(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3864401号
(P3864401)
(45)発行日　平成18年12月27日(2006.12.27)　(24)登録日　平成18年10月13日(2006.10.13)

| (51)Int.Cl. | | F I | | |
|---|---|---|---|---|
| H04L 9/32 | (2006.01) | H04L | 9/00 | 673A |
| G06F 21/24 | (2006.01) | G06F | 12/14 | 530C |
| | | G06F | 12/14 | 550B |

請求項の数 11　(全 50 頁)

(21)出願番号　特願平9-210899
(22)出願日　平成9年8月5日(1997.8.5)
(65)公開番号　特開平11-53264
(43)公開日　平成11年2月26日(1999.2.26)
審査請求日　平成15年6月6日(2003.6.6)
(31)優先権主張番号　特願平9-106104
(32)優先日　平成9年4月23日(1997.4.23)
(33)優先権主張国　日本国(JP)
(31)優先権主張番号　特願平9-143699
(32)優先日　平成9年6月2日(1997.6.2)
(33)優先権主張国　日本国(JP)

(73)特許権者　000002185
ソニー株式会社
東京都品川区北品川６丁目７番３５号
(74)代理人　100082131
弁理士　稲本　義雄
(72)発明者　石黒　隆二
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　大澤　義知
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　刑部　義雄
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

最終頁に続く

(54)【発明の名称】認証システム、電子機器、認証方法、および記録媒体

(57)【特許請求の範囲】
【請求項１】
第１の電子機器と第２の電子機器との間で認証処理を行う認証システムにおいて、
前記第１の電子機器は、
所定の処理を施す情報に対応する第１の鍵を記憶する第１の記憶手段を備え、
前記第２の電子機器は、
自分自身に固有の識別番号を記憶する第２の記憶手段と、
記憶している前記識別番号を前記第１の電子機器に送信する第１の送信手段と
を備え、
前記第１の電子機器は、
前記第２の電子機器から送信されてくる、前記識別番号を受信する第１の受信手段と、
記憶している前記第１の鍵および受信した前記識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出手段と、
あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する第１の乱数生成手段と、
算出した前記ハッシュ値と、生成した前記第１の乱数および前記第２の乱数に基づいて、第１の送信値を生成する第１の送信値生成手段と、
生成した前記第１の送信値を前記第２の電子機器に送信する第２の送信手段と
を備え、

前記第２の電子機器は、
前記第１の電子機器から送信されてくる、前記第１の送信値を受信する第２の受信手段と、
受信した前記第１の送信値に基づいて、前記第２の乱数を生成する第２の乱数生成手段と、
あらかじめ定められたビット数の第３の乱数を生成する第３の乱数生成手段と、
生成した前記第２の乱数および前記第３の乱数に基づいて、第２の送信値を生成する第２の送信値生成手段と、
生成した前記第２の送信値を前記第１の電子機器に送信する第３の送信手段と
を備え、
前記第１の電子機器は、
前記第２の電子機器から送信されてくる、前記第２の送信値を受信する第３の受信手段と、
受信した前記第２の送信値に基づく値と、生成した前記第２の乱数に基づく値とを比較することにより、前記第２の電子機器が正当であるか否かを認証する認証手段と
を備え、
前記認証手段は、比較の結果、前記第２の電子機器が正当でないと認証された場合、認証の処理を終了する
認証システム。
【請求項２】
前記第２の記憶手段は、前記第１の電子機器から送信されてくる所定の情報に所定の処理を施すことに対する許可に対応する第２の鍵を記憶し、
前記第２の送信値生成手段は、前記第２の乱数および前記第３の乱数を結合した値に対して、記憶している前記第２の鍵を用いて暗号的処理を行うことにより、前記第２の送信値を生成する
請求項１の認証システム。
【請求項３】
前記第２の鍵は、前記第１の鍵および前記識別番号に対して、ハッシュ関数を適用して得られるハッシュ値と同一の値である
請求項２の認証システム。
【請求項４】
他の電子機器との間で認証処理を行う電子機器において、
所定の処理を施す情報に対応する第１の鍵を記憶する記憶手段と、
前記他の電子機器に付与された固有の識別番号を受信する第１の受信手段と、
記憶している前記第１の鍵および受信した前記識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出手段と、
あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成手段と、
算出した前記ハッシュ値と、生成した前記第１の乱数および前記第２の乱数に基づいて、第１の送信値を生成する送信値生成手段と、
生成した前記第１の送信値を前記他の電子機器に送信する第１の送信手段と、
前記他の電子機器から送信されてくる、前記第２の乱数および第３の乱数に基づいて生成された第２の送信値を受信する第２の受信手段と、
受信した前記第２の送信値に基づく値と、生成した前記第２の乱数に基づく値とを比較することにより、前記他の電子機器が正当であるか否かを認証する認証手段と
を備え、
前記認証手段は、比較の結果、前記他の電子機器が正当でないと認証された場合、認証の処理を終了する
電子機器。
【請求項５】

所定の処理を施す情報に対応する第１の鍵を記憶しており、他の電子機器との間で認証処理を行う電子機器の認証方法において、
前記他の電子機器に付与された固有の識別番号の受信を制御する第１の受信制御ステップと、
記憶している前記第１の鍵および受信した前記識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出ステップと、
あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成ステップと、
算出した前記ハッシュ値と、生成した前記第１の乱数および前記第２の乱数に基づいて、第１の送信値を生成する送信値生成ステップと、
生成した前記第１の送信値の前記他の電子機器への送信を制御する第１の送信制御ステップと、
前記他の電子機器から送信されてくる、前記第２の乱数および第３の乱数に基づいて生成された第２の送信値の受信を制御する第２の受信制御ステップと、
受信した前記第２の送信値に基づく値と、生成した前記第２の乱数に基づく値とを比較することにより、前記他の電子機器が正当であるか否かを認証する認証ステップと
を含み、
前記認証ステップは、比較の結果、前記他の電子機器が正当でないと認証された場合、認証の処理を終了する
認証方法。

【請求項６】
所定の処理を施す情報に対応する第１の鍵を記憶しており、他の電子機器との間で認証処理を行う電子機器における認証処理用のプログラムであって、
前記他の電子機器に付与された固有の識別番号の受信を制御する第１の受信制御ステップと、
記憶している前記第１の鍵および受信した前記識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出ステップと、
あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成ステップと、
算出した前記ハッシュ値と、生成した前記第１の乱数および前記第２の乱数に基づいて、第１の送信値を生成する送信値生成ステップと、
生成した前記第１の送信値の前記他の電子機器への送信を制御する第１の送信制御ステップと、
前記他の電子機器から送信されてくる、前記第２の乱数および第３の乱数に基づいて生成された第２の送信値の受信を制御する第２の受信制御ステップと、
受信した前記第２の送信値に基づく値と、生成した前記第２の乱数に基づく値とを比較することにより、前記他の電子機器が正当であるか否かを認証する認証ステップと
を含み、
前記認証ステップは、比較の結果、前記他の電子機器が正当でないと認証された場合、認証の処理を終了する
コンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項７】
他の電子機器との間で認証処理を行う電子機器において、
自分自身に固有の識別番号を記憶する記憶手段と、
記憶している前記識別番号を前記他の電子機器に送信する第１の送信手段と、
前記他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および前記識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値を受信する第１の受信手段と、
受信した前記第１の送信値に基づいて、前記第２の乱数を生成する第１の乱数生成手段と、

あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成手段と、
生成した前記第２の乱数および前記第３の乱数に基づいて、第２の送信値を生成する送信値生成手段と、
生成した前記第２の送信値を前記他の電子機器に送信する第２の送信手段と
を備える電子機器。
【請求項８】
前記記憶手段は、前記他の電子機器から送信されてくる所定の情報に所定の処理を施すことに対する許可に対応する第２の鍵を記憶し、
前記送信値生成手段は、前記第２の乱数および前記第３の乱数を結合した値に対して、記憶している前記第２の鍵を用いて暗号的処理を行うことにより、前記第２の送信値を生成する
請求項７の電子機器。
【請求項９】
前記第２の鍵は、前記第１の鍵および前記識別番号に対して、ハッシュ関数を適用して得られるハッシュ値と同一の値である
請求項８の電子機器。
【請求項１０】
自分自身に固有の識別番号を記憶しており、他の電子機器との間で認証処理を行う電子機器の認証方法において、
記憶している自分自身に固有の識別番号の前記他の電子機器への送信を制御する第１の送信制御ステップと、
前記他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および前記識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値の受信を制御する第１の受信制御ステップと、
受信した前記第１の送信値に基づいて、前記第２の乱数を生成する第１の乱数生成ステップと、
あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成ステップと、
生成した前記第２の乱数および前記第３の乱数に基づいて、第２の送信値を生成する送信値生成ステップと、
生成した前記第２の送信値の前記他の電子機器への送信を制御する第２の送信制御ステップと
を含む認証方法。
【請求項１１】
自分自身に固有の識別番号を記憶しており、他の電子機器との間で認証処理を行う電子機器における認証処理用のプログラムであって、
記憶している自分自身に固有の識別番号の前記他の電子機器への送信を制御する第１の送信制御ステップと、
前記他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および前記識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値の受信を制御する第１の受信制御ステップと、
受信した前記第１の送信値に基づいて、前記第２の乱数を生成する第１の乱数生成ステップと、
あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成ステップと、
生成した前記第２の乱数および前記第３の乱数に基づいて、第２の送信値を生成する送信値生成ステップと、
生成した前記第２の送信値の前記他の電子機器への送信を制御する第２の送信制御ステップと
を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。
【発明の詳細な説明】
【０００１】

【発明の属する技術分野】
本発明は、認証システム、電子機器、認証方法、および記録媒体に関し、特に、より安全にデータを授受することができるようにした、認証システム、電子機器、認証方法、および記録媒体に関する。
【０００２】
【従来の技術】
最近、ＡＶ機器やパーソナルコンピュータなどの電子機器を、例えばIEEE１３９４シリアルバスを介して相互に接続し、相互の間でデータを授受することができるようにするシステムが提案されている。
【０００３】
このようなシステムにおいて、例えば一般のユーザが、DVDプレーヤにより再 生出力された映画情報を、１３９４シリアルバスを介してモニタに出力し、表示させる行為は、一般的に、DVD（ディスク）を購入した時点において、映画情報 の著作権者から許容されているものとされる。しかしながら、DVDプレーヤから 再生された映画情報を、光磁気ディスク、その他の記録媒体に記録する行為は、著作権者からの特別な許諾が必要となる。このような場合、例えば、光磁気ディスク装置に、映画情報を記録することが許可されているか否かを表すキーを記憶しておき、このキーを利用して、その光磁気ディスク装置が正当な装置（著作権者からのライセンスを受けた装置）であるか否かを認証するようにし、正当な装置として認証された場合には、その光磁気ディスク装置に映画情報の記録を許容するようにすることが考えられる。
【０００４】
このような場合、映画情報を伝送する側の装置（以下、このような装置をソース（source）と称する）と、伝送を受けた装置（以下、このような装置をシンク（sink）と称する）との間で、相手側の装置が適正な装置であるか否かを認証する必要がある。
【０００５】
図４１は、このような認証を行う従来の方法を表している。同図に示すように、ソースとシンクは、それぞれ著作権者から予め所定の関数ｆを受け取り、それぞれのメモリに記憶しておく。この関数ｆは、その入力と出力から、その関数ｆを特定するのが困難な関数であり、また、知らないものが、その関数ｆに対して任意の入力を与えた場合に得られる出力を推定するのが困難な関数とされる。そして、この関数ｆは、著作権者から許可された装置にのみ与えられ、記憶される。
【０００６】
ソースは乱数ｒを発生し、これを１３９４バスを介してシンクに伝送する。また、ソースは、関数ｆに対して乱数ｒを適用して、ｘ（＝ｆ（ｒ））を生成する。
【０００７】
一方、シンク側においては、ソース側から転送されてきた乱数ｒを関数ｆに適用して、ｙ（＝ｆ（ｒ））を生成する。そして、このｙをソース側に伝送する。
【０００８】
ソース側においては、演算により求めたｘと、シンク側から伝送されてきたｙを比較し、両者が一致するか否か（ｘ＝ｙであるか否か）を判定する。両者が一致していれば、ソース側は、シンク側を正当な装置であると認証し、映画情報を所定のキーで暗号化して、シンク側に伝送する。
【０００９】
このキーとしては、シンクが伝送してきたｙを関数ｆに適用して生成した値ｋ（＝ｆ（ｙ））が用いられる。シンク側においても、同様にして、ｙに関数ｆを適用して、キーｋ（＝ｆ（ｙ））を生成する。そして、このキーｋを用いて、ソース側から伝送されてきた、暗号化されている映画データを復号する。
【００１０】
【発明が解決しようとする課題】
しかしながら、このような方法においては、ソースまたはシンクとして、データを授受す

るすべての電子機器が、同一の関数ｆを秘密裡に保持する必要がある。

【００１１】

その結果、例えば、不正なユーザによって、１つの電子機器に保持されている関数ｆが盗まれてしまったような場合、この不正なユーザは、１３９４バスを介して授受されるデータを監視することにより、鍵ｋを生成することができ、暗号化されているデータを解読することが可能となる。その結果、不正なユーザは、所望の電子機器になりすまして、不正に情報を盗むことが可能となる。

【００１２】

本発明はこのような状況に鑑みてなされたものであり、暗号または復号に必要な情報が盗まれたとしても、不正なユーザが、これを用いて所望の電子機器になりすますことができないようし、より安全性を図るようにするものである。

【００１３】

【課題を解決するための手段】

請求項１に記載の認証システムは、第１の電子機器が、所定の処理を施す情報に対応する第１の鍵を記憶する第１の記憶手段を備え、第２の電子機器が、自分自身に固有の識別番号を記憶する第２の記憶手段と、記憶している識別番号を第１の電子機器に送信する第１の送信手段とを備え、第１の電子機器が、さらに、第２の電子機器から送信されてくる、識別番号を受信する第１の受信手段と、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出手段と、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する第１の乱数生成手段と、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成する第１の送信値生成手段と、生成した第１の送信値を第２の電子機器に送信する第２の送信手段とを備え、第２の電子機器は、さらに、第１の電子機器から送信されてくる、第１の送信値を受信する第２の受信手段と、受信した第１の送信値に基づいて、第２の乱数を生成する第２の乱数生成手段と、あらかじめ定められたビット数の第３の乱数を生成する第３の乱数生成手段と、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成する第２の送信値生成手段と、生成した第２の送信値を第１の電子機器に送信する第３の送信手段とを備え、第１の電子機器は、さらに、第２の電子機器から送信されてくる、第２の送信値を受信する第３の受信手段と、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、第２の電子機器が正当であるか否かを認証する認証手段とを備え、認証手段は、比較の結果、第２の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００１６】

請求項４に記載の電子機器は、所定の処理を施す情報に対応する第１の鍵を記憶する記憶手段と、他の電子機器に付与された固有の識別番号を受信する第１の受信手段と、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出手段と、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成手段と、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成する送信値生成手段と、生成した第１の送信値を他の電子機器に送信する第１の送信手段と、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値を受信する第２の受信手段と、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証する認証手段とを備え、認証手段は、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００１７】

請求項５に記載の認証方法は、他の電子機器に付与された固有の識別番号の受信を制御する第１の受信制御ステップと、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出ステップと、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成ステップと、算出したハッ

シュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成する送信値生成ステップと、生成した第１の送信値の他の電子機器への送信を制御する第１の送信制御ステップと、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値の受信を制御する第２の受信制御ステップと、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証する認証ステップとを含み、認証ステップは、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００１８】

請求項６に記載の記録媒体に記録されたプログラムは、他の電子機器に付与された固有の識別番号の受信を制御する第１の受信制御ステップと、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出する算出ステップと、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成する乱数生成ステップと、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成する送信値生成ステップと、生成した第１の送信値の他の電子機器への送信を制御する第１の送信制御ステップと、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値の受信を制御する第２の受信制御ステップと、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証する認証ステップとを含み、認証ステップは、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００１９】

請求項７に記載の電子機器は、自分自身に固有の識別番号を記憶する記憶手段と、記憶している識別番号を他の電子機器に送信する第１の送信手段と、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値を受信する第１の受信手段と、受信した第１の送信値に基づいて、第２の乱数を生成する第１の乱数生成手段と、あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成手段と、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成する送信値生成手段と、生成した第２の送信値を他の電子機器に送信する第２の送信手段とを備えることを特徴とする。

【００２０】

請求項１０に記載の認証方法は、記憶している自分自身に固有の識別番号の他の電子機器への送信を制御する第１の送信制御ステップと、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値の受信を制御する第１の受信制御ステップと、受信した第１の送信値に基づいて、第２の乱数を生成する第１の乱数生成ステップと、あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成ステップと、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成する送信値生成ステップと、生成した第２の送信値の他の電子機器への送信を制御する第２の送信制御ステップとを含むことを特徴とする。

【００２１】

請求項１１に記載の記録媒体に記録されたプログラムは、記憶している自分自身に固有の識別番号の他の電子機器への送信を制御する第１の送信制御ステップと、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値の受信を制御する第１の受信制御ステップと、受信した第１の送信値に基づいて、第２の乱数を生成する第１の乱数生成ステップと、あらかじめ定められたビット数の第３の乱数を生成する第２の乱数生成ステップと、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成する送信値生成ステップと、生成した第２の送信値の他の電子機

器への送信を制御する第２の送信制御ステップとを含むことを特徴とする。

【００３１】

請求項１に記載の認証システムにおいては、第１の電子機器では、所定の処理を施す情報に対応する第１の鍵が記憶され、第２の電子機器では、自分自身に固有の識別番号が記憶され、記憶している識別番号が第１の電子機器に送信され、第１の電子機器では、第２の電子機器から送信されてくる、識別番号が受信され、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値が算出され、あらかじめ定められたビット数の第１の乱数および第２の乱数が生成され、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値が生成され、生成した第１の送信値が第２の電子機器に送信され、第２の電子機器では、第１の電子機器から送信されてくる、第１の送信値が受信され、受信した第１の送信値に基づいて、第２の乱数が生成され、あらかじめ定められたビット数の第３の乱数が生成され、生成した第２の乱数および第３の乱数に基づいて、第２の送信値が生成され、生成した第２の送信値が第１の電子機器に送信され、第１の電子機器では、第２の電子機器から送信されてくる、第２の送信値が受信され、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、第２の電子機器が正当であるか否かが認証され、比較の結果、第２の電子機器が正当でないと認証された場合、認証の処理が終了される。

【００３２】

請求項４に記載の電子機器、請求項５に記載の認証方法、および請求項６に記載の記録媒体においては、所定の処理を施す情報に対応する第１の鍵が記憶され、他の電子機器に付与された固有の識別番号が受信され、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値が算出され、あらかじめ定められたビット数の第１の乱数および第２の乱数が生成され、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値が生成され、生成した第１の送信値が他の電子機器に送信され、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値が受信され、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かが認証され、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理が終了される。

【００３３】

請求項７に記載の電子機器、請求項１０に記載の認証方法、および請求項１１に記載の記録媒体においては、自分自身に固有の識別番号が記憶され、記憶している識別番号が他の電子機器に送信され、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値が受信され、受信した第１の送信値に基づいて、第２の乱数が生成され、あらかじめ定められたビット数の第３の乱数が生成され、生成した第２の乱数および第３の乱数に基づいて、第２の送信値が生成され、生成した第２の送信値が他の電子機器に送信される。

【００３７】

【発明の実施の形態】

以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【００３８】

請求項１に記載の認証システムは、第１の電子機器（例えば、図１のDVDプレーヤ１（ソース））が、所定の処理を施す情報に対応する第１の鍵（例えば、service_key）を記憶する第１の記憶手段（例えば、図２のEEPROM２７）を備え、第２の電子機器（例えば、図１のパーソナルコンピュータ２（シンク））が、自分自身に固有の識別番号を記憶する第２の記憶手段（例えば、図２のEEPROM５０）と、記憶している識別番号を第１の電子機

器に送信する第１の送信手段（例えば、図３７のステップＳ２８３の処理）とを備え、第１の電子機器は、第２の電子機器から送信されてくる、識別番号を受信する第１の受信手段（例えば、図３７のステップＳ２８４の処理）と、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値（例えば、キーlk）を算出する算出手段（例えば、図３７のステップＳ２８５の処理）と、あらかじめ定められたビット数の第１の乱数（例えば、乱数r1）および第２の乱数（例えば、乱数r2）を生成する第１の乱数生成手段（例えば、図３７のステップＳ２８６の処理）と、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値（例えば、送信値X）を生成する第１の送信値生成手段（例えば、図３７のステップＳ２８７の処理）と、生成した第１の送信値を第２の電子機器に送信する第２の送信手段（例えば、図３７のステップＳ２８８の処理）とを備え、第２の電子機器が、第１の電子機器から送信されてくる、第１の送信値を受信する第２の受信手段（例えば、図３７のステップＳ２８９の処理）と、受信した第１の送信値に基づいて、第２の乱数を生成する第２の乱数生成手段（例えば、図３７のステップＳ２９０の処理）と、あらかじめ定められたビット数の第３の乱数（例えば、乱数r3）を生成する第３の乱数生成手段（例えば、図３７のステップＳ２９１の処理）と、生成した第２の乱数および第３の乱数に基づいて、第２の送信値（例えば、送信値Y）を生成する第２の送信値生成手段（例えば、図３７のステップＳ２９３の処理）と、生成した第２の送信値を第１の電子機器に送信する第３の送信手段（例えば、図３７のステップＳ２９４の処理）とを備え、第１の電子機器が、第２の電子機器から送信されてくる、第２の送信値を受信する第３の受信手段（例えば、図３７のステップＳ２９５の処理）と、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、第２の電子機器が正当であるか否かを認証する認証手段（例えば、図３７のステップＳ２９７の処理）とを備え、認証手段は、比較の結果、第２の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００３９】

請求項２に記載の認証システムは、第２の記憶手段は、第１の電子機器から送信されてくる所定の情報に所定の処理を施すことに対する許可に対応する第２の鍵（例えば、license_key）を記憶し、第２の送信値生成手段は、第２の乱数および第３の乱数を結合した値に対して、記憶している第２の鍵を用いて暗号的処理を行うことにより、第２の送信値を生成することを特徴とする。

【００４０】

請求項３に記載の認証システムは、第２の鍵は、第１の鍵および識別番号に対して、ハッシュ関数を適用して得られるハッシュ値と同一の値であることを特徴とする。

【００４７】

請求項４に記載の電子機器（例えば、図１のDVDプレーヤ１（ソース））は、所定の処理を施す情報に対応する第１の鍵（例えば、service_key）を記憶する記憶手段（例えば、図２のEEPROM２７）と、他の電子機器（例えば、図１のパーソナルコンピュータ２（シンク））に付与された固有の識別番号（例えば、ID）を受信する第１の受信手段（例えば、図３７のステップＳ２８４の処理）と、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値（例えば、キーlk）を算出する算出手段（例えば、図３７のステップＳ２８５の処理）と、あらかじめ定められたビット数の第１の乱数（例えば、乱数r1）および第２の乱数（例えば、乱数r2）を生成する乱数生成手段（例えば、図３７のステップＳ２８６の処理）と、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値（例えば、送信値X）を生成する送信値生成手段（例えば、図３７のステップＳ２８７の処理）と、生成した第１の送信値を他の電子機器に送信する第１の送信手段（例えば、図３７のステップＳ２８８の処理）と、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値（例えば、送信値Y）を受信する第２の受信手段と（例えば、図３７のステップＳ２９５の処理）、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証する認証手段（例え

ば、図３７のステップＳ２９７の処理）とを備え、認証手段は、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００４８】

請求項５に記載の認証方法は、他の電子機器（例えば、図１のパーソナルコンピュータ２（シンク））に付与された固有の識別番号（例えば、ID）の受信を制御する第１の受信制御ステップ（例えば、図３７のステップＳ２８４の処理）と、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値（例えば、キーlk）を算出する算出ステップ（例えば、図３７のステップＳ２８５の処理）と、あらかじめ定められたビット数の第１の乱数（例えば、乱数r1）および第２の乱数（例えば、乱数r2）を生成する乱数生成ステップ（例えば、図３７のステップＳ２８６の処理）と、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値（例えば、送信値X）を生成する送信値生成ステップ（例えば、図３７のステップＳ２８７の処理）と、生成した第１の送信値の他の電子機器への送信を制御する第１の送信制御ステップ（例えば、図３７のステップＳ２８８の処理）と、他の電子機器から送信されてくる、第２の乱数および第３の乱数（例えば、乱数r3）に基づいて生成された第２の送信値（送信値Y）の受信を制御する第２の受信制御ステップ（例えば、図３７のステップＳ２９５の処理）と、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証する認証ステップ（例えば、図３７のステップＳ２９７の処理）とを含み、認証ステップは、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了することを特徴とする。

【００４９】

請求項７に記載の電子機器（例えば、図１のパーソナルコンピュータ２（シンク））は、自分自身に固有の識別番号（例えば、ID）を記憶する記憶手段（例えば、図２のEEPROM５０）と、記憶している識別番号を他の電子機器（例えば、図１のDVDプレーヤ１（ソース））に送信する第１の送信手段（例えば、図３７のステップＳ２８３の処理）と、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵（例えば、service_key）および識別番号から算出されたハッシュ値（例えば、キーlk）、並びに第１の乱数（例えば、乱数r1）および第２の乱数（例えば、乱数r2）に基づいて生成された第１の送信値（例えば、送信値X）を受信する第１の受信手段（例えば、図３７のステップＳ２８９の処理）と、受信した第１の送信値に基づいて、第２の乱数を生成する第１の乱数生成手段（例えば、図３７のステップＳ２９０の処理）と、あらかじめ定められたビット数の第３の乱数（例えば、乱数r3）を生成する第２の乱数生成手段（例えば、図３７のステップＳ２９１の処理）と、生成した第２の乱数および第３の乱数に基づいて、第２の送信値（例えば、送信値Y）を生成する送信値生成手段（例えば、図３７のステップＳ２９３の処理）と、生成した第２の送信値を他の電子機器に送信する第２の送信手段（例えば、図３７のステップＳ２９４の処理）とを備えることを特徴とする。

【００５０】

請求項８に記載の電子機器は、記憶手段は、他の電子機器から送信されてくる所定の情報に所定の処理を施すことに対する許可に対応する第２の鍵（例えば、license_key）を記憶し、送信値生成手段は、第２の乱数および第３の乱数を結合した値に対して、記憶している第２の鍵を用いて暗号的処理を行うことにより、第２の送信値を生成することを特徴とする。

【００５１】

請求項９に記載の電子機器は、第２の鍵は、第１の鍵および識別番号に対して、ハッシュ関数を適用して得られるハッシュ値と同一の値であることを特徴とする。

【００５２】

請求項１０に記載の認証方法は、記憶している識別番号（例えば、ID）の他の電子機器への送信を制御する第１の送信制御ステップ（例えば、図３７のステップＳ２８３の処理）と、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵（例えば、service_key）および識別番号から算出されたハッシュ値（例えば、キーlk）、並

びに第１の乱数（例えば、乱数r1）および第２の乱数（例えば、乱数r2）に基づいて生成された第１の送信値（例えば、送信値X）の受信を制御する第１の受信制御ステップ（例えば、図３７のステップＳ２８９の処理）と、受信した第１の送信値に基づいて、第２の乱数を生成する第１の乱数生成ステップ（例えば、図３７のステップＳ２９０の処理）と、あらかじめ定められたビット数の第３の乱数（例えば、乱数r3）を生成する第２の乱数生成ステップ（例えば、図３７のステップＳ２９１の処理）と、生成した第２の乱数および第３の乱数に基づいて、第２の送信値（例えば、送信値Y）を生成する送信値生成ステップ（例えば、図３７のステップＳ２９３の処理）と、生成した第２の送信値の他の電子機器への送信を制御する第２の送信制御ステップ（例えば、図３７のステップＳ２９４の処理）とを含むことを特徴とする。

【００５９】

図１は、本発明を適用した情報処理システムの構成例を表している。この構成例においては、IEEE１３９４シリアルバス１１を介してDVDプレーヤ１、パーソナルコンピュータ２、光磁気ディスク装置３、データ放送受信装置４、モニタ５、テレビジョン受像機６が相互に接続されている。

【００６０】

図２は、この内のDVDプレーヤ１、パーソナルコンピュータ２、および光磁気ディスク装置３の内部のより詳細な構成例を表している。DVDプレーヤ１は、１３９４インタフェース２６を介して、１３９４バス１１に接続されている。CPU２１は、ROM２２に記憶されているプログラムに従って各種の処理を実行し、RAM２３は、CPU２１が各種の処理を実行する上において必要なデータやプログラムなどを適宜記憶する。操作部２４は、ボタン、スイッチ、リモートコントローラなどにより構成され、ユーザにより操作されたとき、その操作に対応する信号を出力する。ドライブ２５は、図示せぬDVD（ディスク）を駆動し、そこに記録されているデータを再生するようになされている。EEPROM２７は、装置の電源オフ後も記憶する必要のある情報（この実施の形態の場合、鍵情報）を記憶するようになされている。内部バス２８は、これらの各部を相互に接続している。

【００６１】

光磁気ディスク装置３は、CPU３１乃至内部バス３８を有している。これらは、上述したDVDプレーヤ１におけるCPU２１乃至内部バス２８と同様の機能を有するものであり、その説明は省略する。ただし、ドライブ３５は、図示せぬ光磁気ディスクを駆動し、そこにデータを記録または再生するようになされている。

【００６２】

パーソナルコンピュータ２は、１３９４インタフェース４９を介して１３９４バス１１に接続されている。CPU４１は、ROM４２に記憶されているプログラムに従って各種の処理を実行する。RAM４３には、CPU４１が各種の処理を実行する上において必要なデータやプログラムなどが適宜記憶される。入出力インタフェース４４には、キーボード４５とマウス４６が接続されており、それらから入力された信号をCPU４１に出力するようになされている。また、入出力インタフェース４４には、ハードディスク（HDD）４７が接続されており、そこにデータ、プログラムなどを記録再生することができるようになされている。入出力インタフェース４４にはまた、拡張ボード４８を適宜装着し、必要な機能を付加することができるようになされている。EEPROM５０には、電源オフ後も保持する必要のある情報（この実施の形態の場合、各種の鍵情報）が記憶されるようになされている。例えば、PCI(Peripheral Component Interconnect)、ローカルバスなどにより構成される内部バス５１は、これらの各部を相互に接続するようになされている。

【００６３】

なお、この内部バス５１は、ユーザに対して解放されており、ユーザは、拡張ボード４８に所定のボードを適宜接続したり、所定のソフトウェアプログラムを作成して、CPU４１にインストールすることで、内部バス５１により伝送されるデータを適宜受信することができるようになされている。

【００６４】

これに対して、DVDプレーヤ１や光磁気ディスク装置３などのコンシューマエレクトロニクス（CE）装置においては、内部バス２８や内部バス３８は、ユーザに解放されておらず、特殊な改造などを行わない限り、そこに伝送されるデータを取得することができないようになされている。

【００６５】

次に、所定のソースとシンクとの間で行われる認証の処理について説明する。この認証の処理は、図３に示すように、ソースとしての、例えばDVDプレーヤ１のROM２２に予め記憶されているソフトウェアプログラムの１つとしてのファームウェア２０と、シンクとしての、例えばパーソナルコンピュータ２のROM４２に記憶されており、CPU４１が処理するソフトウェアプログラムの１つとしてのライセンスマネージャ６２との間において行われる。

【００６６】

図４は、ソース（DVDプレーヤ１）と、シンク（パーソナルコンピュータ２）との間において行われる認証の手順を示している。DVDプレーヤ１のEEPROM２７には、サービスキー（service_key）と関数（hash）が予め記憶されている。これらはいずれも著作権者から、このDVDプレーヤ１のユーザに与えられたものであり、各ユーザは、EEPROM２７に、これを秘密裡に保管しておくものである。

【００６７】

サービスキーは、著作権者が提供する情報毎に与えられるものであり、この１３９４バス１１で構成されるシステムにおいて、共通のものである。なお、本明細書において、システムとは、複数の装置で構成される全体的な装置を示すものとする。

【００６８】

hash関数は、任意長の入力に対して、６４ビットまたは１２８ビットなどの固定長のデータを出力する関数であり、ｙ（＝hash（ｘ））を与えられたとき、ｘを求めることが困難であり、かつ、hash（ｘ１）＝hash（ｘ２）となるｘ１と、ｘ２の組を求めることも困難となる関数である。１方向hash関数の代表的なものとして、MD５やSHAなどが知られている。この１方向hash関数については、Bruce Schneier著の「Applied Cryptography(Second Edition),Wiley」に詳しく解説されている。

【００６９】

一方、シンクとしての例えばパーソナルコンピュータ２は、著作権者から与えられた、自分自身に固有の識別番号（ID）とライセンスキー（license_key）をEEPROM５０に秘密裡に保持している。このライセンスキーは、ｎビットのIDとｍビットのサービスキーを連結して得たｎ＋ｍビットのデータ（ID || service_key）に対して、hash関数を適用して得られる値である。すなわち、ライセンスキーは次式で表される。

licence_key=hash(ID || service_key)

【００７０】

IDとしては、例えば１３９４バス１１の規格に定められているnode_unique_IDを用いることができる。このnode_unique_IDは、図５に示すように、８バイト（６４ビット）で構成され、最初の３バイトは、IEEEで管理され、電子機器の各メーカーにIEEEから付与される。また、下位５バイトは、各メーカーが、自分自身がユーザに提供する各装置に対して付与することができるものである。各メーカーは、例えば下位５バイトに対してシリアルに、１台に１個の番号を割り当てるようにし、５バイト分を全部使用した場合には、上位３バイトがさらに別の番号となっているnode_unique_IDの付与を受け、そして、その下位５バイトについて１台に１個の番号を割り当てるようにする。従って、このnode_unique_IDは、メーカーに拘らず、１台毎に異なるものとなり、各装置に固有のものとなる。

【００７１】

ステップＳ１において、DVDプレーヤ１のファームウェア２０は、１３９４インタフェース２６を制御し、１３９４バス１１を介してパーソナルコンピュータ２に対してIDを要求する。パーソナルコンピュータ２のライセンスマネージャ６２は、ステップＳ２において、このIDの要求を受信する。すなわち、１３９４インタフェース４９は、１３９４バス１

１を介してDVDプレーヤ１から伝送されてきたID要求の信号を受信すると、これをCPU４１に出力する。CPU４１のライセンスマネージャ６２は、このID要求を受けたとき、ステップＳ３においてEEPROM５０に記憶されているIDを読み出し、これを１３９４インタフェース４９を介して１３９４バス１１からDVDプレーヤ１に伝送する。

【００７２】

DVDプレーヤ１においては、ステップＳ４で１３９４インタフェース２６が、このIDを受け取ると、このIDがCPU２１で動作しているファームウェア２０に供給される。

【００７３】

ファームウェア２０は、ステップＳ５において、パーソナルコンピュータ２から伝送を受けたIDと、EEPROM２７に記憶されているサービスキーを連結して、連結データ（ID || service_key）を生成し、このデータに対して、次式に示すようにhash関数を適用して、キーlkを生成する。

lk＝hash（ID || service_key）

【００７４】

次に、ステップＳ６において、ファームウェア２０は、暗号鍵skを生成する。この暗号鍵skの詳細については後述するが、この暗号鍵skは、セッションキーとしてDVDプレーヤ１とパーソナルコンピュータ２のそれぞれにおいて利用される。

【００７５】

次に、ステップＳ７において、ファームウェア２０は、ステップＳ５で生成した鍵lkを鍵として、ステップＳ６で生成した暗号鍵skを暗号化して、暗号化データ（暗号化鍵）ｅを得る。すなわち、次式を演算する。

ｅ＝Enc（lk,　sk）

【００７６】

なお、Enc（Ａ，Ｂ）は、共通鍵暗号方式で、鍵Ａを用いて、データＢを暗号化することを意味する。

【００７７】

次に、ステップＳ８で、ファームウェア２０は、ステップＳ７で生成した暗号化データｅをパーソナルコンピュータ２に伝送する。すなわち、この暗号化データｅは、DVDプレーヤ１の１３９４インタフェース２６から１３９４バス１１を介してパーソナルコンピュータ２に伝送される。パーソナルコンピュータ２においては、ステップＳ９で、この暗号化データｅを１３９４インタフェース４９を介して受信する。ライセンスマネージャ６２は、このようにして受信した暗号化データｅをEEPROM５０に記憶されているライセンスキーを鍵として、次式に示すように復号し、復号鍵sk'を生成する。

sk'＝Dec（license_key, ｅ）

【００７８】

なお、ここで、Dec（Ａ，Ｂ）は、共通鍵暗号方式で鍵Ａを用いて、データＢを復号することを意味する。

【００７９】

なお、この共通鍵暗号方式における暗号化のアルゴリズムとしては、ＤＥＳが知られている。共通鍵暗号化方式についても、上述した、Applied Cryptography(Second Edition)に詳しく解説されている。

【００８０】

DVDプレーヤ１において、ステップＳ５で生成するキーlkは、パーソナルコンピュータ２のEEPROM５０に記憶されている（license_key）と同一の値となる。すなわち、次式が成立する。

lk＝license_key

【００８１】

従って、パーソナルコンピュータ２において、ステップＳ１０で復号して得たキーsk'は、DVDプレーヤ１において、ステップＳ６で生成した暗号鍵skと同一の値となる。すなわち、次式が成立する。

sk'＝sk

【００８２】

このように、DVDプレーヤ１（ソース）とパーソナルコンピュータ２（シンク）の両方において、同一の鍵sk，sk'を共有することができる。そこで、この鍵skをそのまま暗号鍵として用いるか、あるいは、これを基にして、それぞれが疑似乱数を作り出し、それを暗号鍵として用いることができる。

【００８３】

ライセンスキーは、上述したように、各装置に固有のIDと、提供する情報に対応するサービスキーに基づいて生成されているので、他の装置がskまたはsk'を生成することはできない。また、著作権者から認められていない装置は、ライセンスキーを有していないので、skあるいはsk'を生成することができない。従って、その後DVDプレーヤ１が暗号鍵skを用いて再生データを暗号化してパーソナルコンピュータ２に伝送した場合、パーソナルコンピュータ２が適正にライセンスキーを得たものである場合には、暗号鍵sk'を有しているので、DVDプレーヤ１より伝送されてきた、暗号化されている再生データを復号することができる。しかしながら、パーソナルコンピュータ２が適正なものでない場合、暗号鍵sk'を有していないので、伝送されてきた暗号化されている再生データを復号することができない。換言すれば、適正な装置だけが共通の暗号鍵sk，sk'を生成することができるので、結果的に、認証が行われることになる。

【００８４】

仮に１台のパーソナルコンピュータ２のライセンスキーが盗まれたとしても、IDが１台１台異なるので、そのライセンスキーを用いて、他の装置がDVDプレーヤ１から伝送されてきた暗号化されているデータを復号することはできない。従って、安全性が向上する。

【００８５】

ところで、何らかの理由により、不正なユーザが、暗号化データｅと暗号鍵skを両方とも知ってしまったような場合のことを考える。この場合、ｅは、平文skを、鍵lkで暗号化した暗号文であるので、暗号アルゴリズムが公開されている場合、不正ユーザは、鍵lkを総当たりで試すことにより、正しい鍵lkを得る可能性がある。

【００８６】

不正ユーザによるこの種の攻撃を、より困難にするためには、暗号アルゴリズムの一部または全部を一般に公開せずに秘密にしておくことができる。

【００８７】

または同様に、license_keyから、service_keyを総当たりで調べる攻撃を、より困難にするために、hash関数の一部または全文を一般に公開せずに秘密にしておくようにすることもできる。

【００８８】

図６は、ソース（DVDプレーヤ１）に対して、パーソナルコンピュータ２だけでなく、光磁気ディスク装置３もシンクとして機能する場合の処理例を表している。

【００８９】

この場合、シンク１としてのパーソナルコンピュータ２のEEPROM５０には、IDとしてID１が、また、ライセンスキーとしてlicense_key１が記憶されており、シンク２としての光磁気ディスク装置３においては、EEPROM３７に、IDとしてID２が、また、ライセンスキーとしてlicense_key２が記憶されている。

【００９０】

DVDプレーヤ１（ソース）とパーソナルコンピュータ２（シンク１）の間において行われるステップＳ１１乃至ステップＳ２０の処理は、図４におけるステップＳ１乃至ステップＳ１０の処理と実質的に同様の処理であるので、その説明は省略する。

【００９１】

すなわち、上述したようにして、DVDプレーヤ１は、パーソナルコンピュータ２に対して認証処理を行う。そして次に、ステップＳ２１において、DVDプレーヤ１は、光磁気ディスク装置３に対して、IDを要求する。光磁気ディスク装置３においては、ステップＳ２２

で１３９４インタフェース３６を介して、このID要求信号が受信されると、そのファームウェア３０（図１０）は、ステップＳ２３でEEPROM３７に記憶されているID（ID２）を読み出し、これを１３９４インタフェース３６から、１３９４バス１１を介してDVDプレーヤ１に伝送する。DVDプレーヤ１のファームウェア２０は、ステップＳ２４で、１３９４インタフェース２６を介して、このID２を受け取ると、ステップＳ２５で、次式から鍵lk２を生成する。

lk２＝hash（ID２ || service_key）

【００９２】

さらに、ファームウェア２０は、ステップＳ２６で次式を演算し、ステップＳ１６で生成した鍵skを、ステップＳ２５で生成した鍵lk２を用いて暗号化し、暗号化したデータｅ２を生成する。

ｅ２＝Enc（lk２，sk）

【００９３】

そして、ステップＳ２７で、ファームウェア２０は、この暗号化データｅ２を１３９４インタフェース２６から１３９４バス１１を介して光磁気ディスク装置３に伝送する。

【００９４】

光磁気ディスク装置３においては、ステップＳ２８で１３９４インタフェース３６を介して、この暗号化データｅ２を受信し、ステップＳ２９で次式を演算して、暗号鍵sk２’を生成する。

sk２’＝Dec（license_key２，ｅ２）

【００９５】

以上のようにして、パーソナルコンピュータ２と光磁気ディスク装置３のそれぞれにおいて、暗号鍵sk１’，sk２’が得られたことになる。これらの値は、DVDプレーヤ１における暗号鍵skと同一の値となっている。

【００９６】

図６の処理例においては、DVDプレーヤ１が、パーソナルコンピュータ２と、光磁気ディスク装置３に対して、それぞれ個別にIDを要求し、処理するようにしているのであるが、同報通信によりIDを要求することができる場合は、図７に示すような処理を行うことができる。

【００９７】

すなわち、図７の処理例においては、ステップＳ４１で、ソースとしてのDVDプレーヤ１が、全てのシンク（この例の場合、パーソナルコンピュータ２と光磁気ディスク装置３）に対して同報通信でIDを要求する。パーソナルコンピュータ２と光磁気ディスク装置３は、それぞれステップＳ４２とステップＳ４３で、このID転送要求の信号を受け取ると、それぞれステップＳ４４またはステップＳ４５で、EEPROM５０またはEEPROM３７に記憶されているID１またはID２を読み出し、これをDVDプレーヤ１に転送する。DVDプレーヤ１は、ステップＳ４６とステップＳ４７で、これらのIDをそれぞれ受信する。

【００９８】

DVDプレーヤ１においては、さらにステップＳ４８で、次式から暗号鍵lk１を生成する。

lk１＝hash（ID１ || service_key）

【００９９】

さらに、ステップＳ４９において、次式から暗号鍵lk２が生成される。

lk２＝hash（ID２ || service_key）

【０１００】

DVDプレーヤ１においては、さらにステップＳ５０で、暗号鍵skが生成され、ステップＳ５１で、次式で示すように、暗号鍵skが、鍵lk１を鍵として暗号化される。

ｅ１＝Enc（lk１，sk）

【０１０１】

さらに、ステップＳ５２においては、暗号鍵skが、鍵lk２を鍵として、次式に従って暗号化される。

ｅ２＝Enc（lk２，sk）
【０１０２】
さらに、ステップＳ５３においては、ID１，ｅ１，ID２，ｅ２が、それぞれ次式で示すように連結されて、暗号化データｅが生成される。
ｅ＝ID１ || ｅ１ || ID２ || ｅ２
【０１０３】
DVDプレーヤ１においては、さらにステップＳ５４で、以上のようにして生成された暗号化データｅが同報通信で、パーソナルコンピュータ２と光磁気ディスク装置３に伝送される。
【０１０４】
パーソナルコンピュータ２と光磁気ディスク装置３においては、それぞれステップＳ５５またはステップＳ５６で、これらの暗号化データｅが受信される。そして、パーソナルコンピュータ２と光磁気ディスク装置３においては、それぞれステップＳ５７またはステップＳ５８において、次式で示す演算が行われ、暗号鍵sk１’，sk２’が生成される。
sk１’＝Dec（license_key１，ｅ１）
sk２’＝Dec（license_key２，ｅ２）
【０１０５】
図８は、１つのシンクが複数のサービスを受けること（複数の種類の情報の復号）ができるようになされている場合の処理例を表している。すなわち、この場合においては、例えば、シンクとしてのパーソナルコンピュータ２は、複数のライセンスキー（license_key１，license_key２，license_key３など）をEEPROM５０に記憶している。ソースとしてのDVDプレーヤ１は、そのEEPROM２７に複数のサービスキー（service_key１，service_key２，service_key３など）を記憶している。この場合、DVDプレーヤ１は、ステップＳ８１でシンクとしてのパーソナルコンピュータ２に対してIDを要求するとき、DVDプレーヤ１が、これから転送しようとする情報（サービス）を識別するservice_IDを転送する。パーソナルコンピュータ２においては、ステップＳ８２で、これを受信したとき、EEPROM５０に記憶されている複数のライセンスキーの中から、このservice_IDに対応するものを選択し、これを用いて、ステップＳ９０で復号処理を行う。その他の動作は、図４における場合と同様である。
【０１０６】
図９は、さらに他の処理例を表している。この例においては、ソースとしてのDVDプレーヤ１が、そのEEPROM２７に、service_key、hash関数、および疑似乱数発生関数pRNGを記憶している。これらは、著作権者から与えられたものであり、秘密裡に保管される。また、シンクとしてのパーソナルコンピュータ２のEEPROM５０には、著作権者から与えられたID、LK，LK'、関数Ｇ、および疑似乱数数発生関数pRNGを有している。
【０１０７】
LKは、著作権者が作成したユニークな乱数であり、LK'は、次式を満足するように生成されている。
LK'＝Ｇ＾－１（Ｒ）
Ｒ＝pRNG（Ｈ）　（＋）　pRNG（LK）
Ｈ＝hash（ID || service_key）
【０１０８】
なお、Ｇ＾－１（＾はべき乗を意味する）は、Ｇの逆関数を意味する。Ｇ＾－１は、所定の規則を知っていれば、簡単に計算することができるが、知らない場合には、計算することが難しいような特徴を有している。このような関数としては、公開鍵暗号に用いられている関数を利用することができる。
【０１０９】
また、疑似乱数発生関数は、ハードウェアとして設けるようにすることも可能である。
【０１１０】
DVDプレーヤ１のファームウェア２０は、最初にステップＳ１０１において、パーソナル

コンピュータ２のライセンスマネージャ６２に対してIDを要求する。パーソナルコンピュータ２のライセンスマネージャ６２は、ステップＳ１０２でID要求信号を受け取ると、EEPROM５０に記憶されているIDを読み出し、ステップＳ１０３で、これをDVDプレーヤ１に伝送する。DVDプレーヤ１のファームウェア２０は、ステップＳ１０４でこのIDを受け取ると、ステップＳ１０５で次式を演算する。

H＝hash（ID || service_key）

【０１１１】

さらに、ファームウェア２０は、ステップＳ１０６で鍵skを生成し、ステップＳ１０７で次式を演算する。

ｅ＝sk （＋） pRNG（H）

【０１１２】

なお、Ａ（＋）Ｂは、ＡとＢの排他的論理和の演算を意味する。

【０１１３】

すなわち、疑似ランダム発生キーpRNGにステップＳ１０５で求めたHを入力することで得られた結果、pRNG（H）と、ステップＳ１０６で生成した鍵skのビット毎の排他的論理和を演算することで、鍵SKを暗号化する。

【０１１４】

次に、ステップＳ１０８で、ファームウェア２０は、ｅをパーソナルコンピュータ２に伝送する。

【０１１５】

パーソナルコンピュータ２においては、ステップＳ１０９でこれを受信し、ステップＳ１１０で、次式を演算する。

sk'＝ｅ （＋） G（LK'） （＋） pRNG（LK）

【０１１６】

すなわち、DVDプレーヤ１から伝送されてきたｅ、EEPROM５０に記憶されている関数Ｇに、やはりEEPROM５０に記憶されているLK'を適用して得られる値Ｇ（LK'）、並びに、EEPROM５０に記憶されているLK'を、やはりEEPROM５０に記憶されている疑似乱数発生関数pRNGに適用して得られる結果pRNG（LK）の排他的論理和を演算し、鍵sk'を得る。

【０１１７】

ここで、次式に示すように、sk＝sk'となる。

$$
\begin{aligned}
sk' &= e \ (+)\ G(LK') \ (+)\ pRNG(LK) \\
&= sk \ (+)\ pRNG(H) \ (+)\ R \ (+)\ pRNG(LK) \\
&= sk \ (+)\ pRNG(H) \ (+)\ pRNG(H) \ (+)\ pRNG(LK) \ (+)\ pRNG(LK) \\
&= sk
\end{aligned}
$$

【０１１８】

このようにして、ソースとしてのDVDプレーヤ１とシンクとしてのパーソナルコンピュータ２は、同一の鍵sk，sk'を共有することができる。LK，LK'を作ることができるのは、著作権者だけであるので、ソースが不正に、LK，LK'を作ろうとしても作ることができないので、より安全性を高めることができる。

【０１１９】

以上においては、ソースとシンクにおいて認証を行うようにしたが、例えばパーソナルコンピュータ２には、通常、任意のアプリケーションプログラムをロードして用いることができる。そして、このアプリケーションプログラムとしては、不正に作成したものが使用される場合もある。従って、各アプリケーションプログラム毎に、著作権者から許可を得たものであるか否かを判定する必要がある。そこで、図３に示すように、各アプリケーション部６１とライセンスマネージャ６２との間においても、上述したように、認証処理を

行うようにすることができる。この場合、ライセンスマネージャ６２がソースとなり、アプリケーション部６１がシンクとなる。

【０１２０】

次に、以上のようにして、認証が行われた後（暗号鍵の共有が行われた後）、暗号鍵を用いて、ソースから暗号化したデータをシンクに転送し、シンクにおいて、この暗号化したデータを復号する場合の動作について説明する。

【０１２１】

図１０に示すように、DVDプレーヤ１、あるいは光磁気ディスク装置３のように、内部の機能が一般ユーザに解放されていない装置においては、１３９４バス１１を介して授受されるデータの暗号化と復号の処理は、それぞれ１３９４インタフェース２６または１３９４インタフェース３６で行われる。この暗号化と復号化には、セッションキーＳと時変キーｉが用いられるが、このセッションキーＳと時変キーｉ（正確には、時変キーｉを生成するためのキーｉ’）は、それぞれファームウェア２０またはファームウェア３０から、１３９４インタフェース２６または１３９４インタフェース３６に供給される。セッションキーＳは、初期値として用いられる初期値キーSsと時変キーｉを攪乱するために用いられる攪乱キーSiとにより構成されている。この初期値キーSsと攪乱キーSiは、上述した認証において生成された暗号鍵sk（＝sk'）の所定のビット数の上位ビットと下位ビットにより、それぞれ構成するようにすることができる。このセッションキーＳは、セッション毎に（例えば、１つの映画情報毎に、あるいは、１回の再生毎に）、適宜、更新される。これに対して、攪乱キーSiとキーｉ’から生成される時変キーｉは、１つのセッション内において、頻繁に更新されるキーであり、例えば、所定のタイミングにおける時刻情報などを用いることができる。

【０１２２】

いま、ソースとしてのDVDプレーヤ１から再生出力した映像データを１３９４バス１１を介して光磁気ディスク装置３とパーソナルコンピュータ２に伝送し、それぞれにおいて復号するものとする。この場合、DVDプレーヤ１においては、１３９４インタフェース２６において、セッションキーＳと時変キーｉを用いて暗号化処理が行われる。光磁気ディスク装置３においては、１３９４インタフェース３６において、セッションキーＳと時変キーｉを用いて復号処理が行われる。

【０１２３】

これに対して、パーソナルコンピュータ２においては、ライセンスマネージャ６２が、セッションキーＳのうち、初期値キーSsをアプリケーション部６１に供給し、攪乱キーSiと時変キーｉ（正確には、時変キーｉを生成するためのキーｉ’）を１３９４インタフェース４９（リンク部分）に供給する。そして、１３９４インタフェース４９において、攪乱キーSiとキーｉ’から時変キーｉが生成され、時変キーｉを用いて復号が行われ、その復号されたデータは、アプリケーション部６１において、さらにセッションキーＳ（正確には、初期値キーSs）を用いて復号が行われる。

【０１２４】

このように、パーソナルコンピュータ２においては、内部バス５１が、ユーザに解放されているので、１３９４インタフェース４９により第１段階の復号だけを行い、まだ暗号の状態としておく。そして、アプリケーション部６１において、さらに第２段階の復号を行い、平文にする。これにより、パーソナルコンピュータ２に対して、適宜、機能を付加して、内部バス５１において授受されるデータ（平文）をハードディスク４７や他の装置にコピーすることを禁止させる。

【０１２５】

このように、この発明の実施の形態においては、内部バスが解放されていないCE装置においては、暗号化、または復号処理は、セッションキーＳと時変キーｉを用いて１度に行われるが、内部バスが解放されている装置（パーソナルコンピュータ２など）においては、復号処理が、時変キーｉを用いた復号処理と、セッションキーＳを用いた復号処理に分けて行われる。このように、１段階の復号処理と、２段階に分けた復号処理の両方ができる

ようにするには、次式を成立させることが必要となる。
Dec（Ｓ，Dec（ｉ，　Enc（algo（Ｓ＋ｉ），Data））＝Data
【０１２６】
なお、上記式において、algo（Ｓ＋ｉ）は、所定のアルゴリズムにセッションキーＳと時変キーｉを入力して得られた結果を表している。
【０１２７】
図１１は、上記式を満足する１３９４インタフェース２６の構成例を表している。この構成例においては、アディティブジェネレータ７１により生成したｍビットのデータが、シュリンクジェネレータ７３に供給されている。また、LFSR(Linear Feedback Sift Register)７２が１ビットのデータを出力し、シュリンクジェネレータ７３に供給している。シュリンクジェネレータ７３は、LFSR７２の出力に対応して、アディティブジェネレータ７１の出力を選択し、選択したデータを暗号鍵として加算器７４に出力している。加算器７４は、入力された平文（１３９４バス１１に伝送するｍビットのデータ）と、シュリンクジェネレータ７３より供給されるｍビットのデータ（暗号鍵）とを加算し、加算した結果を暗号文（暗号化されたデータ）として、１３９４バス１１に出力するようになされている。
【０１２８】
加算器７４の加算処理は、mod ２＾ｍ（＾はべき乗を意味する）で、シュリンクジェネレータ７３の出力と平文を加算することを意味する。換言すれば、ｍビットのデータ同志が加算され、キャリオーバを無視した加算値が出力される。
【０１２９】
図１２は、図１１に示した１３９４インタフェース２６のさらにより詳細な構成例を表している。ファームウェア２０から出力されたセッションキーＳのうち、初期値キーSsは、加算器８１を介してレジスタ８２に転送され、保持される。この初期値キーSsは、例えば、５５ワード（１ワードは８ビット乃至３２ビットの幅を有する）により構成される。また、ファームウェア２０から供給されたセッションキーＳのうちの、例えばLSB側の３２ビットで構成される攪乱キーSiは、レジスタ８５に保持される。
【０１３０】
レジスタ８４には、キーｉ’が保持される。このキーｉ’は、例えば１３９４バス１１を介して１個のパケットが伝送される毎に、２ビットのキーｉ’がレジスタ８４に供給され、１６パケット分の（３２ビット分の）キーｉ’がレジスタ８４に保持されたとき、加算器８６により、レジスタ８５に保持されている３２ビットの攪乱キーSiと加算され、最終的な時変キーｉとして加算器８１に供給される。加算器８１は、そのときレジスタ８２に保持されている値と加算器８６より供給された時変キーｉを加算し、その加算結果をレジスタ８２に供給し、保持させる。
【０１３１】
レジスタ８２のワードのビット数が、例えば８ビットである場合、加算器８６より出力される時変キーｉが３２ビットであるので、時変キーｉを４分割して、各８ビットをレジスタ８２の所定のアドレス（０乃至５４）のワードに加算するようにする。
【０１３２】
このようにして、レジスタ８２には、最初に初期値キーSsが保持されるが、その後、この値は、１６パケット分の暗号文を伝送する毎に、時変キーｉで更新される。
【０１３３】
加算器８３は、レジスタ８２に保持されている５５ワードのうちの所定の２ワード（図１２に示されているタイミングの場合、アドレス２３とアドレス５４のワード）を選択し、その選択した２ワードを加算して、シュリンクジェネレータ７３に出力する。また、この加算器７３の出力は、図１２に示すタイミングでは、レジスタ８２のアドレス０に転送され、前の保持値に代えて保持される。
【０１３４】
そして、次のタイミングにおいては、加算器８３に供給されるレジスタ８２の２ワードの

アドレスは、アドレス５４とアドレス２３から、それぞれアドレス５３とアドレス２２に、１ワード分だけ、図中上方に移動され、加算器８３の出力で更新されるアドレスも、図中、より上方のアドレスに移動される。ただし、アドレス０より上方のアドレスは存在しないので、この場合には、アドレス５４に移動する。

【０１３５】

なお、加算器８１，８３，８６では、排他的論理和を演算させるようにすることも可能である。

【０１３６】

LFSR７２は、例えば、図１３に示すように、ｎビットのシフトレジスタ１０１と、シフトレジスタ１０１のｎビットのうちの所定のビット（レジスタ）の値を加算する加算器１０２により構成されている。シフトレジスタ１０１は、加算器１０２より供給されるビットを、図中最も左側のレジスタ$b_n$に保持すると、それまでそこに保持されていたデータを右側のレジスタ$b_{n-1}$にシフトする。レジスタ$b_{n-1}$，$b_{n-2}$，・・・も、同様の処理を行う。そして、さらに次のタイミングでは、各ビットの値を加算器１０２で加算した値を再び、図中最も左側のビット$b_n$に保持させる。以上の動作が順次繰り返されて、図中最も右側のレジスタ$b_1$から出力が１ビットずつ順次出力される。

【０１３７】

図１３は、一般的な構成例であるが、例えば、より具体的には、LFSR７２を図１４に示すように構成することができる。この構成例においては、シフトレジスタ１０１が３１ビットにより構成され、その図中右端のレジスタ$b_1$の値と左端のレジスタ$b_{31}$の値が、加算器１０２で加算され、加算された結果がレジスタ$b_{31}$に帰還されるようになされている。

【０１３８】

LFSR７２より出力された１ビットのデータが論理１であるとき、条件判定部９１は、アディティブジェネレータ７１の加算器８３より供給されたｍビットのデータをそのままＦＩＦＯ９２に転送し、保持させる。これに対して、LFSR７２より供給された１ビットのデータが論理０であるとき、条件判定部９１は、加算器８３より供給されたｍビットのデータを受け付けず、暗号化処理を中断させる。このようにして、シュリンクジェネレータ７３のＦＩＦＯ９２には、アディティブジェネレータ７１で生成したｍビットのデータのうち、LFSR７２が論理１を出力したタイミングのもののみが選択され、保持される。

【０１３９】

ＦＩＦＯ９２により保持したｍビットのデータが、暗号鍵として、加算器７４に供給され、伝送されるべき平文のデータ（DVDからの再生データ）に加算されて、暗号文が生成される。

【０１４０】

暗号化されたデータは、DVDプレーヤ１から１３９４バス１１を介して光磁気ディスク装置３とパーソナルコンピュータ２に供給される。

【０１４１】

光磁気ディスク装置３は、１３９４インタフェース３６において、１３９４バス１１から受信したデータを復号するために、図１５に示すような構成を有している。この構成例においては、シュリンクジェネレータ１７３にアディティブジェネレータ１７１の出力するｍビットのデータと、LFSR１７２が出力する１ビットのデータが供給されている。そして、シュリンクジェネレータ１７３の出力するｍビットの鍵が、減算器１７４に供給されている。減算器１７４は、暗号文からシュリンクジェネレータ１７３より供給される鍵を減算して、平文を復号する。

【０１４２】

すなわち、図１５に示す構成は、図１１に示す構成と基本的に同様の構成とされており、図１１における加算器７４が、減算器１７４に変更されている点だけが異なっている。

【０１４３】

図１６は、図１５に示す構成のより詳細な構成例を表している。この構成も、基本的に図１２に示した構成と同様の構成とされているが、図１２における加算器７４が、減算器１

７４に変更されている。その他のアディティブジェネレータ１７１、LFSR１７２、シュリンクジェネレータ１７３、加算器１８１、レジスタ１８２、加算器１８３、レジスタ１８４，１８５、加算器１８６、条件判定部１９１、ＦＩＦＯ１９２は、図１２におけるアディティブジェネレータ７１、LFSR７２、シュリンクジェネレータ７３、加算器８１、レジスタ８２、加算器８３、レジスタ８４，８５、加算器８６、条件判定部９１、およびＦＩＦＯ９２に対応している。

【０１４４】

従って、その動作は、基本的に図１２に示した場合と同様であるので、その説明は省略するが、図１６の例においては、シュリンクジェネレータ１７３のＦＩＦＯ１９２より出力されたｍビットの鍵が、減算器１７４において、暗号文から減算されて平文が復号される。

【０１４５】

以上のように、１３９４インタフェース３６においては、セッションキーＳ（初期値キーSsと攪乱キーSi）と時変キーｉを用いて、暗号化データが１度に復号される。

【０１４６】

これに対して、上述したように、パーソナルコンピュータ２においては、１３９４インタフェース４９とアプリケーション部６１において、それぞれ個別に、２段階に分けて復号が行われる。

【０１４７】

図１７は、１３９４インタフェース４９において、ハード的に復号を行う場合の構成例を表しており、その基本的構成は、図１５に示した場合と同様である。すなわち、この場合においても、アディティブジェネレータ２７１、LFSR２７２、シュリンクジェネレータ２７３、および減算器２７４により１３９４インタフェース４９が構成されており、これらは、図１５におけるアディティブジェネレータ１７１、LFSR１７２、シュリンクジェネレータ１７３、および減算器１７４と基本的に同様の構成とされている。ただし、図１７の構成例においては、アディティブジェネレータ２７１に対して、ライセンスマネージャ６２から、時変キーｉを生成するためのキーｉ’と、セッションキーＳのうち、時変キーｉを攪乱するための攪乱キーSiとしては、図１５における場合と同様のキーが供給されるが、初期値キーSsとしては、全てのビットが０である単位元が供給される。

【０１４８】

すなわち、図１８に示すように、初期値キーSsの全てのビットが０とされるので、実質的に、初期値キーSsが存在しない場合と同様に、時変キーｉだけに基づいて暗号鍵が生成される。その結果、減算器２７４においては、暗号文の時変キーｉに基づく復号だけが行われる。まだ初期値キーSsに基づく復号が行われていないので、この復号の結果得られるデータは、完全な平文とはなっておらず、暗号文の状態になっている。従って、このデータを内部バス５１から取り込み、ハードディスク４７や、その他の記録媒体に記録したとしても、それをそのまま利用することができない。

【０１４９】

そして、以上のようにして、１３９４インタフェース４９において、ハード的に時変キーｉに基づいて復号されたデータをソフト的に復号するアプリケーション部６１の構成は、図１９に示すように、アディティブジェネレータ３７１、LFSR３７２、シュリンクジェネレータ３７３および減算器３７４により構成される。その基本的構成は、図１５に示したアディティブジェネレータ１７１、LFSR１７２、シュリンクジェネレータ１７３、および減算器１７４と同様の構成となっている。

【０１５０】

ただし、セッションキーＳのうち、初期値キーSsは、図１５における場合と同様に、通常の初期値キーが供給されるが、時変キーｉを生成するための攪乱キーSiとキーｉ’は、それぞれ全てのビットが０である単位元のデータとされる。

【０１５１】

その結果、図２０にその詳細を示すように（そのアディティブジェネレータ３７１乃至FI

FIFO３９２は、図１６におけるアディティブジェネレータ１７１乃至FIFO１９２に対応している）、レジスタ３８４に保持されるキーｉ’とレジスタ３８５に保持される攪乱キーSiは、全てのビットが０であるため、加算器３８６の出力する時変キーｉも全てのビットが０となり、実質的に時変キーｉが存在しない場合と同様の動作が行われる。すなわち、初期値キーSsだけに基づく暗号鍵が生成される。そして、減算器３７４においては、このようにして生成された暗号鍵に基づいて暗号文が平文に復号される。上述したように、この暗号文は、１３９４インタフェース４９において、時変キーｉに基づいて第１段階の復号が行われているものであるので、ここで、初期値キーSsに基づいて第２段階の復号を行うことで、完全な平文を得ることができる。

【０１５２】

光磁気ディスク装置３においては、以上のようにして暗号文が復号されると、CPU３１が、復号されたデータをドライブ３５に供給し、光磁気ディスクに記録させる。

【０１５３】

一方、パーソナルコンピュータ２においては、CPU４１（アプリケーション部６１）が、以上のようにして復号されたデータを、例えばハードディスク４７に供給し、記録させる。パーソナルコンピュータ２においては、拡張ボード４８として所定のボードを接続して、内部バス５１で授受されるデータをモニタすることができるが、内部バス５１に伝送されるデータを最終的に復号することができるのは、アプリケーション部６１であるので、拡張ボード４８は、１３９４インタフェース４９で、時変キーｉに基づく復号が行われたデータ（まだ、セッションキーＳに基づく復号が行われていないデータ）をモニタすることができたとしても、完全に平文に戻されたデータをモニタすることはできない。そこで、不正なコピーが防止される。

【０１５４】

なお、セッションキーの共有は、例えば、Diffie-Hellman法などを用いて行うようにすることも可能である。

【０１５５】

なお、この他、例えばパーソナルコンピュータ２における１３９４インタフェース４９またはアプリケーション部６１の処理能力が比較的低く、復号処理を行うことができない場合には、セッションキーと時変キーのいずれか、あるいは両方をソース側において、単位元で構成するようにし、シンク側においても、これらを単位元で用いるようにすれば、実施的にセッションキーと時変キーを使用しないで、データの授受が可能となる。ただし、そのようにすれば、データが不正にコピーされるおそれが高くなる。

【０１５６】

アプリケーション部６１そのものが、不正にコピーしたものである場合、復号したデータが不正にコピーされてしまう恐れがあるが、上述したようにアプリケーション部６１をライセンスマネージャ６２で認証するようにすれば、これを防止することが可能である。

【０１５７】

この場合の認証方法としては、共通鍵暗号方式の他、公開鍵暗号方式を用いたデジタル署名を利用することができる。

【０１５８】

以上の図１１、図１２、図１５乃至図２０に示す構成は、準同形(homomorphism)の関係を満足するものとなっている。すなわち、キー$K_1$，$K_2$がガロアフィールドＧの要素であるとき、両者の群演算の結果、$K_1$・$K_2$もガロアフィールドＧの要素となる。そして、さらに、所定の関数Ｈについて次式が成立する。

$H(K_1 \cdot K_2) = H(K_1) \cdot H(K_2)$

【０１５９】

図２１は、さらに１３９４インタフェース２６の他の構成例を表している。この構成例においては、セッションキーＳがLFSR５０１乃至５０３に供給され、初期設定されるようになされている。LFSR５０１乃至５０３の幅$n_1$乃至$n_3$は、それぞれ２０ビット程度で、それぞれの幅$n_1$乃至$n_3$は、相互に素になるように構成される。従って、例えば、セッショ

ンキーＳのうち、例えば、上位ｎ１ビットがLFSR５０１に初期設定され、次の上位$n_2$ビットがLFSR５０２に初期設定され、さらに次の上位$n_3$ビットがLFSR５０３に初期設定される。

【０１６０】

LFSR５０１乃至５０３は、クロッキングファンクション５０６より、例えば論理１のイネーブル信号が入力されたとき、ｍビットだけシフト動作を行い、ｍビットのデータを出力する。ｍの値は、例えば、８，１６，３２，４０などとすることができる。

【０１６１】

LFSR５０１とLFSR５０２の出力は、加算器５０４に入力され、加算される。加算器５０４の加算値のうち、キャリー成分は、クロッキングファンクション５０６に供給され、sum成分は、加算器５０５に供給され、LFSR５０３の出力と加算される。加算器５０５のキャリー成分は、クロッキングファンクション５０６に供給され、sum成分は、排他的論理和回路５０８に供給される。

【０１６２】

クロッキングファンクション５０６は、加算器５０４と加算器５０５より供給されるデータの組み合わせが、００，０１，１０，１１のいずれかであるので、これらに対応して、LFSR５０１乃至５０３に対して、０００乃至１１１のいずれか１つの組み合わせのデータを出力する。LFSR５０１乃至５０３は、論理１が入力されたとき、ｍビットのシフト動作を行い、新たなｍビットのデータを出力し、論理０が入力されたとき、前回出力した場合と同一のｍビットのデータを出力する。

【０１６３】

排他的論理和回路５０８は、加算器５０５の出力するsum成分とレジスタ５０７に保持された時変キーｉの排他的論理和を演算し、その演算結果を排他的論理和回路５０９に出力する。排他的論理和回路５０９は、入力された平文と、排他的論理和回路５０８より入力された暗号鍵の排他的論理和を演算し、演算結果を暗号文として出力する。

【０１６４】

図２２は、光磁気ディスク装置３における１３９４インタフェース３６の構成例を表している。この構成例におけるLFSR６０１乃至排他的論理和回路６０９は、図２１におけるLFSR５０１乃至排他的論理和回路５０９と同様の構成とされている。従って、その動作も、基本的に同様となるので、その説明は省略する。ただし、図２１の構成例においては、暗号化処理が行われるのに対して、図２２の構成例においては、復号処理が行われる。

【０１６５】

図２３は、パーソナルコンピュータ２の１３９４インタフェース４９の構成例を表している。この構成例におけるLFSR７０１乃至排他的論理和回路７０９も、図２２における、LFSR６０１乃至排他的論理和回路６０９と同様の構成とされている。ただし、LFSR７０１乃至７０３に初期設定されるセッションキーＳは、全てのビットが０の単位元とされている。従って、この場合、実質的にレジスタ７０７に保持された時変キーｉだけに対応して復号化処理が行われる。

【０１６６】

図２４は、パーソナルコンピュータ２のアプリケーション部６１の構成例を表している。この構成例におけるLFSR８０１乃至排他的論理和回路８０９は、図２２における、LFSR６０１乃至排他的論理和回路６０９と基本的に同様の構成とされている。ただし、レジスタ８０７に入力される時変キーｉが、全てのビットが０である単位元とされている点のみが異なっている。従って、この構成例の場合、セッションキーＳだけに基づいて暗号鍵が生成され、復号処理が行われる。

【０１６７】

なお、図１９、図２０、および図２４に示す処理は、アプリケーション部６１において行われるので、ソフト的に処理されるものである。

【０１６８】

ところで、何らかの理由でlicense_keyが盗まれてしまったような場合には、適宜、これ

を変更（更新）するようにすることができる。勿論、このlicense_keyが実際に盗まれなくても、盗まれるおそれがある場合には、所定の周期で、これを更新するようにすることができる。この場合、例えば、DVD（ディスク）内に、そのとき有効とされるlicense_keyのバージョン（この実施の形態の場合、hash関数の適用回数）が記録される。また、対象となる操作が、DVDプレーヤではなく、例えば衛星を介して伝送されてくる情報を受信する受信装置である場合には、衛星から、そのバージョンの情報が受信装置に向けて伝送される。

【０１６９】

図２５と図２６は、DVDプレーヤにおいて、license_keyを更新する場合の処理例を表している。なお、この実施の形態の場合には、図４に示した情報が、DVDプレーヤ１のEEPROM２７とパーソナルコンピュータ２のEEPROM５０に記憶されている他、EEPROM５０にはhash関数も記憶されている。

【０１７０】

最初に、ステップＳ１５１において、ソースとしてのDVDプレーヤ１は、シンクとしてのパーソナルコンピュータ２に対して、ＩＤを要求する。パーソナルコンピュータ２は、ステップＳ１５２で、このＩＤ要求信号を受け取ると、ステップＳ１５３で、自分自身のＩＤをDVDプレーヤ１に送出する。DVDプレーヤ１は、ステップＳ１５４で、このＩＤを受信する。

【０１７１】

次に、DVDプレーヤ１は、ステップＳ１５５で、次式から鍵lkを演算する。

lk＝hash（ＩＤ　||　service_key）

【０１７２】

以上の処理は、図４のステップＳ１乃至Ｓ５の処理と同様の処理である。

【０１７３】

次に、ステップＳ１５６に進み、DVDプレーヤ１は、ステップＳ１５５で演算した鍵lkが有効なバージョンのものであるか否かを判定する。すなわち、上述したように、DVDには、現在有効なlicense_key（＝lk）のバージョン（hash関数の適用回数）が記録されている。ステップＳ１５５で生成した鍵lkは、hash関数を１回適用して求めたものである。このhash関数の適用回数がバージョンで規定されている回数と等しくない場合、鍵lkは無効と判定される。この場合、ステップＳ１５７に進み、DVDプレーヤ１は、更新回数（演算回数）を示す変数ｇに１を初期設定し、$lk_g$にlkを設定する。そして、ステップＳ１５８において、現在の鍵$lk_g$にhash関数を１回適用し、新たな鍵$lk_{g+1}$を演算する。すなわち、次式を演算する。

$lk_{g+1}$＝hash（$lk_g$）

【０１７４】

ステップＳ１５９では、ステップＳ１５８で求められた鍵$lk_{g+1}$が有効であるか否かを判定する。すなわち、バージョンに規定された回数と同一の回数だけhash関数を適用したか否かを判定する。適用回数がバージョンに規定されている回数に達していないとき、ステップＳ１６０に進み、DVDプレーヤ１は、変数ｇを１だけインクリメントする。そして、ステップＳ１５８に戻り、再び現在の鍵$lk_g$にhash関数を適用し、演算する。

【０１７５】

以上のようにして、バージョンに規定されている回数とhash関数を適用した回数が等しくなるまで、同様の処理が繰り返し実行される。

【０１７６】

なお、この繰り返し回数には、例えば１００回など上限値を設けるようにしてもよい。

【０１７７】

ステップＳ１５９で、バージョンに対応する回数だけhash関数が適用されたと判定された場合（有効な鍵$lk_{g+1}$が得られたと判定された場合）、並びに、ステップＳ１５６で、鍵lkが有効であると判定された場合、ステップＳ１６１に進み、上述した場合と同様にして、暗号鍵skを生成する。ステップＳ１６２では、ステップＳ１５５またはステップＳ１５

８で生成した鍵$lk_g$を鍵として、暗号鍵skを暗号化する。すなわち、次式を演算する。

$e = Enc(lk_g, sk)$

【０１７８】

次に、ステップＳ１６３において、DVDプレーヤ１は、パーソナルコンピュータ２に対して、ステップＳ１６２で暗号化したデータｅと、hash関数の適用回数を表す変数ｇを伝送する。パーソナルコンピュータ２においては、ステップＳ１６４でこれを受信すると、ステップＳ１６５で、パーソナルコンピュータ２におけるhash関数の適用回数を表す変数ｗに１を初期設定する。次に、ステップＳ１６６に進み、ステップＳ１６４で受信した変数ｇと、ステップＳ１６５で設定した変数ｗの値が等しいか否かを判定する。両者が等しくない場合、ステップＳ１６７に進み、パーソナルコンピュータ２のEEPROM５０に記憶されているlicense_keyにhash関数を適用して、新たな$license_key_{w+1}$を次式から求める。

$license\_key_{w+1} = hash(license\_key_w)$

【０１７９】

次に、ステップＳ１６８に進み、ｗを１だけインクリメントして、ステップＳ１６６に戻る。ステップＳ１６６で再び変数ｇと変数ｗが等しいか否かを判定し、両者が等しいと判定されるまで、ステップＳ１６７，Ｓ１６８の処理が繰り返し実行される。

【０１８０】

ステップＳ１６６で、変数ｇが変数ｗと等しいと判定された場合（現在有効な$license_key_w$が得られた場合）、ステップＳ１６９に進み、次式から暗号鍵sk'が演算される。

$sk' = Dec(license\_key_w, e)$

【０１８１】

以上のように、license_key（＝lk）を適宜更新するようにすれば、より安全性を高めることができる。

【０１８２】

なお、図２５と図２６に示した処理例の場合、バージョンを表す変数ｇをソース側からシンク側に伝送するようにしたが、これを伝送しないで、license_keyを更新することも可能である。この場合、図２５の処理に続いて、図２７に示す処理が実行される。

【０１８３】

すなわち、この例の場合、ステップＳ１６３で、DVDプレーヤ１からパーソナルコンピュータ２に対して、暗号化データｅだけが伝送され、バージョンを表す変数ｇは伝送されない。ステップＳ１６４で、パーソナルコンピュータ２がこの暗号化データｅを受信すると、ステップＳ１６５で、この暗号化データｅをlicense_keyを用いて復号する処理が、次式で示すように実行される。

$sk' = Dec(license\_key, e)$

【０１８４】

また、ステップＳ１６６で、DVDプレーヤ１は、ステップＳ１６１で生成した暗号鍵skを用いて、送出するデータを暗号化し、伝送する。パーソナルコンピュータ２は、ステップＳ１６７でこれを受信すると、ステップＳ１６８で、ステップＳ１６５で求めた暗号鍵sk'を用いて、復号する処理を実行する。次に、ステップＳ１６９で、復号した結果得られたデータが正しいか否かを判定する。この判定は、例えばMPEG方式のＴＳ(Transport Stream)パケットが受信されている場合には、そのヘッダ部分に、同期合わせのためのコード（１６進表示で４７）が挿入されているので、このコードが完全であるか否かをチェックすることで行うことができる。

【０１８５】

正しい復号ができなかった場合には、ステップＳ１７０に進み、パーソナルコンピュータ２は、次式に従って、license_keyを更新する。

$license\_key = hash(license\_key)$

【０１８６】

次に、ステップＳ１７１に進み、ステップＳ１７０で求めたlicense_keyを鍵として、ステップＳ１６４で受信した暗号化データｅを、次式に従って復号する。

sk'＝Dec（license_key, ｅ）

【０１８７】

そして、ステップＳ１６８に戻り、ステップＳ１７１で求めた暗号鍵sk'を用いて、ステップＳ１６７で受信した暗号化されているデータを復号する。ステップＳ１６９では、正しい復号が行われたか否かを再び判定する。以上のようにして、ステップＳ１６９で正しい復号が行われたと判定されるまで、ステップＳ１７０，Ｓ１７１，Ｓ１６８の処理が繰り返し実行される。

【０１８８】

以上のようにしても、license_keyを更新することができる。

【０１８９】

また、ソース側における暗号鍵の生成処理とシンク側における復号鍵（暗号鍵）の生成処理は、それぞれ処理対象とするデータと同期を取る必要がある。

【０１９０】

例えば、図２１に示すソース側の１３９４インタフェース２６において、LFSR５０１乃至排他的論理和回路５０８で生成する暗号鍵と、これを用いて暗号化するデータとしての平文の位相関係が、図２２に示すシンク側の１３９４インタフェース３６において、LFSR６０１乃至排他的論理和回路６０８で生成される暗号鍵と、この暗号鍵を用いて復号される暗号文の位相関係と一致している必要がある。そこで、図示は省略しているが、図２１の１３９４インタフェース２６においては、入力される平文に同期して暗号鍵が生成されるようになされており、また、図２２の１３９４インタフェース３６においては、入力される暗号文に同期して、暗号鍵が生成されるようになされてる。

【０１９１】

従って、例えばソース側から１３９４バス１１を介してシンク側に送出された暗号文を構成するパケットや所定のビットが何らかの理由で欠落してしまったような場合、ソース側における平文と暗号鍵の位相に対応する位相を、シンク側の暗号文と暗号鍵において保持することができなくなる。そこで、両者の位相関係が所定のタイミングで確実に更新される（初期化される）ようにすることができる。図２８は、このような処理を行う場合の構成例を表している。

【０１９２】

すなわち、この構成例においては、排他的論理和回路９０１が、乱数発生器９０３が発生する乱数と、入力される平文の排他的論理和を演算し、排他的論理和回路９０４と演算回路９０２に出力するようになされている。演算回路９０２にはまた、セッションキーＳも入力されている。演算回路９０２は、セッションキーＳと排他的論理和回路９０１の出力Ciに対して、所定の演算を施して、その演算結果を乱数発生器９０３に出力するようになされている。

【０１９３】

排他的論理和回路９０４は、排他的論理和回路９０１より入力されたデータと、時変キーｉの排他的論理和を演算し、暗号文として１３９４バス１１に出力するようになされている。

【０１９４】

同様に、シンク側においては、排他的論理和回路９１１が１３９４バス１１を介して入力される暗号文と、時変キーｉの排他的論理和を演算し、排他的論理和回路９１２と演算回路９１３に出力している。演算回路９１３には、セッションキーＳも入力されている。演算回路９１３は、排他的論理和回路９１１からの入力Ciと、セッションキーＳに対して所定の演算を施して、その演算結果を乱数発生器９１４に出力している。乱数発生器９１４は、演算回路９１３から入力される値を初期値として乱数を発生し、発生した乱数を排他的論理和回路９１２に出力している。排他的論理和回路９１２は、排他的論理和回路９１１より供給される暗号文と、乱数発生器９１４より入力される乱数との排他的論理和を演算し、暗号文を復号して平文として出力するようになされている。

【０１９５】

乱数発生器９０３は、例えば図２９に示すように、LFSR９３１乃至クロッキングファンクション９３６により構成されている。これらは、図２１に示したLFSR５０１乃至クロッキングファンクション５０６と同様の構成とされている。

【０１９６】

なお、図示は省略するが、シンク側の乱数発生器９１４も、図２９に示した構成と同様に構成されている。

【０１９７】

また、ソース側の演算回路９０２とシンク側の演算回路９１３は、それぞれ図３０のフローチャートに示すような処理を実行するように構成されている。

【０１９８】

次に、その動作について説明する。

【０１９９】

ソース側の演算回路９０２は、排他的論理和回路９０１からの入力Ciに、所定の関数ｆを適用して、Viを演算する機能、すなわち、次式を演算する機能を有している。

$Vi = f(S, C_i)$

【０２００】

ステップＳ２０１では、上記式における$C_i$に０を初期設定して次式が演算される。

$V_0 = f(S, 0)$

【０２０１】

演算回路９０２は、ステップＳ２０１で演算した値を、ステップＳ２０２で乱数発生器９０３に出力する。乱数発生器９０３では、演算回路９０２の出力$V_0$が、LFSR９３１乃至９３３に入力され、初期設定される。そして、図２１に示した場合と同様にして、乱数が生成され、加算器９３５から出力される。この乱数が排他的論理和回路９０１に供給される。

【０２０２】

排他的論理和回路９０１は、この乱数と入力された平文との排他的論理和を演算し、演算結果$C_i$を演算回路９０２に供給する。

【０２０３】

演算回路９０２においては、次に、ステップＳ２０３において、変数ｉに１が初期設定され、ステップＳ２０４において、排他的論理和回路９０１から入力されたデータが$C_i$に設定される。

【０２０４】

次に、ステップＳ２０５に進み、演算回路９０２は、次式を演算する。

$V_i = f(S, C_i) + V_{i-1}$

【０２０５】

いまの場合、ｉ＝１であるから、次式が演算される。

$V_1 = f(S, C_1) + V_0$

【０２０６】

次に、ステップＳ２０６に進み、いま取り込まれたデータ$C_i$が予め設定してある所定の値Ｔと等しいか否かが判定される。両者の値が等しくない場合、ステップＳ２０７に進み、変数ｉが１だけインクリメントされた後、ステップＳ２０４に戻る。すなわち、いまの場合、ｉ＝２とされ、次に入力されたデータが$C_2$に設定される。

【０２０７】

次に、ステップＳ２０５で次式が演算される。

$V_2 = f(S, C_2) + V_1$

【０２０８】

ステップＳ２０６では、$C_2$の値が所定の値Ｔと等しいか否かが判定され、等しくなければ、ステップＳ２０７に進み、変数ｉが１だけインクリメントされ、再びステップＳ２０４以降の処理が実行される。

【０２０９】

ステップＳ２０６において、$C_i$の値が所定の値Ｔと等しいと判定された場合、ステップＳ２０８に進み、ステップＳ２０５で演算された値$V_i$が乱数発生器９０３に出力される。乱数発生器９０３では、ステップＳ２０２で説明した場合と同様に、この値がLFSR９３１乃至９３３に初期値として設定される。そして、この初期値に対応する乱数が加算器９３５から出力される。

【０２１０】

演算回路９０２は、$V_i$を乱数発生器９０３に出力した後、ステップＳ２０３に戻り、変数ｉを１に初期設定した後、それ以降の処理を繰り返し実行する。

【０２１１】

いま、例えばＴの値が８ビットで表されるものとし、$C_i$の値は、その発生確率が均等であるとすると、２５６（＝$2^8$）回に１回の割合で、$C_i$の値はＴと等しくなることになる。従って、乱数発生器９０３の発生する乱数は、２５６回に１回の割合で初期化（更新）されることになる。

【０２１２】

排他的論理和回路９０１より出力されたデータは、排他的論理和回路９０４に入力され、時変キーｉとの排他的論理和が演算された後、暗号文として１３９４バス１１に出力される。

【０２１３】

シンク側においては、排他的論理和回路９１１が、１３９４バス１１を介して入力された暗号文と、時変キーｉの排他的論理和を演算し、その演算結果$C_i$を、演算回路９１３に出力している。演算回路９１３は、上述したソース側の演算回路９０２と同様の処理を実行し、２５６回に１回の割合で、乱数発生器９１４に初期値$V_i$を供給する。乱数発生器９１４は、入力された値$V_i$を初期値として乱数を発生し、発生した乱数を排他的論理和回路９１２に出力する。排他的論理和回路９１２は、入力された乱数と、排他的論理和回路９１１より入力された暗号化されているデータとの排他的論理和を演算し、演算結果を平文として出力する。

【０２１４】

このように、演算回路９１３は、排他的論理和回路９１１が暗号データを２５６回出力すると１回の割合で初期値を発生する。従って、シンク側に１３９４バス１１を介して入力される暗号データに欠落が生じたとしても、シンク側の暗号文と乱数の位相関係は、暗号データ２５６個に１個の割合で初期化されるため、その時点で位相関係が回復することになる。

【０２１５】

なお、演算回路９０２または演算回路９１３が初期値を出力するのは、Ｔ＝$C_i$となった場合であるから、２５６回に１回の割合で定期的に初期値が出力されるのではなく、平均すると確率的にそのようになるにすぎない。

【０２１６】

なお、送信または受信した暗号データの数をカウントして、同様の処理を実行させるようにすることも可能であるが、そのようにすると、１３９４バス１１上でデータが欠落すると、ソース側におけるデータのカウント値とシンク側におけるデータのカウント値とが異なる値となってしまい、結局、両者の同期を取ることができなくなってしまう。そこで、上記した実施の形態のようにするのが好ましい。

【０２１７】

また、乱数発生器９０３または乱数発生器９１４に供給する初期値としては、排他的論理和回路９０１または９１１の出力するデータ$C_i$をそのまま利用することも可能である。しかしながら、このデータ$C_i$は、１３９４バス１１上を伝送されるデータであり、盗まれるおそれがある。そこで、データ$C_i$を初期値として直接利用せず、これに対して所定の演算を施すことによって生成された値$V_i$を初期値とするようにすれば、より安全性を高めることができる。

【０２１８】

ところで、IEEE１３９４バス１１のデータ転送方式には、asynchronous転送とisochronous転送の２つの方法がある。このうちのasynchronous転送は、２つの機器間での１対１の転送であり、isochronous転送は、１つの機器から１３９４バス１１上のすべての機器に対する同報通信であると考えることができる。従って、例えば図４などに示した認証、鍵共有プロトコルの通信は、同報する必要がないので、asynchronous転送で行われるのが普通である。

【０２１９】

いま、図４の認証、鍵共有プロトコルにおいて、例えばパーソナルコンピュータ２が不正な機器で、license_keyを持っていない場合にも、DVDプレーヤ１からｅを送ってもらうことができる。ここでｅは、セッションキーskを鍵lkで暗号化した暗号文である。不正な機器であるパーソナルコンピュータ２は、license_keyを持っていないので、ｅを復号して正しいskを得ることはできないが、ｅが暗号解読に用いられるおそれがある。

【０２２０】

なんらかの理由によりパーソナルコンピュータ２がセッションキーskを得た場合には、平文skと鍵lkを用いて暗号化した暗号文ｅの両方を手に入れたことになる。その結果、これらが暗号解読に用いられるおそれがある。さらにいえば、攻撃者が平文と暗号文の組をたくさん知るほど、一般的に暗号解読が容易になる。

【０２２１】

また、不正なパーソナルコンピュータ２がＩＤをDVDプレーヤ１に教える際に、虚偽のＩＤを教えると、DVDプレーヤ１は、この虚偽のＩＤに基づいて鍵lkを計算し、これに基づいてセッションキーskを暗号化して送り返してしまう。このような操作を繰り返すことにより、パーソナルコンピュータ２は、ひとつの平文skを複数の暗号鍵lkでそれぞれ暗号化した、複数の暗号文ｅを手に入れることができることになる。

【０２２２】

図３１は、この点を考慮して、シンク機器が不正な機器であった場合には、あるセッションキーskを鍵lkで暗号化した暗号文ｅが２つ以上は、そのシンク機器にわたらないようにする処理例を表している。この図３１における処理は、基本的に、図４に示した処理と同様であるが、ソース機器がシンク機器に対してＩＤを要求する以前に、いくつかの処理が設けられている。

【０２２３】

すなわち、図３１の処理例においては、ステップＳ２０１において、シンク機器としてのパーソナルコンピュータ２が、ソース機器としてのDVDプレーヤ１に対して認証プロトコルの開始を要求する認証要求を転送する。この認証要求は、プロトコルの他の転送と同様に、asynchronous転送によって行われる。

【０２２４】

IEEE１３９４バス１１では、それに接続されているそれぞれの機器が、バスリセット時に固有のノード番号が割り当てられ、各機器は、このノード番号によって、送信機器、受信機器の指定、識別を行うようにしている。

【０２２５】

図３２は、asynchronousパケットのひとつである、write request for data quadletパケットのフォーマットを示している。同図におけるdestination_IDは、受信機器のノード番号を示し、source_IDは、送信機器のノード番号を示している。認証要求を表すパケットでは、quadlet_dataの位置に、あらかじめ定められた認証要求を表すデータが挿入される。

【０２２６】

DVDプレーヤ１は、ステップＳ２０２で認証要求を表すasynchronousパケットを受け取ると、そのパケットを送信した機器のノード番号であるsource_IDを読み取る。そして、ステップＳ２０３において、DVDプレーヤ１は、現在のセッションキーskに関して、このノード番号の機器に対して、暗号文ｅを既に送っているか否かを判定する。暗号文ｅを既に送ったことがある場合には、DVDプレーヤ１は、パーソナルコンピュータ２に対する認証

プロトコルの処理を終了する。これに対して、パーソナルコンピュータ２に対して、まだ暗号文ｅをまだ送信したことがない場合には、さらにステップＳ２０４以降の認証プロトコルが実行される。

【０２２７】

このステップＳ２０４乃至ステップＳ２１３の処理は、図４におけるステップＳ１乃至ステップＳ１０の処理と同様の処理である。

【０２２８】

このような処理が行われた後、DVDプレーヤ１は、ステップＳ２１４において、ステップＳ２１３で読み取ったパーソナルコンピュータ２のノード番号を、EEPROM２７に記憶する。このノード番号は、DVDプレーヤ１が現在のセッションキーskを使い続ける限り保存される。そして、セッションキーskを変更するとき、消去される。

【０２２９】

以上のようにすることにより、ひとつのシンク機器がひとつのセッションキーskについて得られる暗号文ｅの数は高々ひとつであるプロトコルを構成することができる。これにより、より安全性を高めることが可能となる。

【０２３０】

ところで、図４の認証プロトコルのステップＳ７においては、ソース機器がシンク機器に対して送るべきセッションキーskを鍵lkを用いて暗号化し、ｅを生成している。この暗号アルゴリズムのうちで広く用いられているものにブロック暗号がある。このブロック暗号は、平文の一定の長さのブロックを単位として暗号化処理を行うものであり、よく知られているものとして、DES暗号がある。このDES暗号は、平文の６４ビットのブロックを６４ビットの暗号文に変換する暗号アルゴリズムである。

【０２３１】

いま、図４のステップＳ７において、使用する暗号アルゴリズムをｎビットの平文をｎビットの暗号文に変換するｎビットブロック暗号とし、セッションキーskのビット長をｎビットとする。また、ｎビットのセッションキーskを、この暗号アルゴリズムに入力し、鍵lkを用いて変換された結果のｎビットを、そのままｅとするものとする。

【０２３２】

この場合、ソース機器が所定のシンク機器に対して、以前に使用したことのあるセッションキーskを送ろうとした場合、暗号アルゴリズムの入力と鍵が同一であるため、ｅも以前使われたものと同じになり、例えばｅを盗聴した不正者に、以前と同じセッションキーskが使われているという情報を与えてしまうことになる。

【０２３３】

図３３は、この点を考慮した認証プロトコルの例を表している。図３３におけるステップＳ２２１乃至ステップＳ２２６までの処理は、図４におけるステップＳ１乃至ステップＳ６の処理と同様の処理であるので、ここではその説明を省略する。

【０２３４】

ステップＳ２２７において、ソース機器はｎビットの乱数ｒを生成し、ステップＳ２２８において、次式に従って、暗号ｒとセッションキーskの連結を鍵lkで暗号化する。

ｅ＝Enc（lk ｒ || sk）

【０２３５】

このとき、CBCモードという暗号モードが用いられる。図３４は、このCBCモードの構成を表している。同図において、左側半分が暗号化処理を、右側半分が復号化処理を、それぞれ表している。レジスタ１００３とレジスタ１０１２には、同一の値の初期値ＩＶが格納されている。この初期値ＩＶは、システム全体で固定されている。

【０２３６】

暗号化処理においては、まず、平文のｎビットの第１ブロックがレジスタ１００３の値ＩＶと排他的論理和演算回路１００１において排他的論理和演算され、その結果が暗号器１００２に入力される。暗号器１００２のｎビットの暗号文は、第１ブロックとして通信路に送信されるとともに、レジスタ１００３に格納される。

【０２３７】
平文のｎビットの第２ブロックが入力されると、この第２ブロックは、レジスタ１００３に格納されているｎビットの暗号文の第１ブロックと排他的論理和回路１００１において排他的論理和演算される。その演算結果は、暗号器１００２に入力され、暗号化される。暗号器１００２の出力するｎビットの暗号文は、第２ブロックの暗号文として通信路に送信されるとともに、レジスタ１００３に格納される。以上の処理が繰り返し実行される。
【０２３８】
一方、復号側においては、通信路を介して伝送されてきた暗号文の第１ブロックが復号器１０１１により復号され、排他的論理和回路１０１３において、レジスタ１００２に保持されている初期値ＩＶと排他的論理和演算され、平文の第１ブロックが生成される。
【０２３９】
通信路を介して伝送されてきた第１ブロックの暗号文は、レジスタ１０１２に保持される。そして、通信路を介して第２ブロックの暗号文が供給されてきたとき、復号器１０１１が、この第２ブロックの暗号文を復号し、排他的論理和回路１０１３に供給する。排他的論理和回路１０１３は、復号器１０１１の出力する第２ブロックの復号結果とレジスタ１００２に保持されている第１ブロックの暗号文との排他的論理和を演算し、第２ブロックの平文を生成する。
【０２４０】
第２ブロックの暗号文はまた、レジスタ１０１２に保持される。
【０２４１】
以上のような処理が繰り返し実行され、復号化処理が行われる。
【０２４２】
なお、CBCモードに関しては、Bruce Schneier著のApplied Cryptography (Second edition)に詳述されている。
【０２４３】
図３３に戻って、ステップＳ２２８においては、ｎビットの乱数ｒを平文の第１ブロックとし、セッションキーskを平文の第２ブロックとして暗号プロトコルに入力する。従って、第１ブロックの乱数ｒは、レジスタ１００３に保持されている初期値ＩＶと排他的論理和演算された後、暗号器１００２で鍵lkを用いて暗号化される。従って、暗号器１００２から、Enc( lk, r(+)IV ) が出力される。
【０２４４】
この暗号器１００２の出力がレジスタ１００３に保持され、第２ブロックの平文としてのセッションキーskが入力されてきたとき、排他的論理和回路１００１で排他的論理和が演算される。従って、このとき、暗号器１００２の出力は、Enc( lk, sk(+)Enc( lk,r(+)IV )) となる。
【０２４５】
ソース機器は、ステップＳ２２９において、２つのブロックの連結を次式で示すように演算し、シンク機器に送信する。
ｅ＝Enc( lk, r(+)IV ) || Enc( lk, sk(+)Enc( lk, r(+)IV ))
【０２４６】
シンク機器側においては、ステップＳ２３０で送信されてきたｅを受け取り、ステップＳ２３１において、EEPROM５０に記憶されているlicense_keyを用いて、これを復号する。復号して得られた結果のうち、第１ブロックをr'とし、第２ブロックをsk'とする。
【０２４７】
以上のようにして、シンク機器が正しいlicense_keyを持っている場合においてのみ、sk = sk'となり、ソース機器側とシンク機器側において、セッションキーを共有することができる。
【０２４８】
上述のｅの式が意味するところは、同一のセッションキーskが２度以上用いられたとしても、乱数ｒが変化すれば、ｅも変化するということである。このため、ｅを盗聴されたと

しても、盗聴者にはセッションキーskが同一であるかどうか不明であるため、安全性を高めることができる。

【０２４９】

なお、ブロック暗号の利用モードとしてよく知られているものに、上記したCBCモードの他、ECBモード、CFBモード、OFBモードなどがある。このうちの後者の２つはフィードバックを用いているので、図３３に示した処理に用いることができる。また、これ以外の暗号モードについても、フィードバックを用いるモードのものは、適用することが可能である。ブロック暗号の利用モードについても、上記したApplied Cryptography (Second edition)に詳述されている。

【０２５０】

ところで、図４に示した処理例においては、ソース機器がシンク機器に対してセッションキーskを暗号化した暗号文ｅを送り、正当なシンク機器のみが、この暗号文ｅを正しく復号してセッションキーskを得られるので、実質的にソース機器がシンク機器を認証していることになる。この方式では、ソース機器の認証は行われていない。その結果、不正なソース機器がシンク機器に対して、でたらめなデータをｅとして送った場合においても、シンク機器は、それを復号した結果をセッションキーskとして受け入れてしまうことが起こりえる。そこで、これを防止するために、図３５に示すような、処理を行うことができる。

【０２５１】

この図３５の例においては、ステップＳ２４１において、シンク機器としてのパーソナルコンピュータ２が、あらかじめ定められているビット数（例えば６４ビット）の乱数ｒを生成し、ステップＳ２４２において、これをソース機器としてのDVDプレーヤ１に送信する。DVDプレーヤ１は、ステップＳ２４３において、この乱数ｒを受信し、ステップＳ２４４において、パーソナルコンピュータ２に対して、ＩＤを要求する。ステップＳ２４５でこれを受信したパーソナルコンピュータ２は、ステップＳ２４６において、EEPROM５０に記憶されているＩＤを読み出し、DVDプレーヤ１に送信する。DVDプレーヤ１は、ステップＳ２４７で、このＩＤを受信する。

【０２５２】

DVDプレーヤ１はまた、ステップＳ２４８において、次式に基づいて、鍵lkを生成する。

lk = hash( ID || service_key )

【０２５３】

また、DVDプレーヤ１は、ステップＳ２４９において、セッションキーskを生成する。

【０２５４】

さらに、ステップＳ２５０において、DVDプレーヤ１は、次式に基づいて、ｅを生成する。

ｅ = Enc( lk, r || sk )

【０２５５】

このようにして生成されたｅは、ステップＳ２５１において、DVDプレーヤ１からパーソナルコンピュータ２に送信される。

【０２５６】

なお、このときの暗号化モードとしては、CBCモードなど、フィードバックを利用するものが使用される。

【０２５７】

ステップＳ２５２で、ｅを受信したパーソナルコンピュータ２は、ステップＳ２５３において、ｅをlicense_keyを用いて復号した結果を、r'とsk'の連結r'||sk'とする。

【０２５８】

このとき、r'のビット数は、あらかじめ定められているrのビット数と同じビット数になるようにする。

【０２５９】

次に、ステップＳ２５４において、パーソナルコンピュータ２は、ｒ = r' が成り立つか

どうかを検査する。ｒ＝ｒ' が成立する場合、パーソナルコンピュータ２は、DVDプレーヤ１が正当な機器であることを確認し、sk'をセッションキーとして受理する。これは、license_keyを用いて復号した結果のr'がrと等しくなるような暗号文ｅを作成することができるのは、正しい鍵lkを作成することが可能な機器だけであるからである。

【０２６０】

これに対して、ｒ＝ｒ' が成立しない場合には、パーソナルコンピュータ２は、DVDプレーヤ１は、正当な機器ではないと判断し、sk'を破棄する。

【０２６１】

以上のように認証方式を構成することで、シンク機器がソース機器を認証することが可能となる。また、このように認証することにより、図４の処理例において実現されていた、正当なシンク機器だけが正しいセッションキーを得ることができる、という特徴も満足している。

【０２６２】

図３６には、上記と同じ、シンク機器がソース機器の正当性を確認できる認証方式の、別の処理例が示されている。本処理例において、ステップＳ２６１からステップＳ２６６の処理は、図４のステップＳ１からステップＳ６と同様であるので、その説明は省略する。

【０２６３】

ステップＳ２６７において、DVDプレーヤ１は時刻情報をTとする。この時刻情報として具体的には、例えば、IEEE１３９４規格において定められている、３２ビットのCYCLE_TIMEレジスタの値を使用する。

【０２６４】

CYCLE_TIMEレジスタはIEEE１３９４バス上における機器の時刻情報を一定にするために用いられ、バス上に１つあるサイクルマスターと呼ばれる機器からの同報パケットによって各機器のCYCLE_TIMEレジスタが一様に更新される。さらにバス上に共通の24.576MHzのクロックによってもCYCLE_TIMEレジスタが１ずつ加算されるので、レジスタは約４０ナノ秒毎に１度加算される。このことにより、バス上の各機器間の時計合わせが行える。

【０２６５】

DVDプレーヤ１は、ステップＳ２６８において、T||skをlkで暗号化してeを得て、ステップＳ２６９でパーソナルコンピュータ２に送信する。暗号化の際の暗号モードとしては、CBCモードなど、フィードバックを利用するものが使用される。

【０２６６】

パーソナルコンピュータ２はステップＳ２７０でｅを受信し、ステップＳ２７１でlicense_keyを用いてこれを復号し、その結果をT'||sk' とおく。この際、T'の部分のビット数を３２ビットとする。

【０２６７】

ステップＳ２７２で、T'の正当性を検査する。この検査では、パーソナルコンピュータ２自身が持つCYCLE_TIMEレジスタの値と、T'の値を比較し、その差が例えば１００ミリ秒以内であればこれを正しいとし、それを越えていれば不正であるとする。

【０２６８】

この検査に合格した場合、パーソナルコンピュータ２はDVDプレーヤ１が正当な機器であると判断して、sk'をセッションキーとして受理し、不合格の場合にはパーソナルコンピュータ２はDVDプレーヤ１が不正な機器であるとしてsk'を破棄する。これは、license_keyを用いて復号した結果が正しいT'となるような暗号文を生成できるのは、正しいlkを作れる機器だけであるためである。

【０２６９】

以上のように認証方式を構成することにより、シンク機器がソース機器を認証できる方式とすることが可能となる。加えて、この方式でも、図４の処理例で実現されていた、正当なシンク機器だけが正しいセッションキーを得ることができる、という特徴も満たしている。

【０２７０】

図４の処理例においては、license_keyを所有している正当なシンク機器のみがｅを正しく復号してsk=sk'なるsk'を得ることが出来るので、実質的にソース機器がシンク機器を認証する方式を構成している。しかし、この方式ではシンク機器が不正デバイスである場合にも、セッションキーをlkを用いて暗号化した暗号文ｅを得ることができてしまう。不正デバイスであるシンク機器はｅを用いて暗号解読を行ってセッションキーskを得ようとする可能性がある。

【０２７１】

この問題に対し、図３７に示す認証方式の処理例では、ソース機器がシンク機器を正当なものであると確認した後に、セッションキーを暗号化した暗号文を送るようにしている。図３７の処理例を説明する。以下の処理例において、暗号化を行う際の暗号モードは、CBCモードなど、フィードバックを利用するものを使用する。

【０２７２】

ステップＳ２８１からステップＳ２８５は図４の処理例のステップＳ１からステップＳ５と同様であるので説明を省略する。ステップＳ２８６において、DVDプレーヤ１はあらかじめ定められたビット数（例えば６４ビット）の乱数r1とr2を生成し、その連結をM1とする。ステップＳ２８７において、M1を鍵lkで暗号化してXを作り、ステップＳ２８８でXをパーソナルコンピュータ２に送る。

【０２７３】

ステップＳ２８９でXを受け取ったパーソナルコンピュータ２は、ステップＳ２９０で、license_keyを用いてこれを復号し、あらかじめ定められたビット数（例えば６４ビット）ごとに分割してr1'||r2'とする。次にステップＳ２９１であらかじめ定められたビット数（例えば６４ビット）の乱数r3を生成し、ステップＳ２９２で、r3とr2'を連結してM2を得る。ステップＳ２９３で、M2をlicense_keyを用いて暗号化してYを得、ステップＳ２９４で、YをDVDプレーヤ１に送信する。

【０２７４】

ステップＳ２９５でYを受信したDVDプレーヤ１は、ステップＳ２９６でlkを用いてこれを復号し、あらかじめ定められたビット数（例えば６４ビット）ごとに分割してr3'||r2''とする。ステップＳ２９７において、r2''と先に送ったr2が等しいかどうかを検査する。この検査に失敗した場合、DVDプレーヤ１はパーソナルコンピュータ２が正当な機器ではないと判断して認証プロトコルをそこで終了する。この検査に合格した場合、ステップＳ２９８においてDVDプレーヤ１はセッションキーskを生成し、ステップＳ２９９において、r3'とskを連結することによりM3を得る。ステップＳ３００でM3を鍵lkを用いて暗号化して暗号文Zを得て、ステップＳ３０１でこれをパーソナルコンピュータ２に送信する。

【０２７５】

パーソナルコンピュータ２はステップＳ３０２でZを受信し、ステップＳ３０３でZをlicense_keyで復号し、あらかじめ定められたビット数（例えば６４ビット）ごとに分割してr3''||sk'とする。ステップＳ３０４において、r3''が先に送信したr3と等しいかどうかを検査する。 この検査に失敗した場合、パーソナルコンピュータ２はDVDプレーヤ１が正当な機器ではないと判断して認証プロトコルを終了する。この検査に合格した場合、パーソナルコンピュータ２はsk'をセッションキーskとして受理する。

【０２７６】

以上のように認証プロトコルを構成することにより、ソース機器はシンク機器が正当な機器であることを認証した後にセッションキーskを暗号化した暗号文をシンク機器に対して送ることができる。また本処理例では図３３に示した処理例と同様に、たとえソース機器が以前使ったものと同じセッションキーskを用いたとしても、それを鍵lkによって暗号化した暗号文が以前のものと変わるので、情報が漏れにくいという性質も有している。

【０２７７】

ただし、図３７の処理例においては、使用している暗号アルゴリズムがnビットのもので、r1,r2,r3,skのビット数もnビットである時には、問題がある。もしソース機器が不正デバイスであった場合でも、ステップＳ３００において、Zの前半nビットとして、ステップ

Ｓ２９５で受信したYの前半nビットをそのまま使うことにより、シンク機器のステップＳ３０３の検査をパスすることができてしまう。

【０２７８】

この問題に鑑みて、ソース機器がシンク機器の正当性を確認した後にセッションキーを暗号化した暗号文を送信するのみならず、シンク機器がソース機器の正当性を確認できる認証プロトコルの処理例を図３８から図４０に示す。図３８と図３９の処理例は図３７の処理例の変形例である。

【０２７９】

図３８に示した認証プロトコルの処理例を説明する。以下の処理例において、暗号化を行う際の暗号モードは、CBCモードなど、フィードバックを利用するものを使用する。

【０２８０】

ステップＳ３１１からステップＳ３２７は図３７のステップＳ２８１からステップＳ２９７と同様なので説明は省略する。ステップＳ３２８において、DVDプレーヤ１はあらかじめ定められたビット数（例えば６４ビット）の乱数r4とセッションキーskを生成する。ステップＳ３２９においてr4とr3'とskを連結してM3を作り、ステップＳ３３０においてM3を鍵lkで暗号化してZを計算し、ステップＳ３３１においてZをパーソナルコンピュータ２に送信する。

【０２８１】

ステップＳ３３２でZを受信したパーソナルコンピュータ２は、ステップＳ３３３でこれをlicense_keyを用いて復号し、その結果をあらかじめ定められたビット数（例えば６４ビット）ごとに分割してr4'||r3''||sk'とする。ステップＳ３３４でr3''が先に送信したr3と等しいかどうかを検査し、等しい場合のみsk'をセッションキーとして受理する。

【０２８２】

以上のように認証プロトコルを構成すれば、ソース機器がシンク機器の正当性を確認した後にセッションキーを暗号化した暗号文を送信するのみならず、シンク機器がソース機器の正当性を確認できる。

【０２８３】

図３９の処理例も図３７の処理例の変形例である。以下の説明において、暗号化を行う際の暗号モードは、CBCモードなど、フィードバックを利用するものを使用する。

【０２８４】

図３９のステップＳ３５１からステップＳ３６１は図３７のステップＳ２８１からステップＳ２９１と同様であるので説明は省略する。ステップＳ３６２において、パーソナルコンピュータ２はM2をr2'||r3として生成する。ステップＳ３６３において、M2をlicense_keyで暗号化してYを得、ステップＳ３６４でDVDプレーヤ１に送信する。

【０２８５】

ステップＳ３６５でYを受信したDVDプレーヤ１は、ステップＳ３６６でこれを鍵lkを用いて復号してその結果をあらかじめ定められたビット数（例えば６４ビット）ごとに分割してr2''||r3とおく。ステップＳ３６７において、r2''が先に送信したr2と等しいかどうかを検査する。この検査に失敗した場合、DVDプレーヤ１はパーソナルコンピュータ２が正当な機器ではないと判断して認証プロトコルをそこで終了する。この検査に合格した場合、ステップＳ３６８においてDVDプレーヤ１はセッションキーskを生成し、ステップＳ３６９において、r3'とskを連結することによりM3を得る。ステップＳ３７０でM3を鍵lkを用いて暗号化して暗号文Zを得て、ステップＳ３７１でこれをパーソナルコンピュータ２に送信する。

【０２８６】

パーソナルコンピュータ２はステップＳ３７２でZを受信し、ステップＳ３７３でZをlicense_keyで復号し、あらかじめ定められたビット数（例えば６４ビット）ごとに分割してr3''||sk'とする。ステップＳ３７４において、r3''が先に送信したr3と等しいかどうかを検査する。この検査に失敗した場合、パーソナルコンピュータ２はDVDプレーヤ１が正当な機器ではないと判断して認証プロトコルを終了する。この検査に合格した場合、パーソ

ナルコンピュータ２はsk'をセッションキーskとして受理する。

【０２８７】

以上のように認証プロトコルを構成することにより、ソース機器はシンク機器が正当な機器であることを認証した後にセッションキーskを暗号化した暗号文をシンク機器に対して送ることができ、またシンク機器はソース機器が正当な機器であることを確認することができる。また本処理例では図３３に示した処理例と同様に、たとえソース機器が以前使ったものと同じセッションキーskを用いたとしても、それを鍵lkによって暗号化した暗号文が以前のものと変わるので、情報が漏れにくいという性質も有している。

【０２８８】

図４０に示す認証プロトコルの処理例も同様の目的を満たすものである。以下の処理例において、暗号モードとしてはCBCモードなどのフィードバックを利用するものを使用するものとする。図４０において、ステップＳ３８１からステップＳ３８４は図４のステップＳ１からステップＳ４と同様なので説明は省略する。ステップＳ３８５において、DVDプレーヤ１はあらかじめ定められたビット数（例えば６４ビット）の乱数Rsrcを生成し、ステップＳ３８６でパーソナルコンピュータ２に送信する。

【０２８９】

ステップＳ３８７でパーソナルコンピュータ２はRsrcを受信し、ステップＳ３８８であらかじめ定められたビット数（例えば６４ビット）の乱数Rsnkを生成し、ステップＳ３８９でRsnkとRsrcを連結してM1を生成し、ステップＳ３９０でM1を鍵license_keyを用いて暗号化してXを計算してステップＳ３９１でXを送信する。

【０２９０】

DVDプレーヤ１はステップＳ３９２でXを受信し、ステップＳ３９３でパーソナルコンピュータ２のＩＤとservice_keyからlkを計算し、ステップＳ３９４でこのlkを用いてXを復号し、その結果をあらかじめ定められたビット数（例えば６４ビット）ごとに分割してRsnk'||Rsrc'とする。ステップＳ３９５においてRsrc'が先に送信したRsrcと等しいことを検査し、この検査に合格しなければ、パーソナルコンピュータ２が不正デバイスであるとして認証プロトコルを終了する。この検査に合格すれば、DVDプレーヤ１はステップＳ３９６でセッションキーskを生成し、ステップＳ３９７でRsrcとRsnk'とskを連結してM2を生成し、ステップＳ３９８でM2を鍵lkを用いて暗号化してYを得、ステップＳ３９９でパーソナルコンピュータ２に送信する。

【０２９１】

ステップＳ４００でYを受信したパーソナルコンピュータ２は、ステップＳ４０１で鍵license_keyを用いてYを復号してこの結果をあらかじめ定められたビット数（例えば６４ビット）ごとに分割してRsrc''||Rsnk''||sk'とおく。ステップＳ４０２でRsnk''が先に送信したRsnkと等しいことを検査する。この検査に失敗した場合、パーソナルコンピュータ２は、DVDプレーヤ１が不正デバイスであるとしてsk'を破棄する。この検査に合格した場合には、パーソナルコンピュータ２はDVDプレーヤ１が正当なデバイスであることを認め、sk'をセッションキーとして受理する。

【０２９２】

以上のように認証プロトコルを構成することにより、ソース機器はシンク機器が正当な機器であることを認証した後にセッションキーskを暗号化した暗号文をシンク機器に対して送ることができ、またシンク機器はソース機器が正当な機器であることを確認することができる。また本処理例では図３３に示した処理例と同様に、たとえソース機器が以前使ったものと同じセッションキーskを用いたとしても、それを鍵lkによって暗号化した暗号文が以前のものと変わるので、情報が漏れにくいという性質も有している。

【０２９３】

以上においては、DVDプレーヤ１をソースとし、パーソナルコンピュータ２と光磁気ディスク装置３をシンクとしたが、いずれの装置をソースとするかシンクとするかは任意である。

【０２９４】

また、各電子機器を接続する外部バスも、１３９４バスに限らず、種々のバスを利用することができ、それに接続する電子機器も、上述した例に限らず、任意の装置とすることができる。

【０２９５】

なお、上記各種の指令を実行するプログラムは、磁気ディスク、CD-ROMディスクなどの提供媒体を介してユーザに提供したり、ネットワークなどの提供媒体を介してユーザに提供し、必要に応じて内蔵するRAMやハードディスクなどに記憶して利用させるようにすることができる。

【０２９６】

【発明の効果】

以上の如く、請求項１に記載の認証システムによれば、第１の電子機器が、所定の処理を施す情報に対応する第１の鍵を記憶し、第２の電子機器が、自分自身に固有の識別番号を記憶し、記憶している識別番号を第１の電子機器に送信し、第１の電子機器が、第２の電子機器から送信されてくる、識別番号を受信し、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出し、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成し、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成し、生成した第１の送信値を第２の電子機器に送信し、第２の電子機器が、第１の電子機器から送信されてくる、第１の送信値を受信し、受信した第１の送信値に基づいて、第２の乱数を生成し、あらかじめ定められたビット数の第３の乱数を生成し、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成し、生成した第２の送信値を第１の電子機器に送信し、第１の電子機器が、第２の電子機器から送信されてくる、第２の送信値を受信し、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、第２の電子機器が正当であるか否かを認証し、比較の結果、第２の電子機器が正当でないと認証された場合、認証の処理を終了するようにしたので、安全性を向上することが可能となる。

【０２９７】

請求項４に記載の電子機器、請求項５に記載の認証方法、および請求項６に記載の記録媒体によれば、所定の処理を施す情報に対応する第１の鍵を記憶し、他の電子機器に付与された固有の識別番号を受信し、記憶している第１の鍵および受信した識別番号に基づいて、ハッシュ関数を用いて、ハッシュ値を算出し、あらかじめ定められたビット数の第１の乱数および第２の乱数を生成し、算出したハッシュ値と、生成した第１の乱数および第２の乱数に基づいて、第１の送信値を生成し、生成した第１の送信値を他の電子機器に送信し、他の電子機器から送信されてくる、第２の乱数および第３の乱数に基づいて生成された第２の送信値を受信し、受信した第２の送信値に基づく値と、生成した第２の乱数に基づく値とを比較することにより、他の電子機器が正当であるか否かを認証し、比較の結果、他の電子機器が正当でないと認証された場合、認証の処理を終了するようにしたので、より安全に、適正な他の電子機器に対してだけ、所定の情報処理を行わせるようにすることができる。

【０２９８】

請求項７に記載の電子機器、請求項１０に記載の認証方法、および請求項１１に記載の記録媒体によれば、自分自身に固有の識別番号を記憶し、記憶している識別番号を他の電子機器に送信し、他の電子機器から送信されてくる、所定の処理を施す情報に対応する第１の鍵および識別番号から算出されたハッシュ値、並びに第１の乱数および第２の乱数に基づいて生成された第１の送信値を受信し、受信した第１の送信値に基づいて、第２の乱数を生成し、あらかじめ定められたビット数の第３の乱数を生成し、生成した第２の乱数および第３の乱数に基づいて、第２の送信値を生成し、生成した第２の送信値を他の電子機器に送信するようにしたので、より安全性を高めることができる。

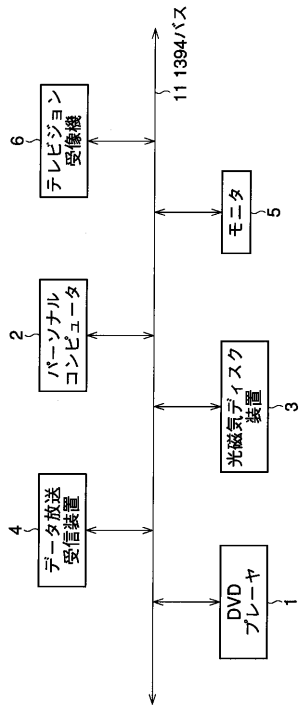
【図面の簡単な説明】

【図１】本発明を適用した情報処理システムの構成例を示すブロック図である。

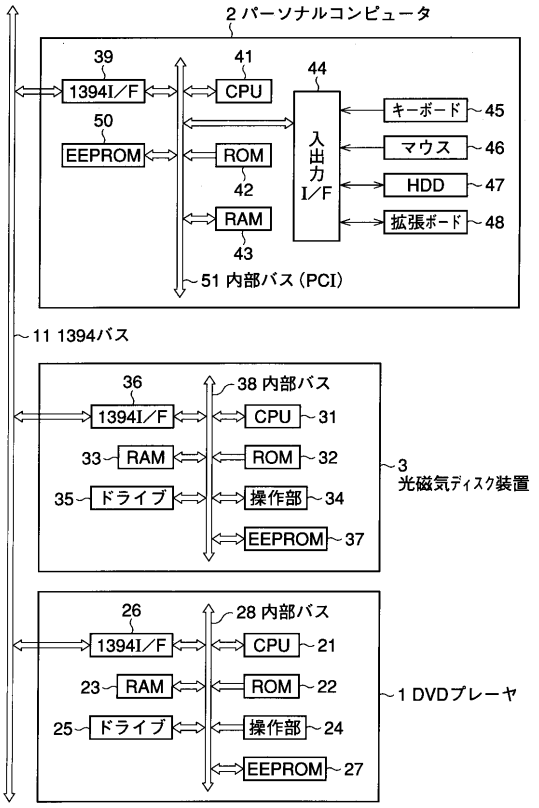
【図２】図１のDVDプレーヤ１、パーソナルコンピュータ２、および光磁気ディスク装置３の内部の構成例を示すブロック図である。
【図３】認証処理を説明する図である。
【図４】認証処理を説明するタイミングチャートである。
【図５】 node_unique_IDのフォーマットを示す図である。
【図６】他の認証処理を説明するタイミングチャートである。
【図７】さらに他の認証処理を説明するタイミングチャートである。
【図８】他の認証処理を説明するタイミングチャートである。
【図９】他の認証処理を説明するタイミングチャートである。
【図１０】暗号化処理を説明するブロック図である。
【図１１】図１０の１３９４インタフェース２６の構成例を示すブロック図である。
【図１２】図１１の１３９４インタフェース２６のより詳細な構成例を示すブロック図である。
【図１３】図１２のLFSR７２のより詳細な構成例を示すブロック図である。
【図１４】図１３のLFSR７２のより具体的な構成例を示すブロック図である。
【図１５】図１０の１３９４インタフェース３６の構成例を示すブロック図である。
【図１６】図１５の１３９４インタフェース３６のより詳細な構成例を示すブロック図である。
【図１７】図１０の１３９４インタフェース４９の構成例を示すブロック図である。
【図１８】図１７の１３９４インタフェース４９のより詳細な構成例を示すブロック図である。
【図１９】図１０のアプリケーション部６１の構成例を示すブロック図である。
【図２０】図１９のアプリケーション部６１のより詳細な構成例を示すブロック図である。
【図２１】図１０の１３９４インタフェース２６の他の構成例を示すブロック図である。
【図２２】図１０の１３９４インタフェース３６の他の構成例を示すブロック図である。
【図２３】図１０の１３９４インタフェース４９の他の構成例を示すブロック図である。
【図２４】図１０のアプリケーション部６１の他の構成例を示すブロック図である。
【図２５】他の認証処理を説明するタイミングチャートである。
【図２６】図２５に続くタイミングチャートである。
【図２７】図２５に続く他のタイミングチャートである。
【図２８】本発明の情報処理システムの他の構成例を示すブロック図である。
【図２９】図２８の乱数発生器９０３の構成例を示すブロック図である。
【図３０】図２８の演算回路９０２の処理を説明するフローチャートである。
【図３１】他の認証処理を説明するタイミングチャートである。
【図３２】パケットフォーマットを示す図である。
【図３３】他の認証処理を説明するタイミングチャートである。
【図３４】 CBCモードの構成例を示すブロック図である。
【図３５】他の認証処理を説明するタイミングチャートである。
【図３６】他の認証処理を説明するタイミングチャートである。
【図３７】他の認証処理を説明するタイミングチャートである。
【図３８】他の認証処理を説明するタイミングチャートである。
【図３９】他の認証処理を説明するタイミングチャートである。
【図４０】他の認証処理を説明するタイミングチャートである。
【図４１】従来の認証方法を説明するタイミングチャートである。
【符号の説明】
１ DVDプレーヤ， ２ パーソナルコンピュータ， ３ 光磁気ディスク装置， １１ １３９４バス， ２０ ファームウェア， ２１ CPU， ２５ ドライブ， ２６ １３９４インタフェース， ２７ EEPROM， ３１ CPU，３５ ドライブ， ３６ １３９４インタフェース， ３７ EEPROM， ４１ CPU， ４７ ハードディスク， ４

８　拡張ボード，　４９　１３９４インタフェース，　５０　EEPROM，　５１　内部バス，　６１　アプリケーション部，　６２　ライセンスマネージャ

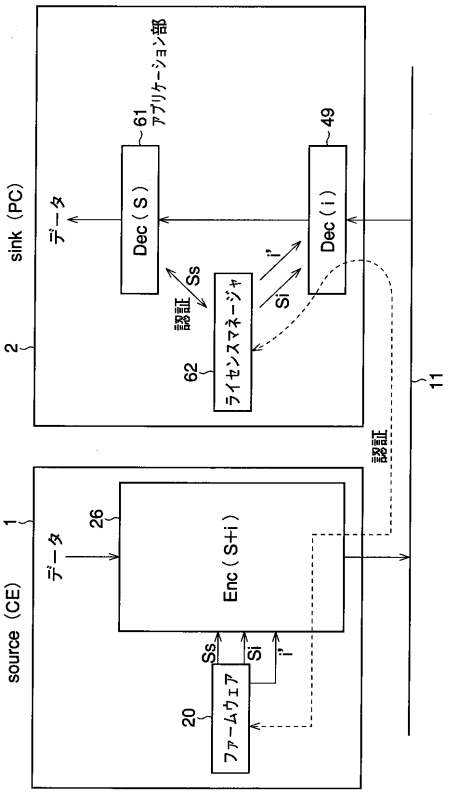
【図１】



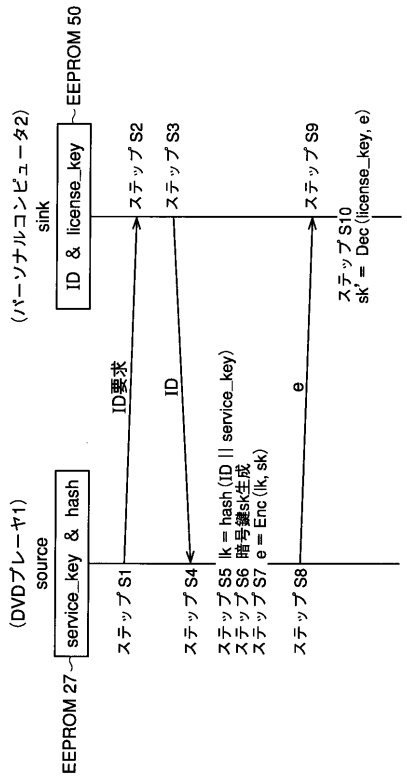

【図２】



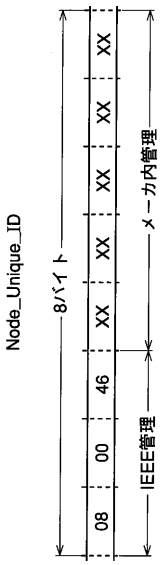
【図３】


source (CE)
1
データ
26
Enc (S+i)
Ss
Si
i'
20
ファームウェア
sink (PC)
2
データ
Dec (S)
61
アプリケーション部
認証
Ss
Si
i'
62
ライセンスマネージャ
Dec (i)
49
認証
11

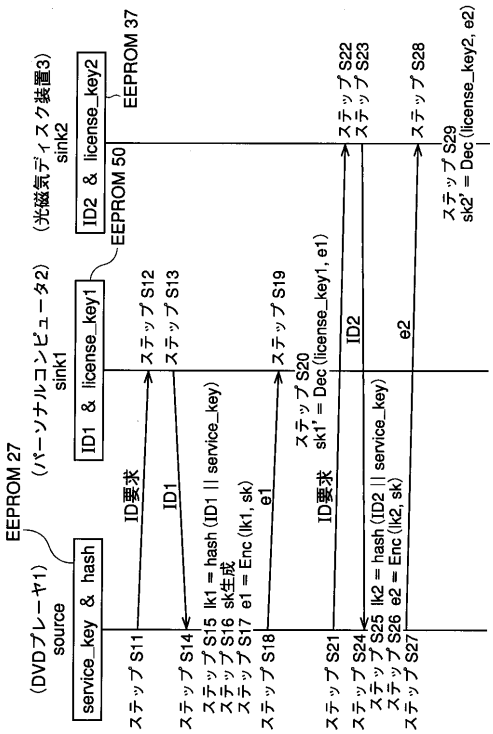
【図４】


(パーソナルコンピュータ2)
sink
ID & license_key
EEPROM 50
(DVDプレーヤ1)
source
service_key & hash
EEPROM 27
ステップ S1
ステップ S2
ステップ S3
ステップ S4
ステップ S5 Ik = hash (ID || service_key)
ステップ S6 暗号鍵sk生成
ステップ S7 e = Enc (Ik, sk)
ステップ S8
ステップ S9
ステップ S10
sk' = Dec (license_key, e)
ID要求
ID
e

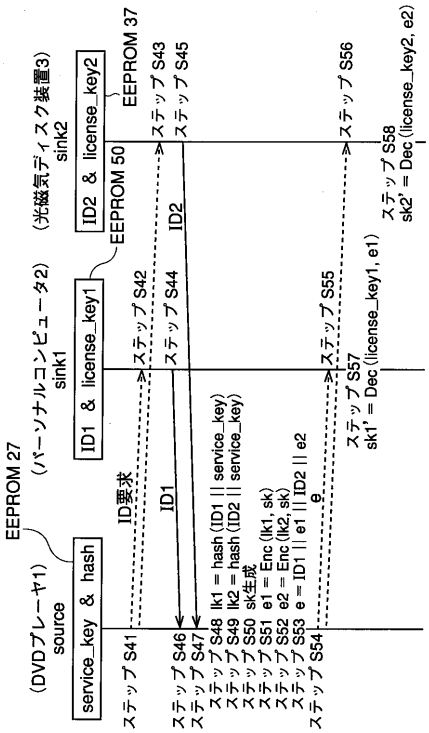
【図５】


Node_Unique_ID
8バイト
08
00
46
XX
XX
XX
XX
XX
IEEE管理
メーカ内管理

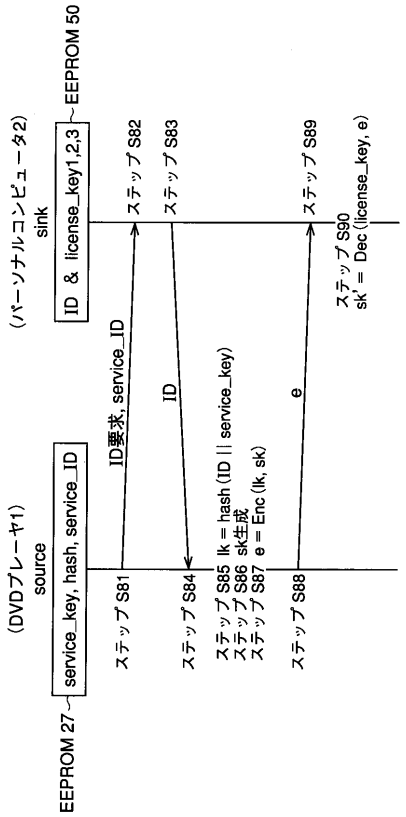
【図６】


(パーソナルコンピュータ2)
sink1
ID1 & license_key1
EEPROM 27
(光磁気ディスク装置3)
sink2
ID2 & license_key2
EEPROM 50
EEPROM 37
(DVDプレーヤ1)
source
service_key & hash
ステップ S11
ステップ S12
ステップ S13
ステップ S14
ステップ S15 Ik1 = hash (ID1 || service_key)
ステップ S16 sk生成
ステップ S17 e1 = Enc (Ik1, sk)
ステップ S18
ステップ S19
ステップ S20
sk1' = Dec (license_key1, e1)
ステップ S21
ステップ S22
ステップ S23
ステップ S24
ステップ S25 Ik2 = hash (ID2 || service_key)
ステップ S26 e2 = Enc (Ik2, sk)
ステップ S27
ステップ S28
ステップ S29
sk2' = Dec (license_key2, e2)
ID1要求
ID1
e1
ID2要求
ID2
e2

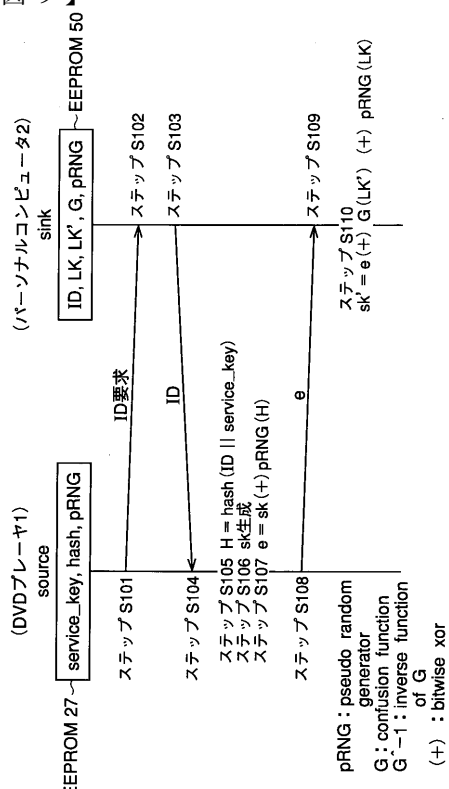
【図７】


(DVDプレーヤ1)
source
service_key & hash
EEPROM 27
(パーソナルコンピュータ2)
sink1
ID1 & license_key1
EEPROM 50
(光磁気ディスク装置3)
sink2
ID2 & license_key2
EEPROM 37
ステップ S41
ID要求
ステップ S42
ステップ S43
ステップ S45
ステップ S46
ID1
ステップ S44
ID2
ステップ S47
ステップ S48 Ik1 = hash (ID1 || service_key)
ステップ S49 Ik2 = hash (ID2 || service_key)
ステップ S50 sk生成
ステップ S51 e1 = Enc (Ik1, sk)
ステップ S52 e2 = Enc (Ik2, sk)
ステップ S53 e = ID1 || e1 || ID2 || e2
ステップ S54
e
ステップ S55
ステップ S56
ステップ S57
sk1' = Dec (license_key1, e1)
ステップ S58
sk2' = Dec (license_key2, e2)

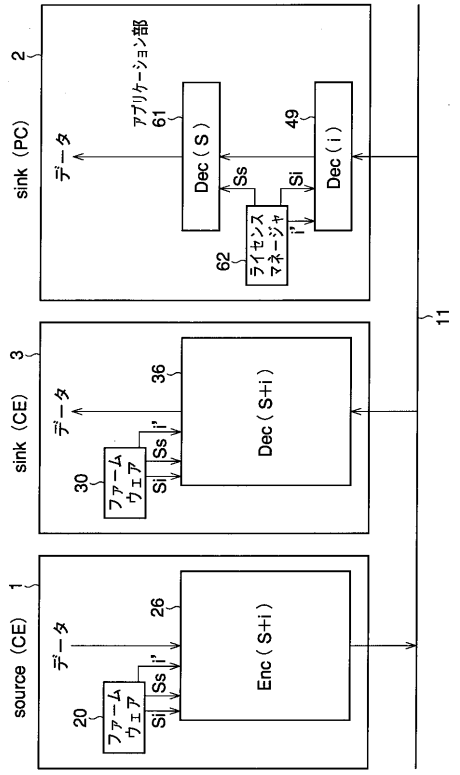
【図８】


(パーソナルコンピュータ2)
sink
ID & license_key1,2,3
EEPROM 50
(DVDプレーヤ1)
source
service_key, hash, service_ID
EEPROM 27
ステップ S81
ID要求, service_ID
ステップ S82
ステップ S83
ID
ステップ S84
ステップ S85 Ik = hash (ID || service_key)
ステップ S86 sk生成
ステップ S87 e = Enc (Ik, sk)
e
ステップ S88
ステップ S89
ステップ S90
sk' = Dec (license_key, e)

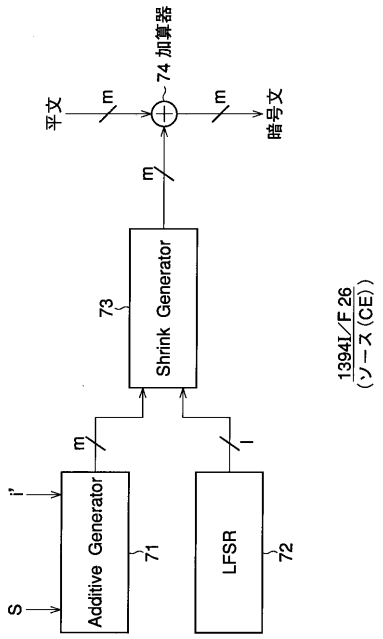
【図９】


(DVDプレーヤ1)
source
service_key, hash, pRNG
EEPROM 27
(パーソナルコンピュータ2)
sink
ID, LK, LK', G, pRNG
EEPROM 50
ステップ S101
ID要求
ステップ S102
ステップ S103
ステップ S104
ID
ステップ S105 H = hash (ID || service_key)
ステップ S106 sk生成
ステップ S107 e = sk (+) pRNG (H)
e
ステップ S108
ステップ S109
ステップ S110
sk' = e (+) G (LK') (+) pRNG (LK)
pRNG : pseudo random generator
G : confusion function
G −1 : inverse function of G
(+) : bitwise xor

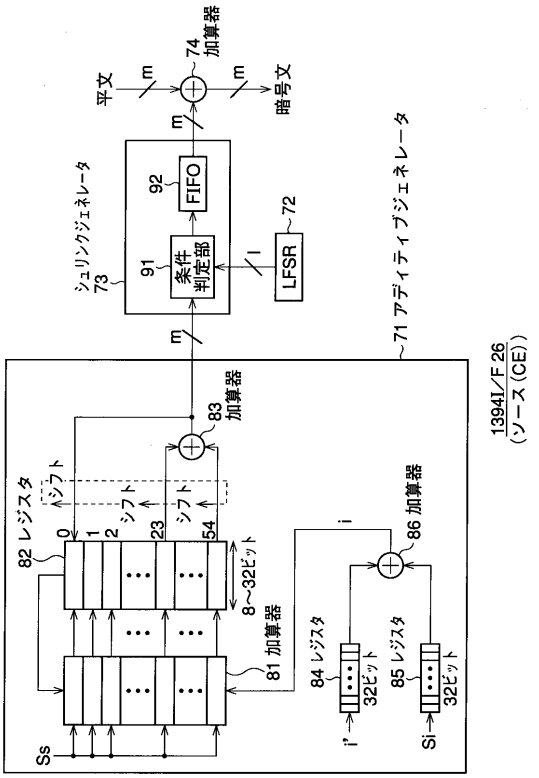
【図１０】


source (CE) 1
データ
20
ファームウェア
Si
Ss
i'
26
Enc ( S+i )
sink (CE) 3
データ
30
ファームウェア
Si
Ss
i'
36
Dec ( S+i )
sink (PC) 2
データ
アプリケーション部
61
Dec ( S )
Ss
62
ライセンスマネージャ
i'
Si
49
Dec ( i )
11

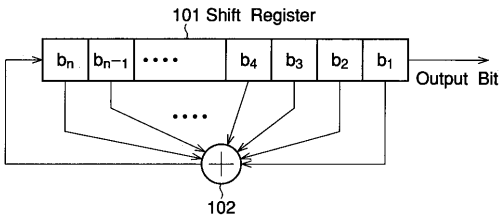
【図１１】




【図１２】

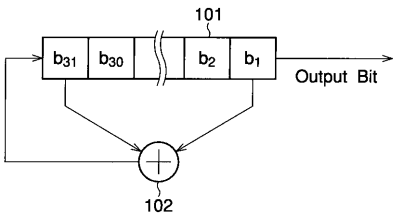



【図１３】



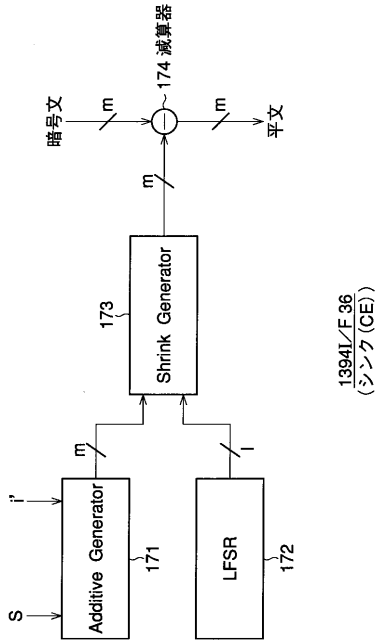

LFSR 72

【図１４】

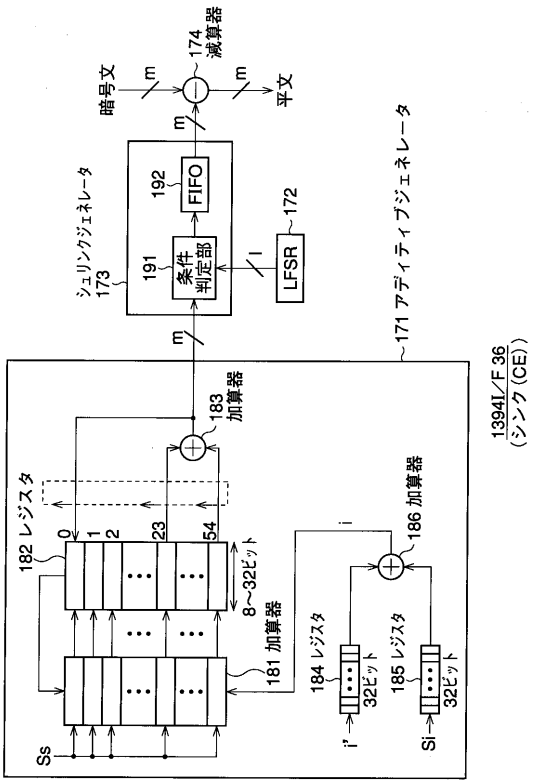



LFSR 72

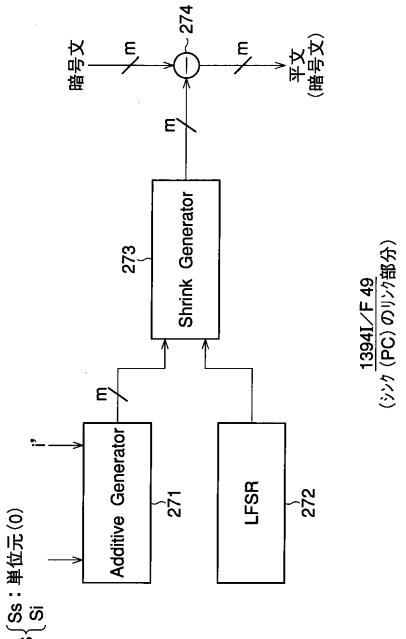
【図１５】



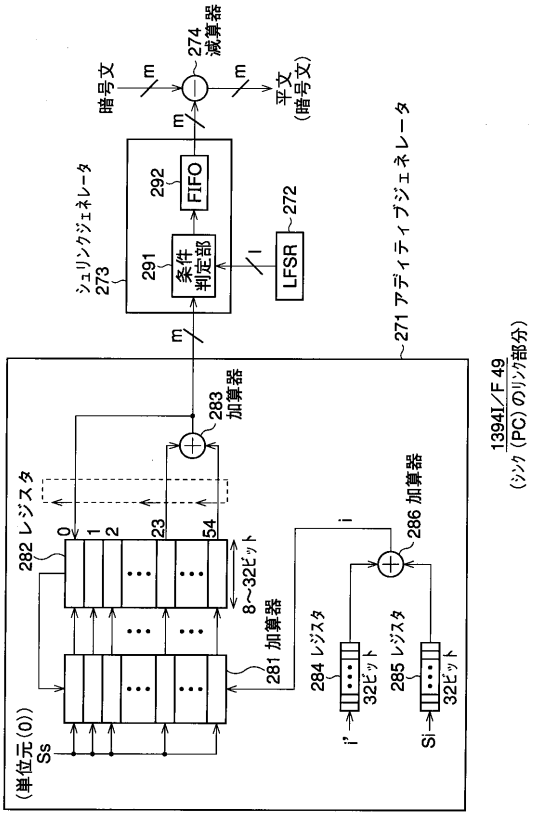
【図１６】



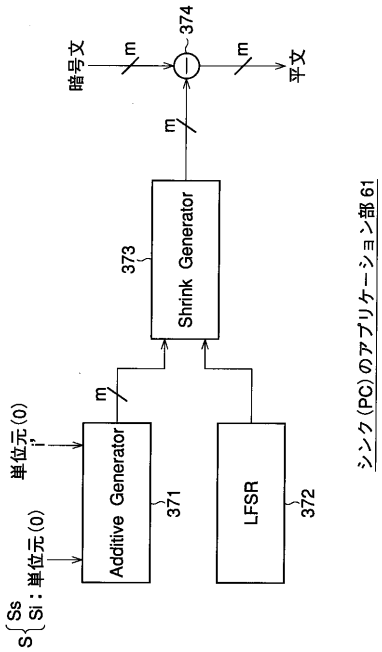

【図１７】



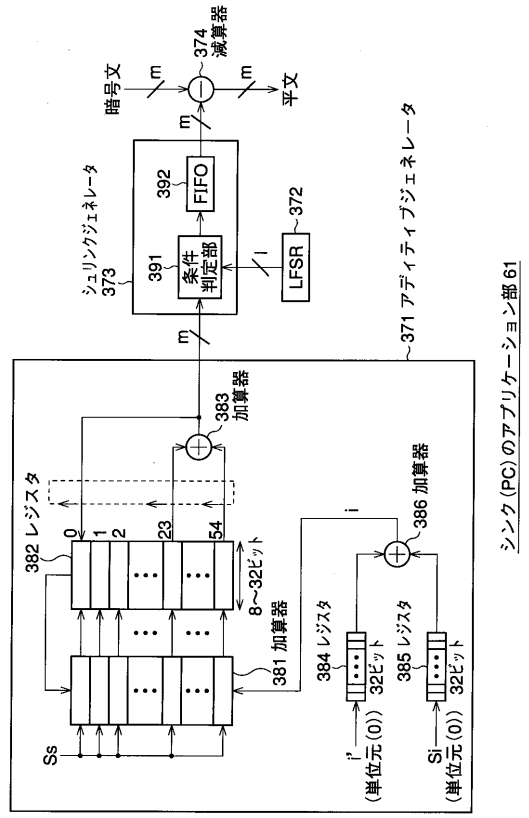

【図１８】



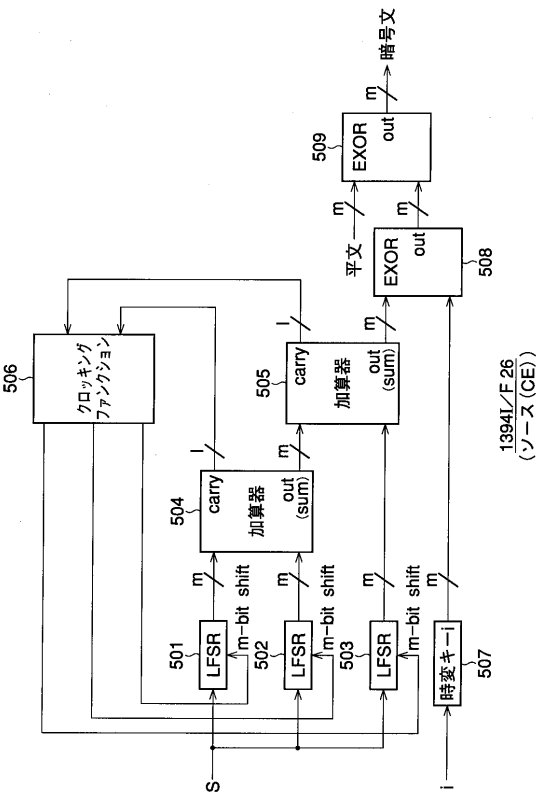

【図１９】



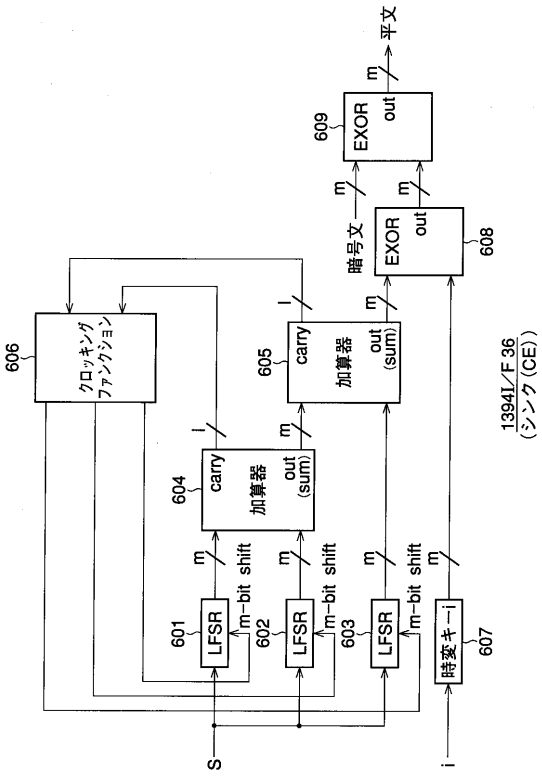
【図２０】



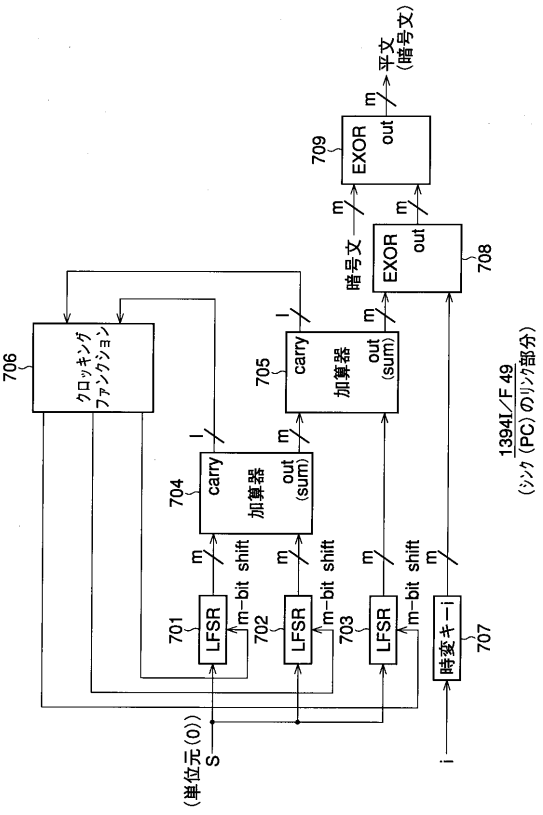

【図２１】



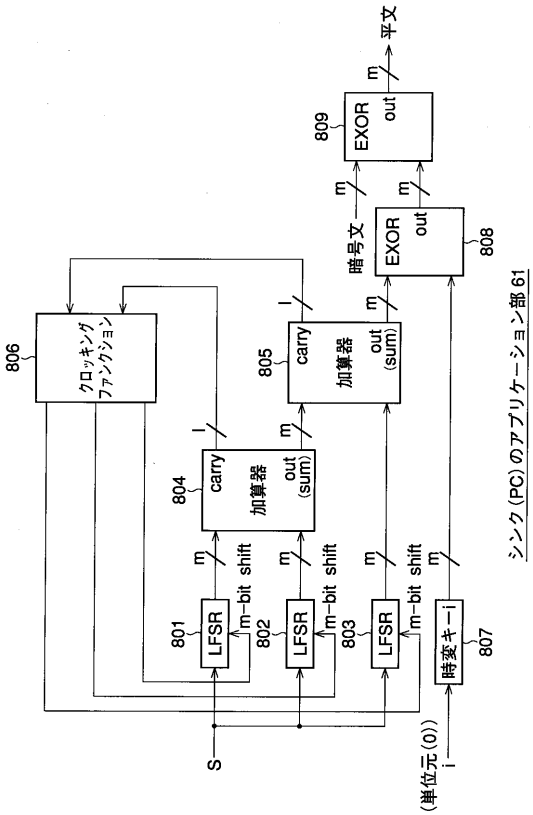

【図２２】



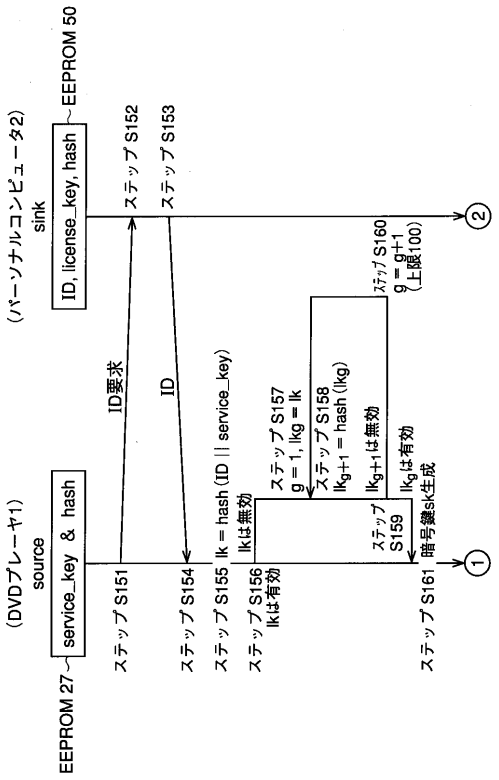

【図２３】



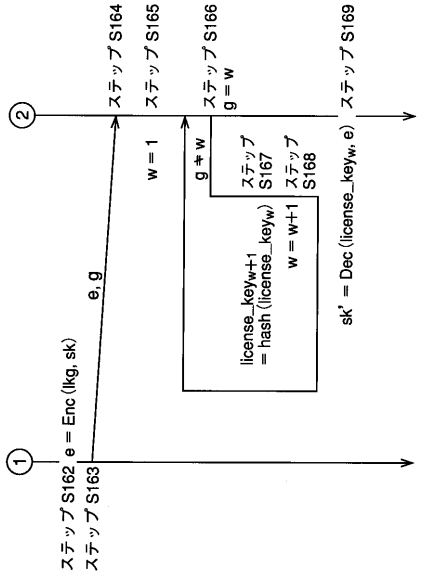
【図２４】



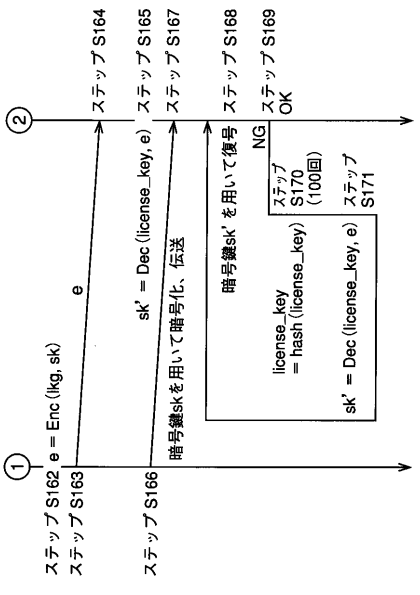

【図２５】



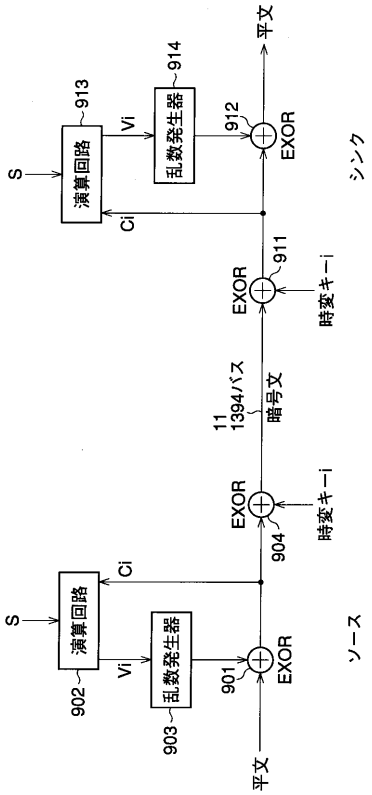

【図２６】



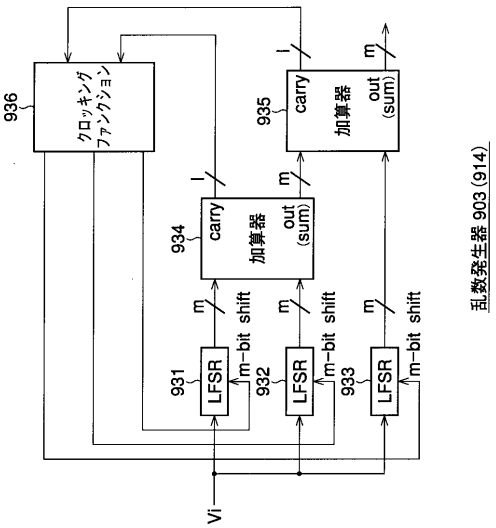

【図２７】



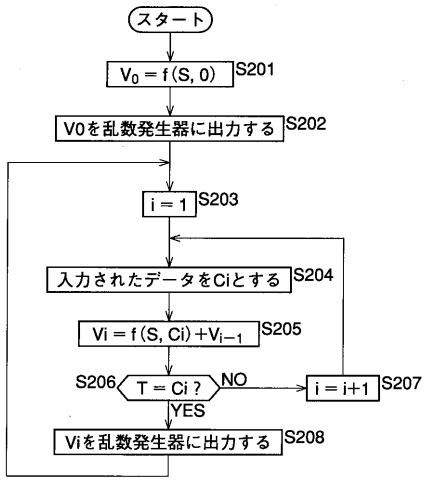
【図２８】



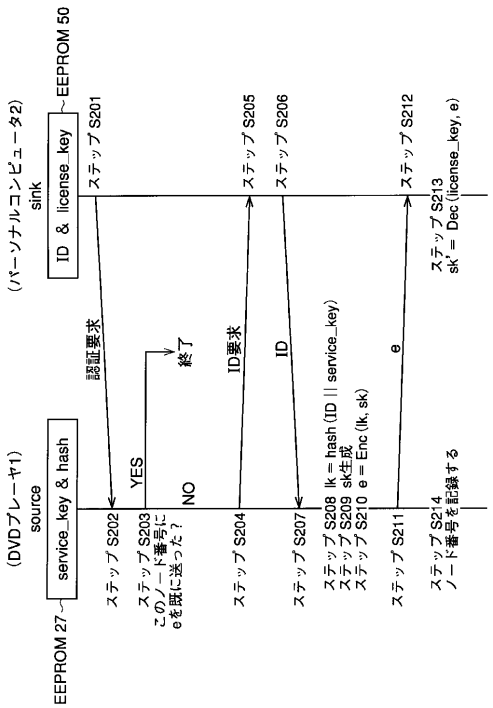

【図２９】



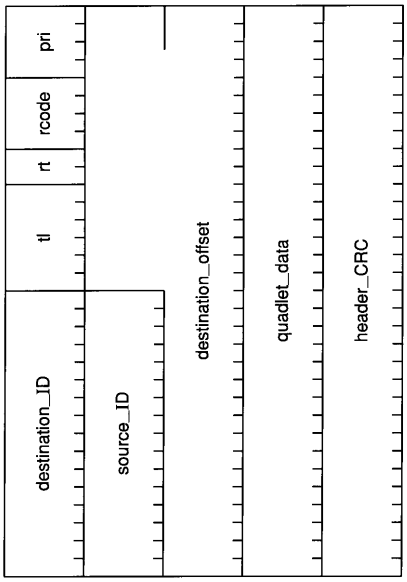

【図３０】



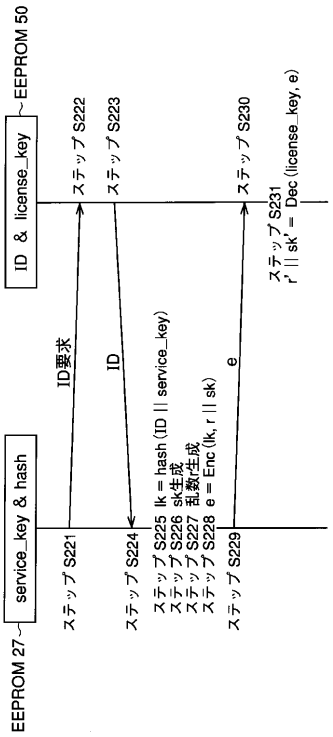

【図３１】



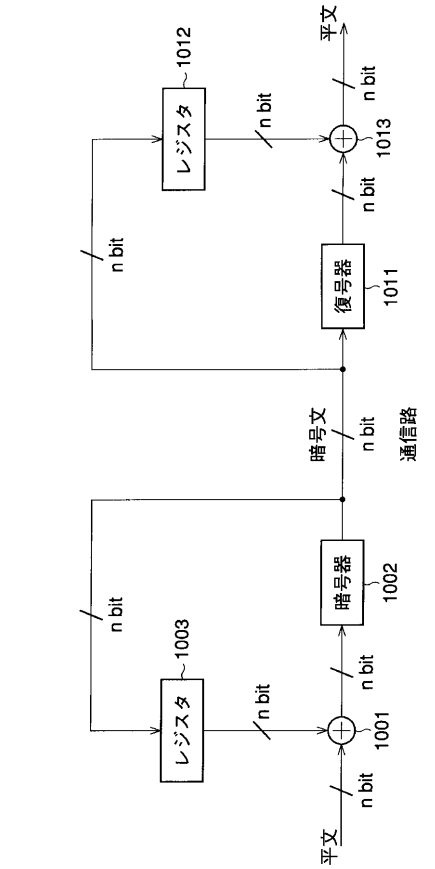
【図３２】



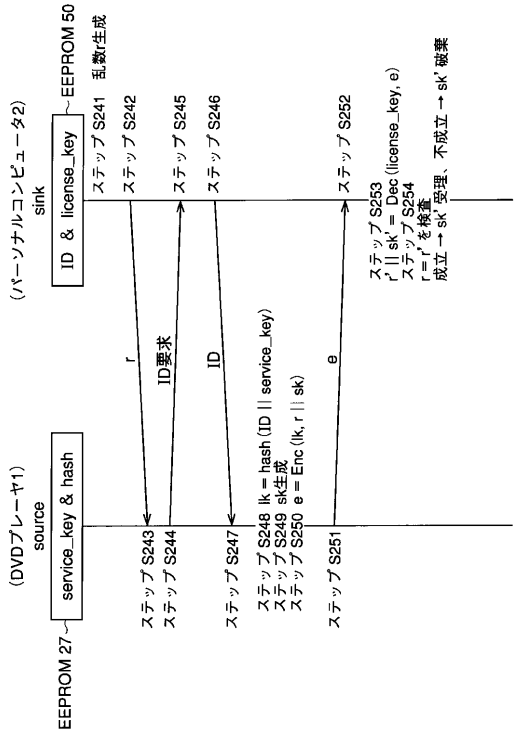

【図３３】



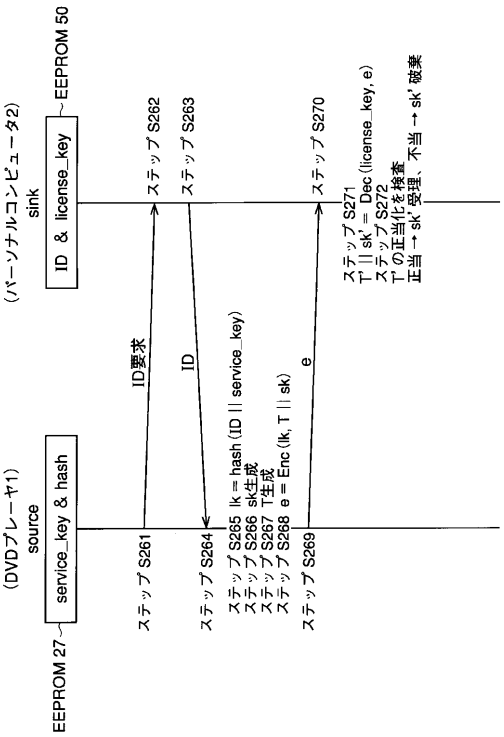

【図３４】



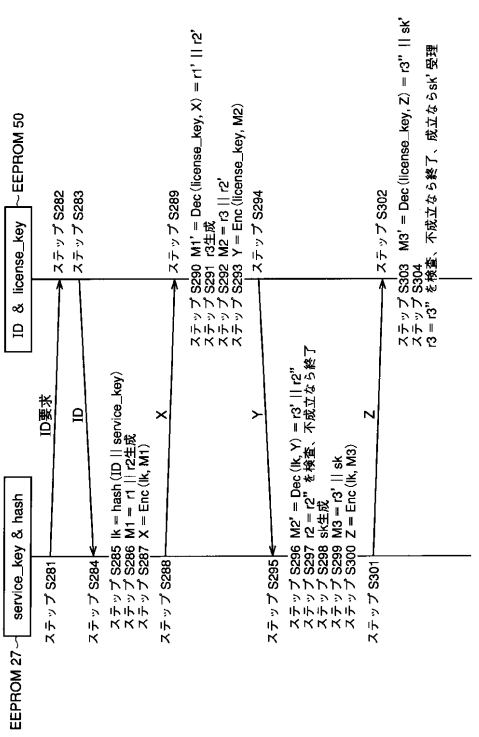

【図３５】



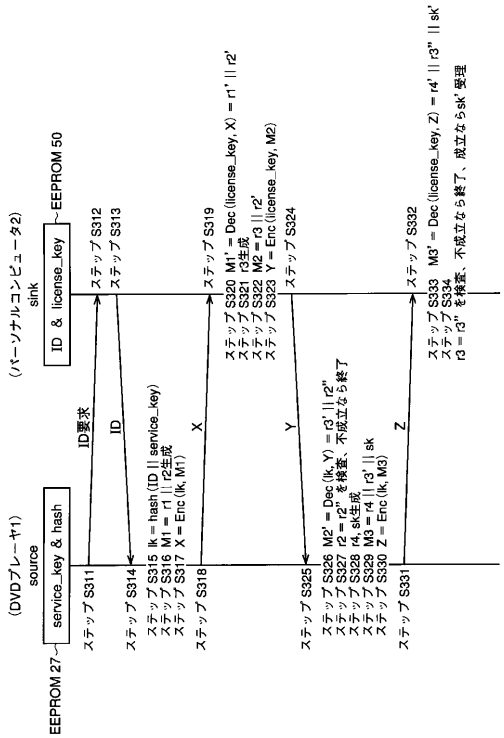
【図３６】



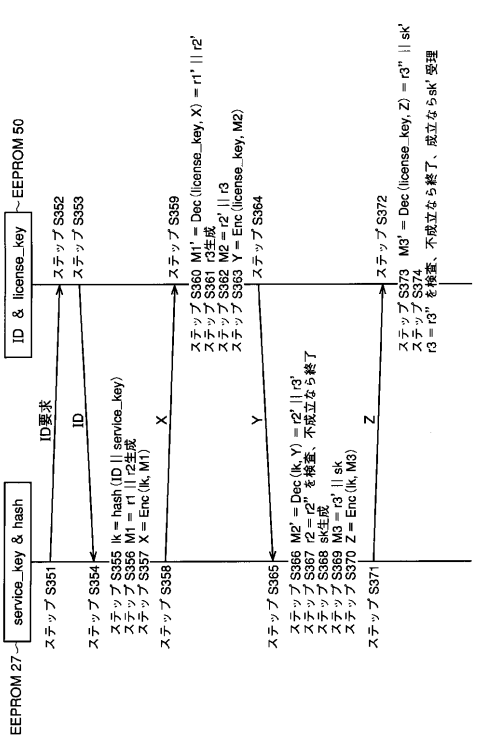

【図３７】



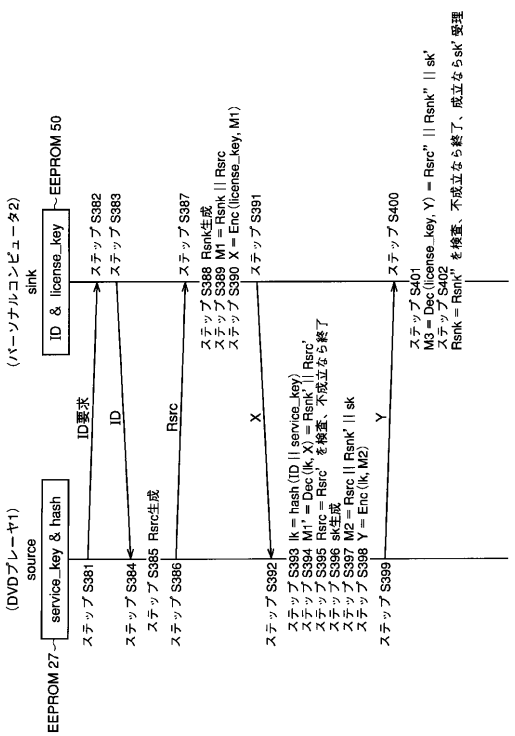

【図３８】



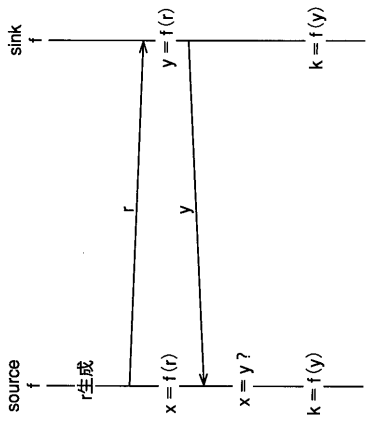

【図３９】



【図４０】


(DVDプレーヤ1)
source
EEPROM 27
service_key & hash
(パーソナルコンピュータ2)
sink
ID & license_key
EEPROM 50
ステップ S381
ID要求
ステップ S382
ステップ S383
ID
ステップ S384
ステップ S385 Rsrc生成
Rsrc
ステップ S386
ステップ S387
ステップ S388 Rsnk生成
ステップ S389 M1 = Rsnk || Rsrc
ステップ S390 X = Enc(license_key, M1)
X
ステップ S391
ステップ S392
ステップ S393 Ik = hash (ID || service_key)
ステップ S394 M1' = Dec (Ik, X) = Rsnk' || Rsrc'
ステップ S395 Rsrc = Rsrc' を検査、不成立なら終了
ステップ S396 sk生成
ステップ S397 M2 = Rsrc || Rsnk' || sk
ステップ S398 Y = Enc (Ik, M2)
Y
ステップ S399
ステップ S400
ステップ S401
M3 = Dec (license_key, Y) = Rsrc" || Rsnk" || sk'
ステップ S402
Rsnk = Rsnk" を検査、不成立なら終了、成立ならsk'受理

【図４１】


sink
f
y = f (r)
k = f (y)
r
y
source
f
r生成
x = f (r)
x = y ?
k = f (y)

フロントページの続き

(72)発明者　佐藤　真
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　嶋　久登
アメリカ合衆国　カリフォルニア州　サラトガ　パセオ・フローレス１２６１０
(72)発明者　中野　雄彦
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　浅野　智之
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

審査官　中里　裕正

(56)参考文献　特開平８－４６９４８（ＪＰ，Ａ）
特開平９－１０７３５０（ＪＰ，Ａ）
特開平１０－２２４３４３（ＪＰ，Ａ）
米国特許第５９４９８７７（ＵＳ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L　9/32
G06F 21/24