セキュアプラットフォームソフトウェア LiSTEE$_{TM}$

LiSTEE$_{TM}$ Secure Platform Software

金井　遵　　磯崎　宏
■KANAI Jun　　■ISOZAKI Hiroshi

近年，組込み機器においてもLinux®(注1)のような大規模な汎用OS（基本ソフトウェア）の搭載事例が増えてきている。大規模なOSには脆弱（ぜいじゃく）性も混入しやすく，脆弱性により，アクセス制御などOSのセキュリティ機構が無効化されてしまうと，組込み機器からの情報漏えいや不正な処理の実行を招くことになる。

そこで東芝は，OSのセキュリティ機構が無効化された際でも重要な処理やデータを保護するために，汎用OSと保護が必要なセキュリティアプリケーションを分離して実行するセキュアプラットフォームソフトウェア LiSTEE$_{TM}$を開発した。LiSTEE$_{TM}$は，汎用OSとセキュリティアプリケーションの高速な切替え機能や，セキュリティアプリケーションのセキュアな更新機能を搭載している。これにより，ハードウェアレベルの保護強度をソフトウェアで実現できるようになり，組込み機器のセキュリティ強化と開発期間短縮を両立できる。

The increase in embedded systems featuring a rich versatile operating system (OS) such as Linux® in recent years has been accompanied by an increased risk of vulnerability. Once security systems including the access control system are defeated through a vulnerability in the OS, information leakages and illegal operations may occur in the embedded system.

As a solution to this issue, Toshiba has developed a secure platform software called LiSTEE$_{TM}$ to protect important processing systems and data when security systems are defeated. LiSTEE$_{TM}$ can implement security applications separately from a rich versatile OS by means of a fast switching function between the OS and a security application, and also provides a secure update function. This software achieves a balance between enhancement of security and shortening of the development period for embedded systems by improving the security to a level comparable to that attained by hardware.

1　まえがき

近年，組込み機器のネットワーク化や高機能化に伴い，組込み機器においてもセキュリティが重要な課題になっている。一方で，組込み機器のOSとしても広く利用されるようになったLinuxは，OSの規模が非常に大きく，攻撃の糸口となる不具合（脆弱性）が混入しやすい。なかでも，管理者権限の不正取得が可能になるような脆弱性が毎月のように報告されている[1]。管理者権限が不正取得されると，Linuxのファイルやプロセスへのアクセス制御が無効化されるため，情報の不正な取得につながるおそれがある。

Linuxのような規模になると，脆弱性を皆無とするような対策は現実的ではない。また，アクセス制御の強化やウイルス対策など，Linuxのセキュリティを強化する試みも多く行われているが[2]，OS自体の脆弱性から，これらのソフトウェアもまた無効化されるおそれがある。したがって，Linuxなどの汎用OSと保護が必要なセキュリティ処理を分離して実行する仕組みが必要になる。この要請に応えるため，組込み機器で多く利用されるARM(注2)プロセッサの一部ではTrustZone®(注3)[3]と呼ばれるセキュリティ機能を搭載している。

東芝は，TrustZone®を利用して汎用OSとセキュリティが求められるアプリケーションとを分離して実行できるプラットフォームソフトウェアLiSTEE$_{TM}$（Light-Weight and Secure TrustZone® based Embedded Environment）を開発している。LiSTEE$_{TM}$は，汎用OSとセキュリティアプリケーションの高速な切替え実行や，セキュリティアプリケーションのセキュアな更新などの特徴的な機能を搭載している。これにより，ハードウェアレベルの保護強度をソフトウェアで実現できるようになり，組込み機器のセキュリティ強化及び開発期間短縮を両立させることができる。また，ハードウェアに比べて更新が容易であるという柔軟性を生かし，例えば暗号化機能などの更新により，長期的に機器の安全性を確保することもできる。

ここでは，LiSTEE$_{TM}$の機能と構成，及びその有用性について述べる。

2　LiSTEE$_{TM}$の概要

2.1　TrustZone®とは

TrustZone®は，セキュアワールドとノンセキュアワールドと

（注1）　Linuxは，Linus Torvalds氏の日本及びその他の国における登録商標又は商標。
（注2），（注3）　ARM，TrustZoneは，英国ARM Limitedの商標。

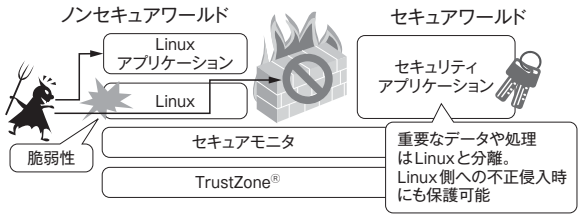

図1. ノンセキュアワールドとセキュアワールドの分離 ― Linuxとセキュリティ処理を分離して実行することで，Linux側への不正侵入時にもセキュリティ処理を保護できる。

Isolation of non-secure and secure worlds

呼ばれる二つの状態（ワールド）でプログラムを分離して実行するARMプロセッサの機能である。

セキュアワールドとノンセキュアワールドで，それぞれ別のOSやアプリケーションを動作させることができる。例えば，ノンセキュアワールドでは汎用処理を動作させるためのLinuxを動かし，セキュアワールドでは小規模で脆弱性混入リスクを最小化したOSを動作させることができる。

また，両ワールドからのメモリやペリフェラル（周辺機器）へのアクセスを制限することもできる。このメモリ保護機能により，セキュアワールドで動作するOSやアプリケーションが利用するメモリを，ノンセキュアワールドから不正にアクセスしたり改ざんしたりすることを防止できる。これにより，たとえノンセキュアワールドで動作するLinuxに脆弱性があり，管理者権限が不正取得されたとしても，セキュアワールドで動作するOSやアプリケーションには直接影響を及ぼすことはない（**図1**）。

ここで，ノンセキュアワールドで動作する汎用OSとセキュアワールドで動作するセキュリティアプリケーションの切替えを行うためのソフトウェアはセキュアモニタと呼ばれ，各ワールドで動作するソフトウェアからセキュアモニタコールと呼ばれる専用の命令で呼び出すことができる。

2.2 セキュアプラットフォームソフトウェアの設計要件

TrustZone®を利用してセキュアプラットフォームソフトウェアを設計する際に求められる要件は，以下のとおりである。これらの要件を満足することで，更新可能なソフトウェアの柔軟性を生かしつつ，セキュリティの向上が実現できる。

(1) セキュアワールド側のデータ改ざんや盗用が不可能
ソフトウェアが動作中に扱う内部データの改ざんと盗用を防止するもので，セキュアプラットフォームのもっとも基本的なセキュリティ要件である。

(2) 二つのワールドで連携して処理が可能　一つのソフトウェアにはセキュリティの観点で，一般に保護すべき処理と保護しなくてもよい処理があり，それらが二つのワールドで連携して動作することが求められる。そのためワールド間で，相互にアプリケーションの呼出しやデータ交換ができる必要がある。

(3) 二つのワールドが高速に切替え可能　二つのワールド間で連携して処理を行う際，ワールドの切替えにかかる時間が長いと処理速度が著しく低下する。このため，ワールド切替えは極力高速である必要がある。

(4) セキュアワールド側のアプリケーションが更新可能
更新が容易なことはソフトウェアで実装する一つのメリットである。組込み機器でもソフトウェア更新の要求は多くあり，セキュアワールドのアプリケーションも更新が可能なことが求められる。

2.3 LiSTEE$_{TM}$の設計と実装

LiSTEE$_{TM}$は，前述した要件を満足する，TrustZone®の機能を活用し汎用OSと保護が必要なセキュリティアプリケーションを分離してセキュアに実行するためのプラットフォームソフトウェアである。**図2**に示すように，LiSTEE$_{TM}$モニタ（2.1節のセキュアモニタに相当），LiSTEE$_{TM}$モニタ上で動作するLiSTEE$_{TM}$アプリケーション（2.1節のセキュリティアプリケーションに相当），及び様々な機器で利用できる共通機能を実装したLiSTEE$_{TM}$コンポーネントから成る。

組込み機器の開発者は，LiSTEE$_{TM}$アプリケーションを実装するとともに，必要に応じてLiSTEE$_{TM}$コンポーネントを組み合わせて組込み機器を開発する。また汎用OSはLiSTEE$_{TM}$モニタ上で動作し，図2で示したようにLinuxアプリケーションからLiSTEE$_{TM}$アプリケーションを呼び出すことができる。データ暗号化処理を例にとると，ストレージやネットワークとの入出力はノンセキュアワールドのLinuxアプリケーションで行い，暗号化処理をセキュアワールドのLiSTEE$_{TM}$アプリケーションで行うことが可能になる。このように実装することで，暗号化に利用する鍵はセキュアワールドだけで扱うため，Linuxアプリケーションへの不正侵入時にも鍵漏えいのおそれがない。

各構成要素について，次に述べる。

2.3.1 LiSTEE$_{TM}$モニタ　設計要件(1)のセキュアワールドのデータ保護や(2)のワールド間の連携処理を実現するに

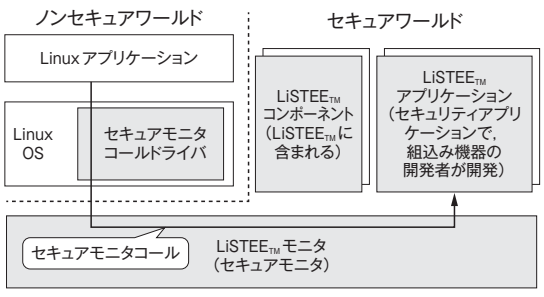

図2. LiSTEE$_{TM}$のソフトウェア構成 ― LiSTEE$_{TM}$は，LiSTEE$_{TM}$モニタや，LiSTEE$_{TM}$アプリケーション，LiSTEE$_{TM}$コンポーネントなどの複数のソフトウェアから構成される。

Software architecture of LiSTEE$_{TM}$

は，セキュアワールドとノンセキュアワールドでOSやアプリケーションを分離して実行する必要がある。そのためLiSTEE$_{TM}$では，時分割で二つのワールドを切り替えて各ワールドのOSやアプリケーションを実行している。このワールドの切替えを行うのがLiSTEE$_{TM}$モニタと呼ばれるソフトウェアで，その主な機能は次のとおりである。

(1) ワールド切替え機能　LiSTEE$_{TM}$モニタは各ワールドで動作するOSやアプリケーションからセキュアモニタコールと呼ばれる命令で呼び出され，セキュアワールドとノンセキュアワールドを切り替える。例えば，ノンセキュアワールドで動作するLinuxからセキュアモニタコールが呼び出された場合にはセキュアワールドに切り替え，LinuxからLiSTEE$_{TM}$アプリケーションを呼び出すことができる。これでワールド間の連携処理が実現できる。

LiSTEE$_{TM}$モニタでは，ワールドを切り替える際，切替え元のワールドで動作していたOSやアプリケーションの状態（レジスタ状態などのコンテキスト）を保存し，切替え先で動作させるOSやアプリケーションのコンテキストを復帰することで，各ワールド間での処理の並行動作を実現する。いわゆるOSでのユーザータスク切替えに似た仕組みを搭載している。

(2) ワールド切替えの高速化機能　ワールド切替え時に，アプリケーションの状態の保存や復帰に必要なコンテキストが多くなるとワールドの切替えは低速になる。そこでLiSTEE$_{TM}$モニタでは，ワールド切替え時に保存，復帰させるコンテキストを最小限にすることで，設計要件(3)のワールド切替えの高速化を実現している[4]。LiSTEE$_{TM}$モニタでは両方のワールドで共通して利用するレジスタだけを保存，復帰させることで，一般的なセキュアモニタに比べワールド切替えを高速化している。

(3) メモリ保護機能　LiSTEE$_{TM}$モニタはセキュアワールドのデータ保護を実現するために，TrustZone®のメモリ保護機能と連携して，ノンセキュアワールドからセキュアメモリ領域への読み書きを禁止している。このセキュアメモリ領域は，LiSTEE$_{TM}$モニタや，LiSTEE$_{TM}$アプリケーション，LiSTEE$_{TM}$コンポーネントで利用するデータやプログラムを格納するためのものである。この機能によりセキュアワールドで動作するデータやプログラムを保護することができ，LiSTEE$_{TM}$アプリケーションやLiSTEE$_{TM}$コンポーネントの機密性及び完全性を保証することが可能である。また，プログラムの保護により処理の不正な停止も防げるため，可用性の対策にもなっている。

一方，ワールド間での連携動作を実現するために，ワールド間でデータを授受するための共有メモリ領域では両ワールドからの読み書きを許可している。

2.3.2 LISTEE$_{TM}$アプリケーション

汎用OSへの不正侵入時などに外部からの攻撃から保護したい処理を，機器の開発者がLiSTEE$_{TM}$アプリケーションとして実装する。LiSTEE$_{TM}$アプリケーションは，LiSTEE$_{TM}$モニタ上のセキュアワールドで実行される。ノンセキュアワールドで動作するOSやアプリケーションからのLiSTEE$_{TM}$アプリケーションが利用するメモリの読み書きは不可能であり，処理やデータを保護することができる。LiSTEE$_{TM}$が有用なユースケースについては4章で述べる。

2.3.3 LiSTEE$_{TM}$コンポーネント

様々な組込み機器を開発する際に共通して使用できるソフトウェアコンポーネントをあらかじめ用意している。LiSTEE$_{TM}$コンポーネントとしては，設計要件(4)を実現するためのLiSTEE$_{TM}$アプリケーションの更新機能などがある。例えば暗号化方式に不具合が発見された場合，この更新機能によりLiSTEE$_{TM}$アプリケーションの暗号化方式の更新を行うことで，機器の交換なしに長期的に安全性を保証する組込み機器を実現できる。LiSTEE$_{TM}$コンポーネントもLiSTEE$_{TM}$モニタ上のセキュアワールドで動作する。

2.3.4 Linux（汎用OS）

前述したように，LiSTEE$_{TM}$では汎用OSとしてLinuxを想定している。Linuxは，組込み機器に多く採用されており，既存のソフトウェア資産を最大限に活用できる。LiSTEE$_{TM}$の汎用OSとして利用するにあたり，Linuxのソースコードは改変せず，両ワールドで利用するメモリ領域が重ならないように，利用するメモリの開始アドレスとサイズをカーネルオプションに設定する。同時に，Linuxアプリケーションから LiSTEE$_{TM}$アプリケーションを呼び出すためのLinux用デバイスドライバを用意した。このドライバは，セキュアモニタコールを発行するインタフェースと，ワールド間のデータ交換に用いる共有メモリの読み書きを行うインタフェースを提供する。

3 評価

LiSTEE$_{TM}$の有用性を評価するため，セキュアワールドのLiSTEE$_{TM}$アプリケーションとして暗号化処理を実装し，暗号化に要する時間を計測した。ノンセキュアワールドではLinuxを動作させている。評価に用いた暗号化のアルゴリズムは，平文に平文と同じ長さの鍵をXOR（排他的論理和）する暗号化と，128ビット，ECB（Electronic Code Book）モードのAES（Advanced Encryption Standard）の2種類である。評価用プログラムは16バイトごとにLinuxから共有メモリを介して与えた平文に対して暗号化を行い共有メモリに出力する。

LiSTEE$_{TM}$の高速化機能を有効，無効にした場合でそれぞれ評価を行った結果を**図3**に示す。ここで，LiSTEE$_{TM}$の高速化機能を有効にした場合にはワールド遷移時に暗号化機能とLinuxの両方で利用する16レジスタだけをコンテキストとし

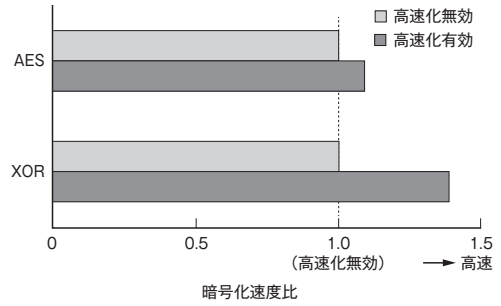

図3．暗号化速度の評価結果 ― LiSTEE$_{TM}$の高速化機能により，暗号化処理において最大38％の高速化効果が確認できた。
Results of evaluation of encryption rate

て保存，復帰させ，高速化機能を無効にした場合には全39レジスタを保存，復帰させている。

この結果から，高速化機能ありの場合，なしの場合と比べXORでは38％の，AESでは9.2％の高速化効果が得られることがわかった。特に暗号化処理自体が軽量なXOR処理では大きな効果が得られている。これにより，従来ワールド間遷移がボトルネックになりTrustZone®を用いた処理の分離の適用を見送らざるをえなかったケースでも，ワールド間遷移の高速化によりTrustZone®を用いた処理の分離が適用可能になると考える。

4　LiSTEE$_{TM}$の組込み機器への展開

近年，組込み機器のネットワークへの接続により，セキュリティを確保すべき場面が増えている。独立行政法人 情報処理推進機構（IPA）は，タブレットやデジタルテレビなどエンターテインメント向けデバイスにおける暗号化されたデジタルコンテンツの復号モジュールに含まれる鍵，タブレットや，ヘルスケア機器，スマートメータなどが扱うクレジットカード情報や健康状態・利用状況情報などの個人情報，並びに車載機器の制御プログラムなどを，保護すべき情報として挙げている[5]。これらはいずれも，機密性（不正な盗用からの保護），完全性（改ざんからの保護），及び可用性（不正な処理停止からの保護）を保証する必要がある。このような機器にLiSTEE$_{TM}$を適用することで，機器のセキュリティ向上が期待できる。

一方で開発者の観点からは，LiSTEE$_{TM}$は開発コスト削減にも寄与できると考える。既に数多くの組込み機器ではLinuxが搭載され，セキュリティの処理とは無関係の様々なアプリケーションソフトウェアが開発されている。機器のセキュリティを向上させつつ，それらのソフトウェア資産を活用することが求められる。LiSTEE$_{TM}$はLinux向けの既存ソフトウェアをそのまま動作させることが可能であるため，セキュリティとは関係ない処理は既存の資産を最大限に活用しつつ，保護が必要なセキュリティ処理だけを分離することで，開発期間の短縮が期待できる。

5　あとがき

TrustZone®を利用して汎用OSとセキュリティアプリケーションを分離して実行するためのセキュアプラットフォームソフトウェア LiSTEE$_{TM}$の概要と有用性について述べた。組込み機器のセキュリティ確保が喫緊の課題となっており，当社はLiSTEE$_{TM}$の適用により，ハードウェアレベルの保護強度をソフトウェアで実現し，セキュリティ強化と組込み機器の開発期間短縮の両立を目指していく。

文　献

(1) MITRE. "Common Vulnerabilities and Exposures (CVE)". <http://cve.mitre.org/>, (accessed 2013-11-29).
(2) Loscocco, P. et al. "Integrating Flexible Support for Security Policies into the Linux Operating System". Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference. Boston, MA, USA, 2001-06, USENIX. 2001, p.29－42.
(3) ARM. "TrustZone". <http://www.arm.com/ja/products/processors/technologies/trustzone.php>, (accessed 2013-11-29).
(4) 金井 遵 他．“高速なOS切り替え機構を有する組み込み向けセキュアモニタLiSTEE”．2013年並列／分散／協調処理に関する北九州サマー・ワークショップ．北九州，2013-08，情報処理学会．2013-OS-126 (19)，p.1－8.
(5) 情報処理推進機構 セキュリティセンター．“組込みソフトウェアを用いた機器におけるセキュリティ”．<http://www.ipa.go.jp/files/000003116.pdf>，(参照 2013-11-29).

金井　　遵　KANAI Jun, D.Eng.
研究開発センター コンピュータアーキテクチャ・セキュリティラボラトリー研究主務，博士（工学）。プラットフォームセキュリティ技術の研究・開発に従事。電子情報通信学会，情報処理学会会員。
Computer Architecture & Security Systems Lab.

磯崎　　宏　ISOZAKI Hiroshi
研究開発センター 研究企画部参事。
ホームネットワーク及びセキュリティ技術の研究・開発に従事。
Research Planning Dept.