MCMPを利用したソフトウェア保護方式に関する一考察

末松俊成[†] 今井秀樹[††]

現在のコンピュータ/ネットワーク社会において，ソフトウェア（プログラムやディジタルデータなど）の違法コピーや不正使用を防ぎ，著作者の権利を保護することには困難な問題が多く残されている．このような現状を改善するには Multipurpose Crypto Microprocessor（MCMP）が有効である．MCMP は，内部に暗号化/復号機能を内蔵したマイクロプロセッサであり，ソフトウェアを安全に実行することができる．しかし，現在すでに多くのコンピュータが普及しており，このすべての CPU を MCMP に切り替えることは容易ではない．本稿では，MCMP が必要な背景について説明し，MCMP を導入する 1 つの手段として，MCMP を搭載した拡張ユニットを用いたソフトウェア保護システムについて考察する．

A Consideration on Software Security Based on MCMP

Toshinari Suematsu[†] and Hideki Imai[††]

In the current computer/network environment, it is very difficult to defend software (programs and digital data, etc.) against infringement or illegal copy for the protection of developer's copyrights. The Multipurpose Crypto Microprocessor (MCMP) is an effective solution for this problem. The MCMP is a microprocessor which contains encryption/decryption units in it. Therefore it can execute software securely. However, too many computer systems are already spreaded in the world, and so it is very difficult to replace all CPU with MCMP. In this paper, we explain the background of the necessity of MCMP then we consider the software protection system which uses an MCMP expansion unit as a measure to introduce a MCMP to the market.

1. はじめに

コンピュータ上で使用されるソフトウェア（プログラムやデータなどの総称）を，改ざん，違法コピーなどの不正使用から防ぐことはコンピュータ社会における重要な課題である．

近年 MP3 という音楽データ圧縮形式で音楽が不正に配布されていることが社会的な問題としてとりあげられている．この問題に代表されるように，従来はソフトウェアを物理的な機構によって防ぐことはあまり行われておらず，ユーザの倫理観に訴えるか，法律によって取り締まることで抑止するのみであった．著作権の侵害という問題は，一般ユーザにはあまり認知されておらず，ソフトウェアの違法コピーなどを行っているほとんどのユーザは罪悪感を感じていないのが現状であり，これらの対策だけでは十分な効果を得ることは難しいと考えられる．

このような状況の中，最近はコンピュータやネットワーク上のセキュリティに関する意識も高まってきており，音楽や映像などのディジタルコンテンツ（本稿ではこれらもソフトウェアとして扱う）を中心に，電子認証や暗号化技術が採用されてきており，ソフトウェア保護の実用化が進みつつある．しかし，コンピュータ上で実行されるソフトウェアなどについては，いまだに多くが無防備に流通しているのが現状である．誰でも手軽にコピーしたり不正使用したりできる環境でありながら，不正使用の禁止を訴えても効果は限られている．このため，ソフトウェアが保護されていることをユーザが一目で理解できるような環境を整えることが先決である．そのうえで初めて倫理や法律が有効に機能すると考えられる．

ソフトウェア保護に有効な一手段として MCMP（Multipurpose Crypto Microprocessor）が知られている[1]．これは，文献 1) では CMP と呼ばれていたが，それ以前に提案されていた専用の組み込みプログラムだけを実行できる簡単な構成の CMP[2),3)]と区別する

† ソニー株式会社大崎東 TEC
Sony Corp. Osaki East Tec.
†† 東京大学生産技術研究所
Institute of Industrial Science, University of Tokyo

表 1 外部攻撃と内部攻撃の比較
Table 1 Comparison of outside attack and inside attack.

	外部攻撃	内部攻撃
攻撃者	第三者	ユーザ自身
防御者	ユーザ，コンピュータ，ソフトウェア	コンピュータ，ソフトウェア
攻撃の対象	ユーザが所有するソフトウェア，データ，通信内容など	他者が権利を有するソフトウェアなど

ため，本稿では MCMP と呼ぶことにする．MCMP は暗号化されたソフトウェアをチップ内部で復号しながら処理することが可能なもので，コンピュータ上で実行するプログラムをハッカーの攻撃や不正使用などから保護するのに適している．しかし，現在すでに従来の CPU を組み込んだ多くのコンピュータが普及しており，このすべての CPU を MCMP に切り替えることは容易ではない．

本稿では，MCMP が必要な背景として，ソフトウェア保護に関する現状の問題点について触れ，ソフトウェア保護の強化が必要であることを示す．次に，現状のコンピュータ社会に MCMP を導入する 1 つの手段として，MCMP を搭載した拡張ユニットを用いたソフトウェア保護システムについて考察する．

2. 外部攻撃と内部攻撃

ソフトウェアの保護について論じるには，まずどのような形で攻撃されるかを把握しておく必要がある．コンピュータ上のソフトウェアに対する攻撃の形態は，コンピュータの外部から行われる外部攻撃とコンピュータの内部から行われる内部攻撃に分類することができる．

2.1 外部攻撃

外部攻撃とは，攻撃の対象となるコンピュータを直接操作しない第三者が，ネットワークなどを通して攻撃を加えるものである．例としては，電子メールなど他人が送受信するデータを盗み見または改ざんすることや，ネットワークを通して他人のコンピュータに侵入し，ソフトウェアの改ざん，データの盗み見，不正コピーを行うことなどがあげられる．外部攻撃においては，各ユーザのプライバシーが侵害されたり，各ユーザのソフトウェアが第三者の攻撃によって損害を被る．これを防ぐため，コンピュータとそのユーザが一体となって外部からの攻撃に対抗する．一例として，暗号化プログラムによってネットワーク上に流すデータを暗号化するなどの対策があげられる．このケースでは，暗号化のプロセスを外部の攻撃者によって解析される可能性は低く，プログラムによる暗号処理を行っても問題は少ない．

2.2 内部攻撃

これに対し，コンピュータを直接操作するユーザ自身が攻撃者となって自分の所有するコンピュータ内のソフトウェアを攻撃する内部攻撃という形態が考えられる．例としては，ソフトウェアの違法コピーや有料ソフトウェアを無料で使用できるように改ざんすることなどがあげられる．この攻撃の防御は，外部攻撃の場合に比べ非常に困難である．なぜなら，暗号化によってデータを保護しようとしても，その処理を行うプログラムは攻撃者であるユーザの手元で実行される．現状のコンピュータにおいては，攻撃者であるユーザがコンピュータと暗号に関する専門知識を持ち，さらにデバッガなどのプログラム解析ツールを使いこなす能力の持ち主であった場合，プログラムを使用した暗号処理によって秘密を守り通すことは原理的に不可能といえる．暗号化や復号などのプロセスを解析され，秘密鍵や復号データなどの重要なデータを盗み出される危険が高い．**表 1** に外部攻撃と内部攻撃の比較を示す．

外部攻撃は，第三者からの攻撃によって個人のプライバシーが侵害されるなどの被害を受けるものであるため，一般的な関心も高い．一方，内部攻撃はユーザ自身が利益を享受するために行うものであり，被害を受けるのはソフトウェア製作・販売者という一般ユーザにとっては身近ではない団体または個人であるため人々の関心は薄い．しかし著作権保護の観点からすると内部攻撃を防ぐことは非常に重要である．

3. MCMP の概要

内部攻撃からソフトウェアを保護するのに有効なのが MCMP である．暗号化機能を内蔵したプロセッサは，従来にも提案されているが[2),3)]用途が限定されるもので，一般的なコンピュータに使用されている CPU に置き替えることは困難なものであった．ここで紹介する MCMP は，一般的なコンピュータの CPU として使用することが可能で，かつ鍵管理方式などのセキュリティを強化し，複雑なアプリケーションにも対応できるようにしたものである．

図 1 は MCMP の一構成例である[1)]．内部に一般の CPU と同じ機能を持ち（以降内部 CPU と呼ぶ），内部 CPU から入出力されるデータを暗号化/復号する

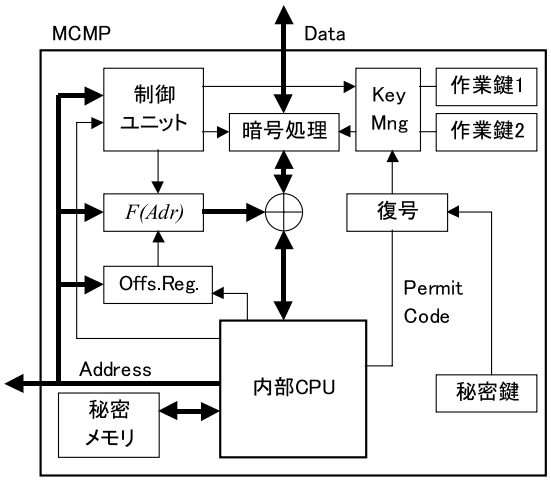

図 1 MCMP 構成例
Fig. 1 A structure of MCMP.

暗号処理ユニットが置かれる．この暗号処理ユニット用の作業鍵（図 1 では 2 つ）のレジスタがあり，その切替えなどは Key Mng（鍵管理ユニット）が管理する．

また暗号処理ユニットと内部 CPU の間には暗号を強化するためのデータスクランブラが置かれる．これは内部 CPU の機械語やデータなどに特徴のあるパターンが繰り返し現れると暗号を解読されやすくなる恐れがあるため，それを防ぐことを目的とする．たとえば，データの中に同じ値が延々と続く部分があったときに，それを暗号化した結果も単調なパターンの繰返しになるようだと，元の値を見破られる恐れがある．データにスクランブルをかけておけば，このような場合でも暗号化した結果をランダムデータのように見せかけることができ安全性が増す．図 1 の例ではメモリ上のデータのアドレスから $F(Adr)$ によって計算したデータと CPU のデータバスに現れるデータとの排他的論理和をとることによってスクランブルをかける．$F(Adr)$ としてはハッシュ関数などが利用できる．$F(Adr)$ に入力されるアドレスは，暗号化したときと復号するときとで同じでなければならない．しかし，一般のコンピュータではソフトウェアはリロケータブルであり，メモリ上のどこにロードされるかは決まっていない．このため，ソフトウェアの先頭などの基準位置からの相対アドレスを $F(Adr)$ の入力とする必要がある．図 1 のオフセットレジスタ（Offs.Reg.）はこのアドレスの調整を行うためのものである．

制御ユニットは内部 CPU の指示により，暗号処理やスクランブル処理をコントロールする．暗号処理に必要な作業鍵は外部から取り込む必要があるが，このとき何の保護もないと，攻撃者に作業鍵を盗み見られる危険が高い．これを防ぐためには作業鍵を暗号化して MCMP に取り込む必要がある．図 1 の MCMP は内部に個々の MCMP 固有の秘密鍵を保管しており，これに対応する公開鍵で暗号化された作業鍵を内部で復号できるようにしている．この階層鍵方式によって安全に作業鍵を取り込むことができる．秘密鍵はシリアル番号と同じように，個々の MCMP 固有のものであり，原則として同じ秘密鍵を持つ MCMP は存在しないものとする．

以上に示した暗号化機能により，MCMP の外部においてはソフトウェアを暗号化された状態でしか見ることができず，プログラム解析ツールを使用したとしても復号鍵を知らないとその処理内容を理解することはできないため，ソフトウェアのセキュリティを飛躍的に向上できる．

図 1 の秘密メモリは MCMP 内部に置かれるメモリで，MCMP の電源が切られても内容が保持される．また，MCMP の外部バスとは切り離されており，外部から直接アクセスすることはできず，MCMP が持つ秘密メモリ管理用の専用コマンドを通してのみアクセスできるという特徴を持つ．

これは，アプリケーションが秘密として扱うデータの一部を MCMP 内部に保存するためのものである．たとえば攻撃者がアプリケーションを騙す目的で記憶装置上のすべてのファイルを不正に書き換えた場合でも MCMP 内部のデータは書き換えられずに残るため，これを利用して不正な変更があったことを検出することができる．秘密メモリに関する詳細については文献 1) を参照されたい．

以上に示したように，暗号化機能と秘密メモリを持った MCMP を利用することによって，ソフトウェアを厳重に保護できるシステムを構築することが可能である．

4. MCMP を使用したソフトウェア保護システム

MCMP によってソフトウェアの保護を実現するには，MCMP を内蔵するコンピュータを開発し普及させなければならない．しかし現在数多く流通している主要な CPU に置きかえられる MCMP を開発し，それを搭載するコンピュータを普及させることは，そう簡単なことではない．MCMP の普及を促すためには，より簡便に導入できる方法を考える必要がある．ここでは，従来のコンピュータに MCMP を搭載する拡張ユニットを追加したシステムによってソフトウェアを

保護する方式について考える．

MCMP 拡張ユニットに最低限必要な機能は以下のとおりである．

- メイン CPU または周辺機器とデータをやりとりする機能．
- メイン CPU の制御下で MCMP 用の暗号化プログラムをダウンロードし実行する機能．

上記の条件が満たされるものであれば，拡張ユニットをコンピュータに接続する手段は何でもよい．例としては，コンピュータの汎用拡張スロット，PC カードスロット，USB（Universal Serial Bus）や SCSI（Small Computer System Interface）などの汎用インタフェースが考えられる．ほかにコプロセッサのような形態も考えられるが，マザーボード上に専用ソケットを用意しなければならないので現状のコンピュータへの適用は困難である．この中で拡張スロットは，バスマスタモードで各種のメモリや記憶装置に直接アクセスできる機能を実現でき，かつデータの転送速度の面でも優れており，最も MCMP 拡張ユニットに適していると考えられる．以降では拡張スロットに増設する MCMP 拡張ユニットについて話を進めることにする．

一般にソフトウェアは，処理を行う主体となるアプリケーションプログラム（以降アプリケーションと呼ぶ）と，それによって生成または処理されるデータとに分類される．例として表計算ソフトウェアについて考えると，表計算ソフトウェアがアプリケーションで，表計算ソフトウェアによって作成されるワークシートがデータである．アプリケーションは当然著作権によって保護される対象であるが，第三者が作成したワークシートも保護の対象となりうる．表計算のワークシートの場合，レイアウト，関数，マクロなどを駆使して複雑な機能を実現することが可能であり，創作物としての条件を満たしていると考えられる．実際，特定の業務に有益なワークシートを作成して販売している例も見られる．

このようにアプリケーションとデータが独立した著作物となるものは多く，他の例としては，ワードプロセッサとそれで作成された小説などの文学作品，音楽・映像の再生アプリケーションとそれらのデータ，CAD アプリケーションと CAD データなどがあげられる．また，辞書や地図などのデータが主体の製品は，現在はこれらを検索したり表示したりするアプリケーションとデータが一体になって販売されていることが多いが，将来的には検索・表示のアプリケーションとデータベース間のインタフェースを統一し，ユーザが好みに合わせて検索・表示アプリケーションとデータを別々に選択できる環境を構築することが期待される．

以上のように，今後のネットワーク社会においては，アプリケーションと個々のデータを別々の著作物として保護することが必要であり，それぞれの暗号化鍵も当然違うものでなければならない．以降ではアプリケーションとデータを別々の鍵で保護する方法について説明する．

ユーザは MCMP 拡張ユニットを組み込んだコンピュータを所有しており，その MCMP のシリアル番号を i とする．以降これを $MCMP_i$ と呼ぶことにする．この $MCMP_i$ は秘密鍵 K_{CSi} を内蔵しており，これと対になる公開鍵 K_{CPi} が存在する．個々の MCMP に組み込まれる秘密鍵は MCMP メーカが管理するもので，ユーザはもとより他の誰も知ることができない．これに対し公開鍵は，MCMP のシリアル番号さえ知っていれば誰でも入手することができるものとする．

アプリケーション m（APP_m）は固有の鍵 K_{Am} で暗号化され，$E_{Am}(APP_m)$ の形で配布される．ここで m は個々のアプリケーションに対応する識別番号とし，$E_{XY}(D)$ とは，鍵 K_{XY} でデータ D を暗号化することを示すものとする．ユーザは所有する $MCMP_i$ の公開鍵 K_{CPi} を開発元に知らせる．開発元はこれで K_{Am} を暗号化して P_{Ami} を作成しユーザに送付する．この作業鍵 K_{Am} を暗号化した P_{Ami} を Permit Code と呼ぶことにする．

$$P_{Ami} = E_{CPi}(K_{Am}) \tag{1}$$

ユーザは P_{Ami} を $MCMP_i$ に取り込ませることで APP_m を実行できる．P_{Ami} の添字 m と i はこの Permit Code が APP_m と $MCMP_i$ の組合せに対してのみ有効であることを示している．

次に，このアプリケーションの下で使用される著作物を $DATA_n$ とし，n は個々のデータに対応する識別番号とする．$DATA_n$ は K_{Dn} で暗号化され，$E_{Dn}(DATA_n)$ の形で配布されるものとする．ユーザは，データの開発元に $MCMP_i$ の公開鍵 K_{CPi} を知らせ，暗号化されたデータとそれを使用するための Permit Code である P_{Dni} を入手する．

$$P_{Dni} = E_{CPi}(K_{Dn}) \tag{2}$$

P_{Dni} の添字 n と i は，この Permit Code が $DATA_n$ と，$MCMP_i$ の組合せに対してのみ有効であることを示している．

以上のアプリケーション，データ，各種の鍵，Permit Code の流れを**図 2** に示す．ユーザからは公開鍵である K_{CPi} が各開発者に渡され，開発者からは暗号化

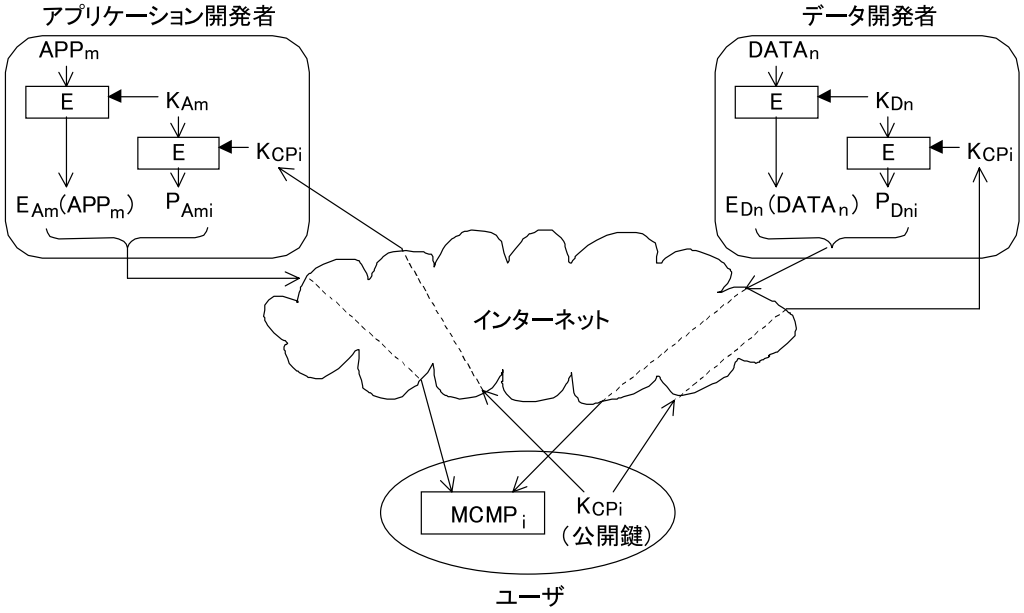

図 2 データの流れ
Fig. 2 Data flow.

されたアプリケーションまたはデータと Permit Code がユーザに渡される．図の中で開発者を示す枠の中の E と書かれた四角い枠は暗号化処理を表し，横から入力される値が鍵を示す．なお，通常はアプリケーションまたはデータの暗号化方式と作業鍵の暗号化方式とは異なるが，この図の中では特に区別していないので注意されたい．

さらに，これらの暗号化アプリケーションとデータを MCMP 拡張ユニットに転送したり，実行を制御したりするための制御ソフトが必要である．この制御ソフトはメイン CPU の下で動作する暗号化されない通常のプログラムである．以上のソフトウェアを次の手順によって実行する．

まずメイン CPU は制御ソフトによって E_{Am} (APP_m) と P_{Ami} を $MCMP_i$ を載せた MCMP 拡張ユニットに転送し，$MCMP_i$ 内部の作業鍵レジスタ 1（図 1 参照）に復号鍵 K_{Am} をセットした後，アプリケーションを起動する．次にアプリケーションがデータを要求するところでメイン CPU は制御ソフトによって $E_{Dn}(DATA_n)$ と P_{Dni} を MCMP 拡張ユニットに転送し，内部の作業鍵レジスタ 2（図 1 参照）に復号鍵 K_{Dn} をセットする．アプリケーションは，MCMP 内部で自身の復号鍵 K_{Am} と $DATA_n$ の復号鍵 K_{Dn} を適宜切り替えながら処理を実行する．本方式の構成を**図 3** に示す．図 3 のうち背景色の付いた部分は暗号化されていないソフトウェアを示している．

この例のように 2 種類の鍵を高速に切り替えて使い分けるためには，図 1 に示したように作業鍵を MCMP の中に複数保管できるようになっているとよい．鍵の管理は，図 1 中の鍵管理ユニット（Key Mng）が行う．作業鍵のレジスタをさらに増やすことも可能なので，コストと効果のバランスを考慮して決定する．

図 3 に示すように，保護しなければならない APP_m と $DATA_n$ は，暗号化された状態のまま MCMP 拡張ユニットに転送され，MCMP 内部で復号されながら処理される．制御ソフトは暗号化ソフトウェアを MCMP 拡張ユニットに転送したり，実行を開始したりするだけのものであるため保護する必要はない．

本方式において，APP_m と $DATA_n$ はともに暗号化によって保護されており，それぞれ対応する Permit Code（P_{Ami}，P_{Dni}）を持つ正規ユーザ以外は使用できない．これらの暗号化されているソフトウェアのコピーは自由に行うことが許され，ユーザ間でアプリケーションやデータを受け渡ししても構わない．たとえばアプリケーションを保管していたハードディスクが壊れたような場合に，そのアプリケーションのコピーを知人からもらうことが合法的に行える．アプリケーションやデータの入手は基本的には無料でよいが，流通経路によっては媒体の代金や必要最小限の手数料などの支払いが必要な場合もありうる．

これに対して Permit Code の入手は通常有料となる．これを紛失した場合は，Permit Code を再発行し

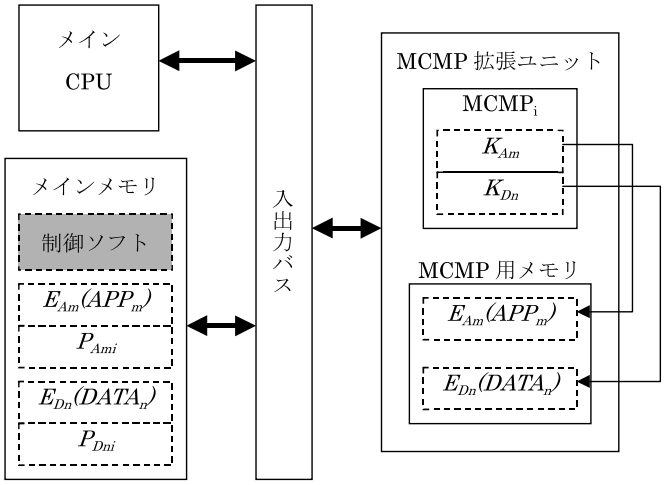

図 3 MCMP 拡張ユニットを使用したシステム例
Fig. 3 A system using MCMP expansion unit.

てもらう必要がある．MCMP のシリアル番号とソフトウェアの種類から，一度発行した Permit Code であることが分かれば，再発行は可能であろう．

本方式は，ソフトウェアを所有するためではなく，使用するために対価を支払うという意味で，超流通[4)~6)]の 1 つの形態といえよう．このシステムをさらに発展させ，使用回数や使用時間などに応じた課金を行うような，より超流通に近いシステムを構築することも可能である．

以上では，説明を簡略化するためアプリケーションとデータともにすべて暗号化されている場合について述べた．暗号化されたアプリケーションは，暗号化しないものに比べて処理速度の面で不利になることは避けられない．また，拡張ボード上で行える処理には制約もあるため，セキュリティを本当に必要とする部分だけを MCMP 拡張ボードで実行し，その他の部分はメイン CPU で実行するような構成にすれば，実行速度をあまり落とさずにセキュリティの向上を図ることが可能である．

たとえば，辞書ソフトウェアにおいて，辞書のデータベースとその検索プログラムを MCMP 拡張ボードが受け持ち，画面の表示などのユーザインタフェース部分をメイン CPU が受け持つといった方法が考えられる．

5. 安全性に関する検討

本方式は，ハッカーなどによる内部攻撃にも耐えることを目的としたもので，保護を必要とするソフトウェアは，すべて MCMP 拡張ユニットに転送し MCMP の下で暗号化したままの状態で処理される．MCMP の耐タンパ性が十分と仮定すれば，拡張ユニット上では攻撃者はいかなる方法をもってしても復号されたデータを見ることができない．メイン CPU が処理する部分は保護されないが，前章で示した例では拡張ユニットで実行するソフトウェアのロードと実行を制御するだけであるので，保護すべきソフトウェアの安全性とは無関係である．

ソフトウェアの実行をメイン CPU と MCMP 拡張ユニットで分担する場合には，秘密にしたい情報の取扱いに十分留意して設計しなければならない．

図 2 に示したように，開発者とユーザがネットワークを通してやりとりする情報は，暗号で保護されたものか公開情報であるため，ネットワーク上においては安全である．

ここで安全性を比較するために，MCMP 拡張ユニットではなく，外部から与えられたデータを単純に暗号化/復号する機能だけを持つ単機能の暗号処理拡張ユニットを使ったシステムについて考えてみる．ここで暗号処理拡張ユニットは MCMP 拡張ユニットと同様に拡張スロットに増設するものとして話を進める．一見この暗号処理拡張ユニットを用いたシステムでもよさそうに思えるが，本方式との間には安全性の面で大きな開きがある．このシステムでは，ソフトウェアを実行するのはメイン CPU であるため，暗号化されたままでは処理できない．暗号化されているソフトウェアと復号鍵を暗号処理拡張ユニットに送り，そこで復号されたソフトウェアを CPU のレジスタやメインメモリなどに置いて処理することになる．このため，メイン CPU と暗号処理拡張ユニットが通信する入出力バスには必ず復号されたデータが現れるので，攻撃

者はそれをモニタして待つだけでよい．またデバッガなどのツールを用いることでも，レジスタやメモリ上に復号されたデータを簡単に見ることができる．したがって，従来のコンピュータに暗号処理拡張ユニットを追加したとしても安全性の向上はほとんど期待できず，暗号化・復号の処理を高速化できる程度の意味しか持たない結果となってしまう．以上に示したように，MCMPが暗号化したままソフトウェアを処理できることが，ソフトウェア保護の面で非常に重要であることが分かる．

本方式においては，ソフトウェアが供給元からユーザに渡り，MCMP拡張ユニットで実行されるまでの過程の間，すべて暗号化されたままの状態で流通・処理されるので，これらの過程においてはソフトウェアの暗号化に採用される暗号の強度以外の問題はない．したがって本方式への攻撃として考えられるものは次のとおりである．

(1) K_{Am}，K_{Dn} を解読する．
(2) K_{CSi} を解読する．
(3) MCMPを分解し内部を解析する．

これらの攻撃に対する強さは，MCMPの耐タンパ性，アプリケーションならびにデータの暗号化方式，MCMPが採用する公開鍵方式に依存し，普通のハッカーなどに対しては十分安全であろう．

なお，安全性に関する以上の議論はMCMP拡張ユニット上に置かれるソフトウェアについてのみ適用されるものである．音楽や映像データは，出力装置を通してアナログ信号に変換され人間の知覚できるものとなる．このアナログ信号に変換された後のデータをコピーされることを防ぐことはできない．また，現在のコンピュータの構成では，ビデオメモリや，サウンドカードのA/Dコンバータへの入力が暗号化されたままだと処理できないため，復号したデータを送る必要があり，ここでデータを盗み取られてしまう可能性がある．これを防ぐには，コンピュータを構成するすべてのデバイス間の通信を暗号化するか，構成するすべての部品を耐タンパモジュールの中に封じ込める必要があるが，現実的とはいえない．このように音楽や映像が主体のソフトウェアに関しては，元データの不正使用を防げたとしても，それから二次的に生成されたデータの不正使用を防ぐことができない場合が存在する．

ただし，以上の攻撃において，元データと同じ形式に戻すためにはディジタル化や圧縮などの手間のかかる作業が必要であり，かつこれによって得られたものが元データと完全に同じになるわけではないので，MP3のように元データを簡単にコピーして利用できるような場合に比べると十分効果があると考えられる．

一方，辞書や文学作品のように情報を視覚で伝えるソフトウェアの場合は，ある時点においてビデオメモリに展開されて画面に表示される情報は全体のごく一部にすぎず，ビデオメモリを通してすべての情報をコピーするにはかなりの手間を要する．またすべての画面をコピーされたとしても，元データが持つデータベースなどの機能は受け継がれないため，攻撃によって受ける影響は小さいといえよう．CADデータについても同様のことがいえ，ビデオメモリに展開されるのは，元のベクトルデータから作り出したある特定の条件下の映像にすぎず，元のベクトルデータの価値に置き換わるものではない．

このように，データ形式の違いにより不正行為によって得られる効果が違うという側面があるものの，いずれの場合においても本方式によって安全性が現状よりも向上するのは間違いなく，ソフトウェア保護に有効である．

6. むすび

本稿では，コンピュータを操作するユーザ自身が攻撃者となる内部攻撃を防ぐことは現在のコンピュータの構成では困難であること示し，次にMCMP拡張ユニットを使用したソフトウェア保護システムの例を示した．本方式の特徴は次のとおりである．

- 従来のコンピュータにMCMPを搭載した拡張ユニットを追加することでソフトウェア保護システムを実現する．MCMP対応のコンピュータを普及させるよりも導入が簡単である．
- メインCPUは暗号化されたソフトウェアをMCMP拡張ユニットに転送し，セキュリティを必要とする処理をこの拡張ユニットに任せる．MCMP拡張ユニット内では暗号化されたままの状態でソフトウェアが処理される．
- アプリケーションとデータを異なる鍵で暗号化し，別個の著作物として保護することができる．これらの鍵の切替えを効率良く行うために，MCMP内部に2つの作業鍵を保管できるようにしている．
- 暗号化されたアプリケーションやデータは，基本的に無料で自由に配布することができる．また，これらのソフトウェアのコピーをとることも自由である．
- 利用者はPermit Codeを入手することによって，ソフトウェアを自身のMCMP上で使用できるようになる．これによりソフトウェアの使用を正規

ユーザに限定できる．

これらの特徴から，本方式はネットワーク上などでソフトウェアを配信するのに適している方式といえよう．

本方式では，従来のコンピュータに MCMP 拡張ユニットを追加するため，その分コストアップになる．最新の CPU はゲート数が 4,000 万以上といわれており，これに対して図 1 に示したような暗号処理機能を追加するにはその 10 分の 1 も必要としないと予想される．MCMP が大量に生産され，量産効果が期待できるようになれば CPU との価格差はかなり縮められよう．MCMP 拡張ユニットは MCMP とメモリを積んでおり，コンピュータのマザーボードの簡略版のようなものであるが，外部機器とのインタフェースなどを必要とせず部品点数が少ないため，マザーボードよりは十分安くできると考えられる．

このような導入段階を経て，将来マザーボード上のメイン CPU の代わりに MCMP を使用するようになれば，コストアップは CPU と MCMP の差額だけになるため，コンピュータ全体のコストに比べるとほとんど無視できる程度にすることも期待できよう．

本稿では，最も防御が困難と考えられる内部攻撃を防ぐという観点から MCMP の有効性について説明してきたが，当然ソフトウェアのセキュリティに関わる他の用途にも有効である．MCMP は暗号化/復号の機能を内蔵しているため，暗号化/復号処理をソフトウェアで行うよりも安全かつ高速に行うことができる．

参 考 文 献

1) 末松俊成，今井秀樹：CMP（Crypto Microprocessor）の一構成方法とその応用例，ISEC98-8 (1998).
2) Best, R.M.: Crypto Microprocessor for Executing Enciphered Programs, United States Patent, 4278837 (July 1981).
3) Best, R.M.: Crypto Microprocessor That Executs Enciphered Programs, United States Patent, 4465904 (Aug. 1984).
4) 森　亮一，河原正治：歴史的必然としての超流通，情報処理学会超編集・超流通・超管理のアーキテクチャシンポジウム論文集，Vol.94, No.1, pp.67–76 (1994).
5) 森　亮一：超流通の構造，防御，人々の利益—定義と基本式，信学技報，ISEC94-13 (1994).
6) 末松俊成，今井秀樹：超流通ラベルリーダを使用しない超流通システムの一構成法，SCIS96 講演論文集，SCIS96-14B (1996).

(平成 12 年 10 月 24 日受付)
(平成 13 年 6 月 19 日採録)

末松　俊成

昭和 56 年弘前大学理学部物理学科卒業．同年ナカミチ（株）入社．平成 10 年ソニー（株）入社．この間，光記録，ディジタル信号処理，誤り訂正符号，情報セキュリティ，ソフトウェア流通等に関する研究・開発に従事．工学博士．電子通信情報学会，情報理論とその応用学会各会員．

今井　秀樹（正会員）

昭和 41 年東京大学工学部電子工学科卒業．昭和 46 年同大学院博士課程修了．工学博士．同年横浜国立大学講師（工学部電気工学科）．昭和 47 年同助教授．昭和 59 年同教授（工学部電子情報工学科）．平成 4 年東京大学教授（生産技術研究所）．郵政省通信総研客員研究官等兼任．現在に至る．この間，符号理論とその応用，暗号と情報セキュリティ，スペクトル拡散方式，データ圧縮，移動通信等の研究に従事．昭和 50 年，平成 2 年電子情報通信学会著述賞，平成 3 年同論文賞，米澤ファウンダーズ・メダル，平成 6 年同業績賞，平成 10 年 IEEE シャノン 50 周年記念論文賞，平成 11 年韓国順天郷大学名誉博士号等受賞．著書「符号理論」（昭晃堂），「情報数学」（昭晃堂），「情報と符号の理論」（岩波書店），「情報理論」（昭晃堂），「符号理論」（電子情報通信学会），「暗号のおはなし」（日本規格協会），「明るい暗号の話」（裳華房），「Essentials of Error-Control Coding Techniques」（Academic Press）等．電子情報通信学会理事，監事，同「基礎・境界ソサイエティ」会長，IEEE 情報理論ソサイエティ理事，国際暗号研究学会（IACR）理事，情報理論とその応用学会会長等を歴任．IEEE Fellow．