組込みシステムのセキュリティ

for SSR: オペレーティングシステムのセキュリティ機能に関する調査研究

櫻庭 健年* 宗藤 誠治† 進 博正‡ 藤田 卓志§ 森尻 智昭¶

2005 年 6 月

概 要

組込みシステムのセキュリティに関する脅威、要求事項、技術動向について述べる。ポイントを述べるにとどめるので、詳細はそこに紹介した参考文献をご覧いただきたい。
Keywords : 組込みシステム、セキュリティ

1 背景：適用の拡大とリスクの増大

組込みシステムは、家電、携帯電話、自動車、鉄道、ビルディングなどの核心部にあって、これらの機器やシステムを制御している。独立したシステムを自律的に制御するものだけでなく、近年は、モバイル、ユビキタス、情報家電といったコンセプトの下で組込みシステムをインターネットや無線ネットワークと結びつけることにより、様々な新サービスが可能になると期待されている。たとえば [1]、医療機関が患者のケアに IP ネットワークを利用する、高圧送電システムを IP ネットワークによって制御する、路側から自動車の速度を監視・制御する、複数のビルの空調機を制御して電力消費を平準化する、あるいは旅客機の主制御を IP ネットワークを介して行う、といった提案もある。

このように、組込みシステムの担う役割は拡大、かつ変質しつつある。これに伴って、組込みシステムのセキュリティの問題がクローズアップされてきている。「ハッカーの標的が、デスクトップ・パソコンから組込みシステムに広がっている」という指摘もある [2]。実際、組込みシステムを IP ネットワークに接続すれば、不特定多数からの攻撃の対象となり得る。また、組込みシステムでは機器の持ち主による、搭載したソフトウエアや暗号鍵の機器からの不正な取出しといった脅威があり得る。携帯機器では紛失や盗難の恐れがあり、組込みシステムに含まれている個人情報を保護する必要がある。適用されるシステムやサービスによっては、社会の基幹システムを支えたり、人命にかかわることもあり、組込みシステムのセキュリティが破られたときのリスクは増大している。

学会でも組込みシステムの研究が盛んになってきている。最近、情報処理学会では組込みシステムを主対象とする研究グループが発足した [3]。ACM では Embedded Systems を対象とした SIG [4] ができ、専門の論文誌が刊行されている。

以下では、組込みシステムについて、その関与者、制約条件、脅威と対策例について整理を試みる。

*株式会社日立製作所 システム開発研究所
†日本アイ・ビー・エム株式会社 東京基礎研究所
‡株式会社東芝 研究開発センター
§株式会社富士通研究所 ユビキタスシステム研究センター
¶東芝ソリューション株式会社 SI 技術開発センター

2　組込みシステムの関与者

組込みシステムには様々な関与者がおり、それぞれの立場でセキュリティ上の要求を持っている。例えば、次のような関与者がいる [5]。

部品の製造者

組込みシステムの主要な部品としてプロセッサや OS がある。これらはノウハウと特長技術の集成であり、これらを製造し、提供する者は、その複製や盗用を嫌う。従って、機器から直接プロセッサや OS などの技術情報の抽出や複製が容易であってはならない。

機器の製造者

機器の製造者は機器を制御するためのファームウエアを組み込んでいる。ファームウエアは機器の製造者の知的財産であり、プロセッサや OS と同様にファームウエアもその複製が容易であってはならない。

基本サービスの提供者

例えば携帯電話の場合、通話・通信サービスが基本サービスである。基本サービスの提供者はサービスの対価として収入を得ており、サービスの只乗り、すなわち不正な無料通話が脅威である。携帯電話は基本サービスを利用するための機器であるが、それを悪用して基本サービスを課金されずに利用できるようなことがあってはならない。

アプリケーションサービスの提供者

例えば携帯電話を通じて行われるサービスの提供者にとっては、サービスの不正な無料利用や、身元を偽っての利用などが脅威である。これらの防止のためには、セキュアなエンド-エンドの通信の保証が必要である。またサービスを受けたにもかかわらず、サービスを受けていないと言い張るような事後否認も防止する（non-repudiation）必要がある。

コンテンツ提供者

例えば携帯電話を通じて行われるサービスとして、音楽や映像のようなコンテンツの提供がある。コンテンツの提供者は、一度配布したコンテンツを再配布されることを嫌う。そのため、このようなコンテンツに関する知的財産権、著作権を保護する DRM（Digital Rights Management）の確立が必要である。

エンドユーザ

利用者は、基本サービスやアプリケーションサービスを正しく利用できることを望む。同時に、それらのサービスを通じて、例えば通信路や機器からユーザの個人情報が漏洩したり、機器に格納した情報が改竄・破壊されたりすることを望まない。ダウンロードしたプログラムを実行できるような場合は、そのプログラムがセキュアに実行できることが保証されなければな

らない。また要求したサービスが不履行となったり、内容が変わったりするようなことがあってはならず、そのようなサービス提供者による事後否認が防止されなければならない。さらに携帯電話などの機器の場合は、紛失したり盗まれたりしたときに、他人によって携帯機器から個人的な情報が抜き出される、といったことも回避したい。

3 組込みシステムの制約条件と脅威

組込みシステムのセキュリティの問題は、現在のデスクトップや企業のコンピュータシステムのセキュリティの問題よりも困難であり、その解決にはより多くの時間を要するという指摘もある [1]。その理由として、組込みシステムには以下に述べるような特有の制限事項があり、それらに伴ってセキュリティ上の問題が発生するが、これらの問題に対してデスクトップやサーバシステムのセキュリティ技術が必ずしも適用できないことがあげられる [5]。

3.1 環境条件

コスト

組込みシステムはコストの制限が厳しく、４ビットや８ビットのプロセッサが使われることも少なくない。このような環境では、たとえば暗号鍵を取り扱うことも難しい。開発者はセキュリティとコストのバランスを図らなければならない。

処理能力

組込みシステムはデッドライン制御をしていることが多い。そのため、CPU を余分に消費させるだけでシステムの動作を不安定にすることができる、という意味で脆弱である。

組込みシステムは、今後ネットワーク機能を備える必要があるが、現行のプロセッサは通信レートの向上、プロトコルの複雑化などに伴って必要な処理能力の増大に必ずしも追随していない。これらはルータ、ファイアウォール、ウェブサーバなどの高トラフィックな環境、PDA、携帯電話、IC カードなどの中程度の処理とメモリを要するシステムにおいて顕著である。

バッテリ

携帯機器など、組込みシステムにはバッテリで動いているものも多く、その場合、消費電力が大幅に制限される。中にはバッテリを１年間持たさなければならないようなシステムもある。このようなシステムに対しては、バッテリを不正に消費させるだけで攻撃者は戦果をあげることができる [6]。セキュリティ処理のために必要なエネルギーの伸びに対して, バッテリ寿命の延びは年間 5-8% と鈍い [7]。バッテリ寿命の延長、エネルギー消費を抑えたセキュリティ設計（プロトコルの最適化、専用セキュリティハードウエアなど）が必要とされている。

管理

組込みシステムには管理者がいない。従って、サーバシステムが前提としているような運用環境を仮定することができない。例えば、次のような問題がある。

- ファイアウォールのような防護システムを誰が動かすのか？
- セキュリティパッチを誰があてるのか？
- 情報家電を DDoS 攻撃などの踏み台として悪用されることを誰が防止するのか？
- 容易に類推されないようなパスワードを設定せよといった、ユーザの指導・監督を誰がするのか？

開発

多くの場合、組込みシステムの開発部隊は小規模であり、セキュリティの専門家を別途確保するということは少ない。一方、わずかなコードであってもセキュリティ保証が必要であるが、厳密なセキュリティ保証を含む開発手法はまだない。一般に、高信頼システムを構築するのは難しく、その上にセキュリティを保証するのはさらに難しい。攻撃者は、設計者が見落としたエラーを探し出して攻撃してくる。システムが複雑になると、設計、開発時にエラーを見逃す可能性が高くなる。

3.2 要求条件

柔軟性

携帯電話のような機器では、ひとつの機器で様々なサービスをサポートすることが多くなってきている。そのため、組込みシステムも様々な機能を備える必要がある。さらに組み込みシステムはこのような要求仕様の拡大、複雑化と同時に、実装コストの削減、開発期間の短縮を求められており、これらに容易に対応できる、柔軟なプラットフォームが必要となっている。

オープンスタンダード、オープンソースの利用

様々な環境とのインターオペラビリティの観点からは、旧世代の携帯電話、無線 LAN 対応を含む、各種のプロトコルのサポートが必要である。特に TCP/IP のサポートは必須であり、その開発コストを考慮して、従来のプロプライエタリな実装から、OS として Linux を採用することも多くなっている。

標準的セキュリティ機能のサポート

セキュリティの観点からは、上記のようなプラットフォームは攻撃者にとってもおなじみのものであり、脆弱性や脅威をそのまま引き継ぐ可能性が高い [8]。そこで組込みシステムとしても VPN、IPSec、SSL、WEP、DRM、などのセキュリティ機能やセキュアプロトコルのサポートが必要となっている。

安全なフィールドアップデート

サービスの追加やプロトコルの変更などに伴う新機能の追加や不良の修正などのために、機器の製品出荷の後もソフトウエアの頻繁な更新が必要となる。オープンシステムでは、配布済みのソフトウエアの更新についてはインターネット上に公開された修正パッチを適用という手法が確立しており、事実上、主要なセキュリティ確保手段となっている。

組込みシステムの場合、オープンシステムと同様のアプローチは困難である。たとえば、情報家電のユーザに頻繁かつ確実に修正パッチを当ててもらうことは不可能であろう。組込みシステムにネットワークでアクセスできる場合は、ソフトウエア更新のためのインタフェースを用意することが可能であるが、こういったインタフェースはしばしば悪用され、セキュリティ上の問題になりやすい。組込みシステムにおいても、ソフトウエアの更新は必要であり、安全な更新方法の確立が今後重要になる [9]。

ダウンロードソフトの安全性

インターネットアクセスがある場合は、ユーザが機器をインターネットに接続してソフトウエアをダウンロードし、使用することが可能となる。携帯電話などはその典型である。このような場合はダウンロードしたソフトウエアの安全性がセキュリティ上の問題になる。携帯電話のウイルス [10] も存在する。ウイルスやトロイの木馬を実行しないようにするのはユーザの責任ではあるが、機器の中の個人情報を含むデータへの不正なアクセスは防止するようにしたい。

持ち主による脅威

情報家電のような機器は、比較的安価で入手しやすいことが多い。すなわち、誰でも容易に機器の持ち主となることができる。機器の持ち主はその機器に対するほぼ無制限のアクセスが可能である。一般に機器の持ち主は、組込みシステムを仔細に調べることによって、サービスを不正に無料利用できるようになるかもしれない、といった動機を持ちうる。持ち主は組込みシステムが想定しなければならない最強の攻撃者と考えられる。

3.3 機器に対する攻撃

ここでは機器、ないし組込みシステムのハードウエアに対して行いうる攻撃についてまとめる。回路や動作時に発生する電磁波など、機器の本来のインターフェースでないところから情報を取り出したり改竄したりするので側面攻撃（side channel attack）と呼ばれることがある。組み込みシステム内の暗号鍵を容易かつ高い確度で推定する暗号解読攻撃として知られているものが多い。

バス探針

組込みシステムのプロセッサや回路を流れる信号を調べることによって例えばサービス利用に用いる暗号鍵のような、持ち主にも秘匿しておきたい情報を読み出すことができることがある。そこで重要情報はチップの外部に出現しないように設計する必要がある。しかし、オンチップの情報や信号を観測することも不可能ではなく [11]、これに対抗する「tamper resistant」なチップや IC カードが検討されている。

書き換え可能なデバイス

書き換え可能な記憶デバイス、例えばEEPROMにデータやコードを保存している場合、それらのデータの改竄の危険がある。フィールドアップデートのインターフェースを攻撃するほか、プログラムが格納されたデバイスを直接攻撃して、プログラムの不正な書き換えやセキュリティ制御フラグの無効化 [11] [12] のような攻撃の対象となりやすい。

処理時間解析（Timing Analysis）

暗号処理時間を計測すると、暗号鍵と処理時間の間の相関を元に、かなりの確度で暗号鍵を推定できることがある [13]。ソフトウエアによる正確な処理時間均等化は難しいことが指摘されている。

消費電力解析（Power Analysis）

処理に要する電流を計測することにより、暗号鍵に関する情報を得ることができることがある [7] [14] [15][16]。

Fault Induction

ICカードなどを放射線浴や高温浴下などで動作させることにより、プロセッサ内でのビット誤りを誘発させ、そのときの処理の結果を調べると、暗号鍵に関する情報が得られることがある。たとえばRSA処理の途中で1ビット誤りが発生したときの結果をもちいると、秘密鍵に関する一種の連立方程式が得られ、しかもこれが容易に解けることがある。実装にエラーがあっても、同様のことが起こりうる [17]。

電磁気解析

通信用機器でなくとも、動作にあわせて電磁波が漏れ出ることがあり、その中に重要情報が含まれることがある [18]。

残留情報

磁気ディスク上のデータは一旦消去・上書きしても元のデータが残留し、それなりの測定器具を使用すれば、元のデータを読み出すことが可能であることが知られており、そのため、徹底したデータ消去のためには7回にわたる乱数データの上書きが必要とされている。メモリデバイス、例えば電源を切った後のSRAM、消去後のEPROMやフラッシュメモリについても残留情報（remanence）が残り、その読み出しが可能であるという研究 [12] もある。

3.4　組込みシステムのセキュリティに対する公開された要求事項

OSDL

OSDL（Open Source Development Labs） [19] では通信事業に利用可能な高信頼性Linuxの仕様の要件をまとめている。特にCarrier Grade Linux Requirements Definition Version 2.0.2 [20] にはSecurity Requirementがある。v3.0はまだ公表されていない。

CE Linux

[21] には、CE Linux Forum における CE デバイスに対するセキュリティ要求と対応技術が簡単にまとめられている。

OMA

OMA（Open Mobile Alliance） [22] の Requirements for Common Security Enablers に大雑把なセキュリティ要件の記述がある。

TMP

TMP（Trusted Mobile Platform） [23] に、携帯端末のハードウエアとソフトウエアのセキュリティアーキテクチャ、およびプロトコルに関するドキュメントが公開されている。内容はやや抽象的である。

4 対策のアプローチ

FIPS 140-2

FIPS 140-2 では以下のような暗号モジュールの４つの物理的セキュリティレベルを定めている [24] [25]。当然、レベルの高いものが、安全性が高いながら、技術的に難しく、コストもかかる。設計者はコストとリスクのバランスを考慮しなければならない。

- レベル１　最低限の保護
- レベル２　シール、密閉などの破壊防止策
- レベル３　検出と対応のメカニズム
- レベル４　 Environmental Failure Protection と厳格な開発工程

TCG

TCG（Trusted Computing Group）はコンピューティングプラットフォームのための信頼性と安全性を持ったハードウエア、ソフトウエアの開発にを業界として取組もうとする業界団体である [26]。TPM（Trusted Platform Module）と呼ばれるセキュリティチップの仕様、および TNC（Trusted Network Connect）と呼ばれる高信頼なネットワーク接続アーキテクチャを提案、公開している。TCG のセキュリティアーキテクチャはソフトウエアによる、あるいはソフトウエアに対する攻撃に対処するために、ハードウエアを基礎とする保護を実現しようとするものである。

TPM 自体は受動的なデバイスであり、利用にはソフトウエアが必要であるが、書き込み保護すべきコードサイズを最小化し、システム全体の設計の自由度を高める為に、チップにソフトウェアの完全性保護機能と報告機能を備えている。これを基礎構造として、TPM がファームウエアの完全性を認証し、このファームウエアが OS の完全性を認証し、この OS がアプリケーションプログラムの完全性を認証する、といった完全性認証の階層を構成することによって、システム全体のセキュリティを構築する。

ソフトウエア品質の確保

ソフトウエアに関係するセキュリティホールは、多くの場合がソフトウエアの品質の悪さに起因している。そこで、セキュリティの高いソフトウエアを開発する手法、環境、ツールなどが提案されている。ソフトウエア開発ライフサイクル（SDLC）においては、早い時期でのセキュリティの作りこみが必要である。要求仕様、構造設計、コード開発などの各段階でのセキュリティ保証が有りうる。また、モデル・ドリブン・セキュリティと呼ばれる開発手法が提案されている [27] [28]。セキュリティに関する要求項目を織り込んだ形でモデル化から実装まで行うことにより、製品の品質を高めることが期待されている。

セキュリティ品質の保証の標準化

セキュリティに関連する製品のセキュリティ関連機能の品質を評価する手法と指標が情報セキュリティ国際評価基準（ISO/IEC15408、JIS X5070 [29] [30]）として標準化されており、これらは CC（Common Criteria [31]）と総称されている。

CC では、評価対象の製品のセキュリティに関する機能要件と保証要件を記述したセキュリティ設計仕様書（ST：Security Target）を作成し、これに基づいて評価を実施する。製品分野ごとにプロテクションプロファイル（PP：Protection Profile）と呼ばれる ST の雛形があり、ST の作成に利用することができる。

評価は各国の公的機関によって行われ、ある国で取得した評価結果は他の国でも通用する。機能要件の設計、開発、実装、テストについて、その厳密性や網羅性の程度に基づいて評価が行われ、合格すればセキュリティ保証レベル（EAL：Eavaluation Assurance Level）をが与えられる。EAL1（機能テスト）から EAL7（形式的検証済み設計とテスト）まであり、商用製品は EAL3 ないし EAL4 の取得が適当とされる。

EAL を取得した製品や利用可能な PP のリストが [30] や [31] にある。最近は EAL3、4、あるいは 4+ を取得した製品が増えている。組込みシステムの場合、適用分野によっては高い保証が必要とされることがある。たとえば、金銭の授受に使われることもある IC カードには EAL5+を取得しているものが多い。

組込みシステム向けアクセス制御モデル

アクセス制御モデルと呼ばれるものが、色々知られている [32] が、多くは実システムにおけるノウハウを抽象化し、客観的分析やより広い適用を可能としたものと考えることができる。組込みシステムに関連するセキュリティとして DRM があるが、DRM に取材したアクセス制御モデルとして UCON モデル [33] がある。UCON モデルは「利用を制御する」モデルとして、従来のアクセス制御とともに、DRM と Client Side Reference Monitor なる概念を含んでいる。

システム全体のセキュリティ設計

システムとしてのセキュリティはシステムを構成する部分（domain）の中で最も脆弱な部分に制約される。このため、組込みシステムの設計は、例えばネットワークやグラフィックス機能などの設計と同時にセキュリティの設計を行うべきであるという指摘がある。そのために、文献 [34] では、各部分を再構成可能であるように構成することを提案している。

また文献 [35] はハードウエアとソフトウエアのセキュリティを考慮した同時設計によって、セキュリティを実現することを提案している。FPGA を利用すれば、そのようなことが柔軟に実行可能であるとしている。

その他の対策アプローチ

各脅威については、脅威の説明の時に参照した文献に対策やそのアプローチが述べられていることが多い。ハードウエアに対する攻撃については [5] [36] [12] などが具体的な対策例を示している。

米国の軍事用組込みシステムについては MILS（Multi Independent Levels of Security）と呼ばれるセキュリティアーキテクチャがある [37]。特徴のひとつはアプリケーション間を厳密に分離することである。

参考文献

[1] P. Koopman, "Embedded System Security," IEEE, Computer Vol.37, No.7, pp.95-97, (2004)

[2] W. Webb, "組み込みシステムに忍び寄るハッカーの脅威," http://www.ednjapan.com/content/issue/2004/11/cover/cover.html

[3] 組込みシステム研究グループ, http://www.ipsj.or.jp/09sig/kenkyukai/newgr-emb.html

[4] Special Interest Group on Embedded Systems (SIGBED), http://www.acm.org/sigbed/

[5] S. Ravi, et al., "Security in Embedded Systems: Design Challenges," ACM, Transaction on Embedded Computing Systems, Vol.3, No.3, pp.461-491, (2004)

[6] T. Martin, et al., "Denial-of-Service Attacks on Battery-powered Mobile Computers," IEEE, Proceedings of the 2nd Pervasive Computing Conference, pp.309-318, (2004) http://www.ece.vt.edu/ tlmartin/power-secure/

[7] K. Lahiri, et al., "Battery-Driven System Design: A New Frontier in Low Power Design," Proceedings of International Conference on VLSI Design, pp.261-267 (2002) http://www.eecs.harvard.edu/ dbrooks/cs246-fall2004/vlsi02_battery.pdf

[8] Z. Shao, et al., "Defending Embedded Systems Against Buffer Overflow via Hardware/Software," IEEE, Proceedings of the 19th Annual Computer Security Application Conference, pp.352-361, (2003)

[9] E. Baissus, "Opportunities and Challenges for Linux in the Mobile Phone Industry," http://www.same-conference.org/same_2004/images/documents/Papers/Session 2/Session2_P2.pdf

[10] Internet Watch, "携帯電話に感染する初のウイルス「Cabir」- 露 Kaspersky が報告," http://internet.watch.impress.co.jp/cda/news/2004/06/15/3488.html

[11] R. Anderson, et al., "Tamper Resistance : a Cautionary Note," The Second USENIX Workshop on Electronic Commerce Proceedings, pp.1-11 (1996)

[12] S. P. Skorobogatov, "Semi-invasive Attacks - A New Approach to Hardware Security Analysis," University of Cambridge Computer Laboratory Technical Report 630 (2005), http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-630.pdf

[13] P. C. Kocher, "Timing Atacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems," Crypt'96, Springer, LNCS 1109, pp.104-113 (1996), http://www.cryptography.com/resources/whitepapers/TimingAttacks.pdf

[14] P.C. Kocher, et al., "Differential Poawer Analysis," Springer, Crypt'99, LNCS 1666, pp.388-397 (1999), http://www.cryptography.com/resources/whitepapers/DPA.pdf

[15] T. S. Messerges, et al., "Investigations of Power Analysis Attacks on Smartcards," Usenix, Proceedings of the 1st Workshop on Smartcard Technology, pp.151-161 (1999), http://www.usenix.org/events/smartcard99/full_papers/messerges/messerges.pdf

[16] K. Okeya, et al., "Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack,"

[17] D. Boneh, et al., "On the Importance of Eliminating Errors in Cryptographic Computations," Springer, Journal of Cryptology, Vol.14, No.2, pp.101-119 (2001)

[18] 新情報セキュリティ技術研究会（IST）, "電磁波セキュリティガイドライン," (2004), http://www.j-netcom.co.jp/ist/oshirase041124.html

[19] Open Source Development Labs, http://www.osdl.org/, http://www.osdl.jp/

[20] Carrier Grade Linux, http://www.osdl.org/lab_activities/carrier_grade_linux/

[21] CE Linux Forum, http://www.celinuxforum.org/

[22] Open Mobile Alliance, http://www.openmobilealliance.org/

[23] Trusted Mobile Platform, http://www.trusted-mobile.org/

[24] IPA, "スマートカードの安全性に関する調査報告書: 安全性評価の標準化動向," (1998), http://www.ipa.go.jp/security/fy11/report/contents/crypto/crypto/report/SmartCard/node57.html

[25] NIST, "Security Requirements for Cryptographic Modules," (2002), http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[26] Trusted Computing Group, https://www.trustedcomputinggroup.org/home

[27] ETH Zurich, "Model Driven Security," http://www.infsec.ethz.ch/people/seyboldb/mds/index

[28] D. Basin, et al., "Model Driven Security: from UML Models to Access Control Infrastructures," ETH-TR No.414, (2003), http://www.informatik.uni-freiburg.de/~tolo/pubs/mdac_tr.pdf

[29] IPA, "ISO/IEC 15408 入門," http://www.ipa.go.jp/security/jisec/about_cc.html

[30] NIST, http://niap.nist.gov/cc-scheme/index.html

[31] Common Criteria, http://www.commoncriteriaportal.org/

[32] IPA, "アクセス制御に関するセキュリティポリシーモデルの調査報告書," (2005), http://www.ipa.go.jp/security/fy16/reports/access_control/policy_model.html

[33] R. Sandhu, et al., "The UCON_{ABC} Usage Control Model," ACM, Transactions on Information and System Security, Vol.7 , No.1, pp.128-174, (2004)

[34] P. Schaumont, et al., "Domain-Specific Codesign for Embedded Security," IEEE, Computer, Vol.36, No.4, pp.68-74 (2003)

[35] J. Zambreno, et al., "SAFE-OPS: An Approach to Embedded Software Security," ACM, Transactions on Embedded Computing Systems, Vol.4, No.1, pp.189-210 (2005)

[36] P. Kocher, et al., "Security as a New Dimension in Embedded System Design," ACM IEEE, Proceedings of DAC 2004, pp.753-760, (2004)

[37] G. Uchenick, "MILS: Architecture for Assurance Systems," RTC magazine, (2005), http://www.rtcmagazine.com/home/article.php?id=100319

[38] TRON PROJECT, http://www.tron.org/