RTOS上でのセキュリティフレームワークの構築
―セキュリティ機構の見える化を実現―

1. 背景

従来のクローズな組込みシステムでは、安全性の上で問題となる脅威にさらされる機会が少なかった。しかしながら、近年、車載システムにように人命を左右する組込みシステムにおいても、汎用コンピュータシステムとの統合が必要となり、汎用コンピュータシステムの脅威が組込みシステム側にも大きく影響を及ぼすようになってきている。このような背景から、組込みシステムにもセキュリティを強化する機構を取り入れる必要性が出てきた。特にRTOS（リアルタイムOS）ではまだセキュリティの機構が確立されておらず、対応が求められている。

2. 目的

本プロジェクトでは、組込みシステム向けコンポーネントにセキュリティの機構を追加したフレームワークを構築し、RTOS上でセキュリティを確保することを目的とする。なお、このセキュリティフレームワークは、組込みシステム内のデータや機能へのアクセス制御に焦点を絞る。

3. 開発内容

既存のコンポーネントベースのソフトウェアに対して、アクセス制御機構を付加するフレームワークを構築する。保護対象外のコンポーネントから保護対象のコンポーネントにアクセスするインタフェース部分に対して、セキュリティコンポーネント、リファレンスモニタから成るアクセス制御機構を付加することで実現する。

○ コンポーネント

本プロジェクトの基にしている組込みコンポーネント仕様では、コンポーネント同士を静的に結合して目標とするアプリケーションを構築することができる。図1に、組込みコンポーネント仕様におけるコンポーネント図の例を示す

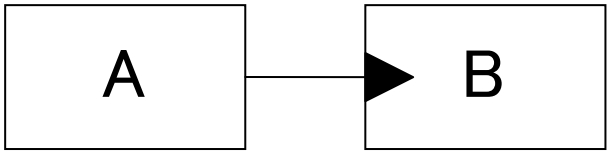

図1　コンポーネント図

図1では、コンポーネントAとBが繋がっており、AがBの機能を呼び出すという関係にある。

○ アクセス制御機構を付加したコンポーネント図

本プロジェクトにおけるアクセス制御機構は、セキュリティエンハンサによって付加される。セキュリティエンハンサは、コンポーネント同士の結合関係と保護対象を調べて、保護対象外から保護対象にアクセスする部分にアクセス制御機構を付加することで、コンポーネントの保護を行う。コンポーネントの保護を行うのは、守るべきリソースがコンポーネントに含まれるためである。セキュリティエンハンサによってアクセス制御機構が付加された様子を図2に示す。

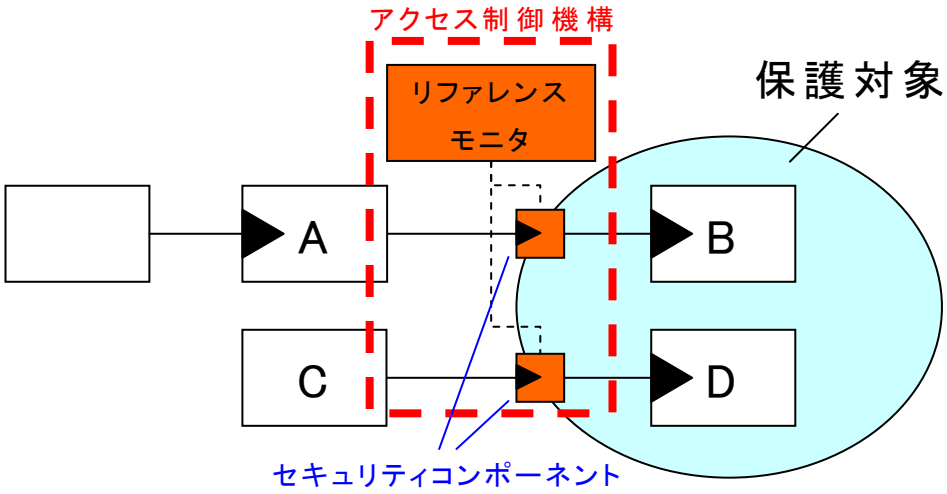

図2　コンポーネントにアクセス制御機構が付加された様子

セキュリティコンポーネントは、保護対象に含まれるコンポーネントへのアクセス制御を行うために付加される。アクセス制御は、アクセスのルールが記述されたリファレンスモニタを参照することで行う。図2の例では、保護対象外のコンポーネント（AやC）から保護対象に含まれるコンポーネント（BやD）へのアクセス時に、セキュリティコンポーネントを経由して、リファレンスモニタでアクセスの可否を検証する。

このアクセス制御機構を適用したコンポーネントベースのアプリケーション開発の手順では、始めにアプリケーション開発者がコンポーネント同士の結合状態を定義する。次に、セキュリティエンハンサがセキュリティコンポーネントとリファレンスモニタを付加する。そして、インタフェースジェネレータがインタフェースコードとヘッダ（共にC言語のソース）を生成する。

○ **インタフェースジェネレータ**

インタフェースジェネレータは、インタフェース記述（インタフェースの定義）、コンポーネント記述（コンポーネントの定義）、組上げ記述（コンポーネント同士の接続の定義）からCソースを生成する。

○ **セキュリティエンハンサ**

セキュリティエンハンサは、ビジュアルエディタ（GUIベースのコンポーネント接続ツール）で指定された保護対象に向かう結合部分に、適切にthrough文を挿入することで、セキュリティコンポーネントを付加するものである。

- **through**

  セキュリティエンハンサの機能を実現するために、組上げ記述に新たにthroughを追加した。throughは、コンポーネントの結合間に、コンポーネントを挿入するためのものである。挿入されるコンポーネントはthrough文で指定されたプラグインの内容に応じたものになる。このコンポーネントは、自身のインタフェースを前後のコンポーネントのインタフェースに自動的に合わせるので、前後のコンポーネント側で変更を加える必要はない。今回の開発では、セキュリティコンポーネントを挿入するためにthroughを使用している。

- **セキュリティエンハンサの実装**

  セキュリティエンハンサは、ビジュアルエディタの一機能として実装され、ビジュアルエディタで選択された保護対象に対して、適切にセキュリティコンポーネントを挿入する。セキュリティコンポーネント自体は、through文を含んだ組上げ記述を生成することによって挿入できる。

○ **セキュリティコンポーネント**

セキュリティコンポーネントは、保護対象外のコンポーネントから保護対象のコンポーネントの機能（関数）を呼び出す部分に挿入され、保護対象へのアクセスを制御するものである。保護対象外のコンポーネントから機能呼び出しが行われると、セキュリティコンポーネントは、現在アクセスして来ているサブジェクトや呼び出された機能といった情報をリファレンスモニタに渡し、返ってきたアクセス可否の情報に応じてアクセス制御を行う。セキュリティコンポーネントはthroughで用いるプラグイン形式で実装されており、既存のコンポーネントに影響を与えずに付加できるようになっている。

○ **リファレンスモニタ**

リファレンスモニタは、セキュリティコンポーネントから渡される関数名と、現在保護対象にアクセスしようとしているサブジェクト名から、保護対象へのアクセスの可否を判定し、その結果をセキュリティコンポーネントに返すものである。アクセスのルールは、ユーザが記述したルール記述ファイルからC言語のソースに変換される。

○ **FatFSコンポーネント**

保護対象の例としてファイルシステム（FatFs）をコンポーネント化

して使用した。FatFsは、フリーソフトウェアとして公開されているFATファイルシステムである。構成は、ファイルシステム全体に関連する機能を持つFATFSコンポーネント、ディレクトリ操作の機能を持つDIRコンポーネント、ファイル操作の機能を持つFILEコンポーネント、ハードウェア依存部を担当するDiskIOコンポーネントから成る。

4. 従来の技術（または機能）との相違

従来、RTOS上では個々のミドルウェアやアプリケーションでセキュリティの機構を作成する必要があったが、今回、統一したフレームワークを作成した。

5. 期待される効果

○ **システムの検証がしやすくなる**

従来はアプリケーション内にセキュリティの機構のコードが埋め込まれていたため、セキュリティ設計の客観的な評価が難しかった。セキュリティコンポーネントは元々存在したコンポーネントとは独立しているため、セキュリティコンポーネントの有無、セキュリティコンポーネントの内容をレビューすることで容易に評価できる。

○ **一括してアクセス制御機構を付加できるようになる**

保護対象が決まれば、セキュリティエンハンサよってセキュリティコンポーネントが一括して付加されるため、アクセス制御の抜けがなくなる。

○ **コンポーネントを安全に再利用できるようになる**

他のアプリケーションで使用されたコンポーネントは、必ずしも安全とは限らない。しかしながら、アクセス制御機構が存在することで、安心して他のコンポーネントを再利用することができる。

○ **コンポーネントベースのアプリケーションのテスト・デバッグに利用できる**

コンポーネント間の呼び出しをチェックできるため、クローズな組込みシステムにおいても、コンポーネントベースのアプリケーションのテスト・デバッグに用いることができる。

なお、アクセス制御機構を付加した場合のオーバーヘッドは、付加しない場合の実行時間の1%前後であり、リアルタイム性への影響は少ないと考えられる。

6. 普及（または活用）の見通し

コンポーネント仕様ワーキンググループを通じて、TOPPERSからオープンソースとして一般に公開する。

7. 開発者名（所属）

- 安積 卓也（名古屋大学大学院 情報科学研究科）
- 山田 晋平（兵庫県立大学大学院 応用情報科学研究科）