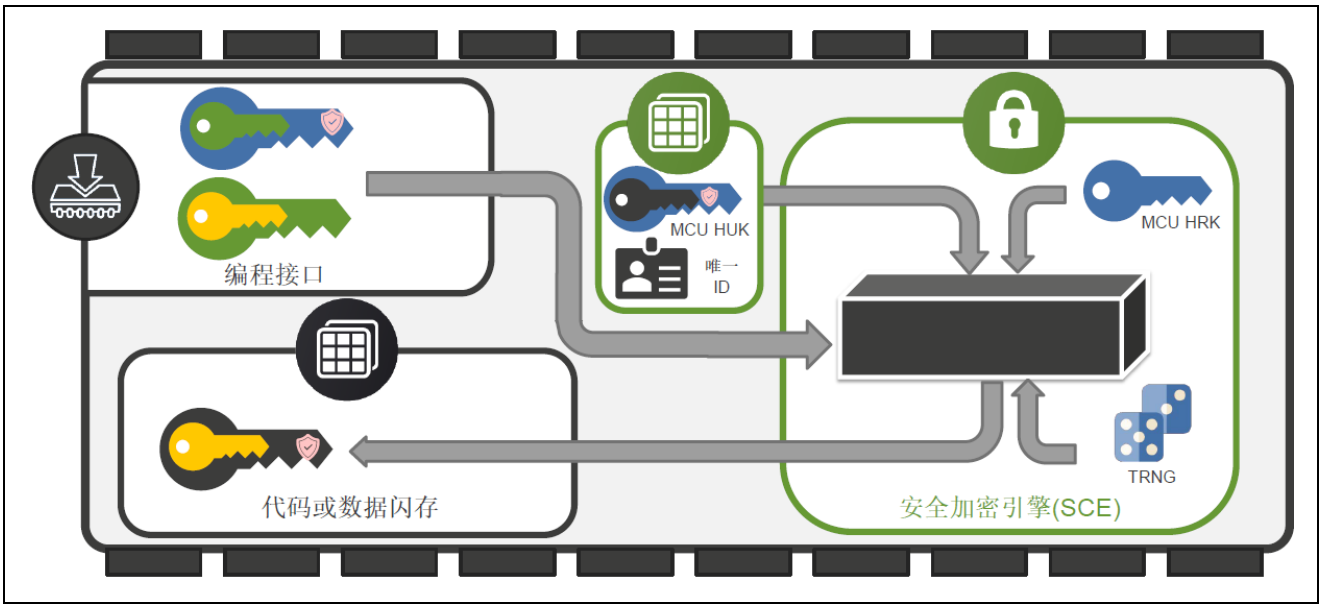

图 1-7 通过串行编程接口注入用户密钥

1.4.2 安全密钥升级的一般步骤

要在系统运行时使用安全密钥安装功能，必须在生产编程/配置过程中注入一个或多个升级密钥 (KUK)。KUK 与其他加密密钥一样，可以存储在代码闪存或数据闪存（如果在 MCU 上可用）中。由于在现场安装新密钥通常是用于替换旧密钥（密钥轮换或重新设置密钥），因此该过程称为“密钥升级”；但是，该过程也可以用于在现场安装新密钥。该过程不会删除之前注入的密钥。

请注意，如果目标器件支持通过编程接口实现密钥注入，则在禁用编程接口之后，无法安装额外的 KUK。在这种情况下，一旦产品位于禁用了编程接口的系统中，则只能通过现存的 KUK 注入新密钥。因此，强烈建议在生产配置过程中注入多个 KUK。这样便允许轮换或撤销 KUK，以遵循架构安全策略或响应密钥泄漏安全漏洞。

密钥升级分三个大的步骤。

1. 第一步是生成并安装 KUK，如*第 1.4.1 节安全密钥*所述。必须注入“KUK”密钥类型的 KUK。

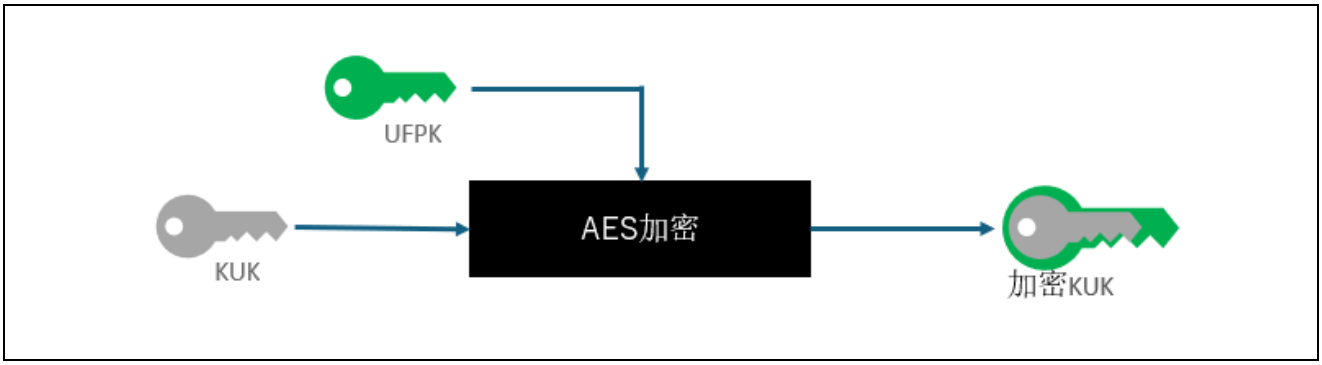

图 1-8 使用 KUK 加密用户密钥

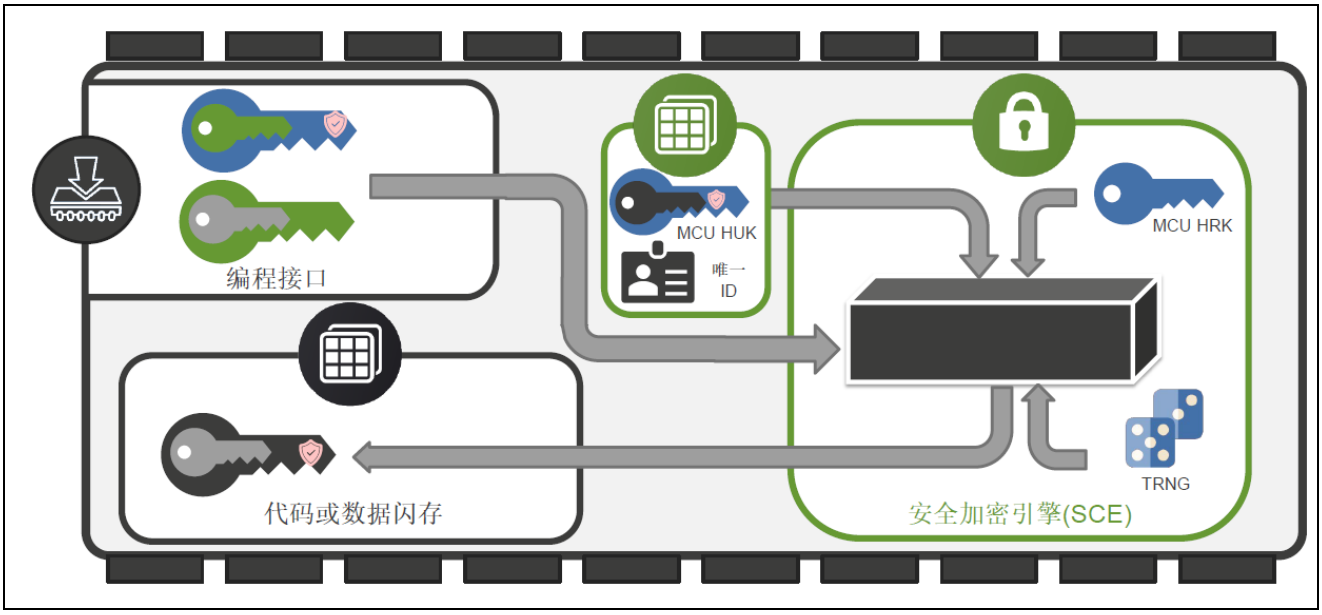

图 1-9 通过串行编程接口注入 KUK

2. 第二步是使用 KUK 封装新用户密钥。这与安全密钥安装类似，但使用的是 KUK 而不是 UFPK。

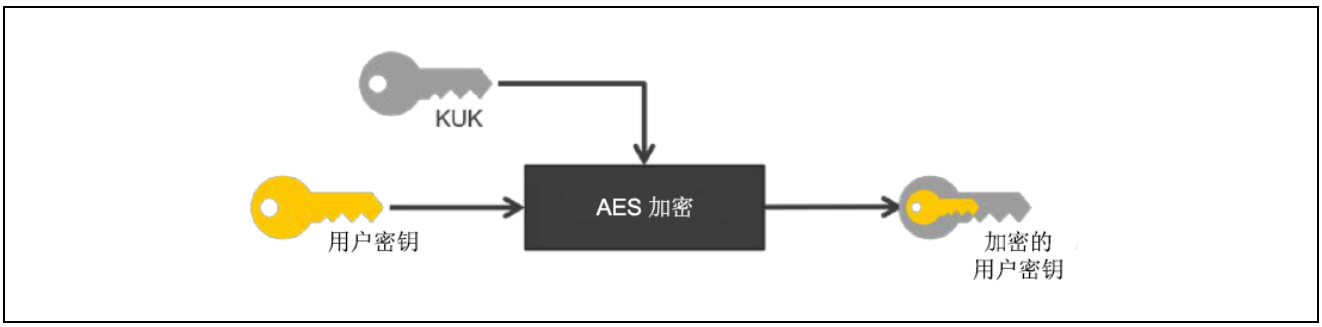

图 1-10 使用 KUK 加密新用户密钥

3. 最后一步是使用相应的器件驱动程序和已注入的 KUK 注入新用户密钥。有关密钥升级 API 及其使用方式的信息，请参见每个 MCU/MPU 器件驱动程序的手册。

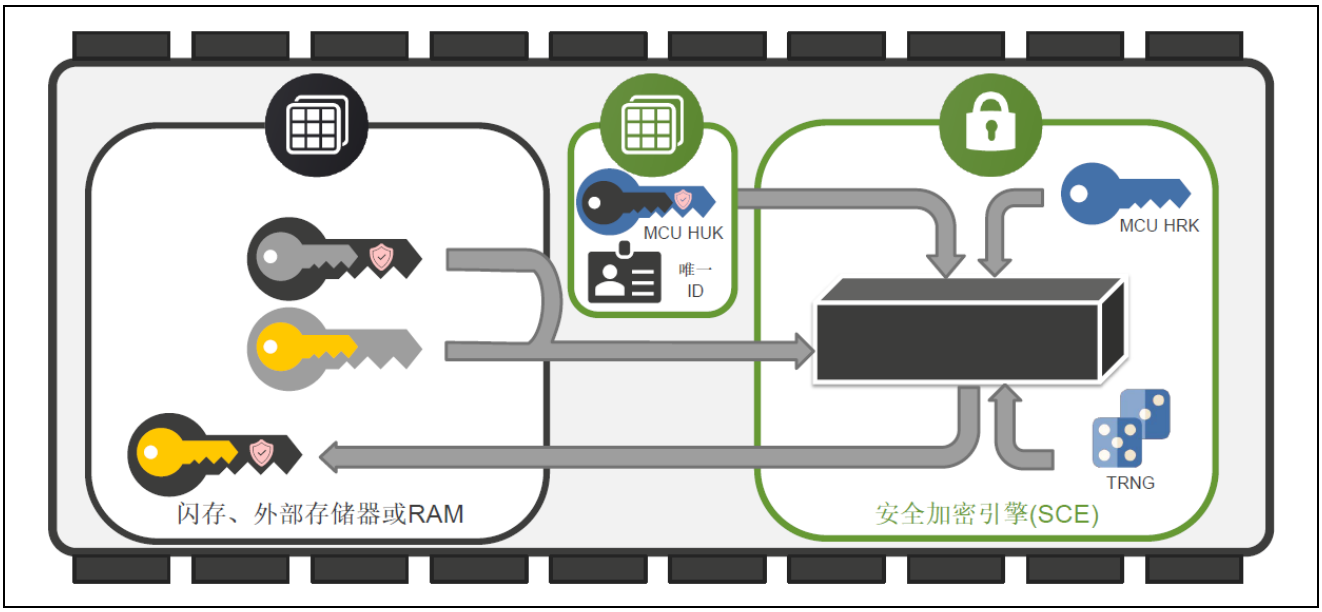

图 1-11 升级用户密钥

1.5 瑞萨安全功能

除了注入及更新以外，瑞萨产品还应用瑞萨安全 IP 提供各种安全功能。安全密钥管理工具支持使用其安全功能所需的功能。支持的功能因 MCU/MPU 而异。请参阅表 1-1MCU/MPU 関連サイト确认各设备的应用指南和驱动程序。

1.5.1 第一阶段引导加载程序

在安全系统中，需要在引导加载程序验证应用程序的合法性后，才能执行该应用程序，以确认待执行的应用程序未发生非法改写。此外，需要确保此引导加载程序本身的合法性，引导加载程序需要配置无法重写的存储器，或由配置在无法重写存储器中的其他引导加载程序来验证其合法性。

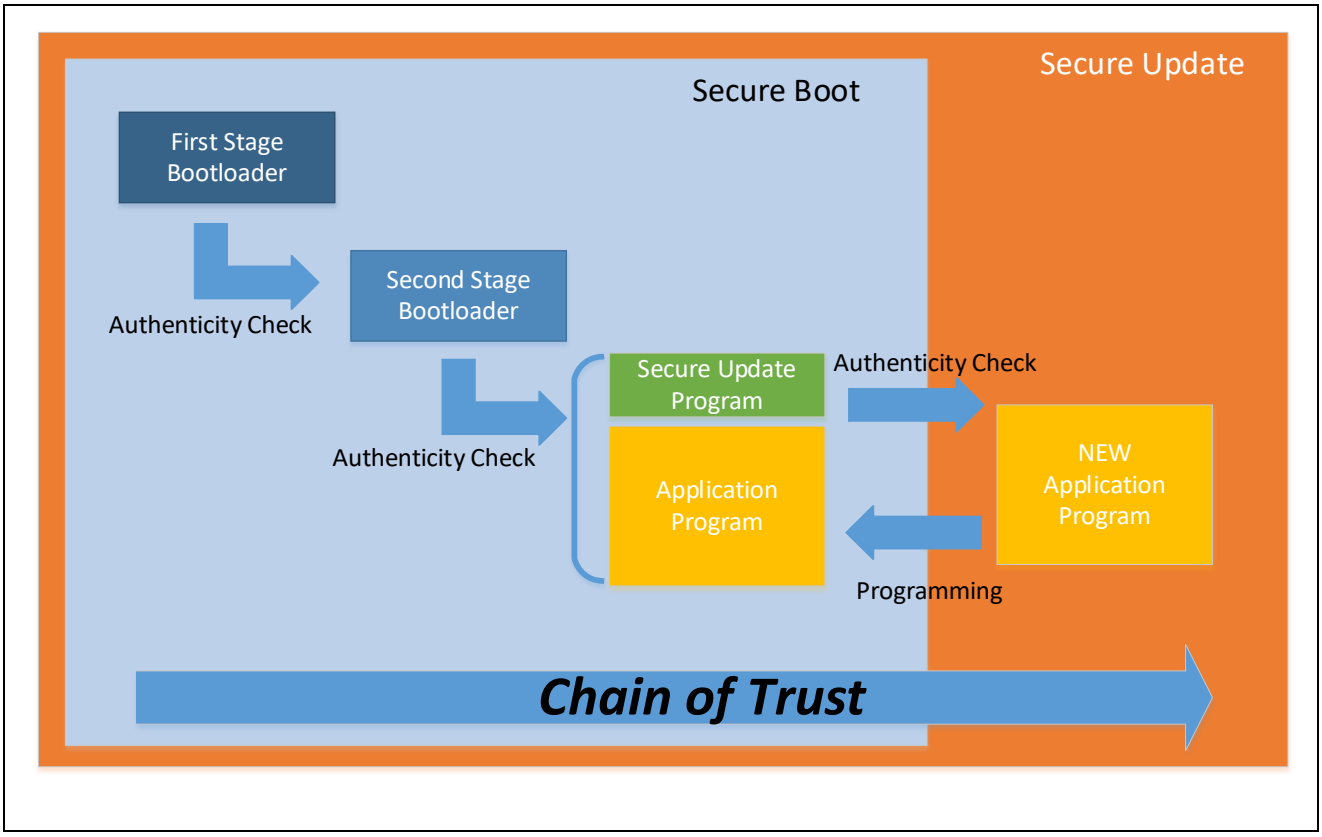

图 1-12 安全系统的Chain of Trust

瑞萨电子的部分 MCU 产品在器件内的 Mask ROM 或 OTP (One Time Programmable) ROM 区域中安装了第一阶段引导加载程序 (FSBL)。OEM 根公钥在器件生产时的编程过程中经已注册。FSBL 使用 OEM 根公钥，进行使用第二阶段引导加载程序公钥的验证。

下图所示的是初始生产编程和常规应用程序执行的安全引导过程，使用了 OEM 根公钥和私钥(OEM_ROOT_PK 及 OEM_ROOT_SK)、OEM 引导加载程序公钥和私钥(OEM_BL_PK 及 OEM_BL_SK)以及 OEM(第二阶段)引导加载程序(OEM_BL)。

安全密钥管理工具将生成验证时要使用的密钥证书及代码证书。详情请参阅 MCU/MPU 的应用指南及驱动程序。

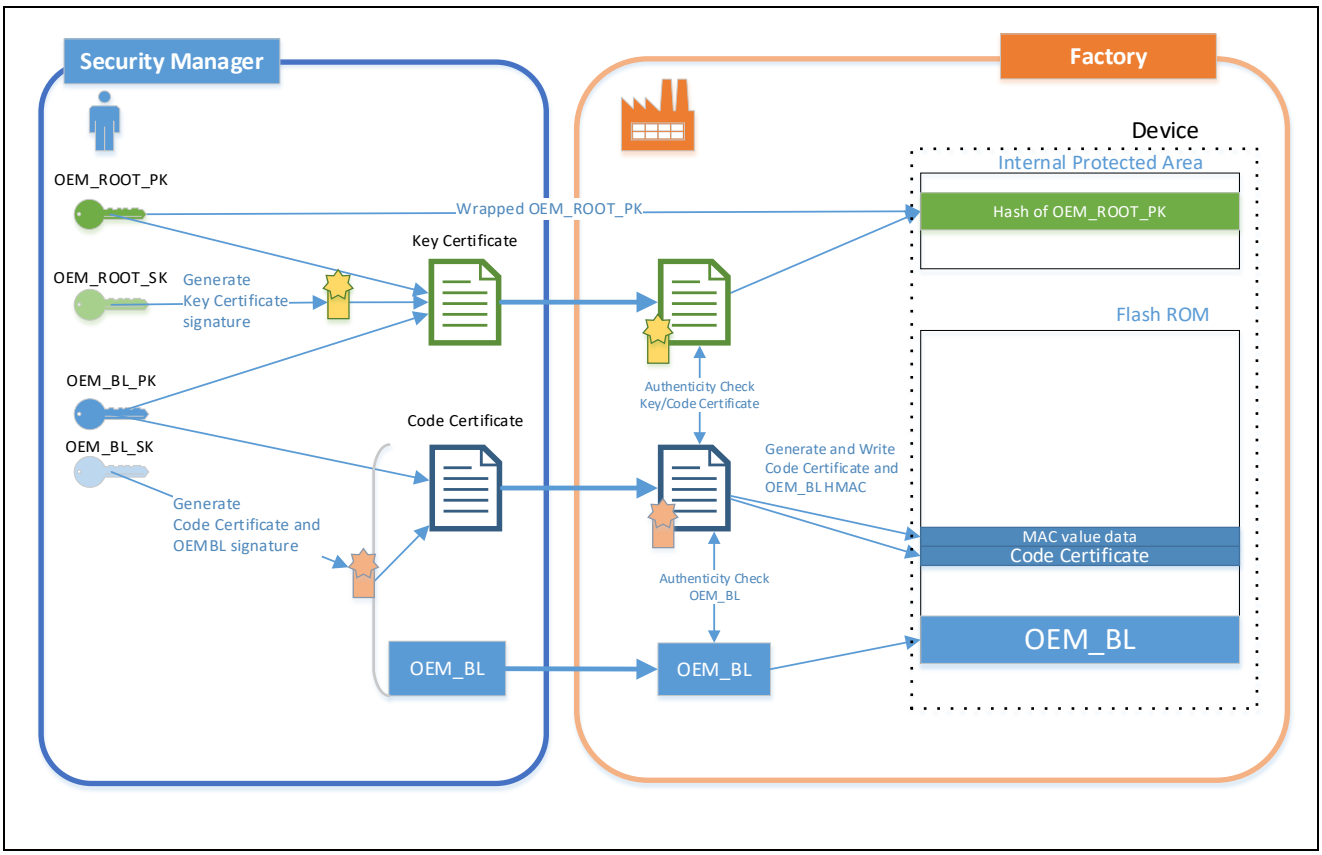

图 1-13 OEM引导加载程序工厂烧录时的Authenticity Check

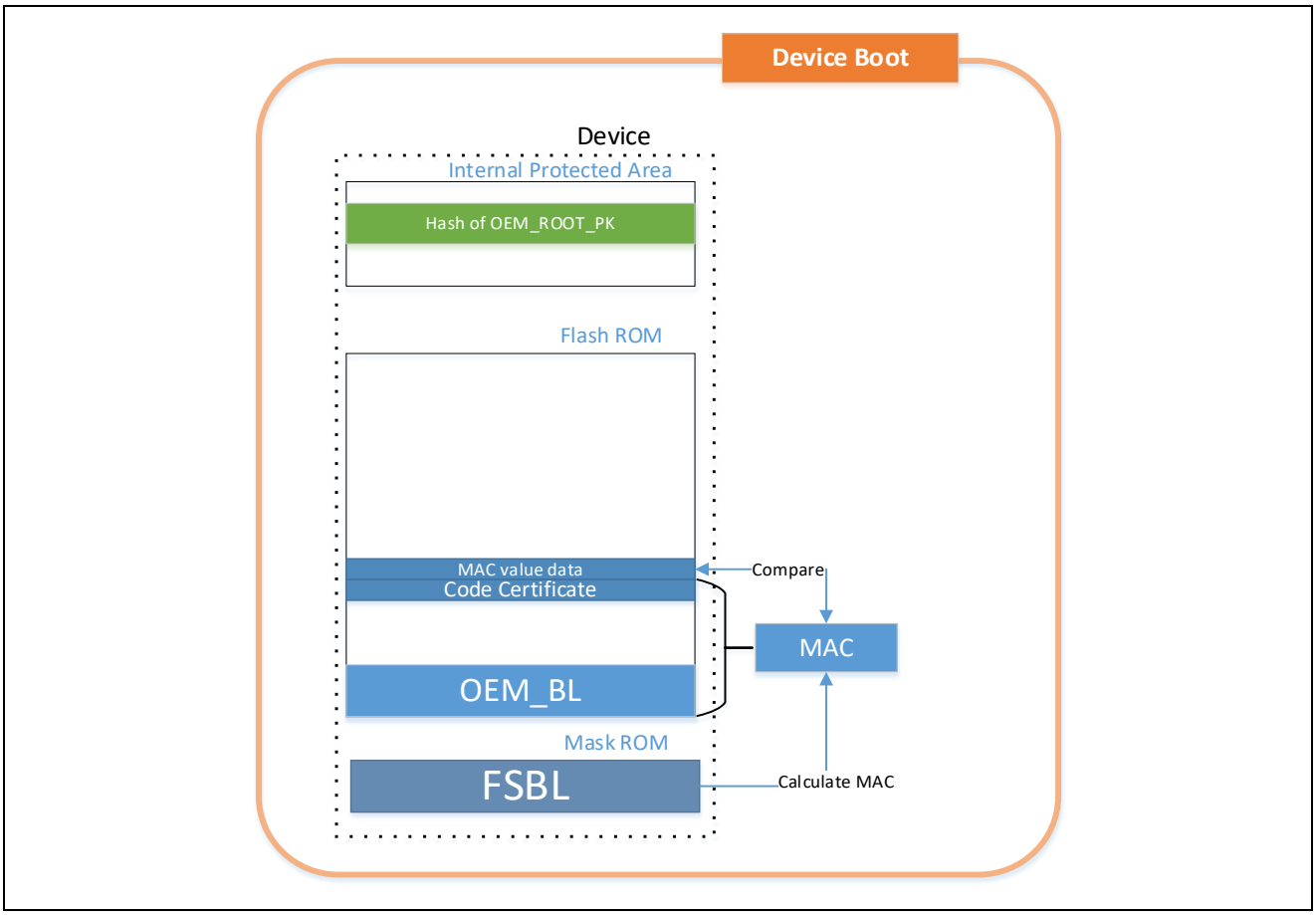

图 1-14 设备启动时的FSBL动作

1.5.2 实时解密

要隐藏存储在外部存储器中的应用程序或机密数据时，需要将加密的应用程序存储到外部存储器中。在 eXecute in Place 中执行在外部 ROM 中加密的应用程序时，实时解密功能将实时解密这些程序。

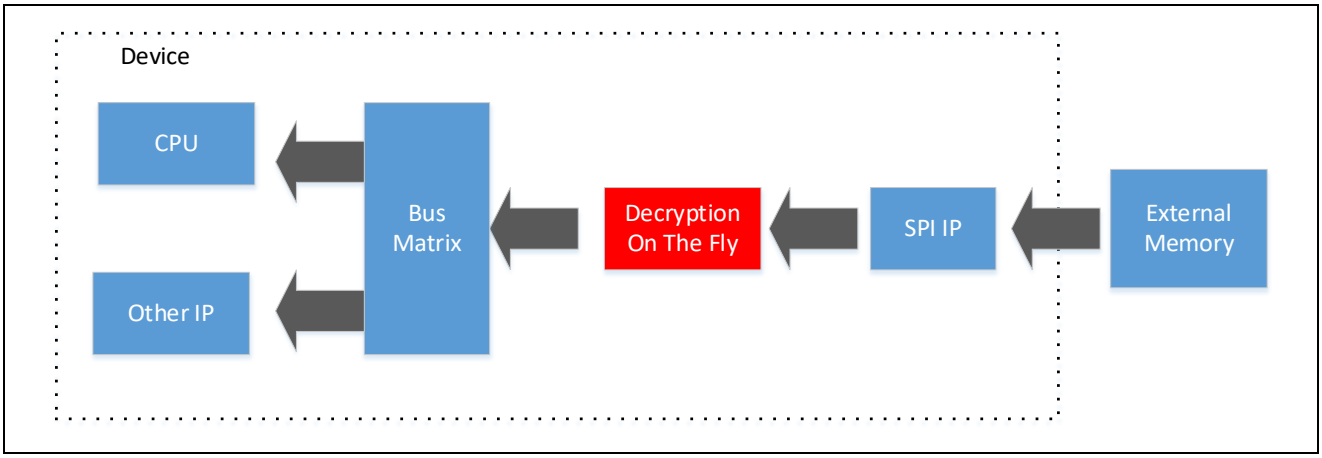

图 1-15 DOTF的MCU/MPU的内部系统总线示例

安全密钥管理工具支持生成要与瑞萨实时解密功能配合使用的应用程序信息（可执行代码或机密数据）。

1.5.3 安全工厂编程

安全工厂编程功能是烧录加密应用程序的功能。即使应用程序文件在工厂烧录时从工厂泄露，也可以防止文件中的程序或数据泄露。

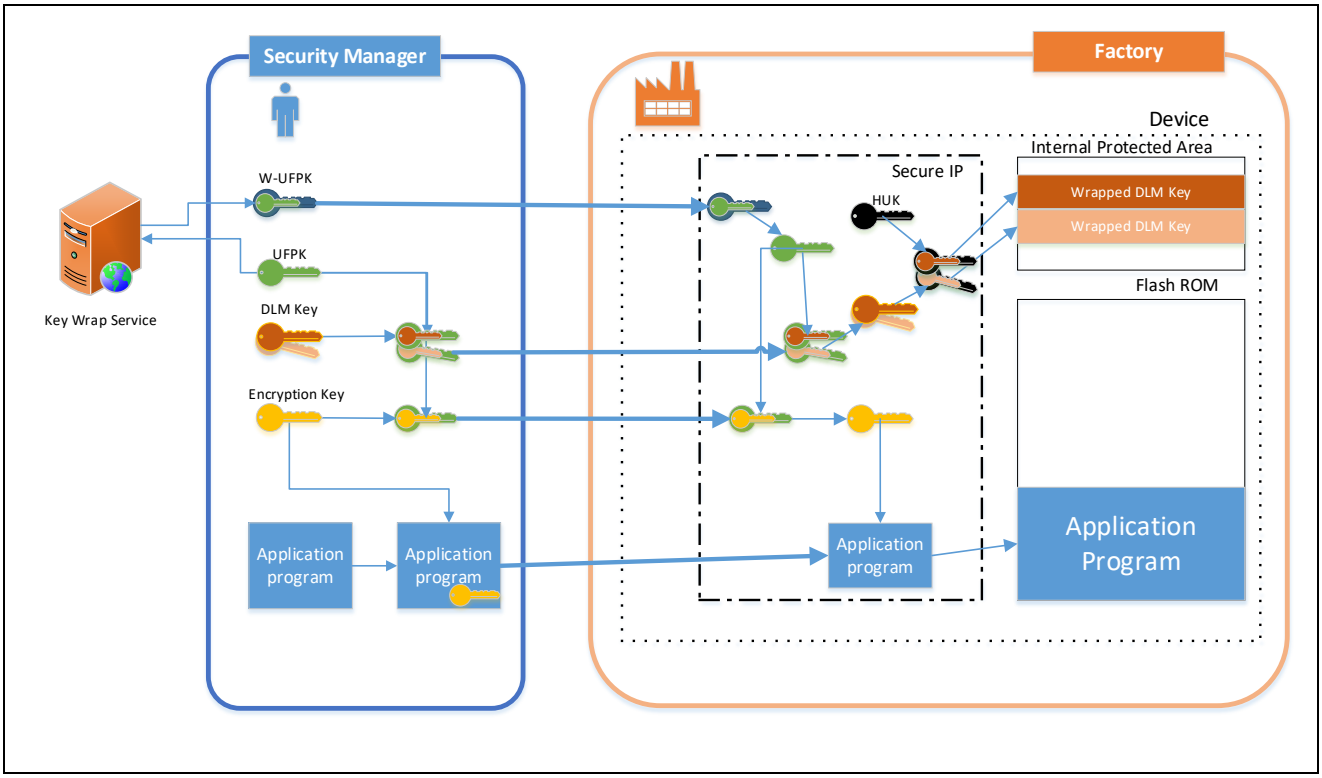

图 1-16 具有安全工厂编程功能MCU/MPU的系统示例

安全密钥管理工具支持以下功能，以实现安全工厂编程功能。

- 用户应用程序的加密
- 使用密钥封装加密
- 同时进行注入器件生命周期的密钥封装
- FSBL 设置。
- 过渡到 DLM 最终状态

请参阅 MCU/MPU 手册，了解特定器件的支持功能列表以及安全工厂编程的限制，如仅支持内部存储器编程。

2. 概述

安全密钥管理工具用于为安全注入和升级准备应用程序和产品生命周期管理 (DLM) 密钥。

该工具有三个版本：

- 命令行接口 – 从操作系统命令行执行单个命令，或者在批处理文件或脚本中包含多个命令。设计用于密钥管理工具和支持小组开发。
- 独立的图形用户界面 – 直观的 GUI，旨在为固件开发人员提供帮助。在使用第三方 IDE 开发时十分有用。
- e² studio 插件 – 集成到瑞萨 e² studio 中的 GUI 接口，用于简化开发支持。

2.1 特性

安全密钥管理工具支持的功能和 MCU/MPU 如下所示。:

2.1.1 注入/更新安全密钥

支持与注入、更新安全密钥相关的以下功能。

关于支持的 MCU/MPU，请参阅表 2-1。

[1] 生成 UFPK 和生成可发送至瑞萨密钥封装服务的格式的文件

[2] 使用注入安全密钥所需的 UFPK 封装 DLM 密钥（仅部分 MCU/MPU 支持 DLM 密钥）

[3] 使用注入安全密钥所需的 UFPK 封装用户密钥

[4] 使用更新安全密钥所需的 KUK 封装用户密钥

表 2-1 MCU/MPU 族 密钥的注入/更新的支持功能

家族	支持功能			
	[1]	[2]	[3]	[4]
RA 产品家族				
RSIP-E51A 安全功能和保护模式	✓	✓	✓	✓
RSIP-E51A 兼容模式	✓	-	✓	-
RSIP-E50D 安全功能和保护模式	✓	✓	✓	✓
RSIP-E50D 兼容模式	✓	-	✓	-
RSIP-E11A 安全功能和保护模式	✓	✓	✓	✓
RSIP-E11A 兼容模式	✓	-	✓	-
SCE9 安全功能和保护模式	✓	✓	✓	✓
SCE9 兼容模式	✓	-	✓	-
SCE7	✓	-	✓	-
SCE5_B	✓	✓	✓	✓
SCE5	✓	-	✓	-
RX 产品家族				
TSIP	✓	-	✓	✓
TSIP-Lite	✓	-	✓	✓
RSIP-E11A 安全功能和保护模式	✓	-	✓	✓
RSIP-E11A 兼容模式	✓	-	✓	-
RZ 产品家族				
RZ/T2M	✓	-	✓	✓
RZ/T2ME	✓	-	✓	✓
RZ/T2L	✓	-	✓	✓
RZ/N2L	✓	-	✓	✓
TSIP	✓	-	✓	✓
瑞萨电子 Synergy 平台				
SCE7	✓	-	✓	✓
SCE5	✓	-	✓	✓

支持下列文件格式的输出(并非所有 MCU/MPU 都支持瑞萨密钥文件)：

- 瑞萨密钥文件
- C 源文件
- 二进制数据
- Motorola 十六进制文件

2.1.2 安全密钥管理工具支持的安全功能

除了安全密钥注入和更新之外，在安全密钥管理工具中还可以生成在以下设备功能中使用的数据。
关于支持的 MCU/MPU，请参阅表 2-2。

[1] 生成 FSBL 所需的密钥证书及代码证书
[2] 在实时解密中对可用的用户程序/数据进行加密
[3] 生成可用于安全工厂编程功能的文件
[4] 使用 TSIP 生成可用于安全更新的文件

表 2-2 MCU/MPU 族 支持的安全功能

家族	支持功能			
	[1]	[2]	[3]	[4]
RA 产品家族				
RSIP-E51A 安全功能和保护模式	✓	✓	✓	-
RSIP-E51A 兼容模式	-	✓	-	-
RSIP-E50D 安全功能和保护模式	✓	✓	✓	-
RSIP-E50D 兼容模式	-	✓	-	-
RSIP-E11A 安全功能和保护模式	-	-	✓	-
RSIP-E11A 兼容模式	-	-	-	-
SCE9 安全功能和保护模式	-	-	-	-
SCE9 兼容模式	-	-	-	-
SCE7	-	-	-	-
SCE5_B	-	-	-	-
SCE5	-	-	-	-
RX 产品家族				
TSIP	-	-	-	✓
TSIP-Lite	-	-	-	✓
RSIP-E11A 安全功能和保护模式	-	-	-	✓
RSIP-E11A 兼容模式	-	-	-	-
RZ 产品家族				
RZ/T2M	-	-	-	-
RZ/T2ME	-	✓	-	-
RZ/T2L	-	-	-	-
RZ/N2L	-	-	-	-
TSIP	-	-	-	-
瑞萨电子 Synergy 平台				
SCE7	-	-	-	-
SCE5	-	-	-	-

2.2 运行环境

2.2.1 硬件环境

(1) 主机 PC

- 处理器：1 GHz 或更快
- 主存储器：1 GB 或更大
- 显示设置：1366 x 768 或更高分辨率
  显示比例 100%（推荐）

注：
如果未按上述分辨率或显示比例进行显示，则可能无法显示 GUI 中的所有选项。

2.2.2 软件环境

(1) 支持 OS

- Windows 10, Windws11（64 位）
- Linux (Ubuntu 22.04 LTS, Ubuntu 24.04 LTS)
- macOS 14 Sonoma (仅支持 Apple 硅芯片。)

(2) e²studio 插件版本 e²studio 运行检查版本

- e²studio 2025-04

3. GUI 功能说明

本章介绍安全密钥管理工具 GUI 版本的界面结构和功能。
独立版本和 e²studio 插件版本具有相同的 GUI 配置。在此只对独立版本进行了说明。

3.1 主窗口

启动后的主窗口包括以下内容：

图 3-1 独立版本的主窗口

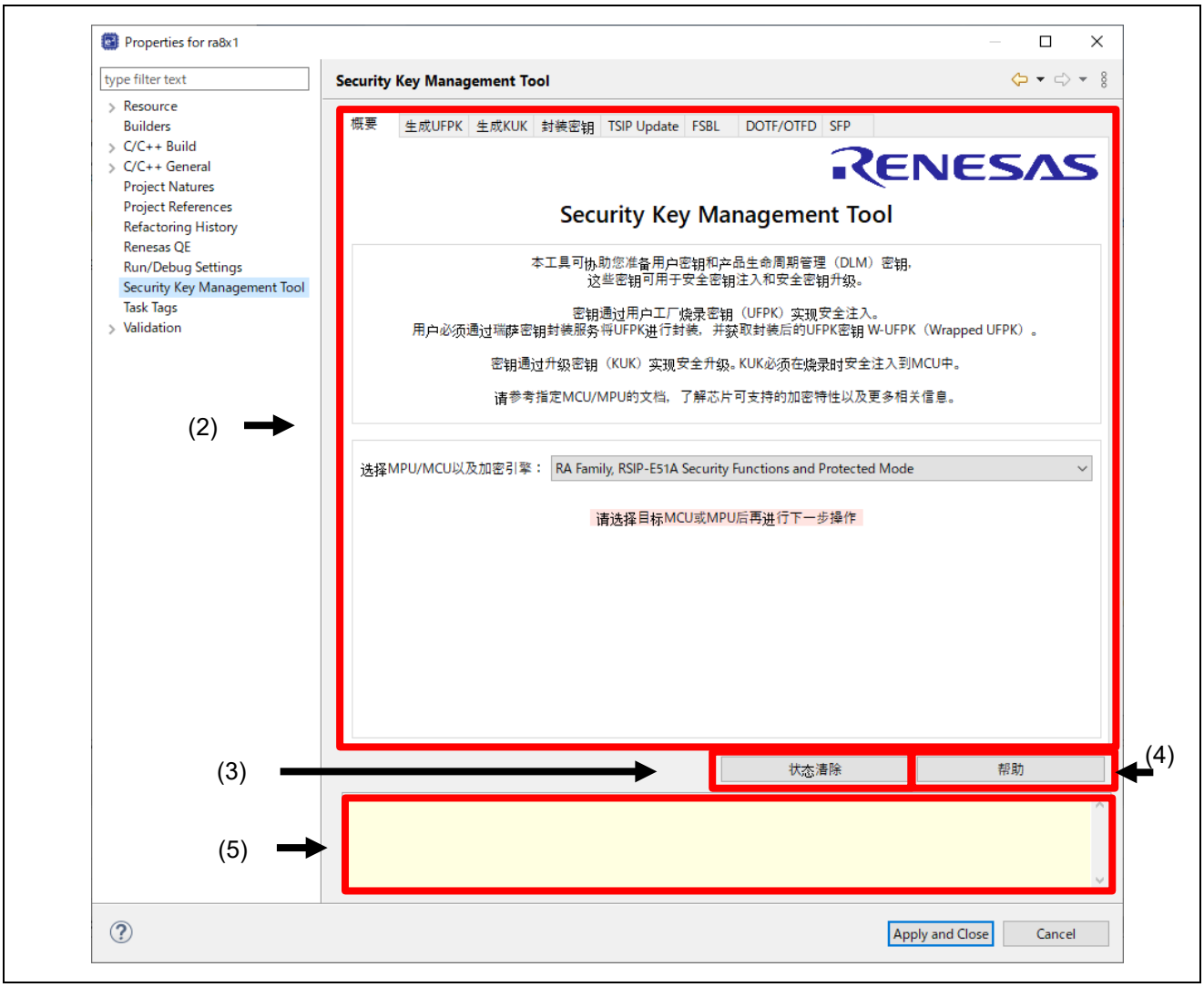

图 3-2 e²studio 插件版本的主窗口

编号	项目	说明
(1)	菜单栏	仅支持独立版本。 有关详细信息，请参见 3.2 菜单栏。
(2)	选项卡窗口	每个选项卡都提供一个界面，用于生成作为安全密钥安装和/或升级过程的一部分的文件。
(3)	**状态清除**	清除状态区域显示的执行结果。在独立版本中，通过菜单栏使用这个功能
(4)	**帮助**	显示对理解瑞萨密钥管理系统有用的资源。在独立版本中，通过菜单栏使用这个功能
(5)	状态	显示请求操作的状态。

3.2 菜单栏

这只是独立版本的一个功能。

3.2.1 [文件] 菜单

从与保存设置相关的功能菜单中选择。

- **[保存…]**

将每个选项卡中设置的输入/输出文件信息保存为 XML 格式的文件。密钥数据不会保存。

- **[加载…]**

加载 **[保存...]** 中保存的设置文件，并反映在每个选项卡中。

在 e²studio 插件版本中，按“**应用并关闭**”按钮将每个选项卡的配置信息保存到 e²studio 项目。

3.2.2 [视图] 菜单

与 GUI 显示相关的菜单。

- **[状态清除]**

清除状态区域显示的执行结果。

3.2.3 [帮助] 菜单

- [关于 Security Key Management Tool…]

帮助对话框，显示对理解瑞萨密钥管理系统有用的资源。

3.3 [概要] 选项卡

在该选项卡中，选择要使用的目标 MCU/MPU 以及加密引擎。确保先选择目标器件，然后再对任何其他选项卡执行操作。其他选项卡的功能取决于在该选项卡上选择的器件。

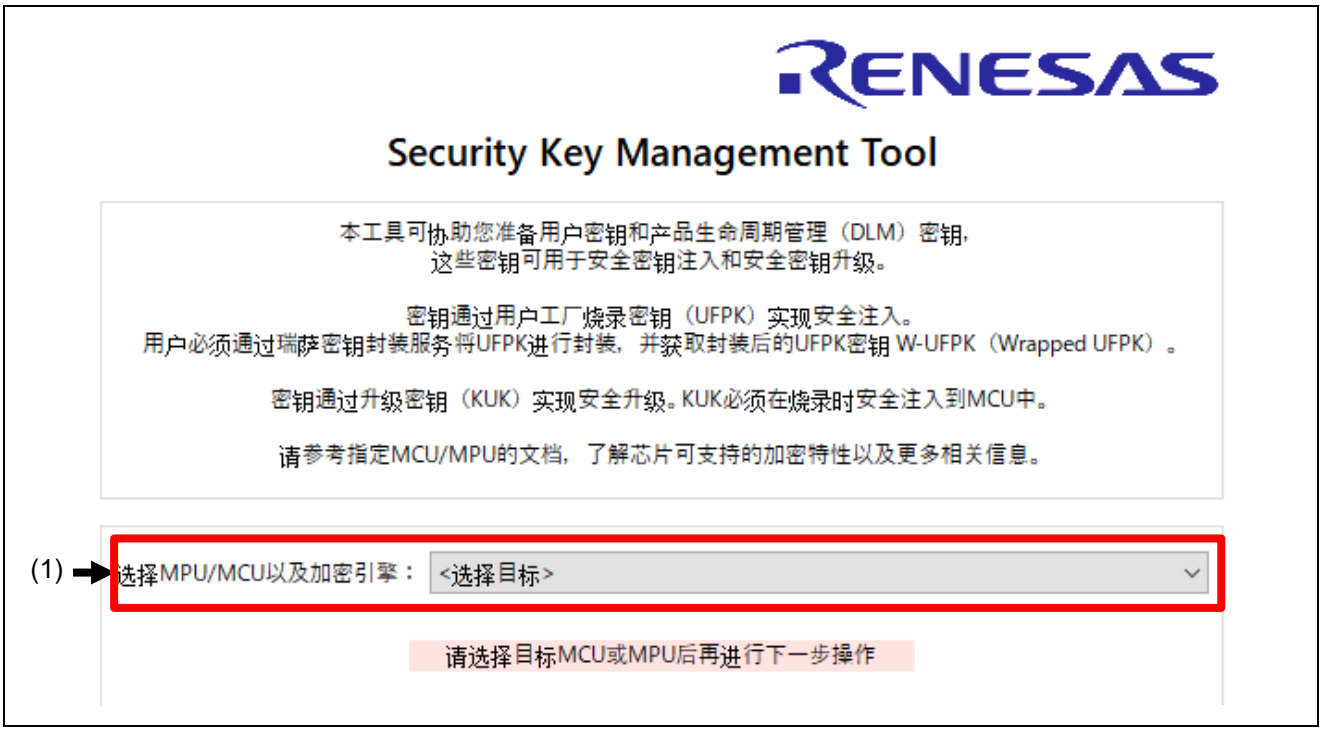

图 3-3 [概要] 选项卡

编号	项目	说明
(1)	选择 MPU/MCU 以及加密引擎	选择目标 MCU/MPU 以及加密引擎进行安全密钥安装和/或升级。

3.4 [生成 UFPK] 选项卡

该选项卡以二进制 `*.key` 文件形式生成用户工厂烧录密钥 (UFPK) 文件。然后必须将该文件发送到瑞萨密钥封装服务(DLM服务器)来获取封装的UFPK (W-UFPK)。UFPK文件还将用于为安全密钥安装准备密钥。

图 3-4 [生成 UFPK] 选项卡

编号	项目	说明
(1)	UFPK 输入格式	选择是使用 (2) 中的输入值作为 UFPK 值还是使用工具生成的随机数值。无法保证该工具随机生成的随机数的具体值。
(2)	UFPK 值	当在 (1) 中选择了**使用指定的值**时，将使用文本框中输入的值作为 UFPK 值。该值必须以大端格式指定为 32 字节十六进制值。
(3)	输出文件 (.key)	选择要输出的 UFPK 文件的路径和文件名。输出文件的扩展名必须设置为 `.key`。
(4)	**生成 UFPK 密钥文件**	根据上面输入的信息生成 UFPK 文件。 在 [概要] 选项卡上选择 MCU/MPU 以及加密引擎时启用。

必须将该选项卡中生成的 UFPK 文件发送到瑞萨密钥封装服务 (https://dlm.renesas.com/keywrap) 来获取 W-UFPK。有关瑞萨密钥封装服务的更多信息，请参见密钥封装服务的常见问题解答以及每个 MCU/MPU 的附加信息（*表 1-1 MCU/MPU 相关信息*）。

3.5 [生成 KUK] 选项卡

该选项卡以二进制 *.key 文件形式生成升级密钥 (KUK) 文件。该文件不仅用于安全安装 KUK，还用于准备其他密钥进行安全密钥升级。

图 3-5 [生成 KUK] 选项卡

编号	项目	说明
(1)	KUK 输入格式	选择是使用 (2) 中的输入值作为 KUK 值还是使用工具生成的随机数值。无法保证该工具随机生成的随机数的具体值。
(2)	KUK 值	当在 (1) 中选择了**使用指定的值**时，将使用文本框中输入的值作为 KUK 值。该值必须以大端格式指定为 32 字节十六进制值。
(3)	输出文件 (.key)	选择要输出的 KUK 文件的路径和文件名。输出文件的扩展名必须设置为 .key。
(4)	**生成 KUK 密钥文件**	根据上面输入的信息生成 KUK 文件。 在 [概要] 选项卡上选择 MCU/MPU 以及加密引擎时启用。

3.6 [封装密钥] 选项卡

使用该选项卡加密用户密钥，并生成安全安装或升级所需的文件。

图 3-6 [封装密钥] 选项卡

编号	项目	说明
(1)	**[密钥类型]** 和 **[密钥数据文件]** 选项卡	在 **[密钥类型]** 选项卡中选择了要加密的用户密钥类型之后，在 **[密钥数据文件]** 选项卡中输入密钥数据文件。 有关 **[密钥类型]** 选项卡的详细信息，请参见*第 3.6.1 节 [密钥类型] 选项卡*。 有关 **[密钥数据文件]** 选项卡的详细信息，请参见*第 3.6.2 节 [密钥数据文件] 选项卡*。
(2)	封装密钥	设置要用于封装的密钥。 有关设置的详细信息，请参见*第 3.6.3 节封装密钥*。
(3)	IV	设置要用于封装的初始向量 (IV)。 有关设置的详细信息，请参见*第 3.6.4 节 IV*。
(4)	输出	选择输出文件格式。请注意，并非所有器件都支持所有可能的输出文件格式。 有关详细信息，请参见*第 3.6.5 节输出*。
(5)	**生成文件**	使用 (1) 和 (2) 中的信息以 (3) 中指定的格式生成用于安装或升级的封装密钥文件。 在 [概要] 选项卡上选择 MCU/MPU 以及加密引擎时启用。

3.6.1 [密钥类型] 选项卡

使用该选项卡指定为安全安装或升级准备的密钥类型。并非所有器件产品家族都支持所有密钥类型和选项。

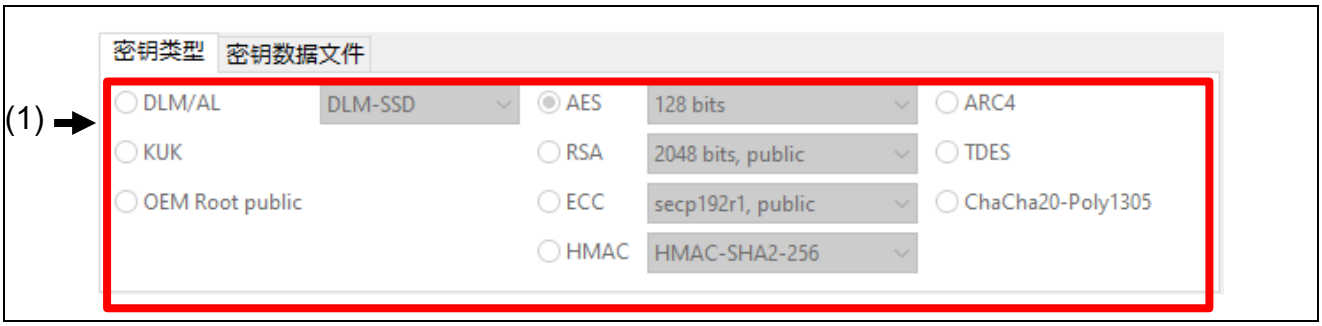

图 3-7 [密钥类型] 选项卡

编号	项目	说明
(1)	密钥类型和密钥长度选择	选择将要安装/升级的密钥的算法和密钥长度。

可以选择下列密钥类型和密钥长度。根据具体的目标 MCU/MPU，支持的密钥类型和密钥长度有所不同。有关支持的密钥类型和密钥长度的详细信息，请参见器件的硬件用户手册和附加器件信息（*表 1-1 MCU/MPU 相关信息*）。

表 3-1 选择了 DLM 时的选项

选项	说明
DLM-SSD	DLM 中 SDD 状态转换的身份验证密钥。
DLM-NSECSD	DLM 中 NSECSD 状态转换的身份验证密钥。
DLM-RMA-REQ	DLM 中 RMA-REQ 状态转换的身份验证密钥
AL2_KEY	DLM 验证级别 AL2 过渡的密钥
AL1_KEY	DLM 验证级别 AL1 过渡的密钥
RMA_KEY	DLM RAM_REQ 状态验证密钥

表 3-2 选择了 AES 时的选项

选项	说明
128 bits	AES 128 位密钥
192 bits	AES 192 位密钥
256 bits	AES 256 位密钥
128 bits, XTS	AES 128 位 XTS 密钥
256 bits, XTS	AES 256 位 XTS 密钥

表 3-3 选择了 RSA 时的选项

选项	说明
1024 bits, public	RSA 1024 位公钥
1024 bits, private	RSA 1024 位私钥
2048 bits, public	RSA 2048 位公钥
2048 bits, private	RSA 2048 位私钥
3072 bits, public	RSA 3072 位公钥
3072 bits, private	RSA 3072 位私钥
4096 bits, public	RSA 4096 位公钥
4096 bits, private	RSA 4096 位私钥
RSA-2048-public-TLS	RSA 2048 位公钥，用于 TLS API

表 3-4 选择了 ECC 时的选项

选项	说明
secp192r1, public	ECC NIST P-192 (secp192r1) 公钥
secp192r1, private	ECC NIST P-192 (secp192r1) 私钥
secp224r1, public	ECC NIST P-224 (secp224r1) 公钥
secp224r1, private	ECC NIST P-224 (secp224r1) 私钥
secp256r1, public	ECC NIST P-256 (secp256r1) 公钥
secp256r1, private	ECC NIST P-256 (secp256r1) 私钥
secp384r1, public	ECC NIST P-384 (secp384r1) 公钥
secp384r1, private	ECC NIST P-384 (secp384r1) 私钥
secp521r1, public	ECC NIST P-521 (secp521r1) 公钥
secp521r1, private	ECC NIST P-521 (secp521r1) 私钥
brainpool P256, public	ECC brainpoolP256r1 公钥
brainpool P256, private	ECC brainpoolP256r1 私钥
brainpool P384, public	ECC brainpoolP384r1 公钥
brainpool P384, private	ECC brainpoolP384r1 私钥
brainpool P512, public	ECC brainpoolP512r1 公钥
brainpool P512, private	ECC brainpoolP512r1 私钥
secp256k1, public	ECC Koblitz 曲线 secp256k1 公钥
secp256k1, public	ECC Koblitz 曲线 secp256k1 私钥
Ed25519, public	EdDSA Ed25519 公钥
Ed25519, private	EdDSA Ed25519 私钥

表 3-5 选择了 HMAC 时的选项

选项	说明
HMAC-SHA-1	HMAC-SHA1 密钥
HMAC-SHA2-224	HMAC-SHA2-224 密钥
HMAC-SHA2-256	HMAC-SHA2-256 密钥
HMAC-SHA2-384	HMAC-SHA2-384 密钥
HMAC-SHA2-512	HMAC-SHA2-512 密钥
HMAC-SHA2-512/244	HMAC-SHA2-512/224 密钥
HMAC-SHA2-512/256	HMAC-SHA2-512/256 密钥
HMAC-SHA3-224	HMAC-SHA3-224 密钥
HMAC-SHA3-256	HMAC-SHA3-256 密钥
HMAC-SHA3-384	HMAC-SHA3-384 密钥
HMAC-SHA3-512	HMAC-SHA3-512 密钥

有关如何设置各键的示例，请参见 6 使用示例。

3.6.2 [密钥数据文件] 选项卡

使用 **[密钥数据文件]** 选项卡指定为安全安装或升级准备的明文密钥材料。请注意，该选项卡的外观取决于在 **[密钥类型]** 选项卡上指定的密钥类型。

3.6.2.1 在 [密钥类型] 选项卡中选择了 DLM/KUK/AES/TDES/ARC4/ChaCha20-Poly1305/ECC 私钥时

图 3-8 选择了 DLM/KUK/AES/TDES/ARC4/ChaCha20-Poly1305/ECC 私钥时的 [密钥数据文件] 选项卡

编号	项目	说明
(1)	输入格式	选择是在 (2) 中提供包含密钥数据的密钥文件，还是在 (3) 中提供密钥的原始明文数据，亦或是通过工具使用 (4) 中指定的输出文件名生成密钥（或密钥对）。
(2)	文件名	在 (1) 中选择**文件**时，选择包含明文密钥的密钥文件。密钥文件必须为二进制文件，且扩展名为 `*.key`。
(3)	明文密钥数据	在 (1) 中选择**明文密钥**时，以十六进制格式输入明文密钥数据。数据量取决于在 **[密钥类型]** 选项卡上指定的密钥类型。
(4)	输出密钥文件	在 (1) 中选择**使用随机数**时，工具将生成密钥数据。指定工具生成的明文密钥数据的输出文件。输出明文密钥文件可以是文本文件或二进制文件。 要生成 ECC 密钥，请选择“私钥”类型。私钥和公钥将同时生成，并且会在指定的文件名中附加 `_public`（对于公钥）和 `_private`（对于私钥）。仅针对私钥生成安装/升级文件。可以使用生成的公钥文件作为输入来为公钥创建密钥安装文件。并非支持所有 ECC 密钥类型。请参见表 3-6 支持的 ECC 密钥生成。 **注**：无法保证该工具随机生成的随机数的具体值。该工具生成的密钥应仅用于原型设计和测试目的。

表 3-6 支持的 ECC 密钥生成

密钥类型和密钥长度
secp256r1、secp384r1、secp521r1 brainpool P256、brainpool P384、brainpool P512 Ed25519

请注意，macOS版本不支持生成brainpool曲线密钥。

3.6.2.2 在 [密钥类型] 选项卡中选择了 RSA 公钥时

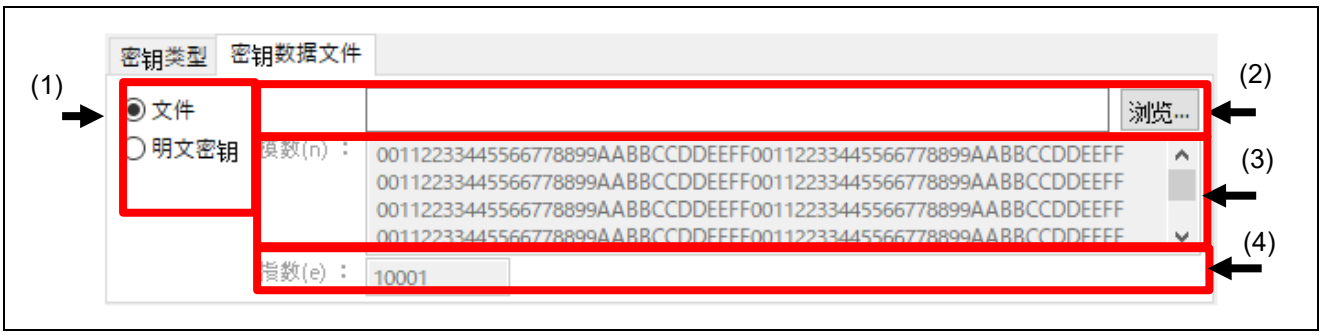

图 3-9 选择了 RSA 公钥时的 [密钥数据文件] 选项卡

编号	项目	说明
(1)	输入格式	选择是在 (2) 中提供包含密钥数据的密钥文件，还是在 (3) 和 (4) 中提供密钥的原始明文数据。
(2)	文件名	在 (1) 中选择**文件**时，选择包含明文密钥的密钥文件。密钥文件必须为二进制文件，且扩展名为 `*.key`。
(3)	模数 (n)	在 (1) 中选择**明文密钥**时，以十六进制格式输入 RSA 模数 n 的明文数据。
(4)	指数 (e)	在 (1) 中选择**明文密钥**时，以十六进制格式输入 RSA 指数 e 的明文数据。

3.6.2.3 在 [密钥类型] 选项卡中选择了 RSA 私钥时

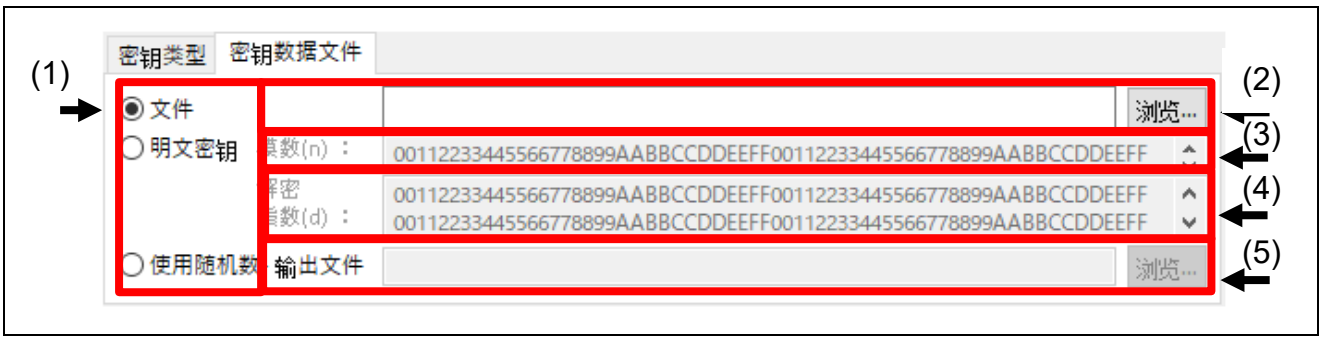

图 3-10 选择了 RSA 私钥时的 [密钥数据文件] 选项卡

编号	项目	说明
(1)	输入格式	选择是在 (2) 中提供包含密钥数据的密钥文件，还是在 (3) 和 (4) 中提供密钥的原始明文数据，亦或是通过工具使用 (5) 中指定的输出文件名生成密钥对。
(2)	文件名	在 (1) 中选择**文件**时，选择包含明文密钥的密钥文件。密钥文件必须为二进制文件，且扩展名为 `*.key`。
(3)	模数 (n)	在 (1) 中选择**明文密钥**时，以十六进制格式输入 RSA 模数 n 的明文数据。
(4)	解密指数 (d)	在 (1) 中选择**明文密钥**时，以十六进制格式输入 RSA 解密指数 d 的明文数据。
(5)	输出密钥文件	在 (1) 中选择**使用随机数**时，工具将生成密钥数据。指定工具生成的明文密钥数据的输出文件。输出明文密钥文件可以是文本文件或二进制文件。 要生成密钥，请选择“私钥”类型。私钥和公钥将同时生成，并且会在指定的文件名中附加 `_public`（对于公钥）和 `_private`（对于私钥）。仅针对私钥生成安装/升级文件。可以使用生成的公钥文件作为输入来为公钥创建密钥安装文件。 注：该工具生成的密钥应仅用于原型设计和测试目的。

3.6.2.4 在 [密钥类型] 选项卡中选择了 ECC 公钥时

图 3-11 选择了 ECC 公钥时的 [密钥数据文件] 选项卡

编号	项目	说明
(1)	输入格式	选择是在 (2) 中提供包含密钥数据的密钥文件，还是在 (3) 和 (4) 中提供密钥的原始明文数据。
(2)	文件名	在 (1) 中选择**文件**时，选择包含明文密钥的密钥文件。密钥文件必须为二进制文件，且扩展名为 `*.key`。
(3)	Qx	在 (1) 中选择**明文密钥**时，以十六进制格式输入 ECC 公钥 Qx 的明文数据。
(4)	Qy	在 (1) 中选择**明文密钥**时，以十六进制格式输入 ECC 公钥 Qy 的明文数据。

3.6.2.5 在 [密钥类型] 选项卡中选择了 OEM Root public

图 3-12 选择了 ECC 公钥时的 [密钥数据文件] 选项卡

编号	项目	说明
(1)	输入格式	选择是在 (2) 中提供包含密钥数据的密钥文件，还是在 (3) 和 (4) 中提供密钥的原始明文数据，亦或是通过工具使用 (5) 中指定的输出文件名生成密钥（或密钥对）。
(2)	文件名	在 (1) 中选择**文件**时，选择包含明文密钥的密钥文件。密钥文件必须为二进制文件，且扩展名为 `*.key`。
(3)	Qx	在 (1) 中选择**明文密钥**时，以十六进制格式输入 ECC 公钥 Qx 的明文数据。
(4)	Qy	在 (1) 中选择**明文密钥**时，以十六进制格式输入 ECC 公钥 Qy 的明文数据。
(5)	输出密钥文件	在 (1) 中选择**使用随机数**时，工具将生成密钥数据。指定工具生成的明文密钥数据的输出文件。输出明文密钥文件可以是文本文件或二进制文件。 **注**：无法保证该工具随机生成的随机数的具体值。该工具生成的密钥应仅用于原型设计和测试目的。

3.6.3 封装密钥

如果为安全安装准备了新密钥，则必须使用 UFPK 对其进行封装，并且必须提供 W-UFPK。如果为安全升级准备了新密钥，则必须使用 KUK 对其进行封装。

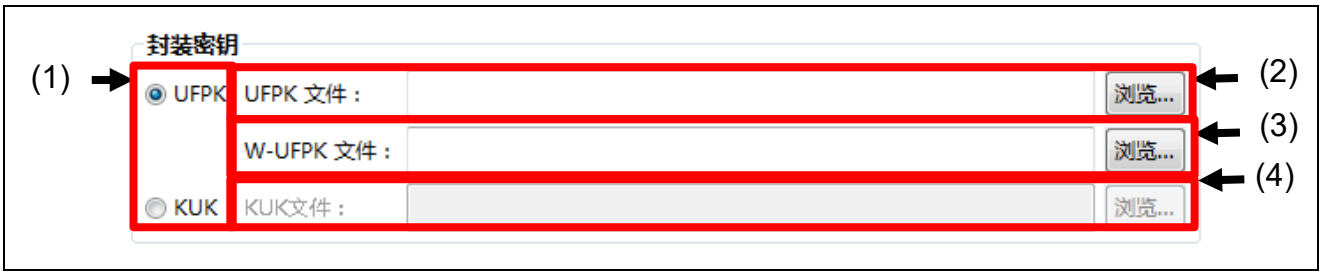

图 3-13 封装密钥选项

编号	项目	说明
(1)	封装密钥类型	选择封装密钥的类型。如果为安全安装准备了新密钥，请选择 **UFPK**，并在 (2) 中提供 UFPK 文件，在 (3) 中提供 W-UFPK 文件。如果为安全升级准备了新密钥，请选择 **KUK**，并在 (4) 中提供 KUK 文件。
(2)	UFPK 文件	如果在 (1) 中选择了 **UFPK**，则输入 UFPK `*.key` 文件。该文件是在 [**生成 UFPK**] 选项卡中生成的。
(3)	W-UFPK 文件	如果在 (1) 中选择了 **UFPK**，则输入对应于指定 UFPK 的 W-UFPK `*.key` 文件。该文件必须从瑞萨密钥封装服务中获得。
(4)	KUK 文件	如果在 (1) 中选择了 **KUK**，则输入 KUK `*.key` 文件。该文件是在 [**生成 KUK**] 选项卡中生成的。

3.6.4 IV

如果为安全安装准备了新密钥，则必须使用 UFPK 对其进行封装，并且必须提供 W-UFPK。如果为安全升级准备了新密钥，则必须使用 KUK 对其进行封装。在这两种情况下，均需要初始向量 (IV)。IV 可以指定，也可以由工具生成。

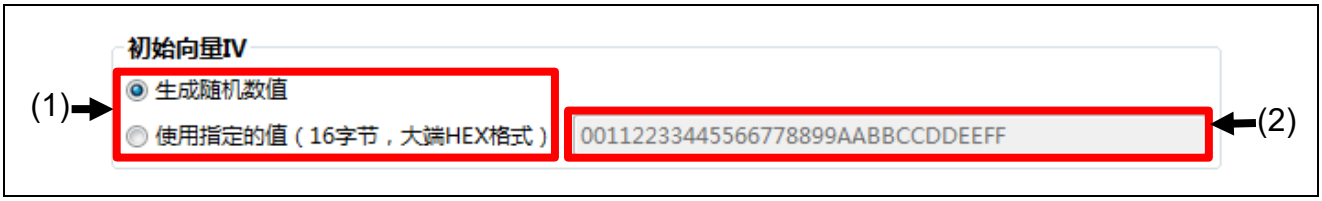

图 **3-14 IV** 选项

编号	项目	说明
(1)	IV 格式	选择是使用 (2) 中的输入值作为 IV 值还是使用工具生成的随机数值。无法保证该工具随机生成的随机数的具体值。
(2)	使用指定的值	当在 (1) 中选择了**使用指定的值**时，加密期间将使用此处输入的值作为初始向量。该值必须指定为大端格式 16 字节十六进制值。

3.6.5 输出

该部分指定为安全密钥安装或升级生成的输出文件的类型。请注意，并非所有 MCU/MPU 产品家族都支持所有选项。例如，RA 产品家族 SCE9 保护模式密钥安装仅支持通过器件编程器实现，因此，如果选择 UFPK 作为封装密钥，则仅支持 RFP 输出格式。

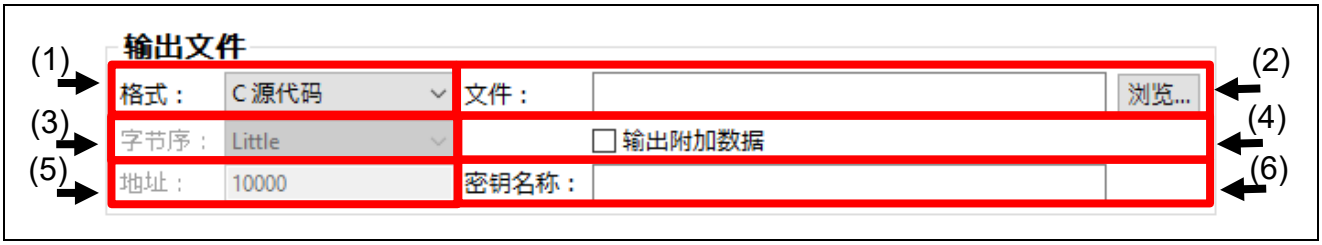

图 3-15 输出文件选项

编号	项目	说明
(1)	格式	选择输出的文件格式。有关可能的格式，请参见*表 3-7 输出文件格式*。
(2)	文件	设置输出文件名和路径。当指定非对称私钥并请求工具生成密钥对时，会在输出的加密密钥文件名中附加“_private”。
(3)	字节序	在 (1) 中选择 "摩托罗拉十六进制 "或 "二进制 "时，将启用此功能。 决定输出数据的数据序列。详见 *4.5.6**bswap** 选项*。
(4)	输出附加数据	在 (1) 中选择 "C 源"、"摩托罗拉十六进制 "或 "二进制 "时，将启用此功能。 如果输出文件名是现有文件名，密钥加密信息将附加到文件末尾。详见 *4.5.5 **fileadd** 选项*。
(5)	地址	当在 (1) 中选择了 **Motorola 十六进制**时，会启用该设置。输入要在 Motorola 十六进制文件中设置的地址。
(6)	密钥名称	当在 (1) 中选择了 **C 源文件**时，会启用该功能。该功能指定了 C 源文件和头文件中定义的结构、变量和数据大小值的 <keyname> 部分。有关如何使用 <keyname> 的详细说明，请参见*第 4.5.4.2 节 **filetype** 的 csource **选项**。*

表 3-7 输出文件格式

选项	说明
C 源文件	输出 C 源文件和头文件。
二进制	输出二进制数据。
Motorola 十六进制	输出 Motorola 十六进制文件。
RFP	以瑞萨密钥文件格式输出密钥数据。

表 3-8 字节序

选项	说明
Little	从 “Big”选项中，输出数据序列被交换为 4 个字节，并以摩托罗拉十六进制文件或二进制文件的形式输出。
Big	按字节顺序输出摩托罗拉十六进制文件或二进制数据，数据格式如表 4-62 和表 4-63 所示。

3.7 [TSIP Update]选项卡

在此选项卡中，加密TSIP Secure Update解决方案的用户程序。

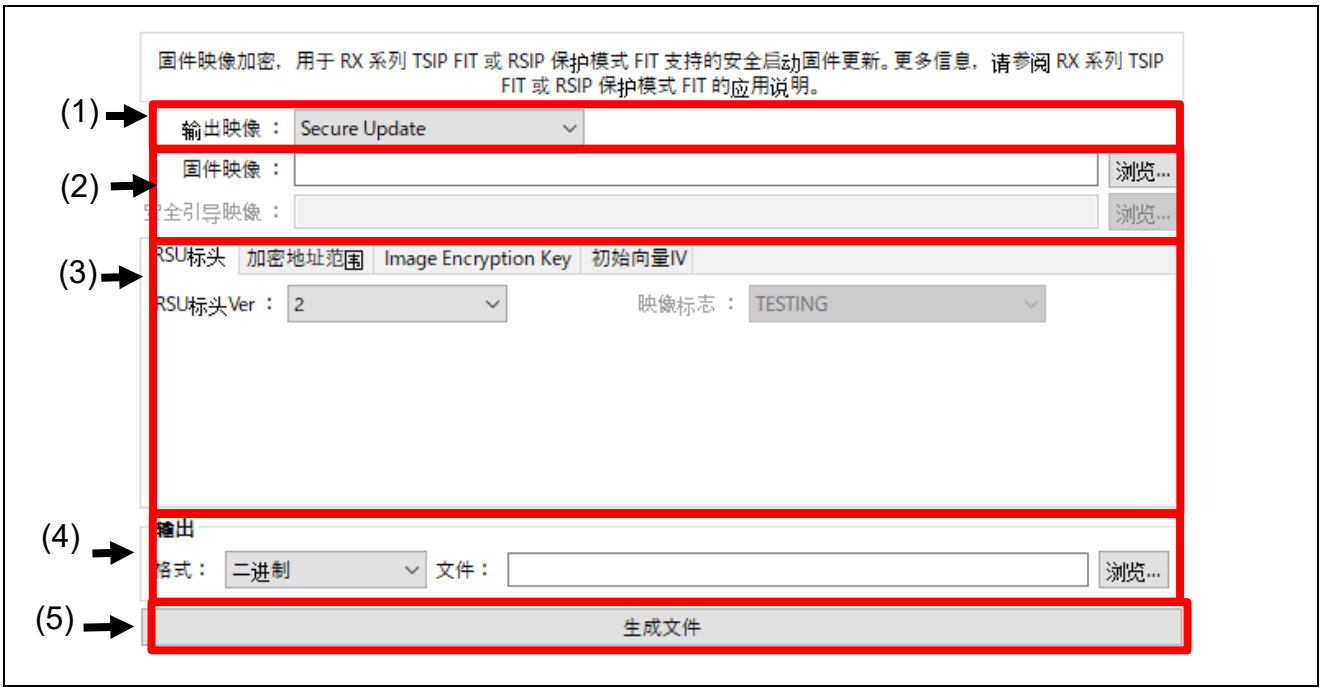

图 **3-16 [TSIP Update]**选项卡

编号	项目	说明
(1)	输出映像	指定要输出的数据类型和要输入的文件。 Factory Programming： 工厂烧录时创建映像的动作 请指定**固件映像**和**安全引导映像**。 Secure Update： 固件更新时创建映像 请指定**固件映像**。 详情请参考 *3.7.1 输出映像*。
(2)	固件映像/ 安全引导映像	**固件映像**处输入要加密的程序的 mot 文件。 **安全引导映像**处输入在输出数据选择 Factory Programming 时要附加到加密固件更新映像的安全引导程序的 mot 文件。
(3)	设置选项卡	进行加密及输出数据相关的设置。 关于各选项卡的设置，请参考 *3.7.3 [加密地址范围]选项卡* *3.7.4 [Image Encryption Key]选项卡* *3.7.5 [IV]选项卡* *3.7.6 [RSU 标头]选项卡*
(4)	输出	设置要输出的文件。 设置详情请参考 *3.7.7 输出*。
(5)	**生成文件**	生成(4)中指定格式的文件。

3.7.1 输出映像

在此选项卡中指定要输出的数据格式。

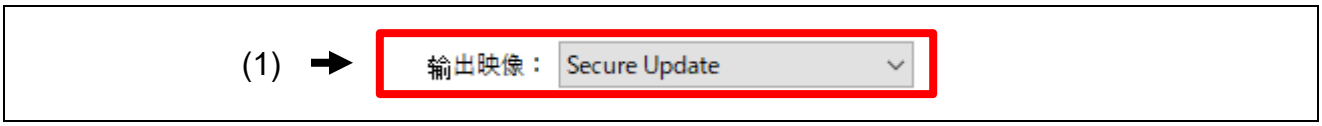

图 3-17 输出映像

编号	项目	说明
(1)	输出映像	在此选项卡中选择要创建的文件类型。

表 3-9 输出映像的显示项目

选择项目	说明
Factory Programming	设想进行工厂烧录，将固件映像的加密数据融入在安全引导中指定的文件，生成 mot 文件。
Secure Update	设想进行固件更新，将输入固件映像的 mot 文件的指定区域进行加密后生成 mot 文件，或者生成附带 RSU 标头的二进制文件。 安全引导部分未加密。写入时，推荐在安全的环境下进行。

3.7.2 固件映像/安全引导映像

指定要加密的固件映像及附加了加密数据的安全引导映像。

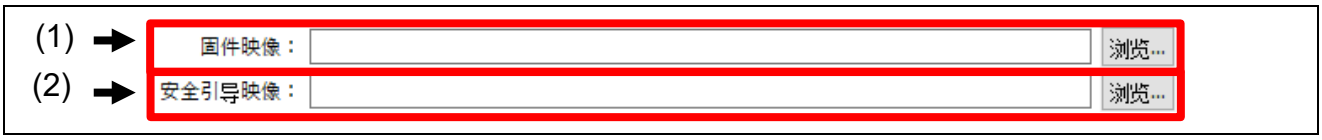

图 3-18 固件映像/安全引导映像

编号	项目	说明
(1)	固件映像	指定要加密的程序的 mot 文件。 面向 TSIP Firm Update 解决方案，对指定 mot 文件内的“加密范围”中指定范围的数据进行加密。关于加密表达式，请参考 TSIP 的 Application Note。
(2)	安全引导映像	在输出数据指定为 **Factory Programming** 时，指定与加密固件映像相匹配的安全引导映像。 安全引导印象未加密。

3.7.3 [RSU 标头]选项卡

在本节中，指定RSU标头的输出文件格式。

图 3-19 [RSU标头]选项卡

编号	项目	说明
(1)	RSU 标头 Ver	指定要附加到加密固件映像的 RSU 标头的版本。 版本可以是 1 或 2。。 关于 RSU 标头 Ver，请参考 4.6.1**ver** 选项。
(2)	映像标志	选择 RSU 标头中 Image Flags 的设置值 如果在 (2) 中选择了 **2**，则无法指定图像标志，因为不需要。 关于可以选择的值，请参考 表 3-10 RSU 标头 映像标志。

表 3-10 RSU 标头 映像标志

选择项目	说明
BLANK	没有固件升级映像。
TESTING	固件升级映像更新中、验证中。
INSTALLING	固件升级映像正在安装初始映像并进行验证。
VALID	启用固件升级映像。
INVALID	禁用固件升级映像。
END_OF_LIFE	固件升级映像寿命结束。

3.7.4 [加密地址范围]选项卡

指定固件映像的加密范围。

图 **3-20 [加密地址范围]选项卡**

编号	项目	说明
(1)	加密开始/加密结束地址	指定在固件映像中指定的 mot 文件内的加密范围。最大加密范围为 8 MB。
(2)	加密映像输出地址	在输出文件指定为 MOT 时，指定输出加密固件映像的地址。
(3)	Flash Write Size	指定闪存写入单元。 以 16 字节为单位指定大小。
(4)	Data Flash	可以指定是否将固件映像中包含的数据闪存数据添加到输出文件中。 更多信息，请参见表 3 11 数据闪存设置。

表 3-11 Data Flash 设置。

選択項目	説明
Add	将固件图像数据闪存数据添加到输出文件中。 输出图像 当选择 **Factory Programming** 时可选择。
Encrypt and append	加密固件映像数据闪存的数据并将其添加到输出文件中；仅 RSU Header Ver 2 可选。
Do not add	输出文件中不附加固件映像数据闪存数据。

3.7.5 [Image Encryption Key]选项卡

指定加密固件映像时要使用的密钥数据。

图 3-21 [Image Encryption Key]选项卡

编号	项目	说明
(1)	Key Encryption Key	指定用于加密会话密钥(Image Encryption Key)的密钥。请以 16 进制，大端 HEX 格式输入。
(2)	Image Encryption Key 的格式	选择加密固件映像的密钥值 选择 **生成随机数值** 时：使用工具的随机数值生成功能生成密钥。 选择 **使用指定的值** 时：使用在(3)中输入的 16 进制数据。
(3)	使用指定的值	在(2)中选择 **使用指定的值** 时，使用此处输入的值作为加密密钥。请以 16 进制，大端 HEX 格式输入。

3.7.6 [IV]选项卡

图 3-22 [IV]选项卡

编号	项目	说明
(1)	IV 的格式	选择加密固件映像时要使用的初始向量值 选择 **生成随机数值** 时：使用工具的随机数值生成功能生成 IV。 选择 **使用指定的值** 时：使用在(2)中输入的 16 进制数据。
(2)	使用指定的值	在(1)中选择 **使用指定的值** 时，使用此处输入的值作为加密固件映像时的初始向量。请以 16 进制，大端 HEX 格式输入。

3.7.7 输出

在本节中，指定要输出的文件格式。

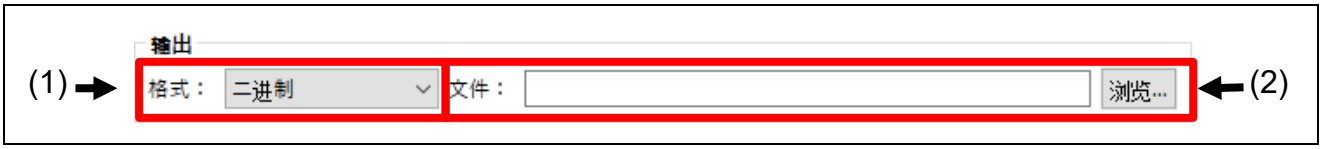

图 3-23 输出

编号	项目	说明
(1)	格式	选择要输出的文件格式 可用格式请参考*表 3-12*。
(2)	文件	设置输出文件名和路径。

表 3-12 可用格式

选择项目	描述
二进制	输出附带 RSU 文件标头的二进制数据。 RSU 仅在输出数据指定为“Secure Update”时才可以指定。
Motorola 十六进制	输出 Motorola 十六进制文件。

3.8 [FSBL]选项卡

在本选项卡中，生成可用于配备第一阶段引导加载程序(FSBL)功能的瑞萨电子器件的密钥证书和代码证书。

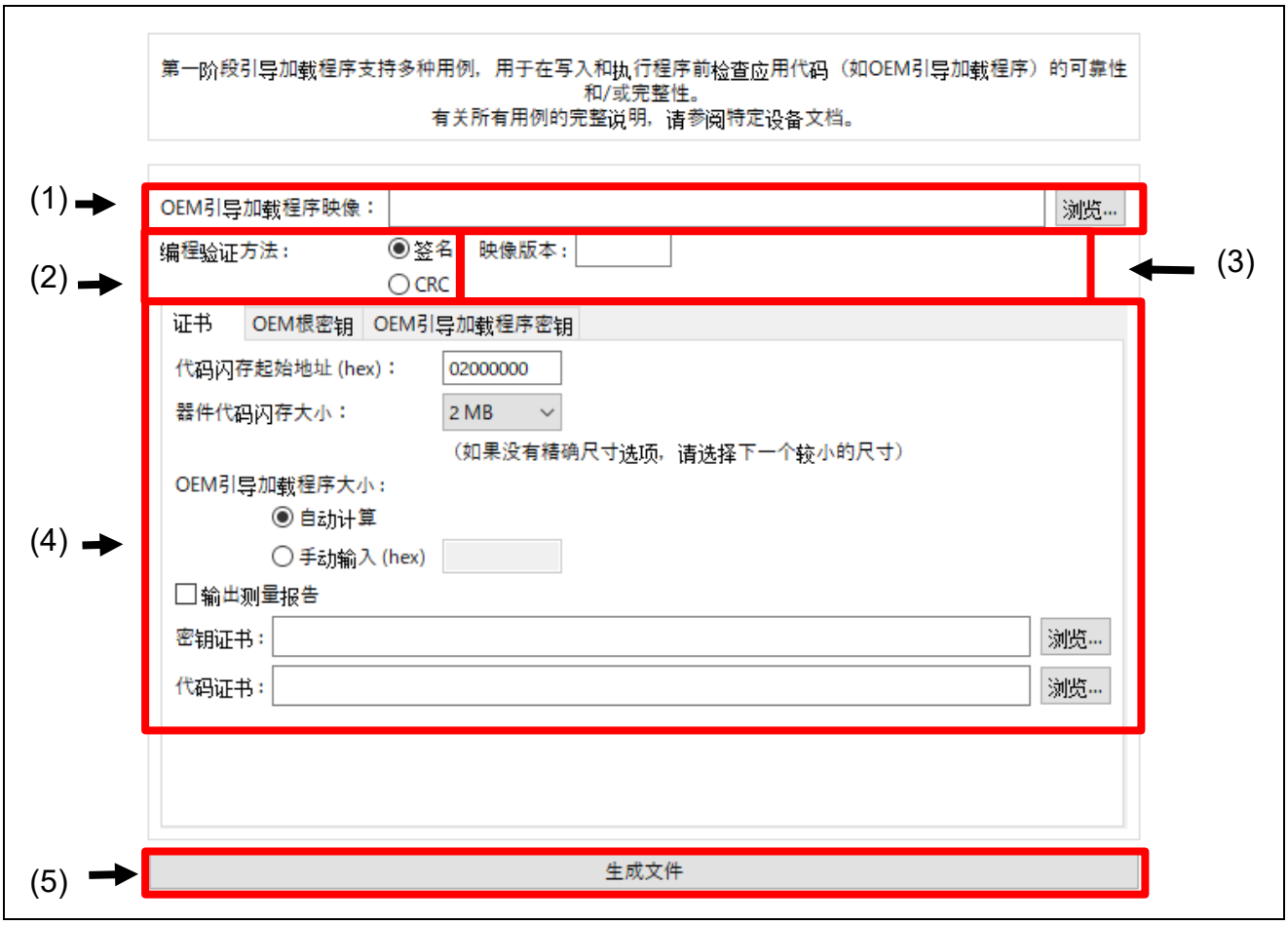

图 3-24 [FSBL]选项卡

编号	项目	说明
(1)	OEM 引导加载程序映像	指定 FSBL 验证对象的 OEM 引导加载程序的 Motorola 十六进制文件。
(2)	编程验证方法	“签名”：生成使用 ECDSA 签名的代码证书和密钥证书。 “CRC”：仅创建使用 CRC 进行简易验证所需的代码证书。 详情请参考 3.8.1 编程验证方法。
(3)	映像版本	在编程验证方法中选择“签名”时，指定代码证书中记载的 OEM 引导加载程序的版本。
(4)	[**证书**] [**OEM 根密钥**] [**OEM 引导加载程序密钥**] 选项卡	[**证书**]选项卡：指定 OEM 引导加载程序信息、密钥证书和代码证书的输出文件名。 [**OEM 根密钥**]选项卡和[**OEM 引导加载程序密钥**]选项卡： 在编程验证方法中选择“签名”时，请输入密钥信息。 有关[**证书**]选项卡的详情，请参阅 3.8.2 [**证书**]选项卡。 有关[**OEM 根密钥**]选项卡的详情，请参阅 3.8.3 [**OEM 根密钥]选项卡**。 有关[**OEM 引导加载程序密钥**]选项卡的详情，请参阅 0 **[OEM 引导加载程序密钥**]选项卡。
(5)	**生成文件**	生成在(4)中指定的密钥证书及代码证书。

3.8.1 编程验证方法

指定OEM引导加载程序的验证方法。

图 3-25 编程验证方法

编号	项目	说明
(1)	签名	使用证明 OEM 引导加载程序的密钥（OEM 引导加载程序密钥）和证明 OEM 引导加载程序密钥的密钥（OEM 根密钥），生成密钥证书和代码证书。 选择“签名”时，需要从[**OEM 根密钥**]选项卡输入 OEM 根密钥，从[**OEM 引导加载程序密钥**]选项卡输入 OEM 引导加载程序密钥。 关于证书的 Chain Of Trust 结构，请参阅具有 FSBL 功能的器件的用户手册。
(2)	CRC	计算 OEM 引导加载程序的 CRC 值，仅生成代码证书。【注】

注： 通过指定 **CRC** 生成代码证书时，将 CRC 值作为虚拟值输出至 Signer ID。在 FSBL 操作模式下选择“CRC + report measurement”，并希望使用 OEM 引导加载程序密钥输出 measurement report 时，请选择“签名”生成代码证书。

3.8.2 [证书]选项卡

指定OEM引导加载程序的信息以及密钥证书和代码证书的文件名。

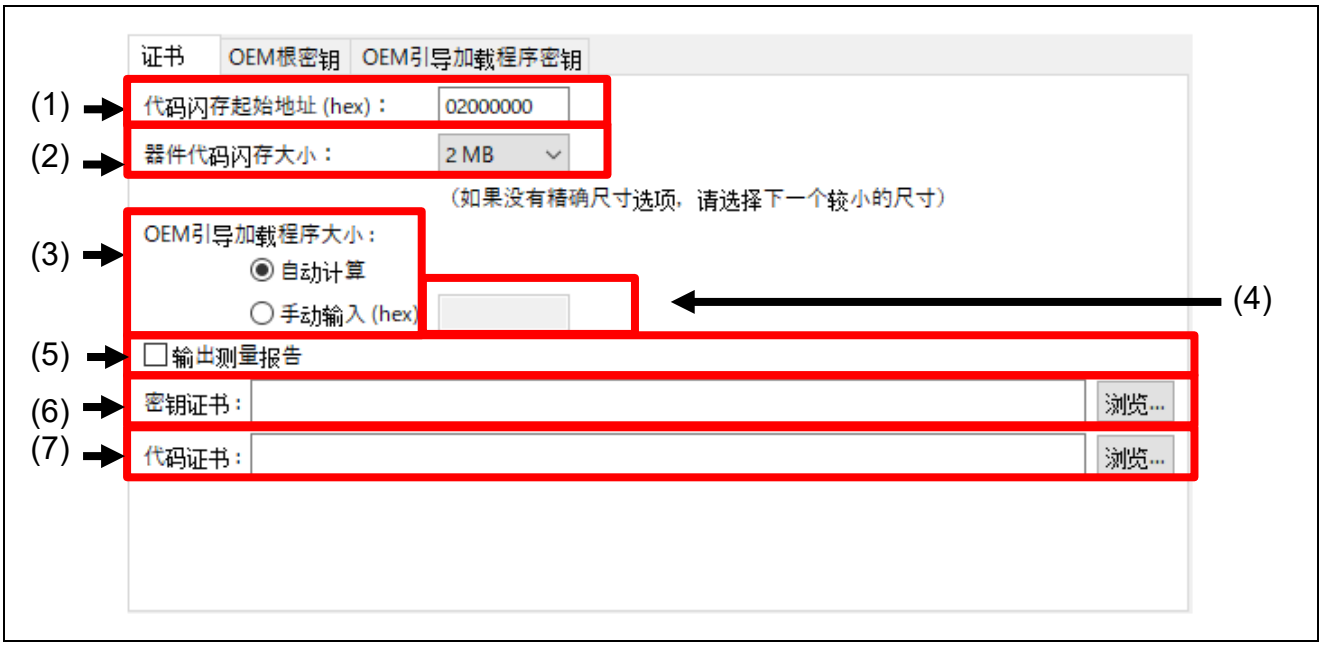

图 3-26 [证书]选项卡

编号	项目	说明
(1)	代码闪存起始地址	指定 OEM 引导加载程序的起始地址。
(2)	器件代码闪存大小	请指定要使用器件的代码闪存大小。 如果没有可用的大小，则指定一个相近的大小。
(3)	OEM 引导加载程序大小	设置 OEM 引导加载程序大小的计算方法。 选择 **自动计算** 时：根据在 OEM 引导加载程序映像中输入的 mot 文件和器件代码闪存大小计算 OEM 引导加载程序的大小。 关于 OEM 引导加载程序的大小计算方法，请参阅 4.7.2 OEM 引导加载程序的签名或 CRC 运算对象区域。 选择 **手动输入** 时：在(4)中输入的值将成为 OEM 引导加载程序的大小。
(4)	OEM 引导加载程序大小的值	在(3)中选择 **手动输入** 时，输入 OEM 引导加载程序的大小。 请输入 16 倍数大小-1 的值。
(5)	输出测量报告	如果选中此选项，则在成功生成密钥和代码证书后，会在状态窗口中输出 Measurement Report 值。
(6)	密钥证书	指定密钥证书的输出文件名。 在编程验证方法中选择“签名”时输入。
(7)	代码证书	指定代码证书的输出文件名。

3.8.3 [OEM 根密钥]选项卡

在编程验证方法中选择 **签名** 时，输入OEM根密钥。

图 **3-27 [OEM根密钥]选项卡**

编号	项目	说明
(1)	OEM 根私钥 输入方法选择	选择 OEM 根私钥的输入方法。 选择 **文件** 时：在(2)中输入密钥文件。 选择 **明文数据** 时：在(3)中输入密钥的明文数据。
(2)	文件	设置 OEM 根私钥的密钥文件。 密钥文件可以是包含公钥信息的 PEM 文件、扩展名为.txt 的文本文件或扩展名为.key 的二进制文件。 指定 PEM 文件时，无需输入 OEM 根公钥。
(3)	明文数据	在(1)中选择了 **明文数据** 时，使用 16 进制输入明文密钥数据。 OEM 根私钥为 secp256r1 时，输入私钥大小（32 字节的数据）。
(4)	OEM 根公钥 输入方法选择	选择 OEM 根公钥的输入方法。 选择 **文件** 时：在(5)中输入密钥文件。 选择 **明文数据** 时：在(6)(7)中输入密钥的明文数据。
(5)	文件	设置 OEM 根公钥的密钥文件。 密钥文件可以是扩展名为.txt 的文本文件或扩展名为.key 的二进制文件。
(6)	Qx	在(4)中选择了 **明文数据** 时，使用 16 进制输入 OEM 根公钥 Qx 的明文密钥数据。
(7)	Qy	在(4)中选择了 **明文数据** 时，使用 16 进制输入 OEM 根公钥 Qy 的明文密钥数据。

3.8.4 [OEM 引导加载程序密钥]选项卡

在编程验证方法中选择 **签名** 时，输入OEM引导加载程序密钥。

图 **3-28 [OEM引导加载程序密钥]选项卡**

编号	项目	说明
(1)	OEM 引导加载程序私钥 输入方法选择	选择 OEM 引导加载程序私钥的输入方法。 选择 **文件** 时：在(2)中输入密钥文件。 选择 **明文数据** 时：在(3)中输入密钥的明文数据。 选择 **使用随机数** 时：指定在(4)中生成的密钥的输出文件名。
(2)	文件	设置 OEM 引导加载程序私钥的密钥文件。 密钥文件可以是包含公钥信息的 PEM 文件、扩展名为.txt 的文本文件或扩展名为.key 的二进制文件。 指定 PEM 文件时，无需输入 OEM 引导加载程序公钥。
(3)	明文数据	在(1)中选择了 **明文数据** 时，使用 16 进制输入明文密钥数据。 OEM 引导加载程序私钥为 secp256r1 时，输入私钥大小（32 字节的数据）。
(4)	输出文件	在(1)中选择了 **使用随机数** 时，则在工具内生成密钥数据。指定在工具内生成的明文密钥数据的输出文件。 输出的明文密钥文件可以指定为文本文件或二进制文件。 同时生成公钥，并以指定的文件名输出文件，其中_public 为公钥，_private 为私钥。

编号	项目	说明
(5)	OEM 引导加载程序公钥 输入方法选择	选择 OEM 引导加载程序公钥的输入方法。 选择 **文件** 时：在(6)中输入密钥文件。 选择 **明文数据** 时：在(7)(8)中输入密钥的明文数据。
(6)	文件	设置 OEM 引导加载程序公钥的密钥文件。 密钥文件可以是扩展名为.key 的二进制文件或扩展名为.txt 的文本文件。
(7)	Qx	在(5)中选择了 **明文数据** 时，使用 16 进制输入 OEM 引导加载程序公钥 Qx 的明文密钥数据。
(8)	Qy	在(5)中选择了 **明文数据** 时，使用 16 进制输入 OEM 引导加载程序公钥 Qy 的明文密钥数据。

3.9 [DOTF/OTFD]选项卡

该选项卡用于加密用户数据（可执行二进制代码或应用程序数据），这些数据可在具有 DOTF 功能的瑞萨器件中使用。

图 3-29 [DOTF/OTFD]选项卡

编号	项目	说明
(1)	明文映像	指定包含要加密的映像的 Motorola 十六进制文件。
(2)	加密范围	指定在(1)中指定的文件内的加密范围。 详情请参考 3.9.1 加密范围。
(3)	执行地址	指定放置和执行数据的地址。 详情请参考 3.9.2 转发地址。
(4)	映像加密密钥	指定用于加密的密钥。 详情请参考 3.9.3 映像加密密钥。
(5)	初始向量 IV	指定用于加密的 nonce 值。 详情请参考 3.9.4 初始向量 IV。
(6)	加密映像	指定加密数据的输出文件名。
(7)	输出映像地址和内容	指定要输出文件的选项。 详情请参考 3.9.5 输出映像地址和内容。
(8)	生成加密映像文件	生成在(6)中指定的加密映像文件。

3.9.1 加密范围

指定在输入的明文映像文件内要加密的范围。

图 3-30 加密范围

编号	项目	说明
(1)	选择加密范围	选择 **加密所有数据** 时：将以明文映像输入的文件的所有数据进行加密。（*注 1） 选择 **加密地址范围** 时：在以明文映像输入的文件内，将在(2)中指定的范围进行加密。（*注 2）
(2)	加密范围	**要加密的起始地**：指定开始加密的地址。 **要加密的结束地**”：指定结束加密的地址。 起始地址应输入 16 字节对齐地址，结束地址应输入 16 字节对齐地址-1 的地址。

注1：如果输入文件的起始地址和结束地址不是16字节对齐，或者输入映像中存在没有数据的区域时，则会在待加密范围内没有数据的位置补充00数据。

注2：如果指定区域内存在没有输入映像数据的区域，则使用00数据进行补充。

3.9.2 目标地址

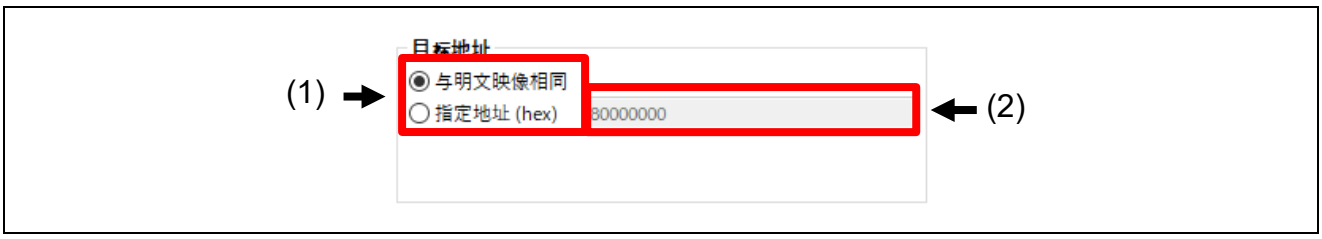

图 **3-31** 目标地址

编号	项目	说明
(1)	选择目标地址	选择 **与明文映像相同** 时：将明文映像中输入的地址指定为目标地址。 选择 **指定地址** 时：将指定的地址指定为目标地址。
(2)	执行地址	在(1)中选择 **指定地址** 时，指定执行加密程序的目标地址。

3.9.3 映像加密密钥

指定加密固件映像时要使用的密钥数据。

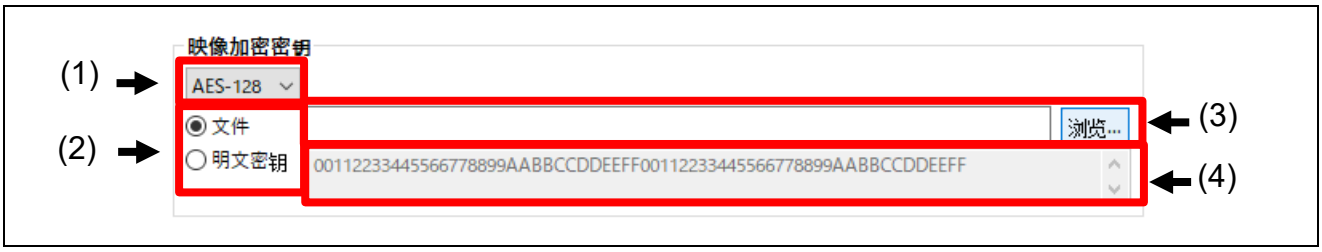

图 3-32 映像加密密钥

编号	项目	说明
(1)	指定密钥长度	指定用于加密的密钥长度。 从 **AES-128**、**AES-192**、**AES-256** 中指定。
(2)	密钥格式	选择映像加密密钥的输入方法。 选择 **文件** 时：在(3)中输入密钥文件。 选择 **明文数据** 时：在(4)中输入密钥的明文数据。
(3)	文件	在(2)中选择 **文件** 时，设置映像加密密钥的密钥文件。可以指定扩展名为 .key 的二进制文件或扩展名为 .txt 的文本文件。
(4)	明文数据	在(2)中选择了 **明文数据** 时，使用 16 进制输入明文密钥数据。 请输入大小与在(1)中选择的密钥长度相匹配的密钥数据。

3.9.4 初始向量 IV

指定加密固件映像时要使用的IV（Counter值）。

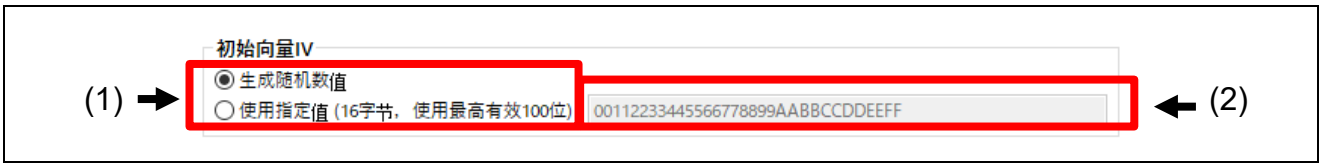

图 3-33 IV

编号	项目	说明
(1)	IV 的格式	选择加密固件映像时要使用的初始向量值。 选择 **生成随机数值** 时：使用工具的随机数值生成功能生成 IV。 选择 **使用指定的值** 时：使用在(2)中输入的十六进制数据。
(2)	使用指定的值	在(1)中选择 **使用指定的值** 时，使用此处输入的值的 100 个高位作为加密时的初始向量。请以 16 进制，大端 HEX 格式输入。

3.9.5 输出映像地址和内容

图 3-34 输出映像地址和内容

编号	项目	说明
(1)	指定地址	指定要输出的映像文件的地址。 选择 **保留原始地址** 时：将输出文件的起始地址设置为与输入文件相同的地址。 选择 **从地址 0 开始** 时：将输出文件的起始地址设置为地址 0。
(2)	包含明文数据	在(1)中选择 **保留原始地址** 时，可以选择此功能。 勾选复选框后，输出文件中允许以明文数据的形式包含输入文件中加密对象范围以外的数据。

3.10 [SFP]选项卡

在本选项卡中，将使用安全工厂编程功能对烧录所需的程序和参数信息进行加密，并输出到文件。

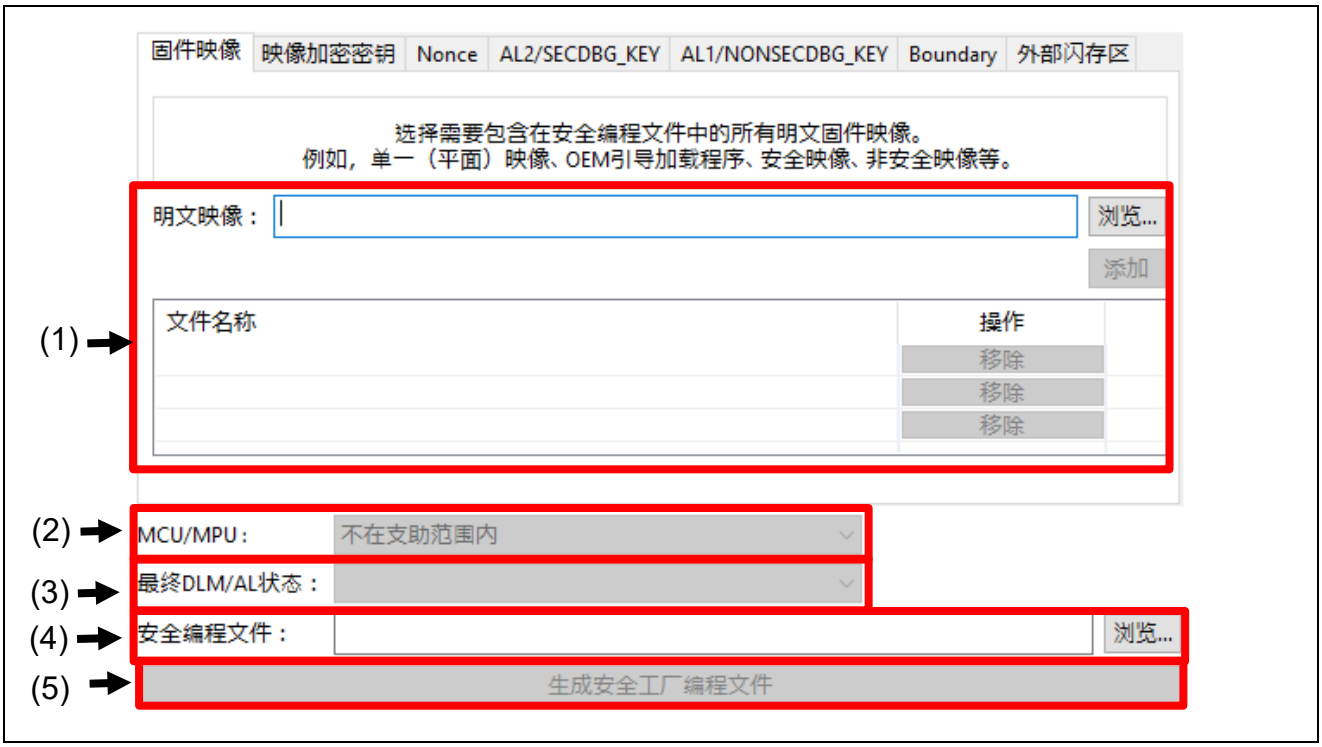

图 3-35 [SFP]选项卡

编号	项目	说明
(1)	[**固件映像**] [**映像加密密钥**] [**Nonce**] [**AL2 /SECDBG_KEY**] [**AL1 /NONSECDBG_KEY**] [**Boundary**] [**外部闪存区**] 选项卡	输入用于生成安全工厂编程文件的各种数据。 关于[**固件映像**]选项卡的详情，请参阅 3.10.1 [固件映像]选项卡。 关于[**映像加密密钥**]选项卡的详情，请参阅 3.10.2[映像加密密钥]选项卡。 关于[**Nonce**]选项卡的详情，请参阅 3.10.3 [Nonce]选项卡。 关于[**AL2/SECDBG_KEY**]选项卡的详情，请参阅 3.10.4 [AL2/SECDBG_KEY]选项卡。 关于[**AL1/NONSECDBG_KEY**]选项卡的详情，请参阅 3.10.5 [AL1/NONSECDBG_KEY]选项卡。 关于[**Boundary**]选项卡的详情，请参阅 3.10.6[Boundary]选项卡。 关于[**外部闪存区**]选项卡的详情，请参阅 3.10.7 卡。
(2)	MCU/MPU	选择使用安全工厂编程的 MCU/MPU。 可使用的 MPU/MPU 取决于在概述选项卡中选择的加密引擎。 有关可选择的 MPU/MPU 和加密引擎，请参见表 3-13 MCU/MPU。
(3)	最终 DLM/AL 状态	指定 DLM 迁移目标信息。 关于可选设置的详情，请参阅表 3-14 最终 DLM/AL 状态。
(4)	安全编程文件	指定安全工厂编程文件的输出路径和名称。
(5)	**生成安全工厂编程文件**	使用(1)至(3)中指定的信息生成安全工厂编程文件。

表 3-13 MCU/MPU

选择	概览选项卡 选择	最终 DLM/AL 状态	说明
RA8D1/M1/T1	RA Family, RSIP-E51A Security functions and Protected Mode	・OEM PL0 with AL2_KEY ・LCK_BOOT”	以 RA8D1、RA8M1 和 RA8T1 的安全工厂编程格式加密指定映像。
RA8D2/M2/T2/P1	RA Family, RSIP-E50D Security functions and Protected Mode	・OEM PL0 with AL2_KEY and AL1_KEY ・OEM PL0 with AL2_KEY ・LCK_BOOT”	以 RA8D2、RA8M2、RA8T2 和 RA8P1 的安全工厂编程格式加密指定映像。
RA4L1	RA Family, RSIP-E11A Security functions and Protected Mode	・DPL with SECDBG_KEY and NONSECDBG_KEY ・LCK_BOOT	以 RA4L1 的安全出厂编程格式对指定映像进行加密。 请务必设置 [**Boundary**] 选项卡。
不在支助范围内	如果选择的加密引擎与上述引擎不同。	-	您在概述选项卡中选择了不支持安全工厂编程功能的加密引擎。

表 3-14 最终 DLM/AL 状态

选择项目	说明
OEM PL0 with AL2_KEY and AL1_KEY	将 DLM 转至 OEM，将保护模式转至 PL0 并写入 AL2_KEY 和 AL1_KEY。
OEM PL0 with AL2_KEY	将 DLM 转至 OEM，将保护模式转至 PL0 并写入 AL2_KEY。 不使用 [**AL1/SECDBG_KEY**] 选项卡。
DPL with SECDBG_KEYand NONSECDBG_KEY	将 DLM 转移到 DPL，并写入 SECDBG_KEY 和 NONSECDBG_KEY。
LCK_BOOT	将 DLM 传输到 LCK_BOOT。 不使用 [**AL1/SECDBG_KEY**] 和 [**AL2/NONSECDBG_KEY**] 选项卡。

请注意，支持安全工厂编程功能的 MCU/MPU 并不支持所有选项。请参阅 MCU/MPU 的《硬件用户手册》，了解支持哪些选项。

3.10.1 [固件映像]选项卡

指定要加密的文件。

图 3-36 [固件映像]选项卡

编号	项目	说明
(1)	明文映像	指定要加密的用户程序。 使用浏览按钮选择文件，使用添加按钮从列表中选择文件。可以使用列表中的移除按钮删除已添加的文件。 最多可以指定 3 个文件。

3.10.2 [映像加密密钥]选项卡

指定用于加密固件映像的密钥，以及用于加密该密钥的IV、UFPK和W-UFPK。

图 3-37 [映像加密密钥]选项卡

编号	项目	说明
(1)	密钥数据	指定要用于用户程序及参数信息加密的 AES128bit 密钥数据。 选择 **文件** 时：在(2)中指定密钥文件。 选择 **明文数据** 时：将在(3)中输入的十六进制数据指定为密钥。 选择 **使用随机数** 时：自动生成密钥数据。设置在(4)中自动生成的密钥数据的文件名和输出路径。
(2)	密钥数据 文件	在(1)中选择 **文件** 时，指定 AES128bit 密钥数据文件。 可以指定二进制文件(*.key)或文本文件(*.txt)。
(3)	密钥数据 明文数据	在(1)中选择 **明文数据** 时，以十六进制数据输入 AES128bit 密钥。
(4)	密钥数据 使用随机数值	在(1)中选择 **生成随机数值** 时，设置自动生成密钥数据的文件名和输出路径。
(5)	初始向量 IV	选择加密密钥数据时的初始向量。 选择 **生成随机数值** 时：使用工具的随机数值生成 IV。 选择 **使用指定的值** 时：使用在(6)中输入的 16 进制数据。
(6)	初始向量 IV 使用指定的值	在(5)中选择“使用指定的值”时，使用此处输入的值作为加密密钥数据时的初始向量。
(7)	UFPK 文件	选择用于加密的 UFPK 文件。
(8)	W-UFPK 文件	选择 W-UFPK 文件。

3.10.3 [Nonce]选项卡

指定用于加密参数信息和用户程序的初始向量值。

图 3-38 [Nonce]选项卡

编号	项目	说明
(1)	初始向量 IV（编程参数）	选择用于加密参数信息的初始向量。 选择 **生成随机数值** 时：使用工具的随机数值生成 IV。 选择 **使用指定的值** 时：使用在(2)中输入的 16 进制数据。
(2)	初始向量 IV（编程参数） 使用指定的值	在(1)中选择“使用指定的值”时，使用此处输入的值作为加密参数信息时的初始向量。
(3)	初始向量 IV（固件映像）	选择用于加密用户程序的初始向量。 选择 **生成随机数值** 时：使用工具的随机数值生成 IV。 选择 **使用指定的值** 时：使用在(4)中输入的 16 进制数据。
(4)	初始向量 IV（固件映像） 使用指定的值	在(3)中选择 **使用指定的值** 时，使用此处输入的值作为加密用户程序时的初始向量。
(5)	初始向量 IV（AL/DLM 密钥）	选择用于加密用户程序的初始向量。 选择 **生成随机数值** 时：使用工具的随机数值生成 IV。 选择 **使用指定的值** 时：使用在(6)中输入的 16 进制数据。
(6)	初始向量 IV（AL/DLM 密钥）	在(3)中选择 **使用指定的值** 时，使用此处输入的值作为加密用户程序时的初始向量。

3.10.4 [AL2/SECDBG_KEY]选项卡

指定AL2_KEY或SECDBG_KEY加密时使用的密钥数据。

图 3-39 [AL2/SECDBG_KEY]选项卡

编号	项目	说明
(1)	密钥数据	指定用于 AL2_KEY 或 SECDBG_KEY 加密要使用的 AES128bit 密钥数据。 选择 **文件** 时：在(2)中指定密钥文件。 选择 **明文数据** 时：将在(3)中输入的十六进制数据指定为密钥。 选择 **使用随机数** 时：自动生成密钥数据。设置在(4)中自动生成的密钥数据的文件名和输出路径。
(2)	密钥数据 文件	在(1)中选择“文件”时，指定 AES128bit 密钥数据文件。 可以指定二进制文件(*.key)或文本文件(*.txt)。
(3)	密钥数据 明文数据	在(1)中选择 **明文数据** 时，以十六进制数据输入 AES128bit 密钥。
(4)	密钥数据 使用随机数值	在(1)中选择 **使用随机数** 时，设置自动生成密钥数据的文件名和输出路径。
(5)	初始向量 IV	选择用于 AL2_KEY 或 SECDBG_KEY 加密的初始向量。 选择 **生成随机数值** 时：根据工具的随机数值生成 IV 选择 **使用指定的值** 时：使用在(6)中输入的十六进制数据
(6)	初始向量 IV 使用指定的值	在(5)中选择“使用指定的值”时，使用此处输入的值作为加密 AL2_KEY 或 SECDBG_KEY 时的初始向量。

3.10.5 [AL1/NONSECDBG_KEY]选项卡

指定AL1_KEY或NONSECDBG_KEY加密时使用的密钥数据。

注：某些 MCU/MPU 不支持此选项卡。 有关详细信息，请参阅 MCU/MPU 用户手册中的支持选项。

图 3-40 [AL1/NONSECDBG_KEY]选项卡

编号	项目	说明
(1)	密钥数据	指定用于 AL1_KEY 或 NONSECDBG_KEY 加密要使用的 AES128bit 密钥数据。 选择 **文件** 时：在(2)中指定密钥文件。 选择 **明文数据** 时：将在(3)中输入的十六进制数据指定为密钥。 选择 **使用随机数** 时：自动生成密钥数据。设置在(4)中自动生成的密钥数据的文件名和输出路径。
(2)	密钥数据 文件	在(1)中选择 **文件** 时，指定 AES128bit 密钥数据文件。 对于文件输入，可以指定二进制文件(*.key)或文本文件(*.txt)。
(3)	密钥数据 明文数据	在(1)中选择 **明文数据** 时，以十六进制数据输入 AES128bit 密钥。
(4)	密钥数据 使用随机数值	在(1)中选择 **使用随机数** 时，设置自动生成密钥数据的文件名和输出路径。
(5)	初始向量 IV	选择用于 AL1_KEY 或 NONSECDBG_KEY 加密的初始向量。 选择 **生成随机数值** 时：根据工具的随机数值生成 IV 选择 **使用指定的值** 时：使用在(6)中输入的十六进制数据
(6)	初始向量 IV 使用指定的值	在(5)中选择 **使用指定的值** 时，使用此处输入的值作为加密 AL1_KEY 或 NONSECDBG_KEY 时的初始向量。

3.10.6 [Boundary]选项卡

指定通过安全工厂编程功能发送到 MCU/MPU 的边界信息。

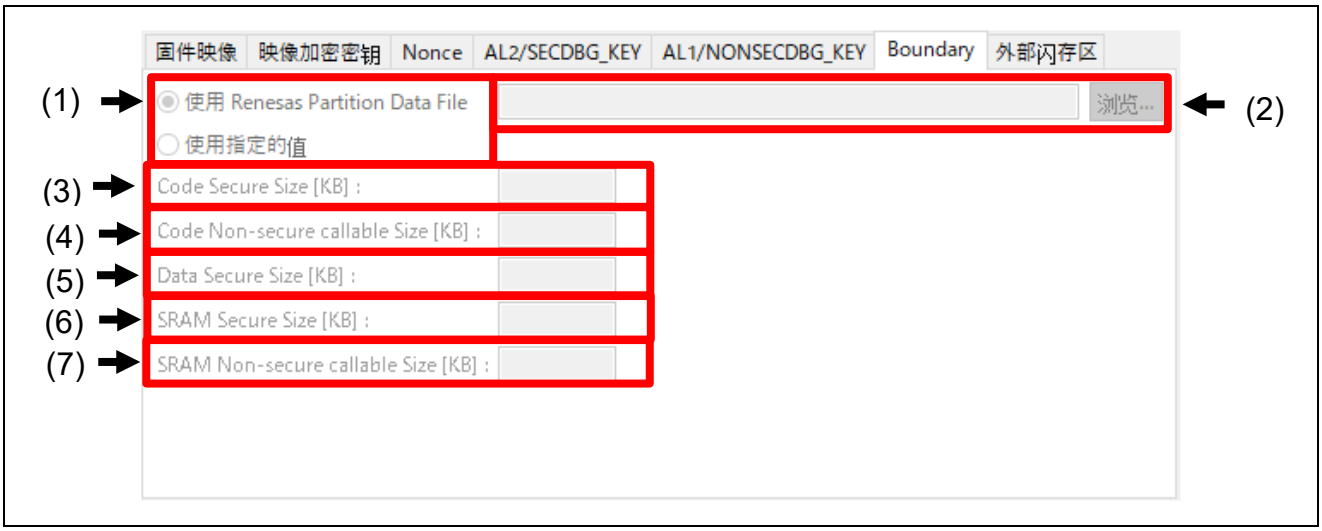

图 3-41 [Boundary]选项卡

编号	项目	说明
(1)	边界设置	**使用 Renesas Partition Data File**：(2) 中指定文件。 **使用指定值** ： 在 (3) 至 (7) 中指定每个区域的大小。
(2)	Renesas Partition Data File 输入	在 (1) 中选择 **使用 Renesas Partition Data File** 时，可指定从 e²studio 输出的 Renesas Partition Data File(`*.rpd`)。
(3)	Code Secure Size	在 (1) 中选择 **使用指定值** 时，以 KB 为单位指定代码闪存安全区域的大小。
(4)	Code Non-secure callable Size	在 (1) 中选择 **使用指定值** 时，以 KB 为单位指定代码闪存非安全可调用区域的大小。
(5)	Data Secure Size	在 (1) 中选择 **使用指定值** 时，以 KB 为单位指定数据闪存安全区域的大小。
(6)	SRAM Secure Size	在 (1) 中选择 **使用指定值** 时，SRAM 安全区域的大小以 KB 为单位指定。
(7)	SRAM Non-secure callable Size	在 (1) 中选择 **使用指定值** 时，SRAM 非安全可调用区域的大小以 KB 为单位指定。

3.10.7 [外部闪存区]选项卡

通过安全工厂编程功能对外部闪存区域的图像进行加密时，指定外部闪存区域和外部闪存的写入单元。
外部闪存区域可指定两个区域如果只使用一个区域，则输入外部闪存区域 0 的设置。

图 3-42 [外部闪存区]选项卡

编号	项目	说明
(1)	配置选项	对于支持外部闪存区安全工厂编程功能的 MCU/MPU，请指定外部闪存区的设置。 **什么都没有**：不设置外部闪存区域。 **设置一个区域**：设置为外部闪存区域 0。 使用 (2) 至 (4) 指定外部闪存区域。 **设置两个区域**：配置外部闪存区域 0 和 1 的设置。 使用 (2) 至 (7) 指定外部闪存区域。
(2)	外部闪存区 0 开始地址	在 (1) 中选择 **设置一个区域** 或 **设置两个区域** 时，指定外部闪存 0 区域的开始地址。
(3)	外部闪存区 0 结束地址	在 (1) 中选择 **设置一个区域** 或 **设置两个区域** 时，指定外部闪存 0 区域的结束地址。
(4)	外部闪存区 0 写入单位	在 (1) 中选择 **设置一个区域** 或 **设置两个区域** 时，指定外部闪存 0 区域的写入单位。写入单位必须以 16 字节为单位指定。
(5)	外部闪存区 1 开始地址	在 (1) 中选择 **设置一个区域** 时，指定外部闪存 1 区域的开始地址。
(6)	外部闪存区 1 结束地址	在 (1) 中选择 **设置一个区域** 时，指定外部闪存 1 区域的结束地址。
(7)	外部闪存区 1 写入单位	在 (1) 中选择 **设置一个区域** 时，指定外部闪存 1 区域的写入单位。写入单位必须以 16 字节为单位指定。

4. CLI 函数说明

4.1 命令行语法

skmt.exe [命令] [选项..]

注：命令、选项和参数不区分大小写。

表 4-1 命令行语法

项目	说明
skmt.exe	可执行文件名。
命令	密钥生成命令。命令以“/”或“-”开头。
选项 ..	指定的密钥生成命令有零个或多个选项。每个选项以“/”或“-”开头。

4.2 命令

命令如下表所示。

表 4-2 命令列表

命令	说明
genufpk	生成 UFPK 文件。 成功后，生成的 UFPK 将显示在控制台上。
genkuk	生成 KUK 文件。 成功后，生成的 KUK 将显示在控制台上。
genkey	加密用户密钥并输出文件。 成功后，控制台将显示 IV、W-UFPK 和加密的用户密钥（包括 MAC）。
gencert	生成密钥和代码证书，可用于具有第一阶段引导加载器（FSBL）功能的 Renesas 设备。
encdotf	对具有 DOTF 功能的 Renesas 器件的用户程序进行加密。
encsfp	对具有安全工厂编程（SFP）功能的瑞萨器件的用户程序进行加密。
enctsip	对在使用 TSIP 进行安全更新、工厂编程时要使用的程序进行加密。
calcresponse	计算 Challenge & Response 认证的响应值并输出至控制台。
H	帮助

该工具支持多种文件类型。通过使用表 4-3 中显示的扩展名指定所需的文件类型。支持绝对路径和相对路径。

表 4-3 文件类型和扩展名

文件类型	扩展名	说明
二进制密钥数据	`*.key`	明文密钥材料的二进制数据文件
瑞萨密钥文件	`*.rkey`	RFP 用于安全用户和 DLM 密钥安装（如果 MCU/MPU 支持）的文件。
Motorola 十六进制文件	`*.mot`、 `*.srec`	Motorola 十六进制文件
二进制数据	`*.bin`	二进制数据文件
C 源文件、头文件	`*.c`、`*.h`	C 源文件和头文件
文本文件	`*.txt`	用 ASCII 字符编写的文件
PEM 文件	`*.pem`	Base64 编码的 x509 ANS.1 密钥文件，用于 OpenSSL
RSU 文件	`*.rsu`	TSIP Firmware Update 使用的映像文件。
安全工厂编程文件	`*.sfp`	安全工厂编程功能使用的图像文件。
Renesas Partition Data File	`*.rpd`	安全工厂编程功能使用的边界信息文件。

4.3 genufpk 命令选项

该命令生成包含用户工厂烧录密钥 (UFPK) 的密钥文件。UFPK 可以指定，也可以由工具生成。

表 4-4 **genufpk** 选项

选项	参数	说明
ufpk	十六进制数据	为 UFPK 指定 32 字节二进制数据。 该选项为可选项。如果省略该选项，则将为 UFPK 生成随机数值。
output	文件路径	指定输出文件名。 该选项为可选项。如果省略该选项，则执行结果将输出到控制台。
nooverwrite	无	该选项为可选项。指定该选项时，如果存在输出文件，则将出现错误。

注：无法保证该工具随机生成的随机数的具体值。

省略 **ufpk** 选项的示例：

> skmt.exe /genufpk /output “D:\example\ufpk.key”

使用 **ufpk** 选项的示例：

> skmt.exe /genufpk
/ufpk “00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff”
/output “D:\example\ufpk.key”

4.4 genkuk 命令选项

该命令生成包含升级密钥 (KUK) 的密钥文件。KUK 可以指定，也可以由工具生成。

表 4-5 **genkuk** 选项

选项	参数	说明
kuk	十六进制数据	指定 KUK 使用的 32 字节二进制数据。 该选项为可选项。如果省略该选项，则将使用工具中生成的随机数值作为 KUK。
output	文件路径	指定输出文件名。 该选项为可选项。如果省略该选项，则执行结果将输出到控制台。
nooverwrite	无	该选项为可选项。指定该选项时，如果在输出文件，则将出现错误。

注：无法保证该工具随机生成的随机数的具体值。

省略 **kuk** 选项的示例：

> skmt.exe /genkuk /output “D:\example\kuk.key”

使用 **kuk** 选项的示例：

> skmt.exe /genkuk /output “D:\example\kuk.key”
/kuk “00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff”

4.5 genkey 命令选项

该命令生成用于安全密钥安装或升级的文件。

以下选项可与 **genkey** 命令一起使用。如表 4-6 中所述，一些数据输入可以指定为十六进制数据或二进制文件形式。当指定文件时，在文件路径的开头添加“`file=`”。

表 4-6 **genkey** 选项 (1)

选项	参数	说明
iv	十六进制数据/文件路径	用于加密用户或 DLM 密钥（16 字节）的初始向量 (IV)。该选项为可选项。如果省略该选项，则将使用工具中生成的随机数值作为 IV。
ufpk	文件路径	包含用于安全密钥安装的 UFPK 的密钥文件。请注意，该 UFPK 必须对应于指定的 W-UFPK。当指定 **kuk** 选项时，无法指定 **ufpk** 选项。
wufpk	文件路径	包含瑞萨密钥封装服务提供的封装 UFPK 的密钥文件。当指定 **kuk** 选项时，无法指定 **wufpk** 选项。
kuk	十六进制数据/文件路径	要用于安全密钥升级的 KUK。当指定 **ufpk** 选项时，无法指定 **kuk** 选项。
mcu	ASCII	目标 MCU/MPU。该值必须为*第 4.5.1 节 mcu 选项*中列出的选项之一。
keytype	ASCII/十六进制数据	为安装或升级准备的密钥类型。可以指定密钥名称或其数字表示。该值必须为*第 4.5.2 节 keytype 选项*中列出的选项之一。
key	十六进制数据/文件路径	DLM 密钥数据或用户密钥数据。有关要输入的密钥数据格式，请参见*第 4.5.3.1 节 十六进制数据直接输入*。该选项为可选项。如果省略该选项，则工具可能会生成密钥数据。但是，在公钥加密模式下，有一些密钥可能无法生成。有关详细信息，请参见*第 4.5.3.3 节省略 key 选项*。
filetype	ASCII	输出文件类型。请参见*第 4.5.4 节 filetype 选项*。 该选项为可选项。如果省略该选项并指定 **output** 选项，则将输出为 bin 文件。如果省略 **filetype** 和 **output** 选项，则将输出到控制台。

注：无法保证该工具随机生成的随机数的具体值。

表 4-7 **genkey** 选项 (2)

选项	参数	说明
address	十六进制数据	要安装密钥的地址。当指定 **mot** 为 **filetype** 时，必须指定该选项。
fileadd	无	**filetype** 选项为 **csource**、**bin**、**mot** 指定时可设置为。指定该选项后，如果输出文件存在，数据将附加到输出文件中。
bswap	ASCII	在 **filetype** 中指定 **bin** 或 **mot** 时可以设置该选项。 指定该选项时，输出数据将被交换。
keyname	ASCII	指定了 C 源文件和头文件中定义的结构、变量和数据大小值的 **<keyname>** 部分。当指定 **csource** 为 **filetype** 时，必须指定该选项。有关如何使用 **<keyname>** 的详细说明，请参见*第 4.5.4.2 节 filetype 的 csource 选项*。
keyfileoutput	文件路径	该选项为可选项。如果省略 **key** 选项，则将由工具生成 DLM 或用户密钥，使用该选项可指定生成的密钥数据的输出路径。输出二进制数据。
output	文件路径	指定输出文件名。该选项为可选项。如果省略该选项，则执行结果将输出到控制台。
nooverwrite	无	该选项为可选项。指定该选项时，如果存在输出文件，则将出现错误。

注：无法保证该工具随机生成的随机数的具体值。

使用 **ufpk** 选项的示例：

> skmt.exe /genkey /ufpk file=”D: \example\ufpk.key” /wufpk file=”D:\example\ufpk_enc.key” /mcu “RA-SCE9” /keytype “AES-128” /key “000102030405060708090A0B0C0D0E0F” /filetype “rfp” /output “D: \example\aes128.rkey”

使用 **kuk** 选项的示例：

> skmt.exe /genkey /kuk file=”D:\example\ufpk.key” /mcu “RA-SCE9” /keytype “AES-128” /key “000102030405060708090A0B0C0D0E0F” /filetype “csource” /output “D:\example\aes128.c”

4.5.1 mcu 选项

mcu 选项指定 MCU/MPU 类型和加密引擎。

表 4-8 **mcu** 选项

ASCII	说明
RA-RSIP-E51A	使用 RA 产品家族 RSIP-E51A 使用安全功能或保护模式时指定
RA-RSIP-E51A-CM	使用 RA 产品家族 RSIP-E51A 兼容模式时指定
RA-RSIP-E50D	使用 RA 产品家族 RSIP-E50D 使用安全功能或保护模式时指定
RA-RSIP-E50D-CM	使用 RA 产品家族 RSIP-E50D 兼容模式时指定
RA-RSIP-E11A	使用 RA 产品家族 RSIP-E11A 使用安全功能或保护模式时指定
RA-RSIP-E11A-CM	使用 RA 产品家族 RSIP-E11A 兼容模式时指定
RA-SCE9	使用 RA 产品家族 SCE9 使用安全功能或保护模式时指定
RA-SCE9-CM	使用 RA 产品家族 SCE9 兼容模式时指定
RA-SCE7	使用 RA 产品家族 SCE7 时指定
RA-SCE5_B	使用 RA 产品家族 SCE5_B 时指定
RA-SCE5	使用 RA 产品家族 SCE5 时指定
RX-TSIP	使用 RX 产品家族 TSIP 时指定
RX-TSIPLite	使用 RX 产品家族 TSIP-Lite 时指定
RX-RSIP-E11A	使用 RX 产品家族 RSIP-E11A 使用安全功能或保护模式时指定
RX-RSIP-E11A-CM	使用 RX 产品家族 RSIP-E11A 兼容模式时指定
RZ-RSIP-T2M	使用 RZ 产品家族 RZ/T2M RSIP 时指定
RZ-RSIP-T2ME	使用 RZ 产品家族 RZ/T2ME RSIP 时指定
RZ-RSIP-T2L	使用 RZ 产品家族 RZ/T2L RSIP 时指定
RZ-RSIP-N2L	使用 RZ 产品家族 RZ/N2L RSIP 时指定
RZ-TSIP	使用 RZ 产品家族 TSIP 时指定
Synergy-SCE7	使用 Synergy 平台 SCE7 时指定
Synergy-SCE5	使用 Synergy 平台 SCE5 时指定

4.5.2 keytype 选项

keytype 选项指定为安全安装或升级准备的密钥类型。该选项可以使用 ASCII 或十六进制值。为了便于阅读，建议使用 ASCII 值。

请注意，并非所有 MCU/MPU 都支持所有密钥类型。

表 4-9 用于 DLM 转换的身份验证密钥

ASCII	**keytype** 值	说明
DLM-SSD	0x01	在 DLM 中转换为 SSD 状态时使用的身份验证密钥。
DLM-NSECSD	0x02	在 DLM 中转换为 NSECSD 状态时使用的身份验证密钥。
DLM-RMA-REQ	0x03	在 DLM 中转换为 RMA_REQ 状态时使用的身份验证密钥。
DLM-AL2	0x01	DLM 验证级别 AL2 过渡的密钥(AL2_Key)
DLM-AL1	0x02	DLM 验证级别 AL1 过渡的密钥(AL1_Key)
DLM-RMA	0x03	DLM RMA_REQ 状态验证密钥(RMA_Key)

表 4-10 用户密钥 AES

ASCII	**keytype** 值	说明
AES-128	0x05	AES-128 位密钥
AES-192	0x06	AES-192 位密钥
AES-256	0x07	AES-256 位密钥
AES-128XTS	0x08	AES-128 位 XTS 密钥
AES-256XTS	0x09	AES-256 位 XTS 密钥

表 4-11 用户密钥 RSA

ASCII	**keytype** 值	说明
RSA-1024-public	0x0A	RSA 1024 位公钥
RSA-1024-private	0x0B	RSA 1024 位私钥
RSA-2048-public	0x0C	RSA 2048 位公钥
RSA-2048-private	0x0D	RSA 2048 位私钥
RSA-3072-public	0x0E	RSA 3072 位公钥
RSA-3072-private	0x0F	RSA 3072 位私钥
RSA-4096-public	0x10	RSA 4096 位公钥
RSA-4096-private	0x11	RSA 4096 位私钥
RSA-2048-public-TLS	-	RSA 2048 位公钥，用于 TLS

注：“RSA-2048-public-TLS”不能通过 **keytype** 值指定，而必须使用 ASCII 指定。

表 4-12 用户密钥 ECC

ASCII	**keytype** 值	说明
secp192r1-public	0x12	ECC NIST P-192 (secp192r1) 公钥
secp192r1-private	0x13	ECC NIST P-192 (secp192r1) 私钥
secp224r1-public	0x14	ECC NIST P-224 (secp224r1) 公钥
secp224r1-private	0x15	ECC NIST P-224 (secp224r1) 私钥
secp256r1-public	0x16	ECC NIST P-256 (secp256r1) 公钥
secp256r1-private	0x17	ECC NIST P-256 (secp256r1) 私钥
secp384r1-public	0x18	ECC NIST P-384 (secp384r1) 公钥
secp384r1-private	0x19	ECC NIST P-384 (secp384r1) 私钥
secp521r1-public	0x24	ECC NIST P-521 (secp521r1) 公钥
secp521r1-private	0x25	ECC NIST P-521 (secp521r1) 私钥
brainpoolP256r1-public	0x1C	ECC brainpoolP256r1 公钥
brainpoolP256r1-private	0x1D	ECC brainpoolP256r1 私钥
brainpoolP384r1-public	0x1E	ECC brainpoolP384r1 公钥
brainpoolP384r1-private	0x1F	ECC brainpoolP384r1 私钥
brainpoolP512r1-public	0x20	ECC brainpoolP512r1 公钥
brainpoolP512r1-private	0x21	ECC brainpoolP512r1 私钥
secp256k1-public	0x22	ECC Koblitz 曲线 secp256k1 公钥
secp256k1-private	0x23	ECC Koblitz 曲线 secp256k1 私钥
Ed25519-public	0x26	EdDSA Ed25519 公钥
Ed25519-private	0x27	EdDSA Ed25519 私钥

表 4-13 用户密钥 SHA-HMAC

ASCII	**keytype** 值	说明
HMAC-SHA1	0x00	HMAC-SHA-1 密钥
HMAC-SHA224	0x1A	HMAC-SHA2-224 密钥
HMAC-SHA256	0x1B	HMAC-SHA2-256 密钥
HMAC-SHA384	0x28	HMAC-SHA2-384 密钥
HMAC-SHA512	0x29	HMAC-SHA2-512 密钥
HMAC-SHA512-224	0x2a	HMAC-SHA2-512/224 密钥
HMAC-SHA512-256	0x2b	HMAC-SHA2-512/256 密钥
HMAC-SHA3-224	0x2c	HMAC-SHA3-224 密钥
HMAC-SHA3-256	0x2d	HMAC-SHA3-256 密钥
HMAC-SHA3-384	0x2e	HMAC-SHA3-384 密钥
HMAC-SHA3-512	0x2f	HMAC-SHA3-512 密钥

注：“HMAC-SHA1”不能通过 **keytype** 值指定，而必须使用 ASCII 指定。

表 4-14 用户密钥 ARC4

ASCII	**keytype** 值	说明
ARC4	0x00	ARC4 密钥

注：“ARC4”不能通过 **keytype** 值指定，而必须使用 ASCII 指定。

表 4-15 用户密钥 DES

ASCII	**keytype** 值	说明
TDES	0x00	三重 DES 密钥

注：“TDES”不能通过 **keytype** 值指定，而必须使用 ASCII 指定。

表 4-16 用户密钥 ChaCha20-Poly1305

ASCII	**keytype** 值	说明
CHACHA20-POLY1305	0x30	CHACHA20-POLY1305 密钥

表 4-17 升级密钥

ASCII	**keytype** 值	说明
OEM_ROOT_PK	0xFD	使用 FSBL 时将注入设备的 OEM 根公钥

表 4-18 升级密钥

ASCII	**keytype** 值	说明
key-update-key	0xFF	升级密钥

4.5.3 **key** 选项

key 选项用于指定需要为安全安装或升级准备的明文 DLM 或用户密钥。明文可以通过直接输入十六进制数据或通过二进制 `*.key` 文件指定。某些密钥类型还可以由工具生成。

4.5.3.1 十六进制数据直接输入

如果明文密钥是通过直接输入十六进制数据提供的，则必须根据指定的 **keytype** 选项提供所需的字节数，具体请参照下表。

表 4-19 DLM-SSD、DLM-NSECSD、DLM-RMA-REQ、DLM-AL2、DLM-AL1、DLM-RMA

字节	数据
0-15	DLM 密钥数据

表 4-20 AES-128

字节	数据
0-15	AES-128 位密钥数据

表 4-21 AES-192

字节	数据
0-23	AES-192 位密钥数据

表 4-22 AES-256

字节	数据
0-31	AES-256 位密钥数据

表 4-23 AES-128XTS

字节	数据
0-15	AES-128 位密钥 1
16-31	AES-128 位密钥 2

表 4-24 AES-256XTS

字节	数据
0-31	AES-256 位密钥 1
32-63	AES-256 位密钥 2

表 4-25 RSA-1024-public

字节	数据
0-127	RSA 1024 位模数 n
128-131	RSA 1024 位指数 e

表 4-26 RSA-1024-private

字节	数据
0-127	RSA 1024 位模数 n
128-255	RSA 1024 位解密指数 d

表 4-27 RSA-2048-public/RSA-2048-public-TLS

字节	数据
0-255	RSA 2048 位模数 n
256-259	RSA 2048 位指数 e

表 4-28 RSA-2048-private

字节	数据
0-255	RSA 2048 位模数 n
256-511	RSA 2048 位解密指数 d

表 4-29 RSA-3072-public

字节	数据
0-383	RSA 3072 位模数 n
384-387	RSA 3072 位指数 e

表 4-30 RSA-3072-private

字节	数据
0-383	RSA 3072 位模数 n
384-767	RSA 3072 位解密指数 d

表 4-31 RSA-4096-public

字节	数据
0-511	RSA 4096 位模数 n
512-515	RSA 4096 位指数 e

表 4-32 RSA-4096-private

字节	数据
0-511	RSA 4096 位模数 n
512-1023	RSA 4096 位解密指数 d

表 4-33 secp192r1-public

字节	数据
0-23	ECC 公钥 Qx
24-47	ECC 公钥 Qy

表 4-34 secp192r1-private

字节	数据
0-23	ECC 私钥 d

表 4-35 secp224r1-public

字节	数据
0-27	ECC 公钥 Qx
28-55	ECC 公钥 Qy

表 4-36 secp224r1-private

字节	数据
0-27	ECC 公钥 d

表 4-37 secp256r1-public/brainpoolP256r1-public/secp256k1-public/OEM_ROOT_PK

字节	数据
0-31	ECC 公钥 Qx
32-63	ECC 公钥 Qy

表 4-38 secp256r1-private/brainpoolP256r1-private/secp256k1-private/Ed25519-private

字节	数据
0-31	ECC 私钥 d

表 4-39 secp384r1-public/brainpoolP384r1-public

字节	数据
0-47	ECC 公钥 Qx
48-95	ECC 公钥 Qy

表 4-40 secp384r1-private/brainpoolP384r1-private

字节	数据
0-47	ECC 私钥 d

表 4-41 brainpoolP512r1-public

字节	数据
0-63	ECC 公钥 Qx
64-127	ECC 公钥 Qy

表 4-42 brainpoolP512r1-private

字节	数据
0-63	ECC 私钥 d

表 4-43 secp521r1-public

字节	数据
0-65	0 Padding(7bit) || ECC Public key Qx
66-131	0 Padding(7bit) || ECC Public key Qy

注： || 表示比特连接

表 4-44 secp521r1-private

字节	数据
0-65	0 Padding(7bit) || ECC Private key d【注】

注： || 表示比特连接

表 4-45 Ed25519-public

字节	数据
0-31	sign(1bit) || ECC Public key Qy(255bit)

注： || 表示比特连接

表 4-46 HMAC-SHA1

字节	数据
0-19	HMAC-SHA-1 密钥

表 4-47 HMAC-SHA224 / HMAC-SHA3-224

字节	数据
0-27	HMAC-SHA2-224 / HMAC-SHA3-224 密钥

表 4-48 HMAC-SHA256 / HMAC-SHA3-256

字节	数据
0-31	HMAC-SHA2-256 / HMAC-SHA3-256 密钥

表 4-49 HMAC-SHA384 / HMAC-SHA3-384

字节	数据
0-47	HMAC-SHA2-384 / HMAC-SHA3-384 密钥

表 4-50 HMAC-SHA512 / HMAC-SHA512-224 / HMAC-SHA512-256 / HMAC-SHA3-512

字节	数据
0-63	HMAC-SHA2-512 / 512/224 / 512/256 / HMAC-SHA3-512 密钥

表 4-51 ARC4

字节	数据
0-255	ARC4 密钥

表 4-52 TDES

字节	数据
0-7	带奇校验 1 的 56 位 DES 密钥
8-15	带奇校验 2 的 56 位 DES 密钥
16-23	带奇校验 3 的 56 位 DES 密钥

注：为每 7 位密钥数据添加奇校验。

示例：

- DES 密钥数据 = 0x00000000000000 -> 0x0101010101010101
- DES 密钥数据 = 0xFFFFFFFFFFFFFF -> 0xFEFEFEFEFEFEFEFE

表 4-53 ChaCha20-Poly1305

字节	数据
0-31	ChaCha20-Poly1305 密钥

表 4-54 升级密钥

字节	数据
0-31	升级密钥

十六进制数据的使用示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RA-SCE9" /keytype "AES-128" /key "000102030405060708090A0B0C0D0E0F" /filetype "rfp" /output "D:\example\aes128.rkey"

4.5.3.2 文件输入

key 选项支持以下文件格式的文件输入

表 4-55 Key 选项输入文件

文件类型	扩展名	格式
二进制文件	`*.key`	二进制数据文件，格式如*第 4.5.3.1 节十六进制数据直接输入*所示。
文本文件	`*.txt`	以十六进制 ASCII 字符表示字节数据的输入文件，格式如*第 4.5.3.1 节十六进制数据直接输入*所示。
PEM	`*.pem`	可以读取 OpenSSL 生成的非对称密钥的 PEM 文件格式密钥文件。仅支持*表 4-56 支持非对称密钥的 PEM 文件*中指定的密钥类型

表 4-56 支持非对称密钥的 PEM 文件

算法	keytype
RSA	RSA-1024-private、RSA-1024-public、RSA-2048-private、RSA-2048-public、RSA-3072-private、RSA-3072-public、RSA-4096-private、RSA-4096-public、RSA-2048-public-TLS
ECC	secp256r1-private、secp256r1-public、secp384r1-private、secp384r1-public、secp521r1-private、secp521r1-public、brainpoolP256r1-private、brainpoolP256r1-public、brainpoolP384r1-private、brainpoolP384r1-public、brainpoolP512r1-private、brainpoolP512r1-public、Ed25519, OEM_ROOT_PK

请注意，macOS 版本不支持为 brainpool 曲线生成密钥。

指定二进制文件的示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RA-SCE9" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "rfp" /output "D:\example\aes128.rkey"

指定文本文件的示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RA-SCE9" /keytype "AES-128" /key file="D:\example\aes128.txt" /filetype "rfp" /output "D:\example\aes128.rkey"

指定 **PEM** 文件的示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RA-SCE9" /keytype "RSA-2048-private" /key file="D:\example\rsa2048.pem" /filetype "rfp" /output "D:\example\rsa2048.rkey"

要手动创建密钥数据，请使用十六进制编辑器或二进制编辑器，并在十六进制编辑器中以大端顺序创建数据。示例如下所示。

- 手动创建 AES 128 位密钥文件的示例：
  AES 128 位密钥 `00112233445566778899AABBCCDDEEFF`

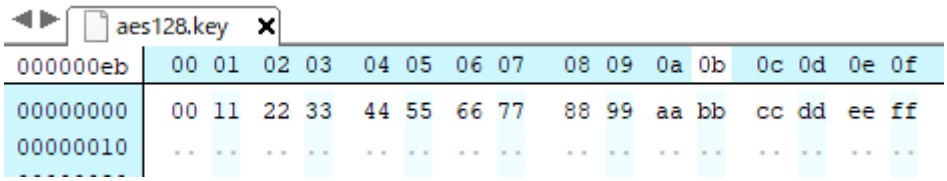

图 4-1 手动创建 AES 128 位密钥文件的示例

- 手动创建 RSA 2048 公钥文件的示例：

```
模数 bad47a84c1782e4dbdd913f2a261fc8b
        65838412c6e45a2068ed6d7f16e9cdf4
        462b39119563cafb74b9cbf25cfd544b
        dae23bff0ebe7f6441042b7e109b9a8a
        faa056821ef8efaab219d21d67634847
        85622d918d395a2a31f2ece8385a8131
        e5ff143314a82e21afd713bae817cc0e
        e3514d4839007ccb55d68409c97a18ab
        62fa6f9f89b3f94a2777c47d6136775a
        56a9a0127f682470bef831fbec4bcd7b
        5095a7823fd70745d37d1bf72b63c4b1
        b4a3d0581e74bf9ade93cc4614861755
        3931a79d92e9e488ef47223ee6f6c061
        884b13c9065b591139de13c1ea292749
        1ed00fb793cd68f463f5f64baa53916b
        46c818ab99706557a1c2d50d232577d1
指数 10001
```

rsa-2048-public.key

00000157	00 01	02 03	04 05	06 07	08 09	0a 0b	0c 0d	0e 0f
00000000	ba d4	7a 84	c1 78	2e 4d	bd d9	13 f2	a2 61	fc 8b
00000010	65 83	84 12	c6 e4	5a 20	68 ed	6d 7f	16 e9	cd f4
00000020	46 2b	39 11	95 63	ca fb	74 b9	cb f2	5c fd	54 4b
00000030	da e2	3b ff	0e be	7f 64	41 04	2b 7e	10 9b	9a 8a
00000040	fa a0	56 82	1e f8	ef aa	b2 19	d2 1d	67 63	48 47
00000050	85 62	2d 91	8d 39	5a 2a	31 f2	ec e8	38 5a	81 31
00000060	e5 ff	14 33	14 a8	2e 21	af d7	13 ba	e8 17	cc 0e
00000070	e3 51	4d 48	39 00	7c cb	55 d6	84 09	c9 7a	18 ab
00000080	62 fa	6f 9f	89 b3	f9 4a	27 77	c4 7d	61 36	77 5a
00000090	56 a9	a0 12	7f 68	24 70	be f8	31 fb	ec 4b	cd 7b
000000a0	50 95	a7 82	3f d7	07 45	d3 7d	1b f7	2b 63	c4 b1
000000b0	b4 a3	d0 58	1e 74	bf 9a	de 93	cc 46	14 86	17 55
000000c0	39 31	a7 9d	92 e9	e4 88	ef 47	22 3e	e6 f6	c0 61
000000d0	88 4b	13 c9	06 5b	59 11	39 de	13 c1	ea 29	27 49
000000e0	1e d0	0f b7	93 cd	68 f4	63 f5	f6 4b	aa 53	91 6b
000000f0	46 c8	18 ab	99 70	65 57	a1 c2	d5 0d	23 25	77 d1
00000100	00 01	00 01	.. ..	.. ..	.. ..	.. ..	.. ..	.. ..

图 4-2 手动创建 RSA 2048 公钥文件的示例

4.5.3.3 省略 key 选项

如果从命令行中省略 **key** 选项，并且为下表中指定的对称算法或非对称算法指定了 **keytype**，则工具将生成随机密钥值。

对于非对称密钥，将生成密钥对。私钥和公钥将同时生成，并且会在指定的文件名中附加 `_public`（对于公钥）和 `_private`（对于私钥）。该选项支持下表中显示的非对称 **keytype** 选项。

建议使用 **keyfileoutput** 选项创建包含明文密钥的密钥文件。

无法保证该工具随机生成的随机数的具体值。该工具生成的密钥应仅用于原型设计和测试目的。

表 4-57 支持非对称密钥生成功能的 **keytype**

算法	keytype
RSA	RSA-1024-private、RSA-2048-private、RSA-3072-private、RSA-4096-private
ECC	secp256r1-private、secp384r1-private、secp521-private、brainpoolP256r1-private、brainpoolP384r1-private、brainpoolP512r1-private、Ed25519, OEM_ROOT_PK

请注意，macOS 版本不支持 PEM 文件输入脑池曲线。

4.5.4 filetype 选项

可以使用 **filetype** 选项指定已准备密钥的输出文件格式。如果省略该选项，则输出二进制数据。如果指定的 **filetype** 与文件扩展名不匹配，则工具将返回错误。

表 4-58 **filetype** 选项

ASCII	说明
rfp	以瑞萨闪存编程器 (RFP) 可读取的文件格式输出文件。输出文件扩展名必须为 `*.rkey`。
csource	输出 C 源文件和头文件。输出文件扩展名必须为 `*.c`。
bin	输出二进制数据文件。输出文件扩展名必须为 `*.bin`。
mot	以 Motorola 十六进制格式输出二进制数据。输出文件扩展名必须为 `*.mot`。 在 address 选项中指定具有 8 个十六进制数字（32 位）的起始地址。

4.5.4.1 filetype 的 rfp 选项

该选项以瑞萨闪存编程器要使用的瑞萨密钥文件格式输出密钥数据。

RFP 文件输出示例：

```
> skmt.exe /genkey /ufpk file="D: \example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key"
  /mcu "RA-SCE9" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "rfp"
  /output "D:\example\abc.rkey"
```

4.5.4.2 filetype 的 csource 选项

该选项输出 C 源文件和头文件。

当使用 **keyname** 选项时，指定的 <keyname> 将用作结构名称、全局变量名称和字节大小定义的一部分。如果不使用 **keyname** 选项，则将使用默认文本。下表显示了这些选项。

表 4-59 **keyname** 选项用法

	设置 **keyname**	省略 **keyname**
结构名称	<keyname>_t	encrypted_user_key_data_t
全局变量名称	g_<keyname>	g_encrypted_user_key_data
encrypted_user_key 字节大小定义值	<KEYNAME>_SIZE *	ENCRYPTED_KEY_BYTE_SIZE

* <keyname> 将转换为大写字母形式

输出 C 源结构如下所示，其中 <keyname> 按照*表 4-59 **keyname** 选项用法*进行替换。

表 4-60 指定 **ufpk** 选项时采用 **csource** 格式的结构输出

名称	类型	说明
g_<keyname>	<keyname>_t	-
keytype	uint32_t	对于使用 keytype 值的安全引擎，输出 keytype 值；对于不使用 keytype 值的安全引擎，输出 0。
shared_key_number	uint32_t	输出 0。当 **mcu** 选项为“RX-TSIP”或“RX-TSIPLite”时不使用。
wufpk[32]	uint8_t	当指定 **wufpk** 选项时输出。如果未指定 **wufpk** 选项，则输出 0。
initial_vector[16]	uint8_t	用于用户密钥加密的初始向量。
encrypted_user_key [<KEYNAME>_SIZE]	uint8_t	加密的用户密钥。 <KEYNAME>_SIZE 表示加密的用户密钥大小。有关加密的用户密钥大小，请参见*表 4-64* 至*表 4-74*。
crc[4]	uint8_t	从 keytype 到 encrypted_user_key 的 CRC-32。

表 4-61 指定 **kuk** 选项时采用“**csource**”格式的结构输出

名称	类型	说明
g_<keyname>	<keyname>_t	-
keytype	uint32_t	对于使用 keytype 值的安全引擎，输出 keytype 值；对于不使用 keytype 值的安全引擎，输出 0。
shared_key_number	uint32_t	输出 0。当 **mcu** 选项位“RX-TSIP”或“RX-TSIPLite”时不使用。
initial_vector[16]	uint8_t	用于用户密钥加密的初始向量。
encrypted_user_key [<KEYNAME>_SIZE]	uint8_t	加密的用户密钥。 <KEYNAME>_SIZE 表示加密的用户密钥大小。有关加密的用户密钥大小，请参见*表 4-64* 至*表 4-74*。
crc[4]	uint8_t	从 keytype 到 encrypted_user_key 的 CRC-32。

C 源文件输出示例：

> skmt.exe /genkey /ufpk file="D: \example\ufpk.key" /wufpk file="D: \example\ufpk_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "csource" /keyname testKey /output "D:\example\abc.c"

4.5.4.3 filetype 的 bin 选项

该选项输出二进制数据文件。输出二进制数据的数据数组如下所示：

表 4-62 指定 **ufpk** 选项时采用 **bin** 格式的二进制数据输出

字节	数据
0	keytype 对于使用 keytype 值的安全引擎，输出 keytype 值；对于不使用 keytype 值的安全引擎，输出 0。
1 - 3	保留（0 填充）
4	shared_key 输出 0。当 **mcu** 选项为“RX-TSIP”或“RX-TSIPLite”时不使用。
5 - 7	保留（0 填充）
8 - 39	WUFPK 当指定 **wufpk** 选项时输出。如果未指定 **wufpk** 选项，则输出 0。
40 - 55	initial_vector
56 - (56+N-1)	encrypted_user_key
(56+N) - 56+N +3	crc 从 keytype 到 encrypted_user_key 的 CRC-32。

注：“N”与 <KEYNAME>_SIZE 值相同。

表 4-63 指定 **kuk** 选项时采用 **bin** 格式的二进制数据输出

字节	数据
0	keytype 对于使用 keytype 值的安全引擎，输出 keytype 值；对于不使用 keytype 值的安全引擎，输出 0。
1 - 3	保留（0 填充）
4	shared_key 输出 0。当 **mcu** 选项为“RX-TSIP”或“RX-TSIPLite”时不使用。
5 - 7	保留（0 填充）
8 - 23	initial_vector
24 - (24+N-1)	encrypted_user_key
(24+N) - 24+N +3	crc 从 keytype 到 encrypted_user_key 的 CRC-32。

注：“N”与 <KEYNAME>_SIZE 值相同。

二进制文件输出示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "bin" /output "D:\example\abc.bin"

4.5.4.4 filetype 的 mot 选项

该选项输出 Motorola 十六进制格式文件。文件格式在*表 4-62* 和*表 4-63* 中指定。必须使用 **address** 选项并指定十六进制 8 位（32 位）起始地址来设置数据的地址。

mot 文件输出示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "mot" /address "FFF00000" /output "D:\example\abc.mot"

每个 **keytype** 选项的 encrypted_user_key 大小如下所示。

表 4-64 DLM 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
DLM-SSD	32
DLM-NSECSD	32
DLM-RMA-REQ	32
DLM-AL2	32
DLM-AL1	32
DLM-RMA	32

表 4-65 AES 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
AES-128	32
AES-192	48
AES-256	48
AES-128XTS	48
AES-256XTS	80

表 4-66 RSA 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
RSA-1024-public	160
RSA-1024-private	272
RSA-2048-public	288
RSA-2048-private	528
RSA-3072-public	416
RSA-3072-private	784
RSA-4096-public	544
RSA-4096-private	1040
RSA 2048 位公钥，用于 TLS	288

表 4-67 ECC 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
secp192r1-public	80
secp192r1-private	48
secp224r1-public	80
secp224r1-private	48
secp256r1-public	80
secp256r1-private	48
secp384r1-public	112
secp384r1-private	64
secp521r1-public	132
secp521r1-private	66
brainpoolP256r1-public	80
brainpoolP256r1-private	48
brainpoolP384r1-public	112
brainpoolP384r1-private	64
brainpoolP512r1-public	144
brainpoolP512r1-private	80
secp256k1-public	80
secp256k1-private	48

表 4-68 Ed25519 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
Ed25519-public	48
Ed25519-private	48

表 4-69 HMAC-SHA 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
HMAC-SHA1	48
HMAC-SHA224	48
HMAC-SHA256	48
HMAC-SHA384	64
HMAC-SHA512	80
HMAC-SHA512-224	80
HMAC-SHA512-256	80
HMAC-SHA3-224	48
HMAC-SHA3-256	48
HMAC-SHA3-384	64
HMAC-SHA3-512	80

表 4-70 ARC4 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
ARC4	272

表 4-71 TDES 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
TDES	48

表 4-72 ChaCha20-Poly1305 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
ChaCha20-Poly1305	48

表 4-73 OEM_ROOT_PK 密钥的 encrypted_user_key 大小

keytype 选项	字节大小
OEM_ROOT_PK	80

表 4-74 升级密钥的 encrypted_user_key 大小

keytype 选项	字节大小
key-update-key	48

4.5.5 fileadd 选项

当文件类型选项指定为 **"csource"**、**"bin "**或 **"mot "**时，可以设置 **fileadd** 选项。当文件类型选项为 **"csource"**、**"bin "**或 **"mot "**时，可以设置 **fileadd** 选项。

当存在与输出文件相同的文件名时，生成的加密密钥信息将附加到现有数据上，而不是覆盖现有数据。

请注意：在文件类型选项中指定 **"csource "**时，不会检查现有数据和新添加数据的定义值是否相同。请使用关键字选项确保它们没有相同的定义值。

fileadd 文件输出示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "csource" /fileadd /keyname "aes128" /output "D:\example"\abc.c"

4.5.6 bswap 选项

bswap 选项指定以 ASCII 字符输出的数据序列。如果省略该选项，则输出为 **"32-big"**。

表 4-75 **bswap** 选项

ASCII	说明
32-big	将*表 4-62* 和*表 4-63* 中的每个数据按大二进制数据顺序输出到文件中。
32-little	*表 4-62* 和*表 4-63* 中的每个数据都以 4 字节交换数据序列输出到文件中。

bswap 文件输出示例：

> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key file="D:\example\aes128.key" /filetype "csource" /bswap "32-little" /keyname "aes128" /output "D:\example\abc.c"

4.5.7 keyfileoutput 选项

如果从命令行中省略 **key** 选项，则工具将生成随机密钥。**keyfileoutput** 选项用于将生成的密钥保存为二进制密钥文件或文本文件。指定的文件扩展名可以是 `*.key` 或 `*.txt`。

该工具只能生成对称密钥对和一些非对称密钥对。要生成非对称密钥对，请将私钥指定为 **keytype**。下表显示了可以生成的非对称密钥对类型。

表 4-76 可以生成的非对称密钥类型

算法	keytype
RSA	RSA-1024、RSA-2048、RSA-3072、RSA-4096
ECC	secp256r1、secp384r1、secp521r1 brainpool P256r1、brainpool P384r1、brainpool P512r1、 Ed25519, OEM_ROOT_PK

无法保证该工具随机生成的随机数的具体值。该工具生成的密钥应仅用于原型设计和测试目的。

当生成非对称密钥对且指定 **/keyfileoutput** 时，会在私钥的文件名中附加“`_private`”，在公钥的文件名中附加“`_public`”。例如，如果指定“`/keyfileoutput abc.key /output abc.rkey`”，则会生成以下文件：

- 包含明文私钥的 `abc_private.key`
- 包含明文公钥的 `abc_public.key`
- 包含私钥安装用加密密钥的 `abc_private.rkey`
- 包含公钥安装用加密密钥的 `abc_public.rkey`

要生成用于安装公钥的密钥安装文件，请使用适当的公钥 **keytype** 运行工具，并使用生成的公钥文件作为密钥数据文件。

有关输出密钥数据格式，请参见*第 4.5.3.1 节十六进制数据直接输入*中定义的格式。

对称密钥文件输出示例：

```
> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key"
  /mcu "RA-SCE9" /keytype "AES-128" /filetype "rfp" /keyfileoutput file="D:\example\aes.key"
  /output "D:\example\abc.rkey"
```

非对称密钥文件输出示例：

```
> skmt.exe /genkey /ufpk file="D:\example\ufpk.key" /wufpk file="D:\example\ufpk_enc.key"
  /mcu "RX-TSIP" /keytype "RSA-1024-private" /filetype "bin"
  /keyfileoutput file="D:\example\rsa1024.key" /output "D:\example\rsa1024_private.bin"
```

4.6 enctsip 命令选项

enctsip命令可使用下列选项。数据输入使用16进制数据或二进制文件进行指定。

表 4-77 **enctsip** 选项(1)

选项	参数	说明
mode	ASCII	指定输出文件的数据格式。 详情请参考 *4.6.2* ***mode*** *选项*
ver	Decimal data	指定要附加到输出文件的 RSU 标头的版本。 可指定版本 1 或 2。省略时指定 2。 详情请参考 *4.6.1****ver*** *选项*。
prg	File Path	指定要加密的 mot 文件。
prg_sb	File Path	**mode** 选项指定为“factory”时，指定与要加密的 mot 文件相匹配的安全引导程序的 **mot** 文件。
enckey	Hex data	指定用于加密 Session Key 的密钥，该会话密钥对在 prg 选项中指定的 **mot** 文件的数据进行加密。(16 字节)
session_key	Hex data	指定 Session Key，该会话密钥对在 prg 选项中指定的 **mot** 文件的数据进行加密。(32 字节) 该选项为可选项。如果省略该选项，则使用工具内生成的随机值作为密钥数据。*1
iv_fw	Hex data	对在 **prg** 选项中指定的 mot 文件的数据进行加密时使用的初始向量 (IV)(16 字节)。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值作为 IV。*1
startaddr	Hex data	指定在 **prg** 选项中指定的 mot 文件内加密区域的开始地址。地址必须采用 16 字节对齐。
endaddr	Hex data	指定在 **prg** 选项中指定的 **mot** 文件内加密区域的结束地址。开始地址和结束地址中指定的加密区域大小必须采用 16 字节对齐。 最大加密范围为 8MB。
destaddr	Hex data	**filetype** 选项指定为“**mot**”时，指定加密用户程序的输出地址。
imgflg	ASCII / Hex data	指定要在 RSU 标头的 Image Flags 中输入的值。 详情请参考 *4.6.3 imgflg 选项*。
filetype	ASCII	指定要输出的文件类型。详情请参考*4.6.4 filetype 选项*。
flash_wsize	Decimal data	只能为 ver 选项 2 指定。指定以 16 字节或更多字节为单位的值。 该选项可以省略。 如果省略，则指定 256。
df_ena	ASCII	该选项将 prg 选项指定的要加密的 mot 文件中数据清除区的数据追加到输出文件中。详见 *4.6.5* ***df_ena*** *选项*。

注：1. 工具内生成的随机值并不能保证足够的随机性。

表 4-78 **enctsip** 选项(2)

选项	参数	说明
motoffset	-	只能与 **ver** 选项 2 一起指定。 添加 **motoffset** 选项后，在将 rsu 文件转换为 mot 文件时，地址偏移将启用；不能与 **df_ena** 选项同时指定。
output	File Path	指定输出文件名。
nooverwrite	无	该选项为可选项。指定该选项时，如果存在输出文件，则会发生错误。

注：1. 工具内生成的随机值并不能保证足够的随机性。

设置mode factory时的使用示例：

```
> skmt.exe /enctsip /mode "factory" /ver "1" /prg "D:\example\userprog.mot"
 /prg_sb "D:\example\secureboot.mot" /enckey "0123456789ABCDEF0123456789ABCDEF"
 /startaddr "FFF80300" /endaddr "FFFEFFFF" /destaddr "FFF00300"
 /filetype "mot" /output "D:\example\factory.mot"
```

设置mode update时的使用示例：

```
> skmt.exe /enctsip /mode "update" /ver "1" /prg "D:\example\userprog.mot"
 /enckey "0123456789ABCDEF0123456789ABCDEF"
 /startaddr "FFF80300" /endaddr "FFFEFFFF" /filetype "rsu" /output "D:\example\update.rsu"
```

4.6.1 ver 选项

ver 选项指定 RSU 文件的版本：RSU 文件版本 1（用 ver 选项指定版本 1 时）和版本 2（用 ver 选项指定版本 2 时）在输出文件中添加的 RSU 标头和加密数据的创建方式上有所不同。

4.6.1.1 指定 ver 选项 1 时

(1) 加密数据准备方法

加密 **prg** 选项指定的 mot 文件中 **startaddr** 和 **endaddr** 选项指定的整个区域。 没有数据的区域将填充 0xFF。

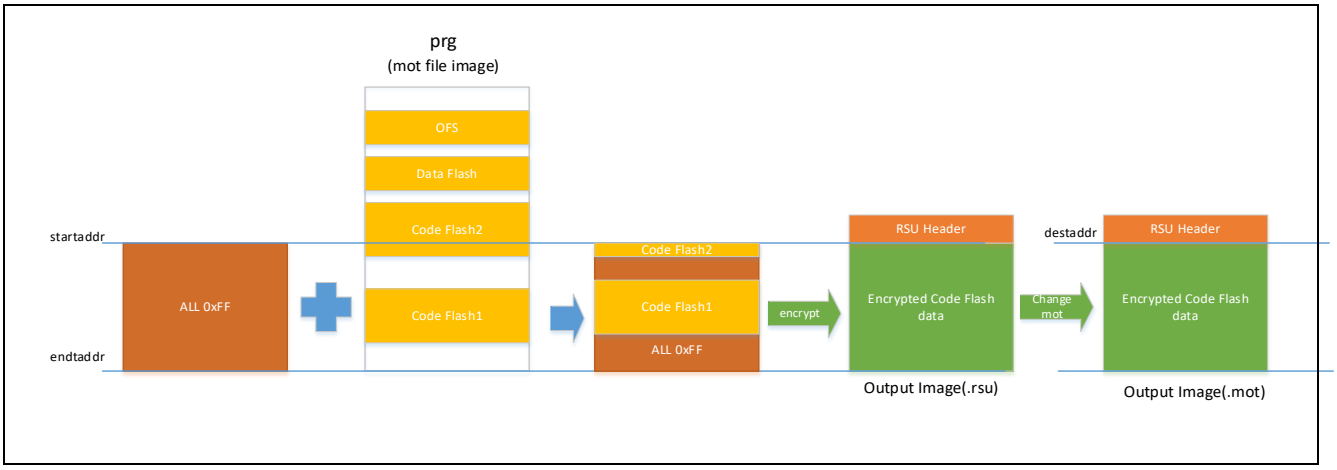

图 4-3 指定 ver 选项 1 时的加密流程

(2) RSU 标头

表 4-79 RSU 标头 V1 中显示的标头。

表 4-79 RSU 标头 V1

偏移量	构成	内容名称	长度	注
0x0000	Header	Magic Code	7	“Renesas”
0x0007		Image Flags	1	在 imgflg 选项中指定的值
0x0008	Signature	Firmware Verification Type	32	ASCII “mac-aes128-cmac-with-tsip”
0x0028		Signature size	4	未使用(0x00)。
0x002C		Signature	256	未使用(0x00)。
0x012C	Option	Dataflash Flag	4	未使用(0x00)。
0x0130		Dataflash Start Address	4	未使用(0x00)。
0x0134		Dataflash End Address	4	未使用(0x00)。
0x0138		Image Size	4	未使用(0x00)。
0x013C		Reserved	124	All 0x00
0x01B8		Format Type*1	4	要输出的文件格式 ASCII 的“rsu”(0x72, 0x73, 0x75, 0x00) 或者“mot”(0x6D, 0x6F, 0x74, 0x00)
0x01BC		IV*1	16	用户程序加密时使用的 IV
0x01CC		SessionKey0*1	16	使用 enckey 对加密用户程序时使用的密钥进行加密后的密钥
0x01DC		SessionKey1*1	16	使用 enckey 对加密用户程序时使用的密钥进行加密后的密钥
0x01EC		加密的用户程序+MAC 的字大小*1	4	加密的用户程序+用户程序加密时生成的 MAC 的合计字大小
0x01F0		MAC*1	16	用户程序加密时生成的 MAC
0x0200	Descriptor	Sequence Number	4	始终为 1
0x0204		Start Address	4	用户程序区域的开始地址
0x0208		End Address	4	用户程序区域的结束地址
0x020C		Execution Address	4	用户程序执行开始地址存储地址 (固定值 0xFFFEFFFC)
0x0210		Hardware ID	4	未使用(0x00)。
0x0214		Reserved(0x00)	236	–
0x0300	Application Binary		N	加密的用户程序
0x0300+N	Dataflash Binary		M	不支持。

注：1. 从 RX Firmware Update FIT V.1.x 中定义的*.rsu 文件格式更改为 TSIP 用格式后的参数。

4.6.1.2 指定 ver 选项 2 时

(1) 加密数据准备方法

在由 **prg** 选项指定的 mot 文件中，由 **startaddr** 和 **endaddr** 选项指定的区域会被划分为由 **flash_wsize** 选项指定大小的区块，只有存在数据的区块才会被加密。
如果数据只存在于按 **flash_wsize** 选项划分的部分块中，则没有数据的区域将填充 0xFF。
在不带 **motoffset** 选项的摩托罗拉六面体文件中输出时，rsu 文件（二进制数据）将以 **destaddr** 地址输出；在带 **motoffset** 选项的情况下，**stardaddr** 地址将转换为 **destaddr** 地址，而输入 mot 文件中的地址信息将被接收并输出。 地址，而输入的 mot 文件中的地址信息将被接收并输出。

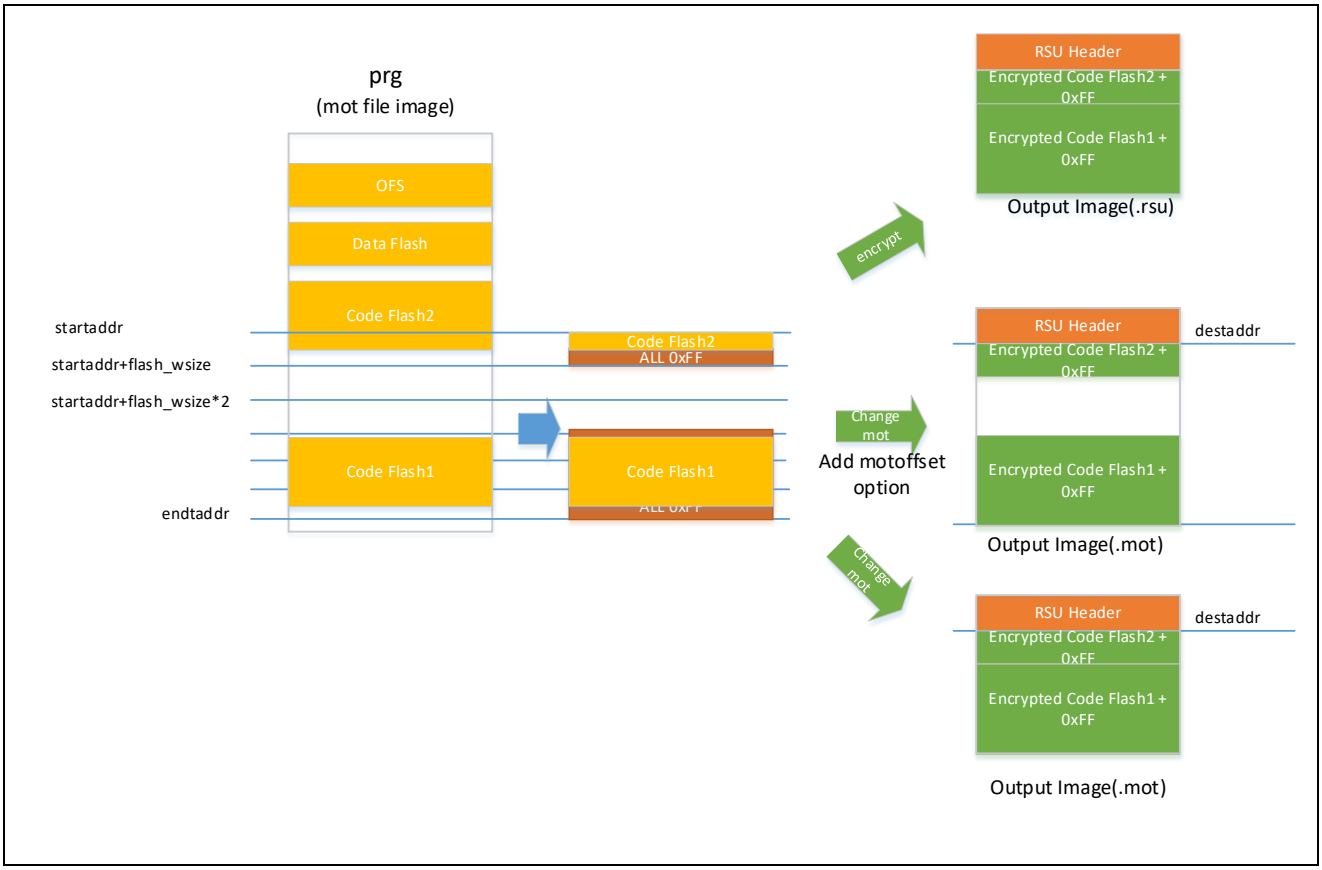

图 4-4 指定 ver 选项 2 时的加密流程

(2) RSU 标头

表 4-80 RSU 标头 V2 中显示的标头。

表 4-80 RSU 标头 V2

偏移量	构成	内容名称	长度	注
0x0000	Header	Magic Code	7	“RELFWV2”
0x0007		Reserved	1	未使用 (0x00)
0x0008	Signature	Firmware Verification Type	32	ASCII “mac-aes128-cmac-with-tsip”
0x0028		Signature size	4	未使用 (0x00)
0x002C		Signature	64	未使用 (0x00)
0x006C	Option	Reserved(0x00)	332	-
0x01B8		Format Type *1	4	要输出的文件格式 ASCII 的“rsu”(0x72, 0x73, 0x75, 0x00) 或者“mot”(0x6D, 0x6F, 0x74, 0x00)
0x01BC		IV *1	16	用户程序加密时使用的 IV
0x01CC		SessionKey0 *1	16	使用 enckey 对加密用户程序时使用的密钥进行加密后的密钥
0x01DC		SessionKey1 *1	16	使用 enckey 对加密用户程序时使用的密钥进行加密后的密钥
0x01EC		加密的用户程序+MAC 的字大小 *1	4	加密的用户程序+用户程序加密时生成的 MAC 的合计字大小
0x01F0		MAC *1	16	用户程序加密时生成的 MAC
0x0200	Descriptor	计划数据的数量	4	用户程序数据数（最多 31 个）
0x0204		Start address[0]	4	用户程序区起始地址 1
0x0208		Data size[0]	4	用户程序区数据大小 第 1 项
0x020C		Start address[1]	4	用户程序区起始地址 2
0x0210		Data size[1]	4	用户程序区数据大小 第 2 项
・ ・ ・		・ ・ ・	・ ・ ・	用户程序区起始地址 3-30 用户程序区数据大小 第 3-30 项
0x02F4		Start address[30]	4	用户程序区起始地址 31
0x02F8		Data size[30]	4	用户程序区数据大小 第 31 项
0x02FC		Reserved(0x00)	4	-
0x0300	Application Binary		N	加密用户程序
0x0300 +N	Dataflash Binary		M	不支持。

注：1. 从 RX Firmware Update FIT V.2.x 中定义的*.rsu 文件格式更改为 TSIP 用格式后的参数。

4.6.2 mode 选项

mode选项用于指定以ASCII字符输出数据的格式。

表 4-81 **mode** 选项

ASCII	说明
factory	设想进行工厂烧录，输出在 prg_sb 中指定的安全引导程序和对加密程序附加了 RSU 的 mot 文件。 文件的示意图请参考*图 4-3 mode 选项 指定 factory 时的文件生成示意图。*
update	设想市场上的 Firmware Update，输出对加密程序附加了 RSU 的二进制文件或 mot 文件。 文件的示意图请参考*图 4.4 mode 选项指定 update 且 filetype 选项指定 rsu 时的文件生成示意图* 或者 *图 4.5 mode 选项指定 update 且 filetype 选项指定 mot 时的文件生成示意图。*

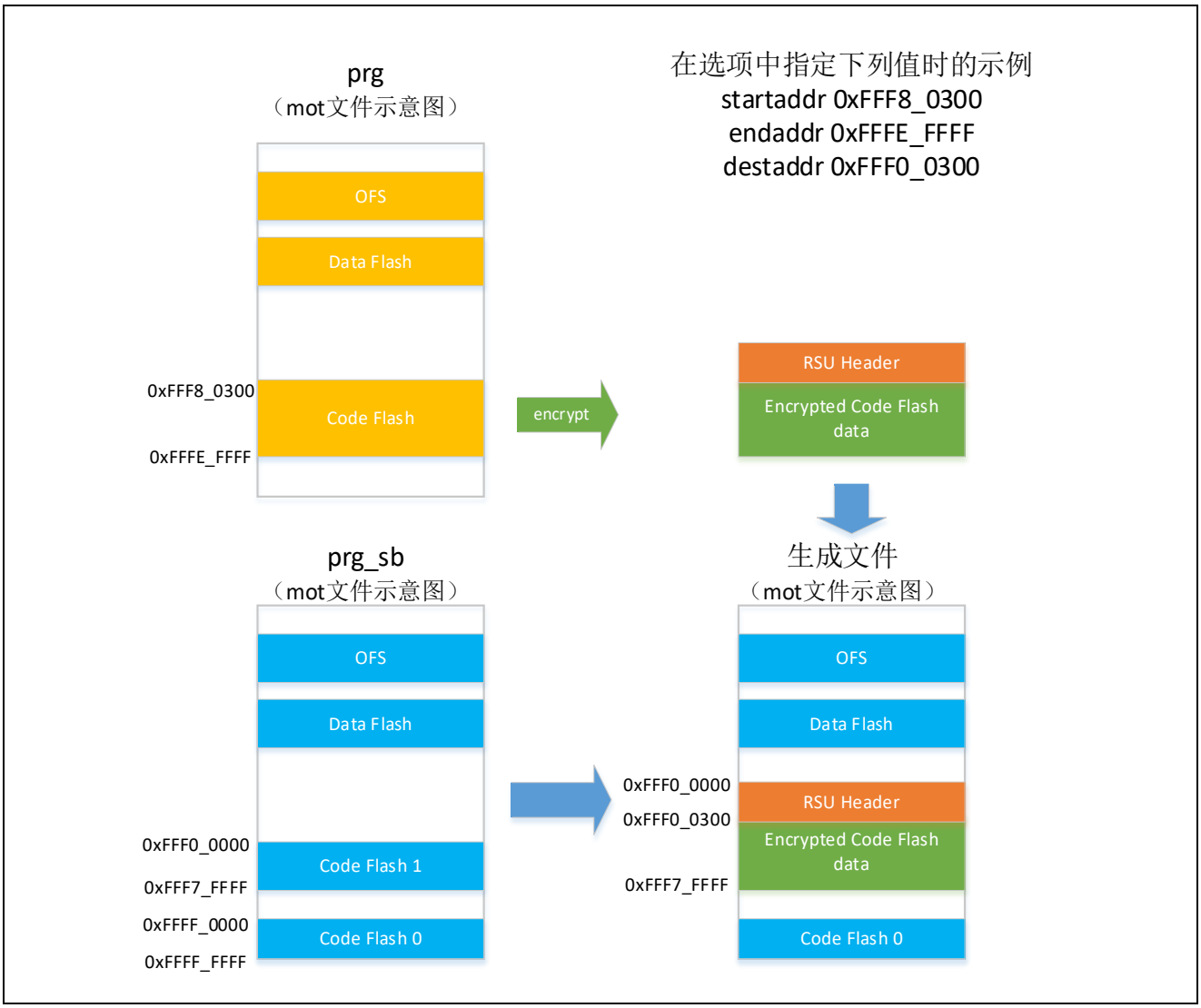

图 4-5 mode选项 指定factory时的文件生成示意图

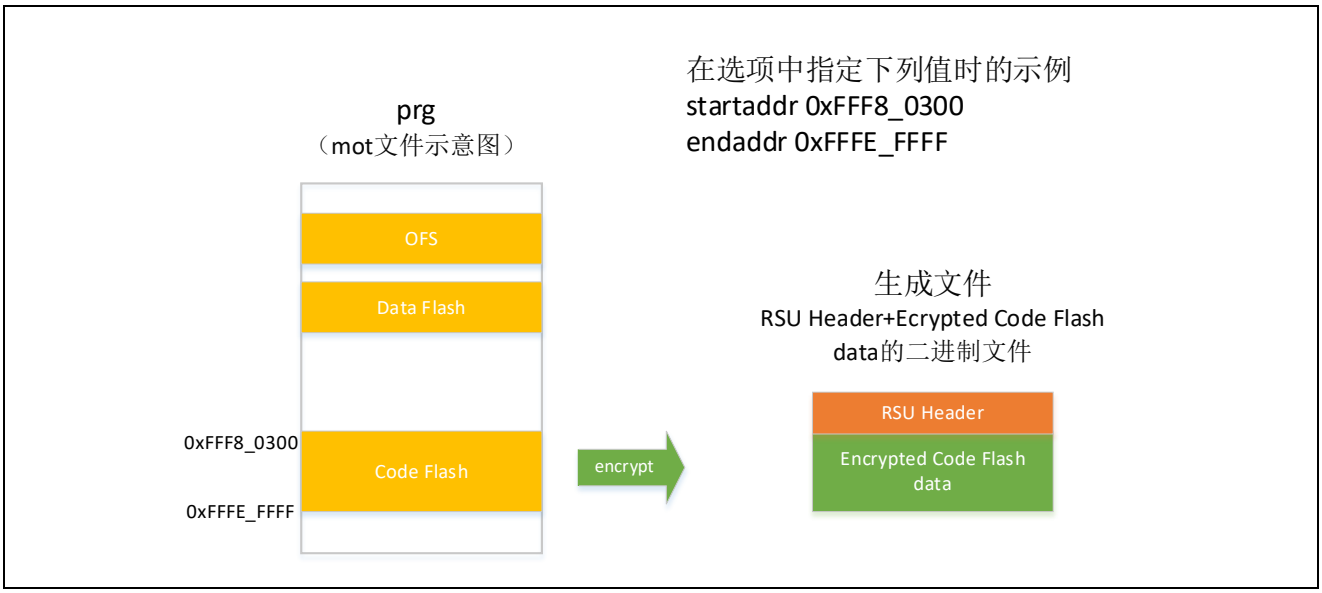

图 4-6 mode选项指定update且 filetype选项指定rsu时的文件生成示意图

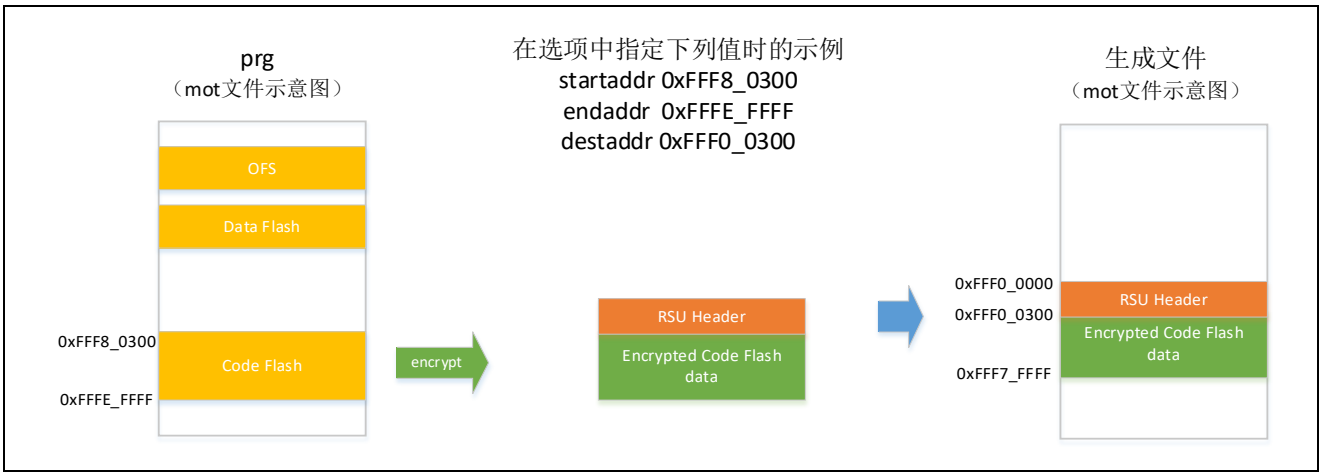

图 4-7 mode选项指定update且 filetype选项指定mot时的文件生成示意图

4.6.3 imgflg 选项

imgflg选项用于指定要写入附加到输出文件的RSU标头的Image Flags字段的值。在该选项中，可以使用ASCII值或16进制数。

表 4-82 **imgflg** 选项

ASCII	值	说明
blank	0xFF	没有写入映像。
testing	0xFE	映像更新中、验证实施中
Installing	0xFC	正在安装初始映像
valid	0xF8	启用应用程序
invalid	0xF0	禁用应用程序
end_of_life	0xE0	应用程序寿命结束

4.6.4 filetype 选项

在**filetype**选项中使用ASCII字符指定输出文件格式。
指定的**filetype**和文件名的扩展名不一致时，会发生错误。

表 4-83 **filetype** 选项

ASCII	扩展名	说明
mot	`*.mot`	输出 Motorola 十六进制格式的文件。
rsu	`*.rsu`	输出对加密用户程序附加了 RSU 标头的二进制数据。 仅在 **mode** 选项指定为 **update** 时才能指定。

4.6.5 df_ena 选项

df_ena 选项指定将 **prg** 选项指定的 mot 文件数据闪存中的数据追加或加密到输出文件中。
使用该选项时，**prg** 选项指定的文件 0x00100000 至 0x0100FFFF 中的数据将被视为数据刷新中的数据。
当使用模式选项指定工厂时，如果地址与使用 **prg_sb** 选项指定的文件中数据闪存的数据重叠，则会发生错误。

表 4-84 **df_ena** 选项

ASCII	说明
add	将由 **prg** 选项指定的 mot 文件中数据刷新区的数据添加到输出文件中。 仅在 **mode** 选项指定为 **factory** 时才能指定。
enc	将 **prg** 选项指定的 mot 文件中的闪存数据加密并附加到输出文件中。 只有在使用 **ver** 选项指定 2 时才能指定。

4.7 gencert 命令选项

gencert命令可使用下列选项。数据输入使用16进制数据或文件进行指定。指定文件时，请在文件路径的开头加上“file=”。

表 4-85 **gencert** 选项(1)

选项	参数	说明
mode	ASCII	“**signature**”（签名）：生成使用 ECDSA 签名的代码证书和密钥证书。 “**CRC**”：仅创建使用 CRC 进行简易验证所需的代码证书。 详情请参考 4.7.1 mode 选项。
loadaddr	Hex data	指定 OEM 引导加载程序的起始地址。
cfsize	Hex data	指定要使用器件的代码闪存大小。 省略 **oembl_size** 时，则将 **loadaddr** 到 **cfsize** 范围内的数据作为签名对象。 关于签名对象区域，请参阅 4.7.2 OEM 引导加载程序的签名或 CRC 运算对象区域。
oembl_size	Hex data	指定 OEM 引导加载程序的大小。大小应指定 16 字节对齐大小。 关于签名对象区域，请参阅 4.7.2 OEM 引导加载程序的签名或 CRC 运算对象区域。
ver	Decimal data	仅当 **mode** 为 **signature**”时才需要此选项。 指定代码证书中记载的 OEM 引导加载程序的版本。 可指定 1-4,294,967,295。 实际可用的版本取决于 MCU/MPU 的规格，请参阅《硬件用户手册》或 MCU/MPU 的应用说明。
oembl	File Path	指定 FSBL 验证对象的 OEM 引导加载程序的 Motorola 十六进制文件。

表 4-86 **gencert** 选项(2)

选项	参数	说明
oembl_private	Hex data / File Path	当 **mode** 为“**signature**”时需要此选项。 指定 OEM 引导加载程序私钥。 指定 HEX 数据时，请输入 32 字节 HEX 数据。 对于文件输入，可以指定 OEM 引导加载程序私钥的二进制文件(`*.key`)、文本文件(`*.txt`)或包含公钥数据的 PEM 文件(`*.pem`)。 省略本选项时，则在工具内部生成 OEM 引导加载程序的 secp256r1 密钥对。【注】
oembl_public	Hex data / File Path	当 **mode** 为“**signature**”时需要此选项。 指定 OEM 引导加载程序公钥。 指定 HEX 数据时，请输入 64 字节 HEX 数据。 对于文件输入，可以指定 Qx、Qy 密钥数据的二进制文件(`*.key`)或文本文件(`*.txt`)。 以 oembl_private 输入 PEM 文件时可以省略。
oemroot_private	Hex data / File Path	当 **mode** 为“**signature**”时需要此选项。 指定 OEM 根私钥。 指定 HEX 数据时，请输入 32 字节 HEX 数据。 对于文件输入，可以指定 OEM 根私钥的二进制文件(`*.key`)、文本文件(`*.txt`)或包含公钥数据的 PEM 文件(`*.pem`)。
oemroot_public	Hex data / File Path	当 **mode** 为“**signature**”时需要此选项。 指定 OEM 根公钥。 指定 HEX 数据时，请输入 64 字节 HEX 数据。 对于文件输入，可以指定 Qx、Qy 密钥数据的二进制文件(`*.key`)或文本文件(`*.txt`)。 以 **oembl_private** 输入 PEM 文件时可以省略。
measurementreport	-	如果指定了该选项，并且成功生成了密钥证书和代码证书，则会根据用于生成和输出的 OEM_BL 和代码证书计算出 measurement report 值。
output_codecert	File Path	指定代码证书的输出文件名。
output_keycert	File Path	当 **mode** 为“**signature**”时需要此选项。 指定密钥证书的输出文件名。
keyfileoutput	File Path	当 **mode** 为“**signature**”且省略 **oembl_private** 选项时，将输出在工具内生成和使用的 secp256r1 密钥对文件。请指定输出文件名。

注：工具内生成的随机值并不能保证足够的随机性。

设置mode CRC时的使用示例：

```
> skmt.exe /gencert /mode "CRC" /loadaddr "02000000" /oembl_size "1000"
/oembl "D:\example\user.mot" /output_codecert "D:\example\ccert.bin"
```

设置mode signature时的使用示例：

```
> skmt.exe /gencert /mode "signature" /loadaddr "02000000" /cfsize "200000" /ver "1"
/oembl "D:\example\user.mot" /oembl_private file="D:\example\is_ec256_priv.pem"
/oemroot_private file="D:\example\ms_ec256_priv.pem"
/output_codecert "D:\example\ccert.bin" /output_keycert " D:\example\kcert.bin"
```

4.7.1 mode 选项

mode选项指定要以ASCII字符输出的证书类型。

表 4-87 **mode** 选项

ASCII	说明
signature	使用签署 OEM 引导加载器的密钥（OEM 引导加载器密钥）和签署 OEM 引导加载器密钥的密钥（OEM 根密钥）生成密钥证书和代码证书。 在 **oemroot_private**、**oemroot_public** 选项中指定 OEM 根密钥对，并使用 **oembl_private**、**oembl_public** 选项指定 OEM 引导加载程序密钥对。 关于证书的 Chain Of Trust 结构，请参阅具有 FSBL 功能的器件的用户手册。
CRC	计算 OEM 引导加载程序的 CRC 值，仅生成代码证书。【注】

注： 通过指定“CRC”生成代码证书时，将 CRC 值作为虚拟值输出至 Signer ID。在 FSBL 操作模式下选择“CRC + report measurement”，并希望使用 OEM 引导加载程序密钥输出 measurement report 时，请选择“signature”生成代码证书。

4.7.2 OEM 引导加载程序的签名或 CRC 运算对象区域

OEM引导加载程序的签名或CRC运算对象为在**oembl**选项中指定的mot文件和从**oembl_size**选项和**cfsize**选项中指定的区域所提取的映像部分。

4.7.2.1 指定 oembl_size 选项时

将从**loadaddr**选项中指定的地址开始至**oembl_size**部分的映像作为签名或CRC运算对象。输入的mot文件中没有**oembl_size**指定数据的区域时，使用0xFF填充数据。

例如，指定**loadaddr**“02000000”、**oembl_size**“4000”时，以下面三个示例进行说明。

示例1：输入mot文件的映像最大至0x02002FFF

示例2：输入mot的映像最大至0x02004FFF

示例3：输入mot的映像内从0x02002000到0x02002FFF没有数据

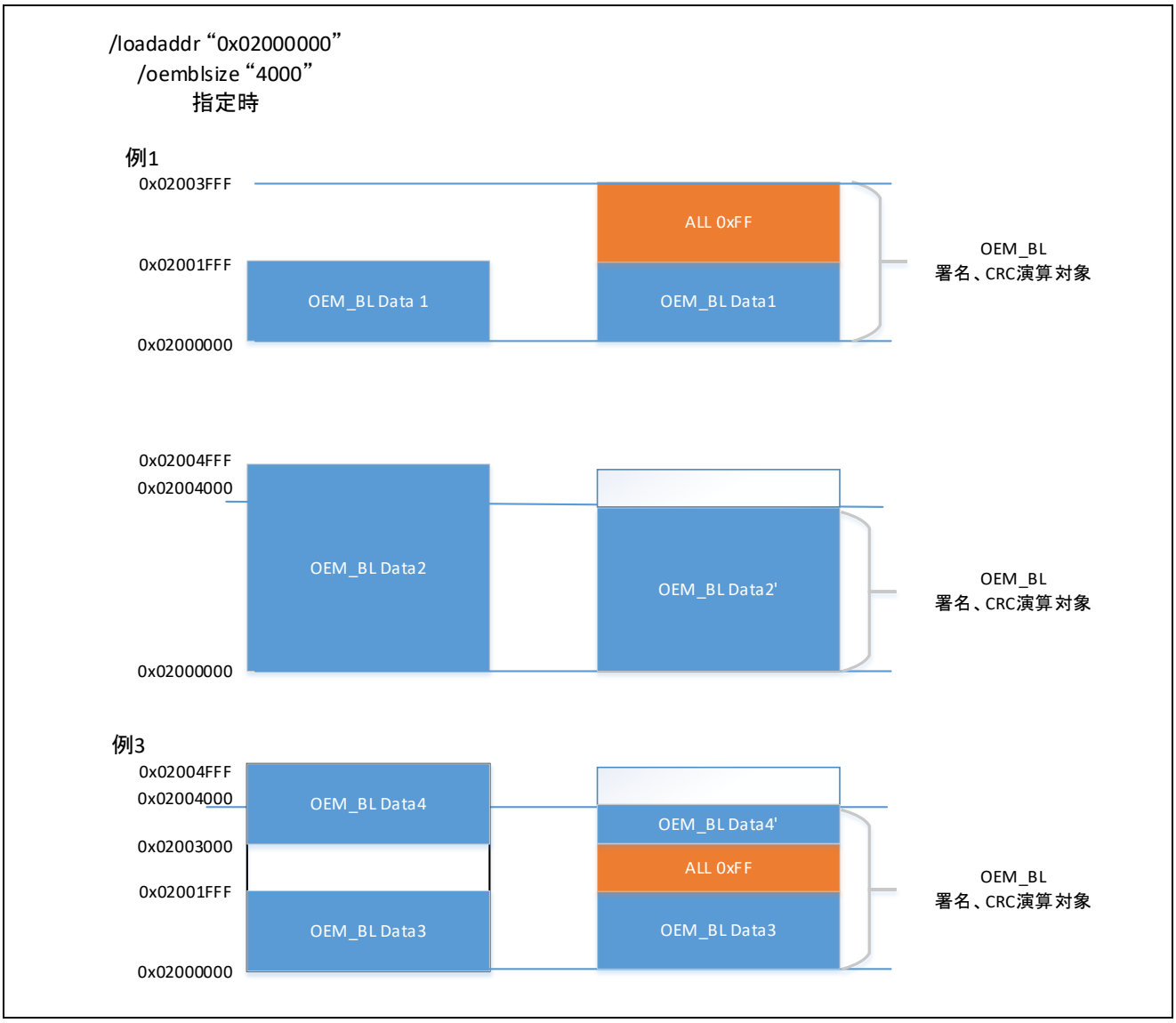

图 4-8 指定oembl_size选项时的签名、CRC运算对象

4.7.2.2 仅指定 cfsize 选项时

省略指定**oembl_size**选项时，在**oembl**选项中指定的mot文件内，将从**loadaddr**中指定的地址开始至**cfsize**指定区域内的数据作为签名或CRC运算对象。

例如，指定**loadaddr**“02000000”、**oembl_size**“10000”时，以下面四个示例进行说明。

示例1：输入mot文件的映像最大至0x02001FF8

示例2：输入mot的映像最大至0x020014FFF

示例3：输入mot的映像内0x02010000以后也存在数据

示例4：输入mot的映像内存在0x02010000之前没有数据的区域

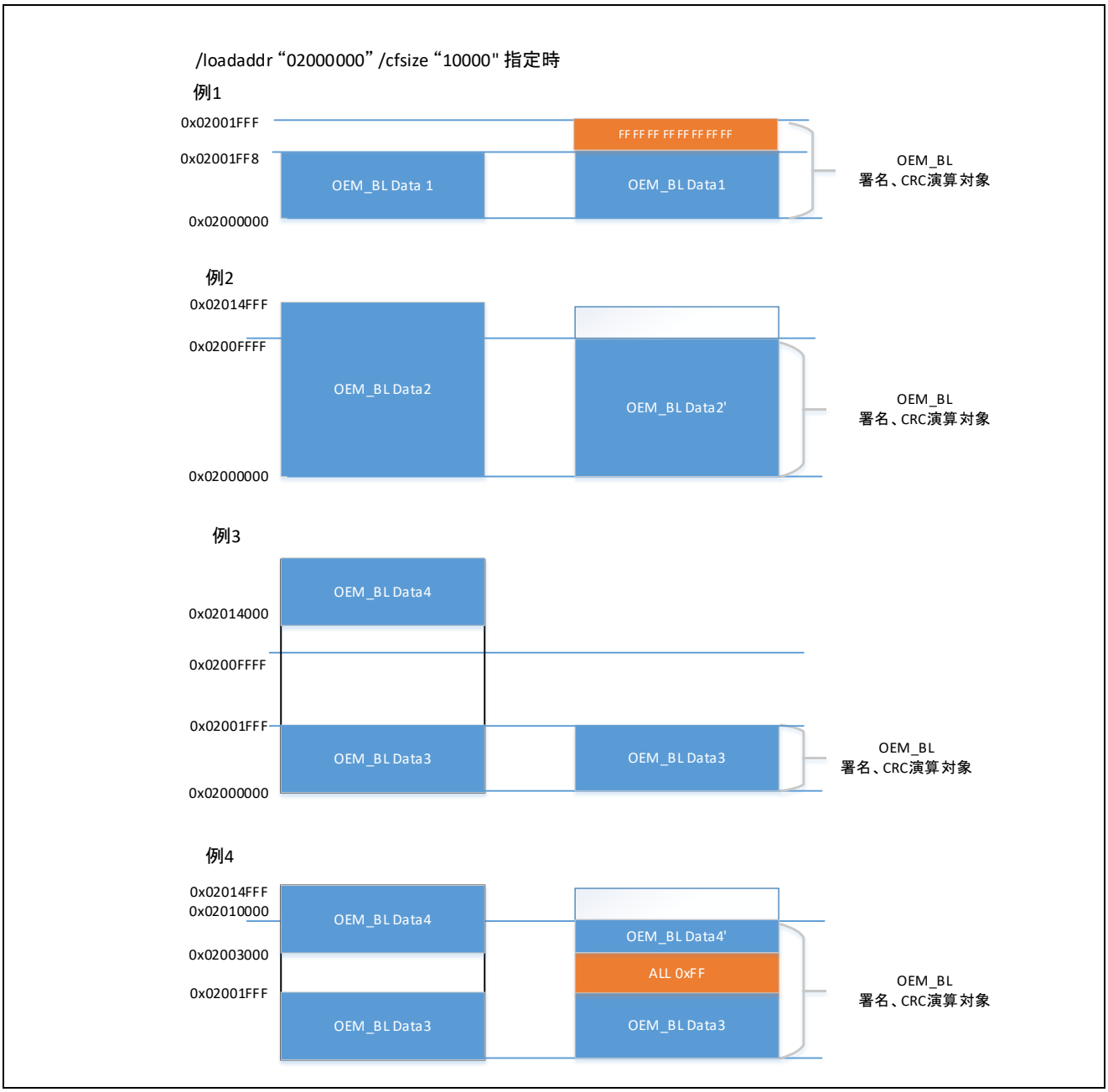

图 4-9 仅指定cfsize选项时的签名、CRC运算对象

4.8 encdotf 命令选项

encdotf命令可使用下列选项。数据输入使用16进制数据或mot文件进行指定。指定文件时，在文件路径的开头加上“file=”。

表 4-88 **encdotf** 选项

选项	参数	说明
keytype	ASCII	指定所用 AES 加密密钥的比特长度。 详情请参考*表 4-89*。
enckey	Hex data / File Path	指定在 **keytype** 选项中指定的位长度的密钥数据或密钥文件。 详情请参考 4.8.2enckey 选项。
nonce	Hex data	指定用于加密的 **nonce** 数据。数据大小为 16 字节。使用输入数据的 100 个高位作为 **nonce**。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注 1）作为 **nonce**。
startaddr	Hex data	指定加密的起始地址。地址大小为 32 位。 该选项为可选项。省略该选项时，则加密整个文件。（*注 2） 请指定 16 字节边界的地址。
endaddr	Hex data	指定加密的结束地址。地址大小为 32 位。 该选项为可选项。省略该选项时，则加密整个文件。（*注 2） 请指定地址，确保从 **startaddr** 选项到 **endaddr** 选项中指定的区域大小为 16 字节。
prg	File Path	指定要加密的文件。
output	File Path	指定输出文件名。
motaddr0	无	指定该选项时，将要输出的 mot 文件的起始地址设为 0。 指定该选项时，无法指定 **incplain** 和 **destaddr** 选项。
incplain	无	指定 **startaddr** 选项及 **endaddr** 选项时，将非加密对象数据附加到输出文件中进行输出。 指定该选项时，无法指定 **motaddr0** 选项。
destaddr	Hex data	指定要加密数据的目标地址。 目标地址用于加密时计数器的 28 个低位。 该选项为可选项。省略该选项时，将要加密范围的起始地址作为目标地址。 指定该选项时，无法指定 **motaddr0** 选项。

注 1： 工具内生成的随机值并不能保证足够的随机性。

注 2： 省略**/staraddr** 和**/endaddr** 选项时，将加密整个文件。没有数据的区域，以及起始地址、结束地址没有进行 16 字节对齐时，将以 0x00 补充数据。

指定加密范围的示例：

```
> skmt.exe /encdotf /keytype "AES-128" /enckey "000102030405060708090A0B0C0D0E0F"
  /nonce "00112233445566778890000000" /startaddr "80000000" /endaddr "80000FFF"
  /prg "D:\work\program.srec" /incplain /output "D:\work\dotf_program.srec"
```

不指定加密范围的示例：

```
> skmt.exe /encdotf /keytype "AES-128" /enckey "000102030405060708090A0B0C0D0E0F"
  /nonce "00112233445566778890000000" /prg "D:\work\program.srec"
  /output "D:\work\dotf_program.srec"
```

4.8.1 keytype 选项

keytype选项以ASCII指定用于加密的密钥长度。

表 4-89 加密所需的密钥长度

ASCII	说明
AES-128	设置密钥长度为 128 位的 AES 密钥。
AES-192	设置密钥长度为 192 位的 AES 密钥。
AES-256	设置密钥长度为 256 位的 AES 密钥。

4.8.2 enckey 选项

enckey选项指定十六进制数据或文件。

按照在 **keytype** 选项中指定的 ASCII 字符，传递以下格式的数据、二进制文件或文本文件。对于数据输入或文本文件输入，则以 16 进制 ASCII 字符串表示 1 个字节。

表 4-90 AES-128

字节	数据
0-15	AES-128 位密钥数据

表 4-91 AES-192

字节	数据
0-23	AES-192 位密钥数据

表 4-92 AES-256

字节	数据
0-31	AES-256 位密钥数据

4.9 encsfp 命令选项

encsfp命令可使用下列选项。数据输入可指定为十六进制/十进制数据、二进制/文本文件或 Renesas Partition Data File。指定文件时，请在文件路径的开头加上“file=”。

并非所有 MCU/MPU 都支持安全工厂编程功能。 请参阅 MCU/MPU 的用户手册和启动固件的应用说明，以了解是否支持该功能或加密范围。

表 4-93 **encsfp** 选项(1)

选项	参数	说明
mcu	ASCII	指定支持安全工厂编程的 MCU 映射信息。 使用 4.9.1mcu 选项中记载的 ASCII 字符进行指定。
enckey	Hex data / File Path	指定要用于用户程序及参数信息加密的 AES128bit 密钥数据。指定 HEX 数据时，请输入 16 字节 HEX 数据。 对于文件输入，可以指定二进制文件(`*.key`)或文本文件(`*.txt`)。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注）作为 **enckey**。
nonce_prm	Hex data	指定用于加密参数信息的 nonce 数据。数据大小为 12 字节。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注）作为 nonce_prm。
nonce_prg	Hex data	指定用于加密用户程序的 nonce 数据。数据大小为 12 字节。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注）作为 **nonce_prg**。
nonce_key	Hex data	当在 **trn** 选项中指定 **DPL_SECDBG_NONSECDBG** 时，将指定用于加密封装的 secdbgkey 和 nonsecdbgkey 的 nonce 数据。 数据大小为 12 字节。该选项可以省略。 如果省略该选项，则使用工具生成的随机值（*注）作为 **nonce_key**。
trn	ASCII / Hex data	指定 DLM 迁移目标信息。 4.9.2 使用 trn 选项中记载的 ASCII 字符进行指定。
prg	File Path	指定要加密的用户程序（mot 文件）。 最多可以指定 3 个用户程序。 详情请参考 4.9.3 prg 选项。
al2key	Hex data / File Path	指定 AL2_KEY。指定 HEX 数据时，请输入 16 字节 HEX 数据。对于文件输入，可以指定二进制文件(`*.key`)或文本文件(`*.txt`)。 在 trn 选项中指定了 **OEM_PL0_AL2** 时，需要输入。 该选项为可选项。在 trn 选项中指定 **OEM_PL0_AL2** 时，如果省略该选项，则使用工具内生成的随机值（*注）作为 al2key。

表 4-94 **encsfp** 选项(2)

选项	参数	说明
secdbgkey	Hex data / File Path	指定 SECDBG_KEY；如果指定 HEX 数据，则输入 16 字节 HEX 数据。对于文件输入，可指定二进制（*.key）或文本文件（*.txt）。 如果使用 **trn** 选项指定了 **DPL_SECDBG_NONSECDBG**，则需要输入。 该选项可以省略；如果在 **trn** 选项中指定了 **DPL_SECDBG_NONSECDBG** 时省略了该选项，则会使用工具生成的随机值（*注）作为 secdbgkey。
nonsecdbgkey	Hex data / File Path	指定 NONSECDBG_KEY；如果指定 HEX 数据，则输入 16 字节 HEX 数据。 对于文件输入，可指定二进制（*.key）或文本文件（*.txt）。 如果使用 **trn** 选项指定了 **DPL_SECDBG_NONSECDBG**，则需要输入。 该选项可以省略；如果在 **trn** 选项中指定了 **DPL_SECDBG_NONSECDBG** 时省略了该选项，则会使用工具生成的随机值（*注）作为 secdbgkey。
ufpk	Hex data / File Path	指定 **enckey** 和 **al2key** 加密时要使用的 UFPK 值，或由 **genufpk** 命令生成的 UFPK 文件。数据大小为 32 字节。
wufpk	File Path	指定由瑞萨密钥封装服务封装了 UFPK 值的 W-UFPK 文件。
iv_enckey	Hex data	指定加密 **enckey** 时要使用的 IV 值。数据大小为 16 字节。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注）作为初始向量 IV。
iv_al2key	Hex data	在 trn 选项中指定”**OEM_PL0_AL2**”时，指定加密 **al2key** 时要使用的 IV 值。数据大小为 16 字节。 该选项为可选项。如果省略该选项，则使用工具内生成的随机值（*注）作为初始向量 IV。
iv_secdbgkey	Hex data	当使用 **trn** 选项指定 **DPL_SECDBG_NONSECDBG** 时，指定封装 secdbgkey 时使用的 IV 值。 数据大小为 16 字节。 此选项可以省略。 如果省略该选项，则使用工具生成的随机值（*注）作为 IV。
iv_nonsecdbgkey	Hex data	当使用 **trn** 选项指定 **DPL_SECDBG_NONSECDBG** 时，指定封装 nonsecdbgkey 时使用的 IV 值。 数据大小为 16 字节。 此选项可以省略。 如果省略该选项，则使用工具生成的随机值（*注）作为 IV。

注：工具内生成的随机值并不能保证足够的随机性。

表 4-95 **encsfp** 选项(3)

选项	参数	说明
boundary	Decimal data / File Path	为代码闪存、数据闪存和 SRAM 指定安全和非安全的可调用边界信息。以及指定五个十进制参数或 Renesas 分区数据文件(`*.rpd`)。 该选项可以省略。 如果省略，则不设置边界信息。 更多信息，请参阅 4.9.4**boundary** 选项。
extarea0	Hex data, Decimal data	指定外部闪存区域的地址和写入单位。 在该选项后，依次输入“起始地址（十六进制）”、“终止地址（十六进制）”和“写入单位（十进制）”。 外部闪存区域和写入单元按以下顺序指定。 以 16 字节为单位指定外部闪存区域和写入单元。 如果加密目标中不包括外部闪存，则不需要此选项。 详见 4.9.5 extarea0/extarea1 选项。
extarea1	Hex data, Decimal data	指定外部闪存区域的地址和写入单位。 在该选项后，依次输入“起始地址（十六进制）”、“终止地址（十六进制）”和“写入单位（十进制）”。 外部闪存区域和写入单元按以下顺序指定。 以 16 字节为单位指定外部闪存区域和写入单元。 如果加密目标中不包括外部闪存，则不需要此选项。 详见 4.9.5 extarea0/extarea1 选项。
output	File Path	指定输出文件名（sfp 文件）。 关于 sfp 文件的格式，请参考附录“安全工厂编程文件格式”。
output_al2key	File Path	在 **trn** 选项中指定 **OEM_PL0_AL2_1** 并省略 **al2key** 选项时，则将用作密钥的在工具内生成的随机值（*注）输出为密钥文件（key 文件）。
output_secdbgkey	File Path	在 **trn** 选项中指定 **DPL_SECDBG_NONSECDBG** 并省略 **secdbgkey** 选项时，则将用作密钥的在工具内生成的随机值（*注）输出为密钥文件（key 文件）。
output _nonsecdbgkey	File Path	在 **trn** 选项中指定 **DPL_SECDBG_NONSECDBG** 并省略 **nonsecdbgkey** 选项时，则将用作密钥的在工具内生成的随机值（*注）输出为密钥文件（key 文件）。
output_enckey	File Path	如果省略 **enckey** 选项，则将用作密钥的在工具内生成的随机值（*注）输出为密钥文件（key 文件）。

注：工具内生成的随机值并不能保证足够的随机性。

mcu 选项 RA8D1_M1_T1 使用 trn 选项指定 OEM_PL0_AL2 的示例：

```
> skmt /encsfp /mcu "RA8D1_M1_T1" /enckey "000102030405060708090a0b0c0d0e0f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /trn "OEM_PL0_AL2" /prg "D:\work\program.mot"
  /al2key "77777777777777778888888888888888"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /iv_enckey "9999999999999999aaaaaaaaaaaaaaaa"
  /iv_al2key "dddddddddddddddddeeeeeeeeeeeeeeee"
  /output " D:\work\ program.sfp"
```

mcu 选项 RA4L1 使用 trn 选项指定 DPL_SECDBG_NONSECDBG 的示例：

```
> skmt /encsfp /mcu "RA4L1" /enckey "000102030405060708090a0b0c0d0e0f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /nonce_key "555555555555666666666666"
  /trn "DPL_SECDEBG_NONSECDBG" /prg "D:\work\program.mot"
  /secdbgkey "77777777777777778888888888888888"
  /nonsecdbgkey "9999999999999999aaaaaaaaaaaaaaaa"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /iv_enckey " dddddddddddddddddeeeeeeeeeeeeeeee"
  /iv_secdbgkey "ffffffffffffffff0000000000000000"
  /iv_nonsecdbgkey "00000000000000001111111111111111"
  /boundary "3" "29" "0" "6" "2"
  /output " D:\work\ program.sfp"
```

mcu 选项 RA8D1_M1_T1 使用 trn 选项指定 LCK_BOOT 的示例：

```
> skmt /encsfp /mcu "RA8D1_M1_T1" /enckey "101112131415161718191a1b1c1d1e1f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /trn "LCK_BOOT" /prg "D:\work\program.mot"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /iv_enckey "9999999999999999aaaaaaaaaaaaaaaa" /output "D:\work\ program.sfp"
```

4.9.1 mcu 选项

mcu选项使用ASCII字符指定MCU信息。

表 4-96 指定 **MCU** 信息

ASCII	说明
RA8D1_M1_T1	指定 RA8D1/M1/T1 的映射信息。
RA8D2_M2_T2_P1	指定 RA8D2/M2/T2/P1 的映射信息。
RA4L1	指定 RA4L1 的映射信息。可指定 **boundary** 选项。

4.9.2 trn 选项

trn选项使用ASCII字符指定DLM迁移信息。可指定的 **trn** 选项取决于 **mcu** 选项中指定的选项。

表 4-97 DLM 迁移和要输入的认证密钥

ASCII	说明
OEM_PL0_AL2_1	将 DLM 转至 OEM，将保护模式转至 PL0，并写入 AL2_KEY 和 AL1_KEY。
OEM_PL0_AL2	将 DLM 转至 OEM，将保护模式转至 PL0，并写入 AL2_KEY。
DPL_SECDBG_NONSECDBG	将 DLM 转移到 DPL，并写入 SECDBG_KEY 和 NONSECDBG_KEY。
LCK_BOOT	将 DLM 传输到 LCK_BOOT。

表 4-98 DLM 迁移和要输入的认证密钥

ASCII	**mcu** 选项		
	RA8D1_M1_T1	**RA8D2_M2_T2_P1**	**RA4L1**
OEM_PL0_AL2_1	-	✓	-
OEM_PL0_AL2	✓	✓	-
DPL_SECDBG_NONSECDBG	-	-	✓
LCK_BOOT	✓	✓	✓

✓：可指定 -：无法指定

4.9.3 prg 选项

prg选项指定要使用安全工厂编程功能加密的用户程序。
prg选项最多可以指定3个。

prg选项 指定3个文件的示例：

```
> skmt /encsfp /mcu "RA8D1_M1_T1" /enckey "101112131415161718191a1b1c1d1e1f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /trn "LCK_BOOT"
  /prg "D:\work\program1.mot" "D:\work\program2.mot" "D:\work\program3.mot"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /iv_enckey "9999999999999999aaaaaaaaaaaaaaaa" /output " D:\work\ program.sfp"
```

4.9.4 boundary 选项

为代码闪存、数据闪存和 SRAM 指定安全和非安全的可调用边界信息。指定方法可以通过参数或文件输入。

4.9.4.1 对于参数输入

按以下从上至下的顺序指定每个区域的大小（KB）。
Code Secure Size, Code Non-secure Callable Size, Data Secure Size, SRAM Secure Size, SRAM Non-secure Callable Size
例如：如果指定 /boundary “128” “64” “0” “16” “8”，则分别使用以下设置。
Code Secure Size : 128KB
Code Non-secure Callable Size : 64KB
Data Secure Size : 0KB
SRAM Secure Size : 16KB
SRAM Non-secure Callable Size: 8KB

4.9.4.2 文件输入

以下参数从 e²studio 生成项目时生成的 Renesas Partition Data File(`*.rpd`) 中读取，并指定为边界信息。
Renesas Partition Data File参数以字节为单位，因此转换为 KB 单位。

表 4-99 Renesas Partition Data File 使用参数

Renesas Partition Data File 使用参数[Byte]	边界信息 [KB]
FLASH_S_SIZE	Code Secure Size
FLASH_C_SIZE	Code Non-secure Callable Size
DATA_FLASH_S_SIZE	Data Secure Size
RAM_S_SIZE	SRAM Secure Size
RAM_C_SIZE	SRAM Non-secure Callable Size

4.9.5 extarea0/extarea1 选项

安全密钥管理工具支持对支持安全工厂编程功能的 MCU/MPU 的启动固件应用说明中定义的外部闪存区域进行 16 字节加密。 该工具支持 16 字节加密。

该选项允许根据要使用的外部闪存的写入单元设置外部闪存区域地址和要使用的闪存写入单元。 使用 extarea0 和 extarea1 选项可以指定两个区域。

指定方法遵循该选项，并按“起始地址（十六进制）”、“终止地址（十六进制）”和“写入单元（十进制）”的顺序输入。 要只指定一个外部闪存区域，请使用 extarea0 选项。

安全密钥管理工具按指定写入划分指定的外部闪存区域，并对写入单元块进行数据加密。 如果写单元块中只有一部分包含数据，则会在没有数据的地址上补上 0xFF 以加密数据。

外部闪存 1 区域设置

0x90000000 - 0x903FFFFF 当指定写入单位为 64 字节时：

```
> skmt /encsfp /mcu " RA8D1_M1_T1" /enckey "10111213141516171819 1a1b1c1d1e1f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /trn "LCK_BOOT"
  /prg "D:\work\program.mot" "D:\work\program_ext.mot"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /extarea0 "90000000" "903FFFFF" "64"
  /iv_enckey "9999999999999999aaaaaaaaaaaaaaaa" /output " D:\work\ program.sfp"
```

外部闪存 2 区域设置

0 区 : 0x80000000 - 0x800FFFFF 当指定写入单位为 128 字节时

1 区 : 0x90000000 - 0x903FFFFF 当指定写入单位为 64 字节时：

```
> skmt /encsfp /mcu " RA8D1_M1_T1" /enckey "10111213141516171819 1a1b1c1d1e1f"
  /nonce_prg "111111111111222222222222" /nonce_prm "333333333333444444444444"
  /trn "LCK_BOOT"
  /prg "D:\work\program.mot" "D:\work\program_ext.mot"
  /ufpk file="ufpk.key" /wufpk "ufpk.key_enc.key"
  /extarea0 "80000000" "800FFFFF" "128"
  /extarea1 "90000000" "903FFFFF" "64"
  /iv_enckey "9999999999999999aaaaaaaaaaaaaaaa" /output " D:\work\ program.sfp"
```

4.10 calcresponse 命令 选项

表 4-100 **calcresponse** 选项

选项	参数	说明
challenge	Hex data	指定 challenge 值。(通常指定器件的唯一 ID) 数据大小为 16 字节。
key	Hex data	指定 DLM 密钥数据。数据大小为 16 字节。
algorithm	Name	指定计算算法。可以选择下列算法。 **HMAC-SHA256** **AES-128-CMAC**

使用**calcresponse**选项时的使用示例：

> skmt.exe /calcresponse /challenge "ABCDEFGHIJKLMNOPQRSTUVWXYZ012345"
/key "000102030405060708090A0B0C0D0E0F" /algorithm HMAC-SHA256

5. 操作程序

5.1 独立

5.1.1 Windows 版本

运行安装程序 SecurityKeyManagementTool_installer_vXXX.exe，并将其安装到任意文件夹中。

注：安装到层级尽可能浅的文件夹中，并且该文件夹应具有写权限，同时文件夹名称中没有“空格”。如果层级太深或文件夹名称太长，则可能无法运行。

5.1.1.1 GUI 版本

安装文件夹中的 SecurityKeyManagementTool.exe 是可执行文件。

图 5-1 安全密钥管理工具 – 从 Windows 启动时的 GUI 对话框

5.1.1.2 CLI 版本

存储在已安装 **CLI** 文件夹中的文件是执行安全密钥管理工具 CLI 版本所需的全套文件。如果要仅使用该工具的 CLI 版本（例如，在生产环境中），则可以将这些文件和**文件夹**（加粗）移动或复制到另一文件夹中以便于操作。下面列出了执行 CLI 版本所需的全套文件。

- skmt.exe
- **device**
- **address**

从命令提示符中输入 `skmt.exe` 命令。

```
C:\work\skmt\tool>skmt.exe /genkey /ufpk file="C:\work\ufpk.key" /wufpk file="C:\work\u
fpk_enc.key" /mcu "RA-SCE9" /keytype "AES-128" /key "000102030405060708090A0B0C0D0E0F"
/filetype "rfp" /output "c:\work\aes128.rkey"
Output File: c:\work\aes128.rkey
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
W-UFPK: 000000004F4C172DEF2789DE4F041294C6B1218D11DC81DC5B37FB72DB899DCFF4BE7158
IV: 46EB124312C83FA226994D17A3F5616A
Encrypted key: DEAE066D5845A01E3FBC0445EC1141F60195D3B3F159D2B624A259BE4965A41D


C:\work\skmt\tool>
```

图 5-2 安全密钥管理工具 - 从命令提示符执行 CLI 示例

5.1.2 Linux 版本

5.1.2.1 GUI 版本

解压缩 Linux 版本包后，将 **SecurityKeyManagementTool** 文件夹放在任意文件夹中。
解压缩时，使用命令 **tar xvzfp xxxxx.tar.gz** 保持执行权限。

图 5-3 安全密钥管理工具 – 从 Linux 启动时的 GUI 对话框

5.1.2.2 CLI 版本

存储在已安装 **CLI** 文件夹中的文件是执行安全密钥管理工具 CLI 版本所需的全套文件。如果要仅使用该工具的 CLI 版本（例如，在生产环境中），则可以将这些文件移动或复制到另一文件夹中以便于操作。下面列出了执行 CLI 版本所需的全套文件。

- skmt
- **device**
- **address**

从终端软件中输入 `skmt` 命令。

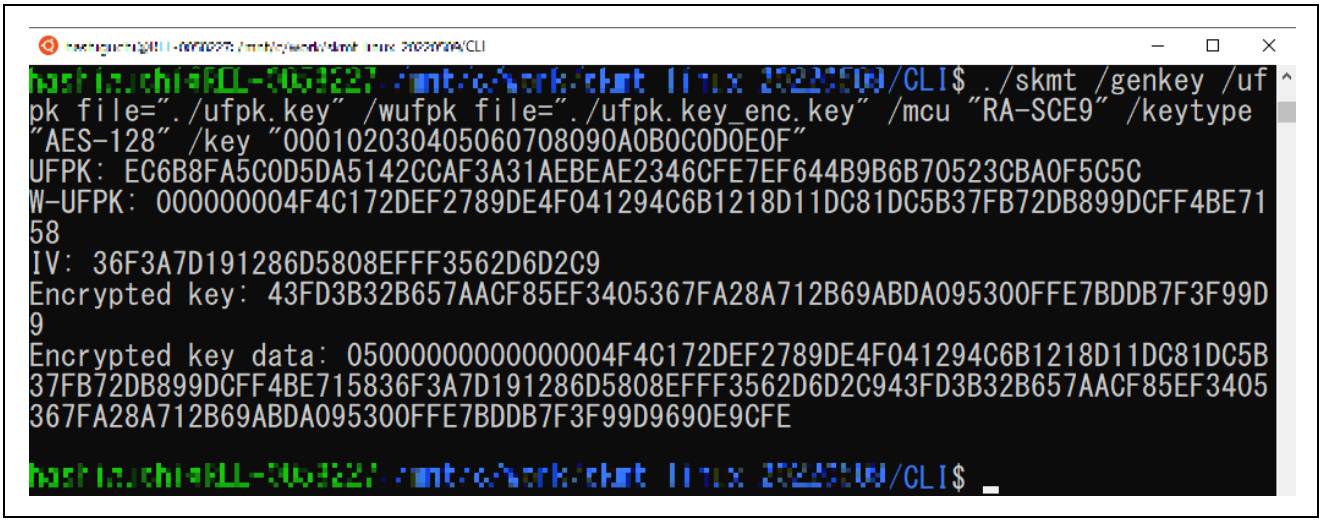

图 5-4 安全密钥管理工具 - 从 Linux 终端软件执行 CLI 示例

5.1.3 macOS 版本

在任意文件夹中解压下载的软件包。
解压缩后的文件结构如下。
SecurityKeyManagementTool.app.zip : GUI 版本
SKMT_CLI.zip : CLI 版本

5.1.3.1 GUI 版本

在任意文件夹中解压 SecurityKeyManagementTool.app.zip 文件，其中 SecurityKeyManagementTool.app 是可执行文件。

图 5-5 安全密钥管理工具 – 从 macOS 启动时的 GUI 对话框

5.1.3.2 CLI 版本

将解压缩文件夹中的 SKMT_CLI.zip 文件解压到所选文件夹。在您选择的文件夹中放入以下文件集。

- skmt
- **device**
- **address**

从终端软件中输入 `skmt` 命令。

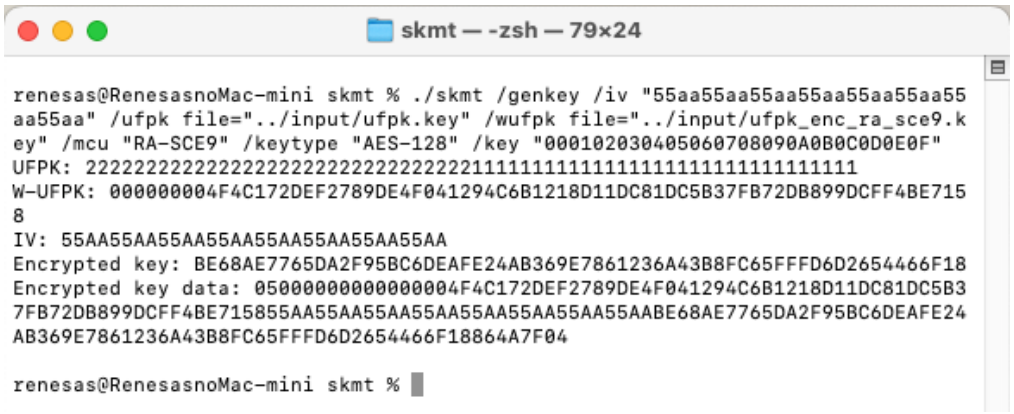

图 5-6 安全密钥管理工具 - 从 macOS 终端软件执行 CLI 示例

5.2 e²studio 插件

Windows，Linux 和 macOS 版本的安装和卸载程序相同。请按照以下步骤安装和卸载。

5.2.1 安装 e²studio 插件版本

1. 从瑞萨网站下载 SecurityKeyManagementTool_Plugin_vXXX_Windows.zip、SecurityKeyManagementTool_Plugin_vXXX_mac.zip 或 SecurityKeyManagementTool_Plugin_vXXX_Linux.zip，并将其放在所选的任意文件夹中。vXXX 为该工具的版本。
2. 启动 e²studio
3. 选择 e²studio 菜单“**帮助(H)**”–“**安装新软件...**”菜单，打开“**安装**”对话框。

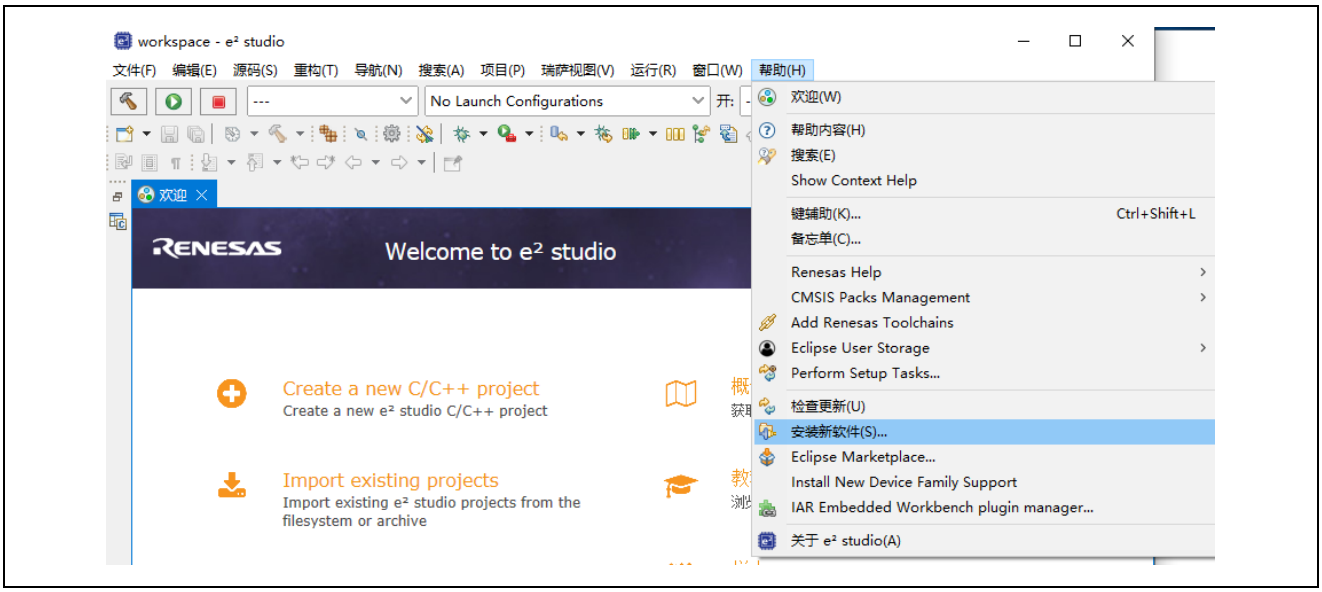

图 5-7 e²studio“帮助(H)”–“安装新软件...”

4. 在“**安装**”对话框中按“**添加**”按钮，打开“**添加资源库**”对话框。

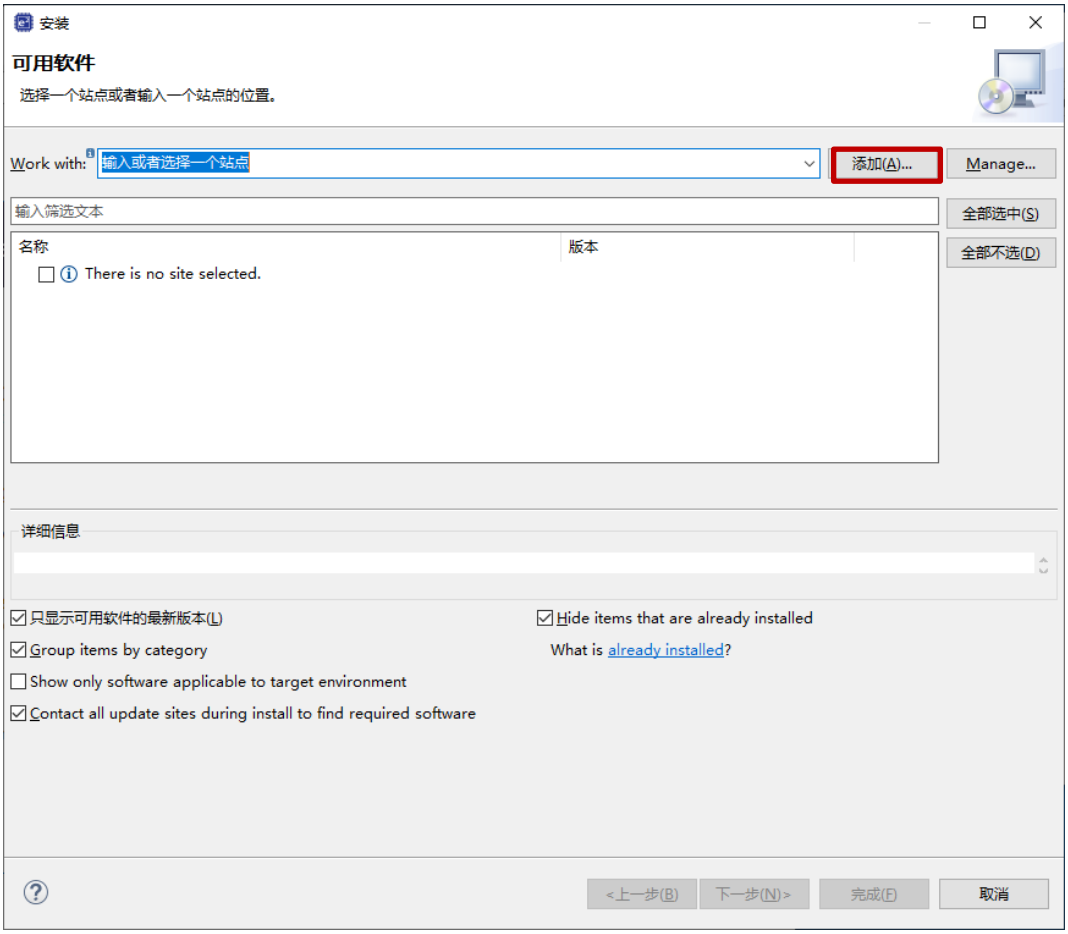

图 5-8“安装”对话框

5. 在“**添加资源库**”对话框中单击“**归档文件(A)…**”，指定在第 1 步中准备的 SecurityKeyManagementTool_Plugin_v_Window/Linux/mac.zip 文件，然后单击“**添加**”按钮。

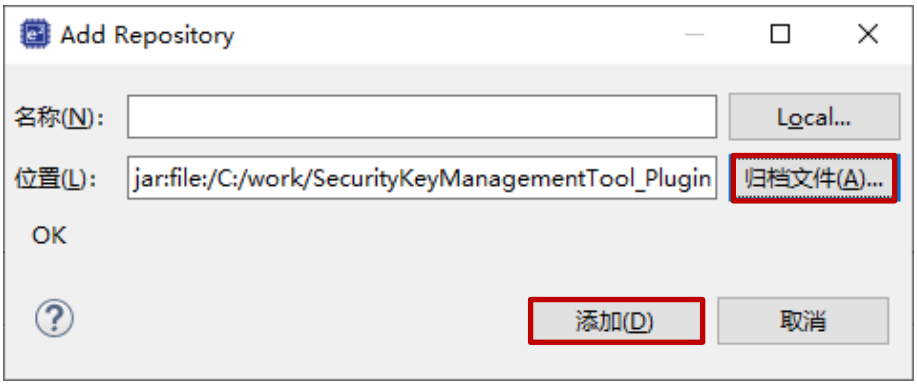

图 5-9“添加资源库”对话框

6. “**Renesas Solution Toolkit**”将添加到“**安装**”对话框中。选中复选框并按“**下一步 >**”按钮。不要勾选“Contact all update sites during install to find required software”。 如果勾选的话，将会增加安装插件所需的时间

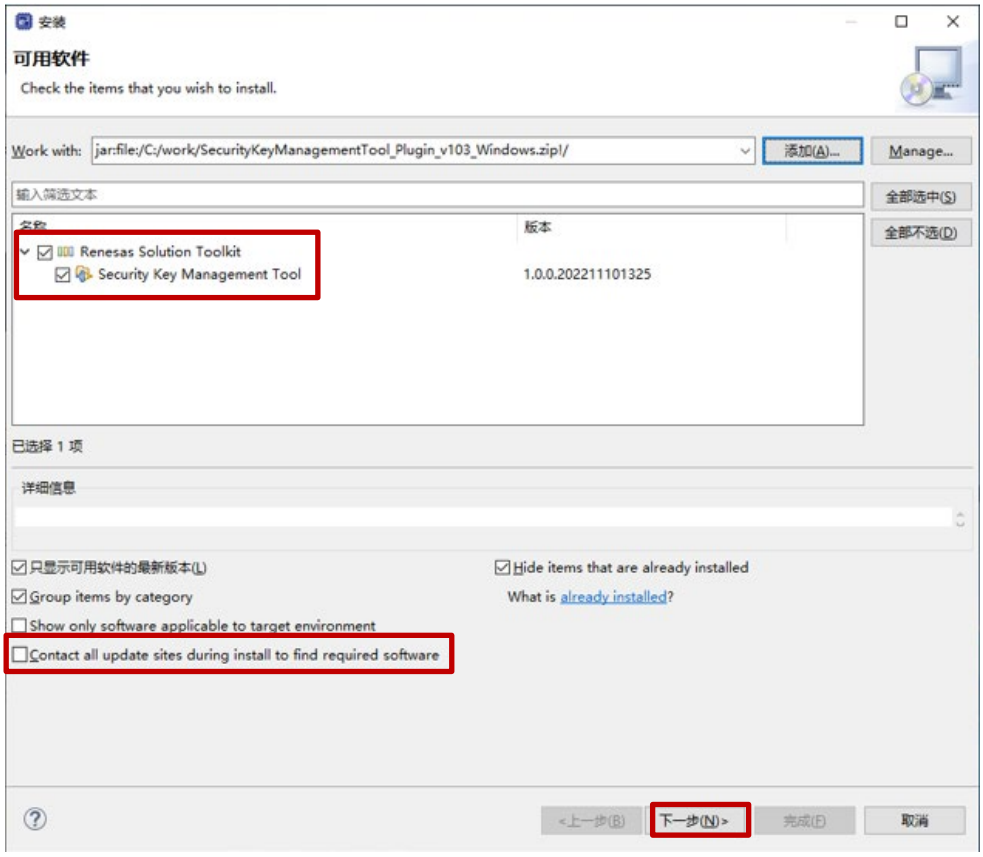

图 5-10“安装”对话框 – 选择“Security Key Management Tool”

7. 当出现“**安装**”对话框时，确认安装“**Security Key Management Tool**”，然后按“下一步”按钮。

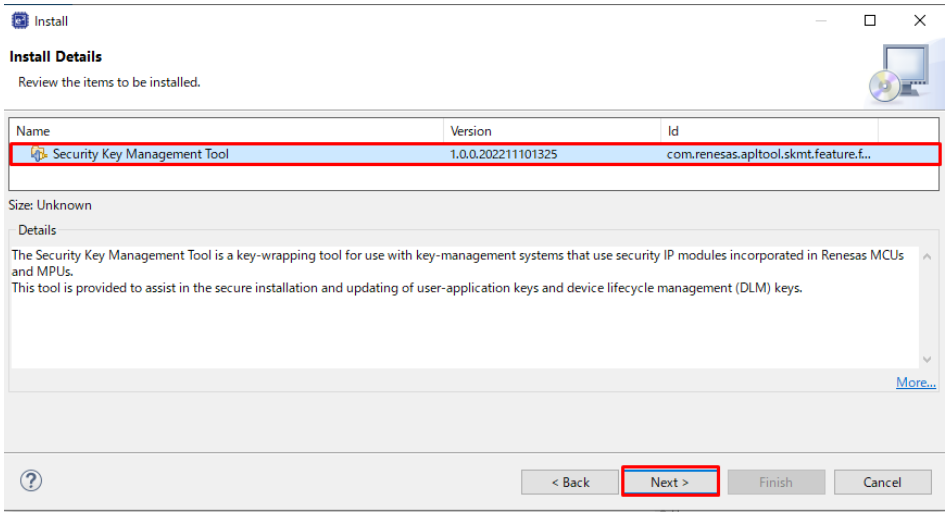

图 5-11“安装”对话框 – 安装详细信息

8. 随后是许可协议确认屏幕。 接受插件下载期间提供的许可条款，可以在对话框中显示的 URL 上查看许可协议的详细内容。选择“I accept the terms of the license agreement”，然后按“完成”按钮。

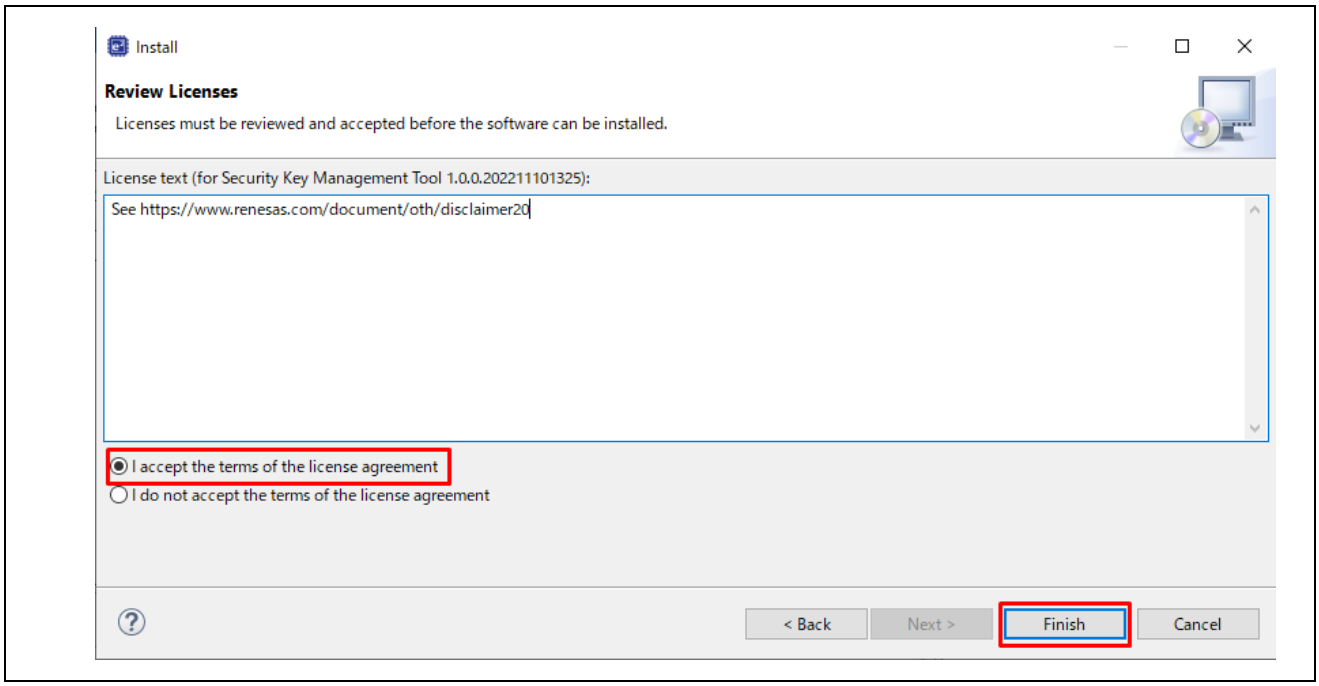

图 5-12“安装”对话框 – 查看许可协议

9. 当出现“**信任**”对话框时，选中显示的证书，然后按“**信任选定项**”按钮继续安装。

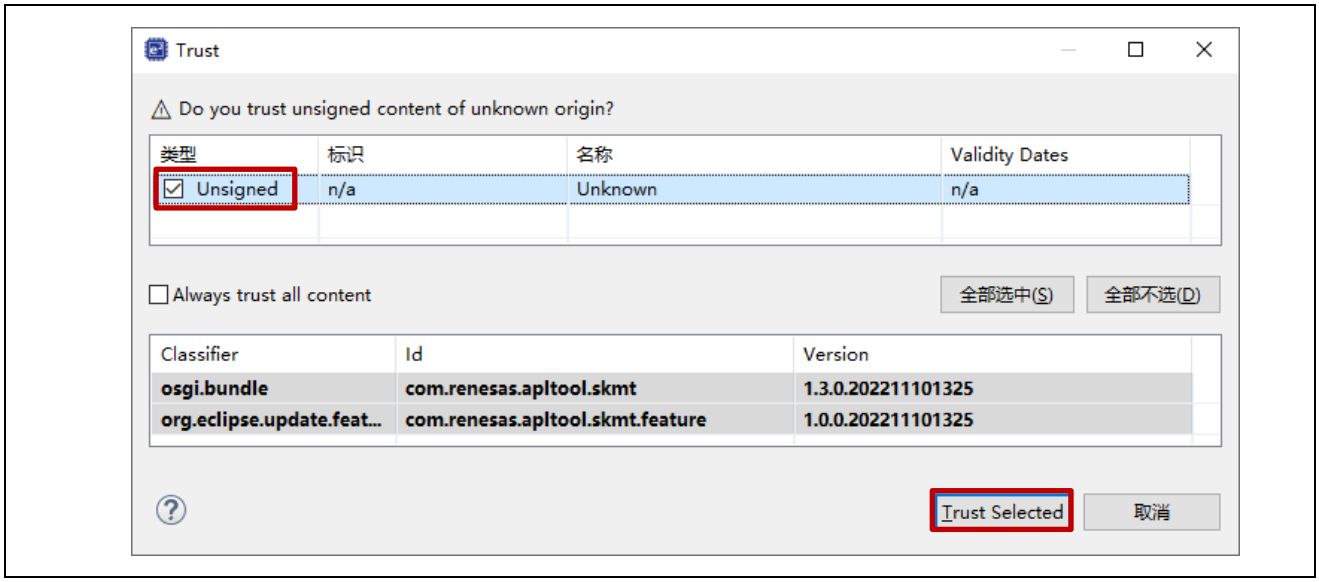

图 5-13 “信任”对话框

10. 当出现提示时，重新启动 e² studio。

11. 安装完成后，安全密钥管理工具将添加到项目的“**属性**”对话框。

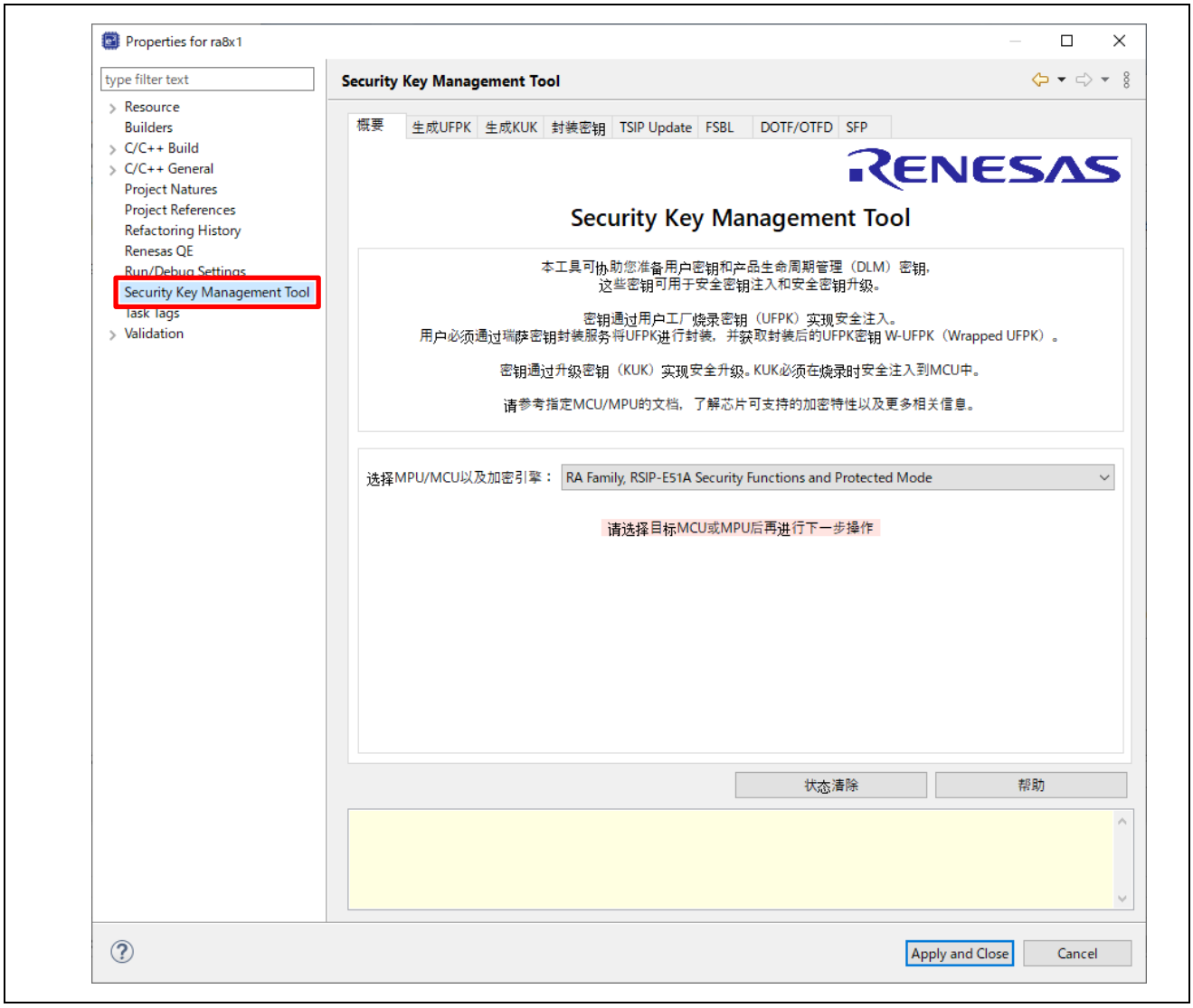

图 5-14 项目“属性”对话框

5.2.2 卸载 e2studio 插件版本

1. 选择 e² studio 菜单“**帮助(H)**”–“**关于 e² studio**”菜单，打开“**关于 e² studio**”对话框。
2. 在“**关于 e² studio**”对话框中按“**安装细节**”按钮。

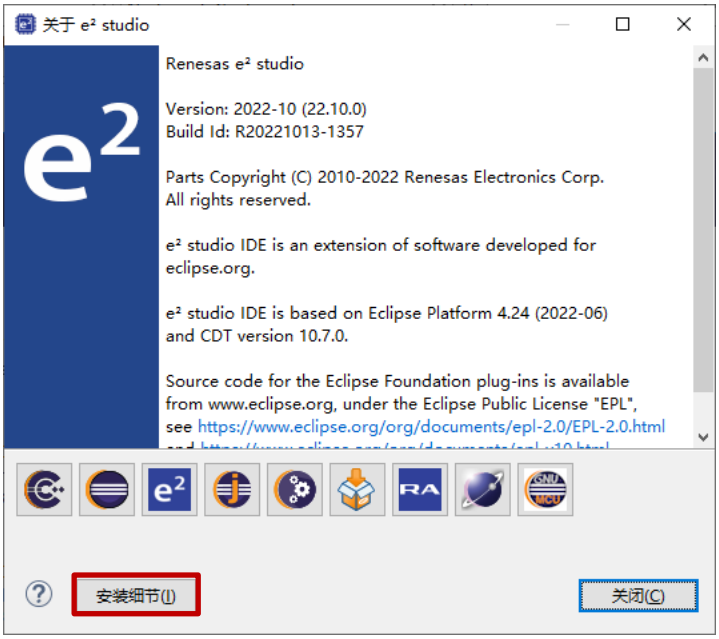

图 5-15“关于 e²studio”对话框

3. 在“**e² studio 安装细节**”对话框中，选择“**已安装的软件**”选项卡 - 安全密钥管理工具，然后按“**卸载...**”按钮。

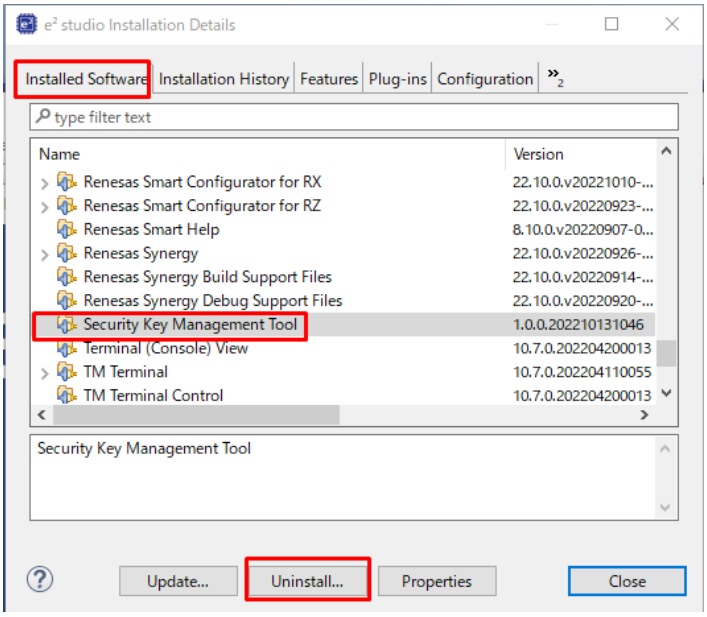

图 5-16“e² studio 安装细节”对话框

4. 当出现“**卸载**”对话框时，选择安全密钥管理工具，然后按“**完成**”按钮。

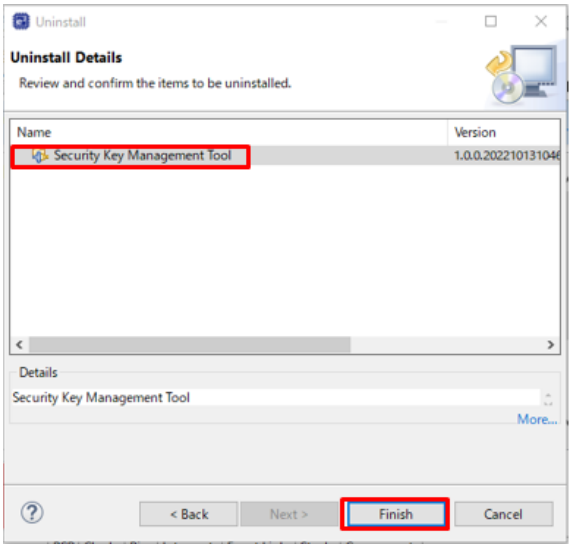

图 5-17“卸载”对话框

5. 系统将提示重新启动 e² studio。

6. 使用示例

6.1 示例 1 – 在具有 TSIP 的 RX 产品家族 MCU 上安装 AES128 密钥

RX 产品家族 TSIP 支持通过在 MCU 上执行固件来实现安全密钥安装。根据*表 1-1 MCU/MPU 相关信息*，可以在 TSIP 库中找到所需的驱动程序。

在生产过程中，经常需要使用相同的密钥对器件组进行编程。密钥安装信息可以嵌入到配置代码中，但是如果存在多个具有不同密钥的组，则最好将密钥信息从配置代码中分离出来。这些步骤可用于创建 Motorola 十六进制文件，该文件可与配置代码一起编程，以便安全安装一个或多个密钥。在以下示例中，创建了一个文件来安装一个 AES128 密钥。

6.1.1 使用 GUI 版本

1. 在 **[概要]** 选项卡上选择 MCU/MPU 以及加密引擎。

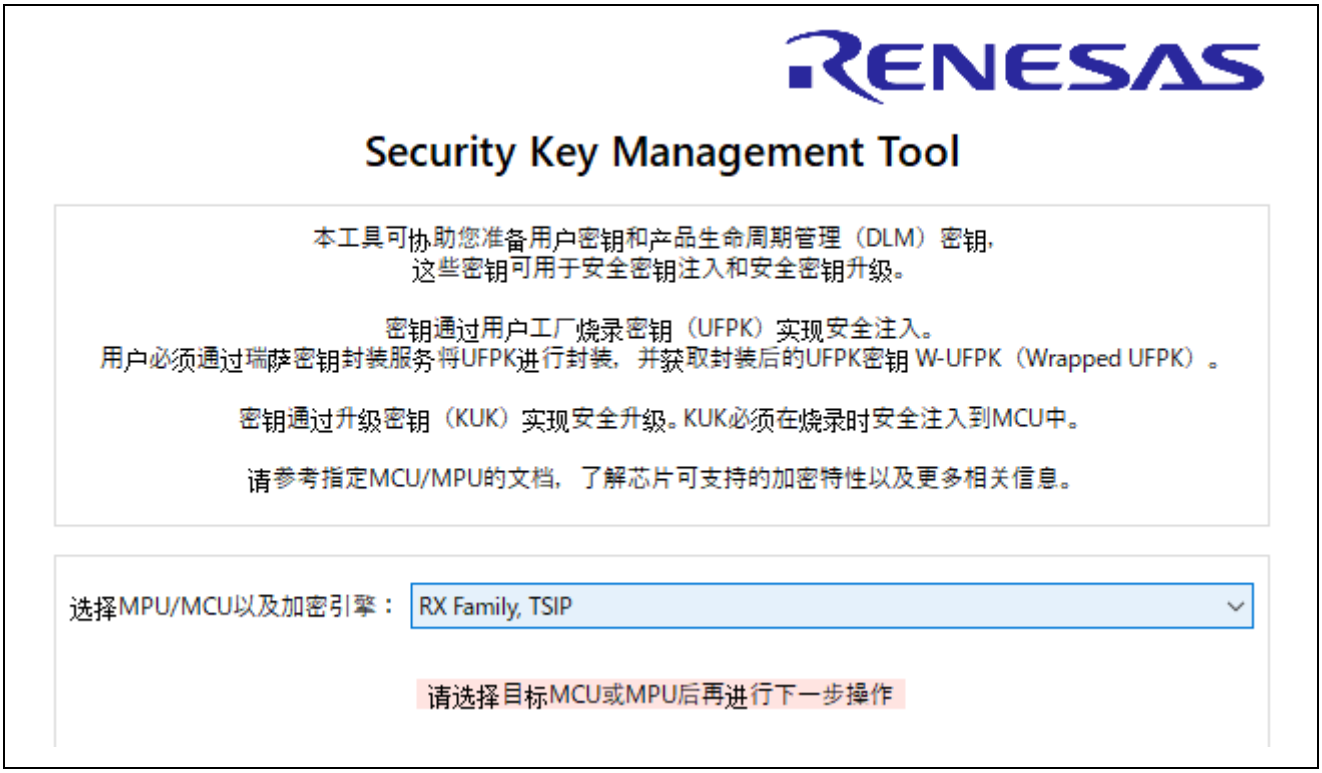

图 6-1 [概要] 选项卡，使用 RX 产品家族 TSIP 安装 AES128 密钥

2. 创建 UFPK。在 **[生成 UFPK]** 选项卡中，输入所需的 UFPK 值，然后输入扩展名为 `*.key` 的输出文件名。在本示例中，使用文件名 `ufpk.key`。

图 6-2 [生成 UFPK] 选项卡，使用指定的值生成 UFPK 的示例

按“**生成 UFPK 密钥文件**”按钮生成 UFPK 文件。如果文件成功生成，该工具将输出以下执行结果。

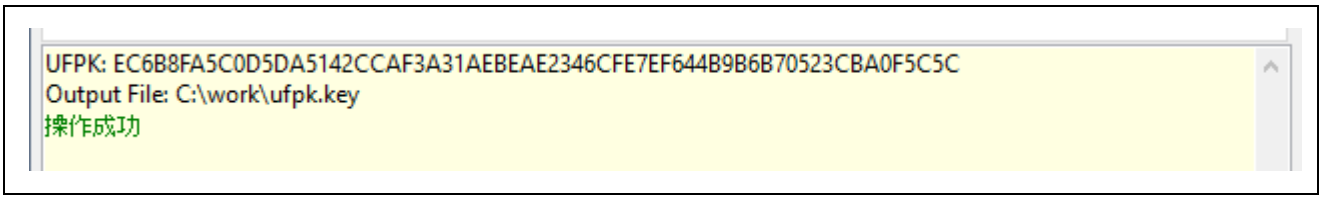

图 6-3 使用指定的值生成 UFPK 的执行结果示例

3. 获取 W-UFPK。将第 2 步中生成的 `ufpk.key` 文件发送到瑞萨密钥封装服务来获取 W-UFPK。有关更多信息，请参见瑞萨密钥封装服务常见问题解答或器件特定的应用笔记。瑞萨密钥封装服务的 URL 为 https://dlm.renesas.com/keywrap。

4. 以 Motorola 十六进制文件格式创建 AES128 密钥文件。在 **[封装密钥]** 选项卡中的 **[密钥类型]** 选项卡下，选择 **AES128**，然后在 **[密钥数据文件]** 选项卡中输入 AES128 数据。对于 **封装密钥**，选择 **UFPK**，然后选择第 2 步中生成的 UFPK 文件和第 3 步中获取的 W-UFPK 文件。为简单起见，在本示例中为 IV 选择了 **生成随机数值**。在 **输出文件** 面板中，为 **格式** 选择 **Motorola 十六进制**，输入扩展名为 `*.mot` 的文件名，然后将 **地址** 字段设置为 FFFF0000。
   指定 **Little** 为内码。

图 6-4 [封装密钥] - [密钥类型] 选项卡，以 Motorola 十六进制格式创建 AES128 密钥文件的示例

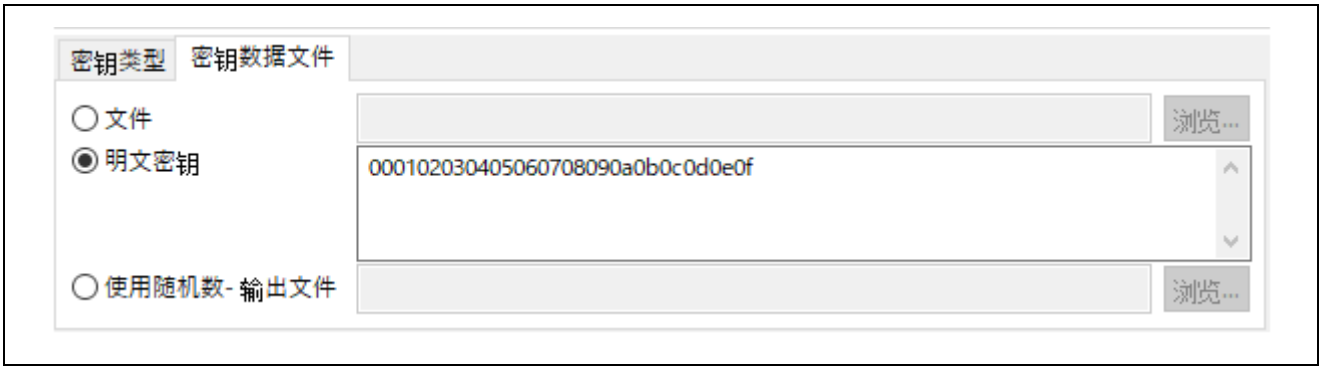

图 6-5 [封装密钥] - [密钥数据文件] 选项卡，以 Motorola 十六进制格式创建 AES128 密钥文件的示例

单击 **生成文件** 按钮生成指定的输出文件。如果文件成功生成，该工具将输出以下执行结果。

```
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
W-UFPK: 00000001102D464621E307E4FF7027D346B9A2DEA8E35A6673DC9685DE0283CBED82DE1E
IV: E2E96270BA2AC0301A70C280A091F425
Encrypted key: 7B095EAD6643CC1AE8BEC0BE8CB7D90F044A44A1F93FEA77756254C76C7900E4
操作成功
```

图 6-6 以 Motorola 十六进制格式创建 AES128 密钥文件的执行结果示例

6.1.2 使用 CLI 版本

1. 使用 **genufpk** 命令创建 UFPK。本示例显示了为 UFPK 使用已知值：

> skmt.exe /genufpk
/ufpk "ec6b8fa5c0d5da5142ccaf3a31aebeae2346cfe7ef644b9b6b70523cba0f5c5c"
/output "C:\work\ufpk.key"

```
C:\work\skmt\tool>skmt.exe /genufpk /ufpk "ec6b8fa5c0d5da5142ccaf3a31aebeae2346cfe7ef64
4b9b6b70523cba0f5c5c" /output "C:\work\ufpk.key"
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
Output File: C:\work\ufpk.key
```

图 6-7 CLI genufpk 命令的执行结果

2. 获取 W-UFPK。将第 1 步中生成的 `ufpk.key` 文件发送到瑞萨密钥封装服务来获取 W-UFPK。有关更多信息，请参见瑞萨密钥封装服务常见问题解答或器件特定的应用笔记。瑞萨密钥封装服务的 URL 为 https://dlm.renesas.com/keywrap。

3. 使用第 1 步中生成的 UFPK 文件和第 2 步中获取的 W-UFPK 文件，通过 **genkey** 命令创建 Motorola 十六进制格式的 AES128 密钥文件：

> skmt.exe /genkey /ufpk file="c:\work\ufpk.key" /wufpk file="C:\work\ufpk.key_enc.key"
/mcu "RX-TSIP" /keytype "AES-128" /key "000102030405060708090A0B0C0D0E0F"
/filetype "mot" /address "FFFF0000" /output "C:\work\aes128.mot"

```
C:\work\skmt\tool>skmt.exe /genkey /ufpk file="c:\work\ufpk.key" /wufpk file="C:\work\u
fpk.key_enc.key" /mcu "RX-TSIP" /keytype "AES-128" /key "000102030405060708090A0B0C0D0E
0F" /filetype "mot" /address "FFFF0000" /output "C:\work\aes128.mot"
Output File: C:\work\aes128.mot
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
W-UFPK: 000000004F4C172DEF2789DE4F041294C6B1218D11DC81DC5B37FB72DB899DCFF4BE7158
IV: 9232A2FD6FC74A7DC0A501397A1A34F3
Encrypted key: 7E66CEB3CA203748A669EDB6B17EF8D231244EA70EF72B839F54CBE41CC1CCC6
```

图 6-8 以 Motorola 十六进制格式创建 AES128 密钥文件的 CLI 示例

6.2 示例 2 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上安装升级密钥

具有 SCE9 保护模式的 RA 产品家族 MCU 支持通过编程接口实现安全密钥安装。瑞萨闪存编程器 (RFP) 支持该功能。

使用编程接口实现安全密钥安装的优势在于，MCU 上无需特殊的配置代码。密钥和应用程序代码可以在同一编程过程中安装。在本示例中，安装一个 KUK，同时在现场启用了密钥升级。

6.2.1 使用 GUI 版本

1. 在 **[概要]** 选项卡上选择 MCU/MPU 以及加密引擎。

图 6-9 [概要] 选项卡，在 RA 产品家族 SCE9 保护模式下安装 KUK

2. 创建 UFPK。在 **[生成 UFPK]** 选项卡中，输入所需的 UFPK 值，然后输入扩展名为 `*.key` 的输出文件名。在本示例中，使用文件名 `ufpk.key`。

图 6-10 [生成 UFPK] 选项卡，使用指定的值生成 UFPK 的示例

按“**生成 UFPK 密钥文件**”按钮生成 UFPK 文件。如果文件成功生成，该工具将输出以下执行结果：

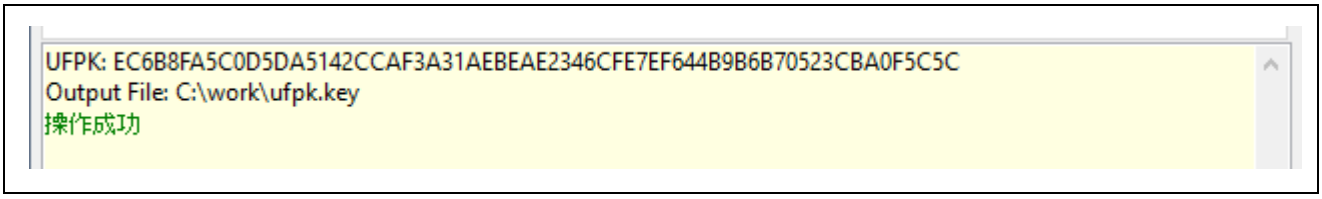

图 6-11 使用指定的值生成 UFPK 的执行结果示例

3. 获取 W-UFPK。将第 2 步中生成的 `ufpk.key` 文件发送到瑞萨密钥封装服务来获取 W-UFPK。有关更多信息，请参见瑞萨密钥封装服务常见问题解答或器件特定的应用笔记。瑞萨密钥封装服务的 URL 为 https://dlm.renesas.com/keywrap。

4. 创建 KUK 密钥文件。在 **[生成 KUK]** 选项卡中，选择是使用工具为 KUK 生成随机数值还是为 KUK 输入 256 位值。输入扩展名为 `*.key` 的输出文件名。在本示例中，使用文件名 `kuk.key`。

图 6-12 [生成 KUK] 选项卡，创建 KUK 密钥文件的示例

单击“**生成 KUK 密钥文件**”按钮生成 KUK 密钥文件。如果文件成功生成，该工具将输出以下执行结果。

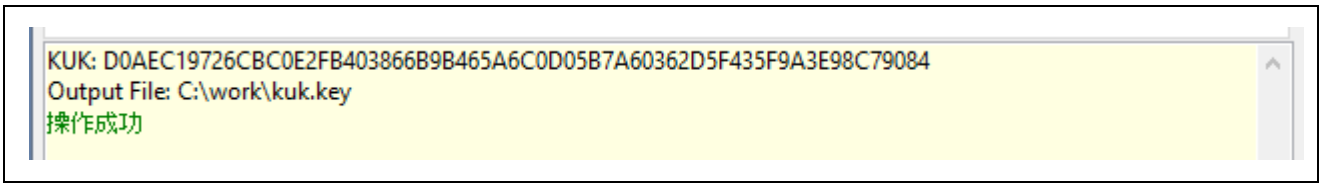

图 6-13 创建 KUK 文件的执行结果示例

5. 为 KUK 安装创建 RFP 文件。在 **[封装密钥]** 选项卡中的 **[密钥类型]** 选项卡中，选择 **KUK**。在 **[密钥数据文件]** 选项卡中，输入上一步中生成的 KUK 密钥文件名（本例中为 `kuk.key`）。对于 **封装密钥**，选择 **UFPK** ，然后选择第 2 步中生成的 UFPK 文件和第 3 步中获取的 W-UFPK 文件。为简单起见，在本示例中为 IV 选择了 **生成随机数值**。在 **输出文件** 面板中，为 **格式** 选择 **RFP**，然后输入扩展名为 `*.rkey` 的文件名。

图 **6-14 [封装密钥] - [密钥类型]** 选项卡，以 **RFP** 文件格式创建 **KUK** 文件的示例

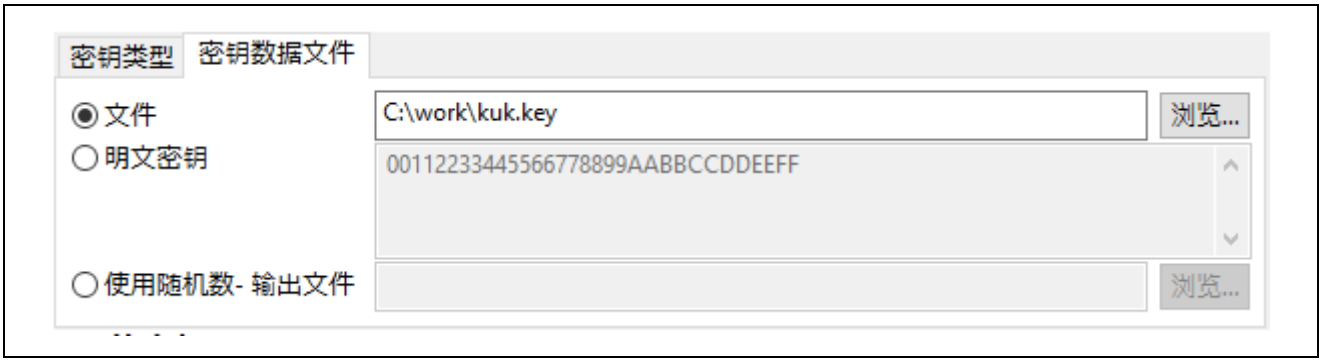

图 **6-15 [封装密钥] – [密钥数据文件]** 选项卡，以 **RFP** 文件格式创建 **KUK** 文件的示例

单击 **生成文件** 按钮生成指定的输出文件。如果文件成功生成，该工具将输出以下执行结果。

```
IV: BF1C8628E8DF1C208485C3ABB4AE5E94
Encrypted key:
0C43895701106C1D4658F181585B0F2E9A1BA427EA4243DE9F183377E7329A848BC0A6C70370C4CFE2A64E828432EA5
9
操作成功
```

图 6-16 以 RFP 文件格式创建 KUK 文件的执行结果示例

6.2.2 使用 CLI 版本

1. 使用 **genufpk** 命令创建 UFPK。本示例显示了为 UFPK 使用指定的值：

> skmt.exe /genufpk
/ufpk "ec6b8fa5c0d5da5142ccaf3a31aebeae2346cfe7ef644b9b6b70523cba0f5c5c"
/output "C:\work\ufpk.key"

```
C:\work\skmt\tool>skmt.exe /genufpk /ufpk "ec6b8fa5c0d5da5142ccaf3a31aebeae2346cfe7ef64
4b9b6b70523cba0f5c5c" /output "C:\work\ufpk.key"
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
Output File: C:\work\ufpk.key
```

图 6-17 CLI genufpk 命令的执行结果

2. 获取 W-UFPK。将第 1 步中生成的 `ufpk.key` 文件发送到瑞萨密钥封装服务来获取 W-UFPK。有关更多信息，请参见瑞萨密钥封装服务常见问题解答或器件特定的应用笔记。瑞萨密钥封装服务的 URL 为 https://dlm.renesas.com/keywrap。

3. 使用 **genkuk** 命令创建 KUK 密钥文件。本示例显示了为 KUK 使用指定的值。

> skmt.exe /genkuk /output "C:\work\kuk.key"
/kuk "d0aec19726cbc0e2fb403866b9b465a6c0d05b7a60362d5f435f9a3e98c79084"

```
C:\work\skmt\tool>skmt.exe /genkuk /output "C:\work\kuk.key" /kuk "d0aec19726cbc0e2fb40
3866b9b465a6c0d05b7a60362d5f435f9a3e98c79084"
KUK: D0AEC19726CBC0E2FB403866B9B465A6C0D05B7A60362D5F435F9A3E98C79084
Output File: C:\work\kuk.key
```

图 6-18 CLI genkuk 命令的执行结果

4. 使用第 1 步中生成的 UFPK 文件、第 2 步中获取的 W-UFPK 文件和第 3 步中生成的 KUK 文件，通过 **genkey** 命令创建要用于 KUK 安装的 RFP 文件：

> skmt.exe /genkey /ufpk file="c:\work\ufpk.key" /wufpk file="C:\work\ufpk.key_enc.key" /mcu "RA-SCE9" /keytype "key-update-key" /key file="C:\work\kuk.key" /filetype "rfp" /output "C:\work\kuk.rkey"

```
C:\work\skmt\tool>skmt.exe /genkey /ufpk file="c:\work\ufpk.key" /wufpk file="C:\work\u
fpk.key_enc.key" /mcu "RA-SCE9" /keytype "key-update-key" /key file="C:\work\kuk.key" /
filetype "rfp" /output "C:\work\kuk.rkey"
Output File: C:\work\kuk.rkey
UFPK: EC6B8FA5C0D5DA5142CCAF3A31AEBEAE2346CFE7EF644B9B6B70523CBA0F5C5C
W-UFPK: 000000004F4C172DEF2789DE4F041294C6B1218D11DC81DC5B37FB72DB899DCFF4BE7158
IV: 4B44A010D7BDF764200BC2424650F976
Encrypted key: 043565AE00E185EBAFF505290DF1976B1A00132FE1F0B25F32C3C83BF92C83A7644968FF
5869821C3EBAD7EB41AA509B
```

图 6-19 以 RFP 文件格式创建 AES128 密钥文件的 CLI 示例

6.3 示例 3 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上升级 RSA 2048 公钥

可以使用之前安装的 KUK 在现场升级密钥。根据*表 1-1 MCU/MPU 相关信息*，可以在器件的支持软件驱动程序中找到所需的驱动程序。

可以通过多种方式在现场执行密钥升级。一种方式是将使用 KUK 封装的密钥作为固件升级的一部分包括在内。有一种简单的方式是将数据作为 C 源文件嵌入。在本示例中，使用之前安装的 KUK（例如，在前一个示例中创建和安装的 KUK）升级一个 RSA 2048 公钥。

6.3.1 使用 GUI 版本

1. 在 **[概要]** 选项卡上选择 MCU/MPU 以及加密引擎。

图 6-20 [概要] 选项卡，在具有 SCE9 保护模式的 RA 产品家族上升级 RSA 2048 公钥

2. 以 C 源文件格式创建 RSA 2048 公钥文件。在 **[封装密钥]** 选项卡中的 **[密钥类型]** 选项卡下，选择 **RSA** 和 **2048 bits, public**，然后在 **[密钥数据文件]** 选项卡中输入 RSA 2048 公钥数据。对于 **封装密钥**，选择 **KUK**，然后将 **KUK 文件** 设置为第 6.2 节示例 2 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上安装升级密钥中生成的文件。为简单起见，在本示例中为 IV 选择了 **生成随机数值**。在 **输出文件** 面板中，为 **格式** 选择 **C 源文件**，然后输入扩展名为 `*.c` 的文件名。

图 **6-21 [封装密钥] - [密钥类型]** 选项卡，以 **C** 源文件格式创建 **RSA 2048** 公钥文件的示例

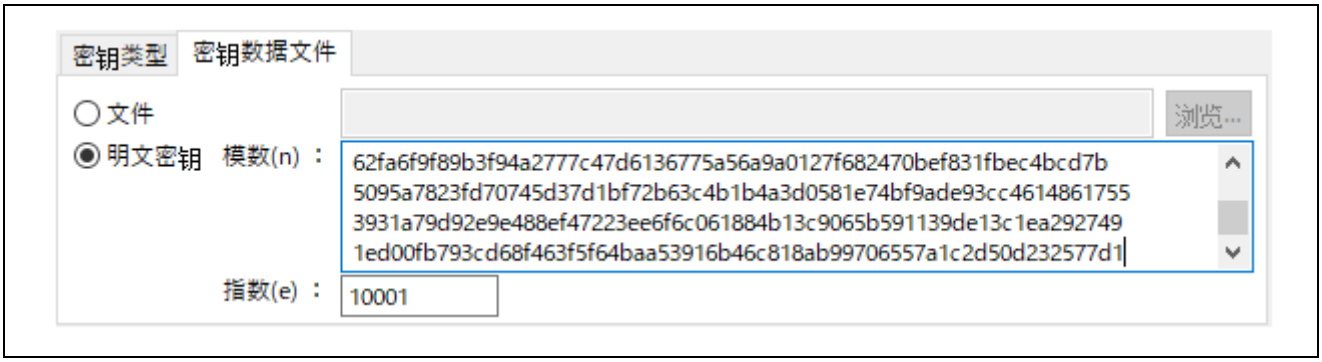

图 **6-22 [封装密钥] – [密钥数据文件]** 选项卡，以 **C** 源文件格式创建 **RSA 2048** 公钥文件的示例

单击 **生成文件** 按钮生成指定的输出文件。如果文件成功生成，该工具将输出以下执行结果。

```
Output File: C:\work\rsa2048public.h
Output File: C:\work\rsa2048public.c
KUK: D0AEC19726CBC0E2FB403866B9B465A6C0D05B7A60362D5F435F9A3E98C79084
IV: 246002BCFE6C3B02533B9D4A9DB0C66C
Encrypted key:
46C95D8AD78C0C91168E9E3B19887ED954CCCF4EDDFA52273D0413ACA9BA69989E98BDA3CDF689863EB3FC599730A63
BC938CA3E5FD488088E49168A43F1D44B8039244A4C342D28E18C68846BB78E4DAA826B4BA4F37FF03FD526EB3CA6BFF
4534ECC8FDAB8F500127BB23BD727DFCF7933CD80E6E42620AE014692CFEBB188F734445DEADCB847EBF9632331F7ECF7
9A9A519AB47B0B2CE5C1C65854C346F17AB7744356FA6EECD66C90F2644185A913E01196267D56517E087A7131C8298AA
C4F573A30E68A3E74F6B8BD62A070D64CBC6EC59E9B14F855E4207C49C26727FE933B11CA8B82D4A2E3D143BF1754CD3
DE457B8AC9593D5C213C13541F9E71F0DE07C43DA1D634EED6773D4601F5B9F3DA0730F2DF42013050CB897EC9F131E
操作成功
```

图 6-23 以 C 源文件格式创建 RSA 2048 公钥文件的执行结果示例

6.3.2 使用 CLI 版本

1. 使用 **genkey** 命令以 C 源文件格式创建 RSA 2048 公钥文件：

```
> skmt.exe /genkey /kuk file="C:\work\kuk.key" /mcu "RA-SCE9" /keytype "RSA-2048-public"
        /key "bad47a84c1782e4dbdd913f2a261fc8b65838412c6e45a2068ed6d7f16e9cdf4
            462b39119563cafb74b9cbf25cfd544bdae23bff0ebe7f6441042b7e109b9a8a
            faa056821ef8efaab219d21d6763484785622d918d395a2a31f2ece8385a8131
            e5ff143314a82e21afd713bae817cc0ee3514d4839007ccb55d68409c97a18ab
            62fa6f9f89b3f94a2777c47d6136775a56a9a0127f682470bef831fbec4bcd7b
            5095a7823fd70745d37d1bf72b63c4b1b4a3d0581e74bf9ade93cc4614861755
            3931a79d92e9e488ef47223ee6f6c061884b13c9065b591139de13c1ea292749
            1ed00fb793cd68f463f5f64baa53916b46c818ab99706557a1c2d50d232577d1
            00010001"
        /filetype "csource" /output "C:\work\rsa2048public.c" /keyname “rsa2048public”
```

使用第 6.2 节示例 2 – 在具有 SCE9 保护模式的 RA 产品家族 MCU 上安装升级密钥中生成的 KUK 文件。

```
C:\work\skmt\tool>skmt.exe /genkey /kuk file="C:\work\kuk.key" /mcu "RA-SCE9" /keytype
"RSA-2048-public" /key "bad47a84c1782e4dbdd913f2a261fc8b65838412c6e45a2068ed6d7f16e9cdf
4462b39119563cafb74b9cbf25cfd544bdae23bff0ebe7f6441042b7e109b9a8afaa056821ef8efaab219d2
1d6763484785622d918d395a2a31f2ece8385a8131e5ff143314a82e21afd713bae817cc0ee3514d4839007
ccb55d68409c97a18ab62fa6f9f89b3f94a2777c47d6136775a56a9a0127f682470bef831fbec4bcd7b5095
a7823fd70745d37d1bf72b63c4b1b4a3d0581e74bf9ade93cc46148617553931a79d92e9e488ef47223ee6f
6c061884b13c9065b591139de13c1ea2927491ed00fb793cd68f463f5f64baa53916b46c818ab99706557a1
c2d50d232577d100010001" /filetype "csource" /output "C:\work\rsa2048public.c" /keyname
"rsa2048public"
Output File: C:\work\rsa2048public.h
Output File: C:\work\rsa2048public.c
KUK: D0AEC19726CBC0E2FB403866B9B465A6C0D05B7A60362D5F435F9A3E98C79084
IV: BED48489C13FF9173A6F7649DEB6EB47
Encrypted key: FBB110AAAA2260E6033B2D9839A71121C137ABE34D1FE59D50C7DB2F0B568AA2D5C0A0CF
92CA7D093A624E931BFC6115A09DBED02CA3E85909940D2FE2921C5589D4D858A349F54FFFEBF8A18D94139
909BF57A0DEB219A99C640993990B3B837132F404CCCCB821AF1D28C8895C968863E293E22A87365BEC0748
1B856275BEC8E44EC9BFA392F586CEBB674C73230834C0B7989011E8DCEA6C71DE314D381788A129A42C541
27CA2FBD315A3F8BA9960B25C7B6A999D915443358C755A52D77608CFE4A48B9EEB46CFDA0AB96CEC209EB2
01F2EA2C2382564C30A3622E57071247B2E386160C5D21A00119ED4DD118B07554E72BB844FFA2AE9EF7C3D
28D9D0C116490388A554EC39D7D515C89C8459A42FD4495B6DF14ADB69D082D356DED
```

图 6-24 以 C 源文件格式创建 RSA 2048 公钥文件的 CLI 示例

6.4 示例 4 – RX 产品家族 TSIP Secure Update 时的使用方法

在使用 TSIP 的安全固件更新解决方案中，可以加密用户程序后发送至目标器件，然后在目标器件内解密用户程序并更新。必要的驱动程序及使用样本进行更新的方法，请参考*表 1-1 MCU/MPU 相关信息*中 RX TSIP 的资料。

6.4.1 使用 GUI 版本的 Security Key Management Tool 时

1. 选择 MCU/MPU 以及加密引擎
   在[概要]选项卡中选择 MCU/MPU 以及加密引擎。

图 6-25 [概要]选项卡

2. 生成固件映像文件

使用**[TSIP Update]**选项卡，生成附加 RSU 标头的固件映像。

在**[RSU 标头]**选项卡中，**映像标志** 设置为 TESTING，**RSU 标头 Ver** 设置为 **1**。

在 **输出映像** 中选择 **Secure Update**，在**“固件映像”**中设置加密固件的 mot 文件。

在**[加密地址范围]**选项卡中，设置要加密的地址范围以及是否添加数据闪存数据。 在本例中，**Data Flash** 选择 **Do not add**。

在**[Image Encryption Key]**选项卡的 **Session Key Encryption Key** 中，设置对 Secure Boot 程序中设置的加密密钥进行加密的密钥。为了便于理解，在本例中，**Image Encryption Session Key** 选择 **生成随机数值**。

在**[IV]**选项卡中，为了便于理解，选择 **生成随机数值**。

输出 的格式选择 **二进制**，指定扩展名为*.rsu 的文件名。

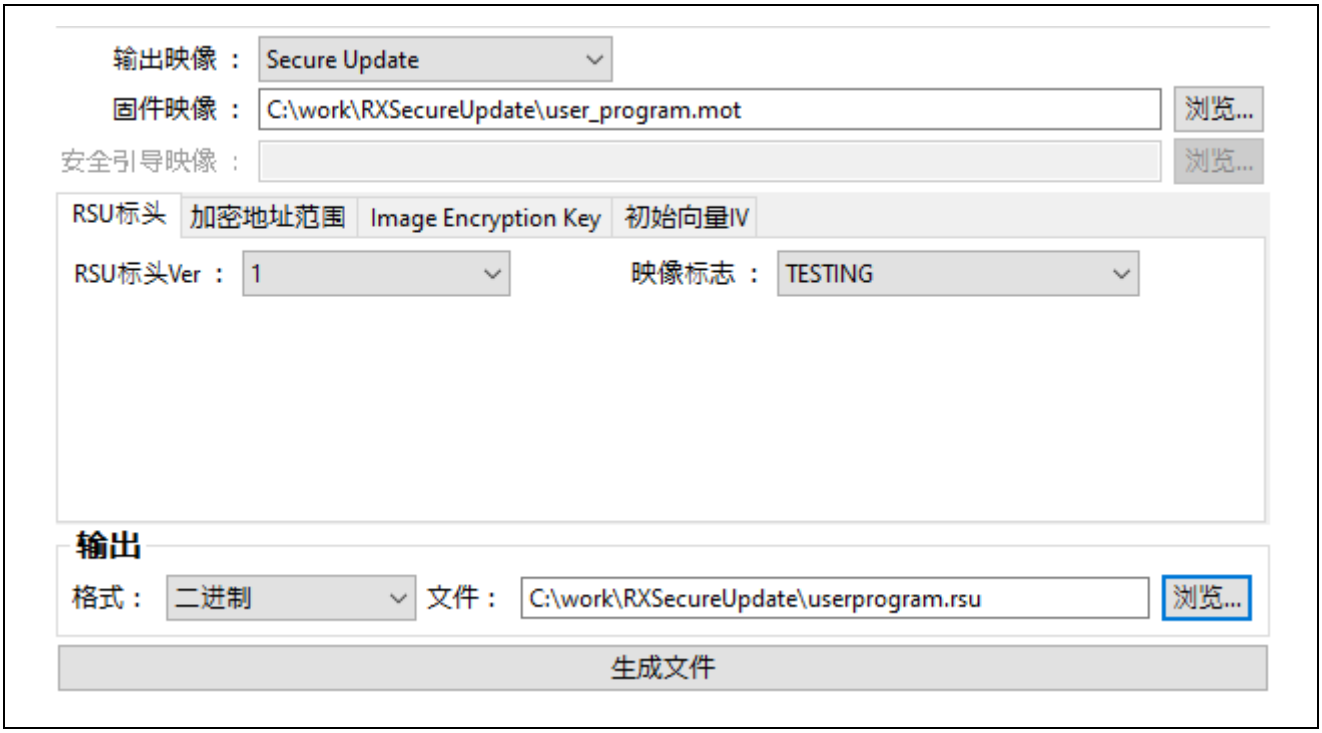

图 **6-26 [TSIP Update] – [RSU标头]选项卡设置示例**

图 **6-27 [TSIP Update] – [加密地址范围]选项卡等的设置示例**

图 6-28 [TSIP Update] – [Image Encryption Key]选项卡设置示例

图 6-29 [TSIP Update] – [IV]选项卡设置示例

如果正常结束，则输出如下。

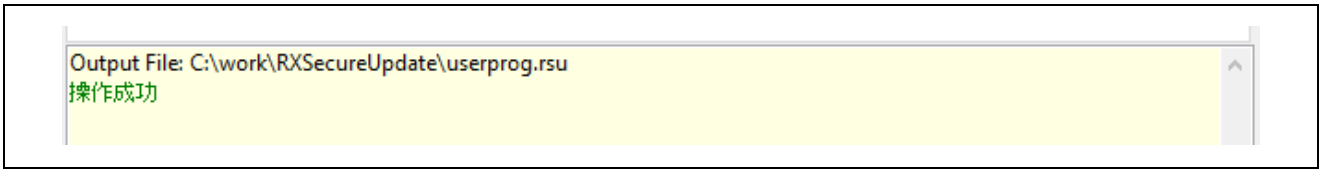

图 6-30 [TSIP Update]选项卡 执行结果

6.4.2 使用 CLI 版本的 Security Key Management Tool 时

1. 在终端软件中执行 **enctsip** 命令。

> skmt.exe /enctsip /mode "update" /ver "1" /prg "C:\work\RXSecureUpdate\user_program.mot"
/enckey "0123456789abcdef0123456789abcdef"
/startaddr "FFE00300" /endaddr "FFFEFFFF" /imgflg "testing" /filetype "bin"
/output "C:\work\RXSecureUpdate\userprog.rsu"

```
C:¥work¥skmt¥tool>skmt.exe /enctsip /mode "update" /ver "1" /prg "C:¥wo
rk¥RXSecureUpdate¥user_program.mot" /enckey "0123456789abcdef0123456789
abcdef" /startaddr "FFE00300" /endaddr "FFFEFFFF" /imgflg "testing" /fi
letype "bin" /output "C:¥work¥RXSecureUpdate¥userprog.rsu"
Output File: C:¥work¥RXSecureUpdate¥userprog.rsu
```

图 **6-31 enctsip**命令执行示例

6.5 示例 5 - RA 产品家族 FSBL 密钥证书/代码证书生成时的使用方法

生成可用于配备第一阶段引导加载程序(FSBL)功能的瑞萨电子器件的密钥证书和代码证书。密钥证书和代码证书用于在使用瑞萨闪存编程器将 OEM 引导加载程序烧录到器件时验证 OEM 引导加载程序的合法性。

必要的工具及示例的使用方法，请参阅*表 1-1* MCU/MPU 相关信息中的资料。

6.5.1 使用 GUI 版本的 Security Key Management Tool 时

1. 选择 MCU/MPU 以及加密引擎
   在[概要]选项卡中选择 MCU/MPU 以及加密引擎。

图 6-32 [概要]选项卡

2. 密钥证书和代码证书的生成

使用**[FSBL]**选项卡生成密钥证书和代码证书。

在本例中，介绍编程验证方法为“签名”时的步骤。

在 **OEM 引导加载程序映像** 中指定签名对象 OEM 引导加载程序的 mot 文件。

编程验证方法 选择 **签名**。

将 **映像版本** 设置为 1。

进行**[证书]**选项卡的设置。

在 **代码闪存起始地址 (hex)** 中设置 OEM 引导加载程序的起始地址。在本例中，设为 02000000。

在 **器件代码闪存大小**中选择要使用器件的代码闪存大小。在本例中，选择 **2MB**。

在 **OEM 引导加载程序大小（16 字节对齐）**中选择 **自动计算**。

在 **密钥证书** 中指定密钥证书的文件名。

在 **代码证书** 中指定代码证书的文件名。

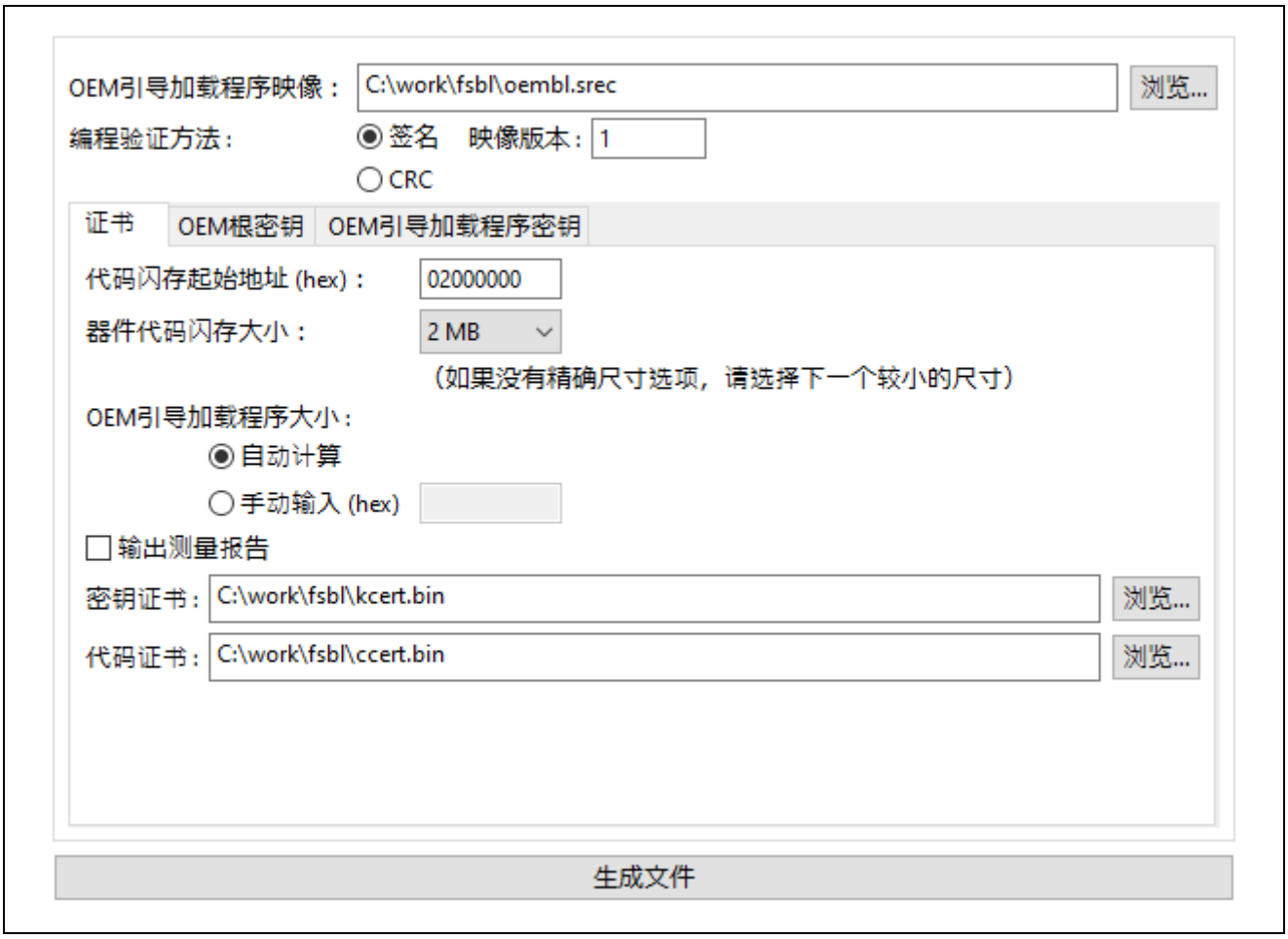

图 6-33 [FSBL] – [证书]选项卡等的设置示例

进行[**OEM 根密钥**]选项卡的设置。

在 **OEM 根私钥** 中选择 **文件**，并指定包含 OEM 引导加载程序私钥的公钥的 PEM 文件。由于已在 **OEM 根私钥** 中指定 PEM 文件，所以无需指定 **OEM 根公钥**。

图 6-34 [FSBL] – [OEM根密钥]选项卡设置示例

进行[**OEM 引导加载程序密钥**]选项卡的设置。

在 **OEM 引导加载程序私钥** 中选择 **文件**，并指定包含 OEM 引导加载程序私钥的公钥的 PEM 文件。由于已在 **OEM 根私钥** 中指定 PEM 文件，所以无需指定 **OEM 根公钥**。

图 6-35 [FSBL] – [OEM引导加载程序密钥]选项卡设置示例

按下“**生成文件**”按钮，生成密钥证书和代码证书。正常生成文件后，输出如下执行结果。

```
Output File: C:\work\fsbl\kcert.bin
Output File: C:\work\fsbl\ccert.bin
操作成功
```

图 6-36 [FSBL]选项卡 执行结果

6.5.2 使用 CLI 版本的 Security Key Management Tool 时

1. 在终端软件中执行 **gencert** 命令。

> skmt.exe /gencert /mode "signature" /loadaddr "02000000" /cfsize "200000" /ver "1"
/oembl "userprog.mot" /oembl_private file="is_ec256_priv.pem"
/oemroot_private file="ms_ec256_priv.pem"
/output_codecert "ccert.bin" /output_keycert "kcert.bin"

```
C:\work\skmt\tool>skmt.exe /gencert /mode "signature" /loadaddr "02000000" /cfsize "200000"
/ver "1" /oembl "C:\work\fsbl\oembl.srec" /oembl_private file="C:\work\fsbl\manifest_signer_
key\ms_ec256_priv.pem" /oemroot_private file="C:\work\fsbl\image_signer_key\is_ec256_priv.pe
m" /output_codecert "C:\work\fsbl\ccert.bin" /output_keycert " C:\work\fsbl\kcert.bin"
Output File: C:\work\fsbl\kcert.bin
Output File: C:\work\fsbl\ccert.bin
```

图 **6-37 gencert**命令执行示例

6.6 示例 6 – RA 产品家族 使用 DOTF 时的程序加密方法

对具有 DOTF 功能的瑞萨器件上的用户应用程序进行加密。

必要的驱动程序及示例的使用方法，请参阅*表 1-1 MCU/MPU 相关信息*中的资料。

6.6.1 使用 GUI 版本的 Security Key Management Tool 时

1. 选择 MCU/MPU 以及加密引擎
   在[**概要**]选项卡中选择 MCU/MPU 以及加密引擎。

图 **6-38 [概要]选项卡**

2. 固件映像的设置

使用**[DOTF/OTFD]**选项卡，加密固件映像。

在 **明文映像** 中设置要加密的固件 mot 文件。

在 **加密地址范围** 中指定在明文映像中输入的加密映像的地址范围。

如果要加密的地址和要执行的地址不同时，请设置 **目标地址**。

在本例中，**加密范围** 为 90000000 至 90000FFF，**目标地址** 设置为 **与明文映像相同**。

图 **6-39 [DOTF/OTFD]**选项卡 **–** 明文映像、加密范围、执行地址的设置示例

3. 加密密钥和 IV 的设置

在 **映像加密密钥** 中设置加密要使用的密钥。在本例中，使用 AES128 的密钥。

在 **初始向量 IV** 中设置加密明文映像要使用的 nonce。在本例中，为了便于理解，IV 选择 **生成随机数值**。

图 **6-40 [DOTF/OTFD]**选项卡 **–** 映像加密密钥、**IV**设置示例

4. 输出文件的设置

在“**加密映像**”中指定要输出的文件名。

在“**输出映像地址和内容**”中，指定要输出的 Motorola 十六进制地址和是否包含未加密的明文数据。

选中“**保留原始地址**”，勾选“包含加密数据范围之外的明文映像数据”。

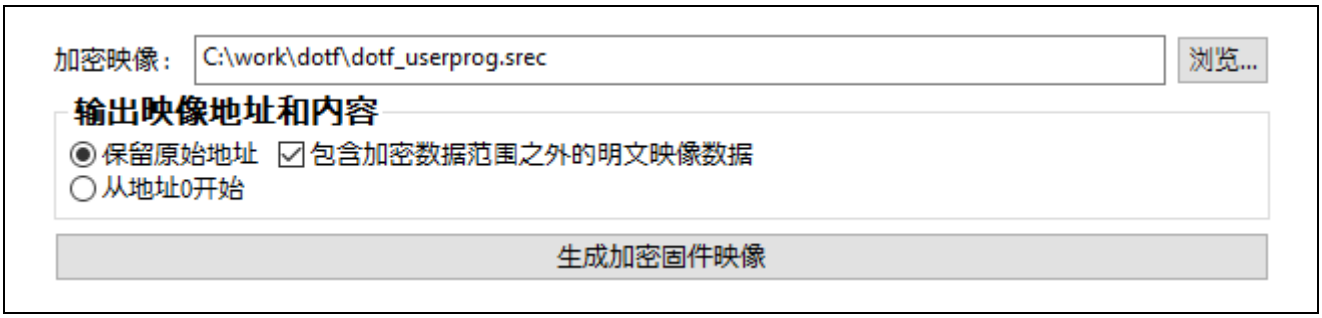

图 6-41 [DOTF/OTFD]选项卡 – 加密映像 输出映像地址和内容的设置示例

如果正常结束，则输出如下。

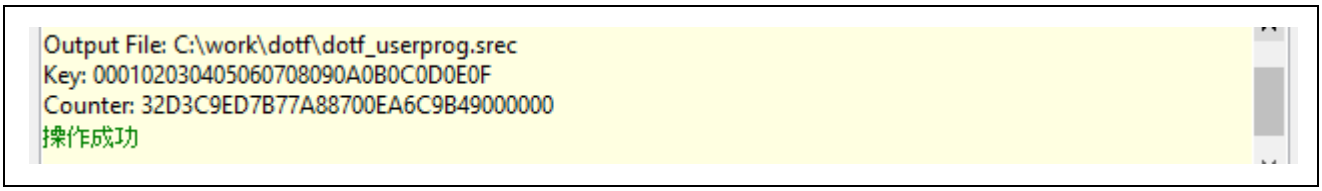

图 6-42 [DOTF/OTFD]选项卡 执行结果

6.6.2 使用 CLI 版本的 Security Key Management Tool 时

1. 在终端软件中执行 **encdotf** 命令。
   > skmt.exe /encdotf /keytype "AES-128" /enckey "000102030405060708090A0B0C0D0E0F" /startaddr "90000000" /endaddr "90000FFF" /incplain /prg "C:\work\dotf\userprog.srec" /output "C:\work\dotf\dotf_userprog.srec"

```
C:\work\skmt\tool>skmt.exe /encdotf /keytype "AES-128" /enckey "000102030405060708090A0B0C0D
0E0F" /startaddr "90000000" /endaddr "90000FFF" /incplain /prg "C:\work\dotf\userprog.srec"
/output "C:\work\dotf\dotf_userprog.srec"
Output File: C:\work\dotf\dotf_userprog.srec
Key: 000102030405060708090A0B0C0D0E0F
Counter: E3B9489DD945947EBC3BE98389000000
```

图 6-43 encdotf命令执行示例

6.7 示例 7 – RA 产品家族 使用安全工厂编程功能时的程序加密方法

输出在安全工厂编程功能中要使用的安全工厂编程文件(`*.sfp`)。

注：安全工厂编程文件中不能包含下表中的功能。这些功能需要通过其他方法输入器件。

表 6-1 **需要添加至 Secure Production Programming** 的文件

项目	必要时	输入方法
用户密钥的注入	应用程序需要安全注入的密钥时	使用 GUI 的[**封装密钥**]选项卡（3.6 [封装密钥] 选项卡）或 **genkey** CLI 命令（4.5 **genkey** 命令选项），生成所有必要的 `*.rkey` 文件。密钥注入要使用的 UFPK 无需与安全工厂编程要使用的 UFPK 相同。
密钥证书	使用带签名认证的 FSBL 时	使用 GUI 的[**FSBL**]选项卡（3.8 [FSBL]选项卡）或 **gencert** CLI 命令（4.7 **gencert** 命令选项），生成必要的二进制密钥证书文件。
代码证书	使用带签名认证或 CRC 验证的 FSBL 时	使用 GUI 的[**FSBL**]选项卡（3.8 [FSBL]选项卡）或 **gencert** CLI 命令（4.7 **gencert** 命令选项），生成必要的二进制代码证书文件。

6.7.1 使用 GUI 版本的 Security Key Management Tool 时

1. 选择 MCU/MPU 以及加密引擎
   在[概要]选项卡中选择 MCU/MPU 以及加密引擎。

图 6-44 [概要]选项卡

2. 生成固件映像文件

使用[**SFP**]选项卡生成安全工厂编程文件(`*.sfp`)。

在[**固件映像**]选项卡的“**明文映像**”中指定要加密的用户程序。选择文件，使用“**添加**”按钮进行添加。

在 “**MCU/MPU** ”下选择要使用的 MCU/MPU 信息。 在本例中，选择 “**RA8D1/M1/T1**”。

在“**最终 DLM/AL 状态**”中选择固件映像写入后的 DLM/AL 状态。在本例中，选择“**OEM PL0 with AL2_KEY**”。

在“**安全编程文件**”中指定输出文件名。

图 **6-45 [SFP] – [**固件映像**]**选项卡等的设置示例

3. 进行[**映像加密密钥**]选项卡的设置。
   在“**密钥数据 (AES-128)**”中指定要用于用户程序及参数信息加密的 AES128bit 密钥数据。为了便于理解，在本例中选择“**使用随机数-输出文件**”，指定密钥数据的输出文件名。
   为简单起见，“**IV**”在本例中选择“**生成随机数值**”。
   在“**UFPK 文件**”中指定 UFPK 文件，在“**WUFPK 文件**”中指定 W-UFPK 文件。
   关于 UFPK 文件及 W-UFPK 文件的生成方法，请参阅 6.1 示例 1 – 在具有 TSIP 的 RX 产品家族 MCU 上安装 AES128 密钥中的步骤 1 至 3。

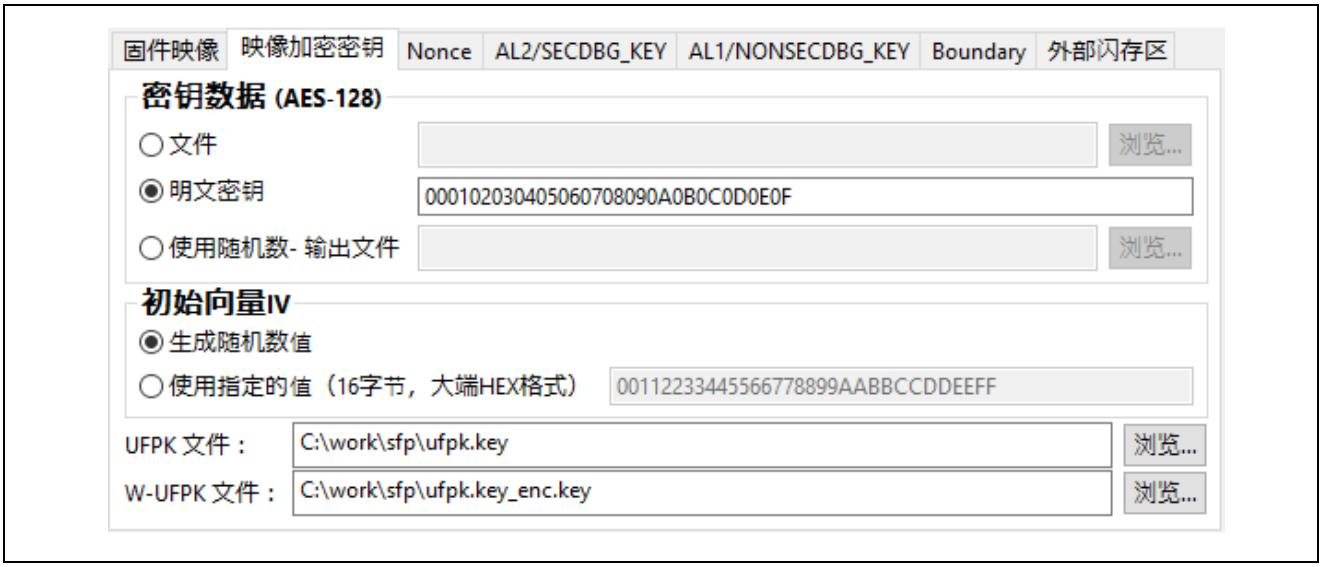

图 6-46 [SFP] – [映像加密密钥]选项卡设置示例

4. 进行**[Nonces]**选项卡的设置。
   为了便于理解，“**初始向量 IV（编程参数）**”及“**初始向量 IV（固件映像）**”在本例中选择“**生成随机数值**”。
   本例中不需要选择“**IV（AL/DLM 键）**”。

图 **6-47 [SFP] – [Nonce]**选项卡设置示例

5. 进行**[AL2/SECDBG_KEY]**选项卡的设置。
   在“**密钥数据**”中指定要用作 AL2_KEY 的 AES128bit 密钥数据。
   为简单起见，“**IV**”在本例中选择“**生成随机数值**”。

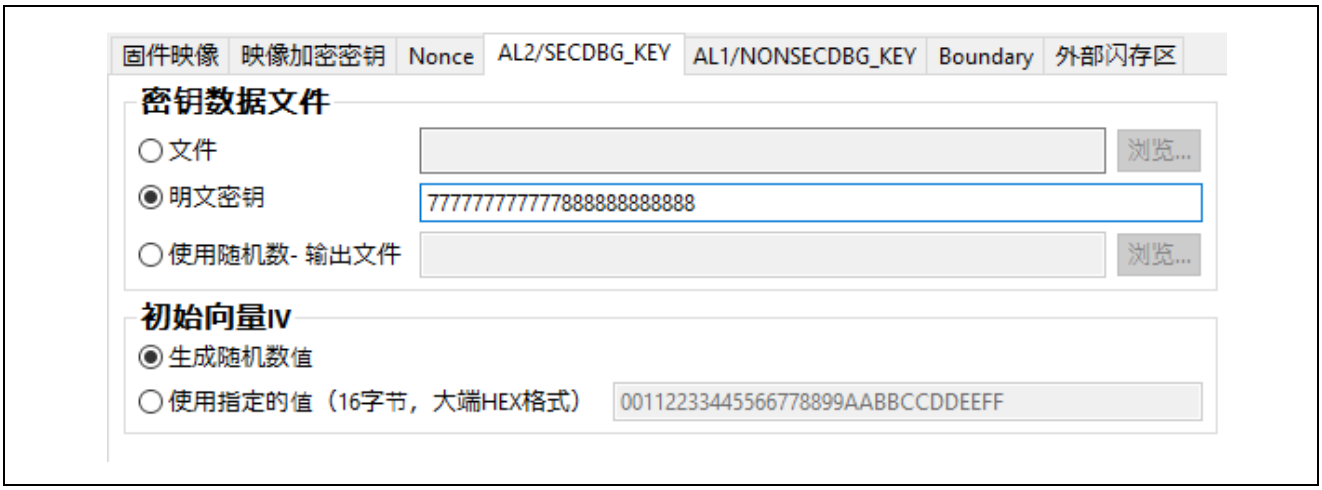

图 **6-48 [SFP] – [AL2/SECDBG_KEY]**选项卡设置示例

按下“**输出文件**”按钮，生成安全工厂编程文件。正常生成文件后，输出如下执行结果。

```
Output File: C:\work\sfp\sfp_userprog.sfp
操作成功
```

图 6-49 [SFP]选项卡 执行结果

6.7.2 使用 CLI 版本的 Security Key Management Tool 时

1. 在终端软件中执行 **encsfp** 命令。

>skmt /encsfp /mcu "RA8D1_M1_T1" /enckey "000102030405060708090a0b0c0d0e0f"
/trn "OEM_PL0_AL2" /prg "C:\work\sfp\userprog.srec"
/al2key "77777777777777778888888888888888"
/ufpk file="C:\work\sfp\ufpk.key" /wufpk "C:\work\sfp\ufpk.key_enc.key"
/output "C:\work\sfp\sfp_userprog.sfp"

```
C:\work\skmt\tool>skmt /encsfp /mcu "RA8D1_M1_T1" /enckey "000102030405060708090a0b0c0d0e0f"
 /trn "OEM_PL0_AL2" /prg "C:\work\sfp\userprog.srec" /al2key "77777777777777778888888888888888
88" /ufpk file="C:\work\sfp\ufpk.key" /wufpk "C:\work\sfp\ufpk.key_enc.key" /output "C:\work
\sfp\sfp_userprog.sfp"
Output File: C:\work\sfp\sfp_userprog.sfp


C:\work\skmt\tool>
```

图 **6-50 encsfp**命令执行示例

7. 注意事项

7.1 使用 Windows 环境时的显示设置

在 Windows 环境中，如果独立 GUI 或 e² studio 插件版本设置为建议设置以外的显示设置，则可能无法显示整个对话框。

可通过以下方式改进显示。

1. 右键单击可执行文件：
   独立版本：`SecurityKeyManagementTool.exe`
   e² studio 插件版本：`e2studio.exe`
   `e2studio.exe` 位于 e² studio 安装的 **eclipse** 文件夹中。
2. 选择**属性**，然后选择**兼容性**选项卡。在**兼容性**选项卡上，单击**更改高 DPI 设置**按钮。
3. 在**高 DPI 缩放覆盖**部分，选中**覆盖高 DPI 缩放行为**复选框，并从下拉列表中选择**系统**。然后单击两个对话框中的**确定**。

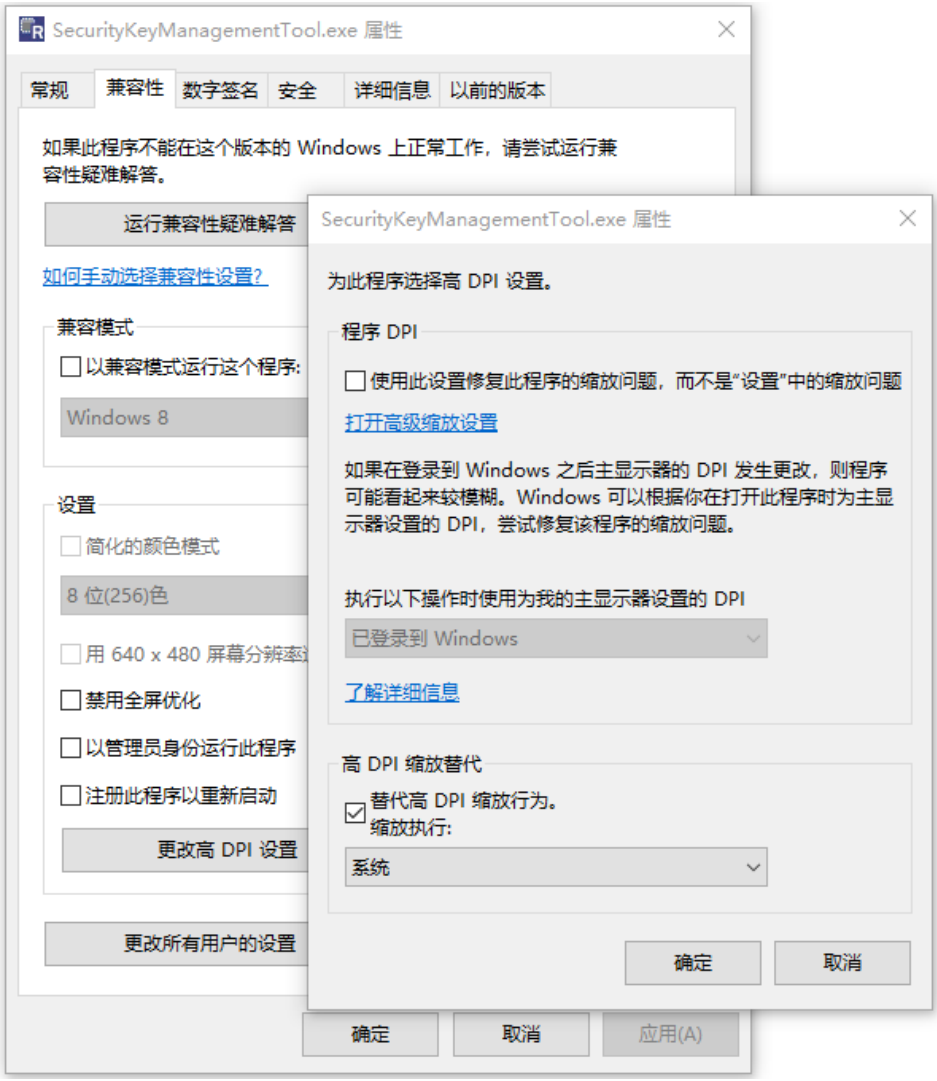

图 7-1 SecurityKeyManagementTool.exe 属性“高 DPI 缩放覆盖”设置

7.2 在 Linux 环境下使用 e²studio 插件版本的注意事项

如果您在 Linux 环境下使用 e² studio 插件版本设置为建议设置以外的显示设置，则可能无法显示整个对话框。安装后必须对命令行可执行文件设置执行权限。 在 e²studio 中安装插件版 Security Key Management Tool 后，为 e²studio 安装文件夹中的以下文件赋予执行权限。

/eclipse/plugins/com.renesas.apltool.skmt_X.X.X.XXXXXXXXXXXX/cli/linux/skmt

7.3 在 macOS 环境下使用 e²studio 插件版本的注意事项

如果您在 macOS 环境下使用 e² studio 插件版本设置为建议设置以外的显示设置，则可能无法显示整个对话框。安装后必须对命令行可执行文件设置执行权限。 在 e²studio 中安装插件版 Security Key Management Tool 后，为 e²studio 安装文件夹中的以下文件赋予执行权限。

/eclipse/plugins/com.renesas.apltool.skmt_X.X.X.XXXXXXXXXXXX/cli/skmt

7.4 macOS 版本的限制

macOS 版本对命令行、单机版和 e2studio 插件版的以下功能有限制

表 7-1 命令行版本的限制

命令选项	选项	限制
genkey	在指定以下特定 /keytype 选项时，省略 /key 选项即可生成密钥。 ”brainpoolP256r1-private” ”brainpoolP384r1-private” ”brainpoolP512r1-private”	不可用。 返回错误″Error: For BRAINPOOLPxxxR1-PRIVATE, key option cannot be omitted when using MAC OS″。
	当指定以下特定 /keytype 选项时，使用 /key 选项输入 PEM 文件。 ”brainpoolP256r1-public”、 ”brainpoolP256r1-private”、 ”brainpoolP384r1-public”、 ”brainpoolP384r1-private”、 ”brainpoolP512r1-public”、 ”brainpoolP512r1-private”、	不可用。 返回错误″Error: For BRAINPOOLPxxxR1-PRIVATE/PUBLIC, PEM file cannot be specified when using MAC OS.″。

表 7-2 对单机版和 e2studio 插件版的限制

标签	操作	限制
封装密钥	在 [封装密钥] 标签中选择以下密钥类型时，请在 [密钥数据文件] 标签中选择 "随机 "来生成密钥。 ”brainpoolP256r1-private” ”brainpoolP384r1-private” ”brainpoolP512r1-private”	密钥生成功能不可用。 当按下生成文件按钮时。 将返回″Error: For BRAINPOOLPxxxR1-PRIVATE, key option cannot be omitted when using MAC OS″错误信息。
	在 [封装密钥] 选项卡上选择以下键类型时、在 [密钥数据文件] 选项卡中，选择 "文件 "并使用 PEM 文件作为输入。 ”brainpoolP256r1-public”、 ”brainpoolP256r1-private”、 ”brainpoolP384r1-public”、 ”brainpoolP384r1-private”、 ”brainpoolP512r1-public”、 ”brainpoolP512r1-private”、	无法处理 PEM 文件。 当按下生成文件按钮时。 将返回″Error: For BRAINPOOLPxxxR1-PRIVATE/PUBLIC, PEM file cannot be specified when using MAC OS.″错误信息。

7.5 安全工厂编程功能的局限性

安全工厂编程功能的用户程序映像大小有上限。

用户程序映像按照 Boot Firmware 应用说明 - Data Packet [Encrypted User Data] - DAT - User data and write address/size 中定义的格式分割。 分割后的用户程序映像的最大总大小为 16 MB。

如果输入的用户程序图像大于最大尺寸，安全密钥管理工具会输出以下错误。

Error: The user program information to be encrypted is too large. The total size of the user program information is less than 16MB.

7.5.1 安全密钥管理工具 User data and write address/size 作成方法

安全密钥管理工具会在用户程序映像的代码闪存、数据闪存和 OFS 区域的每个写入单元（最小 16 字节）中添加 write address/size 信息。如果使用 extarea0/1 指定了外部闪存区域，则会为每个写入单元添加 write address/size 信息。 但是，如果数据是连续的，则连接 User data 以创建 User data and write address/size。User data 部分的最大大小为 1024 字节。

示例：要加密的图像 数据存在于外部闪存区域 0x90000000 至 0x9000040F 中

extarea0 “0x90000000” “0x9000FFFF” “16” 設定時。

Encrypted Data Write Command - Data Packet [Encrypted User Data] – DAT 用的数据

安全密钥管理工具是

第 1 包 DAT：SAD+SIZE+Reserved+（0x90000000-0x900003FF 中的数据）已加密

第 2 包 DAT：SAD+SIZE+Reserved+（0x90000000-0x9000040F 中的数据）已加密

创建数据的方式如下。

7.6 单机版和插件版 在 [概要]选项卡中更改 MCU/MPU 和加密引擎时的注意事项。

- 内容：

使用单机版或插件版时，如果在 [**概要**]选项卡中重新选择 MCU/MPU 和加密引擎，则 [**密钥封装**]选项卡中的 [**密钥数据**]选项卡可能无法正确显示。

- 产生现象的程序：

1. 在[**概要**]选项卡 MCU/MPU 和加密引擎中，选择所需的 MCU/MPU 和加密引擎（例如 RX Family，TSIP）。

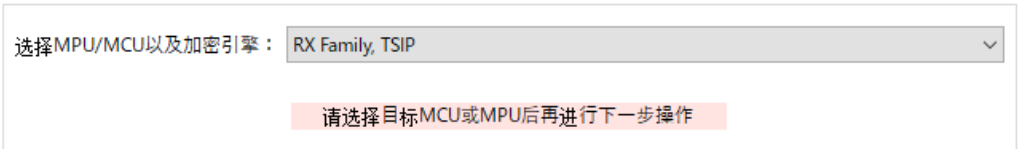

图 **7-2 [概要]选项卡 MCU/MPU 和加密引擎 选择 RX Family，TSIP**

2. 在 [**密钥封装**]选项卡的 [**密钥类型**]选项卡上，选择要使用的密钥类型（在示例中，选择 RSA 2048bits，public）。

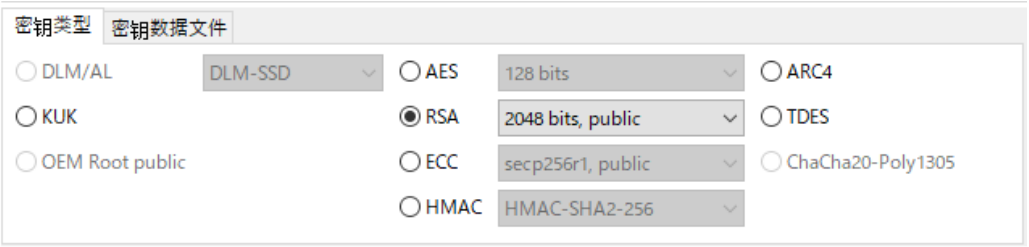

图 **7-3 [密钥封装]选项卡 [密钥类型]选项卡 选择 RSA 2048 位，公开**

3. 重新选择 MCU/MPU 和加密引擎。(在示例中，选择 RX Family，TSIP Lite）。

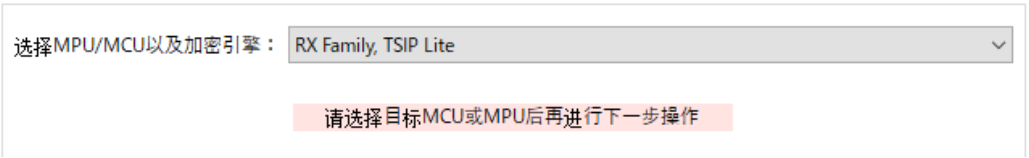

图 **7-4 [概要]选项卡 MCU/MPU 和加密引擎 选择 RX Family，TSIP Lite**

4. [**密钥封装**]选项卡内的 [**密钥类型**]选项卡选择了 AES 128 位，但 [**密钥数据文件**]选项卡显示的是 RSA 公钥的输入显示。

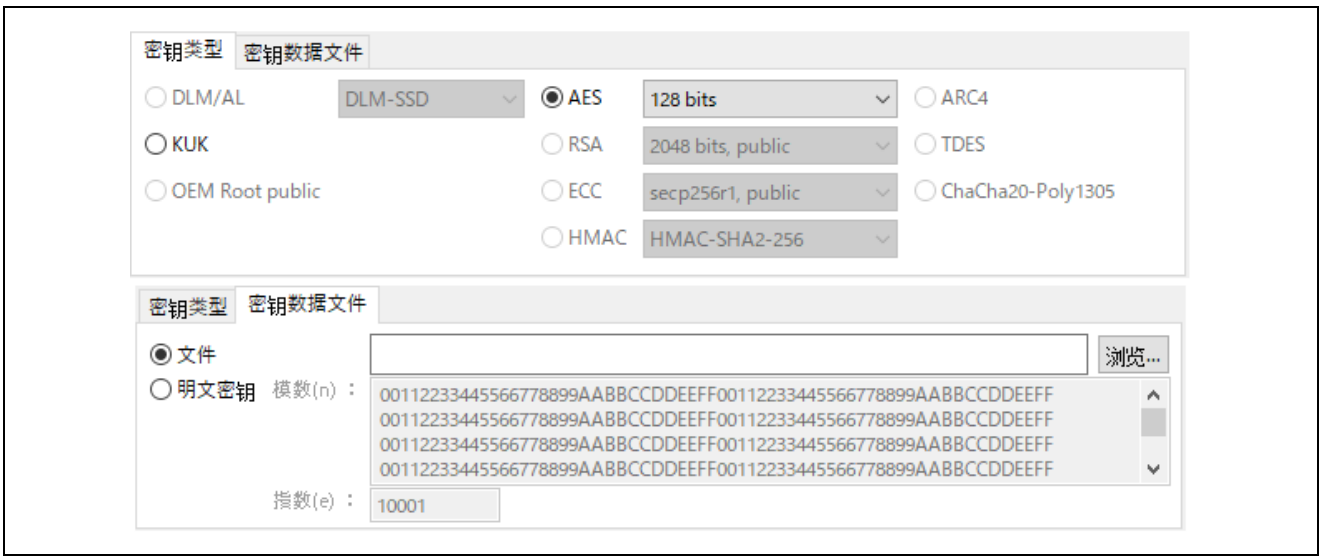

图 **7-5 [密钥封装]选项卡 [密钥类型]选项卡 AES 128bit [密钥数据]选项卡显示 RSA public输入**

- 变通方法：

如果出现这种情况，请在 [**密钥封装**]选项卡的 [**密钥类型**]选项卡上选择不同的密钥类型，[**密钥数据文件**]选项卡将恢复正常。如果要选择 AES，请选择其他密钥类型，然后再次选择 AES，[**密钥数据文件**]选项卡中的显示将恢复正常。

8. 附录

8.1 License

(a) .NET

该工具使用 .NET Foundation 提供的开源软件 .NET。
.NET 许可证见下文。

.NET 许可证：
https://github.com/dotnet/runtime/blob/main/LICENSE.TXT
https://github.com/dotnet/blazor/blob/master/LICENSE.txt

.NET Foundation：
https://dotnetfoundation.org/

(b) Inno Setup

该工具的 windows 版本安装程序是使用 Inno Setup 创建的。
Inno Setup 许可证见下文。

Inno Setup 许可证：
https://jrsoftware.org/files/is/license.txt

Inno Setup：
https://jrsoftware.org/isinfo.php

(c) NSec

本软件的 Ed25519 密钥生成功能使用 NSec 库。
有关 NSec 许可证，请参阅下文。

NSec License :
https://nsec.rocks/license

NSec :
https://nsec.rocks/

(d) Bouncy Castle

本软件的 macOS 版本在 encsfp 命令中使用了 Bouncy Castle。
有关 Bouncy Castle 的许可证，请参阅下文。

Bouncy Castle License :
https://www.bouncycastle.org/about/license

Bouncy Castle :
https://www.bouncycastle.org/

8.2 瑞萨密钥文件（Key File）的格式

8.2.1 文本格式

表 8-1 瑞萨密钥文件的格式 文本格式

项目	定义
字符	仅限 ASCII 字符 （不能使用 Unicode 和多字节字符）
空格	TAB(0x09) / Space(0x20)
每行最大字节长度	80 个字节
换行	CRLF(0x0D,0x0A) / CR(0x0D) / LF(0x0A)
Base64 编码	Chars = 英文字母 / 数字 / "+" / "/" Pad = "=" 行长度= 64 chars （最末行除外）

8.2.2 文件结构

密钥文件由一下三部分组成。

```
<Header>
<Base64Lines>
<Footer>
```

<Header> 和 <Footer> 是以下列出的固定字符串。

Header = "-----BEGIN RENESAS KEY-----"

Footer = "-----END RENESAS KEY-----"

<Base64Lines> 由多行字符组成，代表密钥数据结构，具体定义如下：

密钥数据结构是 Base64 编码的二进制数据。

Footer 后面的行是空行，每一行数据后面的空格都将被忽略。

8.2.3 文件扩展名

".rkey"

8.2.4 密钥数据

8.2.4.1 结构

密钥数据按照以下格式和大小存储，大端字符顺序。

表 8-2 瑞萨密钥文件的格式 关键数据结构

名称	类型	大小	说明
标识符	Char[4]	4 字节	固定的四个字符"REK1"
套件版本	Integer	4 字节	数据格式版本. 当前必须设定为 1。
保留区域	Byte[7]	7 字节	0。
密钥类型	Byte	1 字节	[DLM 密钥] 0。 [用户密钥] keytype 值。 （参考 4.5.2 **keytype**）
加密的密钥长度	Integer	4 字节	"Encrypted Key"的长度（N 字节）
W-UFPK	Byte[36]	36 字节	从瑞萨 DLM 服务器获得的 W-UFPK 文件中的值 最前面 4 个字节是 Shared Key Number 剩余 32 个字节是 WUFPK 的值。
初始向量	Byte[16]	16 字节	I 封装用户密钥使用的初始向量 IV。
加密的密钥	Byte[N]	N 字节	通过 UFPK 加密后的用户密钥数据+MAC 值
数据 CRC	Byte[4]	4 字节	除了这四个字节以外的其他所有数据的 CRC 校验值。 Initial Value = 0xFFFFFFFF Magic number = 0x04C11DB7

8.2.5 加密密钥计算公式

在使用 UFPK 或 KUK 封装用户密钥时，加密和 MAC 的生成按以下公式进行。

- RA 系列、RX 系列和 Synergy 平台的加密密钥计算公式。

```
uint32_t i = 0;
uint32_t n            = User key byte size;
uint8_t IV[16];                                  // Initial Vector(128bit)
uint8_t CBCKey[16]     = UFPK[0:15] or KUK[0:15];   // CBCKEY in either UFPK or KUK
uint8_t CBCMACKey[16] = UFPK[16:31] or KUK[16:31]; // CBCMACKEY in either UFPK or KUK
uint8_t User_Key[n];                             // Plain text User key
uint8_t MAC[16] = {0};
uint32_t encrypted_key[n];                       // Encrypted Key

for (i = 0; i < n; i += 16)
{
   MAC[0:15]    =
      AES128-Enc(CBCMACkey[0: 15], xor_16byte(User_Key[i: i+15], MAC[0: 15]));
   encrypted_key[i: i+15] =
      AES128-Enc(CBCkey[0: 15],    xor_16byte(User_Key [i: i+15], IV[0: 15]));
   IV[0: 15]      = encrypted_key [i: i+15];
}
encrypted_key[i: +15] = AES128-Enc(CBCkey[0: 15], xor_16byte(MAC[0: 15], IV[0: 15]));
```

* : AES128-Enc() 表示密钥长度为 128 位的 AES ECB 模式加密。
第一个参数：加密时使用的密钥 第二个参数：要加密的明文数据

- RZ/T2M2、RZ/T2ME、RZ/T2L 和 RZ/N2L 的加密密钥计算公式

```
uint32_t n            = User key byte size;
uint8_t IV[16];                                  // Initial Vector(128bit)
uint8_t CBCKey[16]     = UFPK[0:15] or KUK[0:15];   // CBCKEY in either UFPK or KUK
uint8_t CMACKey[16]    = UFPK[16:31] or KUK[16:31]; // CMACKEY in either UFPK or KUK
uint8_t User_Key[n];                             // Plain text User key
uint8_t MAC[16] = {0};
uint32_t encrypted_key[n];                       // Encrypted Key

MAC[0:15]           = AES128-CMAC(CMACkey[0: 15], IV[0: 15], User_Key[0: n-1]); *1
encrypted_key[0: n-1] = AES128-CBC(CBCkey[0: 15],   IV[0: 15], User_Key[0: n-1]); *2
encrypted_key[n: n+15] = MAC[0:15];
```

*1 : AES128-CMAC() 表示 AES CMAC 生成操作，密钥长度为 128 位。
第一个参数：加密密钥；第二个参数：初始向量；第三个参数：明文数据。
*2 : AES128-CBC() 表示密钥长度为 128 位的 AES CBC 加密操作。
第一个参数：加密密钥；第二个参数：初始向量；第三个参数：明文数据。

8.3 安全工厂编程文件格式

8.3.1 文件格式

二进制文件，由包含 TLV 字段以外信息的 Pre-Data Field、包含 TLV 字段大小的 “TLV 长度字段 ”和包含加密数据的 “TLV（类型-长度-值）字段 ”组成。
预数据字段的文件格式因 MCU/MPU 而异。 请参见表 8 3，了解哪种 MCU/MPU 具有哪种文件格式。
数据按大端顺序排列。

表 8-3 MCU/MPU 与 Pre-Data Field 对应表

Format Version	MCU/MPU	Format Image
V1	RA8D1_M1_T1	图 8-1
V2	RA4L1, RA8D2_M2_T2_P1	图 8-2

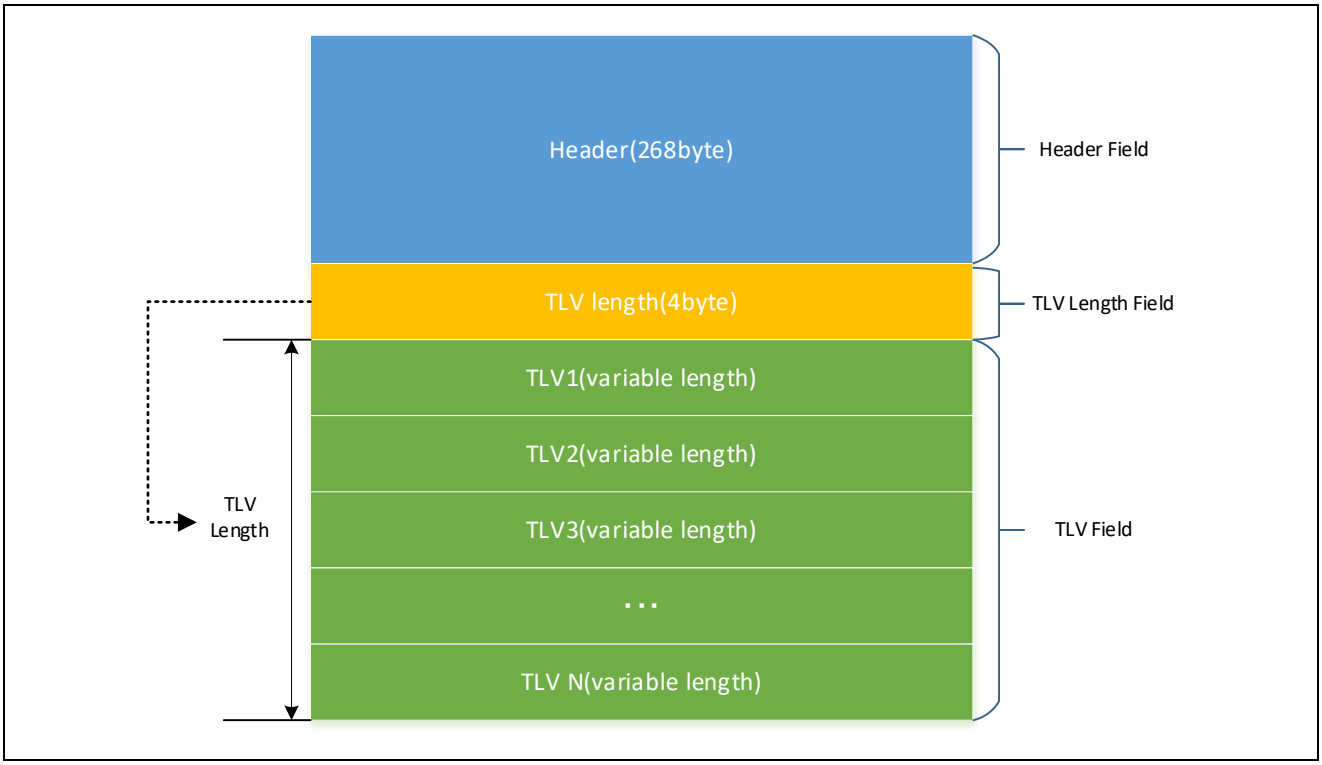

图 8-1 V1格式图像图

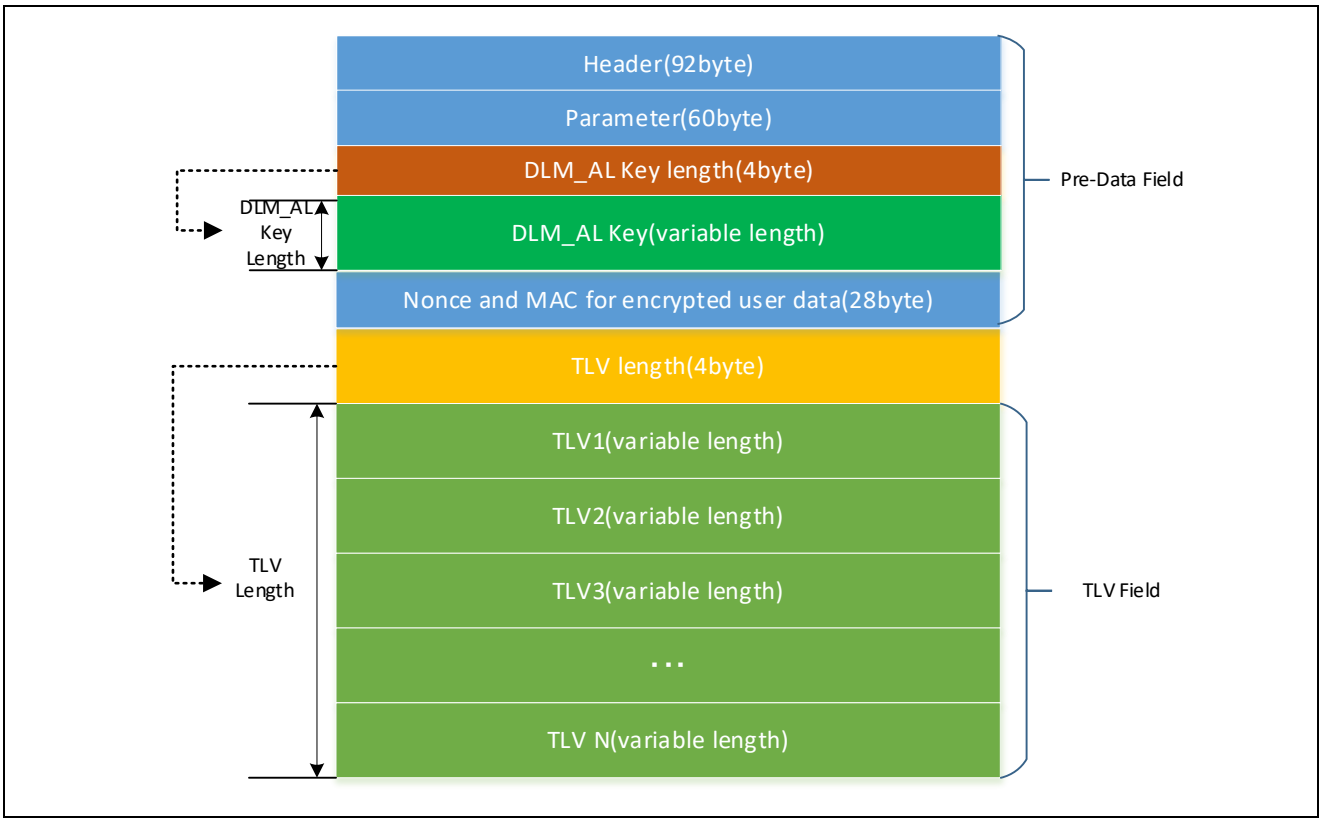
Header(92byte)
Parameter(60byte)
DLM_AL Key length(4byte)
DLM_AL Key(variable length)
Nonce and MAC for encrypted user data(28byte)
TLV length(4byte)
TLV1(variable length)
TLV2(variable length)
TLV3(variable length)
...
TLV N(variable length)
DLM_AL Key Length
TLV Length
Pre-Data Field
TLV Field

图 8-2 V2格式图像图

8.3.1.1 Pre-Data Field

(1) V1 格式的 Pre-Data Field

V1 格式中的前数据字段是一个固定长度字段，长度为 268 字节，用于存储唯一信息。Header 字段是一个 268 字节的字段，用于存储固有信息。

表 8-4 V1 格式的 Pre-Data Field

字段	大小[字节]	说明
Magic	4	使用幻数设置以无符号 4 字节整数表示“sfpr”ASCII 代码的“0x73667072”。
Version	4	格式版本 高位 2 字节表示主版本，低位 2 字节表示副版本。 V.1.00 时设置“0x00010000”。
W-UFPK	36	瑞萨 DLM 服务器发送的 W-UFPK 文件的值 前 4 个字节为共享密钥号 其余 32 个字节为 WUFPK 值
Initialization Vector used for encrypting ENKY	16	封装用于加密用户程序及参数的密钥时的初始向量
ENKY	32	将用于加密参数值及用户程序的密钥，以 UFPK 进行加密后的数据
Nonce used for encrypting parameters	12	加密参数时使用的 Nonce 数据
Encrypted Parameter	16	将参数值以加密密钥加密后的数据
MAC for Encrypted Parameter	16	将参数值以加密密钥加密后生成的 MAC 值
Reserved	3	0xFF
Encrypted AL2 Key Enable	1	Initialization Vector for used encrypted AL2 Key 字段和 Encrypted AL2 Key with MAC 字段的有效无效标志 0：无效 其他：有效
Initial Vector used for encrypting AL2 Key	16	加密 AL2 Key 时使用的初始向量值
Encrypted AL2 Key	32	将 AL2 Key 以 UFPK 加密后的数据＋MAC 值
Reserved	52	0xFF
Nonce used for encrypting user data	12	加密用户程序时使用的 Nonce 值
MAC for encrypted user data	16	加密用户程序时生成的 MAC 值

(2) V2 格式的 Pre-Data Field

V2 格式的预数据字段包含加密用户数据的标 Header、Parameter、DLM_AL Key（包括 DLM_AL Key length）、Nonce 和 MAC。

DLM_AL Key 是一个长度可变的字段。

(a) Header Field

标头字段是一个 92 字节的固定长度字段，包含魔法编号、版本、W-UFPK、ENKY IV 和 ENKY 数据。

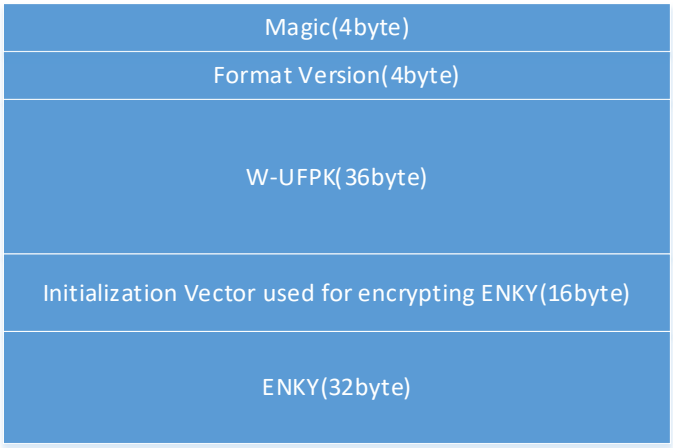

图 8-3 V2 格式 Header Field

表 8-5 V2 格式的 Pre-Data Field

字段	大小[字节]	说明
Magic	4	使用幻数设置以无符号 4 字节整数表示“sfpr”ASCII 代码的“0x73667072”。
Version	4	格式版本 高位 2 字节表示主版本，低位 2 字节表示副版本。 V.2.00 时设置“0x00020000”。
W-UFPK	36	瑞萨 DLM 服务器发送的 W-UFPK 文件的值 前 4 个字节为共享密钥号 其余 32 个字节为 WUFPK 值
Initialization Vector used for encrypting ENKY	16	封装用于加密用户程序及参数的密钥时的初始向量
ENKY	32	将用于加密参数值及用户程序的密钥，以 UFPK 进行加密后的数据

(b) Parameter Field

Parameter是一个 60 字节的固定长度字段，用于存储加密Parameter信息、Nonce 和Parameter加密过程中使用的 MAC 信息。

图 8-4 V2 格式 Parameter Field

表 8-6 V2 格式的 Parameter Field

字段	大小[字节]	说明
Nonce used for encrypting parameter	12	用于加密参数信息的 Nonce
Encrypted Parameters	32	用加密密钥加密的参数信息
MAC for Encrypted Parameter	16	参数信息被加密密钥加密后生成的 MAC 值。

(c) DLM_AL Key Field

DLM_AL Key Field是一个长度可变的字段，包含封装 DLM_AL Key的加密数据（加密 DLM Key数据）、用于加密封装 DLM_AL Key的 Nonce（用于加密 DLM Key数据的 Nonce）和 MAC（用于加密 DLM Key数据的 MAC）。加密 DLM Key数据的 MAC）。

DLM_AL Key长度是一个 4 字节的固定长度字段，用于存储 DLM_AL Key字段的字节数。

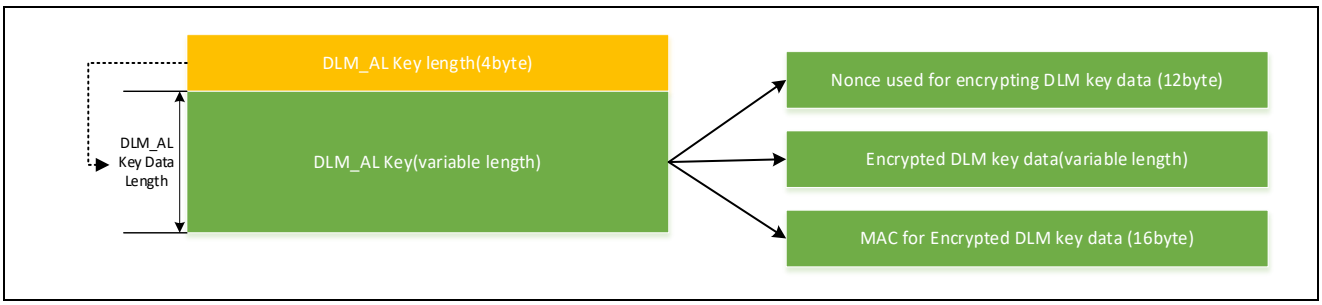

图 8-5 V2 格式 DLM_AL Key Field

表 8-7 V2 格式的 DLM_AL Key Field

字段	大小[字节]	说明
DLM_AL Key length	4	DLM_AL Key Field 的总字节大小
Nonce used for encrypting DLM key data	12	用于加密封装 DLM_AL 密钥的加密 DLM_AL 密钥数据（加密 DLM 密钥数据）的 Nonce。
Encrypted DLM key data	变长	封装 DLM_AL 密钥的加密数据。
MAC for Encrypted DLM key data	16	对封装 DLM_AL 密钥的加密 DLM_AL 密钥数据（加密 DLM 密钥数据）进行加密时使用的 MAC。

(d) Nonce and MAC for encrypted user data Field

Nonce and MAC for encrypted user data Field 是一个 28 字节的固定长度字段，用于存储用户程序加密过程中使用的 nonce 和生成的 MAC 信息。

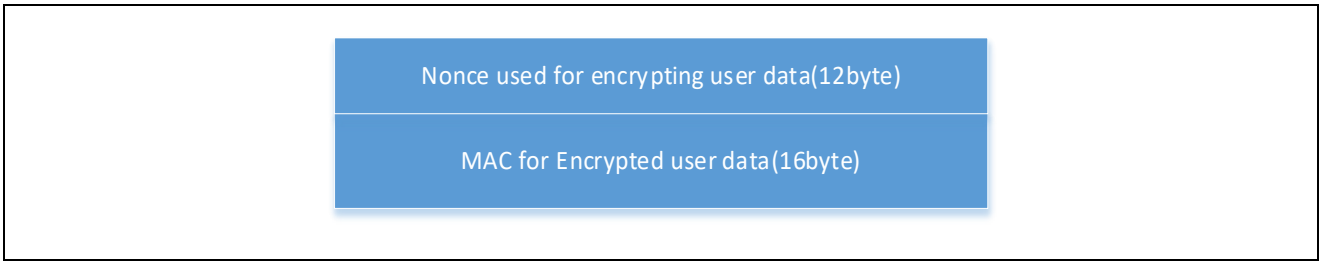

图 8-6 V2 格式 Nonce and MAC for encrypted user data Field

表 8-8 V2 格式的 Nonce and MAC for encrypted user data Field

字段	大小[字节]	说明
Nonce used for encrypting user data	12	加密用户程序时使用的 Nonce 值。
MAC for Encrypted user data	16	加密用户程序时生成的 MAC 值。

8.3.1.2 TLV Length Field

TLV Length是一个固定长度字段，由 4 个字节组成，用于描述 TLV 字段的大小。

表 8-9 安全工厂编程文件格式 TLV Length 字段

字段	大小[字节]	说明
TLV Length	4	TLV 字段的总字节大小

8.3.1.3 TLV Field

TLV 是一个长度可变的字段，其 Type-Length-Value格式为以下值之一。

- Encrypted User Program（加密后的程序）

TLV的格式如图 8.2TLV格式所示。

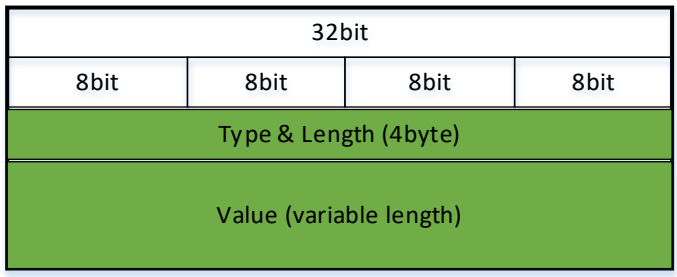

图 8-7 TLV格式

表 8-10 安全工厂编程 TLV 格式

字段	大小[字节]	说明
Type & Length	4	Value 的类型及 word(32bit)大小
Value	可变长度	每种类型的数据 大小是 Length 指定值的 4 倍 （Length 用于指定 word 大小）

(1) Type&Length 字段的详情

Type&Length字段被定义为32位的位字段。

表 8-11 安全工厂编程 TLV 格式 Type&Length 字段

字段	位字段	大小（位）	Bit 位置	说明
Type	Class	4	31:28	Value 的分类 bit 31 28 0 0 0 0 ：加密的用户数据 *其他为预约
	<Class unique>	4	27:24	每个 Class 的字段（详情另行记载）
Length	word size	24	23:0	Value 的 word 大小(1word = 4byte) [值]0～16,777,215

(2) 每个 Class 的唯一字段

下面介绍每个Class的类唯一字段的值。

表 8-12 安全工厂编程 TLV 格式 Class 字段为“加密的用户数据”时

位字段	大小（位）	Bit 位置	说明
Use type	4	27:24	使用类型 [value] bit 27 24 0 0 0 0 ：加密的用户数据 *其他为预约

8.3.2 文件扩展名

".sfp"

8.4 密钥证书/代码证书

8.4.1 密钥证书文件格式

以下数据结构的二进制文件。
详情请参阅支持FSBL功能的器件的用户手册。

表 8-13 密钥证书数据格式

字段		大小[字节]	说明
Header	Magic	4	0x6B657963
	Manifest Version	4	0x00010000
	Flags	4	Reserved (0x00000000)
	Reserved	20	Reserved (ALL 0)
TLV Length		4	密钥证书的 TLV 字段的数据字节长度 0x000000AC
TLV ECC PUBKEY	Type&Length	4	0x00088010
	Value	64	在/gencert/oemroot_public 选项中指定的 OEM 根公钥
TLV KEYHASH	Type&Length	4	0x10144008
	Value	32	在/gencert/oembl_public 选项中指定的 OEM 引导加载程序公钥的 SHA2-256 HASH 值
TLV EXPECTED_SIG	Type&Length	4	0x20088410
	Value	64	签名值

8.4.2 在 mode “signature”下生成的代码证书文件格式

以下数据结构的二进制文件。
详情请参阅支持FSBL功能的器件的用户手册。

表 8-14 **mode** 为“**signature**”时的代码证书数据格式

字段		大小[字节]	说明
Header	Magic	4	0x636F6463
	Manifest Version	4	0x00010000
	Flags	4	0x00000000
	Load Addr	4	在/gencert/loadaddr 选项中指定的 OEM 引导加载程序起始地址
	Dest Addr	4	在/gencert/loadaddr 选项中指定的 OEM 引导加载程序起始地址
	Image size	4	省略在/gencert/oembl_size 选项中指定的 OEM 引导加载程序大小/oembl_size 时，根据在/cfsize 选项和/oembl 中输入的 OEM 引导加载程序文件计算的 OEM 引导加载程序大小
	Image version	4	在/gencert/ver 选项中指定的版本信息
	Build number	4	Reserved (ALL 0)
TLV Length		4	代码证书的 TLV 字段的数据字节长度 0x000000AC
TLV ECC PUBKEY	Type&Length	4	0x00088010
	Value	64	OEM 根公钥
TLV EXPECTED_CRC	Type&Length	4	0x40000001
	Value	4	OEM 引导加载程序的 CRC32 值
TLV SIGNER_ID	Type&Length	4	0x10144008
	Value	32	OEM 引导加载程序公钥的 SHA2-256 HASH 值
TLV EXPECTED_SIG	Type&Length	4	0x25088410
	Value	64	签名值

8.4.3 在 mode “crc”下生成的代码证书文件格式

以下数据结构的二进制文件。
详情请参阅支持FSBL功能的器件的用户手册。

表 8-15 **mode “crc”**时的代码证书数据格式

字段		大小[字节]	说明
Header	Magic	4	0x636F6463
	Manifest Version	4	0x00010000
	Flags	4	0x00000000
	Load Addr	4	在/gencert/loadaddr 选项中指定的 OEM 引导加载程序起始地址
	Dest Addr	4	在/gencert/loadaddr 选项中指定的 OEM 引导加载程序起始地址
	Image size	4	省略在/gencert/oembl_size 选项中指定的 OEM 引导加载程序大小/oembl_size 时，根据在/cfsize 选项和/oembl 中输入的 OEM 引导加载程序文件计算的 OEM 引导加载程序大小
	Image version	4	0
	Build number	4	Reserved (ALL 0)
TLV Length		4	代码证书的 TLV 字段的数据字节长度 0x000000AC
TLV EXPECTED_CRC	Type&Length	4	0x40000001
	Value	4	OEM 引导加载程序的 CRC32 值
TLV SIGNER_ID	Type&Length	4	0x10144008
	Value	32	使用 OEM 引导加载程序的 CRC32 值填补。

8.4.4 CRC32 的计算公式

CodeCert的TLV EXPECTED_CRC的计算值如下。

- CRC32 (x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1)

多项式：0xEDB88320(Bit revesed)

位移方向：右移

输入位反转：无

输出位反转：有

初始值：0xFFFFFFFF

8.5 TSIP Update

8.5.1 用户程序的加密公式

用户程序的加密和 MAC 生成按以下公式进行。

```
uint32_t i = 0;
uint32_t n            = User program byte size;
uint8_t IV[16]        = IV[0:15];                              // Initial Vector(128bit)
uint8_t IEkey[16]     = Image Encryption Key[0:15];
uint8_t IEMACKey[16] = Image Encryption Key[16:31];
uint8_t KEKey[16]     = Key Encryption Key[0:15];
uint8_t User_Program[n];
uint8_t MAC[16] = {0};
uint32_t encrypted_user_program[n];                           // Encrypted user program

for (i = 0; i < (n-16); i += 16)
{
   MAC[0:15]    =
      AES128-Enc(IEMACkey[0: 15], xor_16byte(User_Program[i: i+15], MAC[0: 15]));
   encrypted_user_program[i: i+15] =
      AES128-Enc(IEkey[0: 15],    xor_16byte(User_Program[i: i+15], IV[0: 15]));
   IV[0: 15]    = encrypted_user_program[i: i+15];
}
encrypted_user_program[i: i+15]
                  = AES128-Enc(IEkey[0: 15], xor_16byte(MAC[0: 15], IV[0: 15]));
SessionKey0[0:15] = AES128-Enc(KEkey[0: 15], IEKey[0: 15]);
SessionKey1[0:15] = AES128-Enc(KEkey[0: 15], IEMACKey[0: 15]);
```

* AES128-Enc() 表示 AES ECB 模式密钥长度为 128 位加密。
第一个参数：加密时使用的密钥 第二个参数：纯文本数据

版本更新历史	安全密钥管理工具用户手册

版本	日期	描述	
		页码	概要
1.00	2021 年 12 月 28 日	—	初版发行
1.01	2022 年 3 月 31 日	—	3. 已添加 GUI 功能说明。 5. 已添加 GUI 说明。 6. 已添加 GUI 操作说明
1.02	2022 年 6 月 30 日	—	4.5.1 mcu 选项 RA-SCE5_B 附加。 4.5.3.2 文件输入 pem 文件添加。 5.2.1 使用 Linux 版图形用户界面时的使用方法 更改。
1.03	2022 年 12 月 29 日	—	5.2 添加 e2studio 插件版本说明。 付録 Renesas Key File Format 追記。
1.04	2023 年 9 月 29 日	—	3.7 已添加 [TSIP UPDATE] 选项卡。 4.5.1 mcu 选项 RE-TSIPLite 删除。 4.6 enctsip 命令选项 附加说明。 4.7 calcresponse 命令选项 附加说明。
1.05	2023 年 12 月 22 日	—	1.5 附加安全功能 3.8 已添加 [FSBL] 选项卡。 3.9 已添加 [DOTF] 选项卡。 3.10 已添加 [SFP] 选项卡。 4.7 gencert 命令选项 附加说明。 4.8 encdotf 命令选项 附加说明。 4.9 encsfp 命令选项 附加说明。 6.5 FSBL 密钥证书/代码证书生成方法 附加。 6.6 使用 "即时解密 "附加功能。 6.7 使用安全出厂编程附加功能。
1.06	2024 年 3 月 29 日	P.26 P.47 P.48 P.81 P.91 P.95 P.96 P.98	2.2.2 e2studio 操作检查 2024-01 版更新 3.7.3 已添加 加密范围的上限为 8 MB。 3.7.6 可指定 RSU 标头 Ver 2 更改。 4.5.3.2 pem 文件新增 RSA-2048-public-TLS 支持非对称密钥。 4.6 ver 选项 2 添加。 如果省略，则改为 2。 已添加 endaddr 选项 加密区域大小的上限为 8 MB。 4.6.2 选项 RSU 标头 V1 Sequence Number 始终为 1 更正。 已添加 Execution Address 固定值 0xFFFEFFFC。 RSU 标头 V2 附加。 4.7 可在 ver 选项 1-4,294,967,295 中指定 已添加。

版本	日期	描述	
		页码	概要
1.07	2024 年 8 月 30 日	P.8 P.11 P.24 P.26 P.37 P.44 P.45 P.57 P.63 P.68 P.73 P.74 P.80 P.81 P.90 P.90 P.109 ～110 P.108 P.118 ～119 P.164 ～165 P.168 P.177	术语 加密 KUK 附加 1.2 表 1-2 更新 2.1 表 2-1 和表 2-2 支持 RX RSIP-E11A、RZ/T2M、RZ/T2ME、RZ/T2L、RZ/N2L 2.2.2 支持的操作系统 新增 Ubuntu 22.04 和 macOS 更新 e²studio 版本 2024-07 3.6.2.1 表 3-6 有关新增 macOS 版本的说明 3.6.5 添加 ″字节序″ 和 ″输出附加数据″ 表 3-8 补充。 3.7 TSIP UPDATE -> TSIP Update 标签名称变更 3.9 DOTF -> DOTF/OTFD 标签名称变更 3.10 最后的 DLM/AL 状态 ″ OEM PL0 with AL2_KEY and AL1_KEY ″选项已删除 附加说明 3.10.5 附加说明 4.5 genkey 选项 添加了 fileadd 选项和 bswap 选项 4.5.1 表 4-8 更新 支持 RZ/T2M、RZ/T2ME、RZ/T2L、RZ/N2L、RX-RSIP-E11A 4.5.3.1 表 4-44 Qx → Qy 更正 4.5.3.2 表 4-53、4-54 有关新增 macOS 版本的说明 4.5.5 添加了 fileadd 选项 4.5.6 添加了 bswap 选项 4.9 删除 all_key 选项 4.9 trn ″OEM_PL_AL2_1 ″选项已删除(SPDMID-6670) 5.1.3 添加 macOS 版本(SPDMID-6460) 7.3 macOS 版本 关于使用 e2studio 插件 Plugin 的注意事项 7.4 macOS 版本 使用 e2studio 插件的注意事项 附加说明 8 C 瑞萨密钥文件（Key File）的格式 添加 5) 加密密钥计算公式 F TSIP Update 添加1) 用户程序的加密公式
1.08	2025 年 2 月 19 日	P.4 P.24, 25 P.26 P.37 P.46 P.48 P.49 P.65 -66 P.69 P.70 P.71 P.72 P.73	RSIP-Exxx 已添加到说明中 表 2-1、2-2 增加了 RSIP-E11A PM、CM 2.2.2 更新软件环境 3.6.2.1 表 3-6 已添加的 Ed25519 3.7 更新图 3.16 3.7.3 更新图 3.19 3.7.4 更新图 3.20 已添加的 Flash Write Size, Data Flash 3.10 更新图 3.35 已添加的表 3-13 表 3-14 已添加的 DPL with SECDBG_KEY and NONSECDBG_KEY 3.10.3 更新图 3.38 已添加的 IV(AL/DLM key) 3.10.4 更改 AL2_KEY -> AL2_KEY/SECDBG_KEY 选项卡 3.10.5 更改 AL1_KEY -> AL1_KEY/NONSECDBG_KEY 选项卡 已添加的 3.10.6 Boundary 选项卡 已添加的 3.10.7 **[外部闪存区]**选项卡

		P.76 P.80 P.89 P.91 P.99 P.100 P.100 -104 P.104 P.107 P.118 P.119, 120 P.122 P.123 P.124 P.155 -156 P.169 -173 P.175, 176 P.176 P.177- 178 P.179 -189	4.2 表 4-3 已添加的 Renesa Partition Data File 4.5.1 表 4-8 已添加的 RSIP-E11A PM, CM 4.5.3.2 表 4-54 已添加的 Ed25519 4.5.3.3 表 4-55 已添加的 Ed25519 4.5.7 表 4-73 已添加的 Ed25519 4.6 表 4-74 已添加的 flash_wsize 和 df_ena 选项 更新 4.6.1 ver 选项 表 4-76 删除 本版本中只有一个 已添加的 4.6.5 df_ena 选项 4.9 已删除 MacOS 版本的注释 更新说明。 表 4-89 已添加的 nonce_key, secdbgkey 和 nonsecdbgkey, 选项 表 4-90 ,4-91 已添加的 boundary iv_secdbgkey, iv_nonsecdbgkey, output_secdbgkey,和 output_nonsecdbgkey 选项 4.9.1 表 4-92 已添加的 RA4L1, 将 RA8x1 改为 RA8D1_M1_T1。 4.9.2 表 4-93 已添加的 DPL_SECDBG_NONSECDBG 已添加的表 4-94 已添加的 4.9.4 boundary 选项 已添加的 4.6.5 extarea0/1 选项 6.4.1 从图 6-26 更新至图 6-29 6.7 从图 6-45 更新至图 6-50 已添加 MCU/MPU 选择说明 已添加 IV（AL/DLM 键）说明 更新选项卡名称 7.4 删除 encsfp 的限制 已添加的 7.5 **安全工厂编程功能的局限性** 已添加的 7.6 独立**版或 e²studio 插件版 [SFP] 选项卡 设置文件时的限制** 重新调整了 8 个章节的结构。 8.1 已添加的 NSec 和 Bouncy Castle 已添加的 Format Version 2
1.09	2025 年 7 月 1 日	P.24 P.25 P.26 P.33 P.34 P.55 P.65 P.66 P.72 P.79 P.81	2.1 表 2-1、表 2-2 RA 产品家族 RSIP-E50D 安全功能和保护模式、 RSIP-E50D 兼容模式 追加 RX 产品家族 RSIP-E11A 安全功能和保护模式、 RSIP-E11A 兼容模式 追加 RX 产品家族 RSIP-E11A 削除 2.2.2 支持 OS Windows11 和 Ubuntu 24.04 追加 Ubuntu 20.04 LTS 削除 软件环境更新 3.6 ChaCha20-Poly1305 追加 3.6.1 表 3-5 从 SHA-HMAC-xxx 改为 HMAC-SHA-1 或 HMAC-SHA2-xxx。 HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 追加 3.8.2 输出测量报告追加 3.10 表 3-13 RA8D2/M2/T2/P1 追加 表 3-14 OEM PLO with AL2_KEY and AL1_KEY 追加 3.10.6 配置选项 削除 4.5.1 表 4-8 RA-RSIP-E50D, RA-RSIP-E50D-CM 追加 4.5.2 表 4-13 HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 追加

			表 4-16 ChaCha20-Poly1305 追加
		P.82	4.5.3.1
		P.86	表 4-47 HMAC-SHA3-224 追加
			表 4-48 HMAC-SHA3-256 追加
			表 4-49 HMAC-SHA3-384 追加
			表 4-50 HMAC-SHA3-512 追加
		P.87	表 4-53 ChaCha20-Poly1305 追加
		P.96	4.5.4 表 4-69
			HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512 追加
		P.100	表 4-72 ChaCha20-Poly1305 追加
		P.103	4.6 表 4-77 motoffset 选项追加
			4.6.1.2 说明 motoffset 选项的附加说明。
		P.110	4.7 表 4-85 measurement_report 选项追加
		P.122	4.9.1 表 4-96 RA8D2_M2_T2_P1 追加
			4.9.2 表 4-97,表 4-98 OEM_PL0_AL2_1 追加
			7.6 更新说明。
		P.182	
			8.3.1 表 8-3 RA8D2_M2_T2_P1 追加

安全密钥管理工具 用户手册

出版日期： 版本 1.09 2025 年 7 月 1 日

出版方： Renesas Electronics Corporation

安全密钥管理工具

Renesas Electronics Corporation

R20UT5349CJ0109