

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4884976号
(P4884976)

(45) 発行日 平成24年2月29日 (2012. 2. 29)

(24) 登録日 平成23年12月16日 (2011. 12. 16)

(51) Int. Cl. F I
H O 4 L 9/00 (2006. 01) H O 4 L 9/00 6 1 1 Z
G O 9 C 1/00 (2006. 01) G O 9 C 1/00 6 1 O A

請求項の数 9 (全 24 頁)

(21) 出願番号 特願2006-543679 (P2006-543679)
(86) (22) 出願日 平成16年11月30日 (2004. 11. 30)
(65) 公表番号 特表2007-514193 (P2007-514193A)
(43) 公表日 平成19年5月31日 (2007. 5. 31)
(86) 国際出願番号 PCT/IB2004/052607
(87) 国際公開番号 W02005/060147
(87) 国際公開日 平成17年6月30日 (2005. 6. 30)
審査請求日 平成19年11月29日 (2007. 11. 29)
(31) 優先権主張番号 03104643.6
(32) 優先日 平成15年12月11日 (2003. 12. 11)
(33) 優先権主張国 欧州特許庁 (EP)

前置審査

(73) 特許権者 598036964
イルデト・コーボレート・ビー・ヴィ
オランダ・N L-2 1 3 2・エルエス・ホ
ーフドルプ・タウルサヴェンウー・1 0 5
(74) 代理人 100108453
弁理士 村山 靖彦
(74) 代理人 100064908
弁理士 志賀 正武
(74) 代理人 100089037
弁理士 渡邊 隆
(74) 代理人 100110364
弁理士 実広 信哉

最終頁に続く

(54) 【発明の名称】 各暗号化ラウンドのコア暗号化関数を隠蔽するために置換を用いたブロック暗号化システム

(57) 【特許請求の範囲】

【請求項 1】

ディジタル信号処理関数 f を、不明瞭化された匿名の形態で、それぞれが一意的なインデクス j によって識別される複数の実行装置に供給する方法であって、前記関数 f は、ディジタル信号入力 x を処理してディジタル信号出力を出力するための、複数の信号処理関数 f_i ($1 \leq i \leq N$) を含む関数縦続を含み、前記方法は、

$2N$ 個の反転可能な置換 p_i ($1 \leq i \leq 2N$) のセットを選択するステップと、

N 個の関数 g_i のセットを算出するステップと、を含み、ここで関数 g_i は、 $1 \leq i \leq N$ について

【数 1】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等であり、前記方法は更に、

各装置 j について、前記装置及び／又は前記装置のユーザについて一意的な $2N$ 個の反転可能な置換 $p_{j,i}$ の対応するセット及び／又はシーケンスを選択するステップと、

各実行装置 j について、 $N-1$ 個の関数 $h_{j,i}$ の対応するセットを算出するステップと、を含み、ここで $h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 2】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等であり、前記方法は更に、

各前記実行装置 j に

【数 3】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含むそれぞれの実行装置関数縦続 $ED_j (y_1, \dots, y_N)$ を備えるステップと、
各前記実行装置 j にそれぞれのローダ関数

【数 4】

$$loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

10

を備えるステップと、

を含み、ここで $l_{j,i}$ は

【数 5】

$$p_{j,2i}^{-1} \circ p_{2i}$$

に関数的に同等であり、 $r_{j,i}$ は

【数 6】

$$p_{2i-1}^{-1} \circ p_{j,2i-1}$$

に関数的に同等であり、前記方法は更に、

20

前記実行装置に関数 g_1 乃至 g_N を供給するステップと、

前記実行装置において、 $ED_j(loader_j(g_1, \dots, g_N))$ を実行する
ステップと、

を含む方法。

【請求項 2】

各前記実行装置について同一のコンテンツを用いて、放送及び／又は記憶媒体上の配布
を通して各前記実行装置に前記 g_1 乃至 g_N を供給するステップを含む、請求項 1 に記載
のデジタル信号処理関数 f を供給する方法。

【請求項 3】

各前記実行装置について同一のコンテンツを用いて、放送及び／又は記憶媒体上の配布
を通して各前記実行装置に前記デジタル信号入力 x を供給するステップを更に含む、
請求項 2 に記載のデジタル信号処理関数 f を供給する方法。

30

【請求項 4】

装置特有のコンテンツを用いて、1 対 1 の通信チャネル及び／又は記憶媒体を通して、
対応する関数 $h_{j,i}$ 、 $l_{j,i}$ 又は $r_{j,i}$ のセットの少なくとも 1 つを、実行装置 j
に供給するステップを含む、請求項 1 に記載のデジタル信号処理関数 f を供給する方法
。

【請求項 5】

前記関数 f は、Feistel 暗号ネットワークに基づく復号化関数であり、前記信号処理関
数 f_i の各々が、それぞれの Feistel 復号化ラウンド関数である、請求項 1 に記載のデ
ジタル信号処理関数 f を供給する方法。

40

【請求項 6】

置換 p_i の各々は Feistel 変換器であり、連続的な対 $\langle x, y \rangle$ に演算する関数 Q は、
反転可能な関数 Q_x 及び Q_y が存在し且つ $Q(\langle x, y \rangle) = \langle Q_x(x), Q_y(y) \rangle$
である場合には、Feistel 変換器であり、ここで、

【数 7】

$$Q_x(x) \oplus Q_y(y) = Q_x(x \oplus y)$$

及び

【数 8】

$$Q_y(x) \oplus Q_y(y) = Q_y(x \oplus y)$$

である、請求項 5 に記載のデジタル信号処理関数 f を供給する方法。

【請求項 7】

実行装置 j 中のプロセッサに、デジタル信号処理関数 f を実行させるように動作可能なコンピュータプログラムであって、前記関数 f は、デジタル信号入力 x を処理してデジタル信号出力を出力するための、複数の信号処理関数 f_i ($1 \leq i \leq N$) を含む関数縦続を含み、本方法は、

前記実行装置について一意的であり

10

【数 9】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続をロードするステップを含み、ここで y_1 乃至 y_N は関数パラメータであり、前記方法は更に

ローダ関数

【数 10】

$$\text{loader}_j(x_1, \dots, x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,N}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

をロードするステップと、

20

関数 g_1 乃至 g_N のセットをロードするステップと、

前記ローダ関数を前記関数 g_1 乃至 g_N のセットに適用して関数 $g_{j,1}$ 乃至 $g_{j,N}$ のセットを出力し、前記実行装置関数縦続を前記関数 $g_{j,1}$ 乃至 $g_{j,N}$ のセットに適用するステップと、

を含み、ここで、

g_i は、 $1 \leq i \leq N$ について

【数 11】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等であり、

30

p_i は反転可能な置換 ($1 \leq i \leq N$) であり、

$h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 12】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等であり、

$l_{j,i}$ は、

【数 13】

$$p_{j,2i}^{-1} \circ p_{2i}$$

と関数的に同等であり、

40

$r_{j,i}$ は、

【数 14】

$$p_{2i-1}^{-1} \circ p_{j,2i-1}$$

と関数的に同等であり、

$p_{j,i}$ は、反転可能な置換 ($1 \leq i \leq 2N$) であり、前記装置及び／又は前記装置のユーザについて一意的である、コンピュータプログラム。

【請求項 8】

デジタル信号処理関数 f を、不明瞭化された匿名の形態で、複数の実行装置に供給す

50

るシステムであって、前記システムは、サーバと、それぞれが一意的なインデクス j によって識別される複数の実行装置とを含み、前記関数 f は、デジタル信号入力 x を処理してデジタル信号出力を出力するための、複数の信号処理関数 f_i ($1 \leq i \leq N$) を含む関数縦続を含み、

前記サーバは、プログラムの制御下で、

2 N 個の反転可能な置換 p_i ($1 \leq i \leq 2N$) のセットを選択し、

N 個の関数 g_i のセットを算出し、

各装置 j について、前記装置及び／又は前記装置のユーザについて一意な 2 N 個の反転可能な置換 $p_{j,i}$ の対応するセット及び／又はシーケンスを選択し、

各実行装置 j について、 $N-1$ 個の関数 $h_{j,i}$ の対応するセットを算出し、

各前記実行装置 j に

【数 15】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含むそれぞれの実行装置関数縦続 ED_j ($y_1, \dots, y_N$) を備え、

各前記実行装置 j にそれぞれのローダ関数

【数 16】

$$loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

を備え、

前記実行装置に関数 g_1 乃至 g_N を供給するためのプロセッサを含み、ここで関数 g_i は、 $1 \leq i \leq N$ について

【数 17】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等であり、 $h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 18】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等であり、 $l_{j,i}$ は

【数 19】

$$p_{j,2i}^{-1} \circ p_{2i}$$

に関数的に同等であり、 $r_{j,i}$ は

【数 20】

$$p_{2i-1}^{-1} \circ p_{j,2i-1}$$

に関数的に同等であり、

各前記実行装置 j は、

前記サーバから前記関数 g_1 乃至 g_N を取得する手段と、

プログラムの制御下で、

前記実行装置について一意的であり

【数 21】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続をロードし、

ローダ関数

【数 22】

$$loader_j(x_1, \dots, x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

10

20

30

40

50

をロードし、

前記ローダ関数を前記関数 g_1 乃至 g_N のセットに適用し、関数 $g_{j,1}$ 乃至 $g_{j,N}$ のセットを出力し、

前記実行装置関数縦続を、 $g_{j,1}$ 乃至 $g_{j,N}$ のセットに適用するためのプロセッサと、

を含み、ここで y_1 乃至 y_N は関数パラメータであるシステム。

【請求項 9】

請求項 8 に記載のシステムにおける使用のための実行装置であって、前記実行装置は、一意的なインデクス j によって識別され、

前記サーバから前記関数 g_1 乃至 g_N を取得する手段と、

プログラムの制御下で、

前記実行装置について一意的であり

【数 2 3】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続をロードし、

ローダ関数

【数 2 4】

$$loader_j(x_1, \dots, x_N) \equiv (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

をロードし、

前記ローダ関数を前記関数 g_1 乃至 g_N のセットに適用して関数 $g_{j,1}$ 乃至 $g_{j,N}$ のセットを出力し、

前記実行装置関数縦続を、前記関数 $g_{j,1}$ 乃至 $g_{j,N}$ のセットに適用するためのプロセッサと、

を含み、ここで y_1 乃至 y_N は関数パラメータであり、

g_i は、 $1 \leq i \leq N$ について

【数 2 5】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等であり、

p_i は反転可能な置換 ($1 \leq i \leq N$) であり、

$h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 2 6】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等であり、

$l_{j,i}$ は、

【数 2 7】

$$p_{j,2i}^{-1} \circ p_{2i}$$

と関数的に同等であり、

$r_{j,i}$ は、

【数 2 8】

$$p_{2i-1}^{-1} \circ p_{j,2i-1}$$

と関数的に同等であり、

$p_{j,i}$ は、反転可能な置換 ($1 \leq i \leq 2N$) であり、前記装置及び／又は前記装置のユーザについて一意的である、実行装置。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】

【0001】

本発明は、セキュアであり及び／又はパーソナライズされた方法で、縦続された信号処理関数を実行装置に供給する方法に関する。本発明はまた、セキュアであり及び／又はパーソナライズされた方法で、縦続された信号処理関数を実行装置に供給するシステムに関する。本発明は更に、セキュアであり及び／又はパーソナライズされた方法で供給された縦続接続された信号処理関数を実行する実行装置に関する。

【背景技術】

【0002】

インターネットは、デジタルコンテンツへの便利でユビキタス的なアクセスをユーザに提供する。強力な配布チャネルとしてのインターネットの可能性のため、多くの消費者向け電子機器が、インターネットへの最も有力な入口であるPCプラットフォームと相互運用するように努めている。著作権により保護されるコンテンツのための配布媒体としてのインターネットの使用は、コンテンツプロバイダの利益を確保するための抑え難い困難をもたらす。とりわけ、コンテンツプロバイダの著作権及びビジネスモデルを保証することが必要とされる。再生ソフトウェアの制御は、コンテンツが利用され得る期間及び条件を含む、コンテンツの所有者の利益を強化する一方法である。とりわけPCプラットフォームについては、ユーザは、コンテンツへのアクセスを提供するハードウェア及びソフトウェアへの完全な制御と、いずれのコンテンツ保護メカニズムをも攻撃及び回避するための無制限の時間及びリソースを持つことが仮定される必要がある。結果として、コンテンツプロバイダは、全てのユーザが信頼されるわけではないコミュニティへの敵対的なネットワークによって、正当なユーザにコンテンツを配信しなければならない。PCへ配布される保護されたコンテンツのためのデジタル権利管理における一般的なアプローチは、デジタルコンテンツを暗号化し（例えばDESを利用して）、PCのハードディスク上の所謂ライセンスデータベースに復号化鍵（即ち「ライセンス」）を保存することである。PC上のデジタルコンテンツは一般に、Microsoft社のMedia Player、Real社のRealOne Player、Apple社のQuickTime（登録商標）Playerのようなメディアプレイヤーを利用して再生される。斯かるプレイヤーは、特定のコンテンツの形式に対して、当該形式に専用のデコードを実行するための、それぞれのプラグインをロードすることができる。これらのコンテンツの形式は、AVI、DV、Motion JPEG、MPEG-1、MPEG-2、MPEG-4、WMV、Audio CD、MP3、WMA、WAV、AIFF/AIFC、AU等を含み得る。プレイヤー及びプラグインの構造は、図1に示される。図1においては、メディアプレイヤー100が、コアプレイヤー100及び幾つかの形式に専用のプラグイン（図示されるものはプラグイン120、122及び124）を含む。コア（core）プレイヤー100は例えば、プレイヤーを制御するためのユーザインタフェースを提供する。各プラグインはそれぞれのデコーダを含む。各プラグインは、デコードされたコンテンツを、サウンドカードのような再生ハードウェア／ソフトウェアに直接送信しても良いし、又は更なる処理のためにコアプレイヤー100に送っても良い。セキュアな再生のため、当該形式のコンテンツをデコードするだけでなく、該コンテンツを復号化するセキュアなプラグインが利用される。このことは図2に示される。図2においては、暗号化されたコンテンツが最初に復号化器230に送られ、次いで復号化されたコンテンツが当該形式に専用のデコーダ220に送られる。復号化器230は、ライセンスデータベース210から復号化鍵／ライセンスを受信しても良い。

【0003】

暗号化に依存するデジタル権利管理の最大の脆弱性は、鍵の配布及び処理である。再生のために、ソフトウェアプレイヤーは、ライセンスデータベースから復号化鍵を取得する必要がある。次いで当該復号化鍵を、暗号化されたコンテンツの復号化のためにメモリのどこかに保存する必要がある。このことは、ソフトウェアプレイヤーにおける鍵の処理の攻撃のために、攻撃者に2つの選択肢を残す。第1に、ライセンスデータベースのアクセス関数のリバースエンジニアリングが、全てのライセンスデータベースからアセット鍵（as

10

20

30

40

50

set key) を取得することが可能なブラックボックスソフトウェアに帰着する（即ち、攻撃者がソフトウェア関数の内部の動作を理解する必要がない）ことである。第2に、コンテンツの復号化の間に利用されるメモリへのアクセスの監視によって、アセット鍵を取得することが可能であることである。

一般に、デジタル権利管理システムは、ラウンド（round）と呼ばれる暗号化／復号化ステップのシーケンスを利用して、ブロック毎にデータストリームを処理するブロック暗号に基づく暗号化手法を利用する。i-1 番目のラウンドの出力が、i 番目のラウンドの入力である。従って、Nラウンドのシステムについては、アルゴリズムは関数の縦続（cascade）

【数 5 8】

$$f_N \circ \dots \circ f_1(x)$$

10

として記述されることが出来る。ここで、関数 f_i は、ラウンド i の関数を表す。殆どのブロックアルゴリズムは、Feistelネットワークである。斯かるネットワークにおいては、偶数長 n の入力データブロック x は、長さ $n/2$ の 2 つの半分（通常 L 及び R と呼ばれる）に分割される。1 番目のラウンドに供給される入力 x は $x = \langle L_0, R_0 \rangle$ と与えられる。i 番目のラウンド（ $i > 0$ ）は、関数 f_i を実行する。ここで f_i は、

【数 5 9】

$$f_i(\langle L_{i-1}, R_{i-1} \rangle) = \langle R_{i-1}, (L_{i-1} \oplus F(R_{i-1}, K_i)) \rangle$$

20

と定義される。 K_i は i 番目のラウンドにおいて利用される副鍵（subkey）であり、F は任意のラウンド関数（round function）である。

【発明の開示】

【発明が解決しようとする課題】

【0004】

本発明の目的は、Feistelネットワークのような縦続された信号処理関数のより優れた保護を提供することにある。

【課題を解決するための手段】

【0005】

本目的を達成するため、デジタル信号処理関数 f を不明瞭化された形態で実行装置に供給する方法であって、前記関数 f は、デジタル信号入力 x を処理してデジタル信号出力を出力するための、複数の信号処理関数 f_i （ $1 \leq i \leq N$ ）を含む関数縦続（例えば

30

【数 6 0】

$$FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$$

）を含む前記方法は、

2N 個の反転可能な置換 p_i （ $1 \leq i \leq 2N$ ）のセットを選択するステップと、

N 個の関数 g_i のセットを算出するステップと、を含み、ここで関数 g_i は、 $1 \leq i \leq N$ について

【数 6 1】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

40

と関数的に同等であり、前記方法は更に、

N-1 個の関数 h_i のセットを算出するステップを含み、ここで関数 h_i は、 $2 \leq i \leq N$ について

【数 6 2】

$$p_{2i-1}^{-1} \circ p_{2i-2}$$

と関数的に同等であり、前記方法は更に、

50

【数 6 3】

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続を前記実行装置に備えるステップを含み、ここで y_1 乃至 y_N は関数パラメータであり（例えば

【数 6 4】

$$ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

）、前記方法は更に、

関数 g_1 乃至 g_N を前記実行装置に供給するステップと、

前記実行装置において、前記実行装置関数縦続を関数 g_1 乃至 g_N に適用する（例えば $ED_1(g_1, \dots, g_N)$ ）ステップと、
を含む。

本発明によれば、構成 (constituent) 関数 f_i は、 g_i としてカプセル化された形態で提供される。ここで g_i は、 $1 \leq i \leq N$ について、

【数 6 5】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

に関数的に同等である。カプセル化のために利用される関数 p_i もまた、 $2 \leq i \leq N$ について、

【数 6 6】

$$p_{2i-1}^{-1} \circ p_{2i-2}$$

の乗算されたバージョンである h_i の形態で供給されることにより隠蔽される。インターリーブされた態様で実行装置において関数 g_i 及び h_i を実行する（例えば図 4 に示されるように）ことにより、 f_i を直接に認識可能とすることなく、関数の縦続の機能が達成される。とりわけ、 f_i が Feistel 暗号のラウンド関数を表す場合には、該ラウンド関数に埋め込まれたラウンド鍵は直接には認識されない。 f_i の不明瞭にされた配信がセキュリティを増大させる。実行関数装置縦続が、メディアプレイヤのコア機能を形成し得る。ここで、セット $g_1, \dots, g_N$ が、該プレイヤが f_1 から f_N まで（ f_N を含む）を含む関数縦続を実行することを可能とする。

従属請求項 2 及び 3 は、関数縦続の（関数的な）開始を保護する 2 つのそれぞれの代替例を示す。請求項 2 の例においては、実行装置関数縦続は p_1^{-1} から開始し、例えば

【数 6 7】

$$ED_2(y_1, \dots, y_N) = y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$$

である。これを $g_1, \dots, g_N$ に適用することにより、前記装置において実行される関数シーケンスの関数的な開始として

【数 6 8】

$$\dots \circ g_2 \circ h_2 \circ g_1 \circ p_1^{-1} = \dots \circ p_3^{-1} \circ f_2 \circ p_2 \circ p_2^{-1} \circ f_1 \circ p_1 \circ p_1^{-1} = \dots \circ p_3^{-1} \circ f_2 \circ f_1$$

となり、このようにして前記実行装置は f_1 を明示的に実行する。請求項 3 の例においては、 p_1^{-1} を隠蔽することを支援する開始関数 f_0 を用いて関数縦続を拡張することにより、セキュリティが増大させられる。該関数縦続は例えば、

【数 6 9】

$$FC_2(x) = f_N \circ \dots \circ f_1 \circ f_0(x)$$

10

20

30

40

50

であっても良い。前記実行装置関数縦続は関数 S_1 から開始し、例えば

【数 7 0】

$$ED_3(y_1, \dots, y_N) = y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$$

である。ここで S_1 は

【数 7 1】

$$p_1^{-1} \circ f_0$$

に関数的に同等である。 S_1 は単に f_0 で乗算された形態で p_1^{-1} を表すため、 p_1^{-1} は特定のメモリ位置を読み取ることのような直接的な方法で前記実行装置から取得されることはできない。好ましくは、 f_0 はグローバルに秘密 (global secret) である。

10

【0 0 0 6】

従属請求項 4 及び 5 は、請求項 2 及び 3 と類似した方法で関数縦続の (関数的な) 末尾を保護する 2 つのそれぞれの代替例を示す。

【0 0 0 7】

従属請求項 6 の方法によれば、選択された置換 p_i のシーケンスが装置に対して一意的である。このようにして、関数縦続が、不明瞭にされた形態でだけではなく、パーソナライズされた形態で、実行装置に供給される。例えば、埋め込まれた復号鍵を持つ Feistel 暗号を関数縦続が表す場合、暗号解読又は力任せの攻撃は、 g_1 、 $\dots$ 、 g_N のブラックボックス機能を得ることに帰着し得る。この破壊された機能はこのとき、対応する実行装置関数縦続と組み合わせてのみ機能し、他のいずれの実行装置とも機能しない。このことは、成功する攻撃の影響をかなり制限する。

20

【0 0 0 8】

従属請求項 7 の方法によれば、実行装置関数縦続は、例えばメディアプレイヤー又はメディアプレイヤー用のプラグインの形態で、プログラムに埋め込まれる。該実行装置はかくして、セキュアでパーソナライズされたソフトウェアを備える。

【0 0 0 9】

従属請求項 8 の方法によれば、関数 g_1 、 $\dots$ 、 g_N は、前記プログラム用のプラグインを形成する。前記プログラム自体がプラグインである場合には、関数 g_1 、 $\dots$ 、 g_N は事実上該プラグイン用のプラグインである。代替として、従属請求項 9 の方法によれば、関数 g_1 、 $\dots$ 、 g_N は前記実行装置関数縦続と同じプログラムに埋め込まれても良い。

30

本発明の目的を達成するため、実行装置中のプロセッサに、デジタル信号処理関数 f を実行させるように動作可能なコンピュータプログラムであって、前記関数 f は、デジタル信号入力 x を処理してデジタル信号出力を出力するための、複数の信号処理関数 f_i ($1 \leq i \leq N$) を含む関数縦続 (例えば

【数 7 2】

$$FC_1(x) = f_N \circ \dots \circ f_1(x)$$

) を含み、前記関数 f の実行は、

40

【数 7 3】

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続をロードし、

関数 g_1 乃至 g_N のセットをロードし、

前記関数 g_1 乃至 g_N のセットに前記実行装置関数縦続を適用することにより実行され、ここで y_1 乃至 y_N は関数パラメータであり、

g_i は、 $1 \leq i \leq N$ について

【数 7 4】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同値であり、

h_i は、 $2 \leq i \leq N$ について

【数 7 5】

$$p_{2i-1}^{-1} \circ p_{2i-2}$$

と関数的に同値であり、

p_i は、 $1 \leq i \leq 2N$ について反転可能な置換であるコンピュータプログラムが提供される。

本発明の目的を達成するため、ディジタル信号処理関数 f を、不明瞭化された匿名の形態で、それぞれが一意的なインデクス j によって識別される複数の実行装置に供給する方法であって、前記関数 f は、ディジタル信号入力 x を処理してディジタル信号出力を出力するための、複数の信号処理関数 f_i ($1 \leq i \leq N$) を含む関数縦続（例えば

【数 7 6】

$$FC_1(x) \equiv f_N \circ \dots \circ f_1(x)$$

) を含む前記方法は、

$2N$ 個の反転可能な置換 p_i ($1 \leq i \leq 2N$) のセットを選択するステップと、

N 個の関数 g_i のセットを算出するステップと、を含み、ここで関数 g_i は、 $1 \leq i \leq N$ について

【数 7 7】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等であり、前記方法は更に、

各装置 j について、前記装置及び／又は前記装置のユーザについて一意的な $2N$ 個の反転可能な置換 $p_{j,i}$ の対応するセット及び／又はシーケンスを選択するステップと、

各実行装置 j について、 $N-1$ 個の関数 $h_{j,i}$ の対応するセットを算出するステップと、を含み、ここで $h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 7 8】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等であり、前記方法は更に、

各前記実行装置 j に

【数 7 9】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

を含むそれぞれの実行装置関数縦続 $ED_j(y_1, \dots, y_N)$ を備えるステップと、

各前記実行装置 j にそれぞれのロード関数

【数 8 0】

$$loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

を備えるステップと、

を含み、ここで $l_{j,i}$ は

【数 8 1】

$$p_{j,2i}^{-1} \circ p_{j,2i-1}$$

に関数的に同等であり、 $r_{j,i}$ は

【数 8 2】

$$p_{2,i}^{-1} \circ p_{j,2,i}$$

に関数的に同等であり、前記方法は更に、

前記実行装置に関数 g_1 乃至 g_N を供給するステップと、

前記実行装置において、 $ED_j(\text{loader}_j(g_1, \dots, g_N))$ を実行するステップと、
を含む。

【0010】

関数 f_i は、請求項 1 に記載のものと同様に、関数 g_1 、 $\dots$ 、 g_N の形態で不明瞭にされる。関数 g_1 、 $\dots$ 、 g_N は各装置に対して同一であり、1つのデフォルト／匿名の装置に対応するように見える。実行装置は装置に特有の（「パーソナライズ」された）実行装置縦続を備える。装置に特有のローダ（loader）関数が、それぞれの匿名の関数 g_i を、実行装置縦続に送られる対応する装置に特有の関数へと変換するために利用される。前記ローダ関数は、公然にされていない置換 $p_{j,i}$ のセット／シーケンスに基づく変換関数 $l_{j,i}$ 及び $r_{j,i}$ を利用する。

【0011】

従属請求項 12 の方法によれば、関数 g_i は、例えば放送又は CD-ROM 若しくは DVD のような記憶媒体を利用して、全ての装置に同一の方法で供給されることができる。

【0012】

本発明のこれらの及び他の態様は、以下に説明される実施例を参照しながら説明され明らかとなるであろう。

【発明を実施するための最良の形態】

【0013】

図 3 は、本発明が利用され得る先行技術のシステムのブロック図を示す。図 3 の例においては、コンテンツ（典型的にはオーディオ及び／又はビデオコンテンツ）が媒体 310 に配布される。前記媒体は、各プレイヤについて同一であっても良い。前記媒体は、オーディオ CD、DVD、固体メモリ等のような、いずれの適切なタイプのものであっても良い。前記媒体上のコンテンツは、好ましくは Feistel 暗号のような暗号化アルゴリズムを利用のもとに暗号化されることにより、コピー保護されている。前記記憶媒体は、復号化鍵に関する情報を含んでも良い。代替として前記記憶媒体は、プレイヤが、例えばインターネットにおけるサーバから該情報をダウンロードすることによって、前記情報を取得することを可能とする情報 312（識別子のような）を含んでも良い。前記復号化鍵は、復号化鍵 326 を算出する（324）ため、鍵特有の鍵 322 及び情報 312 を利用することによりセキュアモジュール 320 において生成される。前記復号化鍵は、第 2 のモジュール 330 において取得される（332）。第 2 のモジュール 330 は、媒体 310 のコンテンツ 314 を復号化し（334）、デコードし（336）、再生する（338）。

図 4 は、本発明による方法を示す。デジタル信号処理関数 f は、不明瞭化された形態で実行装置に供給される。関数 f は、複数の信号処理関数 f_i （ $1 \leq i \leq N$ ）を含む関数縦続を含む。例えば、前記関数縦続のコアは、

【数 8 3】

$$FC_1(x) = f_N \circ \dots \circ f_1(x)$$

によって形成されても良い。ここで慣例的な数学的表記：

【数 8 4】

$$g \circ f(x) = g(f(x))$$

10

20

30

40

50

が利用されていることに留意されたい。原則的に、前記関数縦続はいずれのデジタル信号処理関数であっても良い。好適な実施例においては、前記関数縦続は暗号を含む。例えば、関数 f_i はFeistel暗号の i 番目のラウンド ($i > 0$) を表しても良い。斯かる場合においては、 f_i は、

【数 8 5】

$$f_i(L_{i-1} \oplus R_{i-1}) = R_{i-1} \oplus (L_{i-1} \oplus F(R_{i-1}, K_i))$$

と定義される。ここで K_i は i 番目のラウンドにおいて利用される副鍵であり、 F は任意のラウンド関数である。

本発明によれば、 $2N$ 個の反転可能な置換 p_i ($1 \leq i \leq 2N$) が選択される。次いで、 N 個の関数 g_i のセットが算出される。ここで g_i は、 $1 \leq i \leq N$ について

【数 8 6】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

に関数的に同等である。ここで関数的に同等であるとは、全てのとり得る入力値について、 g_i が同じ入力（例えば x ）に適用された場合に、

【数 8 7】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

が当該入力に適用された場合と等しい結果が得られることを意味する。合成関数 $p_{2i}^{-1} \circ f_i$ 及び p_{2i}^{-1} は、個別には認識できない。 g_i は、

【数 8 8】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

のブラックボックス機能を提供する。図 5 は、当該アプローチを非常に単純な 1 次元関数について説明する。本例においては、

【数 8 9】

$$p_4(x) = \sqrt{x}; p_4^{-1}(x) = x^2; p_3(x) = \frac{x}{3}; p_3^{-1}(x) = 3x; f_2(x) = x + 3$$

である。従って、

【数 9 0】

$$g_2(x) = p_4^{-1} \circ f_2 \circ p_3(x) = p_4^{-1} \circ f_2(p_3(x)) = p_4^{-1} \circ f_2\left(\frac{x}{3}\right) = p_4^{-1}\left(\frac{x}{3} + 3\right) = \left(\frac{x}{3} + 3\right)^2$$

である。コンピュータのコンパイラ構築の分野においては、

【数 9 1】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

のブラックボックス機能が所謂部分評価 (partial evaluation) を用いてどの様に達成されるかが良く知られている。N. D. Jones、C. K. Gomard 及び P. Sestoft による「Partial Evaluation and Automatic Program Generation」の第 1 章は、部分評価の概念を記載している。このことは、より詳細にはここでは説明されない。有用な置換を実行することを可能とするため、デジタル信号入力 x は、例えば 64 又は 128 ビットブロック／ベクトルの、多次元パラメータであることは理解されるであろう。本発明によれば、 $N-1$ 個の関数 h_i のセットが算出される。ここで h_i は、 $2 \leq i \leq N$ について

10

20

30

40

【数 9 2】

$$p_{2i-1}^{-1} \circ p_{2i-2}$$

と関数的に同等である。図 5 の単純な例を利用すると、

【数 9 3】

$$h_2(x) = p_3^{-1} \circ p_2(x) = 3 \bullet p_2(x); h_3(x) = p_5^{-1} \circ p_4(x) = p_5^{-1}(\sqrt{x})$$

である。これらの定義を利用すると、 f_2 を隠蔽する前記実行装置の縦続の部分は、

【数 9 4】

$$\begin{aligned} \cdots h_3 \circ g_2 \circ h_2 &= \cdots (p_5^{-1}(\sqrt{x})) \circ (\frac{x}{3} + 3)^2 \circ (3 \bullet p_2(x)) = (p_5^{-1}(\sqrt{x})) \circ (\frac{3 \bullet p_2(x)}{3} + 3)^2 \\ &= (p_5^{-1}(\sqrt{x})) \circ (p_2(x) + 3)^2 = p_5^{-1}(\sqrt{(p_2(x) + 3)^2}) = p_5^{-1}(p_2(x) + 3) \end{aligned}$$

となる。これは実際に

【数 9 5】

$$p_5^{-1} \circ f_2 \circ p_2(x)$$

と関数的に同等であることが分かる。かくして、当該縦続を実行した実行装置は、 f_2 の明示的な知識を持つことなく f_2 を実行したことになる。

【0 0 1 4】

別の例においては、 $N = 2$ であり、 f_1 及び f_2 がそれぞれ

f_1 : {0 → 3、1 → 1、2 → 6、3 → 2、4 → 7、5 → 5、6 → 4、7 → 0、8 → 8}、

f_2 : {0 → 4、1 → 1、2 → 5、3 → 7、4 → 6、5 → 2、6 → 0、7 → 8、8 → 3}

によって与えられるマッピングテーブルに対して評価される。本例においては、 f_1 は 0 と 8 との間の数を 0 と 8 との間の数に変換する反転可能な関数である。例えば値 0 は値 3 に、値 1 は値 1 に、値 2 は値 6 に等変換される。以下の 4 つのそれぞれの置換が、本例において利用される：

p_1 : {0 → 5、1 → 3、2 → 1、3 → 7、4 → 0、5 → 6、6 → 2、7 → 8、8 → 4}、

p_2 : {0 → 8、1 → 6、2 → 7、3 → 3、4 → 4、5 → 2、6 → 0、7 → 1、8 → 5}、

p_3 : {0 → 3、1 → 5、2 → 7、3 → 1、4 → 6、5 → 0、6 → 2、7 → 8、8 → 4}、

p_4 : {0 → 3、1 → 0、2 → 5、3 → 2、4 → 7、5 → 8、6 → 1、7 → 4、8 → 6}

本例に対しては、以下の 3 つの逆置換が利用される：

p_2^{-1} : {0 → 6、1 → 7、2 → 5、3 → 3、4 → 4、5 → 8、6 → 1、7 → 2、8 → 0}、

p_3^{-1} : {0 → 5、1 → 3、2 → 6、3 → 0、4 → 8、5 → 1、6 → 4、7 → 2、8 → 7}、

p_4^{-1} : {0 → 1、1 → 6、2 → 3、3 → 0、4 → 7、5 → 2、6 → 8、7 → 4、8 → 5}

これら 3 の関数が与えられると、

10

20

30

40

【数 9 6】

$$h_2(x) = p_3^{-1} \circ p_2(x)$$

は以下のように与えられる：

h_2 : { 0 → 7、1 → 4、2 → 2、3 → 0、4 → 8、5 → 6、6 → 5、7 → 3、8 → 1 }

例えば、 p_2 は 0 を 8 へマッピングし、 p_3^{-1} は 8 を 7 にマッピングする。従って、

【数 9 7】

$$h_2(0) = p_3^{-1} \circ p_2(0) = 7$$

10

である。同様に、

【数 9 8】

$$g_1(x) = p_2^{-1} \circ f_1 \circ p_1(x)$$

は、

g_1 : { 0 → 8、1 → 5、2 → 7、3 → 6、4 → 3、5 → 4、6 → 1、7 → 0、8 → 2 }

によって与えられ、

【数 9 9】

20

$$g_2(x) = p_4^{-1} \circ f_2 \circ p_3(x)$$

は、

g_2 : { 0 → 4、1 → 3、2 → 5、3 → 6、4 → 1、5 → 7、6 → 2、7 → 0、8 → 8 }

によって与えられる。

前記実行装置は、

【数 1 0 0】

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

30

を含む実行装置関数縦続を備える。ここで、 y_1 、 $\dots$ 、 y_N は関数パラメータである。このことは図 4 において、関数 h_N 、 h_{N-1} 、 $\dots$ 、 h_2 のシーケンス 4 1 0 として示される。実行装置関数縦続の例は、

【数 1 0 1】

$$ED_1(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

である。更に、関数 g_1 、 $\dots$ 、 g_N が前記実行装置に供給される。このことは図 4 において、関数 g_N 、 g_{N-1} 、 $\dots$ 、 g_1 のシーケンス 4 2 0 として示される。前記実行装置において、前記実行装置関数縦続が関数 g_1 、 $\dots$ 、 g_N に適用される。このことは例えば、前記実行装置における全体の信号処理関数 $ED_1(g_1, \dots, g_N)$ を与える。該関数は次いでデジタル信号入力 x に適用される。

40

【数 1 0 2】

$$h_{i+1} \circ g_i \circ h_i$$

のような連鎖の中央部分を見ると、該部分は

【数 1 0 3】

$$h_{i+1} \circ g_i \circ h_i = p_{2i+1}^{-1} \circ p_{2i} \circ p_{2i}^{-1} \circ f_i \circ p_{2i-1} \circ p_{2i-1}^{-1} \circ p_{2i-2} = p_{2i+1}^{-1} \circ f_i \circ p_{2i-2}$$

を与える。該式の最初の項及び最後の項は、それぞれの g 項によって除去される。全体の結果として、前記実行装置は、いずれの関数 f_i へのアクセスをも持つことなく、関数縦続

【数 1 0 4】

$$f_N \circ \dots \circ f_1(x)$$

10

を含む関数を実行する。これらの関数はかくして不明瞭化される。

好適な実施例においては、前記連鎖の先頭及び末尾の処理について選択肢が与えられる。いずれの更なる手段をも用いずに、前記実行装置における結果の全体の信号処理関数が、

【数 1 0 5】

$$ED_1(g_1, \dots, g_N) \equiv p_{2N-1}^{-1} \circ f_N \circ \dots \circ f_1(x) \circ p_1$$

となり得る。例えば、項 p_1 は

20

【数 1 0 6】

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$$

を含む実行装置関数縦続を利用することにより除去され得る。例えば、

【数 1 0 7】

$$ED_2(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ p_1^{-1}$$

である。好ましくは、項 p_1^{-1} は前記実行装置においてセキュアに維持される。このことを実行する好適な方法は、更なる信号処理関数 f_0 を用いて関数縦続を拡張することである（例えば

30

【数 1 0 8】

$$FC_2(x) \equiv f_N \circ \dots \circ f_1 \circ f_0(x)$$

）。このとき前記実行装置関数縦続は

【数 1 0 9】

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$$

を含む（例えば

【数 1 1 0】

$$ED_3(y_1, \dots, y_N) \equiv y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1 \circ S_1$$

40

）。ここで S_1 は、

【数 1 1 1】

$$p_1^{-1} \circ f_0$$

に関数的に同等である。このようにして、個々の項 p_1^{-1} 及び f_0 は明らかにされる必要がなく、乗算された形式

【数 1 1 2】

$$p_1^{-1} \circ f_0$$

のみが存在する。好ましくは、 f_0 はグローバルに秘密である。即ち、 f_0 を知っている必要があるものには知られているが、それ以上は配布されない。グローバルな秘密はそれ自体知られており、グローバルな秘密をセキュアな方法で通信する方法もまた知られているが、ここではこれ以上議論されない。

対応する方法で、項 p_{2N-1}^{-1} を処理するための方法がとられても良い。例えば、前記実行装置関数縦続は、

10

【数 1 1 3】

$$p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

を含んでも良い（例えば

【数 1 1 4】

$$ED_4(y_1, \dots, y_N) \equiv p_{2N} \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

）。 p_{2N} のより優れた保護のため、前記関数縦続は更なる信号処理関数 f_{N+1} で終了しても良い（例えば

20

【数 1 1 5】

$$FC_3(x) \equiv f_{N+1} \circ f_N \circ \dots \circ f_1(x)$$

）。このとき前記実行装置関数縦続は

【数 1 1 6】

$$S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

を含む（例えば

【数 1 1 7】

$$ED_5(y_1, \dots, y_N) \equiv S_2 \circ y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

）。ここで S_2 は、

【数 1 1 8】

$$f_{N+1} \circ p_{2N}$$

と関数的に同等である。

【0 0 1 5】

図 6 は、本発明が利用され得るシステムを示す。システム 600 は、サーバ 610 と、少なくとも 1 つの実行装置 620 とを有する。前記サーバは、例えばウェブサーバのようなサーバとして利用されるプラットフォーム又はファイルサーバのような、従来のコンピュータプラットフォーム上に実装されても良い。前記サーバは、プロセッサ 612 を含む。プロセッサ 612 は、プログラムの制御下で動作する。前記プログラムは、埋め込まれた ROM のように埋め込まれた記憶装置において前記プロセッサに常設的に埋め込まれても良いが、ハードディスク（図示されていない）のようなバックグラウンドの記憶装置からロードされても良い。前記プログラムの制御の下、プロセッサ 612 は以下を実行する：

40

- ・ $2N$ 個の反転可能な置換 p_i ($1 \leq i \leq 2N$) のセットを選択する。
- ・ N 個の関数 g_i のセットを算出する。ここで g_i は、 $1 \leq i \leq N$ について

【数 1 1 9】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等である。

・

$N-1$ 個の関数 h_i のセットを算出する。ここで h_i は、 $2 \leq i \leq N$ について

【数 1 2 0】

$$p_{2i-1}^{-1} \circ p_{2i-2}$$

10

と関数的に同等である。

前記置換は、（好ましくはセキュアな）記憶装置（図示されていない）に保存されていても良い、置換の非常に大きなセットから（例えばランダム的に又は疑似ランダム的に）選択されても良い。前記サーバは、前記置換を生成するために適切なプログラムも利用しても良い。反転可能な置換を生成する方法は良く知られており、ここではこれ以上説明されない。

加えて、前記サーバは、

【数 1 2 1】

20

$$y_N \circ h_N \circ y_{N-1} \circ h_{N-1} \circ \dots \circ y_1$$

を含む実行装置関数縦続を前記実行装置に供給するための手段 6 1 4 を含む。ここで y_1 、 $\dots$ 、 y_N は関数パラメータである。これらのサーバは、このことをいずれの適切な形態で実行しても良い。例えば工場では、項 h_i は、実行装置 6 2 0 の製造の間、前記実行装置の記憶モジュールに保存されても良い。図 6 は、前記項が、インターネット 6 3 0 によって直接に実行装置 6 2 0 にダウンロードされることを示す。サーバ 6 1 0 はまた、関数 g_1 、 $\dots$ 、 g_N を実行装置 6 2 0 に供給するための手段 6 1 6 を含む。関数 g_i は、それぞれ関数 f_i を組み込む。関数 f_i は、デジタル信号入力 x 用に特に選択される。例えば、各ビデオタイトルが、対応する暗号化関数を用いて暗号化される（例えば同一の暗号化だがコンテンツ特有の鍵を利用して）。この目的のため、サーバ 6 1 0 はまた、コンテンツ 6 4 0 を暗号化するようにプロセッサ 6 1 2 を制御するためのソフトウェアを含んでも良く、また、例えば記憶媒体による又はインターネットのような通信媒体を通じた配布のために、配布媒体に暗号化されたコンテンツ 6 4 2 を供給しても良い。

30

【0 0 1 6】

実行装置 6 2 0 は、サーバ 6 1 0 から関数 g_1 、 $\dots$ 、 g_N を得るための手段 6 2 6 を含む。これらの手段は前記サーバの手段 6 1 6 と協働するものであり、更には説明されない。実行装置 6 2 0 は更に、プロセッサ 6 2 2 を含む。前記プロセッサは、パーソナルコンピュータから知られたプロセッサ又は埋め込まれたマイクロコントローラのような、いずれの適切なタイプのものであっても良い。プロセッサ 6 2 2 は、プログラムの制御下で動作する。前記プログラムは、埋め込まれた ROM のように埋め込まれた記憶装置を利用してプロセッサ 6 2 2 に常設的に埋め込まれても良いが、ハードディスク（図示されていない）のようなバックグラウンドの記憶装置からロードされても良い。前記プログラムの制御の下、プロセッサ 6 2 2 は、実行装置関数縦続をロードし、例えば ED_1 (g_1 、 $\dots$ 、 g_N) を実行することにより、該ロードされた実行装置関数縦続を関数 g_1 、 $\dots$ 、 g_N に適用する。結果の信号処理関数は次いで、信号入力 x （例えば媒体から受信されたコンテンツ）に適用されても良い。プロセッサ 6 2 2 は、いずれの適切な形態で前記実行装置関数縦続をロードしても良い。例えば前記縦続は、製造の間記憶装置に予め保存され、直接のメモリ読み取りアクセスへのロードを減少させても良い。図 6 の例においては、実行装置 6 2 0 は、例えばインターネット 6 3 0 を通して、又は媒体 6 5 0 から、

40

50

前記縦続（又は前記縦続の項）を取得するための手段624を含む。同様に、実行装置620は、媒体650から暗号化されたコンテンツ652を取得し、プロセッサ622を利用して該コンテンツを復号化しても良い。前記プロセッサは、前記復号化されたコンテンツをデコードしても良い。

【0017】

図7は、実行装置関数縦続が、プロセッサ622による実行のためのソフトウェアプログラム710に埋め込まれて実行装置620に供給される、好適な実施例を示す。図7における同一の番号は、図6において利用されたものと同一のアイテムを示す。ソフトウェアプログラム710は、メディアプレイヤーのようなプログラム用のプラグインであっても良い。従って図7の手段614は、該プラグイン710を、インターネットを介して供給しても良く（例えば図7のアイテム630）、又は製造の間に実行装置620に直接埋め込んでも良い。

【0018】

実施例においては、関数 g_1 、 $\dots$ 、 g_N が、プログラム710用のプラグインの形で実行装置620に供給される。プログラム710が既にプラグインである場合には、関数 g_1 、 $\dots$ 、 g_N は事実上プラグイン用のプラグインである。代替としては、関数 g_1 、 $\dots$ 、 g_N は、前記実行装置関数縦続を関数パラメータ g_1 、 $\dots$ 、 g_N に適用することにより、関数 g_1 、 $\dots$ 、 g_N をソフトウェアプログラム710に埋め込むことにより、実行装置620に供給される。このようにして、プログラム710は、関数 h_i と g_i との両方を埋め込む。

【0019】

実施例においては、各実行装置及び／又は該実行装置のユーザは一意的であり、一意的な識別子（例えば一意的な番号 j ）によって識別される。本発明によるシステム及び方法においては、シーケンス g_i 及び h_i は、包含されるユーザ／団体（party）に対して一意的であることが保証される。このことは、実行装置及び／又は実行装置のユーザの一意的な識別子 j を得ることによって、及びそれぞれ得られた識別子に対して一意的な $2N$ 個の反転可能な置換 p_i のセットによって、実現されることができる。同様に、置換の同一のセットを利用して、置換の一意的なシーケンスが選択され得る。両方の手法が（置換の異なるセット又は置換の異なるシーケンスを選択する）組み合わせられても良い。好ましくは、各一意的な識別子について一意的なセット／シーケンスを、サーバが（セキュアな方法で）保存する。このようにして、パーソナルコンピュータ中の各ソフトウェアメディアプレイヤーが、メディアタイトルを復号化及び／又はデコードするための一意的なプラグインを供給されることができる。媒体自体は一意的である必要はない。暗号化されたコンテンツは、単に暗号化関数に依存し、置換の一意的なセット／シーケンスには依存しない。前記ソフトウェアが前記識別子に対応するか否かを定期的に（例えばメディアプレイヤーの起動時に）チェックし、一致が確認された場合にのみ前記ソフトウェアを実行することにより、プレイヤーソフトウェアが、該プレイヤーソフトウェアが属さないPC上で実行されないことが保証される。偶然にハッカーが装置特有の置換を得たとしても、該置換は包含されるPCにおいてのみ利用されることができ、異なる暗号化（異なる関数 f_i に帰着する）を用いて保護されたコンテンツには利用可能であるが、異なるプラットフォームにおいては利用されることができない。

以上では、信号処理関数の縦続が不明瞭な態様で実行装置に供給される方法及びシステムが説明された。各装置について、置換の同一のセット／シーケンスが利用されても良いし、又は装置特有のセット／シーケンスが利用されても良い。以下では、各装置について同一である不明瞭化された態様で信号関数縦続（「鍵」）を配布し、共通鍵（common key）を装置特有の鍵に変換する変換ルーチン（「ローダ」）を利用することにより、装置特有のセット／シーケンスを実現するための好適なアプローチが説明される。「共通鍵」は上述したものと略同じ方法で生成される。前記共通鍵は基本的にリファレンスプレイヤー又は匿名プレイヤーを「アンロック（unlock）」するが、しかしながらこれらのプレイヤーは、本実施例においては、いずれの実際の実行装置によっても実行されない。上述のものと同

10

20

30

40

50

様に、本方法は $2N$ 個の反転可能な置換 p_i のセット ($1 \leq i \leq 2N$) を選択するステップと、 N 個の関数 g_i を算出するステップとを有する。ここで g_i は、 $1 \leq i \leq 2N$ について

【数 1 2 2】

$$p_{2i}^{-1} \circ f_i \circ p_{2i-1}$$

と関数的に同等である。本方法は加えて、それぞれが一意的なインデクス j によって識別される各実行装置について、該装置及び／又は該装置のユーザについて一意的な対応する $2N$ 個の反転可能な置換 $p_{j,i}$ のセット及び／又はシーケンスを選択するステップを有する。該セットは、一意的な「プレイヤ」を各装置に備えるために利用される。該一意的なプレイヤは、各実行装置 j について $N-1$ 個の関数 $h_{j,i}$ の対応するセットを算出し、各実行装置 j に

【数 1 2 3】

$$y_N \circ h_{j,N} \circ y_{N-1} \circ h_{j,N-1} \circ \dots \circ y_1$$

を含むそれぞれの実行装置関数縦続 $ED_j(y_1, \dots, y_N)$ を備えることにより形成される。ここで $h_{j,i}$ は、 $2 \leq i \leq N$ について

【数 1 2 4】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

に関数的に同等である。しかしながら、該装置特有のセット h_i は、セット h_i を利用するリファレンスプレイヤを「アンロック」することができる不明瞭化された関数縦続とは一致しない。該後者のセット／プレイヤのセットは、いずれの実行装置にも利用可能とはされない。その代わりに、実行装置 j がそれぞれのローダ関数

【数 1 2 5】

$$loader_j(x_1, \dots, x_N) = (l_{j,1} \circ x_1 \circ r_{j,1}, \dots, l_{j,N} \circ x_N \circ r_{j,N})$$

を備える。ここで $l_{j,i}$ は

【数 1 2 6】

$$p_{j,2i}^{-1} \circ p_{j,2i-1}$$

と関数的に同等であり、 $r_{j,i}$ は

【数 1 2 7】

$$p_{j,2i-1}^{-1} \circ p_{j,2i-2}$$

と関数的に同等である。上述のものと同様に、各実行装置は同一の関数 $g_1, \dots, g_N$ を備える。前記実行装置はこのとき $ED_j(loader_j(g_1, \dots, g_N))$ を実行する。本式において、 $loader_j(g_1, \dots, g_N)$ は事実上匿名鍵 $g_1, \dots, g_N$ を、実行装置関数縦続 $ED_j(y_1, \dots, y_N)$ に最適に一致する装置特有の鍵に変換する。定義： $loader_j(g_1, \dots, g_N) = (g_{j,1}, g_{j,2}, \dots, g_{j,N})$ を用いると、 $loader_j(g_1, \dots, g_N)$ の i 番目の要素は

【数 1 2 8】

$$g_{j,i} = l_{j,i} \circ g_i \circ r_{j,i}$$

である。上述の定義を用いると、このことは

【数 1 2 9】

$$g_{j,i} = p_{j,2i}^{-1} \circ p_{2i} \circ p_{2i}^{-1} \circ f_i \circ p_{2i-1} \circ p_{2i-1}^{-1} \circ p_{j,2i-1}$$

を与え、該式は

【数 1 3 0】

$$g_{j,i} = p_{j,2i}^{-1} \circ f_i \circ p_{j,2i-1}$$

と書き換えられる。このことは、装置特有のセット $h_{j,i}$ が置換を除去する、装置特有の置換セット／シーケンスを利用する場合と同様である。

10

【0 0 2 0】

匿名の不明瞭化された鍵及び装置特有のローダを利用する着想はまた、図 8 に示される。匿名プレイヤ P 1-R 8 1 0 は、関数 h_i を組み込む。匿名プレイヤ P 1-R は、不明瞭化された信号処理関数 f_i をセット g_i の形で含む、対応する鍵 K-R 8 1 2 によって開錠されることができる。匿名プレイヤ P 1-R は、いずれのユーザ／団体にも開示されていない。その代わり各ユーザ／団体は、一意的な装置特有のプレイヤ（図示されているものはプレイヤ P 1-1 (8 3 0) 及び P 1-2 (8 4 0)）を備えられる。共通鍵 K-R は全てのユーザ／団体に供給される。しかしながら、該共通鍵は、特定のプレイヤに合致しない。それ故、各ユーザ／団体はまた装置特有の鍵ローダ K-L（図示されているものは 8 2 0 及び 8 2 5）を供給される。ローダ 8 2 0 及び 8 2 5 は、匿名鍵 K-R 8 1 2 を装置特有の鍵 K-j に変換するために利用される。この目的のため、ローダ K-L_i は関数 $l_{i,j}$ 及び $r_{j,i}$ を含む。図 8 に示されたように、原則的に、装置特有のローダが利用される。図 9 に更に示されたように、実際には、前記ローダは同一であっても良いが、装置特有の関数 $l_{j,i}$ 及び $r_{j,i}$ と共に供給される。図 9 の例においては、 $l_{1,i}$ 及び $r_{1,i}$ と共に供給されることにより匿名鍵 K-R 8 1 2 を装置 1 用の装置特有の鍵 8 3 2 に変換し、 $l_{2,i}$ 及び $r_{2,i}$ と共に供給されることにより匿名鍵 K-R 8 1 2 を装置 2 用の装置特有の鍵 8 4 2 に変換する。装置特有のプレイヤ 8 3 0 及び 8 4 0 が次いで、それぞれ装置特有の鍵のセット $h_{1,i}$ （それぞれ 8 3 2 及び 8 4 2）を利用してアンロックされる。これらの例においては、「鍵」及び「プレイヤ」なる語句は交換可能である。なぜなら、関数の 2 つの連鎖が相互にロックするからである。図 4 の例は、両方の連鎖を鍵として説明している。同様に、このことは 2 つの相互にロックするプレイヤとしても説明され得る。

20

30

【0 0 2 1】

匿名プレイヤ 8 1 0 ($g_N, \dots, g_1$ を組み込む) は有利にも、放送及び／又は記憶媒体における配布を通じて、各実行装置について同一のコンテンツを用いて、各実行装置に供給されても良いことは理解されるであろう。このことは単に、該プレイヤが各装置について同一であるからである。同様に、各実行装置によって処理されるべきデジタル信号入力 x は、放送及び／又は記憶媒体における配布を通じて、各実行装置について同一のコンテンツを用いて配布されても良い。ローダに特有の態様は好ましくは、少なくとも 1 つの対応する関数のセット $h_{j,i}$ 、 $l_{j,i}$ 又は $r_{j,i}$ を用いて装置特有のコンテンツを用いて、「1 対 1 通信チャネル」及び／又は記憶媒体を介して、実行装置 j に供給される。「1 対 1 通信チャネル」は、いずれの適切な方法で実現されても良い。好ましくは、前記サーバは、インターネットを利用してセキュアなリンク（例えば SSL）を介して、前記装置特有の情報をダウンロードする。

40

上述したように、関数 f は Feistel 暗号ネットワークに基づく復号化関数であっても良く、信号処理関数 f_i の各々は、それぞれの Feistel 復号化ラウンド関数である。斯かる場合には、置換 p_i の各々は好ましくは、Feistel 変換器である。ここで連続的な対 $\langle x, y \rangle$ に演算する関数 Q は、反転可能な関数 Q_x 及び Q_y が存在し且つ $Q(\langle x, y \rangle) = \langle Q_x(x), Q_y(y) \rangle$ である場合には、Feistel 変換器である。ここで、

【数 1 3 1】

$$Q_x(x) \oplus Q_x(y) = Q_x(x \oplus y)$$

及び

【数 1 3 2】

$$Q_y(x) \oplus Q_y(y) = Q_y(x \oplus y)$$

である。これらの条件が満たされる場合、関数 f_i は最適に隠蔽される。実際には、多くの斯かるFeistel変換器が存在し、装置に特有な置換の選択のための十分な余地を与える。Feistel変換器の定義は、上述の定義を用いて、定義： $swap(<x, y>) = swap(<y, x>)$ 及び

10

【数 1 3 3】

$$involutory_F(\langle x, y \rangle) = \langle x, y \oplus F(x) \rangle$$

を用いると、Feistelラウンド

【数 1 3 4】

$$f_i(\langle L_{i-1}, R_{i-1} \rangle) = \langle R_{i-1}, (L_{i-1} \oplus F(R_{i-1}, K_i)) \rangle$$

が

20

【数 1 3 5】

$$f_i = swap \circ involutory_F$$

として見る事ができるという洞察に基づく。このとき $swap^{-1} = swap$ 及び $involutory_F^{-1} = involutory_F$ が成り立つ。

【0 0 2 2】

本発明は、本発明を実施するために適応されたコンピュータプログラム、特に担体上のコンピュータプログラムにも適用されることは理解されるであろう。前記プログラムは、ソースコード、オブジェクトコード、部分的にコンパイルされた形式のようなソースコードとオブジェクトコードの中間のコード、又は本発明による方法の実装における利用に適した他のいずれの形態であっても良い。前記担体は、前記プログラムを担持することが可能ないずれのエンティティ又は装置であっても良い。例えば前記担体は、例えばCD-ROM若しくは半導体ROMといったROMのような記憶媒体、又は例えばフロッピー（登録商標）ディスク若しくはハードディスクのような磁気記録媒体を含んでも良い。更に前記担体は、電気若しくは光ケーブルによって、無線によって、又はその他の手段によって伝達され得る、電気又は光信号のような伝送可能な担体であっても良い。前記プログラムが斯かる信号で実施化される場合には、前記担体は斯かるケーブル又は他の装置若しくは手段によって構成されても良い。代替として、前記担体は、前記プログラムが埋め込まれた集積回路であって、本方法を実行するために、又は本方法の実行における使用のために適応された集積回路であっても良い。

30

40

【0 0 2 3】

上述の実施例は本発明を限定するものではなく説明するものであって、当業者は添付する請求項の範囲から逸脱することなく多くの代替実施例を設計することが可能であろうことは留意されるべきである。請求項において、括弧に挟まれたいずれの参照記号も、請求の範囲を限定するものとして解釈されるべきではない。動詞「有する (comprise)」及びその語形変化の使用は、請求項に記載されたもの以外の要素又はステップの存在を除外するものではない。要素に先行する冠詞「1つの (a又はan)」は、複数のかような要素の存在を除外するものではない。本発明は、幾つかの別個の要素を有するハードウェアによって、及び適切にプログラムされたコンピュータによって実装されても良い。幾つかの手段を列記した装置請求項において、これら手段の幾つかは同一のハードウェアのアイテム

50

によって実施化されても良い。特定の手段が相互に異なる従属請求項に列挙されているという単なる事実は、これら手段の組み合わせが有利に利用されることができないことを示すものではない。

【図面の簡単な説明】

【0024】

【図1】 先行技術のプラグインベースのデコードのブロック図を示す。

【図2】 先行技術ベースの復号化のブロック図を示す。

【図3】 先行技術の一体化された復号化／デコードシステムのブロック図を示す。

【図4】 本発明による不明瞭化を示す。

【図5】 不明瞭化の単純な例を示す。

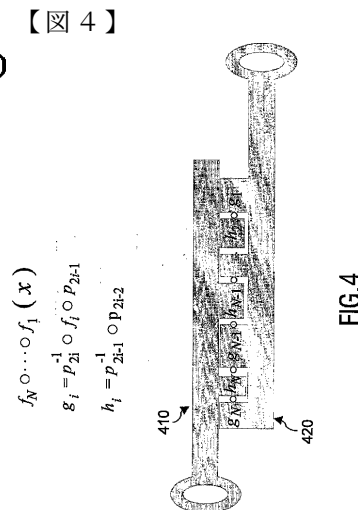
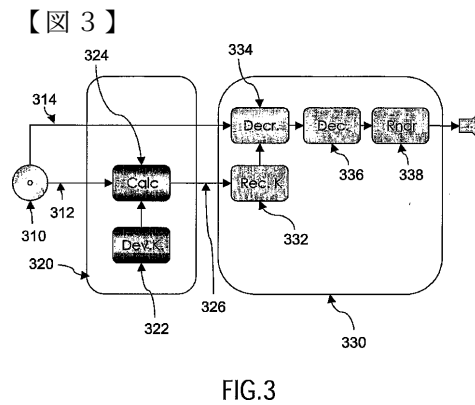
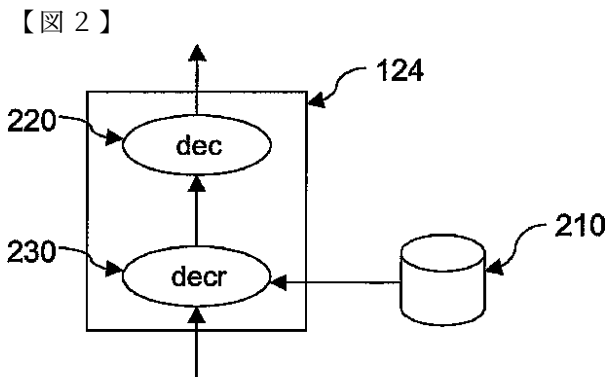
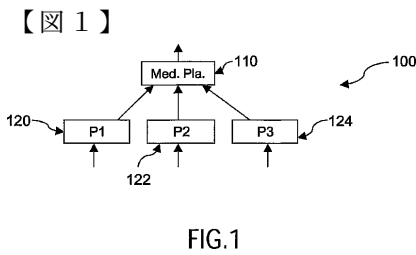
【図6】 本発明によるシステムのブロック図を示す。

【図7】 本発明によるシステムの更なる実施例を示す。

【図8】 本発明による匿名の不明瞭化を示す。

【図9】 匿名の不明瞭化のための代替実施例を示す。

10



【図 5】

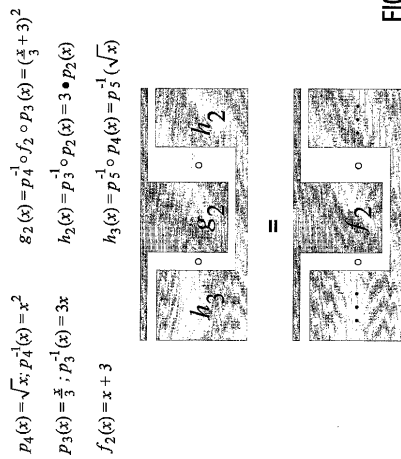


FIG.5

【図 7】

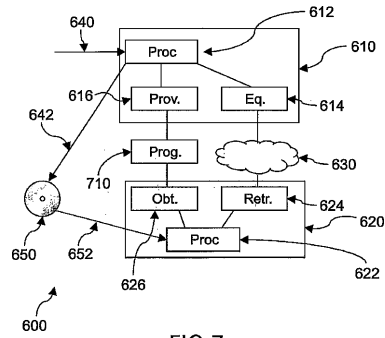


FIG.7

【図 8】

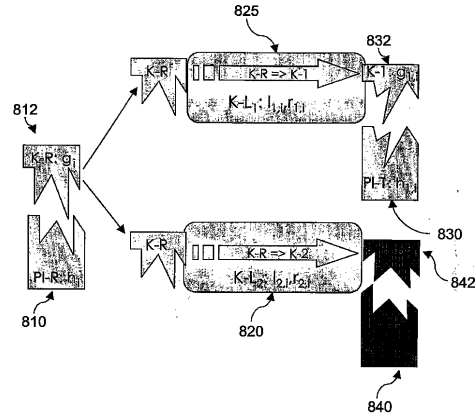


FIG.8

【図 6】

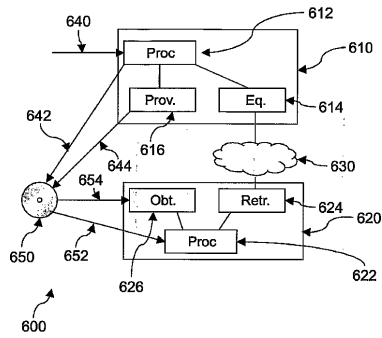


FIG.6

【図 9】

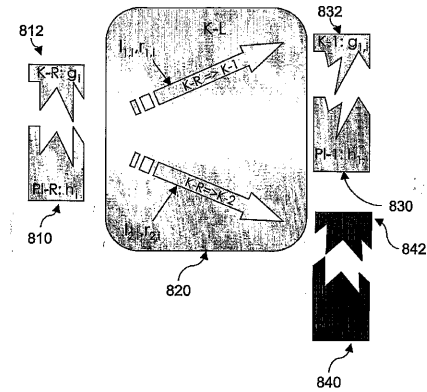


FIG.9

フロントページの続き

- (72)発明者 ホリッセン パウルス エム エイチ エム エイ
オランダ国 5 6 5 6 アーアー アインドーフエン プロフ ホルストラーン 6
- (72)発明者 トレシエル ヨアヒム エイ
オランダ国 5 6 5 6 アーアー アインドーフエン プロフ ホルストラーン 6
- (72)発明者 スタリング アントニウス エイ エム
オランダ国 5 6 5 6 アーアー アインドーフエン プロフ ホルストラーン 6
- (72)発明者 マッロン ウィレム シー
オランダ国 5 6 5 6 アーアー アインドーフエン プロフ ホルストラーン 6
- (72)発明者 トレップェルス メンノ エイ
オランダ国 5 6 5 6 アーアー アインドーフエン プロフ ホルストラーン 6

審査官 中里 裕正

- (56)参考文献 特開平08-016385 (JP, A)
国際公開第02/046890 (WO, A1)
特表2002-514333 (JP, A)
特開2001-067270 (JP, A)
松岡 賢 Masashi MATSUOKA, 鍵内蔵型暗号ソフトウェアの技術者による耐タンパー性評価法の事例紹介 A Case Study for Evaluating Tamper Resistance of Key-built-in Cryptographic Software by Engineers, 電子情報通信学会技術研究報告 Vol. 101 No. 311 IEICE Technical Report, 日本, 社団法人電子情報通信学会 The Institute of Electronics, Information and Communication Engineers, 第101巻

- (58)調査した分野(Int.Cl., DB名)

H04L 9/06

G09C 1/00

JSTPlus/JMEDPlus/JST7580(JDreamII)