(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4723141号
(P4723141)

(45)発行日　平成23年7月13日(2011.7.13)　(24)登録日　平成23年4月15日(2011.4.15)

(51)Int.Cl.		F I		
H04L 9/08	(2006.01)	H04L	9/00	601B
H04L 9/16	(2006.01)	H04L	9/00	643
H04L 12/22	(2006.01)	H04L	12/22	

請求項の数 8　(全 39 頁)

(21)出願番号	特願2001-512739 (P2001-512739)
(86)(22)出願日	平成12年7月20日 (2000.7.20)
(65)公表番号	特表2003-505978 (P2003-505978A)
(43)公表日	平成15年2月12日 (2003.2.12)
(86)国際出願番号	PCT/GB2000/002813
(87)国際公開番号	WO2001/008348
(87)国際公開日	平成13年2月1日 (2001.2.1)
審査請求日	平成19年7月4日 (2007.7.4)
(31)優先権主張番号	99305870.0
(32)優先日	平成11年7月23日 (1999.7.23)
(33)優先権主張国	欧州特許庁 (EP)

(73)特許権者　390028587
ブリティッシュ・テレコミュニケーションズ・パブリック・リミテッド・カンパニー
ＢＲＩＴＩＳＨ　ＴＥＬＥＣＯＭＭＵＮＩＣＡＴＩＯＮＳ　ＰＵＢＬＩＣ　ＬＩＭＩＴＥＤ　ＣＯＭＰＡＮＹ
イギリス国，イーシー１エー・７エー　ジェイ，ロンドン，ニューゲート・ストリート　８１
(74)代理人　100084618
弁理士　村松　貞男
(74)代理人　100092196
弁理士　橋本　良郎
(74)代理人　100095441
弁理士　白根　俊郎

最終頁に続く

(54)【発明の名称】データ分配

(57)【特許請求の範囲】

【請求項１】

データを複数のユーザ端末に安全に配信する方法であって、

（ａ）データサーバが、複数のデータユニットをそれぞれ、キーの第一のシーケンスのうちの異なるキーで暗号化する段階と、

（ｂ）前記データサーバが、暗号化したデータユニットを複数のユーザ端末に向けて通信網を経由して通信する段階と、

（ｃ）前記データサーバが、少くとも一つのシード値をキー管理ノードを介してユーザ端末に向けて通信する段階と、

（ｄ）前記ユーザ端末が、該少くとも一つのシード値から、該ユーザ端末に向けて通信されたシード値の数よりも大きな数のキーの第二のシーケンスを生成する段階と、

（ｅ）前記キーの第二のシーケンスを用いて該ユーザ端末がデータユニットを復号化する段階、とを備え、

段階（ｄ）では、段階（ａ）のキーの第一のシーケンスの中の任意に二重に境界を画成した部分を構成しているキーのシーケンスが生成され、

前記部分の下側の境界と上側の境界との中のシーケンス内の位置が、段階（ｃ）で通信された少くとも一つのシード値によって決められる段階を特徴とする、方法。

【請求項２】

（Ａ）少くとも一つの初期シード値に作用して、初期シード値をブラインドする、より大きな数の中間シード値を生成する段階と、

（Ｂ）段階（Ａ）で生成された前記中間シード値にさらに作用して、段階（Ａ）で生成された前記中間シード値をブラインドする、さらに大きな数の別のシード値を生成する段階と、

（Ｃ）生成された別のシード値の数が段階（ａ）で必要とされるキーの数以上となるまで、段階（Ｂ）を繰返す段階とにより、

段階（ａ）で使用されるキーの第一のシーケンスが生成される請求項１に記載の方法。

【請求項３】

段階（ｄ）が複数の異なるシード値から得られた値を組合せる段階を含む請求項１に記載の方法。

【請求項４】

前記段階（ｄ）は、

（Ｉ）異なるそれぞれのシード値を有する第一と第二のブラインド機能チェーンからそれぞれ得られた第一と第二の値を組合せる段階と、

（ＩＩ）第一の値の位置に続く第一のブラインド機能チェーン内の位置から得られた値と、第二の値の位置に先行する第二のブラインド機能チェーン内の位置から得られた値とを組合せて、それにより別の次のシードまたはキー値を生成する段階とを含む請求項３に記載の方法。

【請求項５】

段階（ＩＩ）を繰返して、それによって、別のキー値を生成し、

各繰返しでは、第一のブラインド機能チェーン内の前の位置に続く位置からの値と、第二のブラインド機能チェーン内の前の位置に先行する位置からの値とを組合せる請求項４に記載の方法。

【請求項６】

段階（ｄ）が、複数の異なるブラインド機能の各々を用いて、複数のシード値に作用する段階を含む請求項１に記載の方法。

【請求項７】

（Ｉ）一組の異なるブラインド機能の各々を用いて、少くとも一つのルートシード値に作用して、それにより複数の別の値を生成する段階と、

（ＩＩ）該一組の異なるブラインド機能の各々を用いて、段階（Ｉ）で生成された該別の値もしくはそこから得られた値に作用する段階と、

（ＩＩＩ）段階（ＩＩ）を繰返して、それにより、各繰返しにより、値のツリー内の次の継続するレイヤを生成する段階と、

（ＩＶ）段階（ＩＩＩ）で生成されたレイヤのうちの少なくとも一つレイヤにおけるシードのシーケンスから得られた値を、キーのシーケンスとして段階（ａ）において用いる段階と、

（Ｖ）段階（ｃ）において、ユーザ端末に向けて、ツリーの本体内部からの少くとも一つの値を通信して、値のツリー内の位置がユーザ端末に通信されることによって、データユニットを復号化するときに使用するためにユーザにとって利用可能なキーのシーケンスの部分の位置と拡がりとを判断する段階、とをさらに含む請求項６に記載の方法。

【請求項８】

前記段階（Ｉ）では、

（ｉ）該一組の異なるブラインド機能を用いて、複数の異なるシード値に作用する段階と、

（ｉｉ)異なるブラインド機能の各々について、一つのブラインド機能を用いてシード値のうちの一つのシード値に作用した結果と、同じ又は別のブラインド機能を用いてそれぞれのシード値のうちの別のシード値に作用した結果とを組合せて、それにより複数の別の値を生成する段階、とをさらに含む請求項７に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、データへの制御されたアクセスを与えるデータ分配（配信）システム（データデストリビューションシステム）に関する。例えばこのシステムはマルチキャストオーディオまたはビデオプログラム素材への限定された時間についてのアクセスを、ユーザによる前もってする支払に対して提供するために使用できる。しかし、この発明は当然のことながらマルチキャストパケット網で使用することに限定されるものではなく、例えば他のバルクデータ分配チャンネルと一緒に使用されてもよく、その中にはディジタルバーサタイルディスクＤＶＤのような記憶媒体が含まれている。

【０００２】

【従来の技術】

マルチキャスト技術はインターネットで使用するために開発されてきており、データソースから多数の受け側に向けてデータを効率的に分配できるようにしている。しかしながら、既存のマルチキャスト用プロトコルの効率とスケーラビリティとは部分的にはデータソースがデータ受け側についての知識をなにも必要としていないという事実に依存している。しかしこのことが問題を呈示しており、データソースと受け側との間で安全な（セキュリテイのある）関係を設定しようとするときに、例えばそれによってテレビジョンプログラムのような流れとなっているビデオデータがそのプログラムを受取るために支払をした加入者だけに送られるようにするときに問題を生ずる。

【０００３】

【発明が解決しようとする課題】

一般に、このような安全な関係はデータソースにおいてデータを暗号化（エンクリプト）することによって、またその上でユーザによるアクセスをそのデータを解暗号化（解号化、デクリプト）するのに必要とされるキーについて制御することによって設定することができる。一つの単純なやり方はデータを暗号化するために単一のセッションキーを使用することである。このセッションキーは新しいユーザがそのデータへのアクセスを希望するまでは、あるいは既存ユーザの一人が排除されるまでは不変に保たれる。この時点で、新しいセッションキーが必要とされて、そのキーがすべてのユーザに分配されなければならない。キー分配機構の効率がキーの階層構造を用いることにより幾分かは改善できるのではあるが、特定の顧客を排除したり、含めたりするために変更される必要があるのはそのうちの一部だけであり、このようなスキーム（機構）では、かなりの伝送オーバーヘッドが新しい顧客の群（グループ）への加入か退去と関係して不可避のままとなっている。

【０００４】

この発明の出願人の未決国際特許出願ＰＣＴ／ＧＢ９８／０３７５３（ＢＴ事件番号：Ａ２５７２８／ＷＯ）に記述されている代りのやり方では、データソースにおけるデータが一連の応用データユニット（ＡＤＵ）と分けられて異なるキーが各ＡＤＵに対して使用されている。このキーは初期シード（種子）値からシーケンスとして組織的に生成されている。このシード値はまた各顧客端末にある安全なモジュールに向けて通信され、またこの安全なモジュールがエンドユーザに向けたキーの利用可能性を制御している。

【０００５】

【課題を解決するための手段】

この発明を第一の観点でとらえると、

（ａ）複数のデータユニットをそれぞれキーのシーケンスの一つで暗号化することと；

（ｂ）暗号化したデータユニットを複数のユーザ端末に向けて通信することと；（ｃ）少くとも一つのシード値をユーザ端末に向けて通信することと；

（ｄ）該シード値から、該ユーザ端末に向けて通信したシード値の数よりも大きな数のキーのシーケンスを生成することと；

（ｅ）前記キーのシーケンスを用いて該ユーザ端末でデータユニットを解暗号化することとを備え、

該段階（ｄ）では、段階（ａ）のキーのシーケンスについての任意に二重に境界を画成した部分を構成しているキーのシーケンスが生成されることと、

前記部分の下側と上側との境界のシーケンス内の位置が段階（ｃ）で通信された少くとも

一つのシード値によって決められることとを特徴とするデータを分配（配信）する方法が提供されている。

【０００６】

この発明はデータを分配（配信）する方法を用意しており、そこでは、出願人の上記引用の未決出願に開示されているシステムのように、継続するデータユニットが異なるキーのシーケンスを用いて暗号化されている。しかしながら、この発明の方法はもっと著しい利点を与えており、それは各ユーザにとって利用可能なキーのシーケンスの程度（拡がり）が潜在的には無制限というのではなく、これは初期のシステムと似ていることであるが、しかし、二重に境界が画成されていて（ダブリイバウンデッド）、言い換えると、ユーザにとって利用可能なキーのシーケンスの初めと終りとが前もって決められている。さらに後述することになるように、データの送り側、あるいはデータの送り側に代って動作しているキー発行当事者は任意に出発点と終了点とを決めることができ、したがってユーザにとって利用可能なキーのシーケンス長はどのシード値がユーザに対して送られるかを選ぶことによって任意に決められる。キーのシーケンスのいずれもの所望部分について、一組のシードが存在し、これがその部分に対し、かつその部分だけに対してアクセスを与えることになる。ユーザに、潜在的に無制限なキーの組ではなく、二重に境界を画成したキーの組へのアクセスを与えることにより、この発明は、各顧客端末において安全なモジュールをもち、データ送り側の制御下でユーザのキーへのアクセスを制御する必要性を除去している。

【０００７】

好ましいのは、段階（ａ）で使用されるキーのシーケンスが、

（Ａ）いくつかの初期シード値に作用して中間シード値として初期シード値をブラインドとするより大きな数の中間シード値を生成することと；

（Ｂ）さらに前段階（Ａ）で作られた中間シード値に作用して、前段階で作られた中間シード値をブラインドとし、それによりさらに大きな数の別の値を生成することと；

（Ｃ）段階（Ｂ）を作られた値の数が段階（ａ）で必要とされるキーの数以上となるまで繰返すこととにより生成される、ことである。

【０００８】

好ましいのは、この方法が、

（ｉ）一つのルートシード値を異なるブラインド機能の組の各々と作用させてそれにより複数の別の値を作ることと；

（ii）該異なるブラインド機能の各組を前段階で作られた該別の値もしくはそこから得られた値に作用させることと；

(iii）段階（ii）を繰返して、それにより、各繰返しにより値のトリー内で次の継続するレイヤを作ることと；

（iv）段階（ａ）では、段階(iii)で作られたいくつかのレイヤにおけるシードのシーケンスから得られた値をキーのシーケンスとして用いることと；

（ｖ）段階（ｃ）では、ユーザ端末に向けて、トリーの本体内部からの少くとも一つの値を通信して、ユーザ端末に向けて通信された値のトリー内の位置が、それによって、データユニットを解暗号化するために使用するためのユーザにとって利用可能なキーのシーケンスの部分の位置と拡がりとを判断することとを含むことである。

【０００９】

ブラインド機能は入力値に作用して出力値を作る機能であり、ここでは、その機能が既知であってもなお、入力相が出力値からすぐには得ることができない。ブラインド機能は、例えば、ＭＤ５（メッセージダイジェスト番号５）のような暗号記法上のハッシュ機能を備えていてよい。

【００１０】

発明者らは、一連のキーを組織的に生成しながら、ユーザにとって利用可能とされるシーケンスの部分の位置と拡がりとについて容易に制御できるようにするとくに効率的な方法が、異なるブラインド機能の組を用いて反復繰返される動作によって値のトリー（木）を

生成することを発見した。以下詳細に記述する例では、二進トリーが一対の対称的なブラインド機能から形成されて使用される。一つの機能は右回転シフトであり、それに続いてハッシュ機能がされ、また他の機能は左回転シフトであり、それに続いてハッシュ機能が行なわれる。この発明のこの特徴は、しかしながら、二進のトリーで使用するのに限定されてはおらず、三進あるいはより高次のトリーを用いても実施できる。

【００１１】

好ましいのは、段階（ｃ）で、シード値が顧客端末に向けて通信網を経由して通信されることであり、さらにこの場合にはシード値が顧客端末に向けて複数のキー管理ノードから通信されるのがよく、このノードは異なるそれぞれの位置で網に接続されている。

【００１２】

この発明の好ましい実施例における別な利点は、符号化されたデータへのアクセスを用意するためにシード値を分配するプロセスが異なる位置にある多数のキー管理ノードに向けて展開できることであり、これらの位置の一部もしくは全部はデータソースから遠隔にあってもよい。このやり方では、データ制御システムは大きな数の受信側で使用するために直ちにスケールを合わせるスケーラビリティが備わっている。

【００１３】

一般にデータユニットとシード値とは同じ通信網上で分配される。しかし、これは必要条件ではない。例えば、データユニットはバルクデータ記憶媒体であるＤＶＤのようなものの上に分散されていて、シード値が後にインターネットを経てオンラインで顧客に通信されるようにしてよい。こういった例は、例示として与えたにすぎず、様々な違った実施が採用されてよいことは明らかであろう。

【００１４】

全体のキーシーケンスの特有のサブレンジについてのキーを構築するためにいずれかの受け側により求められるシードは各シードがどれであるかを黙示的（明示的の反対）に識別する順序で通信されるのが好ましい。この場合にシードとシードを通信するために前もって配列された順序とのインデックスは求められている最小と最大の値の知識から推論され、各シードについてのインデックス番号を明白にリストすることはなしでよい。好ましいのは、各暗号化されたデータユニットが暗号化されていないインデックス番号を帯していて、いずれもの受け側を識別するようにしていて、この受け側のシーケンス内のキーがデータユニットの解号化に使用されるべきことである。

【００１５】

この発明を別な観点でとらえると、分配用にデータを暗号化する方法であって：

（ａ）少くとも一つのルートシード値にいくつかのブラインド機能を作用させて、それにより複数の別な値を作ることと；

（ｂ）いくつかのブラインド機能を前段階で作られたかそこから得られた別の値に作用させることと；

（ｃ）段階（ｂ）を繰返して、各繰返しにより値のトリー内に次の継続するレイヤを作ることと；

（ｄ）段階（ｃ）によって生成されたいくつかのレイヤから得られたキー値のシーケンスを用いて複数のデータユニットを暗号化することとを備えた方法が提供されている。

【００１６】

この発明を別な観点でとらえると、複数のデータユニットの各々をキーのシーケンスの一つで暗号化して、該暗号化したデータユニットを複数のユーザ端末に向けて通信することとを備えたデータを分配する方法であって、

該キーのシーケンスはキー構成アルゴリズムにより応用データユニットに向けて生成されて割当てられることと、

該キー構成アルゴリズムのコピイが複数のキーマネージャに分配されて、それにより使用時には受け側が、データ送り側を参照することなく、キーマネージャから該データの任意の部分に向けてアクセスするためのキーを得られるようにすることを特徴とする方法が提供されている。

【００１７】
この発明のこの特徴はデータ分配のやり方として、キー管理が多数のキー管理ノードで展開でき、システムが多くの数のユーザに対して拡げられるようにスケールをとることができるようにする。
【００１８】
この発明の別の特徴によると、先に述べた特徴による方法で使用するために暗号化された複数のデータユニットを含んでいるデータキャリヤが用意されている。このデータキャリヤは例えばＤＶＤディスクのようなデータ記憶媒体であり、計算機メモリのある領域であったり、データユニットとともに符号化されたデータ伝送信号であってよい。
【００１９】
この発明は、顧客端末、データサーバ、及びキーマネージャでこの発明と一緒に使用するもの、このようなデバイスとそれらを含む網で使用する方法にも展開される。
【００２０】
この発明を実施するシステムを添付の図面を参照して詳述して行く。
【００２１】
【発明の実施の形態】
データ通信システムはデータサーバ１を含み、データサーバ１はデータ通信網２を経て多数の顧客端末３に接続されている。例示を簡単にするために、僅かな顧客端末だけが示されているが、実際には、データサーバ１は同時に多数の端末と通信する。この例では、データ通信網２は公開インターネットであり、多数のサブ網２Ａ～２Ｃがノード４によって相互接続されて形成されている。サブ網と関係するルータとはインターネットプロトコル（ＩＰ）マルチキャステングをサポートしている。
【００２２】
この例では、データサーバ１はビデオサーバである。データサーバはビデオデータストリームを大形メモリデバイスから読取って、ＭＰＥＧ２のような適当な圧縮アルゴリズムを用いてデータを圧縮する。データサーバ１内の暗号モジュールはそこで圧縮したビデオデータストリームを応用データユニット（ＡＤＵ）に分ける。例えば、各ＡＤＵはビデオ信号の１分間分に対応するデータを備えている。暗号アルゴリズムがそこで使用されて、ＡＤＵを組織的に発生されたキーのシーケンスで暗号化する。各ＡＤＵに対しては異なるキーが使用される。適当な暗号アルゴリズムにはＤＥＳ（data encryption standard）（US Federal standard FIPSPUB 46）が含まれている。これは従来形のプライベート（私的）キーアルゴリズムである。キーのシーケンスを発生する際に使用されるシード値もデータサーバ１から多数のキー管理ノードへ送られる。キー管理ノードは異なる位置でデータ通信網を介して拡散されている。顧客端末の一つからの要求に応答して、キー管理ノードはその端末に向けて多数のシード値を通信する。シード値を発する前に、それぞれのキー管理ノードはチェックを実行して、例えば関係している顧客端末が要求されたデータをアクセスする権利をもつことを設定するようにしてよい。例えば、顧客はビデオデータサーバ１からマルチキャストされている特定のフィルムに対するアクセス権利を要求していてよい。このフィルムがペイパービューベースで（pay-per-view basis，支払いをすると見ることができるシステムで）利用できる場合には、キー管理ノードはその顧客がビデオデータサーバのオペレータのところに口座をもっていて、そのフィルムに対して適切な支払いを済ませていることをチェックする。こういった条件が適えられていると、キー管理ノードは選ばれたシード値を顧客に向けて発して、その顧客がそのフィルムを構成しているＡＤＵを暗号化するためにデータサーバで使用されたキーシーケンスの部分に対応しているキーを生成できるようにする。以下にさらに記述するように、キーシーケンスの生成に使用されるアルゴリズムは、シード値の適切な選択がもとのキーシーケンスの任意に境界を画成した部分へのアクセスを与えるために使用することができる。
【００２３】
図２は顧客端末３の一つの主な機能部品を示す。網インターフェース（Ｉ／Ｆ）２２はデータ通信網２との間でＡＤＵを通信する。ＡＤＵはインターフェース２２からアクセスモ

ジュール２３へ進む。前のシステムではアクセスモジュール２３が別個の安全なモジュール内部、例えばスマートカード上に置かれるようにできたのとは対照的に、この発明を実施するシステムでは、アクセスモジュールは単に、顧客端末の主プロセッサ上で実行されているソフトウェアモジュールとすることができる。アクセスモジュール２３は解号化モジュールＤと、キー生成モジュールＫと、シードメモリＳＳとを備えている。シードメモリはキー管理ノードから受領したシード値を記憶して、これらシード値を、キー構成アルゴリズムで後に詳述するようなものを用いて、処理して一連のキーを生成する。このキーシリーズは始点と終点としてシードメモリＳＳ内に保持されているシード値で決まるものを有している。このシーケンスからのキーが解号化モジュールＤにシーケンシャルに送られる。解号化モジュールＤはインターフェース２２から受領した一連のＡＤＵを解号化して、それらを応用レイヤモジュール（ＡＰＰ）２４に送る。ここがさらに処理をするが、例えばＭＰＥＧ２解号化アルゴリズムを用いて処理し、その結果データを出力デバイスに送り、そこはこの例ではビデオディスプレイユニットＶＤＵ２５となっている。好ましい実施としては、インターフェース２２はハードウェアでＩＳＤＮモデムによって実施してよいし、またソフトウェアではＴＣＰ－ＩＰ（Transport Control Protocol-Internet Protocol）スタックで実施してよい。

【００２４】

顧客端末は多数の形式のいずれか一つで実施されてよい。例えば、端末は適当な網インターフェース、インテリジェントモバイルフォン、及びテレビジョンと一緒にインターネットアクセスを提供するように設計されたセットトップボックスを備えたパーソナルコンピュータを含むことができる。

【００２５】

図３は図１の網で使用するためのキー管理ノードの一例の構造を示す。このノードはパケットをデータ送り側と顧客端末との両方と通信するし、あるいはＴＣＰ－ＩＰスタックを経て“受け側（receiver）”と通信する。パケットは安全なソケットレイヤ（ＳＳＬ）３２上で通信され、公開キー暗号アルゴリズムが通常形式で使用される。キー管理応用３３はシード値をデータ送り側から受領し、シード値を顧客端末へ向けてさらに詳述するやり方で発する。データメモリ３３０はキー管理応用３３と関係していて、各データ送り側から受領したシード値を保持している。ユーザはキー管理応用とユーザインターフェース３４を経て対話するが、ユーザインターフェース３４は、例えばＨＴＭＬ（ハイパーテキストマークアップ言語）とサーバウエブページに向けたＣＧＩとを顧客端末に向けて使用してよい。

【００２６】

図４は図１の網で使用するためのデータ送り側の一例の構造を示す。データ応用４１はデータを出力し、このデータは、この例では、ＭＰＥＧ２ビデオストリームで応用データユニットに分けられたものを備えている。ビデオプログラム素材はメモリ４１０から得られる。ＡＤＵはアクセスモジュール４２へ向けて送られる。これは暗号サブモジュールＥと、キー生成サブモジュールＫとシードメモリＳＳとを含んでいる。サブモジュールＫはキーのシーケンスを生成するが、ランダムに発生されたシード値をキー構築アルゴリズム（後に詳述するようなもの）と一緒に使用して生成する。シード値はまたアクセスモジュール４２から安全なソケットレイヤ（ＳＳＬ）４３とＴＣＰ－ＩＰスタック４４とを経て出力されてもよい。暗号化されたＡＤＵはまたＴＣＰ－ＩＰスタック４４を経て出力される。

【００２７】

データサーバとキー管理ノードとの両方は市販のプラットホームであるCOMPAQ Proliant^{TM}サーバとかSun Microsystems Enterprise 5000^{TM}サーバを用いて実施されてよい。

【００２８】

図５はデータ送り側により出力されたデータフレームの一つのフォーマットを示す。これは暗号化したＡＤＵと、キーインデックスk_iと、セッション識別子ＳＥとを運んでいるデータペイロードを含んでいる。キーインデックスとセッション識別子とは暗号でない平

文で送られる。
【００２９】
一般にキー値は顧客端末に向けてＡＤＵの分配で使用したのと同じ通信網上で分配されてよい。
【００３０】
“応用データユニット（ＡＤＵ）”という用語は、この明細書及び請求の範囲ではセキュリテイ（安全）もしくは商用という視点から有用であるとされたデータの最小単位（ユニット）を記述するために使用されている。ＡＤＵの大きさは応用と必要とされるセキュリテイにより変わってよい。それは初期化フレームでありまた関係する“Ｐ－フレーム”の組でビデオシーケンス内にあるものであってよいし、網ゲームへのアクセスの１０分間（時間）であってもよい。暗号用に使用されるＡＤＵは応用の異なるレイヤで使用されるものとは異なっていてよい。例えば、現在の例では、ＡＤＵはＭＰＥＧ圧縮アルゴリズムで処理されるビデオフレームに対して異なる継続時間を有していて、また顧客により購入される個々のプログラムアイデアからもまた異なる継続時間を有していてよい。このシステムの性能を強化するために、ＡＤＵは部分的に限り暗号化され、他の部分は平文で送られてよい。ＡＤＵの大きさはコンテンツに依存するストリームの継続時間を介して可変とされてよい。応用データユニット（ＡＤＵ）の大きさは、システムのスケーラビリティにとっての主たる決定的要因（プライマリイデターミナント）であり、とくに百万の受け側が１５分間内に一つのマルチキャストデータストリームに加わろうとしているが、ＡＤＵの大きさもまた１５分であるという条件の下では、これが一つの再キーイベントを必要とするだけのこととなるからである。違ったキーシーケンス構築アルゴリズムがもっと詳細に記述されることになる。
【００３１】
送り側が結合を解いた構造（Sender-decoupled architecture）
この発明は、図１の方法のような単純な時間シーケンスでの使用に限定されない。例えば、この発明は大規模な網ゲームに応用されてよい。
【００３２】
このようなゲームでは、ＡＤＵの経済的価値は時間とかデータ量とかには関係せず、完全に応用特有の因子だけに関係する。この例では、参加者は“ゲーム－時間”当りの課金がされ、継続時間は実時間の時分（一分、二分．．．の分）とは厳密には関係していないが、ゲームの時間監視者（タイムキーパ）によって規定され、信号を送られる。このゲームは数多くの仮想ゾーンで構成され、その各々は異なるゾーンコントローラによって調整（モデレート）されている。ゾーンコントローラは背景となる事象（バックグラウンドイベント）と、寿命に係るゾーンを与えるデータとを提供し、ゾーンについてのマルチキャストアドレス上で暗号化したこのデータを送るが、同じＡＤＵインデックスが、したがってキーが、すべてのゾーンである一時刻に使用される。こうして、全体のゲームが一つの単一の安全なマルチキャストセッションとなり、数多くのマルチキャストアドレスにまたがって拡散していてもそれが行なわれる。遊戯者（プレーヤ）は現在のキーをもっている限りは長きにわたって、いずれかのゾーンについての背景データに同調することができる。そのゾーン内で遊戯者により作られたフォアグラウンド（前面に出た）イベントは暗号化されていないが、この背景データを参照することなしには意味のないものとなっている。
【００３３】
図６はこのようなゲームでのデータの流れを示す。ゲームの安全に関係する流れと、ゲームが進行中（設定中ではない）に一度送られた流れだけが示されている。すべての遊戯者がデータを送っているが、この図は暗号化している送り側Ｓすなわちゾーンコントローラだけを示している。同じように解号化している受け側Ｒだけが示されており、これがゲーム遊戯者である。ゲームコントローラはゲームの安全を設定するが、この安全は図示されておらず、以下で記述される。キー管理作用（操作、動作、オペレーション）は多数のレプリカとしたキーマネージャＫＭに対して委ねられていて、ＫＭは安全なウエブサーバ技術を使用する。

【００３４】
安全なマルチキャストセッションへのキーはシーケンス内のゲーム時分毎に（ＡＤＵ毎に）変更される。暗号化されたデータはその頭にＡＤＵインデックスが平文で付けられていて、これがその解号化で必要とされるキーを参照している。設定フェーズ後に、ゲームコントローラと、ゾーンコントローラとキーマネージャとは初期シードを保持し、これがコントローラとマネージャに対してゲームの全継続時間中使用されることになるキーのシーケンスを計算できるようにしている。代って、段階を追う（ステージとした）設定も使用できる。
【００３５】
ゲーム設定
１．ゲームコントローラ（図示せず）は共用“制御セッションキー”をすべてのＫＭとＳとに向けてそれらの識別の認証をそれ自体で満足なものとした後にユニキャストする。すべてのＳはすべてのＫＳとともに安全なウエブサーバを運用していて、それによりセッションキーが各々に向けて、クライアントが認証した安全ソケットレイヤ（ＳＳＬ）通信を用いて、各公開キーで暗号化したものを送られるようにできる。ゲームコントローラはまた制御メッセージ用に使用することになるマルチキャストアドレスをすべてのＫＭとＳとに通知して、直ちに参加するようにする。
２．ゲームコントローラはつぎに初期シードを生成して全体のキーシーケンスを構築するようにし、これらのシードをすべてのＫＭとすべてのＳとに向けてマルチキャストし、制御セッションキーでメッセージを暗号化して、信頼性のあるマルチキャストプロトコルで関与しているターゲットの恐らくは少数に対して適切とされているものを用いるようにする。
３．そのゲームが認証されたセッションディレクトリイアナウンスメント内でアナウンスされる。このアナウンスメントはMark Handler（ユニバシティカレッジロンドン，ＵＣＬ）の“On Scalable Internet Multimedia Conferencing System”ＰｈＤ学位請求論文（１４－１１－１９９７）に記述されており、マルチキャスト（図示せず）上で規則的に繰返しがされる。認証されたアナウンスメントはゲームの収入を集めるためにいんちきな（spoof）支払いサーバを設定する攻撃者（アタッカ）を排除する。このアナウンスメントプロトコルはキーマネージャアドレスの詳細と、ゲーム時分当りの価格とを含むために強化される。このキーマネージャはこのアナウンスメントを受け側と一緒によく聴いて、ゲーム時分の現在価格を得るようにする。このアナウンスメントはまたどのキーシーケンス構築コンストラクション）が使用されているかを特定することもしなければならない。
【００３６】
受け側セッション設定、継続及び終結（Receiver session set-up, duration and termination）
１．ゲームに参加するために支払をしたいとする受け側は、セッションディレクトリイ内での広告がされていることを聞いた上で、ＫＭウエブサーバと接触をとり、適当な形式を用いてある数のゲーム時分を要求する。このことが図６の“ユニキャスト設定”として示されている。ＲはＫＭに要求したゲーム時分の対価を支払うが、おそらくはその者のクレジットカード詳細を与えるか、何らかの形式の電子マネーが前のゲームで獲得したトークンを与えることになる。そのお返しに、ＫＭは中間シードの組を送り、それがＲにその者が購入したキーシーケンスのサブレンジだけを計算できるようにする。このキーシーケンス構築は次節で記述されていて、構築を効率的に可能なものとしている。こういったすべてのことが安全なソケットレイヤ（ＳＳＬ）通信上で認証を必要とするＫＭだけで行なわれ、Ｒによらない。
２．Ｒはその者が購入した中間シードを用いて関係するキーと生成する。
３．Ｒはゲーム応用によって決められた関係するマルチキャストに参加する。このマルチキャストの一つは、常に、一つのＳからの暗号化された背景ゾーンデータである。Ｒは前段階で計算されたキーシーケンスを用いてこれらのメッセージを解号化して、それによりゲームデータのその余の部分を意味あるものとする。

４．タイムキーパが新しいゲーム時分の信号を（制御マルチキャスト上で）送るときはいつでも、すべてのゾーンコントローラはそのＡＤＵインデックスを歩進（インクレメント）させて、シーケンス内の次のキーを使用する。ゾーンコントローラはすべてが同じＡＤＵインデックスを用いる。各ＲはＡＤＵインデックスでＳからのメッセージの中にあるものが歩進されて、適切なシーケンス内の次のキーを使用していることに注目する。
５．ゲーム時分インデックスが、Ｒが購入したシーケンスの終りに近付いたときには、この応用は遊戯者に“硬貨挿入”警報をその者がアクセスを失う前に与えている。ゲーム時分はこの点に達するまでは歩進を続け、達すると必要とされる。キーはＲが計算をすることが可能な範囲外となる。Ｒがもっとゲーム時分を購入しなければ、その者はゲームから外されなければならない。

【００３７】
このシナリオは、キーマネージが各ＡＤＵインデックスの経済的な値もしくは各ＡＤＵに対するアクセスポリシイを何らかの予備的な構成（プレアレンジメント）を介して知ることを条件として、どのようにして送り側がすべての受け側が活動に参加しまた退去することから完全に結合を解く（分解する）ことができるかを示している。ここではキーマネージャと送り側との間で何らの通信をする必要はない。送り側はいずれの受け側の活動について知ることを決して要求されない。もしキーマネージャがすでに送ってしまったＡＤＵを売ることを回避する必要があるとすれば、キーマネージャは単に送り側からのＡＤＵシーケンス番号の変化しているストリーム（流れ）と同期をとる必要があるだけである。例をあげると、キーマネージャはマルチキャストデータそれ自体に聴き入ることにより同期する。他のシナリオでは同期は純粋に時間応用であり、明白な同期信号を介するか、一日のうちの時間同期を黙示的にとるかのいずれかによる。また別なシナリオでは（例えば商用ソフトウェアのマルチキャストでは）、伝送の時刻は重要とされない。例えば、伝送は規則的に繰返され、受け側はシーケンスの一部についてのキーが売られていて、受け側ではその後のいつでもよい時刻にチューニング（同期をとる）をすることができる。

【００３８】
この例では、前納がシード購入に使用されている。これはキーマネージャがその者の顧客についての状態を何も保持しないことを確かなものとしており、これが意味するところは中央の状態保管場所を必要としないので、無限に応答ができることであり、もしシードが口座で購入され、顧客の口座状態がチェックされることを要するといった場合と異なっている。

【００３９】
キー構築の別な方法を記述して行く。

【００４０】
キーシーケンス構築（構成、コントラクション）
以下に述べるキーシーケンス構築では、次の記法が使用される。
・ｂ（ｖ）はｖの値をブラインドする機能について使用される記法である。言い換えると、計算機上で制限されている（ゲームの）相手（adversary）はｂ（ｖ）からｖを見付けることができない。ブラインデングもしくは片道機能（ワンウェイファンクション）の例はハッシュ機能であり、その例はＭＤ５ハッシュ［IETF RFC 1321］もしくは標準のSecure Hash１［NIST Sha-1］である。良いハッシュ機能は一般に軽量の計算機処理資源だけを必要とする。ハッシュ機能は固定の大きさの出力にまでいずれの大きさの入力も減らすように設計されている。すべての場合に、ここでは出力と同じ大きさとすでになっている入力を使用することとするが、単にハッシュというブラインデング道具（プロパティ）を用いるだけであり、大きさを減少する道具としてではない。
・b^h（ｖ）は機能ｂ（　）が繰返し前の結果に適用され、その回数が全部でｈ回であることを意味する。
・ｒ（ｖ）はいずれかの計算機上で高速の１対１機能であり、これが入力値の組からそれ自体への写像を行なう。円形の（回転式、ロータリイ）ビットシフトがこのような機能の一例である。

・ｃ（v_1，v_2，…）は値v_1，v_2等を組合せる機能であり、それによって結果と１を除くすべてのオペランドとが与えられると、残りのオペランドが自明として容易に求めることができる。Ｃ（　）は次のように選ばれなければならない。すなわちオペランドのビットが独立していて、バイアスされていないとすると、結果のビットもまた独立していてバイアスされないものとなることである。ＸＯＲ機能はこのような組合せ機能である。ｃ（　）はまた理想的にはその余のオペランドを自明なものとして求めるために使用できるものであり、これはＸＯＲの場合のようなものであって、言い換えるとv_1＝ｃ（ｃ（v_1，v_2，…），v_2，…）である。

【００４１】

すべての構造についての共通なモデルは４．５節で与えられるが、それ自体の用語について先ず各スキームを導入する方がよりはっきりすると思う。

【００４２】

双方向性ハッシュチェーン（Bi-directional hash chain,BHC）

双方向性ハッシュチェーン構造は限られた形式の中で安全であることを証明するだけであるが、発明者らはこれを限られたバージョンが後のスキームの基礎を形成するとして記述することに固執する。限定されていない形式が使用されるというシナリオもまたあってよい。

１．送り側はランダムに二つの初期シード値ｖ（０，０）とｖ（０，１）とを生成する。一つの具体例として、これらの値が１２８ビット幅であるとする。

２．送り側は要求された最大キーシーケンス長Ｈについて判断をする。

３．送り側は繰返して同じブラインド機能を各シードに適用して等しい長さＨの二つのシードチェーンを作るようにする。この値はしたがってｖ（０，０）ないしｖ（Ｈ－１，０）とｖ（０，１）ないしｖ（Ｈ－１，１）である。項Ｈ－１はしばしば出現するので、別の定数Ｇ＝Ｈ－１を導入することとする。

【００４３】

したがって、正式には、

$v(h,0)=b^h(v(0,0))$；$v(h,1)=b^h(v(0,1))$　　(4.1.1)

である。

４．キーk_0を作るために、送り側はチェーンゼロ、ｖ（０，０）からの第一のシードをチェーン１、ｖ（Ｇ，１）からの最後のものと組合せる：

キーk_1を作るために、送り側はチェーンゼロ、ｖ（１，０）からの第二のシードをチェーン１、ｖ（Ｇ－１，１）からの終りから二番目のものと組合せる。こうしたことが続く：

正式には、$k_h=c(v(h,0),v(G-h,1)$　　(4.1.2)

である：

厳密には、使用されるストリームサイファ（流れの暗号）は１２８ｂのキーを必要とはしないので、より短いキーがこの組合せの結果から最上位（または最下位）ビットを切り落として、一般には６４ｂとして求められてよい。ストリームサイファは早くて安全である限り不適切とはならない。

５．送り側はストリームのマルチキャステングを開始し、ADU_0（応用データユニット０）をk_0で、ADU_1をk_1でというふうに暗号化するが、少くともＡＤＵシーケンス番号は平文で残すようにする。

６．もし送り側がキー管理を委ねていれば、二つの初期シード値をキーマネージャに私的に（公開せずに）通信しなければならない。新しい初期シード対が生成されることが可能で、キーマネージャに向けてもっと早くに計算されたキーで暗号化されたストリーミングデータと並列に通信できる。

【００４４】

受け側はシーケンスの一部を次のように再構築する：

１．受け側がADU_mからADU_nへのアクセスを許可されると、送り側（もしくはキー

マネージャ）はシードｖ（ｍ，０）とｖ（Ｇ－ｎ，１）とをその受け側へユニキャストする。
２．その受け側はシードチェーンｖ（ｍ，０）ないしｖ（ｎ，０）とｖ（Ｇ－ｎ，１）ないしｖ（Ｇ－ｍ，１）とを、式（4.1.1）を用いて送られたシードに対してブラインド機能を繰返して適用することによって作る。
３．受信側は、送り側がしたように、式（4.1.2）を用いてキーk_m ないしk_n を作る：しかしながら、いずれかのシードｖ（ｈ，０）ただし（ｈ＜ｍ）、またはｖ（ｈ，１）ただし（ｈ＜ｎ）は、この受け側にとっては可能性としては知られることがなく、知るにはブラインドとしたシードについての膨大なサーチを必要とし、それが送り側が啓示（レビール）したことに“先行する”ものとする。したがって、k_n ないしk_m の範囲（レンジ）外のキーはこの受け側では現実には計算できないものである。
４．いずれもの他の受け側には完全に異なる範囲のＡＤＵへのアクセスが与えられるようにでき、そのためにはその範囲の境界で関連のシードを送ることがされる。第一のチェーンからは“開始”シードが、第二のチェーンからは“終末”のシードが送られる。

【００４５】
図７では、暗い灰色の背景をもつシードの範囲が第一の記述した受け側からブラインドされたものを表わしている。これが暗い灰色の背景をもつキーがこの受け側からもブラインドされることに通じている。

【００４６】
したがって、各受け側はキーの連続している範囲へのアクセスを受け側につきまたセッションにつき二つだけのシードを送ることによってアクセスが与えられるようにできる。不運にも、この構成は、各受け側が一つの送り側シーケンス内で啓示されたキーの一つの範囲をもっているものだけに限定できない限り、制限された使用となる。もし受け側が初期の範囲へと次に他の後の範囲へのアクセスを許されていると（例えばk_0 ないしk_1 であれば、次にk_{G-1} ないしk_G ）、受け側はそこでこの二つ（k_0 ないしk_G ）の間のすべての値を計算できる。これはシードｖ（０，０），ｖ（Ｇ－１，１），ｖ（Ｇ－１，０）及びｖ（Ｇ，１）が啓示されるようにならなければなるが、ｖ（０，０）とｖ（Ｇ，１）だけが全体のシーケンスを啓示することになることが理由とされる。

【００４７】
一巡してみると、この限定は規則正しく第二のチェーンを新しいシード値で（すなわちＨを低く保って）再スタートさせていて、互のＨのＡＤＵ内で一方の受け側について二つのアクセスを許さないようにしている。しかしながら、このことはキーマネージャのところで顧客状態を保持することを必要とする。ニッチな（niche）応用があってもよく、そこではこの機構は適切なものとされ、その例は顧客が加入を拡張することができるだけであって、退去して、再始動しなくてもよいといった商用モデルがそれにあたる。このような場合にはこれは極めて効果的な機構である。

【００４８】
二巡目のものは、この限定は二つの離れた（接合していない）チェーンが、二つの最少と言えるほど短いチェーンの間のギャップのための余地が存在することを条件としてのみ可能とされることに注目することである。言い換えるとＨ＜４のチェーンが常に安全となることである。このような短いチェーンは多く使用されるとは見えないが、後に、この特徴を使用して、短いＢＨＣフラグメント（部分）から混成（ハイブリッド）構成を築くためにこの特徴を使用することになる。

【００４９】
二進ハッシュトリー（ＢＨＴ）
二進ハッシュトリーは二つのブラインド機能、b_0（ ）とb_1（ ）とを必要とすることがよく知られている。ここではこれらを“左”と“右”とのブラインド機能と呼ぶことにする。一般にこれらは単一のブラインド機能ｂ（ ）から、二つの単純な１対１機能、r_0（ ）とr_1（ ）と、の一方をブラインド機能の前に適用することによって構築できる。図８に示すところである。

【００５０】
したがって：b_0（ｓ）＝ｂ（r_0（ｓ））；b_1（ｓ）＝ｂ（r_1（ｓ））　例えば、第一のよく知られているブラインド機能は１ビット左円形シフトの後にＭＤ５ハッシュを行うものであってよく、その間に第二のブラインド機能は１ビット右円形シフトし、その後にＭＤ５ハッシュが続くものとすることができる。他の代るものは一つのブラインド機能で１とのＸＯＲを伴うものかよく知られているワードでの連結を伴うものが先行している。二つの機能として極小でしかもプロセッサ資源の等量を消費するものを選ぶのが好都合と思われ、その理由はこれがあらゆる場合にロードとバランスをとりチャンネルを切り換えるサスセティビリティ（感受性）に制限を与えるからである（この感受性はプロセッサロードのレベルが実行されている機能の選択を啓示することを条件としているが、そうでないときに出現するものである）。代って、効率の点についてはハッシュ機能の二つの変形が使用される。例えば二つの異なる初期化ベクトルを備えたＭＤ５である。しかしながら、試行及び試験アルゴリズム（tried-and-tested algorithms）でタンパーする（変更を加える）とするのは誤った勧告と思われる。

【００５１】
　このキーシーケンスは次のように構築される：
１．送り側は初期シード値ｓ（０，０）を無作為にランダムに生成する。ここでもまた具体例として、その値を１２８ビット幅とする。
２．送り側は、要求された最大トリー深さ（奥行き）Ｄについて決定をし、これが新しい初期シードが必要とされる前に最大キーシーケンス長$N_0 = 2^D$に通じることとなる。
３．送り側は二つの“左”と“右”との第一のレベルの中間シード値を生成し、それぞれ“左”と“右”とのブラインド機能を初期シードに適用する：
　ｓ（１，０）＝b_0（ｓ（０，０））；ｓ（１，１）＝b_1（ｓ（０，０））
送り側は四つの第二のレベルの中間シード値を生成する：
　ｓ（２，０）＝b_0（ｓ（１，０））；ｓ（２，１）＝b_1（ｓ（１，０））；
　ｓ（２，２）＝b_0（ｓ（１，１））；ｓ（２，３）＝b_1（ｓ（１，１））,
等々であり、Ｄレベルの深さまで中間シード値の二値トリーを作る。正式には、もし$s_{d,i}$が中間シードであって初期シード$s_{0,0}$の下ｄレベルであるとすると、
$$s_{d,i} = b_p(s_{(d-1),i/2}) \qquad (4.2.1)$$
であり、ここでｉが偶数であればｐ＝０、奇数であればｐ＝１である。
４．キーシーケンスが次に前と同じようにトリーの葉をまたいだシード値もしくはそれらから端を切り落とした（トランケートした）誘導されたものから構築される：
　すなわち、もしＤ＝５，k_0＝ｓ（５，０）であれば、k_1＝ｓ（５，１）；…k_{31}＝ｓ（５，３１）である：
　正式には　$k_i = s_{D,i}$　　(4.2.2)
である。
５．送り側はストリームのマルチキャストを開始し、ADU_0をk_0で、ADU_1をk_1でというように暗号化するが、少くともＡＤＵシーケンス番号は平文のまま残す。
６．もし送り側がキー管理を委ねていれば、私的に（公開せずに）キーマネージャと初期シードを通信する必要がある。新しい初期シードが生成できてキーマネージャに向けて、先に計算されたキーで暗号化されたストリーミングデータと並列に通信することができる。

【００５２】
受け側はシーケンスの一部を次のように再構築する：
１．受け側がADU_mからADU_nへのアクセスを許可されるときは、送り側（あるいはキーマネージャ）はその受け側に向けてシードの組を（例えばＳＳＬ）を用いてユニキャストする。この組は中間シードで構成されていて、このシードは必要とされる範囲のキーの計算をこの範囲外のいずれのキーの計算を可能とすることなく、可能とするトリーの根（ルート）に最も近いものである：
こういったものは最小及び最大のシードのインデックスｉを試験することによって識別さ

れ、その際には、偶数のインデックスが常に“左”の子供であり、また奇数のインデックスが常に“右”の子供であるという事実が使用される。試験はトリーの各レイヤで実行され、葉から始まって上方へ向って動作が進む。“右”の最小もしくは“左”の最大はレヘルを上へ移動する前に啓示することを常に必要とする。もしあるシードが啓示されると、インデックスは一シードだけ内側にシフトされ、それによって、レイヤを上へ移動する前に最小と最大とが常にそれぞれ偶数と奇数となる。レイヤを上へ移動するためには、最小と最大のインデックスは半分とされて、必要であれば丸められる。これは両者間の差が２だけ予測可能に減ることを確かにしている。奇数／偶数試験は新しいインデックスについて繰返されて、前のように“右”最小もしくは“左”最大を啓示する。このプロセスは最小と最大が交わるか一致するまで続く。これらが交わることができるのは一方か両方かが内側へシフトされた後である。両方が上側へシフトされた後に一致することができ、この場合は、一致したところでのシードはプロセスが終る前に啓示される必要がある。このプロセスはもっと正式には付録ＡにＣのようなコードで記述されている。

２．明らかに、各受け側は与えられた各シードがトリーのどこにあるかを知る必要がある。このシードとそれらのインデックスとはそれらが啓示されるときに明らかに対とすることができる。代って、必要とされる帯域幅を減らすために、プロトコルは順序を特定してよく、この順序でシードが送られて、それにより各インデックスが最小と最大のインデックスとシードの順序とから黙示的に計算できる。これが可能なのは、キーのいずれか一つの範囲の再作成が許されているのはシードの唯一の一番小さな組しかないことによる：各受け側はそこでこういった中間シード上に同じ対のブラインド機能を繰返すことができる。これは送り側がk_mからk_nまでのキーのシーケンスを再作成するためにしたのと同じである（式4.2.1及び4.2.2）。

３．いずれか他の受け側は中間シードの異なる組を送られることによってＡＤＵの完全に異なる範囲へのアクセスを与えられるようにできる。

【００５３】

キーシーケンスでＤ＝４についての作り方がグラフとして図９に示されている。

【００５４】

例として、一つの受け側がk_3からk_9までのキーシーケンスを再作成することができるようにする関連の中間シードに丸印を付けてある。シードとキーとでこの受け側からブラインドされて残っているものは灰色の背景上に示されている。無論、Ｄの値で４よりも大きいものは実用上一般的なものである。

【００５５】

各レイヤは、ユニークにレイヤを識別する限りは任意のｄの値を指定されることに注意したい。ｄとかＤとかの実際の値に頼っているものは何もないので、送り側にとってはどのくらい遠くまでトリーが上側に延びているかを啓示する必要はなく、したがってセキュリテイ（安全性）が改善されている。

【００５６】

ときにはセッションが開始時に未知の継続期間をもつことになる。明らかにＤの選択はある一つの出発点からのキーシーケンスの最大長を制限する。一番簡単な作業の一巡は新しい初期シードを生成して、必要ならば古いものに沿って新しい二進ハッシュトリーを開始するだけのことである。もしＤが送り側と受け側とによって知られていれば、キーの範囲で最大キーインデックス2^Dをオーバーフローするものは、すぐにすべての当事者にとって明らかなものとなる。このような場合には、“トリーのＩＤ（識別子）”を各新しいトリーに割当てて、これを各トリーについてのシードと一緒に特定することが賢いものとされる。

【００５７】

この上の方の限界を回避する別なやり方は、Ｄを定数ではなく変数とすることであり、例えば$D = D_0 + f(i)$とする。図１０が示しているのはこのような連続するＢＨＴであり、ここで$D_0 = 4$であり、またＤはＭのキー毎に１だけ上昇する。この例ではＭは固定値７をとる。しかし、この複雑を増すには僅かなポイントが存在しており、トリーの別な

枝に対して共通なシードだけがトリーの遠方の右側の枝、$s_{d,2}$ ｄ、に沿っているものとなっていることである。こういったもののうちのいずれかが啓示されていたとすると、全体の将来のトリーは啓示されたことになる。したがって、この“改良”は受け側へキーの任意の範囲を啓示するときには効率を高めるためには決して使用できないし、節約されるものの全ては、送り側が非常にまれに些細なメッセージ内で新しい初期シードをキーマネージャに向けて送るということである。これとは対照的に、膨大なサーチ量が値うちのあるものとされる“無限大”の値のシードの組を作ることが理由となってそれがセキュリテイの弱点を導入する。他方、新しい初期シードを規則正しく生成しなければならないことは、第一の作業一巡でのように、攻撃のためのＢＨＴの攻撃の受け易さについてシーリング（上限）を設定することになる。

【００５８】

二進ハッシュチェーントリーハイブリット（ＢＨＣ－Ｔ）

この構成はハイブリッド（混成）とよばれているが、その理由は二進ハッシュトリー（ＢＨＴ）が双方向性ハッシュチェーン（ＢＨＣｓ）の一部でちょうど２シード長であるものから作られていることによる。理解のためだけの目的で根から葉への方向でトリーを構築する例で始めることとし、これは図１１に示したようにＢＨＣの部分を構築するための例である。これは理解を容易にすることを目的としている。後にトリーを構築するための最善方法は根からではなく側部からであることを勧めることになる。

１．ランダムに生成された二つの初期シード値；ｓ（０，０）とｓ（０，１）とがあると仮定する。ここでもまた、具体的な例として、その値が１２８ビット幅であるとする。

２．同じ構築機能を各シードに適用して二つのブラインドしたシードｖ（１，０）とｖ（１，１）とを作る。

３．子供のシードｓ（１，１）を作るために、第一のシードｓ（０，０）をブラインドした第二のシードｖ（１，１）と組合せる：

子供のシードｓ（１，２）を作るために、第二のシードｓ（０，１）をブラインドした第一のシードｖ（１，０）と組合せる。

４．第三の初期シードｓ（０，２）をランダムに生成して、それをブラインドしてｖ（１，２）を作るとすると、第二と第三の初期シードと、それらの相手側のブラインドした値を同じやり方で組合せて、二つの別な子供シードｓ（１，３）とｓ（１，４）とを作る。これが意味するのはどの親シードも４人の子供を作るということであり、二つは（近親相姦的に）片親共通（sibling）で組合されるのが一方の側で、また他の二つは他方の側に半分片親共通（half-sibling）で組合されている場合である。その結果、この構成はもし新しい子供のシードがブラインドされて、それらの親があったように組合されるとすると二進のトリーを作ることになり、その理由はシードの数が各世代で倍増していることによる。しかし、このトリーはシードの最上行の中央（２以上の初期シードがこの行に沿って作られていると仮定している）の下でのみ枝分れをする。トリーの縁は最上から作られるとすると内側に“しぼむ”ことになる（後述参照）：

正式には、

$$s(d,i) = c(s(d-1), i/2), v(2d-1, i/2+1) \quad i\text{奇数}$$
$$= c(s(d-1), i/2), v(2d-1, i/2-1) \quad i\text{偶数} \qquad (4.3.1)$$

ただし、

$v(h,j) = b(s((h-1)/2, j))$。

【００５９】

図１１ａはＢＨＣ－Ｔハイブリッドの親シードの二対、＜ｓ（０，０），ｓ（０，１）と＜ｓ（０，１），ｓ（０，２）を示す。輪は各対にとって共通な親シードを識別しており、外側の輪の中にある外側の数値はこの図の縁からはみ出しているが、それはここでは中心の親シードｓ（０，１）だけの子孫について注目していることによる。図１１ｂは同じ三人の親を示し、同じ四人の子供を作っているが、視野からはブラインドしたシードは隠れていて、その理由はそれらが決して通信をしていないことにあり、またどのように二進のトリーが形成されるかをより良く示すためでもある（注：親子のことであるので単位と

して「人」を用いることにした）。輪を付けた親シードで下側の図にあるものは、同じ三つのリングされたシードで上側の図に示されたものを表わしている。二つの破線矢印でこのシーケンスを右へ続けているものは、どのように親シードｓ（０，２）が別の二人の子供を作ることになるかを右に別の親があることを条件として示している。矢印の各対を接合している破線はこの線の上の両方の親が組合されてその下の両方の子供を作るという事実を表わしている。この構成を簡単にした形成を用いて後の図で表わすことにする。

【００６０】

図１２はハイブリッドトリーの例の一部を示す。二進ハッシュトリーの場合のように、継続するＡＤＵを暗号化するのに使用されるキーはトリーの葉におけるシードのシーケンスであるかそれらからの頭を切り落した誘導されたものかである。この図は特定の受け側に向けた輪で結んだシートを啓示することによって、キーk_3ないしk_9の範囲の例がどのように啓示されるかを示している。

【００６１】

ここでルート（根）ではなくサイド（側部）からトリーをどのように構築するかを説明するために、この構成での別なトゥイスト（取り組み方）へ移ることとする。先に、ＸＯＲ機能が選ばれたことを示し、その理由として、もし二つのオペランドのＸＯＲが第三の値を作るとすると、これら三つの値のうちのいずれか二つがＸＯＲされて第三のものを作ることができるとした。これが図１３に示されていて、すべてのシードの値が図１１と同じにな成っている。もしｓ（０，１）が最初に未知であるが、ｓ（０，０）とｓ（１，２）とが既知であれば、ｓ（０，１）と次にｓ（１，１）とがこの“ツイスト”性質によつて求められる：

ｓ（０，１）＝ｃ（ｓ（１，２），ｂ（ｓ（０，０））であり、そのときは、

ｓ（１，１）＝ｃ（ｓ（０，０），ｂ（ｓ（０，１））である。

【００６２】

図１４は受け側がどのようにＢＨＣ－Ｔハイブリッド構成を“側部（サイド）”から作ることができるかを示している。シードを作成の順序は番号を付けた円で示されている。いずれかの順序で作ることができるシードが同じ番号の後に区別用の文字が来るものをすべて割当てられている。陰影のある円で輪をつけたノードの次に来るものはランダムに生成されるべきシードを表わしている。これらを一次（プライマリイ）シードと呼ぶ。これらが次の輪をつけたノードまでの後続の全中間シードの値を固定している。

１．送り側はシード０の１２８ビット値をランダムに生成する。

２．シード１と２とが次に生成される。これらは四つのシードの箱の対角の隅を形成し、したがって対向する角の値３と４とを“ツイスト”アルゴリズムによって設定する：

正式には、

ｓ(d-1，i/2)＝ｃ（ｓ(d,i)，ｖ(2d-1，i/2+1)）　　ｉ奇数

＝ｃ（ｓ(d,i)，ｖ(2d-1，i/2-1)）　　ｉ偶数　　（4.3.2）

ただし、

ｖ(h,j)＝ｂ（ｓ((h-1)/2，ｊ)）である：

もし根（ルート）シードに対してｄ＝０であると、ｄは葉から根へ向う方向で漸進的に負となることに注意したい。

３．シード５は次に生成されなければならず、対角隅の別の対を２で形成する。

４．これが対向する角、シート６と７を式（4.3.2）で啓示する。

５．シード７と２とが次に４の別の箱の上側の隅を形成し、式（4.3.1）によりシード８ａと８ｂとを設定する。

６．このパターンは同じ様な形式で、シード９がランダムに生成された後に続く。この構成の利点は、トリーが限界なしに生長できるということであり、前もって何らかの限界を決める必要がない。

７．送り側はこのストリームのマルチキャストを開始し、ADU_0をk_0で、ADU_1をk_1でといったように暗号化するが、少くともＡＤＵシーケンス番号は平文として残しておく：

すなわち、$k_i = s(D, i)$であり、ここでD＝０である　　（4.3.3）。
８．もし送り側がキー管理を委ねていれば、キーマネージャに向けて一次シードを私的に（非公開で）通信しなければキーならない。新しい一次シードが生成できて、キーマネージャに向けて、先に計算したキーで暗号化したストリーミングデータと並列に通信される。

【００６３】
受信側はシーケンスの一部を次のように再構築する：
１．受け側がADU_mからADU_nへのアクセスを許されるときには、送り側（もしくはキーマネージャ）はその受信側にシードの組をユニキャストする。この組はトリー内の中間シードの最も小さい組で成り、必要とされているキーの範囲の計算を可能とするものである：
これらは最小と最大のシードのインデックスｉをＢＨＴに対して同じようであるがミラーとした（鏡側を作る）やり方で試験することにより識別される。“左”の最小か“右”の最大かがレベルを上に移す前に常に啓示されることを要する。もしシードが啓示されると、インデックスが内側に一シードだけシフトされて、それによりレイヤを上へ移動する前に、最小と最大とが常に奇数と偶数とにそれぞれなる。レイヤを上へ移すために、最小と最大のインデックスは半分とされ、必要があれば丸められる。これが両者間の差を１だけ予測可能に減らす。奇数／偶数試験が新しいインデックスについて繰返される。このプロセスは最小と最大とが２もしくは３離れているようになるまで続く。もし二つが２だけ離れていると、二つは両者間でそのシードと一緒に啓示される。３だけ離れていると、両者の間で両方のシードと一緒に啓示されるがその条件は最小が偶数である場合に限られる。もし奇数であるともう一つレイヤを上へ移動する価値があり、それによって、何も啓示されず、もう一巡が許される試験が開始された後に、例外的な初期条件が試験されるが、それは要求された範囲がすでに二つ幅よりも小さくなっている場合である。このプロセスはもっと正式には付録ＢにＣに似たコードで記述されている。
２．明らかに、各受け側はそれが与えられる各シードがトリーのどこにあるかを知る必要がある。これらのシードとそのインデックスとは啓示されるときには明白に対とされることができる。これに代って、必要とされる帯域幅を減らすためには、プロトコルは順序を特定してよく、この順序でシードが送られ、それにより各インデックスが最小と最大のインデックスとシードの順序とから黙示的に計算できるものとなる。例えば、付録Ｂにあるアルゴリズムは、同じキーの範囲について同じ順序で同じシードをいつも啓示する。
３．各受け側はそこでこれらの中間シードについてブラインドと組合せ機能の同じ対を繰返すことができ、これは送り側がキーk_mないしk_nのシーケンスを再作成するためにしたのと同じである（式4.3.1, 4.3.2, 及び4.3.3）。
４．他のいずれかの受け側は中間シードの異なる組を送られることによってＡＤＵの完全に異なる範囲に向けたアクセスを与えられるようにできる。

【００６４】
ＢＨＣ－Ｔは側部から構築されるようにできるので、未知の継続期間のセッションにとっては理想的である。続いているランタル生成を新しい中間ルートシードについてすることは攻撃に対する攻撃の受け易さを制限するが、シーケンスの連続計算を許す。攻撃の受け易さをさらに制限するためには、送り側は将来のシードの生成を遅らせることができて、いずれもの受け側もシーケンス内のある将来点を越えたところでのキーを計算するための能力を否定するようにする。これが、シード空間についての暴力的サーチについて利用できる時間を制限することになる。それにも拘らず、側部からトリーを構築することは各新しいルートシードに依存するキーの数（したがってシードに対する攻撃の値）が指数関数的に成長する原因となっている。

【００６５】
ルートシードの値は規則正しく葉レベルとなるように定義されたレベルを歩進させ（第一のものを除く）、Ｍ個のキーの各シーケンスの後のルートに近い１つのレイヤに移動することによって境界を画成することができる。

【００６６】
正式には、このことは式（4.3.3）が下記により置換されることを要する。
【００６７】
$k_i = s(-i/M, i)$　　i＜Mに対して、
$k_i = s(1-i/M, i)$　　i＝Mに対して　　（4.3.4）。
【００６８】
これが図１５にM＝８について示されている。無論、実際にはMはもっと大きく、合理的な長さの受け側セッションがトリーの上左手の枝にあたることがないように効率的に記述されることができることを確かなものとしている。
【００６９】
先に、ＢＨＣはH＜４のときに限り真性に安全であることに注目した。ＢＨＣ部分でこのチェーントリーハイブリッドで使用されるものはH＝２であり、これがハイブリッド機構の安全を確かなものとしている。このことはまた、二進チェーントリーハイブリッドが長さ３（H＝３）のチェーン部分から安全性に妥協を生じさせずに構築できることを示唆している、この場合に、各親シードは六人の子供を作り、その条件は片親共通的もしくは半片親共通的に対を作るときとしており、それによって各レベルでのトリーの幅で三通りの生長を与えている（三重のトリー：ＢＨＣ３－Ｔ）。この構造が図２４に示されているが、完全な解析は将来の作業のために残されている。若干複雑となるが、ＢＨＣ－Ｔよりもより効率的となる潜在性をもっている。
【００７０】
二進ハッシュトリーII（ＢＨＴ２）
ここで別な二進トリー応用構造であってＢＨＴとＢＨＣ－Ｔとのやり方を組合せて、暴力的攻撃（ブルートフォースアタック）に対抗するセキュリテイを大いに強固にするようにしたものを提供する。シード$s_{d,i}$について同じ記号法を用いるが、ｄについてのもと（原点）はＢＨＴについてのようにルートにあり、その値は葉に近づくとともに上昇する。トリーの一つの要素は図１６に示されている。ここでは二つのブラインド機能をこの構成で使用し、b_0（　）とb_1（　）とであり、これをそれぞれ“左”と“右”と呼ぶことにするのはＢＨＴの場合と同じである。
１．ここで二つのランダムに生成された初期シード値ｓ（０，０）とｓ（０，１）とがあると仮定する。ここでもまた、具体例としてその値が１２８ビット幅をとるとする。
２．送り側は必要とされる最大トリー深さ（奥行き）Ｄを決める。２つのブラインドとした値を各初期シードから作り、各ブライインド機能について一つを作る：
ｖ（１，０）＝b_0（ｓ（０，０））；ｖ（１，１）＝b_1（ｓ（０，０））；
ｖ（１，２）＝b_0（ｓ（０，１））；ｖ（１，３）＝b_1（ｓ（０，１））。
３．子供シードｓ（１，１）を作るために、二つの左のブラインドとしたシードｖ（１，０）とｖ（１，２）とを作る：
子供のシードｓ（１，２）を作るために二つの右のブラインドとしたシードｖ（１，１）とｖ（１，３）とを作る。
４．ここで第三のシードｓ（０，２）をランダムに生成するとすると、第二と第三との初期シードを同じように組合せて、二つの別な子供のシードｓ（１，３）とｓ（１，４）とを作ることができる。ＢＨＣ－Ｔハイブリッドと同じように、これはすべての親シードが二人の子供を作り、二進のトリーを構築できるようにするが、縁では内側に“しぼんだもの”となる。事実、もしレイヤｄがn_dシードを含んでいると、$n_{(d+1)} = 2n_d - 2$である。二以上の初期シードが使用される限りは、トリーは二進トリーに向う傾向がある：
正式には、
$s(d,i) = c(v(2d-1, i/2), v(2d-1, i/2+1))$　　i奇数
$= c(v(2d-1, i/2), v(2d-1, i/2-1))$　　i偶数　　（4.4.1）
ただし、
$v(h,j) = b(s((h-1)/2, j))$
である。

５．キーシーケンスはそこでトリーの葉をまたいでシード値から構築される：
正式には　$k_i = s_{D,i}$　　（4.4.2）
である。
６．送り側はこのストリームのマルチキャストを開始し、ADU_0 を k_0 で、ADU_1 を k_1 でというように暗号化するが、少くともＡＤＵ番号は平文で残す。
【００７１】
図１６ａはＢＨＴ２の二つの新シード対＜ｓ（０，０），ｓ（０，１）と＜ｓ（０，１），ｓ（０，２）とを示している。輪は親シードを識別し、これは図のａとｂとについて共通であり、ＢＨＣ－Ｔハイブリッドを示すために使用したのと全く同じやり方である。前と同じように、図１６ｂはＢＨＴ２で構築したシードのトリーがどのように表わされるかを示していて、はっきりと見えるようにするために中間のブラインドした値は隠されている。一旦これらの中部値が隠されると、結果のＢＨＴ２は図１１ｂのＢＨＣ－Ｔハイブリッドと同一に見える。
【００７２】
どのシードがキーの範囲を啓示するために啓示されるべきかを計算するアルゴリズムもまた付録ＢにあるＢＨＣ－Ｔハイブリッドについてのものと同一であり、したがって図１２で輪をつけたシードは依然として特定の受信側に対して k_3 ないし k_9 を啓示している。
【００７３】
三つの初期シードでレイヤ０にあるものから深さＤまで構築したＢＨＴ２の葉をまたいだキーの最大数は $2^D + 2$ である。もし連続したトリーが必要であれば、キーは中間シードのレイヤを段々に下るように定義できて、そこにまたがるレベルに留まるのではないことは図１０に示した連続したＢＨＴと似ている。
【００７４】
四つのブラインドした値の式（4.4.1）での二つ組合せを用いてのみの二進トリーの構築方法を示してきた。一時に四つの値の二つをとると六つの可能な組合せがある。
【００７５】
ｃ１＝ｃ（ｖ（１，０），ｖ（１，１））
ｃ２＝ｃ（ｖ（１，２），ｖ（１，３））
ｃ３＝ｃ（ｖ（１，０），ｖ（１，２））
ｃ４＝ｃ（ｖ（１，１），ｖ（１，３））
ｃ５＝ｃ（ｖ（１，０），ｖ（１，３））
ｃ６＝ｃ（ｖ（１，１），ｖ（１，２））
ｃ１とｃ２とはそれぞれ一人の親シードにだけ依存している。したがって、親を単独で啓示することは、子供を啓示し、両方の使用を規制している。さらに、ｃ６＝ｃ（ｃ３，ｃ４，ｃ５）とｃ５＝ｃ（ｃ３，ｃ４，ｃ６）などとなるので、したがって、こういった組合せのいずれか三つを啓示することは第四のものを黙示的に啓示している。それにも拘らず、これらの組合せのいずれか三つがＢＨＴ２で使用されたたった二つよりも使用されることができる。結果として生じた三次のトリー（ＢＨＴ３）の解析は将来の作業として残されている。
【００７６】
共通モデル
四つのキー構成を呈示した上で、共通モデルを呈示することとし、このモデルはこれらの機構も、他の同様な機構も同じ条項（用語）で記述することができるようにする。
【００７７】
二つの座標平面を定義する。
・“ブラインド”平面は離散的な値ｖを有し、それが座標（ｈ，ｊ）にあって、一般に一方のｈ座標での値がブラインドされて値をｈ＋１に作り、特定のマッピングがこの機構に依存している。
・“組合せ”平面は離散的な値ｓを有し、それが座標（ｄ，ｉ）にあって、ブラインド平面からの値を組合せた結果となっており、この組合せもまたこの機構に依存するようにし

ている。

【００７８】

各構成が初等数学的“分子（モレキュール）”からブラインド平面内の内で構築される。図２０～２４はこういった分子が太い黒い矢印の集合として示しており、この矢印はｈ＝０軸から始まって、ｖの一つの値から次へのマッピングをするブラインド機能を表わしている。どのように構成がｊ軸の方向に生成するかを示すために、太いがしかし非常に薄い灰色の矢印は次の分子を完成させる隣の値のブラインドを表わしている。分子は三つの定数により定義される。

・Ｈ，これはブラインド平面のｈ軸に沿った一つの分子の高さである。

・Ｐ，これは一つの分子内で使用されたブラインド機能の数である。

・Ｑ，これは組合せ平面内で各値を作るためにブラインド平面内の各分子から組合される値の数である。

【００７９】

ブラインド平面内の一つの分子の初期値ｖは組合せ平面内の前の値から直接にマップする（これが図２０～２４に鎖線で示されている）：

もし、ｈ mod Ｈ＝０であれば、ｖ（ｈ，ｊ）＝ｓ（ｈ／Ｈ，ｊ）　　（4.5.1）

である。ブラインド平面分子の後続の値は前の値から（太い矢印で示すように）ブラインドされる：

もし、ｈ mod Ｈ＝０であれば、

$v(h, j) = b_p(v((h-1), j/p))$　　（4.5.2）

であり、ここでｐ＝ｊ mod ｐである。

【００８０】

結果としてのブラインド平面分子内の最終値はそこで組合されて、組合せ平面内の次の値を作るために組合される（細い線で図示されている）：

$s(d,i) = c(v(h_0, j_0), \cdots v(h_q, j_q), \cdots v(h_{(Q-1)}, j_{(Q-1)}))$
（4.5.3）

ここでh_qとj_qとはパラメータｑの関数として各構成に対し定義される。

【００８１】

したがって、ｄはブラインド平面内でｈ軸に沿ってすべてのＨについて組合せ平面内でｄが１だけ歩進する。

【００８２】

表１はＨ，Ｐ及びＱの値を与え、また、h_qについての式は各構成を定義するj_qである。この表はまたこの共通モデルを用いる各構成を例示する数値を参照している。

【００８３】

【表１】

表１．各キーシーケンス構成を定義する共通モデルの係数

	ＢＨＴ２	ＢＨＴ	ＢＨＣ	ＢＨＣ－Ｔ	ＢＨＣ３－Ｔ
図	２０	２１	２２	２３	２４
H	2	2	H	2	3
P	2	2	1	1	1
Q	2	1	2	2	2
h_q	Ｈｄ－１		Ｈ(d-1)＋ｑ(H-1)＋(1-2q)(i mod H)		
j_q	ｉ－１＋２ｑ	ｉ	ｉ／Ｈ＋ｑ		

【００８４】
すべての場合、連続する構成が望ましい場合を除けば、このシーケンスから構成されたキーは次式により定義される：
k_i ＝ｓ（Ｄ，ｉ）　　（4.5.4）
ここで、Ｄ＝log（N_0）であり、N_0 は必要とされるキーの最大数である：
付随的に、一方向性機能トリー（one-way function tree OFT）［McGrew 98］が｛…？｝を設定することから生じている（訳者注：この２行はＰＣＴ原文の記載不備と思われる）。
【００８５】
処理に対する記憶のトレードオフ（兼合い）
すべてのＭＡＲＫＳ構成では、少数のシードが使用されて、暗号化前の送り側と、解号化前の受け側の両方でもっと大きな数のキーを生成するために使用される。いずれの場合も、キーシーケンスについての記憶能力の制限があってもよく、その記憶能力はシードよりも指数関数的に増大する記憶を必要とする。このような場合に、第一の僅かなキーが計算され、その際に他のものが計算できるようにするシードを記憶するようにしている。一般に、記憶もしくは処理時間が最論の供給であるかどうかに依存して、記憶がセーブできるか、あるいは同じ計算の繰返しを避けることができるかのいずれかとなる。
【００８６】
ＢＨＣについてみると、第一のキーが計算できる以前に、全体の逆のチェーンがトラバースされなければならない。しかし、すべての値が記憶される必要はない。道中葉での値、３／４点での値等は、記憶されてよく、残りは無視される。シーケンスがこの逆チェーンを浸食（eat back、ここではトラバースの意）していくと、次の値が常に再計算され、それには前に記憶した値からのハッシュチェーンを再度走り、必要とされるところによりその進路でより多くの値を記憶することが行なわれる。
【００８７】
すべてのトリー構成についてみると、いずれかの中間シードで第一のキーに対するトリーの枝を下って行くものがセッションが開始できる前に計算される必要があるが、ここでもまた、それらのすべてが記憶される必要はない。葉に最も近いものが記憶（キャッシュ）される必要があり、その理由は、そういったものが次のいくつかのキーを計算するためにすぐに必要とされることになるからである。根により近い中間シードが必要とされるときは、そういったものはキーマネージャによってもともと送られたシードが無視されないでいる限り再計算できる。
【００８８】
効率

すでに述べたように、Ｈ３を伴うＢＨＣは極めて効率的であるがセキリティ（安全性）が乏しい。したがって、ここでは発明者らが完全に解析した二進トリー応用の構成について議論を限定することとする。表２は、安全なマルチキャストセッションについてＢＨＴ，ＢＨＣ－Ｔ，及びＢＨＴ２の各種パラメータを示す。

【００８９】

ここでＲ，Ｓ，及びＫＭは受け側、送り側、及びキーマネージャをそれぞれ表わし前節で定義したものであり；Ｎ（＝ｎ－ｍ＋１）は受け側が必要とするキーのレンジの長さであり、キー空間内にランダムに位置しており；w_s はシードのサイズ（一般に１２８ｂ）であり、；w_h はプロトコルヘッダオーバーベッドのサイズであり；t_s はシードをブラインドするためのプロセッサの時間（それに比較的無視できる循環シフト及び／又は組合せ操作１を加える）である。

【００９０】

【表２】

表２．安全なマルチキャストセッションについてのＢＨＴ，ＢＨＣ－Ｔ及びＢＨＴ２の各種パラメータ

			ＢＨＴ	ＢＨＣ－Ｔ	ＢＨＴ２
Per R	(unicast message size)/w_s-w_h or (min storage)/w_s	min	1	3	3
		max	2(log(N+2)-1)	2 logN	2 logN
		mean	O(log(N)-1)	O(log(N))	O(log(N))
Per R	(processing latency)/t_s	min	0	0	0
		max	log(N)	2(log(N)-1)	4(log(N)-1)
		mean	O(log(N)/2)	O(log(N)-1)	O(2(log(N)-1))
Per R or S	(processing per key)/t_s	min	1	2	4
		max	log(N)	2(log(N)-1)	4(log(N)-1)
		mean	2	4	8
Per S or KM	(min storage)/w_s		1	3	3
Per S	(min random bits)/w_s				

【００９１】

各受け側のセッション設定についてのユニキャストメッセージサイズは各受け側が必要としている最少記憶量と等しくなることが示されている。このことが言えるのは、受け側が上記のように処理のための記憶をトレードオフするように選ぶことを条件としている場合に限られる。同じトレードオフが最小の送り側メモリ行についても使用されている。処理のための遅れは一つの受け側がそのセッションについてユニキャスト設定メッセージを受領した後に到来するデータを解号化する用意ができるまでに必要とされる時間である。他のメンバーが参加したり、退去するときには遅れというコストは存在せず、それは計画されていない立退き（eviction）について用命に応ずる（ケイターする）スキームの中にあることによる。キーについての処理のための数字はキーへの継続的なアクセスを仮定している。この場合に、一番効果的な値でいずれものセッションの間に記憶されるものは（トリーの構造を可能とするために啓示された最少組を除けば）根から現在使用中のキーまで

の枝の上にあるものとなる。キーについての平均の処理はそこで全体の記の中でのハッシュ操作の数を葉におけるキーの数で割ったものとなる。送り側だけが（あるいは複数送り側があるときにはグループの制御側が）初期シードについてランダムなビットを生成することが求められている。必要とされるビットの数はこういった初期シードの最小送り側記憶に明らかに等しいものとなる。

【００９２】

グループ会員（メンバーシップ）のサイズ（大きさ）に依存するパラメータだけが受け側についてのパラメータであることが認められる。こういった二つ（記憶と処理の遅れ）のコストがグループ会員間に分散され、それによって受け側では一定のものとなる。ユニキャストメッセージサイズだけがキーマネージャでコストを生じ、これがグループ会員サイズと線形に上昇して行くが、コストは受け側セッションについて一度だけ負担される。たしかに受け側にとってのコストはそれ自体がグループサイズに依存しており、その理由はすべてのスキームが計画されていない立退きを認めていることによる。したがって、提供される構成のすべては高度にスケーラブルなものとなっている。

【００９３】

このスキームは相互に比較してみると、恐らくは驚くべきこととして、混成の（ハイブリッドの）ＢＨＣ－ＴとＢＨＴ２とはメッセージ送りという項目ではＢＨＴと非常に近い程効率的である。これらは共に受け側セッション設定メッセージについて平均して一つだけ余計なシードを必要とするだけである。もしＮが大きいとすると、このことは受け側セッションについて必要とされるキーの数に比較して重要なこととはならない。平均して、ＢＨＣ－Ｔは処理２倍を、ＢＨＴ２は４倍をＢＨＴに対して必要としている。しかしながら、安全の改良はコストと同様に価値があることを知ることになる。

【００９４】

ＢＨＴ

ＢＨＴについてみると、トリー内の各シードは可能性としてその子供の２倍の価値がある。したがって、そのトリー内で現在一番高いとして知られているシード値についてブラインドする正しい値を求めてシード空間を手間ひまをかけて探査するための誘因（インセンティブ）が存在している。ＭＤ５ハッシュについては、平均して2^{127}回のＭＤ５操作が含まれることになる。ある値が正しくないが、既知の値と衝突する値に対してブラインドするとして見つかる可能性がある（一般的には、ＭＤ５での毎2^{64}操作で見つかることになる）。これが明らかになるのはシードを用いてキーのレンジを作り、それで仮定として暗号化したあるデータについてその一つを試験するときだけである。一つのレベルを破ることに成功すると、次のレベルは再び２倍価値のあるものとなるが、破るためには同じ野蛮な（勇猛な）力（ブルートフォース）のいる努力が必要となる。一つのＭＤ５ハッシュ（ポータブルソース）を１２８ｂ入力についてすることはSun SPARCサーバ１０００で約４μｓかかるので、2^{128} ＭＤＳは４ｅ２５（4×10^{25}）年かかることになる。

【００９５】

ＢＨＣ－Ｔ

ＢＨＣ－Ｔハイブリッドについてみると、攻撃に対する強度はその攻撃がどの方向をとっているかに依存する。もしＢＨＣ－Ｔの単一要素をとるとすると、そこには四つのシード値があり、２人の親と２人の子供であり、これが表３と図１７とにも示されている。四つの値のうちのいずれか一つだけが与えられるとすると、他のいずれもが計算されることはできず、その理由は、正確さを試験するためには不十分な情報しか存在しないことによる。もしちょうど二つだけの値が与えられたとすると、この表は他の二つを計算することが、どの二つが与えられたかに依存して、どんなにむづかしいかを表示している。文字ｉは入力値を表わし、セル内の値は入力が与えられたとして、出力値の対を見付けることを保証するのに必要なブラインド機能操作の数を表わしている。ｗは数字空間内のビット数であり、ＭＤ５については１２８である。図１７は同じ情報を図式的に示し、入力値は丸を付してあり、またブラインドされた値は灰色の（グレイ）背景上に示されている。

【００９６】

【表３】

表３．ＢＨＣ－Ｔにおけるシード対の啓示とブラインド

親	s(0,0)	i	$1+2^{(w+1)}$	i	$1+2^{w}$	i	2
	s(0,1)	i		$1+2^{w}$	i	2	i
子供	s(1,1)	2	i	i	$1+2^{w}$		i
	s(1,2)		i	$1+2^{w}$	i	i	2

【００９７】
もし、親と子供の“方形”の片側下が与えられると、反対側の親が大層な規模で探査でき、各値がそれをブラインドすることにより、また二つの与えられた値のＸＯＲで比較されることにより試験される。こうして成功が“側面”からの攻撃について2^{w}回のブラインド操作後に保証される。
【００９８】
もし、二つの子供の値だけが与えられると、親の一方についての大規模な探査が僅かに多く関与することになる。すなわち、一方の親の値ｓ（０，１）が推量されて、次のことが真であれば正しいものとされる。
【００９９】
ｃ(ｓ(０，１)，ｂ(ｃ(ｓ(１，１)，ｂ(ｓ(０，１)))))＝ｓ(１，２)
したがって、“上方”への攻撃について$2^{(w+1)}$ブラインド操作後に成功が保証される。
【０１００】
二つの与えられた値と両立可能性があるが正しい値の対ではない二つの未知の値（二重の衝突）を見付けることの確率はこの構成では小さい。このような対が生ずる（turn up）すると、それらの値でキーを作り暗号化したデータをキーを試験することによってのみそれらの値が試験されるようにできる。二重衝突のより小さな確率はそこで攻撃者の課題の複雑を僅かに減らすことになる。
【０１０１】
側部攻撃は最高のシードですでに知られているのと同じレベルでたかだか一つのシードを得ることだけができる。右への攻撃は偶数のインデックスが付いている子供で終るが、それは右への次の“箱”の中で一つの値だけが知られていることによる。同じように左への攻撃は奇数のインデックスが付いている子供によって阻止される。上への攻撃はそこで唯一の残っている選択肢ということになる。一つの成功する上への攻撃は余分のキーを何も与えないが、側部攻撃が続くときには最終の側部攻撃のキーの二倍を啓示する。
【０１０２】
ＢＨＴ２
攻撃に対するＢＨＴ２の強度はＢＨＣ－Ｔハイブリッドのものに対するのと同じ形式をとり、例外は上への攻撃に対する強度がもっと大きくなるように設定されていることである。ＢＨＣ－Ｔについてのように、四つの“方形”からのたった一つの既知の値は他のいずれをも決して啓示することはできない。しかしながら、ＢＨＣ－Ｔとは違って、三つの値は直ちに第四のものを与える必然性はない。もし、一方の親だけが未知であるとすると、2^{w}のブラインド操作がそれを見付けることを保証するために必要とされる。二つだけの値が与えられたとすると、表４はどの二つが与えられているかに依存して、他の二つを計算することがどのくらいむづかしいかをリストとしてある。前のように、セル内の値は入力が与えられたとして、出力値の対を見付けることを保証するのに必要なブラインド操作の数を表わしている。図１８は同じ情報を図式的に示し、入力値が丸を付してあり、ブラインドした値はグレイ背景上に示されている。
【０１０３】

【表４】

表４．ＢＨＴ２におけるシード対の啓示とブラインド

親	s (0,0)	i	2^{2w}	i	$3+2^w$	i	$3+2^w$
	s (0,1)	i		$3+2^w$	i	$3+2^w$	i
子供	s (1,1)	4	i	i	$3+2^w$		i
	s (1,2)		i	$3+2^w$	i	i	$3+2^w$

【０１０４】
図１８はＢＨＴ２におけるシードサブセットの啓示とブラインドを示している。
【０１０５】
もし一つの親といずれかの子供で“方形”の片側の下のものとが与えられると、反対側の親が大規模に探査でき、各値はそれをブラインドして二つの既知の値のＸＯＲとそれを比較することにより試験される。こうして成功が“側部”攻撃についての2^wのブラインド操作後に保証される。一人の親と反対側の子供とが与えられるとすると同じことが適用される。
【０１０６】
もし二人の子供の値だけが与えられるとすると、両親についての大規模な探査が設計されて、ＢＨＴ２に含まれるもっと多くのものとなる。右の親の値ｓ（０，１）での各推測に対して、それが左ブラインドされねばならず、そこで左の親の値が大規模に探査されて、左ブラインドされた値を見付けるようにしなければならず、この値は第一の左ブラインドされた推測と組合されるときには左の子供の与えられた値を与える。しかしながら、これら二つの親の推測は右ブラインドされるときには、正しい右の子供をあたえるために組合されそうもない。したがって、右の親での次の推測は左親のブラインドされた値の大規模な探査と組合されなければならず、このようなことが続行される。これは次の同時に成立する方程式を解くのと同じであり、条件はｓ（１，１）とｓ（１，２）とだけが与えられているものとする。
【０１０７】
$c(b_0(s(0,0), b_0(s(0,1)))) = s(1,1)$
$c(b_1(s(0,0), b_1(s(0,1)))) = s(1,2)$
成功を保証するためには、ここで二つの親の組合せの方形マトリックスについての大規模な探査を必要とし、これが2^{2w}ブラインド操作となる。野蛮な力の攻撃を子供の中から親に向けてすることに対抗するより大きな力が濃いグレイ背景により図示されている。代りとなるものは左と右との、一方の親のブラインドした値のすべてを記憶して、再計算を維持するようにすることである。しかし、一方の親の可能とされる値のすべてについてインデックスを付けていない左ブラインド値だけでメモリの５ｅ２７ＴＢ（テラバイト）以上を使ってしまうことになり、このコストは他の攻撃手段がもっと経済的に価値があるものとしてしまう。
【０１０８】
二重の衝突についての同じコメントがＢＨＣ－Ｔについてしたのと同じようにＢＨＴ２に対して適応するが、値についての悪い対は、四つのハッシュ衝突が同時に遭遇するとして現れるという例外があるが、これはほとんどないに等しい小さな確率の事象である。
【０１０９】
ＢＨＴ２の側部攻撃は、それらがＢＨＣ－Ｔ内であるのでたかだか一つの“箱”にいずれにしても限定される。したがって、キーについて何らかの顕著な数を得るためには上への攻撃が間もなく直面されるようにならなければいけない。側部攻撃についての2^wのブラインド操作は恐らくは攻撃されているキーを合法的に取得するよりも経費がかかるものと

なろう。一旦、上への攻撃に直面しなければならなくなると、2^w のブラインド操作が他の方法を見付けるための刺戟にたしかになる。

【０１１０】

一般的なセキュリテイ（安全性）

一般に、一つのトリー（木）を構築するために必要とされるランダム値が多くなると、より一層各新しいランダムシードから作られる境界内部に向けた受けることになる攻撃を含むことができる。しかしながら、長く実行されているセッションについては、セキュリテイと連続しているキー空間の便利さとの間のトレードオフが存在することは連続するトリーとの関係で前述した通りである。ランダムに生成されたシードのランダムさは正しく設計されなければならないとされる弱さについての別の潜在的な領域である。

【０１１１】

すべてのＭＡＲＫＳ構成は有効なグループメンバー間の通謀（collusion）に向けての攻撃を受け易いものである。もしメンバーのサブグループがメンバーの中で同意して、各々がキー空間の異なるレンジを購入するとすると、メンバー達は送られたシードを皆んな共用できて、それにより皆んながそうでなければ別個なキー空間を統合したものをアクセスすることができる。アービトラージュ（arbitrage）はメンバーの通謀（すでに論じた）の変形である。これは、一つのグループメンバーが全体のキーシーケンスを購入して、次にその一部をその売り値よりも安く売り、それでもなお、大部のキーが一人の顧客ではなく多勢によって買われるとすると利益を得る場合である。グループでないメンバーとの通謀に対する保護についてはウォーターマーキング（すかしのしるし）について別節（シーケンシャルでない、またはマルチシーケンシャルなキーアクセス）で記述する。

【０１１２】

最後に、いずれかの特定の応用についての全体のシステムのセキュリテイは明らかにセッションを設定するときに使用されるセキュリテイの強度に依存している。上記の例にあげたシナリオは強調される必要のある問題点（issues）を記述しており、適切とされる標準的な暗号技術を示唆している。いつもそうであるように、いずれかのＭＡＲＫＳ構成を用いる応用の全体のセキュリテイは一番弱い部分と同じに強いものである。

【０１１３】

キー管理スキームで現在の仕事の中で記述されているものは他の機構（メカニズム）とモジュール形式で組合せるためにそれ自体を役立たせていて、以下に記述する追加の商用要件に適うようにしている。

【０１１４】

複数の送り側のマルチキャスト

複数の送り側のマルチキャストセッションは、同じキーシーケンスをすべての送り側が使用する限りは、ＭＡＲＫＳ構成を用いて安全とされるようにできる。使用するキーがすべて同じシーケンスの部分である限りは、同じキーを同時にすべてが使用していることを要しない。受け側はどのキーを使うべきかを知ることができ、それは各送り側が他のものとシーケンスから外れていても言えることであるが、ＡＤＵインデックスが平文で暗号化されたＡＤＵについてのヘッダとして送られることを条件とする。例としてあげたシナリオはどのようにして複数の送り側が使用しているＡＤＵインデックスを同期するかを、もしこれが応用の商用モデルにとって重要であったとして、記述した。

【０１１５】

もし複数の送り側のマルチキャストで各送り側が異なるキーまたはキーシーケンスを使用するとすると、各送り側は、皆が同じマルチキャストアドレスを使用する場合であっても、異なるマルチキャストセッションを作っている。これはマルチキャストセッションと前出の節で定義した安全なマルチキャストセッションとの間の差異から続いているものである。このような場合には、各安全なマルチキャストセッションは作られて、他のものからは別に維持されなければならない。しかしながら、“amortised initialisation”［Baleｎ 99］（償還した初期化）と呼ばれているものについての若干のスコープ（見通し）があってよい。言い換えると、はっきりとした安全なマルチキャストセッションがすべて同じ

設定データをメッセージ送りをセーブするために使用できる。例えば、各々からともかくもいずれかを購入することを条件に、商用モデルは、顧客が常に同じＡＤＵを関係している送り側の組の各個から購入しなければならない。このようなシナリオでは、各送り側はその送り側に特有の長期間キーをもつすべての送り側に共通のキーのＭＡＲＫＳシーケンスを組合せることになる。顧客はキーの共通レンジについての関連するシードを購入することがでたときには、その者が解号したいと望んでいる各送り側についての追加の長期間キーを購入する。

【０１１６】

シーケンシャルでない、またマルチシーケンシャルなキーアクセス

ＭＡＲＫＳ構成は、各受け側にキーシーケンスへのアクセスを与えるときに効果があるように設計されていて、このキーシーケンスはより広いシーケンスの任意のサブレンジであるが、しかしデータがシーケンシャルではない場合ではないし、あるいはシーケンスの任意のばらばらにされた部品が必要とされる場合ではないとする。したがって、ＭＡＲＫＳは、実時間マルチメディアストリームのような一次元で自然にシーケンシャルとなっているデータストリームでターゲットとされる。

【０１１７】

しかし、受け側がキーのレンジにアクセスをもつようになると、明らかにそこにシーケンシャルな順序でアクセスすることを強制されない。例えば受け側はＭＡＲＫＳキーシーケンスの一つを用いて暗号化されて、インターネット上でマルチキャストされている音楽の流れのサブレンジを記憶できる。ダウンロードされたトラックのインデックスを用いて、受け側は後にランダムな順序で聴くためにトラックをピックアップし、その際にＭＡＲＫＳシーケンスから順序を外れて採った関連のキーを使用する。

【０１１８】

ＭＡＲＫＳはまたシーケンシャルではあるが複数のディメンション（次元）にあるデータへのアクセスを制限するために使用することもできる。このような応用のいくつかの例は、M.Fuchs, C.Diot, T.Turletti, M.Hoffman,“A Naming Approach for ALF Design”, proceedings of HIPPARCH workshop, London, (June 1998)に記述されている。三次元のキーシーケンス空間は図１９に示されている。

【０１１９】

例えば、マルチキャストストック引用（quote）へのアクセスは加入についての継続期間によるのと加入する将来のマーケットのレンジによるのとの両方で販売されるようにできる。各引用はそこで二つの中間キーを一緒にＸＯＲしたもので暗号化されなければならなくなる。こうして“最終のキー”として暗号化に使用されるものが、

$k_{i,j} = c(k'_{0,i}, k'_{i,j})$ となる。

【０１２０】

一つの中間キーはシーケンス $k'_{0,i}$ からのものとなろう、ここでｉは毎分歩増（インクレメント）する。他の中間キーはシーケンス $k'_{i,j}$ からとすることができ、ここでｊは引用の将来にわたる月数を表わす。一年ないし二年先に特定している商人はその者がどのくらい長く加入を望んでいるかに依存して $k'_{0,i}$ の関連するサブレンジを購入することになるだけでなく、中間キー $k'_{1,12}$ ないし $k'_{1,24}$ のレンジをも購入することになる。

【０１２１】

Ross Anderson & Charalampos Manifavas(Cambridge Uni),“Chameleon-A New Kind of Stream Cipher”Encryption in Haifa(Jan 1997)（前述）、のようなやり方がデータのストリームを解号化するために使用されるキーをウォーターマークする（透かしを入れる）ために使用できる。このようにして、ＭＡＲＫＳのいずれかの構成により生成されるキーは中間キーとして取扱うことができる。送り側は前節で述べたように長ターム（長期間）のキーブロック（具体的な例では５１２ｋＢ（キロバイト））をもつ各中間キーを組合せることによって最終のキーのシーケンスを作る。各受け側は、同じブロックの長タームウォーターマークとしたバージョンを与えられて、中間キーについての受け側のシーケンスから最終キーのウォーターマークとしたシーケンスを作り、これによってウオーターマー

クした解号化を強制する。

【０１２２】

しかしながら、このやり方はChameleon（カメレオン）での一般的な不備（flaw）を背負うことになる。このやり方は反逆的な商人された受け側により完全に承認されていない受け側（すなわち受け側には長タームキーブロックをもっていない。）へ送られたいずれかのキーもしくはデータについての監査試行を作り出す。このような場合には、反逆者でキーもしくはデータを啓示するものは、そのキーまたはデータがトレースされているとすると、トレースされることができる。しかし、中間キー（最終キーではない）はいずれもの受け側にも送られるようにでき、この受け側はある時刻に依然有効である長タームキーブロックを与えられるものとする。したがって、受け側で、中間キーの組のうちの特定のキー（これはウォーターマークされていない）についての資格を与えられていないものが、受け側キーブロックでウォーターマークされた最終キーを作ることができて、それにより、暗号ストリームを解号できる。作られたキーとデータとは受け側自体のウォーターマークでスタンプされているけれども、これは単に監査試行を洩れのターゲット（ソースではない）に与える。したがって、“内部の”反逆者のこの形式に対してはほとんど抑止力が存在しない。

【０１２３】

ＭＡＲＫＳ構成の特定の場合に戻ると、Cameleonの一般的な不備は、中間シードもしくは中間キーのいずれかが監査試行についての恐れなしに内部で送られるようにできることを意味している。例えば、上述の網ゲーム例では、遊戯者のグループが通謀することができて、それぞれが違ったゲーム時間を購入してそれぞれが購入できる中間シードを仲間の間で共用できるようにする。実際のキーを作るために、各遊戯者はそこで自分自身のウォーターマークした長タームのキーブロックでゲームを遊ぶのにその者が必要とするものを使用できる。監査試行はウォーターマークされていない中間シードを誰が送ったかについてのトレースするために作られることがない。しかし、遊戯者のいずれかが最近にゲームをしていない誰かにウォーターマークしたキーまたはデータを送ることを試みておらず、したがってその者たちの自体の有効な長タームキーブロックをもっていないとすると、監査試行が存在する。同様に、代ってもし遊戯者の一人がその長タームキーブロックを送ったとすると反逆的受け側にとってトレース可能なウォーターマークを含んでいるので、監査試行が存在する。

【０１２４】

計画されていない立退き（unplanned eviction）

すでに指摘したように、ＭＡＲＫＳ構成は、任意の時刻にグループから立退くことを許しているが、その条件はその時刻に各受け側のセッションが設定されていることに限るとしている。もし予め計画された立退きが普通の場合であるが、時たま計画されていない立退きが必要とされるのであれば、ＭＡＲＫＳスキームのいずれかが他のスキーム、例えばＬＫＨ++［Chang 99］、と組合されて、時たまの計画されていな立退きを許すようにすることができる。これを達成するためには、上述のウォーターマークでのように、ＭＡＲＫＳ構成のいずれかによって生成されたキーシーケンスが中間キーとして取扱われる。これらが分散されたグループキーと組合され（例えばＸＯＲされて）、例えばＬＫＨ++を用いて、データストリームを解号するために使用される最終キーを作るようにする。ＭＡＲＫＳ中間キーとＬＫＨ++中間キーとの両方はいずれかの一時に最終キーを作るのに必要とされる。

【０１２５】

実際には、中央キーのいずれかの数が組合せされて（例えばＸＯＲを用いて）複数の要件を同時に適うようにできる。例えば、ＭＡＲＫＳ，ＬＫＨ++とChameleon中間キーは組合されて、低コストの計画された立退きと、時たまの計画されていない立退きと、ウォーターマークした監査試行で長タームグループ外部へのもれに対抗するものとを同時に達成するようにできる。

【０１２６】

正式には、最終キーは　$k_{i,j,\ldots} = c(k'_{0,i}, k'_{1,j,\ldots})$
であり、ここで中間キーｋ′はＭＡＲＫＳ構成もしくはウォーターマーキングと多次元キーシーケンスについて前二節で記述してようないずれかの他の手段を用いるシーケンスから生成されるようにできる。

【０１２７】
一般に、このやり方の組合せは、個別の部品スキームの和であるメモリコストをもつ併合したスキームを作る。しかしながら、ＬＫＨ++をＭＡＲＫＳとを組合せ、そこでは大部分の立退きが計画されているようにすることは、計画されていない立退きが実際に必要とされている場合を除けば、ＬＫＨ++の再キーイングメッセージのすべてを切り捨てることになる。

【０１２８】
この発明は無論のことマルチキャストデータ網とともに使用することに限定されない。他の二つの使用分野が例として以下に記述される。

【０１２９】
仮想私設網（ＶＰＮ）
大きな会社はその従業員と契約者とがその会社の他の当事者といずれの場所からもインターネット上でＶＰＮを設定することによって通信することができるようにすることができる。これを行うための一つの方法はすべての作業者に全社で使用されるグループキーを与えることである。その結果、ある作業者が会社に加わりあるいはそこから立ち去る度毎に、グループキーが変更されなければならない。これに代って、キーはＭＡＲＫＳ構成の一つによって決められたシーケンスの中で定期的に変更されなければならないとし、作業者が参加もしくは退去したかしないかによらないとする。新しい顧用契約が設定される毎に、シードが各作業者に与えられて、各作業者がシーケンス内の次のキーを計算できるようにし、その者の契約が交信されるまではそのようにする。いずれかの作業者で早期に立ち去る者は計画されていない立退きとして処理される。

【０１３０】
ディジタルバーサタイルディスク（ＤＶＤ）
ＤＶＤはもともとはディジタルビデオディスクであったが、その理由はその容量がこの媒体に適していたからであった。しかしながら、これが僅かなメモリ空間を必要とするソフトウェアやオーディオといったコンテンツを記憶するのに使用されてもよい。オーディオトラックもしくはソフトウェアタイトルの各選択のために異なるまばらに満たされたＤＶＤをプレスするのに代って、この発明を用いて、各ＤＶＤは数百もの関係するトラックもしくはタイトルをもつ容量を一杯とするように作られる。各トラックもしくはタイトルはＡＤＵを構成している。各ＡＤＵはＭＡＲＫＳ構成の一つを用いて作られたシーケンスからの異なるキーで暗号化されるようにできる。こういったＤＶＤは大量生産できて、空き（フリー）を、例えばマガジン上のカバーディスクのように、与えられる。こういったＤＶＤの一つを保有している者は誰でもそこでインターネット上でシードを購入することができ、これがＤＶＤ上のＡＤＵのあるもののロックを解くためのキーのレンジへのアクセスを与えることになる。ＭＡＲＫＳはこのようなシナリオに対して理想的に適していて、その理由は暗号キーはＤＶＤが一度プレスされると変化されるように出来ず、したがって商用のモデルであって物理メディアを使用するものが計画されていない立退きに頼る傾向がないことによる。このスキームはキーとデータとをウォーターマークするためにChameleonと有用に組合されるようにできる。

【０１３１】
非常に大きなグループのキーを管理するための解を上述してきた。それによると、受け側が開始したインターネットマルチキャストのスケーラビリティを、送り側をすべての受け側の参加と退去活動から完全に切り離すことにより保存している。送り側はまたこの受け側の活動を吸収するキーマネージャから完全に切り離されてもいる。たくさんの商用応用がモデルとして状態のない（着手する処理についての特定状態を記憶していない、換言すれば有限状態機械でない）キーマネージャだけを必要とするものをもつことを示した。こ

の場合には制限されていないキーマネージャの複製（レプリケーション）が実現可能となる。状態のないキーマネージャの複製された組の一つが故障したときには、仲間のサーバでの進行中のトランザクション（業務）に何も影響を与えないので、問題からの全体のシステムの隔離が弾力性を改善している。こういった点を例示するために、大規模網のゲームで分毎に課金される作業ずみの例を呈示した。

【０１３２】

こういった利得は受け側の参加もしくは退去活動で再度のキーイングを駆動するのではなく、組織的なグループキー変更を使用することにより達成されている。切り離し（デカップリング）は、送り側とキーマネージャによってマルチキャストデータストリームにおける経済的価値の単位（“応用データユニット”で課金と関係しているもの）を前もってこしらえておくことによって達成される。組織的なキー変更は、そこでそのデータ内で宣言されたＡＤＵインデックスを歩増させることによって信号を発するようにできる。このモデルを用いると、受け側にはなにも脇の効果（サイドエフェクト）がないし、一つの受け側が参加したり退去したりするときに送り側にも効果を生じない。マルチキャストがキー管理のために使用されず、データ転送のためだけであることを確めた。したがって再度のキーイングはランダムな伝送損失に対して脆弱ではなく、マルチキャストを使用するときにスケーラブルに修復することが複雑な損失を受けにくい。伝統的なキー管理解法が技術のスケーラビリティを成功裡に改善して、グループメンバーの計画されていない立退きを許すようにしたが、しかし、最善の技術は依然としてメッセージングという項目でコストのかかるものとなっている。これとは対照的に、ここでは計画された立退きの問題に焦点をあてた。すなわち、ある任意の将来のＡＤＵの後の立退きであるが受け側があるセッションを要求した時刻に計画されたものである。たくさんと商用シナリオで前支払いもしくは加入を基礎としたものは計画されていない立退きを必要としないが、任意の計画された立退きを求めている点を主張した。この例は、ペイＴＶとか、ペイパービューＴＶとかネットワークゲーミングなどである。

【０１３３】

計画されてはいるが任意の立退きを達成するために、ここではキーシーケンス構成であって、送り側によって組織的にグループキーを変更するために使用されるものの選択を設計した。こういった構成はシーケンスのいずれものサブレンジが少数のシード（それぞれ１６Ｂ）を啓示することによって再構成されるようにできるよう設計されている。したがって、受け側はデータシーケンスの任意のサブレンジに対するアクセスを与えられるようにできる。実用的なスキームのすべてはＯ（log(N)）シードを用いて各受け側へのＮのキーを啓示することができる。このスキームは各キーを計算するための処理のロード（負荷）で違っていて、これがセキュリティとのトレードオフとなっている。一番重いスキームは平均でちょうどＯ（２(log(N)－１)）高速ハッシュ操作を開始するのに必要とし、続いて、平均でちょうど１６余計にハッシュをシーケンス内で新しい各キーを計算するのに必要とし、これは先行して実行できる。一番軽い機構はこれよりも４倍少い処理を必要とする。

【０１３４】

この作業を文脈に加えて、ペイＴＶで秒当り課金され、一千万人の看者の１０％が１５分間に同調をとったり、同調を外したりする場合についてみると、最善の代替スキーム（Chang et al.）では、再キーメッセージとしてすべてのグループメンバに向けてマルチキャストする毎秒数千バイト（ｋＢ）の１０倍オーダーを生成することになる。この発明の作業は数百バイトのユニキャストメッセージを多分四時間の視聴の開始時に各受け側に向けて一度だけ送ればよい。

付録Ａ－ＢＨＴについて中間シードの最小組を識別するためのアルゴリズム

次のＣのようなコードの部分では

・機能odd(x)はｘが奇数であるかどうか試験をする。また、

・機能reveal(d,i)は受け側に対してシード$s_{d,i}$を啓示する。

【０１３５】

【表５】

```
min=m; max=n;
if (min max) error();  // reject min max
for(d=D; d=0; d--) {    // working from bottom of tree...
                        // move up the tree one level each loop
    if (min == max) {   // min & max have converged...
        reveal(d,min);  // ...so reveal root of sub-tree...
        break;          // ...and quit
    }
    if odd(min) {       // odd min values are never left
children...
        reveal(d,min);  // ...so reveal odd min seed
        min++;          // and step min inwards one seed to right
    }
    if !odd(max) {      // even max values are never right
children...
        reveal(d,max);  // ...so reveal even max seed
        max--;          // and step max inwards one seed to left
    }
    if (min max) break; // min & max were cousins, so quit
    min/=2;             // halve min ...
    max/=2;             // ... and halve max ready for...
}                       // ... next level up round loop
```

【０１３６】

付録Ｂ－ＢＨＣ－Ｔについての中間シードの最小組を識別するためのアルゴリズム。

【０１３７】

次のＣのようなコードの部分では、

・機能odd(x)はｘが奇数であるかどうか試験をする。また、

・機能reveal(d,i)は受け側に対してシード$s_{d,i}$を啓示する。

【０１３８】

【表６】

```
min=m; max=n;
if (min max) error();   // reject min max
d=0;                    // working from bottom of tree
    if (max <= min+1) {   // requested min & max are adjacent/the
same...
        reveal(d,min);    // ...so reveal left...
        if (max < min)   // requested min & max are not the same...
            reveal(d,max); // ...so reveal right too...
        break;            // ...and quit
    }
for(d=0; ; d++) {         // move up the tree one level each loop
    if (max <= min+3) {   // min & max are two or three apart...
        if (max < min+3) {    // min & max were two apart...
            reveal(d,min);    // ...so reveal left...
            reveal(d,max);    // ...and right
            reveal(d,min+1);  // ...and centre...
            break;            // ...and quit
        } else {                  // min & max were three apart,
so...
            if (!odd(min)) {      // ...only if min is even...
                reveal(d,min+1);  // ...reveal left centre...
                reveal(d,max-1);  // ...and right centre...
                break;            // ...and quit
            }
        }
    }
    if !odd(min) {        // even min values are never right
children...
        reveal(d,min);    // ...so reveal even min seed
        min++;            // and step min inwards one seed to right
    }
    if odd(max) {         // odd max values are never left


 children...
        reveal(d,max);    // ...so reveal odd max seed
        max--;            // and step max inwards one seed to left
    }
    min/=2;               // halve min ...
    max/=2;               // ... and halve max ready for...
}                         // ... next level up round loop
```

【０１３９】

【図面の簡単な説明】

【図１】　この発明を実施する網の模式図。

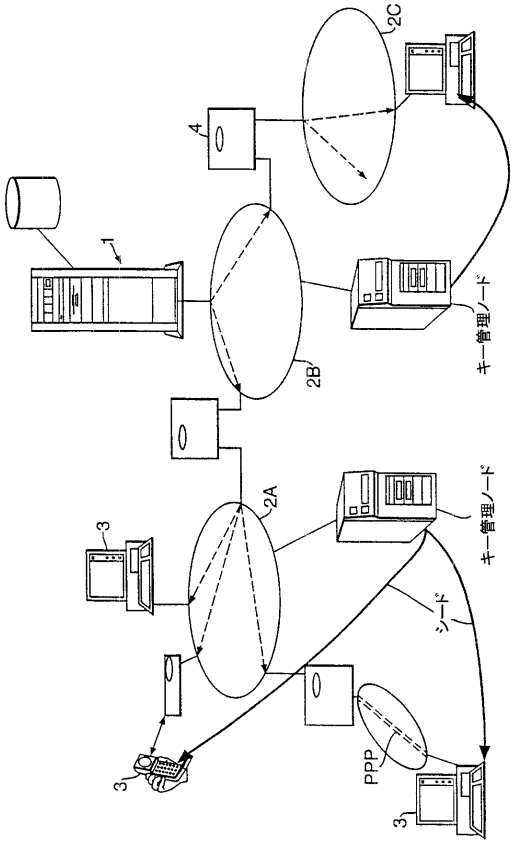
【図２】　図１の網とともに使用するための顧客端末の構造を示す図。

【図３】　図１の網とともに使用するキー管理ノードの構造を示す図。

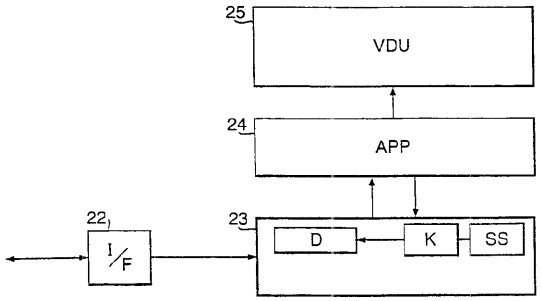
【図４】　図１の網とともに使用するデータ送り側の構造を示す図。

【図５】　図１の網上を送られるデータパケットのフォーマットを示す図。
【図６】　キー管理ノードを経たキーの分配を示す図。
【図７】　双方向ハッシュチェーンを示す図。
【図８】　二つのブラインデング機能の生成を示す図。
【図９】　二進ハッシュトリーを示す図。
【図１０】　連続する二進ハッシュトリーを示す図。
【図１１】　ハイブリッド二進ハッシュチェーントリーの要素を示す図。
【図１２】　ハイブリッド二進ハッシュチェーントリーを示す図。
【図１３】　ハイブリッド二進ハッシュチェーントリーの構造を示す図。
【図１４】　二進ハッシュチェーントリーの成長のシーケンスを示す図。
【図１５】　連続する二進ハッシュチェーントリーハイブリッドを示す図。
【図１６】　第二の二進ハッシュトリーの要素を示す図。
【図１７】　ＢＨＣ－Ｔにおけるシード対の啓示とブラインデングを示す図。
【図１８】　第二の二進ハッシュトリーにおけるシードの啓示とブラインデングを示す図。
【図１９】　多次元キーシーケンスキーを示す図。
【図２０】　普通モデルを示す図。
【図２１】　普通モデルを示す図。
【図２２】　普通モデルを示す図。
【図２３】　普通モデルを示す図。
【図２４】　普通モデルを示す図。

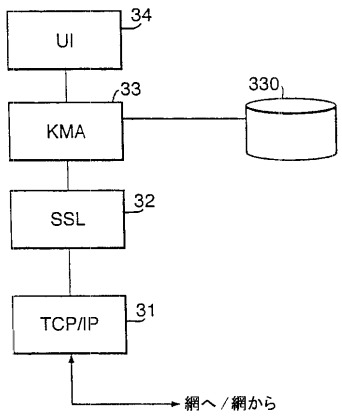
【図１】



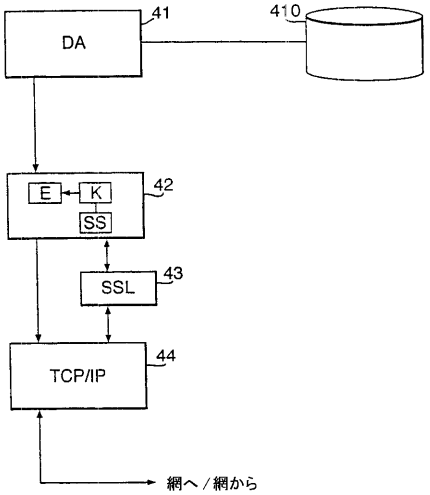
【図２】



【図３】



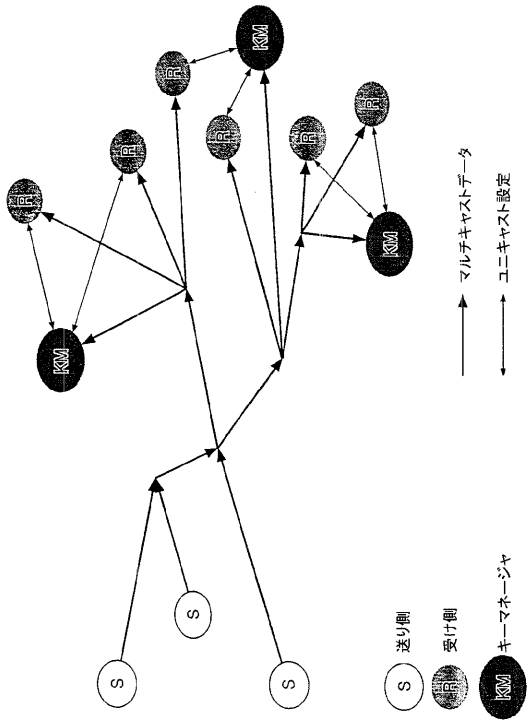
【図４】


DA
41
410
E
K
SS
42
SSL
43
TCP/IP
44
網へ／網から

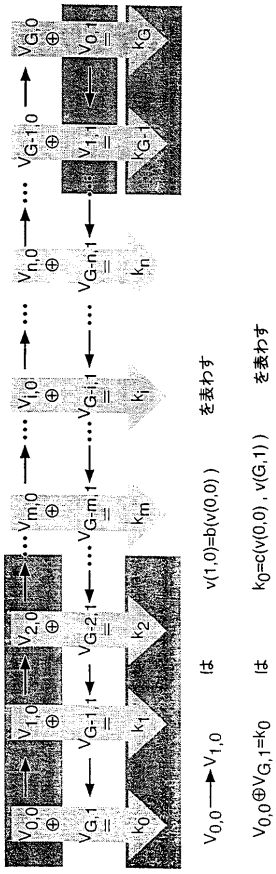
【図５】


データ
ki
SE

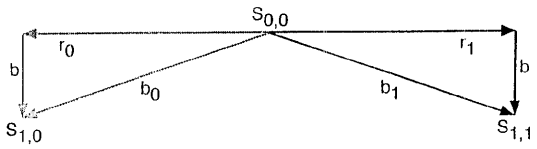
【図６】


KM
RP
S
マルチキャストデータ
ユニキャスト設定
送り側
受け側
キーマネージャ

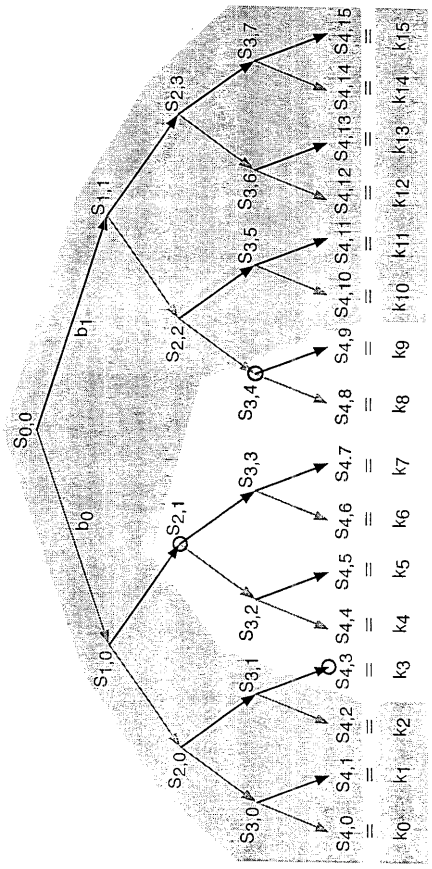
【図７】


V0,0
VG,1
k0
V1,0
VG-1,1
k1
V2,0
VG-2,1
k2
Vm,0
VG-m,1
km
Vi,0
VG-i,1
ki
Vn,0
VG-n,1
kn
VG-1,0
V1,1
G-1
VG,0
V0,1
kG
V0,0⊕VG,1=k0 は k0=c(v(0,0) , v(G,1)) を表わす
V0,0 → V1,0 は v(1,0)=b(v(0,0)) を表わす

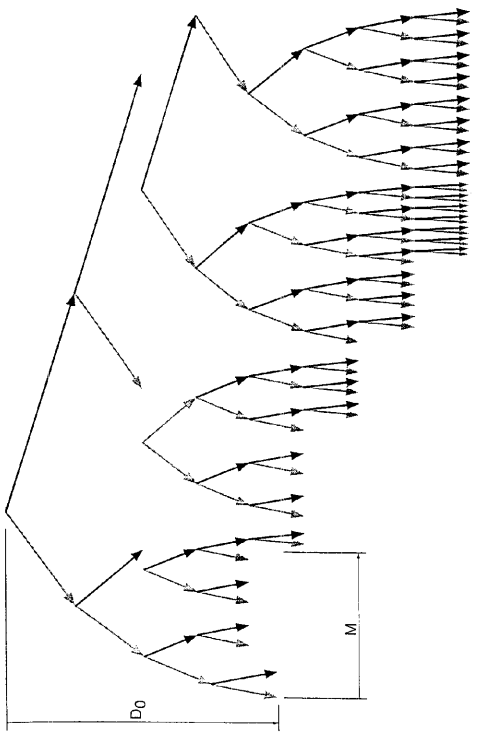
【図８】


S0,0
r0
r1
b
b
b0
b1
S1,0
S1,1

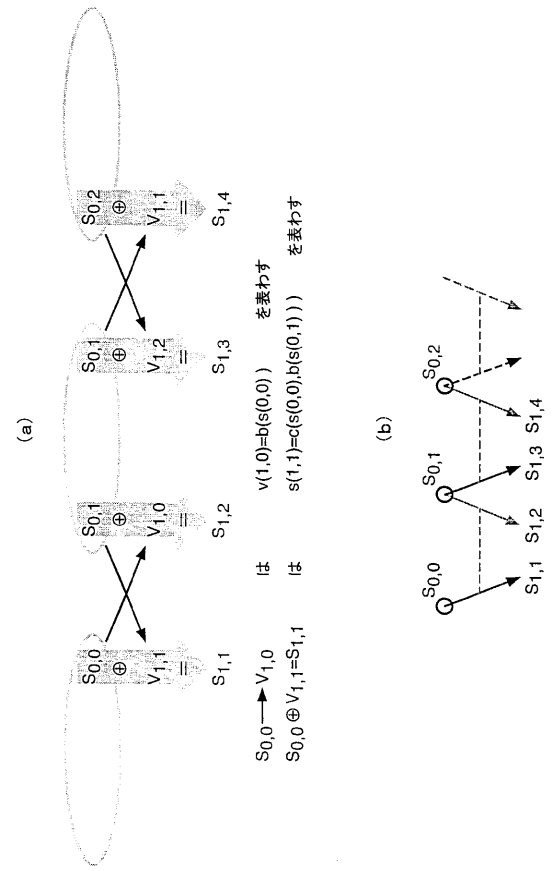
【図９】



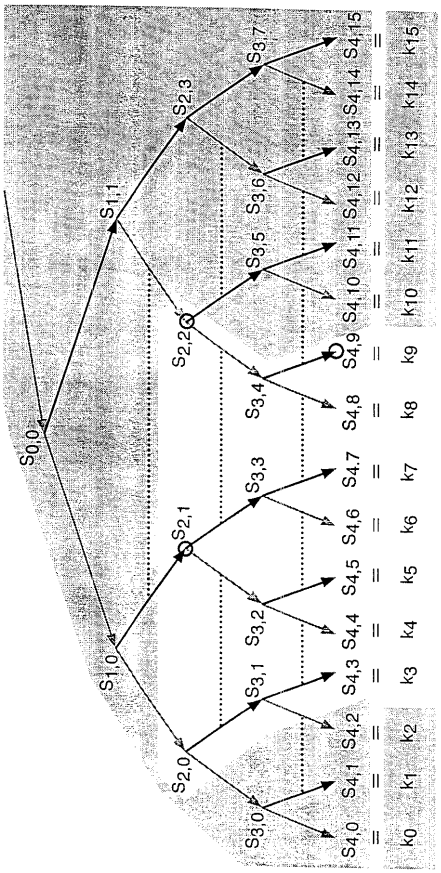
【図１０】



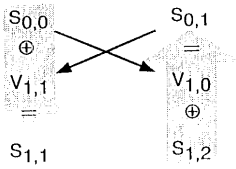
【図１１】



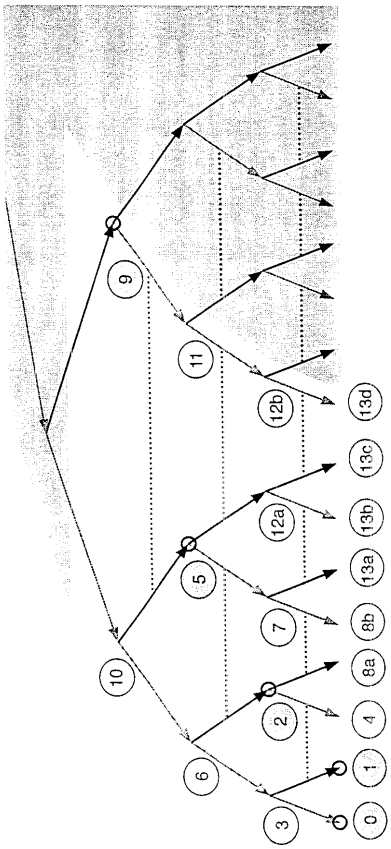
【図１２】



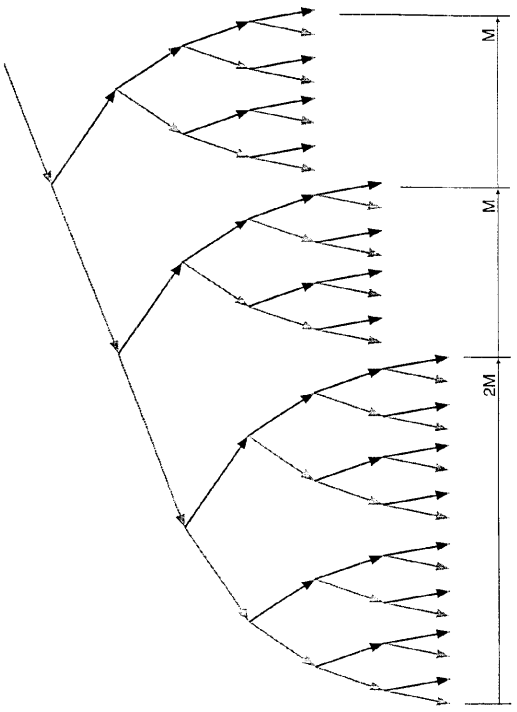
【図１３】



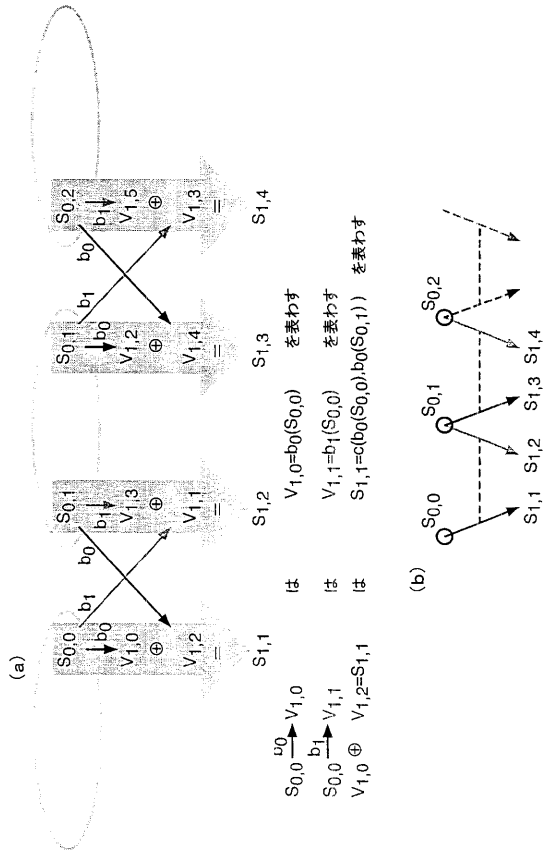
【図１４】



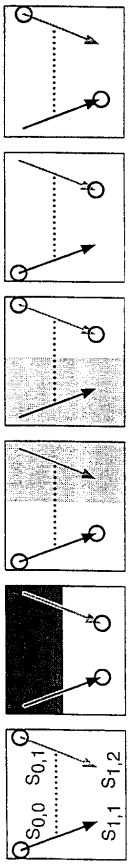
【図１５】



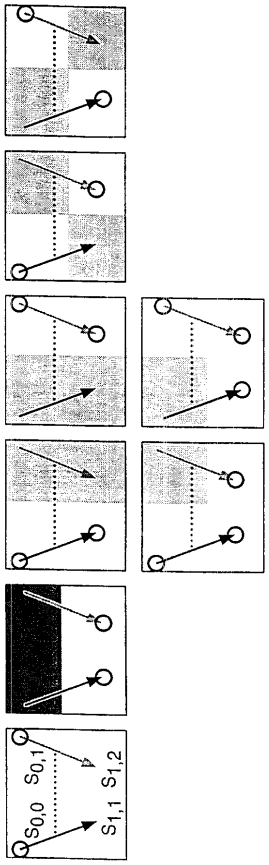
【図１６】



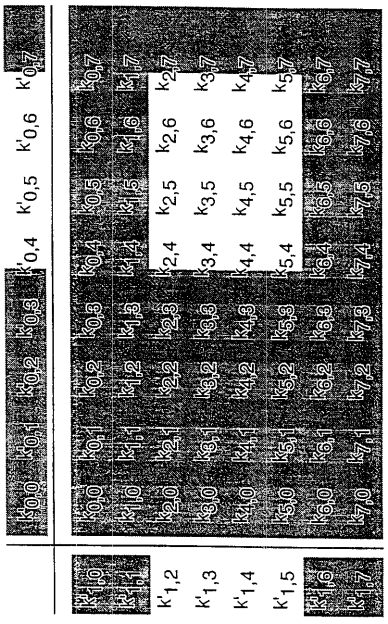
【図１７】



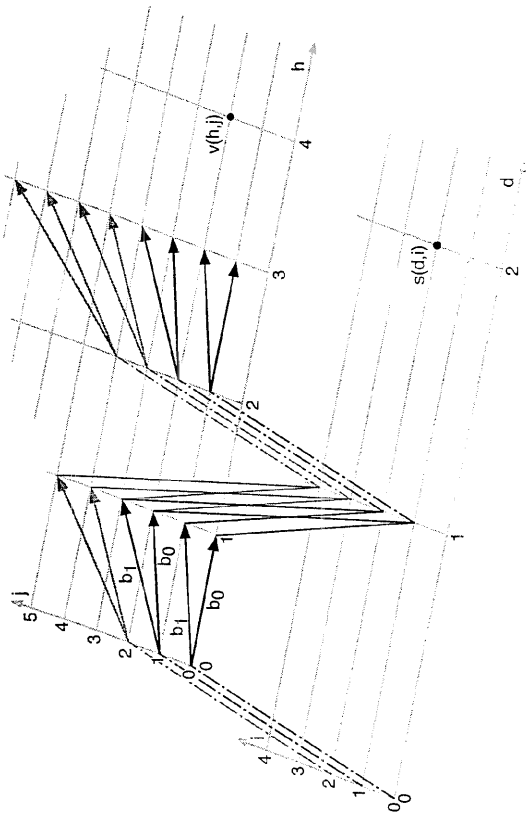
【図１８】



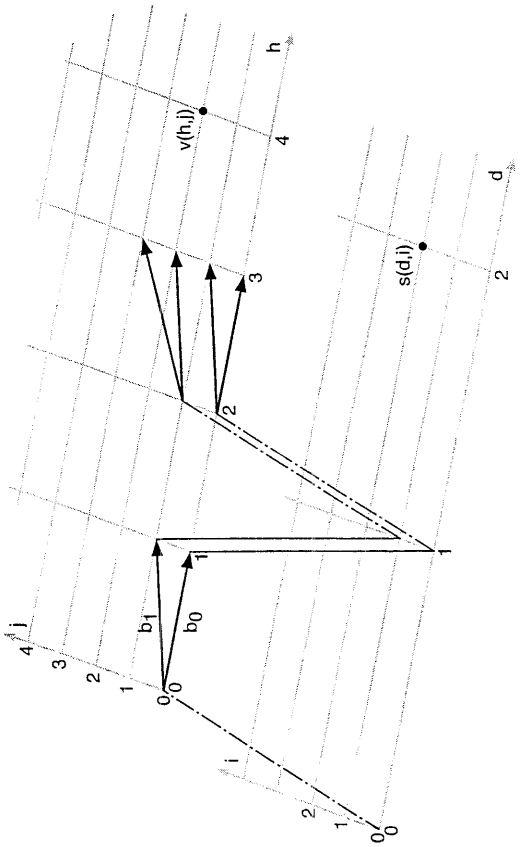
【図１９】



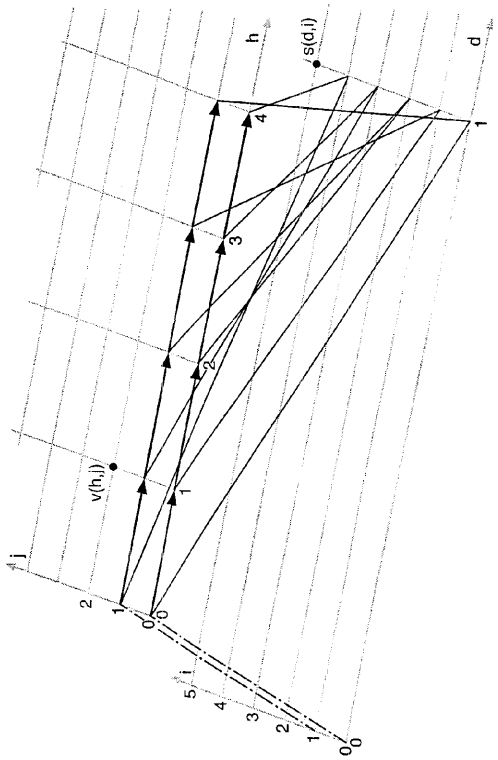
【図２０】



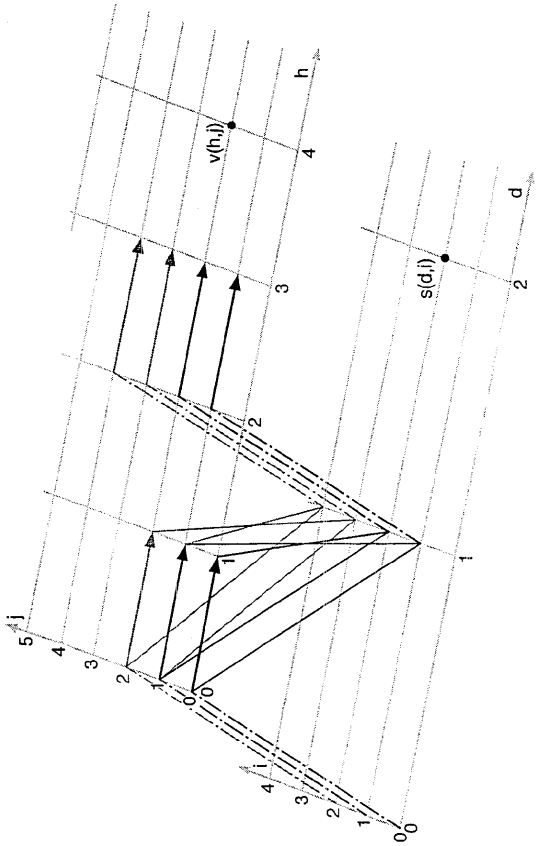
【図２１】



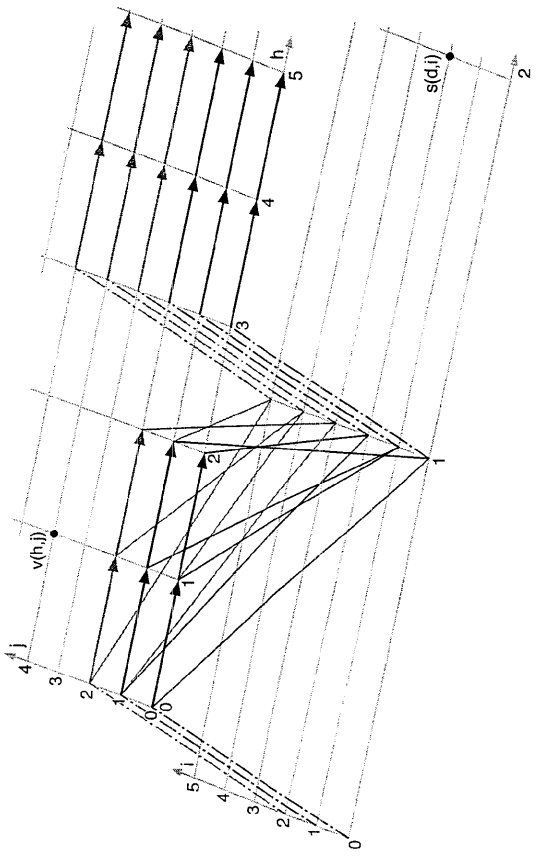
【図２２】



【図２３】



【図２４】



フロントページの続き

(72)発明者　ブリスコー、ロバート・ジョン
イギリス国、アイピー１３・９エヌダブリュ、サフォーク、ウッドブリッジ、パーラム、ホーム・ファーム（番地なし）

審査官　青木　重徳

(56)参考文献　特開平１１－０８５０１４（ＪＰ，Ａ）
特開平１１－０４１２８０（ＪＰ，Ａ）
特開平１０－０６３３６４（ＪＰ，Ａ）
特開平１０－００３２５６（ＪＰ，Ａ）
国際公開第９８／０４５９８０（WO，Ａ１）
D. Wallner, E. Harder, R. Agee, “RFC2627: Key Management for Multicast: Issues and Architectures”, RFC Editor, [online], １９９９年　６月, p.1-23, [retreved on 2010-08-02]. Retreved from the Internet, ＵＲＬ, http://portal.acm.org/citation.cfm?id=RFC2627
Chung Kei Wong, Mohamed Gouda, Simon S. Lam, “Secure Group Communications Using Key Graphs”, ACM SIGCOMM Computer Communication Review, [online], １９９８年１０月, Volume 28, Issue 4, p.68-79, [retreved on 2010-08-02]. Retreved from the Internet, ＵＲＬ, http://portal.acm.org/citation.cfm?id=285243.285260

(58)調査した分野(Int.Cl., ＤＢ名)
H04L　9/08
H04L　9/16
H04L　12/22