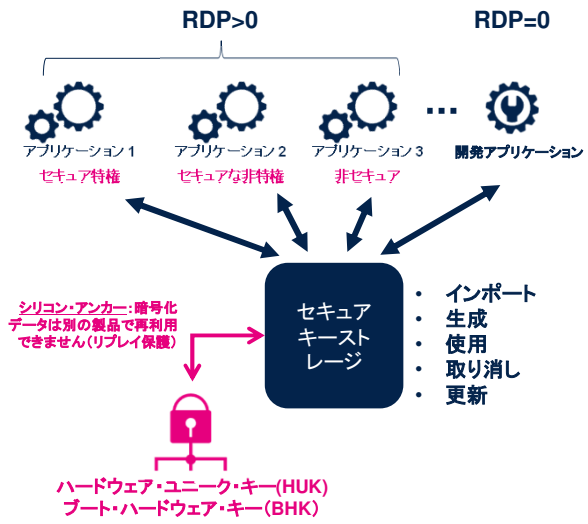

STM32U5 強化セキュアキーストレージ機能のプレゼンテーションへようこそ。ここでは、セキュアキーストレージアプリケーションに使用される SAES モジュールの機能について説明します。

# セキュアキーストレージアプリケーション




- 複数のアプリケーションから、デバイスに格納されたキーにアクセスする必要があります
- STM32 には、この重要な機能の**堅牢性を高める**ハードウェアメカニズムがあります

**適用の利点**

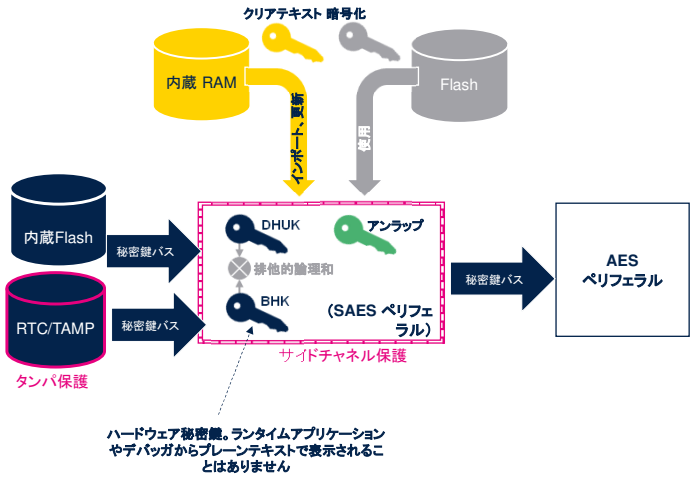
- 管理するキーの機密性を保護します
- キーのライフサイクル、つまりインポート、生成、使用、取り消し、更新を管理します。
- 各キーを許可されたユーザのみが利用できるようにします（理想的に）

life.augmented

2

セキュアキーストレージアプリケーションは、セキュリティにとって重要です。
STM32 には、堅牢性を高めるためのハードウェアメカニズムが含まれています。このトレーニング モジュールでは、この点について説明します。
完全なハードウェア機能である読出し保護（RDP）メカニズムでは、デバイスのデバッグ、テスト、およびプロビジョニングされた機密情報へのアクセスを制御します。
セキュアキーストレージでは、キーのライフサイクル（インポート、生成、使用、取り消し、更新）が管理されます。

# 強化セキュアキーストレージ




## 適用の利点

- 最新のセキュリティ標準(PSA、SESIP など)により、キーの保護の向上がますます求められるようになっています
- チップごとに固有の ID を持つオンチップの暗号化ストレージ技術を有効にします
  - 物理的クローン不可機能(PUF)の代替
  - ST ソリューションではシステム Flash (RHUK)に含まれる機密情報とシリコンに含まれる機密情報を利用

life.augmented

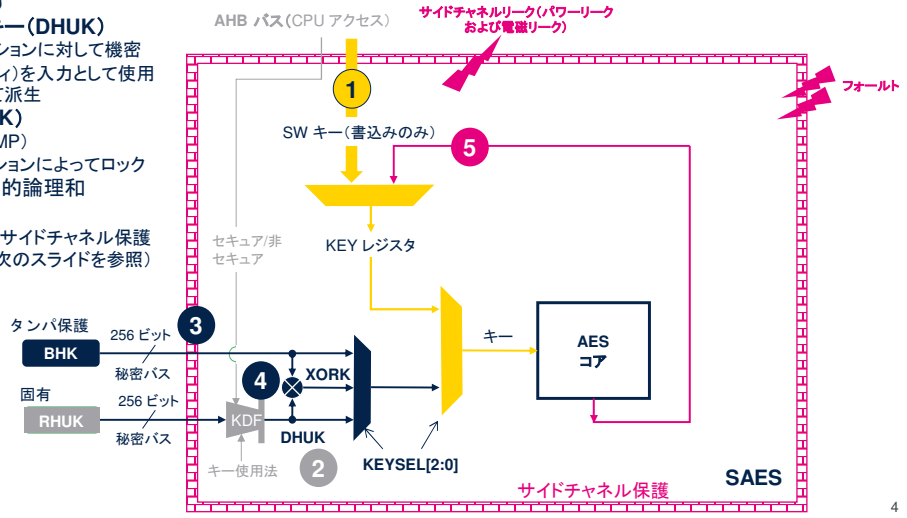
3

キーをより適切に保護するため、サイドチャネル保護された SAES ペリフェラルでは特殊なハードウェア秘密鍵 DHUK および BHK を使用して、ランタイムアプリケーションやデバッガにはプレーンテキストで表示されることのない重要なアプリケーションキーを作成します。
派生ハードウェア・ユニーク・キー(DHUK)を利用することにより、ユーザは物理的クローン不可能機能(PUF)
実装することができます。
このモジュールでは、次のユースケースについて説明します。
- ハードウェア秘密鍵によるキーの暗号化(キーのラップ)
- ハードウェア秘密鍵によるキーの復号化(キーのアンラップ)
- AES ペリフェラルのためのキーのプロビジョニング(共有鍵のラップ)
- AES ペリフェラルのためのキーの復号化(共有鍵のアンラップ)
- SAES におけるハードウェアキー保護

# SAES で可能な鍵の選択

1. **ソフトウェアキー**
   - CPU によって書き込まれます
2. **派生ハードウェア・ユニーク・キー(DHUK)**
   - 不揮発性であり、アプリケーションに対して機密
   - キー用途(モード、セキュリティ)を入力として使用するキー派生関数を使用して派生
3. **ブート・ハードウェア・キー(BHK)**
   - 揮発性であり、タンパ保護(TAMP)
   - 書込み後に、ブートアプリケーションによってロック
4. **XORK**:BHK と DHUK の排他的論理和
5. **ラップキー**
   - キーとして直接書き込まれたサイドチャネル保護された AES 復号化の結果(次のスライドを参照)

- キー 2 から 4 は、ランタイム CPU やデバッガによって読み出されることはありません
- キー 5 も、キー 2 から 4 で暗号化されている場合、ランタイム CPU やデバッガによって読み出されることはありません




life.augmented

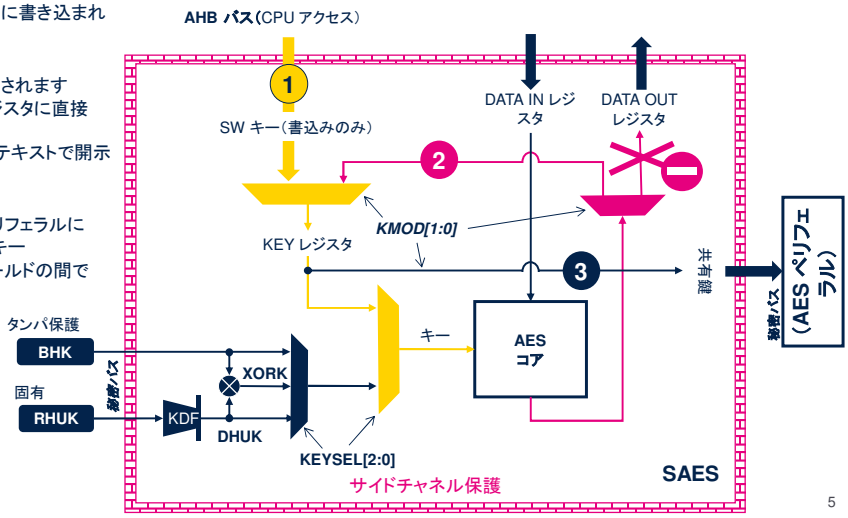
4

サイドチャネルで保護された SAES 内では複数の鍵を使用できます。
- ソフトウェアキー。CPU によって書込み専用キーレジスタに書き込まれます。
- 不揮発性でアプリケーションに対して機密の派生ハードウェア・ユニーク・キー(DHUK)。DHUK の派生には、内部キー派生関数が使用されます。図の KDF を参照してください。
- ブートハードウェアキー(BHK)。揮発性で、TAMP ペリフェラル内でタンパ保護されます。BHK は書込み後に、ブートアプリケーションによってロックされます。
- BHK と DHUK の排他的論理和
- ラップキーは、理想的にはハードウェア秘密鍵で暗号化され、ハードウェアに対して機密でもあります。

したがって、SAES ペリフェラルは、ハードウェア秘密鍵 DHUK を使用してアプリケーションキーのラップとアンラップを行うことができ、アプリケーションキー BHK により排他的論理和を取ることも、取らないこともできます。
ラップ/アンラップは、鍵の暗号化/復号化と同じ意味です。
この機能により、AES キーをプレーンテキストで公開されることなく、アプリケーションソフトウェアで使用可能にできます。

# SAES で可能な鍵の使用

1. **通常キーモード**
   - CPU によって書込み専用レジスタに書き込まれます
2. **ラップキーモード**
   - CPU によって書き込まれ、暗号化されます
   - 復号化された鍵が書込み専用レジスタに直接ロードされた場合
   - アプリケーションに対してプレーンテキストで開示されることはありません
3. **共有鍵モード**
   - 保護されていない高速な AES ペリフェラルによってロードできる、特殊なラップキー
   - セキュアワールドと非セキュアワールドの間でキーを共有する簡単な方法

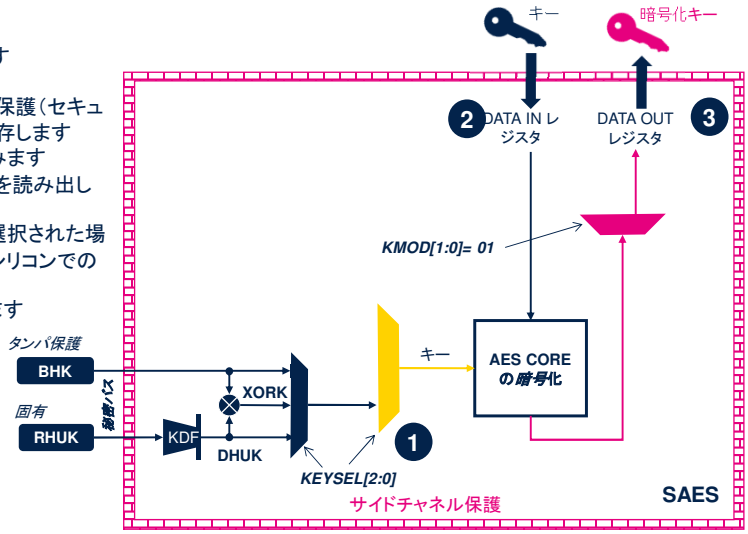



SAES にキーをロードする際、アプリケーションには 3 つのオプションがあります。

- キーレジスタに直接書き込みます。
- 暗号化キーを書き込んで、キーレジスタに直接復号化します。
- 暗号化キーをキーレジスタに直接書き込んで復号化して、AES ペリフェラルに秘密鍵バスを介してロードさせます。

# プロビジョニングの使用法：キーのラップ

- **ラップモード**を使用したキーの暗号化
  1. ハードウェア秘密鍵を選択します（DHUK、BHK、XORK）
     - KDF は KMOD & SAES 保護（セキュアまたは非セキュア）に依存します
  2. プレーンテキストキーを書き込みます
  3. 暗号化され、ラップxされたキーを読み出します
     - DHUK または XORK が選択された場合、キーの復号化はこのシリコンでのみできます
  4. キーを任意の Flash に保存します

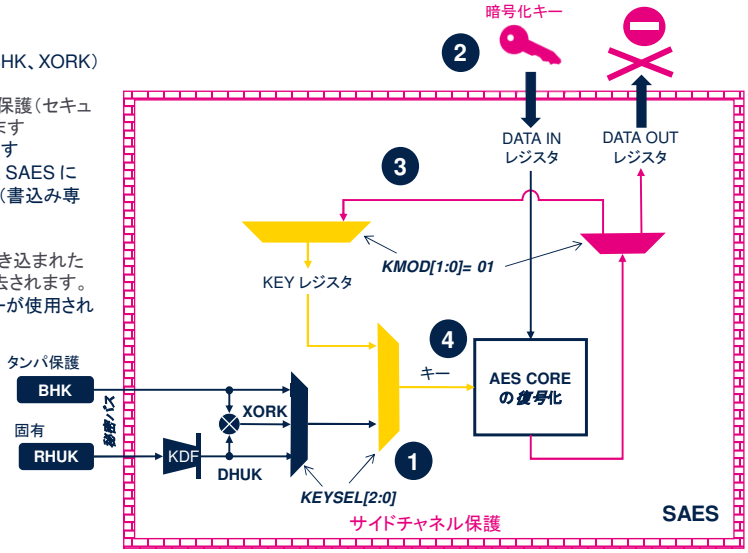





このスライドでは、ラップキーモードを使用してキーを暗号化する方法について詳しく説明します。生成された暗号化キーは、任意の Flash に格納できます。
最初の手順では、ハードウェア秘密鍵を選択します。キー派生関数は、TrustZone の状態と、KMOD フィールドでプログラムされたキーの使用状態に依存します。
次に、ソフトウェアによりプレーンテキストキーが書き込まれます。
最後に、AES コアにより、生成された暗号化キーが読み出されます。

# プロビジョニングの使用法：キーのアンラップ

- **ラップモード**を使用したキーの復号化
  1. 正しいハードウェア秘密鍵（DHUK、BHK、XORK）を選択します
     - KDF は KMOD および SAES 保護（セキュアまたは非セキュア）に依存します
  2. ラップされた暗号化キーを書き込みます
  3. 復号化され、アンラップされたキーが、SAES によってキーレジスタに書き込まれます（書込み専用）
     - DOUT はゼロを返します
     - キーレジスタが s/w によって書き込まれた場合、キー全体が自動的に消去されます。
  4. 必要に応じて、アプリケーションでキーが使用されます。

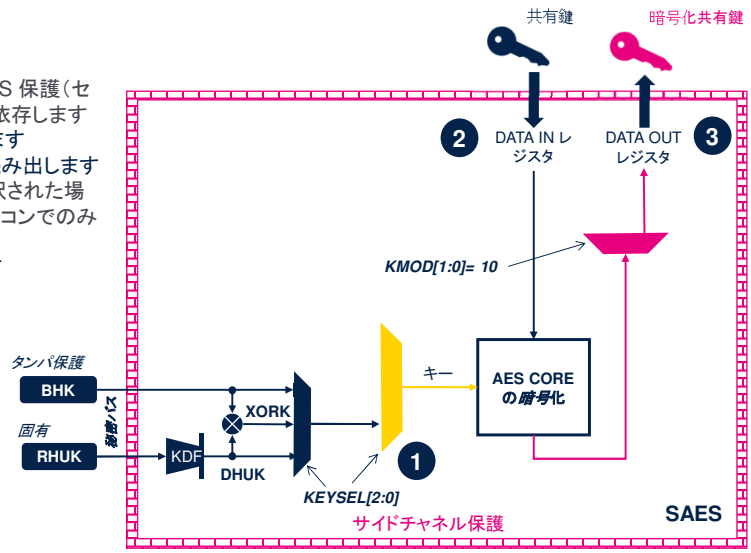





ラップキーモードで暗号化されたキーは、正しいキーを使用して、同じモードで復号化する必要があります。結果は、書込み専用キーレジスタに自動的に格納されます。
前のスライドで生成した暗号化キーは、暗号化および復号化の目的で使用する前に、復号化する必要があります。
最初の手順では、ハードウェア秘密鍵を選択します。
手順 2 で、ソフトウェアによりラップされたキーが DATA IN レジスタに書き込まれます。
手順 3 で、AES コアによりこのキーがアンラップされ、元のキーがソフトウェアではアクセスできない SAES モジュール内の書込みレジスタに格納されます。
その後、アプリケーションでは、必要に応じて、アンラップされたキーを使用できます。
このアンラップされたキーは、ソフトウェアによりキーレジスタに書き込まれるときに自動的に消去されることに注意してください。

# プロビジョニングの使用法：共有鍵のラップ

- **共有モード**を使用したキーの暗号化
  1. ハードウェア秘密鍵を選択します（DHUK、BHK、XORK）
     - KDF は KMOD および SAES 保護（セキュアまたは非セキュア）に依存します
  2. プレーンテキストキーを書き込みます
  3. 暗号化され、ラップされたキーを読み出します
     - DHUK または XORK が選択された場合、キーの復号化はこのシリコンでのみできます
  4. キーを任意の Flash に保存します

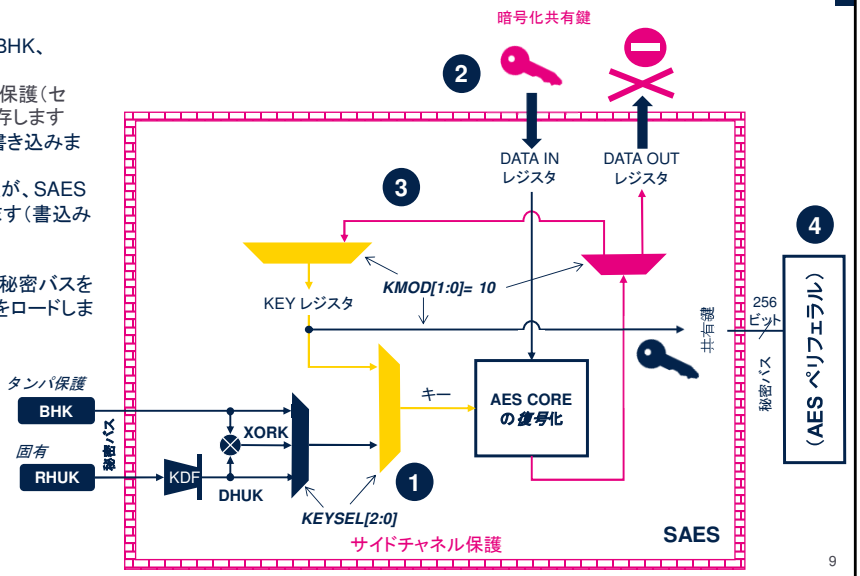





このスライドでは、共有鍵モードで共有鍵を暗号化する方法について詳しく説明します。生成された暗号化キーは、任意の Flash に格納できます。
SAES で AES ペリフェラルとキーを共有するには、キーを 1 回暗号化する必要があります。
共有鍵の暗号化シーケンスはラップされたキーの場合と同じですが、KMOD は 01 ではなく 10 に設定されます。

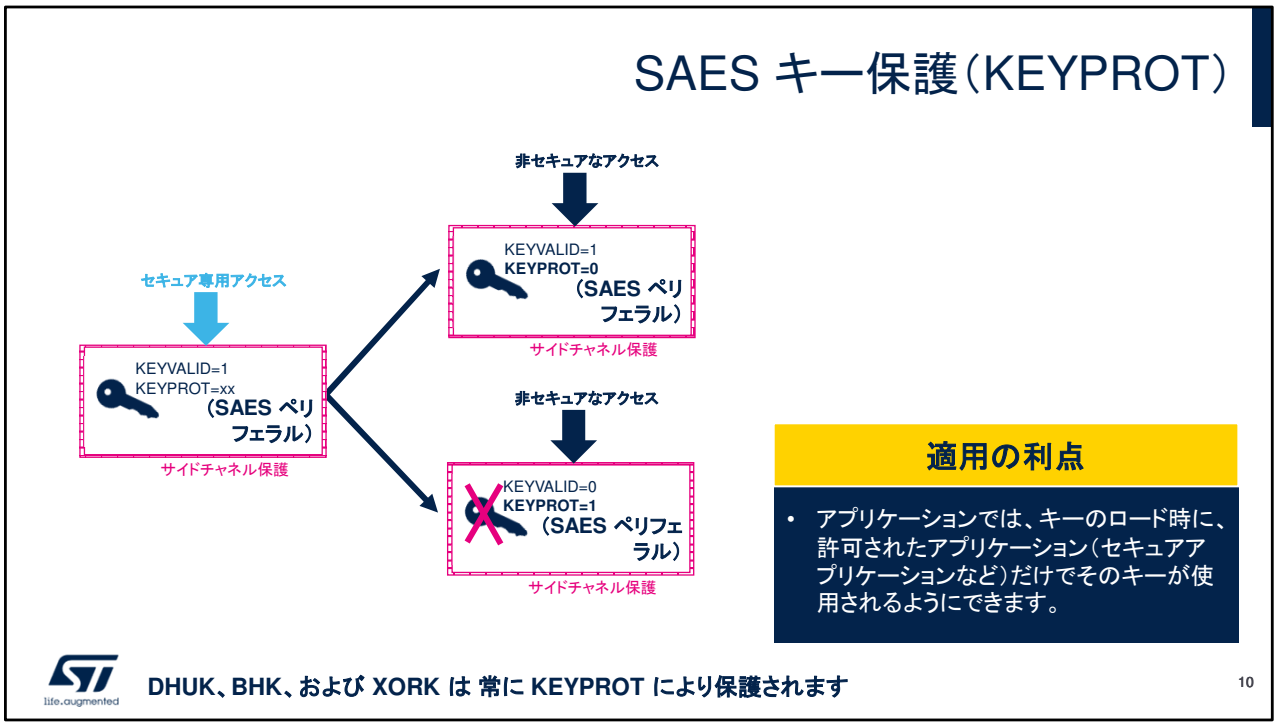
# 共有鍵のアンラップ

- **共有モード**を使用したキーの復号化
  1. 正しいハードウェア秘密鍵（DHUK、BHK、XORK）を選択します
     - KDF は KMOD および SAES 保護（セキュアまたは非セキュア）に依存します
  2. ラップされて暗号化された共有鍵を書き込みます
  3. 復号化され、アンラップされた共有鍵が、SAES によってキーレジスタに書き込まれます（書込み専用）
     - DOUT はゼロを返します
  4. キーを SAES で使用するか、または秘密バスを使用して AES ペリフェラルからキーをロードします。

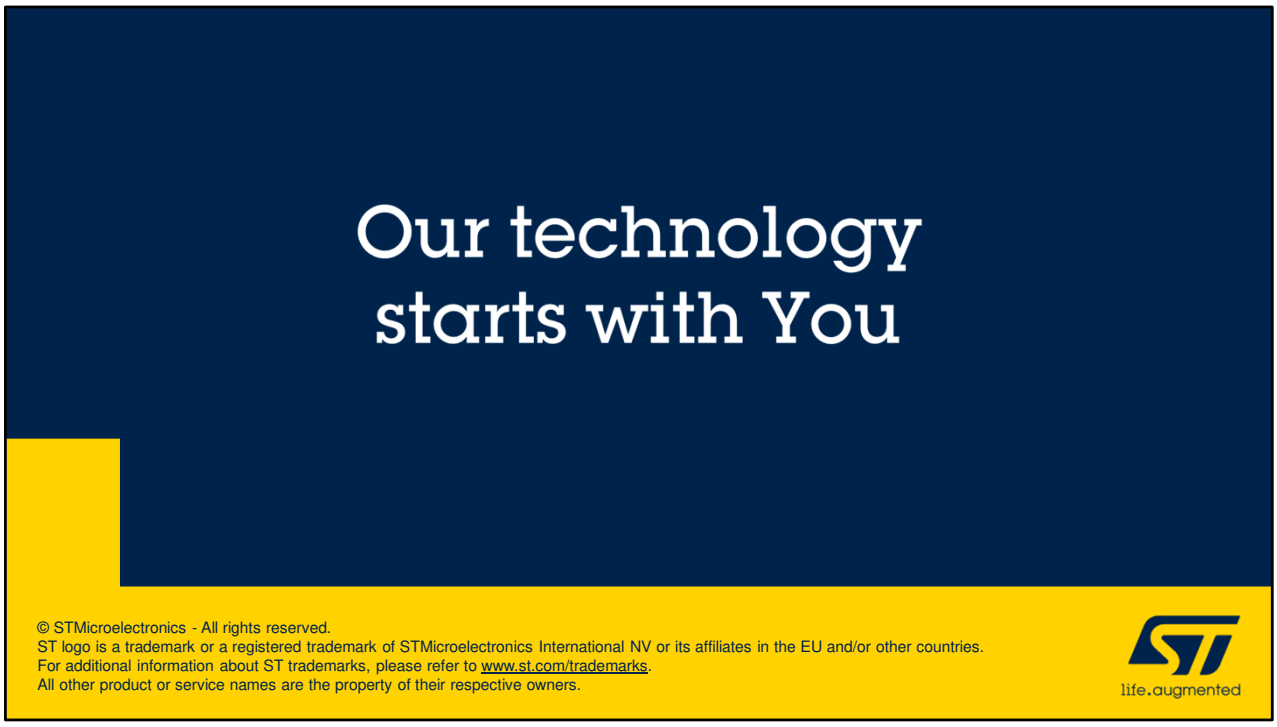





SAES で AES ペリフェラルとキーを共有する必要がある場合は必ず、共有暗号鍵を SAES で復号化し、AES によってロードする必要があります。
共有鍵の復号化シーケンスはラップされたキーの場合と同じですが、KMOD は 01 ではなく 10 に設定されます。
結果は書込み専用キーレジスタに自動的に格納され、SAES ステータスレジスタの KEYVALID が 1 のときに、AES ペリフェラルによってロードできます。




SAES にキーをロードするとき、アプリケーションで許可されたアプリケーション（セキュアアプリケーションなど）だけでそのキーが使用されるようにするには、KEYPROT ビットに 1 をセットします。
KEYPROT が 1 の場合、SAES では、ロードされたキーのセキュリティレベルと一致しないアクセスが検出されると直ちに、キーデータが消去され、KEYVALID がクリアされます。
DHUK、BHK、および XORK は、常に KEYPROT により保護されます。




SAES モジュールの詳細については、対称暗号のトレーニングモジュールを参照してください。