社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－134 (14)
2008－CSEC－ 40 (14)
2008／3／6

Cipher Mode On The Fly 暗号の考察

中嶋 純† 八百 健嗣† 小沼 良平† 中井 敏久†

†沖電気工業株式会社 研究開発本部 ユビキタスシステムラボラトリ
〒541-0053 大阪市中央区本町 2-5-7 大阪丸紅ビル 4階
E-mail: {nakashima264, yao282, konuma765, nakai365}@oki.com

概要 プログラムやデータをバイト単位またはワード単位で暗号化してメモリに保存しておき，命令フェッチ時にプロセッサ内部で動的に復号化するバス暗号化セキュリティ（本稿では on the fly 暗号と呼ぶ）技術がある．プログラムには分岐や繰り返しがあるために従来は暗号モードの適用がされていなかった．本論文ではジャンプ時に初期化処理を行うことで CFB 暗号モードを適用を可能としたバス暗号化方式を提案し，他のバス暗号化方式の改良方式との比較考察を行う．提案手法はデータバス幅が狭い場合であっても適用が可能で，なおかつ高い安全性を保つことができる．

Consideration of a Cipher Mode on-the-fly encryption

Jun NAKASHIMA† Taketsugu YAO† Ryohei KONUMA† and Toshihisa NAKAI†

†Ubiquitous System Laboratories, Corporate Research & Development Center,

Oki Electric Industry, Co., Ltd.

2-5-7 Honmachi, Chuo-ku, Osaka, 541-0053 Japan
E-mail: {nakashima264, yao282, konuma765, nakai365}@oki.com

abstract It is called "on-the-fly cryptography" that program codes are pre-encrypted by the bytes or words in the memory, and decrypted dynamically during execution. However, any cipher mode has not been applied to the "on-the-fly cryptography" because the programs have branches or/and loops. In this paper, we propose an on-the-fly cryptography which applies the CFB mode by initializing the register on branch/jump. And we also compare our proposal with other improved on-the-fly cryptographies. Even when the width of the data bus is narrow, our proposal is applicable and keeps the system secure.

1. はじめに

組み込み機器とは内部にマイコンやメモリを搭載している電子機器である．組み込み機器の一般的な製品の例としては携帯電話や AV 機器がある．また，家電製品の高機能化に伴い，近年出荷されている炊飯器や冷蔵庫も組み込み機器と呼べるようになった．

これまで組み込み機器のセキュリティの必要性はあまり認識されてこなかった．しかし近年ではインターネットに接続可能なデジタル情報家電などを初めとして，無線 LAN や，あるいは ZigBee™※などの無線ネットワークに接続する組み込み機器も登場し始めている．

さらに将来は，組み込み機器のプラットフォームも汎用化・オープン化することが予測されている[1]．汎用化・オープン化が進めば，ノ

※ ZigBee は Koninklijke Philips Electronics N. V.の登録商標です．

ウハウの蓄積が容易になることから，開発効率が向上し，さらに，組み込み機器ベンダーに依存しない汎用的なソフトウェアを開発できるようになるだろうと考えられる．

上記のように，組み込み機器のネットワーク化や，汎用化・オープン化に伴い，新たな脅威も発生する．組み込み機器がネットワークに接続されると，ネットワークを経由した，遠隔からの攻撃も可能となる．また，プラットフォームが規格化されると，ソフトウェア開発ツールの入手が容易になり，一般ユーザーが，組み込み機器の虚弱を解析するといった脅威も考えられる．

本論文では，この観点から，組み込み機器への解析耐性と攻撃困難性（耐タンパ性）を高める技術として，バス暗号化方式に関連した提案と考察を行う．バス暗号化方式では，プログラムやデータをバイト単位，または，ワード単位で暗号化してメモリに保存しておき，命令フェッチ時にプロセッサ内部で動的に復号化する．

この技術が有効であると考える理由を次に述べる．攻撃者は，攻撃するに先立ってまず，攻撃対象について理解することから始めると考えられる．場合によっては，リバースエンジニアリングを行うことも考えられる．

例えば，メモリの中身は，ROMのリーダー等を利用すれば読み出すことができる．さらに，回路上のデータバスにプローブを当て，ロジックアナライザを利用して，データバス上の信号を取得すれば，動作を観察することもできる．

バス暗号化方式は，メモリ内を暗号化し，実行するプログラムだけを動的に復号化するため，上述のような解析に対する耐性を高めるのに有用であると考える．さらには，攻撃者が不正なコードを実行させようとする場合でも，プログラムを暗号化して書き込む必要があるため，攻撃者が暗号化鍵を知ることなしに，不正なプログラムを作成し，実行させることが困難になる．

以降，第2章で関連研究について紹介し，第3章で提案法の説明と安全性に関する考察，課題について述べ，第4章でまとめる．

2. 関連研究

我々が着目する脅威は，機器の解析によって秘密情報が漏洩することや，攻撃の足掛かりを見つけ出されること，さらには，改竄により不正なコードを埋め込まれてしまうことである．以下では，上述の問題に対する対策技術について，幾つか紹介する．

2.1. TCGプラットフォーム

TCG(Trusted Computing Group)とは，信頼できるコンピューティング・プラットフォーム環境を構築することを目的とした業界団体である[2]．TCGでは，ハードウェア・ソフトウェアを連携させて，プラットフォームの改竄などを検知することを目的としており，BIOSやOSやソフトウェア等のハッシュ値を，耐タンパモジュールTPM(Trusted Platform Module)に保持しておき，起動時などにハッシュ値を計測・照合して改竄の有無を検査する．

2.2. ワンチップ化

ワンチップ化は，メモリをCPUコアと同一チップ上に実装することで，動作中に流れるデータへのプローブ解析を困難にする．ただし，バッファオーバフロー問題への対策や，秘密情報に対するアクセスコントロールを厳密にしなければ，安全とはいえない．また，利用するメモリサイズが大きい場合，外部に増設メモリを設けて利用することは，セキュリティの観点から推奨されない．したがって，プログラムサイズがメモリサイズよりも大きくなる場合はこの方法は適用できない．

2.3. On the fly(OTF)暗号

CPUと外部メモリの間に，暗号・復号化関数を挿入することで，バス上を流れるデータを暗号化するセキュリティ方式がある[3]．以下では文献[8]に倣いon the fly(OTF)暗号と呼ぶことにする．文献[4]で紹介されているon the fly暗号の基本構成を図1に示す．

(OTF暗号の基本構成)

CPUコアと暗号・復号化関数は，同一チッ

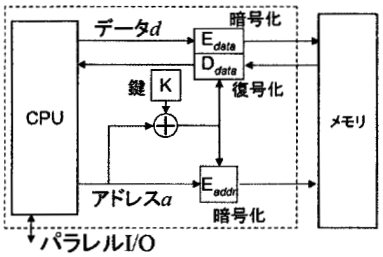

図１：on the fly 暗号

プ上に置かれる．このとき，鍵をストアするための記憶領域も，チップに含まれる．CPU が，メモリにデータを出力する際は，暗号化回路 E_{data} によって暗号化される．逆に，メモリからチップへ命令やデータを入力するときは，暗号化された状態で入力され，復号化回路 D_{data} によって復号化される．暗号化は，１バイト単位で，もしくは，プログラムの場合にはワード単位で，暗号化される．さらに，暗号化鍵 K に，メモリアドレス a（暗号化前）を，排他的論理和によって加えることで，鍵の探索攻撃への耐性を高めている．

(OTF の問題点)

1998 年に M.Kuhn は，on the fly 暗号が組み込まれたセキュアマイクロプロセッサチップ DS5002FP[5]に対して，選択暗号文攻撃を行い，暗号化されたメモリ上のデータを，平文で得ることに成功した[4]．Kuhn は，最終的に，８つのメモリアドレス毎の暗号化の変換テーブルを取得し，これを利用して，８つの命令からなるプログラムを暗号文で書き，このプログラムによりメモリ上のすべてのデータをパラレルポートに平文で出力した．

2.4. ブロック単位の on the fly 暗号 (OTF-BLK)

前述の点を鑑みて，複数の命令を束ねた上でブロック暗号を利用する on the fly 暗号の手法がある．本稿では，この手法を OTF-BLK と表記する．この方法を採っている AIGIS アーキテクチャでは，さらにブロックをまとめて，キャッシュブロックと呼ばれる情報単位で暗号化を行っている[6]．復号化した情報は，キャッシュメモリにストアされる．

2.5. 命令解釈の On the fly 暗号(OTF-FSM: Finite State Machine)

命令の解釈を置換し，有限状態機械を用いてこの置換テーブルを動的に切り替えることで，暗号ブロックの連鎖を実現する on the fly 暗号が，Monden らによって提案されている[7][8]．本稿では，この手法を，OTF-FSM(Finite State Machine)と表記する．この方法では，プロセッサ内の命令デコーダに復号器（状態遷移機械）をもたせており，“状態”が変化すると，同一命令コードであっても，命令の解釈方法が変わるようになっている．上述の“状態”は複数あり，どの“状態”に遷移するかは，入力される命令コードの種類によって決定される．このように，入力される命令コードの種類によって“状態”が決定される場合，分岐やジャンプが起こった場合に，暗号化時に利用した“状態”と，復号時に利用する“状態”とが，異なってしまう可能性がある．そこで OTF-FSM では，分岐やジャンプが起こった場合にも，特定の“状態”に遷移することで正しく復号されるようにしている．この特定の”状態”に遷移するために，分岐の合流ポイントの直前にダミーコードを挿入している．

図２を用いて，OTF-FSM において暗号化された命令を復号化する過程を以下で説明する．まず，状態の遷移について述べる．状態 $q0$→状態 $q1$ へと遷移するのは，状態 $q0$ において，add 命令が入力されたときである．一方，状態 $q1$→状態 $q0$ へと遷移するのは，状態 $q1$ において，add が入力されたときである．図２に示す暗号命令において，４行目の sub 命令が add 命令に置き換わり，５行目の add が sub に置き換わるところまでが，初期状態 $q0$ となる．５行目で add が入力されることに伴い，状態が $q1$ へと遷移する．オリジナルのプログラムコードでは，このまま loop へジャンプするが，状態が $q1$ のままなので正しく復号できない．しかし，ここでは６行目の add 命令（ダミーコード）により，状態が $q1$ から $q0$ へと遷移しており，loop へジャンプしても正しく復号で

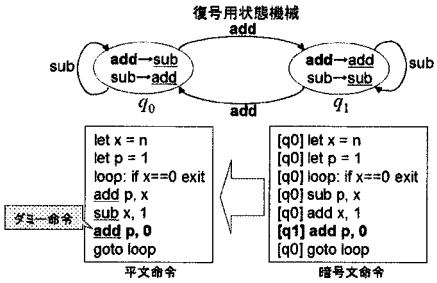

図２：命令解釈の OTF

きる．

2.6. 課題

ハードウェアサイズが小さい組み込み機器では，データバスのバス幅が限定される場合がある．例えば，データバスのバス幅が，8 ビットと小さい組み込み機器を想定する．例えば，上述の OTF-BLK 方式で，暗号化関数に 128 ビットブロック暗号を利用する場合，メモリに繰り返しアクセスし，128 ビットのデータを用意して初めて，OTF-BLK 方式が適用可能となる．また，データを用意している間は NOP を CPU に入力するような制御機構を追加で用意する必要がある．

一方，OTF-FSM はバス幅が 8 ビットに限定されていても追加の制御機構なく適用できる．しかし，解析耐性は必ずしも高いとはいえない．その原因は大きく二つある．

（1） オペコードは必ずしも変換されず，平文のままとなっているケースがある．全てのオペコードを置換すればこの問題は発生しない．しかし，テーブルのサイズは置換するオペコードの数と状態の数に比例する．さらに変換テーブルは秘密にしなければならないため，復号化回路内に実装する必要がある．ハードウェア量の制約のある組み込み機器では実際には置換されない命令も残る可能性が高い．これが解析時のヒントになり得る．

（2） オペコードの種類が多様でないため，オペコード同士の変換だけでは，暗号強度が弱い．（3.2 節で後述）

以上の課題を踏まえて，本稿では，データバスのバス幅が限定される組み込み機器においても適用可能な，on the fly 暗号を提案する．

3. 暗号モード on the fly 暗号 (OTF-CM: Cipher Mode)

3.1. 提案方式

本章では，on the fly 暗号において CFB 暗号モードを利用する方式（以下 OTF-CM と呼ぶ）を提案する．

図３に，６４ビットブロック暗号方式を用いた入力データ長８ビットの CFB 暗号モードの，暗号化と復号化を示す．図３において，K は鍵であり，DR_l はシフト量 l のシフトレジスタである．DR_l には最初，初期値が設定されている．暗号化関数 F はシフトレジスタ DR_l のデータを鍵 K で暗号化し，出力のうちの上位 8 ビットを平文 P の暗号化鍵として利用する．平文 P の暗号化は，F 関数の出力の上位 8 ビットとの排他的論理和によって暗号文 C となる．そして C はシフトレジスタ DR_l の下位 8 ビット部分にフィードバックされ，シフトレジスタ DR_l は 8 ビットだけ左へシフトする．復号化は暗号化と同じ手順をとる．すなわちまず F 関数が DR_l を鍵 K で暗号化して，その出力の上位 8 ビットと暗号文 C との排他的論理和により平文 P を得る．そして暗号文 C をシフトレジスタ DR_l へフィードバックする．

ところで，プログラムにはループや分岐が存在するのが一般的である．暗号モードを適用するためには，任意の命令からのジャンプ先となる命令を，考慮しなければならない．すなわち，直前の命令から順当に実行される場合と，ジャンプ命令から実行が移る場合のうち，どちらの場合でも，正しく復号できなければならない．提案法では，復号時にジャンプ命令を検出すると，シフトレジスタ DR_l を IV(Initial Vector) に初期化するというルールを追加することとし，さらに，ジャンプ先の対象となる命令の直前には，『直下の命令へのジャンプ命令』を追

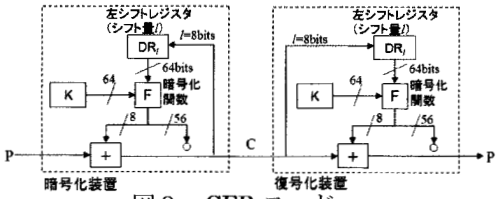

図３：CFB モード

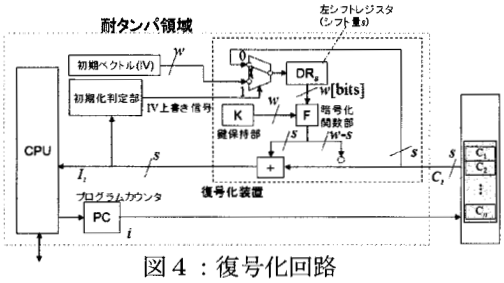

図４：復号化回路

加している．これにより，ジャンプ先となる命令は，必ずジャンプ命令から実行が移るようになり，この命令が復号化されるときは，必ずシフトレジスタが IV で初期化された状態となるようにしている．本方式におけるプログラムの暗号化処理は，大きく 2 つのフェーズからなる．

（暗号化フェーズ１）

ジャンプ先の対象となる命令の直前に，直下の命令へのジャンプ命令を追加する．

（暗号化フェーズ２）

1) $C_i = \mathrm{F_K}(\mathrm{DR}_s) \oplus I_i$
2) $\mathrm{DR}_s = \mathrm{DR}_s[56,1] \,\|\, C_i$
3) I_i が条件分岐，または，ジャンプ命令の場合，シフトレジスタ DR_s 全体を初期ベクトル IV で初期化する．
4) i を 1 だけインクリメントして 1)へ戻る．

ただし，$\mathrm{DR}_s[56,1]$は DR_sの 1(LSB)～56 ビット目(MSB-8 ビット目）のビットパターンである．‖はビット連結演算である．また $\mathrm{F_K}(\cdot)$はデータ・を鍵 K で暗号化することを表す．⊕は排他的論理和である．

図４に提案手法の復号化回路を示す．また，復号化手順を以下に示す．図中の s は，提案手法の入力データ長であり，w は内部暗号化関数のブロック長である．

1) プログラムカウンタの値 i に対応する命令 C_i が命令メモリから復号化装置へ入力される．
2) $I_i = \mathrm{F_K}(\mathrm{DR}_s) \oplus C_i$
3) $\mathrm{DR}_k = \mathrm{DR}_s[56,1] \,\|\, C_i$
4) I_i が条件分岐またはジャンプ命令の場合シフトレジスタ DR_s 全体を初期ベクトル IV で初期化する．

このように提案手法では，プログラム内のジャンプ命令への対応として，ジャンプ時にシフトレジスタを初期化することとし，ジャンプ先の命令への移行は，必ずジャンプ命令によって行われるようにすることで，CFB 暗号モードを適用可能とした．

3.2. 他の OTF 方式との比較

本節では，第 2 章で述べた，ブロック単位の on the fly 暗号 OTF-BLK と，命令解釈型の on the fly 暗号 OTF-FSM と，提案手法の OTF-CM の 3 手法について比較考察する．

OTF-FSM と OTF-CM は共に，1 命令単位で復号化できるため，データバス幅が 1 命令分であるという環境下であってもデータ読み出しに関する制御が不要であり，また復号化後のキャッシュも必要としない．よって，データバスの幅が限定されるような，組み込み機器に対しても適用可能である．一方 OTF-BLK では前述の通り追加の制御機構が必要となる．

次に，解析困難性について比較考察する．2.6 節で述べたように，OTF-FSM では，解析困難性に課題がある．2.6 節の（１）より，例えば，図２中の暗号文命令における状態 $q0$ が，最初から 5 行目まで連続しているように，ある状態はランダムな長さで，バースト的に続くことがわかる．このため，ある命令の周辺は，共通の状態である可能性が高い．さらに，平文のままとなっている命令もあることから，例えば，ソートアルゴリズムなどのよく知られたアルゴリズムであれば，推測される可能性がある．また 2.6 節の（２）について，近年安全とされる共通鍵暗号は，平文と暗号文の空間が 2^{128} という大きさであるのに対して，OTF-FSM では（一部の）命令コードの置換のみを行うため，

平文空間, 暗号文空間は最大でも命令セットの命令数で制限される. このことから, ある状態の置換テーブルを暗号化関数とみなした場合, 暗号化関数自体の暗号強度は低い. そこで OTF-FSM では様々な置換のパターンを用意して, それらを遷移させていくことで暗号強度の補強を図っている. しかし, 暗号化関数自体の強度が低いため, 例えば, もし暗号化されたプログラムと, それに対応する平文のプログラムの対が入手できた場合, これらの対応関係を元に置換テーブルを復元できる可能性がある. 一方, 提案法では, 十分高い強度の暗号方式を用いれば, このような場合であっても, 鍵を推測することは困難となる.

一方, OTF-CM では, 暗号化関数のビット長がワード長よりも長くできたことで, 必要十分な鍵長に対して十分大きな平文空間と暗号文空間を確保できるようになり, OTF-FSM 方式よりも暗号化強度が高くなる.

以上, 各 OTF 方式の特徴を表 1 にまとめる.

表 1 : 各方式の特徴 (組み込み機器に適用時)

方式 / 特徴	OTF-BLK	OTF-FSM	OTF-CM
入力データ長	暗号ブロック長	1 命令	1 命令
キャッシュ	必要	不要	不要
暗号強度	高	低	高

4. まとめ

本稿では組み込み機器への解析攻撃に耐性を持たせる手法として on the fly 暗号に着目して課題を探し, CFB 暗号モードを on the fly 暗号に適用する手法を提案した. 提案法は, データバス幅が制限されているような組み込み機器において, 高い暗号強度を保ちつつ, 複雑な制御機構を必要としないという利点がある. 今後の課題としては, ダミーコードによるコード量の増加を削減することが挙げられる.

参考文献

[1] (独)情報処理推進機構 セキュリティセンター, 組み込みソフトウェアを用いた機器におけるセキュリティ, 2006 年 4 月.

[2] 財団法人情報処理相互運用技術協会, 平成 16 年情報家電セキュリティ調査報告書, 2005 年.

[3] R.M. Best, "Preventing Software Piracy with Crypto-Microprocessors,"Proc. IEEE Spring COMPCON '80, pp. 466-469, San Francisco, 25-28 Feb. 1980.

[4] Markus G. Kuhn, "Cipher Instruction Search Attack on the Bus-Encryption Security Microcontroller DS5002FP," IEEE TRANSACTIONS ON COMPUTERS, VOL. 47, NO. 10, OCTOBER 1998.

[5] DS5002FP Data sheet http://datasheets.maxim-ic.com/en/ds/DS5002FP.pdf

[6] Suh, G., Clarke, D., Gassend, B., van Dijk, M. and Devadas, S. : AEGIS: Architecture for Tamper-evident and Tamperresistant Processing, Proc 17th International Conference on Supercomputing (ICS2003), pp.160-171, June 2003.

[7] Monden, A., Monsifrot, A, and Thomborson, C. "Tamper-Resistant Software System Based on a Finite State Machine," IEICE TRANS. FUNDAMENTALS, VOL.E88-A, NO.1 pp.112-122, January 2005.

[8] 門田 暁人, Clark Thomborson: "ソフトウェアプロテクションの技術動向(後編) - ハードウェアによるソフトウェア耐タンパー化技術" No.46, No.5, pp.558-563, May 2005.