RZ/G1 Trusted Secure IP (TSIP) Software

アプリケーションノート 機能編

要旨

本書は、Verified Linux Package 上で動作する、RZ/G1 Trusted Secure IP Software について、その機能や使用方法を記載しています。RZ/G 製品上で Verified Linux Package を利用してセキュリティ開発を行う開発者を対象としています。

対象デバイス

RZ/G1M-PF
RZ/G1E-PF
RZ/G1N-PF
RZ/G1C-PF

目次

1. 概要 3
1.1 特徴 4
1.2 提供物 5
1.3 参照ドキュメント 6
1.4 用語 7

2. ソフトウェア構成 8
2.1 ソフトウェア間のインターフェース 9

3. 機能 10
3.1 セキュアストレージ 10
3.2 セキュアソフトウェアアップデート 10
3.3 基本暗号化機能 10
3.4 暗号化カーネルブート 10

4. API 仕様(Security Library) 11

5. API 仕様(Security Driver for Boot/Provisioning) 12

6. 鍵と鍵束 13
6.1 鍵の種類 13
6.2 鍵束 13

7. プロビジョニング処理と暗号化カーネルブート 15
7.1 プロビジョニング処理 15
7.1.1 一時暗号化 16
7.1.2 再暗号化 17

7.2 暗号化カーネルブート 18

8. ユーザ鍵のインジェクション 19
8.1 鍵の使用 19

9. セキュアソフトウェアアップデート 20
9.1 一時暗号化 21
9.2 再暗号化 21

10. 運用と廃棄 22
10.1 安全な運用 22
10.2 製品の廃棄 22

11. プロビジョニングツール 23
11.1 機能 23
11.2 動作環境 23
11.3 ファイル構成 24
11.4 使用方法 26
11.4.1 ツールの起動/終了方法 26
11.4.2 鍵を生成する 27
11.4.3 プロビジョニング用鍵束データを作成する 28
11.4.4 プロビジョニング用/アップデート用 Linux カーネルを作成する 29
11.4.5 更新用鍵束データを作成する 30
11.4.6 鍵データの一時暗号化を行う 31
11.4.7 エラー表示 32

改訂記録 33

1. 概要

本パッケージでは、RZ/G シリーズに内蔵する Trusted Secure IP (以降 TSIP) を利用し、耐タンパ性の高いセキュリティ機能を提供します。

- ✓ セキュアストレージ
- ✓ セキュアソフトウェアアップデート
- ✓ 基本暗号化機能

これらの機能は、Linux のユーザ空間で動作する”Security Daemon”と”Security Library”によって実現します。ユーザアプリケーションは Security Library の API を経由し、Security Daemon が提供するサービスを呼び出して使用します。Security Daemon は”Security Driver”を内包しており、TSIP を使用して各機能を実行します。

- ✓ 鍵束、ユーザデータの再暗号化
- ✓ 鍵束、ユーザデータの復号、検証

これらの機能は、事前に暗号化しておいた鍵束やユーザデータ(uImage/DeviceTree を想定、以降同様)を起動時に復号、検証(暗号化カーネルブート)するために使用します。本ソフトウェアパッケージでは、事前の暗号化に使用する Linux のユーザ空間で動作する”Security Driver for Provisioning(Linux)”と、起動時の復号、検証に使用する non-OS 環境で動作する”Security Driver for Boot(non-OS)”を提供します。TSIP の各機能を使用するためには暗号化カーネルブートを行う必要があります。ブート時に復号、検証に失敗した場合は TSIP を使用することができないため、暗号化したデータをより強固に守ることが可能となります。

1.1　特徴

本ソフトウェアパッケージでは、RZ/G の持つ暗号用 IP である TSIP を使用して各種のセキュリティ機能を実現しています。TSIP は、個々のデバイス固有の鍵(デバイス固有鍵)を持っており、これを使用して各種データの暗号化、復号を行います。TSIP のデバイス固有鍵は外部からアクセスできない仕組みになっており、高い耐タンパ性を持っています。

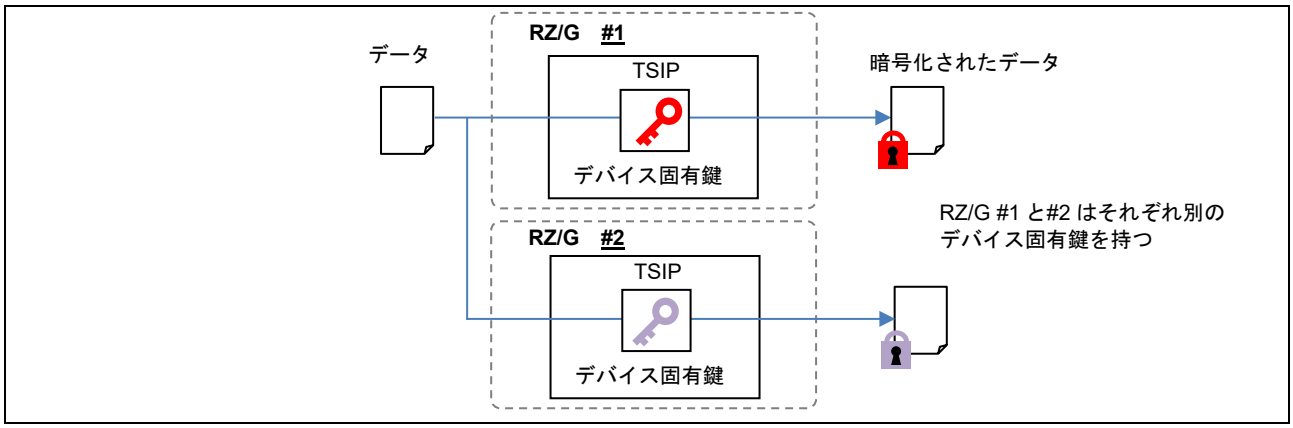

図 1-1　TSIP のデバイス固有鍵

デバイス固有鍵で暗号化されたデータは、暗号化したデバイスでのみ復号が可能です。そのため、Linux カーネルやユーザデータを不揮発性メモリに保存して運用する場合においても、デバイス固有鍵で各データを暗号化しておくことで、安全に運用できます。不揮発性メモリ上の Linux カーネルやユーザデータが別の環境へ不正コピーされた場合、復号することができません。また、同じデバイス上のデータであっても、改ざんされた場合は復号することができないため、ブートプログラムでの改ざん検出が可能となります。

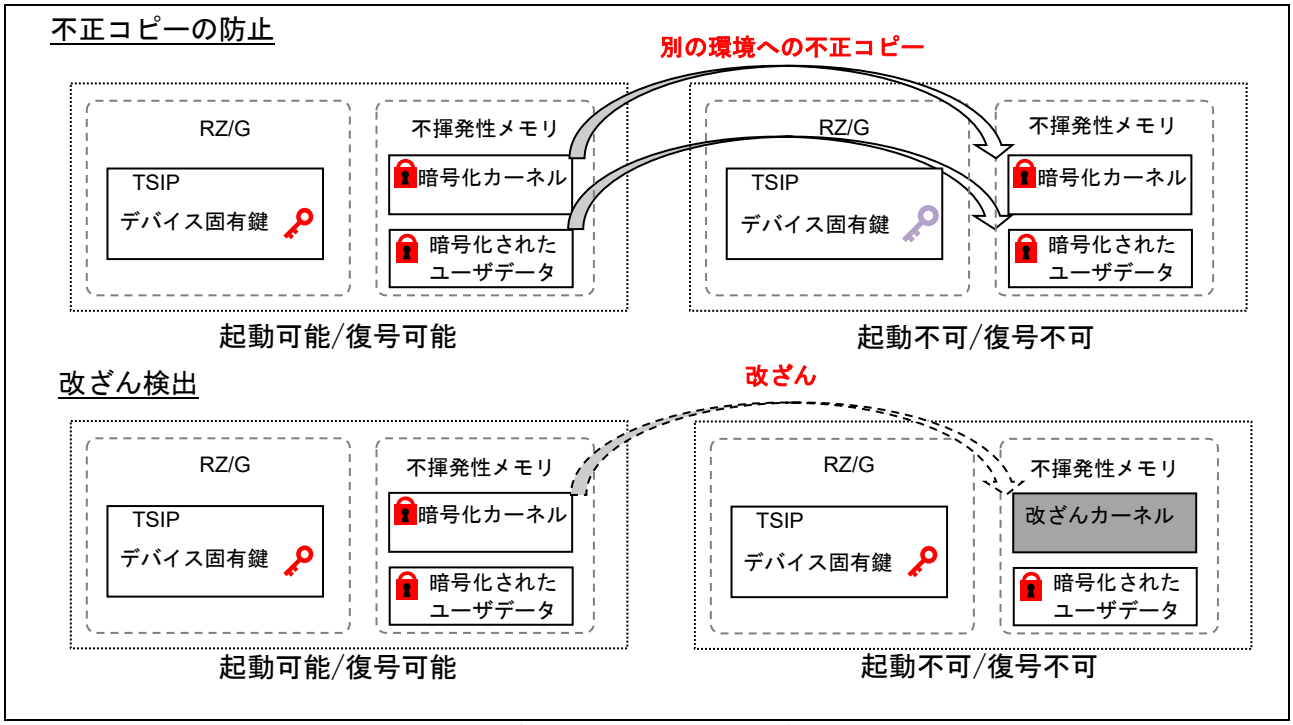

図 1-2　デバイス固有鍵によるデータ保護

1.2 提供物

本パッケージには、以下のバイナリが含まれています。

表 1-1 提供物(バイナリ)

type	name	description
Binary	Security Library	Linux のユーザ空間で動作するアプリケーションです。ユーザアプリケーションとのインターフェースを提供します。Security Daemon と組み合わせて使用します。
	Security Daemon	Linux のユーザ空間で動作するアプリケーションです。Security Library からのリクエストを処理します。内部に Security Driver を含んでおり、TSIP の制御を行います。
	Security Driver for Provisioning (Linux)	暗号化カーネルブート用に、プロビジョニング処理にて事前にデータを TSIP で再暗号化するために使用する、Linux 用のドライバです。(Security Daemon が含んでいる Security Driver とは別のドライバです。Security Library/Security Daemon が持つ各種機能は本ドライバにはありません)
	Security Driver for Boot (non-OS)	暗号化カーネルブート用に、ブート時に鍵束の検証、ユーザデータの復号、検証を行うために使用する、ベアメタル環境用ドライバです。(Security Library/Security Daemon が持つ各種機能は本ドライバにはありません)
	Security Module	Security Driver for Boot(non-OS)を使用した、ブートローダ用プログラムです。 起動時に、プロビジョニング処理にて再暗号化済みの鍵束の検証や、ユーザデータの復号/検証を行うための u-boot アプリケーションです。u-boot から実行し、TSIP を有効にします。

また、環境作成のためのサンプルや、使用時の手順確認用のサンプルとして、以下を用意しています。

表 1-2 提供物(サンプル)

type	name	description
Sample Tool	Provisioning Tool	プロビジョニング処理時に、製品に持ち込むためのデータを一時暗号化するための Windows アプリケーションソースコード/プログラムです。ユーザ鍵の一時暗号化にも使用します。
	Install Tool	プロビジョニング処理時に、一時暗号化したデータを製品上の TSIP を使用して再暗号化するための Linux アプリケーションソースコード/プログラムです。Security Driver for Provisioning(Linux)を使用しています。
Sample Program	Injection sample	Security Library を使用した Linux のサンプルコードです。 ユーザ鍵のインジェクションの手順参考用です。
	Secure Storage Sample	Security Library を使用した Linux のサンプルコードです。 セキュアストレージ機能での暗号化/復号手順参考用です。
	Verification Sample	Security Library を使用した Linux のサンプルコードです。 RSA による署名生成、署名検証の手順参考用です。
	Encrypt/Decrypt Sample	Security Library を使用した Linux のサンプルコードです。 AES によるデータの暗号化、復号の手順参考用です。
	Secure Update Sample	Security Library を使用した Linux のサンプルコードです。 Update 機能による更新用ファイルの作成手順参考用です。

提供物のファイル構成は以下のとおりです。

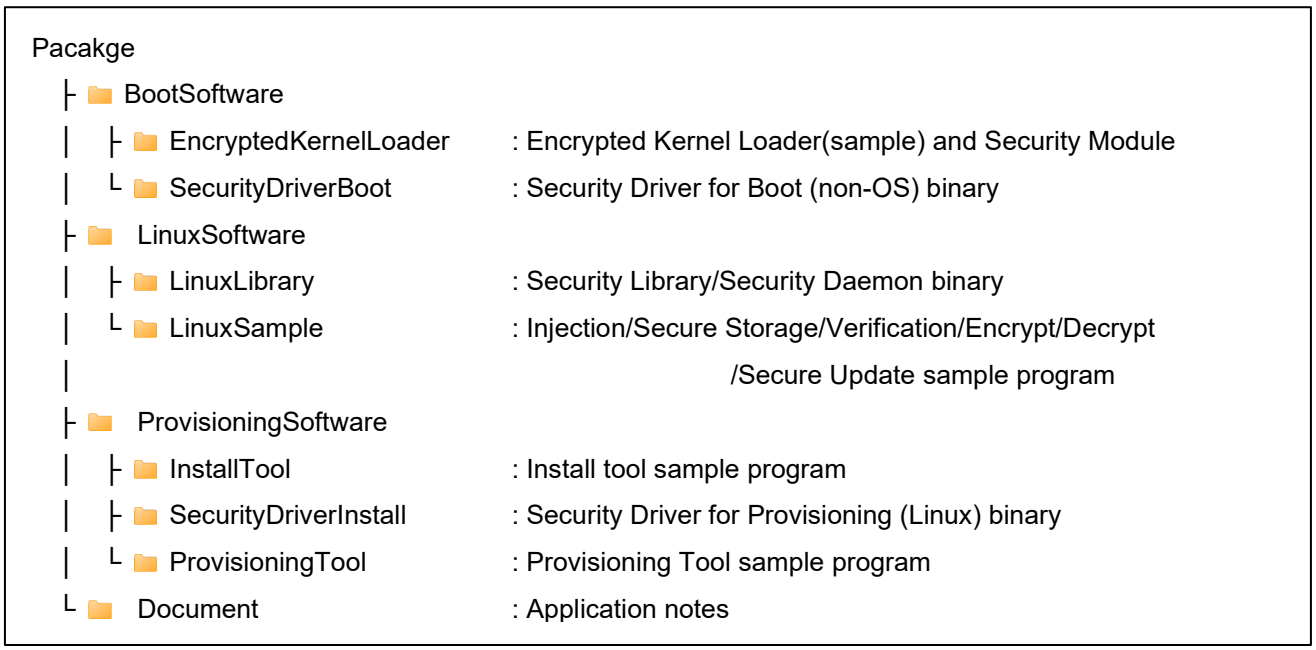

図 1-3 提供物のファイル構成

1.3 参照ドキュメント

RZ/G1 Trusted Secure IP (TSIP) ソフトウェアの関連ドキュメントを以下に示します。

- RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート 機能編 : R01AN5788JJ(本ドキュメント)
  RZ/G1 Trusted Secure IP (TSIP) Software の各機能について説明しています。
- RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート API 編 : R01AN5789JJ
  RZ/G1 Trusted Secure IP (TSIP) Software を利用するためのライブラリ/ドライバの API について説明しています。また、ライブラリ/ドライバを使用したツールやサンプルプログラムについて説明しています。

1.4 用語

name	description
Trusted Secure IP (TSIP)	RZ/G に内蔵されたセキュリティ IP
デバイス固有鍵	TSIP 内部のデバイス固有の鍵。TSIP 外部から読み出すことはできません。
一時暗号化	TSIP で扱う鍵、鍵束、ブート時検証対象データを、TSIP 外部でいったん暗号化すること。
再暗号化	一時暗号化された鍵、鍵束、ブート時検証対象データを、TSIP 内部で復号し、デバイス固有鍵で再度暗号化すること。
プロビジョニング処理	鍵束やユーザデータを一時暗号化し、その後 TSIP で再暗号化し、不揮発性メモリに格納しておく処理。TSIP を使用するためにはプロビジョニング処理を事前に実施する必要があります。
暗号化カーネルブート	事前に TSIP で再暗号化しておいた鍵束やユーザデータの復号、検証をブート時に行い、TSIP の各種暗号化/復号機能を使用できる状態にする処理のこと。
インジェクション	鍵、鍵束の再暗号化のこと。

2. ソフトウェア構成

RZ/G セキュリティは、以下の表のようなソフトウェア構成になります。

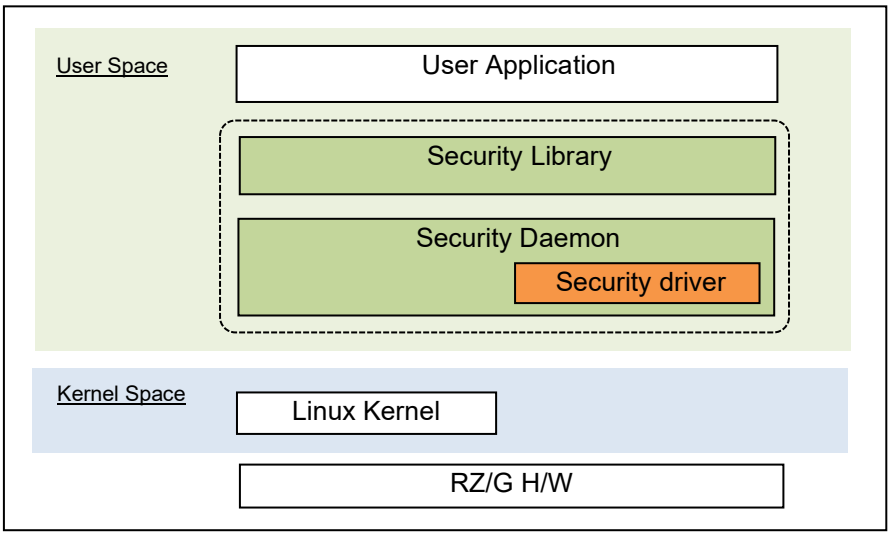

図 2-1　ソフトウェア構成

表 2-1 構成ソフトウェアの一覧

Software Name	Description
Security Library	Application とのインターフェースである API を提供します
Security Daemon	Application からの要求に従って実行するサービスになります
Security Driver	TSIP のデバイスドライバ

2.1　ソフトウェア間のインターフェース

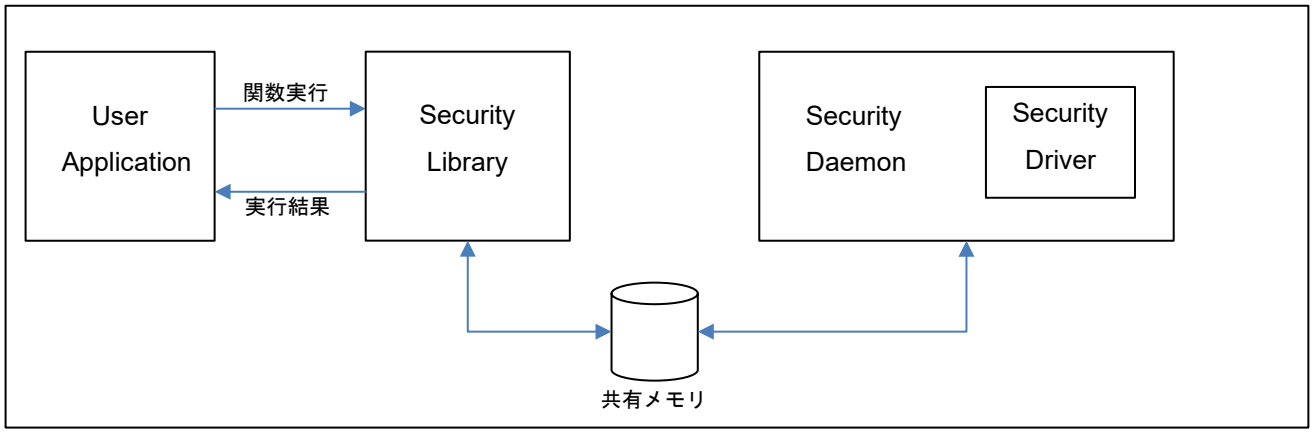

図 2-2　ソフトウェア間インターフェースのイメージ

Application は、Security Library の API 関数を実行して、実行結果を取得します。Security Library では、実行結果の通知方法として、同期方式と非同期方式を選択できます。

同期方式では、API 関数の戻り値として実行結果を通知します。非同期方式では、Security Library 内部でスレッドを生成し、そのスレッドで処理を実行します。API 関数は処理完了を待たずに正常終了します。スレッドでは、処理完了後、引数で指定したコールバック関数を実行して Application に実行結果を通知します。

表 2-2 実行結果の通知方式

方式	実行結果の通知方法
同期方式	API 関数の戻り値
非同期方式	コールバック関数

Security Library と Security Daemon 間の通信では、共有メモリを使用します。Security Library が生成するメモリと Security Daemon が生成するメモリが存在します。Security Library では、API 関数を実行する毎に必要となる共有メモリを生成して、処理が完了した時点で廃棄します。Security Daemon では、起動時に必要な共有メモリを生成します。

3. 機能

3.1 セキュアストレージ

セキュアストレージ機能として、TSIP 内で生成する固有の鍵で、データを保護する機能を API として提供します。TSIP 毎に固有となる鍵を生成するため、保護した RZ/G 製品以外では保護を解除できません。

表 3-1 セキュアストレージ用提供機能

暗号化機能	TSIP 内部の固有の鍵を使用したユーザデータの暗号化
復号機能	TSIP 内部の固有の鍵を使用したユーザデータの復号

3.2 セキュアソフトウェアアップデート

セキュアソフトウェアアップデートでは、ブート時の復号、検証を行う対象、鍵束データの更新用データを作成します。

表 3-2 セキュアソフトウェアアップデート用提供機能

鍵束の更新	鍵束の更新データを作成します
起動時検証データの更新	起動時検証データの更新データを作成します

3.3 基本暗号化機能

基本暗号化機能として、以下の表に示す TSIP が持つ暗号化機能を API として提供します。

表 3-3 基本暗号化機能提供アルゴリズム

対称暗号	AES CBC (128bits, 256bits)
非対称暗号	RSA (1024bits, 2048bits)
ハッシュアルゴリズム	SHA-1, SHA-256
MAC	HMAC (SHA-1, SHA-256) CMAC (AES-128, AES-256)

3.4 暗号化カーネルブート

暗号化カーネルブートでは、TSIP を使用して、事前に署名暗号化したデータの復号検証を行います。暗号化カーネルブート用として、以下の機能を持つドライバを提供します。

表 3-4 暗号化カーネルブート用提供機能

鍵束の再暗号化	一時暗号化された鍵束を TSIP のもつ固有の鍵で再暗号化します
ユーザデータの再暗号化	一時暗号化されたユーザデータを TSIP のもつ固有の鍵で再暗号化します
鍵束の検証	再暗号化された鍵束を検証し、TSIP を有効にします
ユーザデータの復号、検証	再暗号化されたユーザデータを復号、検証し、TSIP を有効にします

4. API 仕様(Security Library)

Security Library がユーザに提供する API 関数の一覧を以下に示します。

表 4-1 Security Library 提供 API 一覧

No	Function Name	Description
1	SEC_Initialize	Security Library と Security Daemon の初期化処理を行います
2	SEC_Finalize	Security Library と Security Daemon の終了処理を行います
3	SEC_ContentsKeyConvert	暗号化/復号で用いる鍵データの再暗号化を行います
4	SEC_ContentsKeyGenerate	暗号化/復号や署名付加/検証に必要な鍵データを生成します
5	SEC_ContentsDataEnc	再暗号化した鍵データを用いて、任意のデータを暗号化します
6	SEC_ContentsDataDec	再暗号化した鍵データを用いて、暗号化したデータを復号します
7	SEC_ContentsDataMakeVerifyCode	再暗号化した鍵データを用いて、任意のデータの検証用コードを計算・生成します
8	SEC_ContentsDataVerify	再暗号化した鍵データと検証用コードを用いて、データを検証します
9	SEC_SecureStorageEnc	デバイス固有の鍵を用いてデータを暗号化します
10	SEC_SecureStorageDec	デバイス固有の鍵を用いてデータを復号化します
11	SEC_Kernel_Update	ブート時に復号検証するカーネルイメージの更新データを作成します
12	SEC_DevTree_Update	ブート時に復号検証するデバイスツリーの更新データを作成します
13	SEC_Key_Update_	鍵束の更新データを作成します
14	SEC_GetPackageVersion	バージョンを取得します

API の詳細については、「RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート API 編」を参照してください。

5. API 仕様(Security Driver for Boot/Provisioning)

暗号化カーネルブートで使用する Security Driver for Boot/Provisioning が提供する API 関数の一覧を以下に示します。なお、Security Driver for Boot および Security Driver for Provisioning の提供する機能は同じです。

表 5-1 Security Driver for Boot/Provisioning 提供 API 一覧

No	Function Name	Description
1	R_TSIP_Init	TSIP を初期化します
2	R_TSIP_Inject_Key	一時暗号化された鍵束を TSIP で再暗号化します
3	R_TSIP_Install_uImage	一時暗号化されたカーネル(uImage)を TSIP で再暗号化します
4	R_TSIP_Install_DeviceTree	一時暗号化された Device Tree を TSIP で再暗号化します
5	R_TSIP_SB_uImage	TSIP で再暗号化されたカーネル(uImage)の復号、検証を行います。
6	R_TSIP_SB_DeviceTree	TSIP で再暗号化された Device Tree の復号、検証を行います。

API の詳細については、「RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート API 編」を参照してください。

6. 鍵と鍵束

6.1 鍵の種類

本ソフトウェアパッケージでは、大きく分けて、以下の鍵を使用します。

表 6-1 Trusted Secure IP (TSIP) Software で使用する鍵

名称	説明
Hidden Root Key	ユーザには提供されない、Renesas のみが管理する鍵
Provisioning Key	鍵束の一時暗号化、メッセージ認証に使用します。二つの鍵をバイナリ連結したものです。 (Provisioning Key = 鍵束用一時暗号化鍵 || 鍵束用 MAC 鍵)※
EncProvisioning Key	Provisioning Key を Hidden Root Key で暗号化したものです。TSIP が Provisioning Key で一時暗号化された鍵束を復号するために使用します。
鍵束	ユーザが製品で利用する鍵の束(表 6-3 鍵束のフォーマット参照) 事前に TSIP で再暗号化して保持しておきます。
ユーザ鍵	各暗号処理に使用するユーザ鍵。

※“||”はバイナリ連結を表しています。

6.2 鍵束

RZ/G に内蔵された TSIP では、用途毎に別々の鍵を使用します。用途毎の鍵はユーザが作成します。本ソフトウェアパッケージでは、この用途毎のユーザ鍵をまとめて鍵束として管理します。鍵束は、各鍵や IV(Initialization Vector)をバイナリ結合したものです。

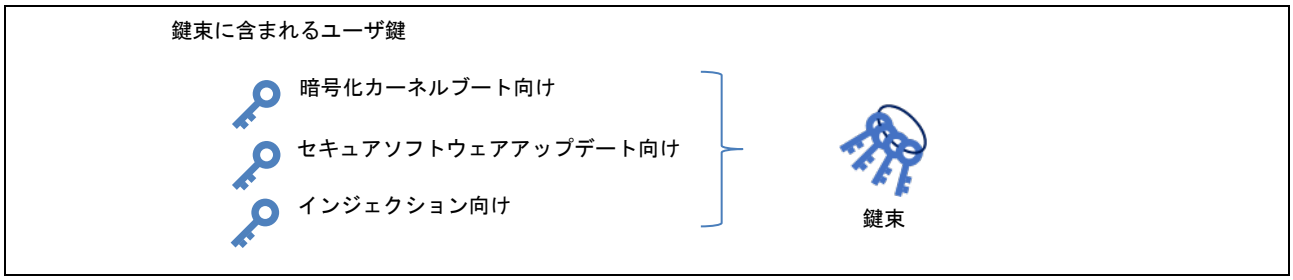

図 6-1 鍵束のイメージ

表 6-2 鍵束に含まれる鍵

鍵束に含まれるユーザ鍵	説明
暗号化カーネルブート向け	boot 時に復号/検証する対象データの復号、署名検証に使用します
セキュアソフトウェアアップデート向け	boot 時に復号/検証するデータ、および鍵束のアップデート時に鍵束の再暗号化、MAC 値の生成、MAC 値の検証に使用します
インジェクション向け	ユーザ鍵のインジェクション時に、鍵の再暗号化、MAC 値の生成、MAC 値の検証に使用します

鍵束のフォーマットを以下に示します。鍵束に含まれる各鍵データとして、ユニークな鍵を用意する必要があります。鍵束は「プロビジョニングツール」で作成することができます。

表 6-3 鍵束のフォーマット

用途		種類	Size (Byte)
暗号化 カーネルブート向け	uImage 一時暗号化鍵	AES-128 鍵	16
		IV	16
	DeviceTree 一時暗号化鍵	AES-128 鍵	16
		IV	16
	uImage/DeviceTree 一時署名検証鍵※	PublicKey(n)	256
		PublicKey(0^15 Padding || e || 0^96 Padding)	16
Reserved		(0 固定)	272
セキュアソフトウェア アップデート向け	鍵束用一時暗号化鍵	AES-128 鍵	16
	鍵束用 MAC 鍵	AES-128 鍵	16
インジェクション向け	ユーザ鍵用 一時暗号化鍵	AES-128 鍵	16
	ユーザ鍵用 MAC 鍵	AES-128 鍵	16

note※ ： ペアとなる秘密鍵(ユーザデーター時署名生成鍵)がプロビジョニング処理にてユーザデータの署名に使用されます。

7. プロビジョニング処理と暗号化カーネルブート

TSIP の各種暗号化/復号機能を使用するためには、暗号化カーネルブートの実施が必要です。暗号化カーネルブートを行うためには、事前にプロビジョニング処理を行う必要があります。鍵束やユーザデータを TSIP で再暗号化し、不揮発性メモリに格納しておく処理をプロビジョニング処理と呼びます。事前に再暗号化しておいた鍵束やユーザデータの復号、検証を行い、TSIP の各種暗号化/復号機能を使用できる状態にする処理を暗号化カーネルブートと呼びます。

7.1 プロビジョニング処理

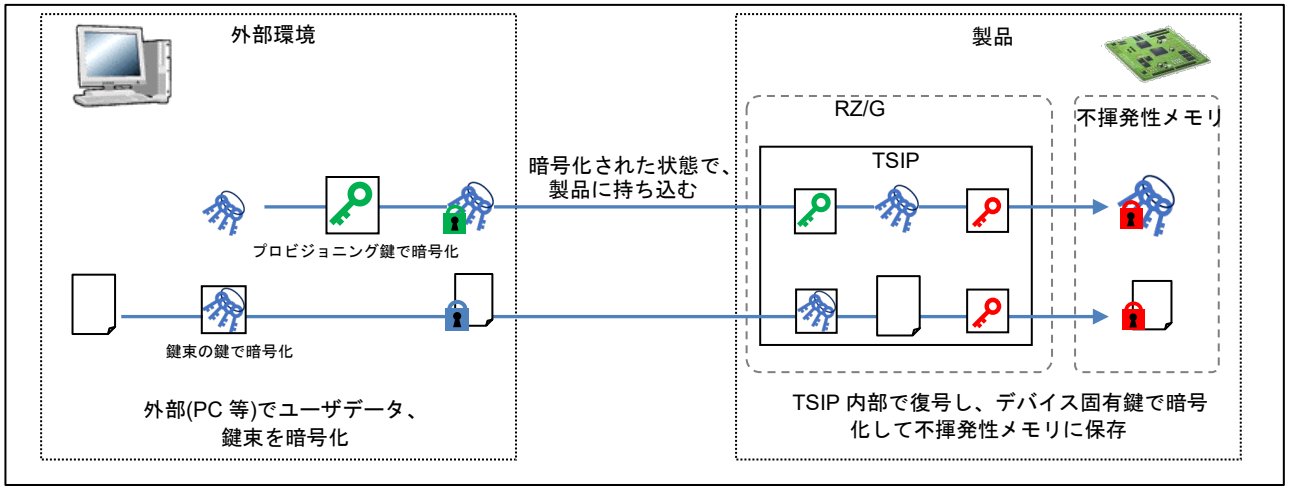

図 7-1 プロビジョニング処理のイメージ

プロビジョニング処理では、以下を行います。

1. 外部で、起動時に復号検証したいユーザデータや鍵束を暗号化(一時暗号化)し、暗号化された状態で各データを製品に持ち込む。
2. 製品上の RZ/G に内蔵された TSIP で復号し、デバイス固有鍵で暗号化(再暗号化)し、不揮発性メモリへと保存する。

鍵束や起動時に復号検証するユーザデータを製品に持ち込む際には一時暗号化した状態で行い、一時暗号化されたデータは TSIP 内部で復号し、再暗号化されたデータが TSIP 外部に出力されます。これにより、製品にデータを持ち込む際にも情報漏洩を防ぐことが可能となっています。

7.1.1　一時暗号化

プロビジョニング処理で使用する鍵と、製品に持ち込むためのデータや鍵束の一時暗号化について説明します。以下の手順で、製品に持ち込むための各データを作成できます。

1)　Hidden Root Key を使用して、Provisioning Key を暗号化します。本工程はルネサスにて行い、ユーザには暗号化された EncProvisioningKey が提供されます。EncProvisioningKey の提供については、「RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート API 編」を参照ください。

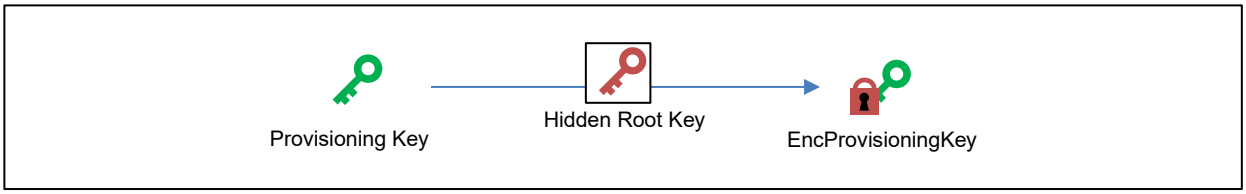

図 7-2 Provisioning Key の暗号化のイメージ

2)　ユーザ鍵を用意し、鍵束の作成と暗号化を行います。本工程はサンプル提供している「プロビジョニングツール」で実施できます。鍵束の暗号化には Provisioning Key(=Provisioning Key1(鍵束用一時暗号化鍵) || Provisioning Key2(鍵束用 MAC 鍵))を使用します。

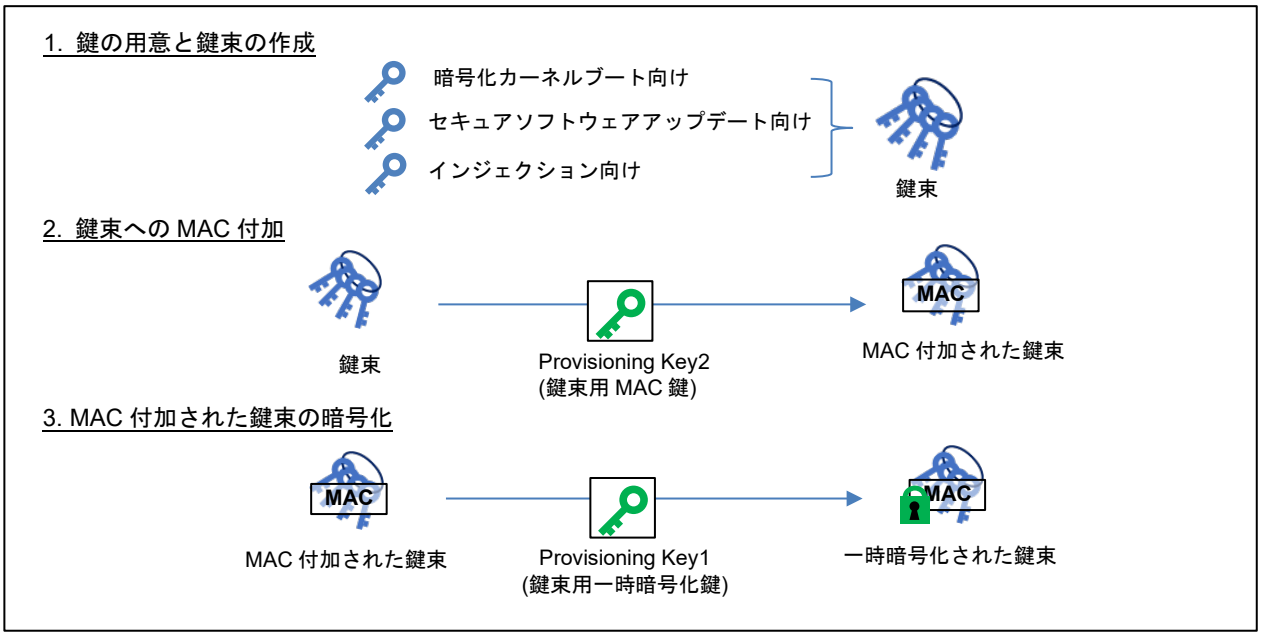

図 7-3 鍵束の一時暗号化のイメージ

使用する暗号アルゴリズムは以下のとおりです。

表 7-1 鍵束一時暗号化処理方式

処理	暗号方式	鍵	IV
MAC	CBC-MAC with AES-128	Provisioning Key2 (鍵束用 MAC 鍵)	0
暗号化	AES-128-CBC	Provisioning Key1 (鍵束用一時暗号化鍵)	IV0※

※ IV0=0x85c1673483d5d291f0d0713e3ea434a3

3)暗号化カーネルブート向け鍵を使用して、ユーザデータ(uImage/DeviceTree)に対して署名付加と暗号化を行います。本工程はサンプル提供している「プロビジョニングツール」にて実施できます。

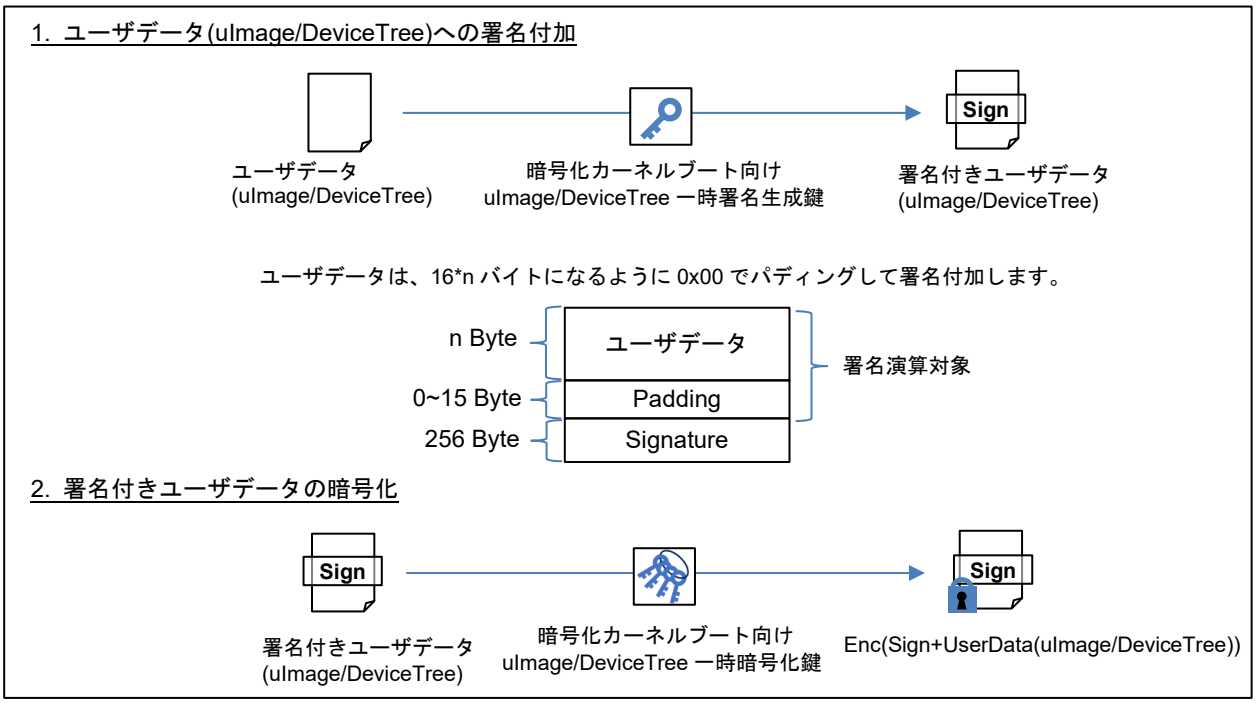

図 7-4 ユーザデータの一時暗号化処理のイメージ

使用する暗号アルゴリズムは以下のとおりです。

図 7-5 ブート時復号検証対象ユーザデーター時暗号化処理方式

処理	暗号方式	鍵	IV
署名	RSASSA-PKCS1-v1_5 SHA256	uImage/DeviceTree 一時署名生成鍵	0
暗号化	AES-128-CBC	uImage/DeviceTree 一時暗号化鍵	IV※

※ 暗号化カーネルブート向けユーザデーター時暗号化鍵の IV

uImage、DeviceTree それぞれに対して、上記作業を行います。

以上で、製品に事前に持ち込むデータの作成が完了します。

7.1.2 再暗号化

一時暗号化した鍵束やユーザデータを製品に持ち込み、TSIP で再暗号化します。各データは暗号化(一時暗号化)された状態で持ち込むため、安全な運用が可能です。TSIP での再暗号化には Security Driver for Provisioning(Linux)を使用します。再暗号化したデータを不揮発性メモリに保存しておき、暗号化カーネルブートで Security Driver for Boot (non-OS)を使用して復号、検証します。

7.2　暗号化カーネルブート

暗号化カーネルブートは、プロビジョニング処理で製品に持ち込まれた鍵束、ユーザデータの復号、検証を行い、TSIP の各種暗号化/復号機能を使用できる状態にする処理です。検証、復号に成功しない限り、TSIP の各種暗号化/復号機能を使用することはできません。

ブート時の鍵束、ユーザデータの復号、検証には Security Driver for Boot (non-OS)を使用します。Security Driver for Boot(non-OS)の詳細については、「RZ/G1 Trusted Secure IP (TSIP) Software アプリケーションノート API 編」を参照してください。本パッケージで提供するサンプルでは、Security Driver for Boot を使用した Security Module を使用して、u-boot 上でカーネルデータの復号検証を行っています。

8. ユーザ鍵のインジェクション

暗号化カーネルブート後は、TSIP の各種暗号化/復号機能を使用できます。TSIP でユーザ鍵を使用する場合、ユーザ鍵のインジェクションが必要です。ユーザ鍵のインジェクションとは、外部で一時暗号化されたユーザ鍵を、デバイス固有鍵で再暗号化する処理を指します。TSIP を使用した各種暗号化/復号機能を利用するためには、インジェクションされた鍵を使用する必要があります。

ユーザ鍵は、一時暗号化された状態で製品に持ち込むことになるため、鍵の安全な運用が可能です。インジェクション後の鍵は TSIP のデバイス固有鍵で暗号化された状態で運用され、鍵データ自体は TSIP 内部でのみ使用されるため、流出、漏洩することがありません。

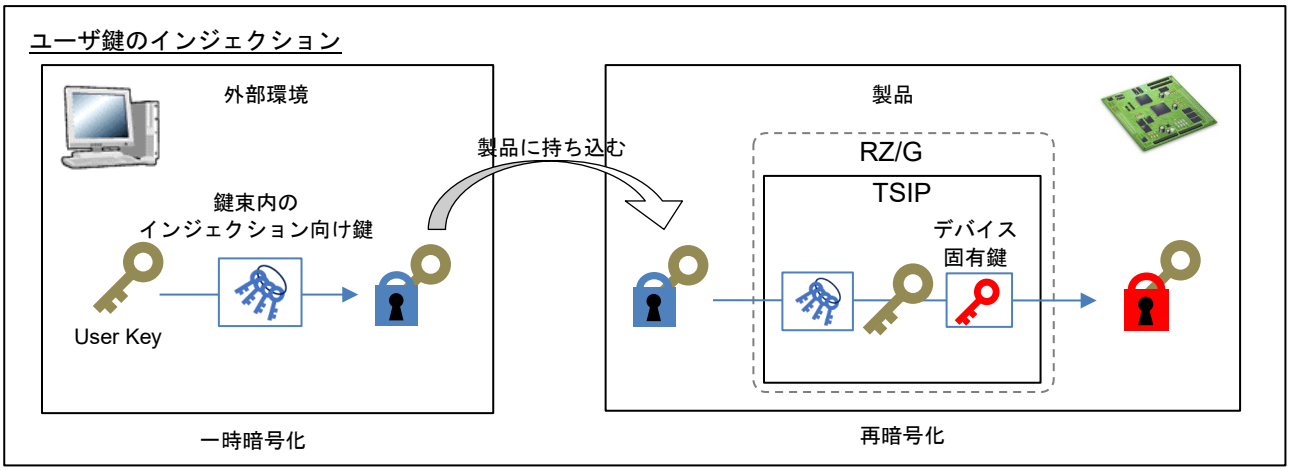

図 8-1 ユーザ鍵のインジェクション処理のイメージ

一時暗号化には、鍵束に含めたインジェクション向け鍵(ユーザ鍵用一時暗号化鍵、ユーザ鍵用 MAC 鍵)を使用します。ユーザ鍵用 MAC 鍵で MAC 付加し、ユーザ鍵用一時暗号化鍵で暗号化します。

図 8-2 ユーザ鍵インジェクション時一時暗号化のアルゴリズム

処理	暗号方式	鍵	IV
MAC 処理	CBC-MAC with AES-128	ユーザ鍵用 MAC 鍵	0
一時暗号化処理	AES-128-CBC	ユーザ鍵用一時暗号化鍵	IV※

※ IV はプロビジョニングツールが自動的に付加します。

8.1 鍵の使用

インジェクションしたユーザ鍵を使用して、TSIP の各種暗号化/復号処理を使用できます。TSIP では、インジェクションしていない鍵は使用できません。インジェクションした鍵を使用して TSIP の各種暗号化/復号処理を行うには、Security Library を使用します。

9. セキュアソフトウェアアップデート

製品運用開始後に、プロビジョニング処理にて再暗号化した鍵束、ユーザデータ(uImage/DeviceTree)のアップデートを行うために、それぞれを更新するためのデータを作成することができます。更新する鍵束、ユーザデータは外部環境において一時暗号化した状態で製品に持ち込み、TSIP で再暗号化して出力します。出力されたデータを既存のデータと差し替えて更新します。新しい鍵束やユーザデータは暗号化された状態で製品に持ち込み、TSIP 内部で再暗号化されるため、安全に鍵束、ユーザデータの更新が可能です。

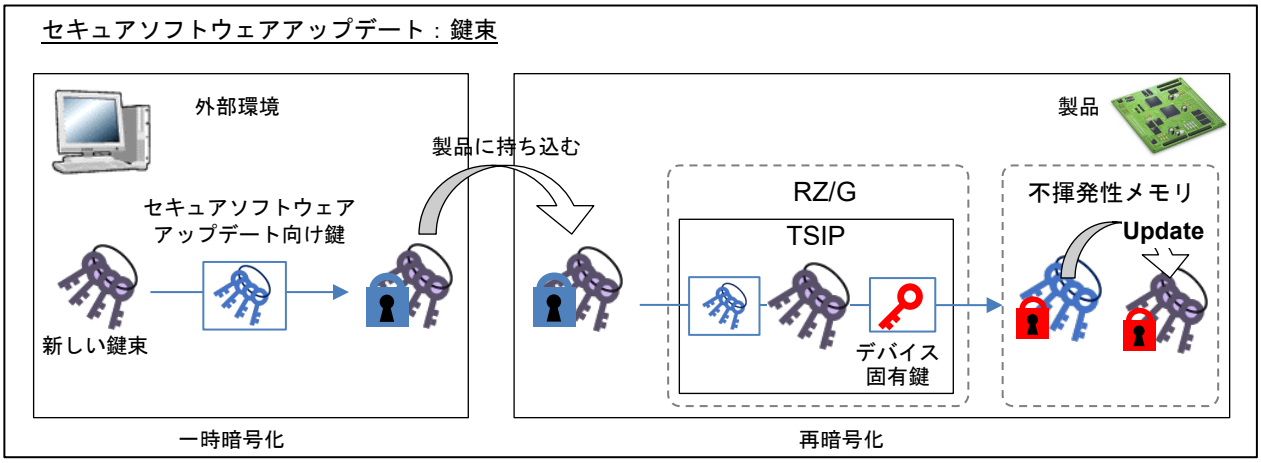

図 9-1 鍵束のアップデート処理のイメージ

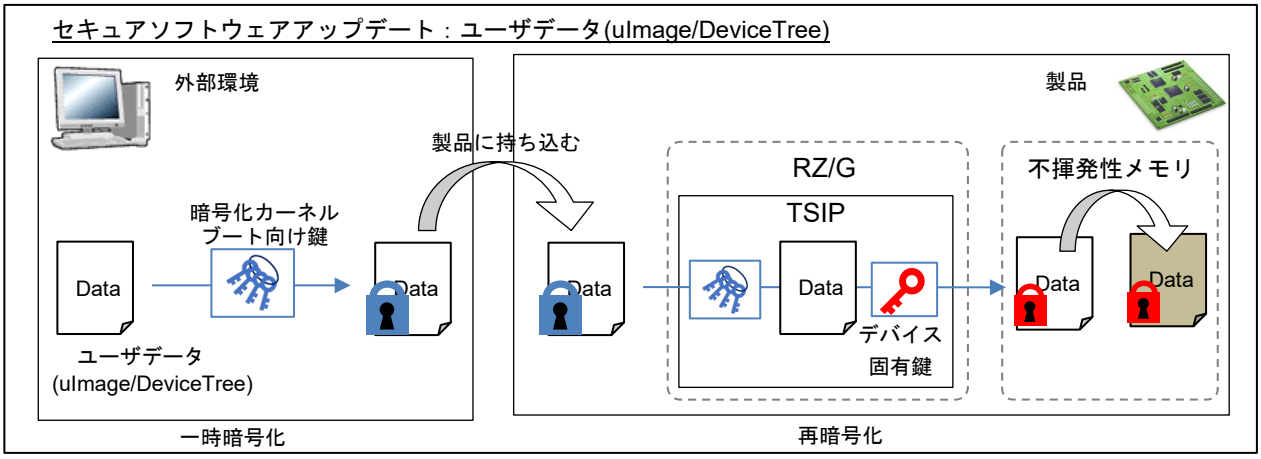

図 9-2 ブート時検証対象ユーザデータ(uImage/DeviceTree)のアップデート処理のイメージ

9.1　一時暗号化

新しい鍵束の一時暗号化には、すでにプロビジョニング済みの鍵束内のセキュアソフトウェアアップデート向け鍵を使用します。プロビジョニング処理での鍵束の MAC 付加、暗号化と同様に、一時暗号化データを作成します。

新しいユーザデータの一時暗号化には、鍵束内の暗号化カーネルブート向け鍵を使用します。プロビジョニング処理での署名付加、暗号化と同様に一時暗号化データを作成します。このとき、再暗号化時に、先に鍵束の再暗号化を実施する場合は、新しい鍵束内の暗号化カーネルブート向け鍵を使用する必要があることにご注意ください。

9.2　再暗号化

一時暗号化された新しい鍵束、ユーザデータは、製品に持ち込まれた後、TSIP 内で再暗号化します。再暗号化処理は、Security Library を使用して実施します。

10. 運用と廃棄

10.1 安全な運用

本ソフトウェアパッケージでは、平文としての鍵はすべて TSIP 内部でのみ使用しており、TSIP 内部へ入力する際および TSIP 外部へ出力する際は、暗号化および MAC 処理を実施しています。そのため、製品運用時においてユーザ鍵漏洩の危険性を最低限にすることが可能ですが、一時暗号化を行う段階では、平文で扱うユーザ鍵/起動時検証対象データ/鍵束/プロビジョニング鍵は安全な環境で扱う必要があります。安全な環境で一時暗号化したデータを作成し、それらを含む ROM データとして量産拠点へ提供し、ユーザ量産拠点などで再暗号化することで、必要とされる安全な環境の範囲を最小限にしたまま、製品固有の鍵で暗号化された状態のデータを作成することができます。

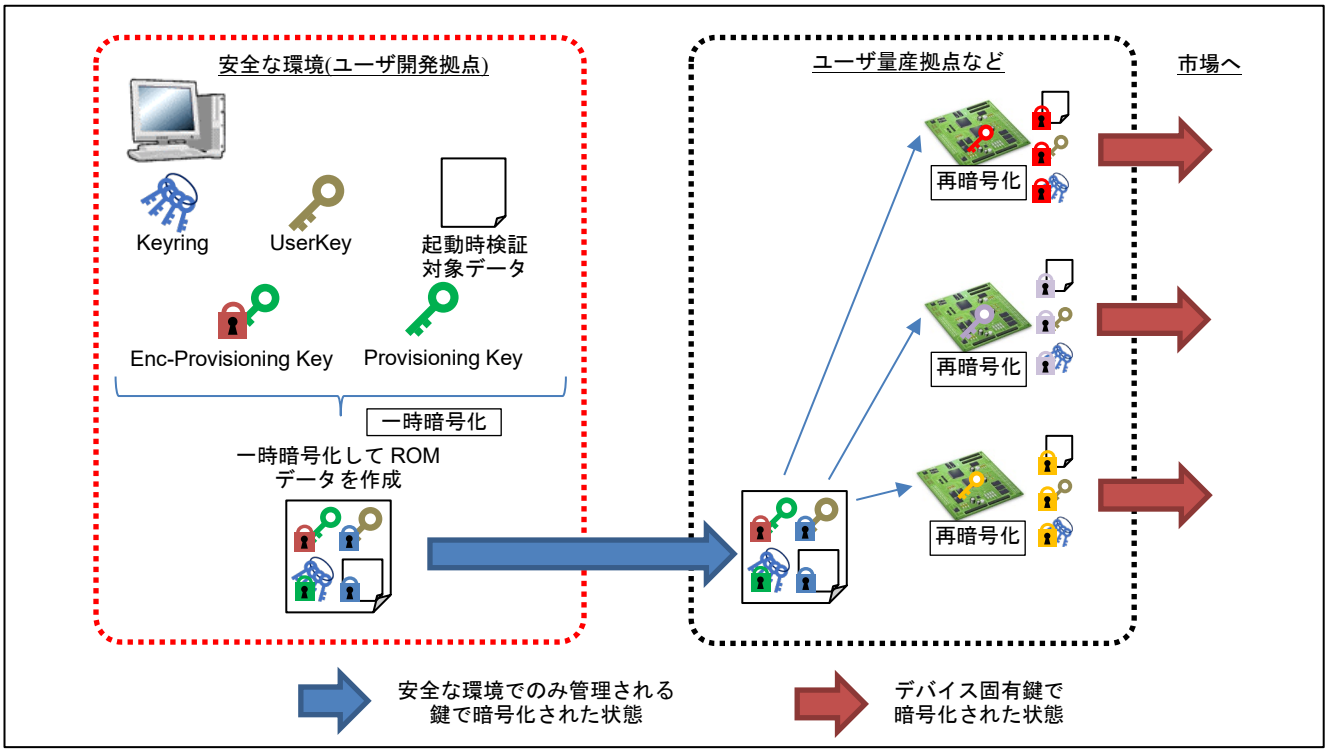

図 10-1 安全な製品出荷

10.2 製品の廃棄

製品の使用用途によっては、製品廃棄時にユーザの秘匿データを漏れない状態にする必要があります。本ソフトウェアパッケージにおいては、プロビジョニング処理を実施した鍵束を廃棄し、暗号化したデータが復号できない状態にすることで、安全な廃棄が可能となります。プロビジョニング後の暗号化された鍵束を格納している不揮発メモリの領域を 0 クリア、固定値で上書きする等、鍵束が使用できない状態にしてください。鍵束データを 0 クリアすることで、暗号化カーネルブートに失敗し、TSIP の復号機能は使用できなくなるため、暗号化されたデータが復号されることはありません。

11. プロビジョニングツール

プロビジョニングツールは、RZ/G1 Trusted Secure IP (TSIP) Software のプロビジョニング処理に使用する各種データを一時暗号化するための Windows アプリケーションです。乱数による鍵の作成も行うことができます。本パッケージにはプロビジョニングツールのソースコードが含まれています。

11.1 機能

本ツールでは、以下の機能を提供します。

鍵束の作成と一時暗号化
起動時検証/復号対象のユーザデータ(uImage/DeviceTree)の一時暗号化
更新用鍵束の作成と一時暗号化
更新用起動時検証/復号対象のユーザデータ(uImage/DeviceTree)の一時暗号化
ユーザ鍵の一時暗号化

11.2 動作環境

本ツールは、以下の環境で動作実績があります。

表 11-1 プロビジョニングツール動作環境

項目	内容
対応 OS	Windows 10 (64bit)
使用するライブラリ	Microsoft .net Framework 3.5

なお、鍵の生成のための乱数ジェネレータとして、.NET Framework の RNGCryptoServiceProvider クラスの GetBytes メソッドを使用しています。

11.3 ファイル構成

本ツールのファイル構成を以下に示します。

図 11-1 ファイル構成

※1　Output 内のフォルダはツール実行後に自動生成されます。鍵生成ツールの出力結果は、フォルダ名の最後に”_KEY”が付きます。暗号化カーネルブート用データ作成ツールの出力結果は、フォルダ名の最後に”_ENC”が付きます。署名付加の出力結果は、フォルダ名の最後に”_CERT”が付きます。鍵データのインジェクション処理の出力結果は、フォルダ名の最後に、”_INJ”が付きます。

※2　Input 内には、あらかじめサンプルの各 Key を評価用として配置しています。

※3　ProvisioningKey フォルダには、プロビジョニング鍵を配置してください。本パッケージには評価に使用できる ProvisioningKey サンプルが含まれています。

ツール内のフォルダ名、ファイル名は固定になります。本ツールの生成機能以外で作成した鍵やデータを Input データとして利用する場合は、リネームして配置してください。Input フォルダ内の各入力ファイルは以下のとおりです。

図 11-2 Input フォルダ内の各ファイル

フォルダ/ファイル	説明	種類
ProvisioningKey	プロビジョニング鍵を配置します。	
C1-Key.bin	鍵束用一時暗号化鍵	AES-128
C2-Key.bin	鍵束用 MAC 鍵	AES-128
ProvisioningKey	鍵束用一時暗号化鍵 || 鍵束用 MAC 鍵 ProvisioningKey がある場合は、C1-Key.bin、C2-Key.bin はツール内部で自動的に作成されます。	
EncProvisioningKey	ProvisioningKey を Hidden Root Key にて暗号化したもの。 -	
UpdateKey	鍵束をアップデートするときに使用する鍵。インジェクション済みの鍵束内の、セキュアソフトウェアアップデート用鍵(Up1-Key.bin、Up2-Key.bin)をリネームして配置する必要があります。	
SS-Up1-Key.bin	鍵束用一時暗号化鍵	AES-128
SS-Up2-Key.bin	鍵束用 MAC 鍵	AES-128
KeyRing	鍵束に含まれる各鍵	
EncryptedKernelBooting	暗号化カーネルブート向けの鍵	
E1-Key.bin	uImage 用一時暗号化鍵	AES-128 || IV
E2-Key.bin	DeviceTree 用一時暗号化鍵	AES-128 || IV
SBS-Key.der	uImage/DeviceTree 用一時署名検証鍵	RSA 204 PublicKey※
EncryptedCommunications	(Reserved)	
RMP-Key.der	(Reserved)	-
RMS-Key.der	(Reserved)	-
UpdatingOfKeyring	セキュアソフトウェアアップデート向けの鍵	
Up1-Key.bin	鍵束用一時暗号化鍵	AES-128
Up2-Key.bin	鍵束用 MAC 鍵	AES-128
InjectionKeys	インジェクション向けの鍵	
BCF1-Key.bin	ユーザ鍵用一時暗号化鍵	AES-128
BCF2-Key.bin	ユーザ鍵用 MAC 鍵	AES-128
Injection	ユーザ鍵をインジェクションするときに対象となる鍵	
AES128-Key.bin	ユーザ鍵	AES-128
AES256-Key.bin	ユーザ鍵	AES-256
HMAC-SHA1-Key.bin	ユーザ鍵	HMAC-SHA1
HMAC-SHA256-Key.bin	ユーザ鍵	HMAC-SHA256
RSA1024-Private-Key.der	ユーザ鍵	RSA1024 PrivateKey-CRT-※
RSA1024-Public-Key.der	ユーザ鍵	RSA1024 PublicKey※
RSA2048-Private-Key.der	ユーザ鍵	RSA2048 PrivateKey-CRT-※
RSA2048-Public-Key.der	ユーザ鍵	RSA2048 PublicKey※
LinuxKernel	暗号化カーネルブートの Linux カーネル	
uImage	Linux カーネル (uImage)	-
DeviceTree	Linux カーネル (DeviceTree)	-

※ der 形式

11.4 使用方法

11.4.1 ツールの起動/終了方法

ProvisioningTool.exe をダブルクリックで実行すると、以下のツール画面が表示されます。

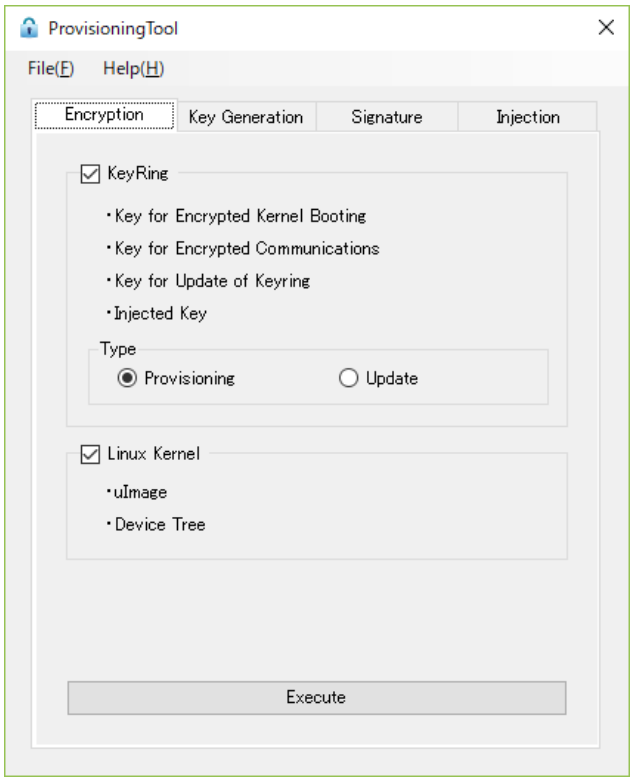

図 11-3 起動画面

終了時は、ツール右上の”×”をクリックする、または、メニューから”File”→”Exit”を選択します。

11.4.2 鍵を生成する

Input フォルダ内の鍵として使用できる各鍵を作成します。

暗号化カーネルブートに必要な鍵を生成する手順は以下になります。

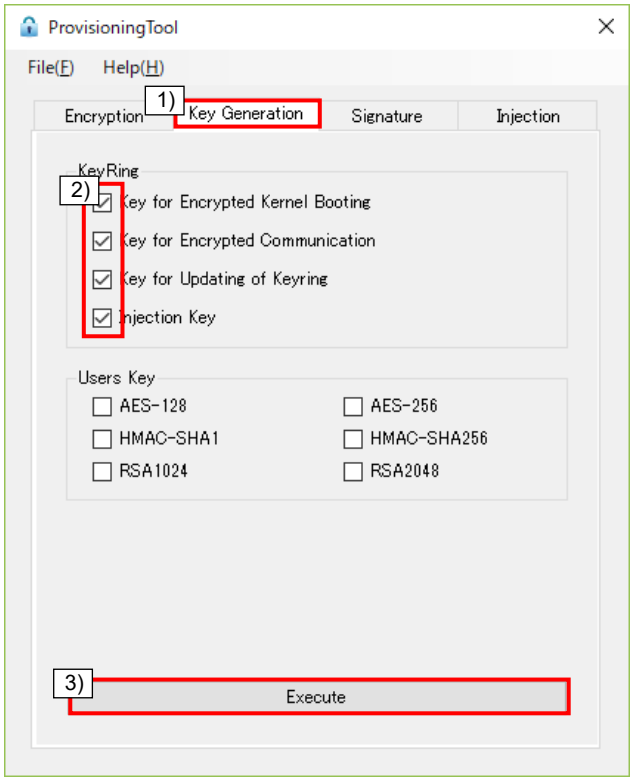

図 11-4 鍵生成のツール画面

- 1)　“Key Generation”タブを選択する
- 2)　生成する鍵のチェックボックスを選択する
- 暗号化カーネルブートでは、“Keys for Encrypted Kernel Booting”にチェックします。
- 3)　“Key Generate”ボタンを押下する

“Keys for Encrypted Communication”、“Keys for Updating of Keyring”、“Injection Keys”のチェックボックスを選択しておくと、暗号化通信、セキュアソフトウェアアップデート、基本暗号化の各機能を使用するためのユーザ鍵を同時に作成することができます。Users Key の各チェックボックスを選択しておくと、各暗号アルゴリズムの鍵を同時に作成することができます。ここで作成した鍵は、Input フォルダに配置して使用します(図 11-1 ファイル構成参照)

11.4.3 プロビジョニング用鍵束データを作成する

KeyRing フォルダ内の各鍵から、鍵束を作成し、プロビジョニング処理で使用できるように暗号化します。ProvisioningKey フォルダ内の鍵と、KeyRing フォルダ内の鍵が使用されます。Keyring フォルダ内の各鍵をバイナリ連結し、鍵束のフォーマットに従って鍵束データを作成します。ProvisioningKey がフォルダ内の ProvisioningKey を C1-Key.bin および C2-Key.bin に分解(ProvisioningKey=C1-Key.bin || C2-Key.bin)します。鍵束に対して C2-Key.bin で MAC 値が生成され、鍵束データに MAC 値を付加して C1-Key.bin で暗号化します。

処理	暗号方式	鍵	IV
MAC 演算	CBC-MAC with AES-128	鍵束用 MAC 鍵(C2-Key.bin)	0
暗号化	AES-128-CBC	鍵束用一時暗号化鍵(C1-Key.bin)	IV0※

※ IV0=0x85c1673483d5d291f0d0713e3ea434a3

なお、鍵束データには、SBS-Key.der から作成された公開鍵が含まれます。また、RMS-Key.der は鍵束には含まれません。

プロビジョニング処理に必要となる鍵束データを作成するための手順は以下になります。

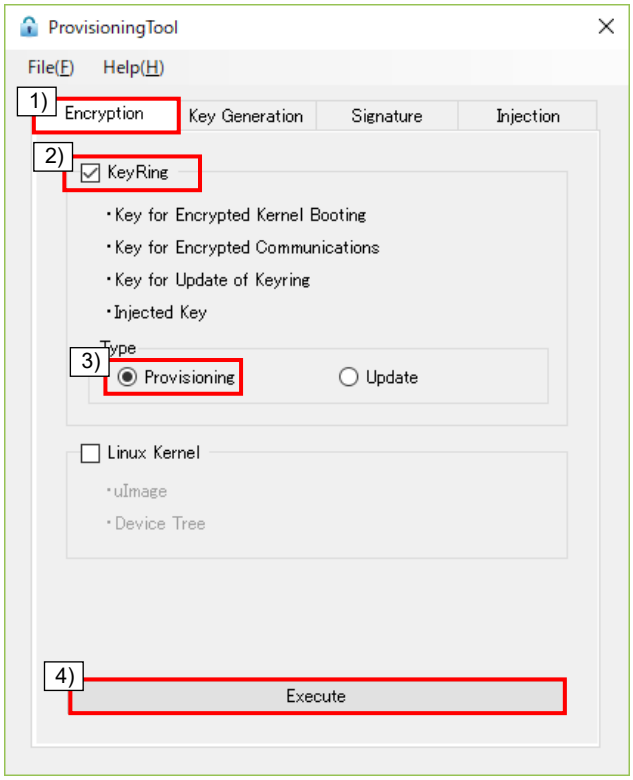

図 11-5 鍵束データのツール画面

- 1) “Encryption”タブを選択する
- 2) “KeyRing”のチェックボックスを選択する
- 3) “Type”の”Provisioning”を選択する
- 4) “Execute”ボタンを押下する

11.4.4 プロビジョニング用/アップデート用 Linux カーネルを作成する

Linux カーネル(uImage/DeviceTree)をプロビジョニング処理、セキュアソフトウェアアップデート処理用に一時暗号化して出力します。LinuxKernel フォルダ内の uImage および DeviceTree のサイズが 16*n バイトになるように 0x00 でパディングします。パディングされたそれぞれのデータに対して SBS-Key.der で署名を行い、署名付き uImage、署名付き DeviceTree をそれぞれ、E1-Key.bin、E2-Key.bin で暗号化します。

処理	暗号方式	鍵
署名	RSASSA-PKCS1-v1_5 SHA256	uImage/DeviceTree 用一時署名検証鍵(SBS-Key.der)
暗号化	AES-128-CBC	uImage 用一時暗号化鍵(E1-Key.bin)※ DeviceTree 用一時暗号化鍵(E2-Key.bin)※

note※ ：IV には E1-Key.bin および E2-Key.bin の後半 16byte が使用されます(図 11-2 Input フォルダ内の各ファイル参照)。

プロビジョニング処理/セキュアソフトウェアアップデート処理に必要となる Linux カーネルデータを作成するための手順は以下になります。

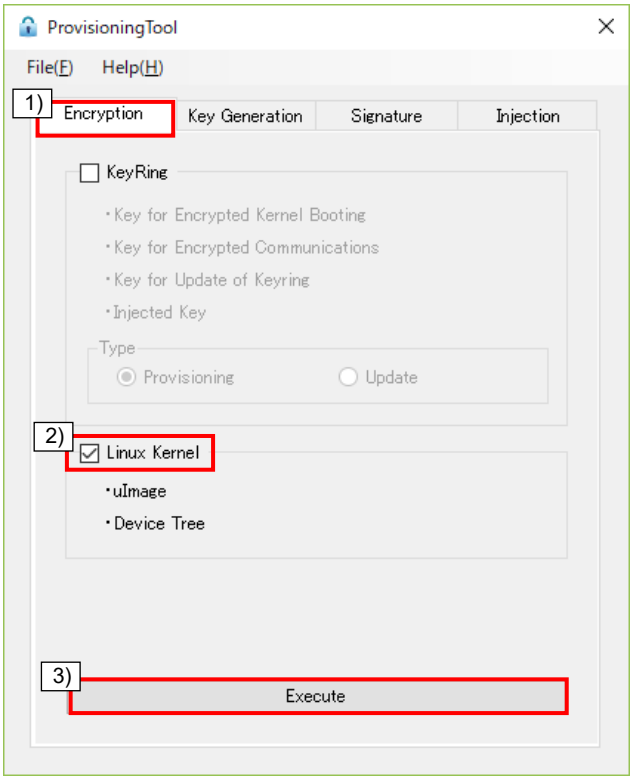

図 11-6 Linux カーネルデータ作成のツール画面

- 1) “Encryption”タブを選択する
- 2) “Linux Kernel” のチェックボックスを選択する
- 3) “Execute” ボタンを押下する

11.4.5 更新用鍵束データを作成する

KeyRing フォルダ内の各鍵から、鍵束を作成し、セキュアソフトウェアアップデート処理で使用できるように暗号化します。 UpdateKey フォルダ内の鍵と、KeyRing フォルダ内の鍵が使用されます。Keyring フォルダ内の各鍵をバイナリ連結し、鍵束のフォーマットに従って鍵束データを作成します。作成された鍵束に対して SS-UP2-Key.bin で MAC 値が生成され、鍵束データに MAC 値を付加して SS-UP1-Key.bin で暗号化します。

処理	暗号方式	鍵	IV
MAC 演算	CBC-MAC with AES-128	鍵束用 MAC 鍵(C2-Key.bin)	0
暗号化	AES-128-CBC	鍵束用一時暗号化鍵(C1-Key.bin)	IV0※

※ IV0=0x85c1673483d5d291f0d0713e3ea434a3

セキュアソフトウェアアップデート処理に必要となる鍵束データを作成する手順は以下になります。

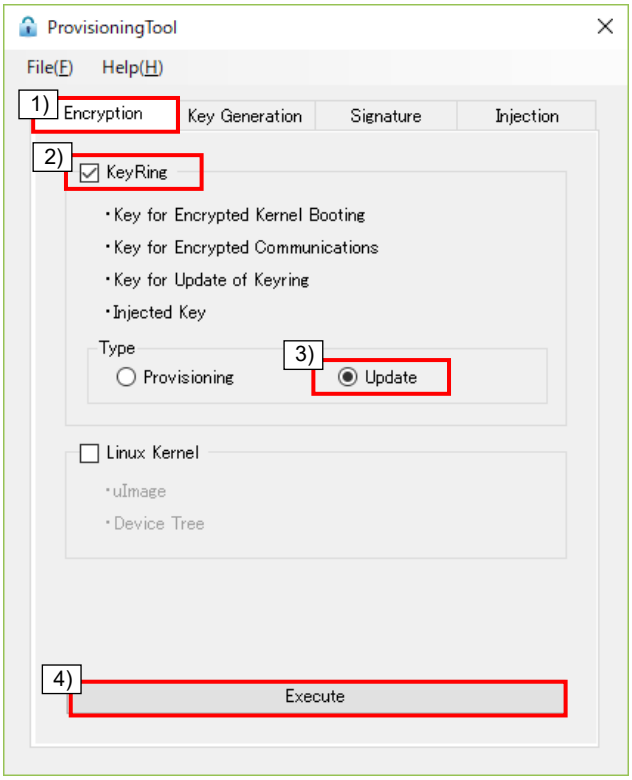

図 11-7 更新用鍵束データ作成のツール画面

- 1) “Encryption”タブを選択する
- 2) “KeyRing”のチェックボックスを選択する
- 3) “Type”の”Update”を選択する
- 4) “Execute”ボタンを押下する

11.4.6 鍵データの一時暗号化を行う

ユーザ鍵データの一時暗号化を行います。Injection フォルダ内の各ユーザ鍵を、Keyring フォルダ内の BCF2-Key.bin で MAC 演算を行い、MAC 付加して BCF1-Key.bin で暗号化します。

処理	暗号方式	鍵	IV
MAC 値演算	CBC-MAC with AES-128	ユーザ鍵用 MAC 鍵(BCF2-Key.bin)	0
暗号化	AES-128-CBC	ユーザ鍵用一時暗号化鍵(BCF1-Key.bin)	※

※ツール内で自動生成されます。

ユーザ鍵データを RZ/G で使用するための一時暗号化処理を行う手順は以下になります。

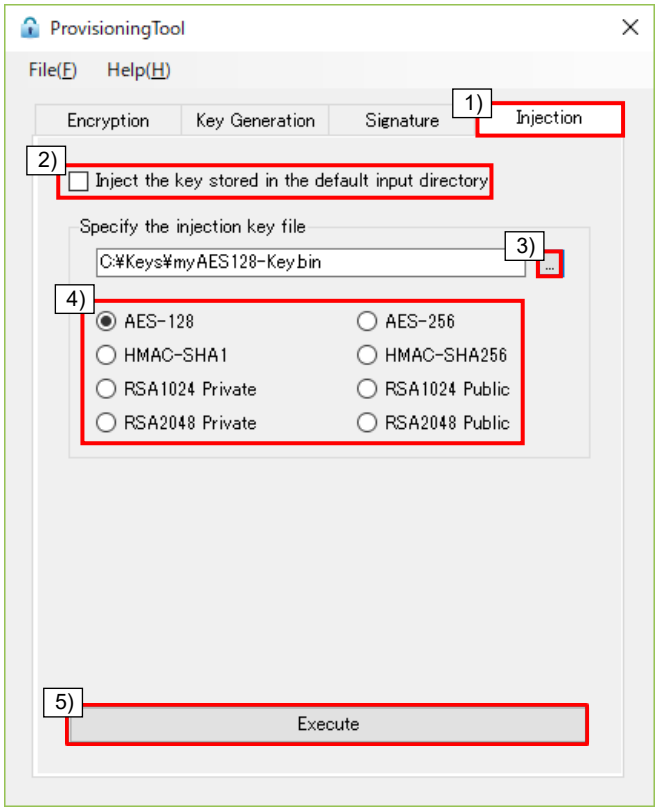

図 11-8 ユーザ鍵一時暗号化処理のツール画面

- 1) “Injection”タブを選択する。
- 2) “Inject the key stored in the default input directory”のチェックボックスをオフにする。※
- 3) “Specify the injection key file”のファイル選択ダイアログボタンを押下し、ファイル選択ダイアログにて対象となる鍵データを選択する。
- 4) “Specify the injection key file”にて、対象となる鍵データの暗号方式を選択する。
- 5) “Execute”ボタンを押下する。

note※: “inject the key stored in the default input directory”のチェックボックスを選択していると、Input/injection フォルダにある鍵データが、インジェクションの対象となる鍵データとして自動的に選択されます。

11.4.7 エラー表示

必要なファイルが存在しない場合、以下のエラーメッセージを表示します。

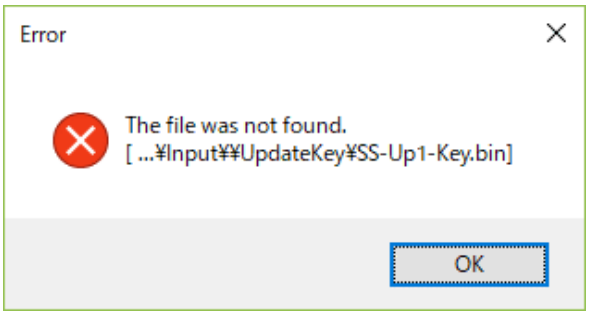

図 11-9 エラーメッセージ

指定パスにファイルが存在するか確認して、再実行してください。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec. 25, 2019	-	新規

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。