マルチメディア通信と分散処理　101－16
コンピュータセキュリティ　12－16
（2001．2．21）

再構成可能計算機を用いた不正コピーの防止

井熊 徹
静岡大学大学院
理工学研究科

西垣 正勝
静岡大学
情報学部

曽我 正和
岩手県立大学
ソフトウェア情報学部

田窪 昭夫
三菱電機株式会社
情報システム製作所

ディジタルコンテンツの不正コピーを完全に防止するためには，復号に関連する一連の情報をユーザから隔離しなければならない．この考えに基づき，すでにCMPおよびスクラッチング方式などの不正コピー防止技術が提案されている．しかしこれらの方式においては，CPU にハードウェア的な拡張を施す必要があり，柔軟性が乏しい．そこで本稿では，CPU の一部をプログラマブルデバイスにより実装することにより，不正コピー防止のためのハードウェア機構をソフトウェア的に再構成することが可能なCPUを作成することを提案する．本方式により，各メーカが独自に，コンテンツに応じた最新の不正コピー防止方式を採用することが可能となる．

キーワード：不正コピー防止，再構成CPU，CMP，スクラッチング，再暗号化

Copy Protection on A Reconfigurable CPU

Tohru Ikuma
Graduate school of
Science and Engineering,
Shizuoka University

Masakatsu Nishigaki
Faculty of Information,
Shizuoka University

Masakazu Soga
Faculty of Software and
Information Science,
Iwate Prefectural University

Akio Takubo
Mitsubishi
Electric Corp.

To protect digital contents from illegal duplication, some hardware modification is required. However, the hardware-based copy protection systems have little flexibility. This paper proposes to construct copy protection systems on a reconfigurable CPU. This system is always ready to implement the modules for any protection scheme on the reconfigurable CPU. Therefore, we can use the best suitable scheme to protect a given content.

Keywords: copy protection, reconfigurable CPU, CMP, scratching, rights enforcement encryption

1. はじめに

ディジタル社会においては，ディジタルコンテンツがネットワーク経由で流通し，使用に応じて課金される．ここで，不正コピーの完全な防止は重要な命題である．既に，ディジタルコンテンツの不正コピー防止のために様々な暗号化技術が用いられている．コンテンツは暗号化された形で配信されるため，復号鍵を持たない第三者がこれを使用することはできない．だが，購入者（正規ユーザ）には復号鍵が知らされるため，一度，コンテンツが正規ユーザの手に渡って復号されてしまうと，その後の不正コピーに対しては無防備となる．すなわち，通常の暗号化技術では流通過程におけるセキュリティを維持することしかできない．電子透かし[1]を用いることにより，コンテンツが正規ユーザの手に渡った後の不正コピーを監視することがある程度は可能となる．しかし，電子透かしは不正コピーの抑止力として働くものであり，不正コピーを防止する技術ではない．また電子透かしには，情報中に膨大な冗長成分を持つ画像には適用しやすいがプログラムには適用しにくい，コンテンツがクラッカーにより盗み出されて不特定多数にばらまかれた場合に犯人を特定する力はない，などの問題がある．

結局，不正コピーを完全に防止するためには何らかのハードウェア的な方策が不可欠となる．これに対する一つの有効なアプローチとして，CPU全体またはその中の一部をハードウェア的に封印した上で暗号化技術を活用するという方法が提案されている[2]-[7]．この方法においては，コンテンツは暗号化されたままの状態で保存されることになり，実行直前に復号され，使用される．そして，暗号化コンテンツを復号するための鍵をはじめとする復号に関連する一連の情報や結果は，何らかの方法でユーザから隔離される．

しかしながら，ハードウェアによる不正コピー防止方式は柔軟性に欠ける．そこで本稿では，CPUの一

部をプログラマブルデバイスにより実装することにより，不正コピー防止のための機構を再構成することが可能なCPUを作成することを提案する．本方式においては，再構成部分の回路情報が正規購入者にのみ，購入者のCPUの公開鍵により暗号化された上で送られることになる．CPUの秘密鍵およびプログラマブルデバイスにより実装される部分がハードウェア的に封印される．本方式により，各メーカが独自に，コンテンツに応じた最新の不正コピー防止方式を採用することが可能となる．

2. 専用ハードウェアによる不正コピー防止方式

2.1 CMP

CMP(crypto microprocessor)[2]-[4]は従来のCPUの機能に加え，暗号化／復号機構，秘密メモリを内蔵したプロセッサである．暗号化／復号機構と秘密メモリはハードウェア的に封印されている．ソフトウェアはハードディスクやメモリ内など，CPUの外部においては暗号化された状態で保存される．CMPはソフトウェアを順次，部分的に復号し，秘密メモリにデータを書き込みながらこれを実行していく(図1)．秘密メモリ内のデータは正規ユーザでさえ参照することはできない．すなわち，いかなる者もオリジナルのソフトウェアを見ることはできない．ソフトウェアの暗号化は各ソフトウェアごとに用意された作業鍵によって行われる．通常，速度向上のため，この暗号化には共通鍵暗号が用いられる．すなわち，作業鍵は共通鍵となる．作業鍵はCMPごとに割り当てられる公開鍵によって暗号化され，ソフトウェアとともにユーザに配信される．CMPの秘密鍵はCPU製造時にCMP内に封印され，ユーザにも知らされることはない．したがって，ユーザであっても暗号化されたソフトウェアを復号することはできない．作業鍵の暗号化に使用された公開鍵に対応する秘密鍵をもつCMPのみがソフトウェアを復号し，利用することが可能である．

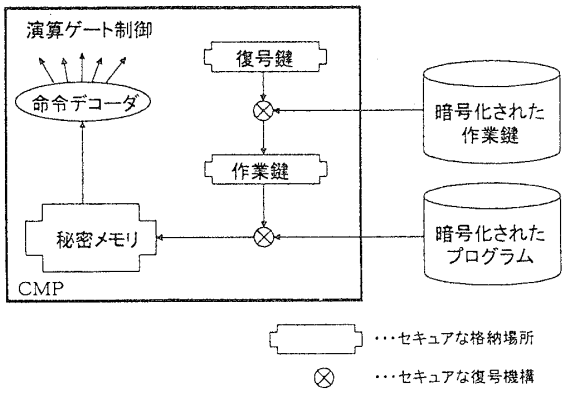

図1：CMPの復号機構

2.2 プログラムのスクラッチングと動的復号

スクラッチングは，プログラムの一部を意図的に改竄(暗号化)することにより，不正コピーを無効化する方法である(図2)．プログラム中の改竄部分はプログラムに付けられた傷(スクラッチ)に相当する．傷を付けられたプログラムをスクラッチトプログラムと呼ぶ．傷を修復するためのクリーナ情報がなければ，スクラッチトプログラムをそのまま実行したとしても正常な動作には至らない．図2に示されるように，クリーナ情報は傷を修復するための定数とスクラッチング箇所を特定するための先行命令から成る．クリーナ情報はユーザの使用するCPUの公開鍵により暗号化された上で送られる．この公開鍵に対応する秘密鍵はCPU製造時にCPU内に封印されている．よって，正規ユーザであってもクリーナ情報を不正に復号することはできない．プログラムはスクラッチトプログラムの状態で配信され，ユーザのファイルシステムに格納される．更にプログラムは，実行の段階で主メモリにロードされた時点においても傷付いたままである．CPUにはクリーナ情報を用いてプログラム中の傷(暗号化部分)を動的に復元(復号)するための機構が付加されており，プログラムの傷は命令が実行される際に命令デコーダの直前にて修復され，正しく実行される．

動的復号機構を図3に示す．スクラッチング方式においては，CMPの秘密メモリに相当する安全な記憶装置として，セキュアレジスタ，セキュアROMが用意されている．CPUの秘密鍵はCPU製造時にセキュアROMに封印されている．また，暗号化されているクリーナ情報を復号してセキュアレジスタに格納するファームウェア命令が用意されている．スクラッチトプログラムが実行されると，各命令は命令レジスタに読み込まれ実行されるとともに，過去数命令が命令SRに留まる．通常は命令SR内の命令群は先行命令に一致せず，比較器から出力される修復フラグはオフであり，命令レジスタの命令がそのまま命令デコーダに送られ，実行される．一方，命令デコーダ内にスクラッチト命令が読み込まれた場合には，命令SR内の命令群と先行命令が一致し，比較器から出力される修復フラグがオンとなり，命令は修復器において定数とXORがとられて正規の命令となって命令デコーダに送られる．ここで，修復フラグがオンになった

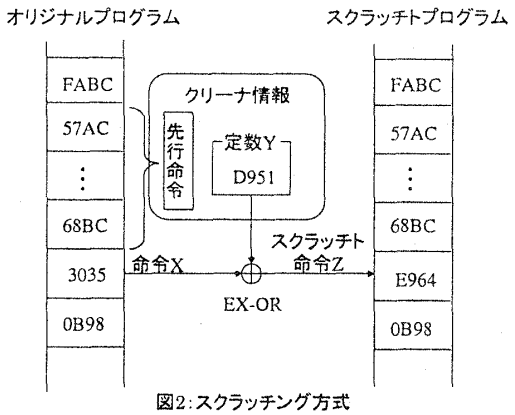

図2:スクラッチング方式

時点の命令SRの内容がユーザにスキャンされると先行命令が漏洩するため，命令SR，修復フラグに関してもセキュアレジスタに格納する必要がある．オリジナルプログラムはいかなる記憶装置上にも残らない．動的復号のプロセスは基本的にはXORのみであり，そのオーバヘッドも無視できる．また，本方式の動的復元機構は既存のアーキテクチャに上位互換的に付加可能である．

2.3 問題

CMPやスクラッチング方式により不正コピーを完全に防ぐことが可能となるが，ハードウェアによる不正コピー防止方式はその導入が困難であるという短所を持つ．CMPにおける暗号化／復号機構やスクラッチング方式における動的復号機構など，各方式専用のモジュールはCPUの開発時に組み込まれていなければならない．このため，独自の不正コピー防止方式を開発し，これを採用することができるのは，CPUの開発能力のあるメーカに限られることになる．また，実装される不正コピー防止方式が一種類に固定化されてしまうため，

- 新たに開発されたより良い不正コピー防止方式をリアルタイムに採用することは難しい，
- プログラムや画像など，コンテンツにはその種類に応じた適切な不正コピー防止方式が考えられ得ると思われるが，コンテンツに応じて方式を変更することはできない，
- もし，その不正コピー防止方式がブレイクされてしまった場合には，全てのコンテンツに被害がおよぶ(同種のコンテンツにおいても異なる不正コピー防止方式を採用することができれば，ある方式がブレイクされてもその被害の範囲を抑えることができる)

という柔軟性に関する問題を抱える．

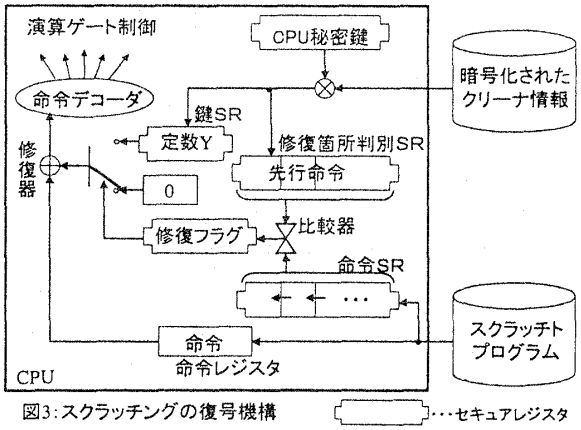

図3:スクラッチングの復号機構

3. 再構成 CPU による不正コピー防止方式

本稿では，CPUの一部をプログラマブルデバイスにより実装することにより，不正コピー防止のための機構を再構成することが可能なCPUを作成することを提案する．本方式により，CPUメーカに限らず，コンテンツプロバイダを含む全てのメーカが独自に，コンテンツに応じた最新の不正コピー防止方式を採用することが可能となる．すなわち，本方式は専用ハードウェアによる不正コピー防止方式の利便性に加え，安全性をも向上させる．本章ではその構成と動作を示す．

3.1 CPU の拡張

本方式においては以下に示すハードウェアの拡張がCPUに施される．この拡張は通常のノイマン型計算機に対して上位互換的に行なうことが可能である．すなわち，この拡張にはCPUの機種に依存する部分はない．よって，コンテンツプロバイダが不正コピー防止方式を採用する際に，ユーザのCPUの種類を意識する必要はない．

1. 各CPUの入出力部はプログラマブルデバイスにより実装される．現在プログラマブルデバイス上に構成されている回路をユーザがスキャンすることはできない．
2. 各CPUには製造時に公開鍵暗号方式の鍵のペアが割り当てられる．これらを「CPU公開鍵」，「CPU秘密鍵」と呼ぶ．CPU秘密鍵はCPU内に秘密裡に封印され，正規ユーザにも知らされることはない．また，CPU公開鍵は認証局により署名された上で公開される．ここで，認証局の持つ公開鍵暗号方式の鍵のペアを「公的公開鍵」，「公的秘密鍵」と呼ぶ．
3. 各CPUは，CPU公開鍵により暗号化された構成

情報に従ってプログラマブルデバイスを再構成するための命令を持つ．この命令を「DecryptLOAD命令」と呼ぶ．DecryptLOAD命令を実行するのはCPUとは別途に用意された専用装置である．DecryptLOAD命令によりこの専用装置が起動され，CPUに封印されているCPU秘密鍵を用いて構成情報が復号された上で，この構成情報に従いプログラマブルデバイスが書き替えられる．また，この命令を使用せずにプログラマブルデバイスを再構成することはできない．

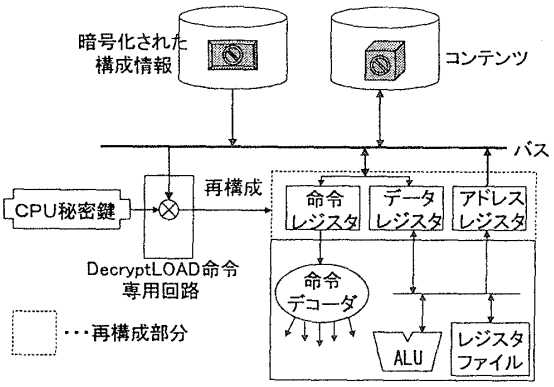

図4：再構成CPU

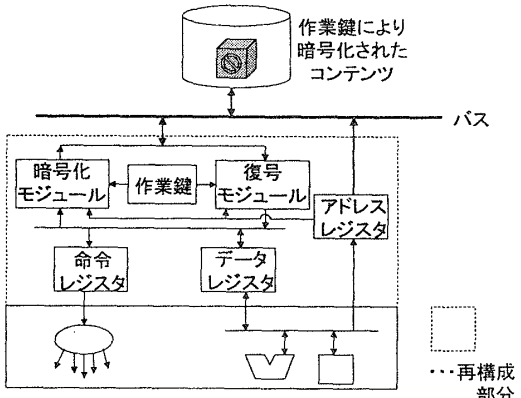

図5：CMPとして再構成されたCPU

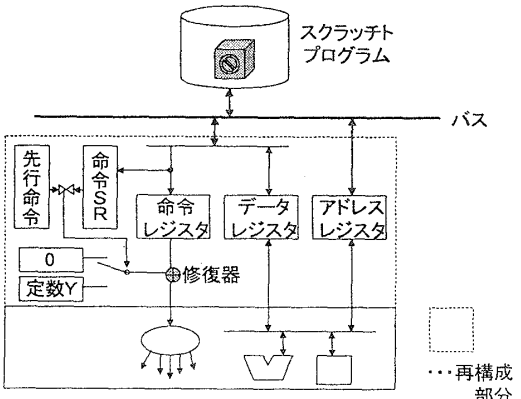

図6：スクラッチング方式として再構成されたCPU

再構成CPUの概念を図4に示す．また，再構成CPUによりCMP，スクラッチング方式を実装した状態を模式的に図5，図6に示す．

以降，CPU公開鍵，CPU秘密鍵をCp，Cq，公的公開鍵，公的秘密鍵をEp，Eqと記す．また，鍵Kによりメッセージを暗号化したものをK{M}と記す．プログラマブルデバイスの構成情報をCIと記す．

3.2 コンテンツの暗号化と配送

コンテンツプロバイダが不正コピー防止方式としてスクラッチング方式を採用した場合を例にとり，コンテンツプロバイダにより作成されたコンテンツと構成情報が購入者に渡されるまでの流れを説明する（図7）．なお，ここではコンテンツプロバイダはコンテンツ製作者とコンテンツ配布者を兼ねているが，実際には両者が別機関であっても構わない．購入者とプロバイダ間の通信は一般のネットワーク回線を用いて行われるが，署名付メッセージを用いるなどの方法により両者間のメッセージの正当性は保証されているという前提を置く．

1. コンテンツプロバイダはコンテンツを作成する．
2. コンテンツプロバイダは任意の不正コピー防止方式を採用することができる．ここではコンテンツがプログラムであるとし，プロバイダはスクラッチング方式を採用したとする．
3. コンテンツプロバイダはプログラムに傷を付け，スクラッチトプログラムとクリーナ情報を生成する．
4. コンテンツプロバイダはスクラッチング方式の動的復号機構に応じたCPUの構成情報CIを作成する．CPUの構成情報は具体的には図6の点線枠内の回路となる．クリーナ情報である先行命令および定数の値は，修復箇所判別SR，鍵SRの初期値として与えられる．
5. 購入者はコンテンツプロバイダに購入の意志とともに，使用する計算機のCPU公開鍵をEq{Cp}の形で伝える．
6. コンテンツプロバイダはEq{Cp}を公的公開鍵Epで復号することで購入者のCPU公開鍵Cpを得る．この時，認証局による署名により，メーカは確かにCPU公開鍵であることを確認した上でCpを得ることが可能である．
7. コンテンツプロバイダは構成情報CIを購入者のCPU公開鍵Cpにより暗号化してCp{CI}

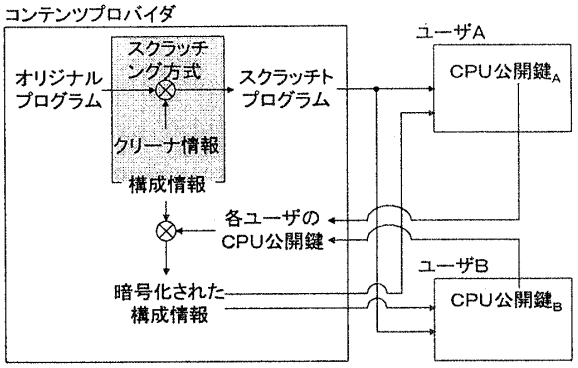

図7:コンテンツの配信

を作り，スクラッチトプログラムとともに購入者に配信する．また，必要に応じて課金する．

本方式においてはCPU全体をプログラマブルデバイスによって再構成するのではなく，CPUの入出力部分のみを書き換える．よって，コンテンツプロバイダはCPU全体を開発する必要はなく，購入者に配布される構成情報のサイズも小さく抑えられる．また，一般的にプログラマブルデバイスによって構成される回路は動作速度が遅いという問題点も軽減される．

上記例ではコンテンツプロバイダがCPUの構成情報を作成しているが，実際には，不正コピー防止方式を専門に開発するメーカの存在を考えることもできる．このメーカは各種不正コピー防止方式に対するCPUの構成情報を公開(販売)する．

手順6において，コンテンツプロバイダは購入者が申請した公開鍵を無条件に信頼してはいけない．購入者が公開鍵pと秘密鍵qのペアを知っていた場合，pによって構成情報が暗号化されて購入者に送られると，購入者が不正に構成情報を復号することが可能となってしまうためである．購入者がコンテンツプロバイダに示した公開鍵がCPU公開鍵であることを確認するために，CPU公開鍵には認証局の署名が付けられている．

従来のスクラッチング方式[6]においては，クリーナ情報である先行命令および定数の値は，その都度，修復箇所判別SR，鍵SRにロードされる．これに対し，本方式においては回路の構成情報とクリーナ情報をまとめてしまうことが可能である．手順4では，先行命令および定数の値が修復箇所判別SR，鍵SRの初期値として与えられている回路を構成情報として生成している．これにより，当該スクラッチトプログラムに対応するクリーナ情報がセット済みの動的復号機構を，直接，プログラマブルデバイス上に構成している．これは，CPUがCMPとして再構成された場合にも同様であり，CPUの構成情報内に作業鍵の値をセットしてしまうことが可能である．

3.3 CPUの再構成およびコンテンツの復号

購入者が構成情報を復号し，コンテンツを実行する手順を示す．前節に続き，不正コピー防止方式としてスクラッチング方式が採用されている場合を例にとり説明する．

1. 購入者はDecryptLOAD命令により，構成情報をプログラマブルデバイス上に再構成する．DecryptLOAD命令は，CPU公開鍵Cpにより暗号化されている構成情報Cp{CI}をCPU内部のCPU秘密鍵Cqにより復号した上で，CIに従ってCPUを再構成する．
2. 購入者はスクラッチトプログラムを実行する．スクラッチトプログラムはメモリにロードされた後に順次CPUに読み込まれる．CPUの入出力部分がスクラッチングの動的復号機構を含んだ形に再構成されているため(図6)，スクラッチトプログラムは正常に実行される．

従来のスクラッチング方式[6]においては、クリーナ情報などはセキュアレジスタに格納されていた。本方式においては、現在のプログラマブルデバイスの回路情報をユーザがスキャンすることができないような実装方式を採っており、クリーナ情報などが外部に漏洩することがないようになっている。

4. 考察

4.1 マルチタスクへの対応

現在，CPUはマルチタスク処理の形態で利用されることがほとんどである．本方式によりコンテンツごとに異なる不正コピー防止方式を採用することが可能となるが，プログラマブルデバイス上に一種類の回

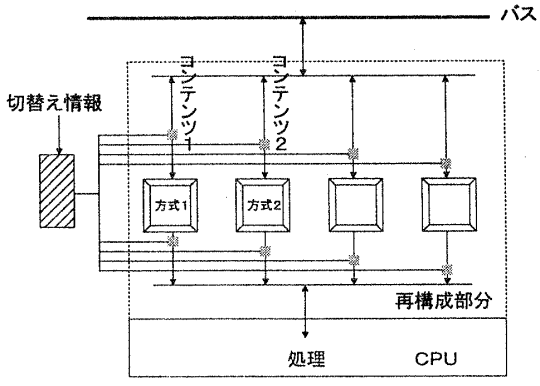

図8:マルチタスクへの対応

路を展開することしかできないと，タスクの切替えごとに当該コンテンツの不正コピー防止方式に応じて再構成部を書き換えなければならない．通常のプログラマブルデバイスの書き換えには時間を要するため，これはCPUの処理速度に対して大きな負荷となる．これを避けるため，プログラマブルデバイスには複数の回路を展開することができるようにする必要があると思われる[8]．タスクの切替えに応じて使用する回路を変更する(図8)．回路の切替えは単純な論理ゲートのみによって行うことが可能である．切替え命令はOSが発する．

4.2 ユーザビリティ

本方式では，構成情報は各ユーザの所有するCPUに割り当てられたCPU公開鍵によって暗号化され，配信される．しかし，このような暗号化にはユーザの使用できる計算機が固定されてしまうという問題が残る．各 CPU には固有の秘密鍵が封印されているため，ある CPU 公開鍵によって暗号化された構成情報を別の CPU で使用することは不可能である．すなわち，コンテンツの正規ユーザであっても，自身の所有する2台目以降の計算機においては，そのコンテンツを利用することができない．この問題は IC カードを用いることにより解決できると思われる[7]．

4.3 本方式の専用機への適用

4.1節および4.2節に示した問題は汎用計算機における問題と言える．用途が固定された機器においてはこれらの問題を考慮する必要がないことが多く，専用計算機においては本方式がより効果的に働くことが予想される．例えば，家庭用ゲーム機を考えた場合，ユーザが同じ機種のゲーム機を2台所有することはなく，また，同時に複数のゲームが実行されることもない．本方式はゲームソフトごとに異なる不正コピー防止方式を採用することが可能であり，各ソフトハウスごとにゲームが開発されている現状にも適すると思われる．

4.4 CPU 秘密鍵の安全性

本方式をつきつめると，CPU に封印されている CPU 秘密鍵が本方式のセキュリティの根底となっていることが分かる．すなわち，万一，CPU に封印されている CPU 秘密鍵が漏洩してしまうと，本方式の効力は全て失われる．ハードウェア的に封印されている CPU 秘密鍵が不正に読み出される危険は少ないかもしれないが，近い将来，公開鍵暗号がブレイクされる可能性は否定できない．本方式においては複数の不正コピー防止方式を自由に選ぶことが可能であるが，今後は封印されている CPU 秘密鍵をも必要に応じて書き換えることができるように改良を施す必要があるだろう．

5. まとめ

CPU の一部をプログラマブルデバイスにより実装することにより，不正コピー防止のためのハードウェア機構をソフトウェア的に再構成することが可能な CPU を作成することを提案した．本方式により，各コンテンツプロバイダが独自に，コンテンツに応じた最新の不正コピー防止方式を容易に導入することが可能となる．今後は本方式をより詳細に渡って検討することにより，本方式の安全性を確認するとともに，本方式を著作者とユーザ間のさまざまな契約状態に対応可能なものに拡張していく予定である．

参考文献

[1] J.Zhao and E.Koch, “Embedding robust labels into images for copyright protection”, Proceedings of International Conference on Intellectual Property Right for Information, 1995-8 .

[2] R.M.Best , “Crypto Microprocessor for Executing Enciphered Programs”,US Patent, No.4,278,837, 1981-07 .

[3] R.M.Best , “Crypto Microprocessor for Executing Enciphered Programs”,US Patent, No.4,465,901, 1984-08 .

[4] 末松,今井, “CMP(Crypto Microprocessor)の一構成方法とその応用例”, 信学技報, ISEC98-8, 1998-05.

[5] D.L.Davis, US Patent, No.5,805,706, 1996-4.

[6] 西垣,曽我,井熊,田窪, “データのスクラッチングと動的復元による実行形式プログラムの不正コピー防止”, 電子情報通信学会論文誌(A)Vol.J83-A No.11 pp.1288-1299, 2000-11.

[7] 井熊,西垣,曽我,田窪, “IC カードから CPU への秘密情報の送信”, CSS’99 論文集, pp.49-54, 1999-10.

[8] NEC, “新しい計算機の設計概念を実現する「動的再構成 LSI」の開発について”, http://www.nec.co.jp/japanese/today/newsrel/9902/1502.html.