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

特許第5779434号
(P5779434)

(45)発行日　平成27年9月16日(2015.9.16)　(24)登録日　平成27年7月17日(2015.7.17)

(51)Int.Cl.　F I
G06F 12/14 (2006.01)　G06F 12/14 510A
H04L 9/08 (2006.01)　H04L 9/00 601C
H04L 9/00 601E

請求項の数 11　(全 25 頁)

(21)出願番号　特願2011-156722(P2011-156722)
(22)出願日　平成23年7月15日(2011.7.15)
(65)公開番号　特開2013-25374(P2013-25374A)
(43)公開日　平成25年2月4日(2013.2.4)
審査請求日　平成26年4月1日(2014.4.1)

(73)特許権者　514315159
株式会社ソシオネクスト
神奈川県横浜市港北区新横浜２丁目１０番２３
(74)代理人　100094525
弁理士　土井　健二
(74)代理人　100094514
弁理士　林　恒徳
(72)発明者　山下　晋
神奈川県横浜市港北区新横浜二丁目１０番２３　富士通セミコンダクター株式会社内

審査官　青木　重徳

最終頁に続く

(54)【発明の名称】セキュリティ装置及びセキュリティシステム

(57)【特許請求の範囲】

【請求項１】

スクランブル演算機能を有するプロセッサと、第１の認証コードが記憶された第１の記憶ユニットとを有し、データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置に接続されるセキュリティ装置であって、

前記第１の認証コードが記憶された第２の記憶ユニットと、

前記乱数を生成する乱数生成ユニットと、

前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、

前記スクランブル演算機能を有するコントローラとを有し、

前記コントローラは、

前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、

前記ホスト装置から、前記第１のスクランブルキーから取得した前記乱数に従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信し、前記スクランブルデータと前記乱数とを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティ装置。

【請求項２】

スクランブル演算機能を有するプロセッサと、第１の認証コードと第２の認証コードとが記憶された第１の記憶ユニットとを有し、データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードと前記第２の認証コードとが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の記憶ユニットの前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
前記ホスト装置から、前記第１のスクランブルキーから取得された前記乱数と前記第１の記憶ユニットの前記第２の認証コードとが前記スクランブル演算されて生成された第２のスクランブルキーに従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信し、前記スクランブルデータと前記第２のスクランブルキーとを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティ装置。

【請求項３】

スクランブル演算機能を有するプロセッサと、第１の認証コードが記憶された第１の記憶ユニットとを有し、データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記乱数とを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置の前記プロセッサに、前記第１のスクランブルキーから取得された前記乱数に従って前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得させるセキュリティ装置。

【請求項４】

スクランブル演算機能を有するプロセッサと、第１の認証コードと第２の認証コードとが記憶された第１の記憶ユニットとを有し、データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードと前記第２の認証コードとが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、

前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の記憶ユニットの前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記第２のスクランブルキーとを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置の前記プロセッサに、前記第１のスクランブルキーから取得された前記乱数と前記第１の記憶ユニットの前記第２の認証コードとが前記スクランブル演算されて生成された前記第２のスクランブルキーに従って、前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得させるセキュリティ装置。
【請求項５】
請求項１乃至４のいずれかにおいて、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の記憶ユニットの前記第１の認証コードによる前記スクランブル演算によって前記乱数が取得されるセキュリティ装置。
【請求項６】
請求項１乃至４のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前回の送受信セッションの乱数との前記スクランブル演算によって生成し、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記前回の送受信セッションの乱数による前記スクランブル演算によって、前記今回の送受信セッションの乱数が取得されるセキュリティ装置。
【請求項７】
請求項１乃至４のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前記第２の記憶ユニットの前記第１の認証コードとの前記スクランブル演算によって生成し、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の記憶ユニットの前記第１の認証コードによる前記スクランブル演算によって、前記今回の送受信セッションの乱数が取得されるセキュリティ装置。
【請求項８】
データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードが記憶された第１の記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、

前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数に従って暗号化対象データをスクランブル演算しスクランブルデータを生成して前記セキュリティ装置に送信し、
前記コントローラは、受信した前記スクランブルデータと前記乱数とを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティシステム。

【請求項９】
データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードと第２の認証コードとが記憶された第１の記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードと第２の認証コードとが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の記憶ユニットの前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数をさらに前記第１の記憶ユニットの前記第２の認証コードと前記スクランブル演算して生成した第２のスクランブルキーに従って、暗号化対象データをスクランブル演算しスクランブルデータを生成して前記セキュリティ装置に送信し、
前記コントローラは、受信した前記スクランブルデータと前記第２のスクランブルキーとを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティシステム。

【請求項１０】
データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードが記憶された第１の記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、

前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記乱数とを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得された前記乱数に従って前記スクランブルデータをスクランブル演算して前記スクランブルデータから前記復号化データを取得するセキュリティシステム。

【請求項１１】
データを暗号化または復号化する暗号化ユニット及び乱数を生成する乱数生成ユニットを備えていないホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードと第２の認証コードとが記憶された第１の記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードと第２の認証コードとが記憶された第２の記憶ユニットと、
前記乱数を生成する乱数生成ユニットと、
前記データを暗号化または復号化し、前記スクランブル演算機能とは異なる暗号機能を有する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第２の記憶ユニットの前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の記憶ユニットの前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記第２のスクランブルキーとを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数と前記第１の記憶ユニットの前記第２の認証コードとを前記スクランブル演算して生成した前記第２のスクランブルキー従って、前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得するセキュリティシステム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、セキュリティ装置及びセキュリティシステムに関する。

【背景技術】
【０００２】
近年、組込み機器のストレージ（外部記憶装置）等に格納されたデータや、回路基板の配線に流れるデータの盗難による情報漏洩が増加しており、セキュリティ機能追加の要望が高まっている。そこで、例えば、組み込み機器の既存システムにデータ暗号化機能を有するセキュリティチップ（セキュリティ装置）を追加搭載することが考えられる。これにより、回路基板上のホストチップ（ホスト装置）からマイクロプロセッサ等のストレージに格納されるデータや回路基板上に流れるデータを、セキュリティチップにより暗号化することができ、情報漏洩を防ぐことができる。
【０００３】

一方で、ホストチップが処理するデータをセキュリティチップで暗号化するためには、ホストチップからセキュリティチップに、配線を介して暗号化対象のデータが転送されることになる。つまり、ホストチップとセキュリティチップ間の配線に暗号化前のデータが流れることになる。そのため、ホストチップとセキュリティチップ間に流れる暗号化前のデータについての盗難による情報漏洩が新たに懸念される。

【０００４】

データの暗号化には、従来から、例えば、ＳＳＬ/ＴＬＳやストリーム暗号方式等が用いられる。例えば、ＳＳＬ／ＴＬＳでは、サーバが公開鍵をクライアントに送信し、クライアントは受信した公開鍵に基づいて乱数を暗号化（ＲＳＡ、ＤＨなど）してサーバに送信する。そして、サーバは、暗号化された乱数を公開鍵と対になる秘密鍵を用いて復号化し乱数を取得する。このように両者は乱数を共有し、共有した乱数を元に共通の暗号アルゴリズム（ＡＥＳ、ＤＥＳなど）に基づいて共通鍵を生成し、その共通鍵に基づいてデータを暗号化し送受信する。

【０００５】

また、ストリーム暗号方式では、送信側（例えば、クライアント）、受信側（例えば、サーバ）はカウンタ列、共通鍵を共有し、ブロック暗号（ＡＥＳ、ＤＥＳなど）を用いて共通のストリーム暗号鍵列（乱数列）をそれぞれ生成する。そして、送信側は、送信対象のデータとストリーム暗号鍵列とを排他的論理和演算（ＸＯＲ/ＥＯＲ)してスクランブル化データを生成し受信側に送信すると共に、受信側は、スクランブル化データとストリーム暗号鍵列とを排他的論理和演算（ＸＯＲ/ＥＯＲ）して元のデータに復号化する。

【先行技術文献】

【特許文献】

【０００６】

【特許文献１】特開２００７－３３６５０６号公報

【特許文献２】特開平１０－２２２４６８号公報

【発明の概要】

【発明が解決しようとする課題】

【０００７】

しかしながら、従来の暗号化技術では、サーバ（受信側）、クライアント（送信側）双方に暗号化手段（ＲＳＡ、ＡＥＳ等）が必要となる。例えば、ＳＳＬ／ＴＬＳでは共通鍵の元になる乱数の共有時、及び共通鍵の生成時に、ストリーム暗号方式ではストリーム暗号鍵列の生成時に、双方に暗号化手段が必要となる。

【０００８】

従って、従来の暗号化技術によると、ホストチップとセキュリティチップ間のデータを暗号化するためには、ホストチップにも暗号化手段が搭載されている必要があり、既存のホストチップにセキュリティチップを搭載するだけでホストチップ内のデータを暗号化するという初期の目的を達成できない。また、暗号化処理が高負荷であることから、ホストチップに暗号化手段を搭載することは、システム全体の負荷を高くしてしまい好ましくない。

【０００９】

そこで、本発明は、ホスト装置に暗号化手段を設けることなく、ホスト装置とセキュリティ装置間の配線に流れるデータの情報漏洩を困難にするセキュリティ装置及びセキュリティシステムを提供する。

【課題を解決するための手段】

【００１０】

第１の側面は、スクランブル演算機能を有するプロセッサと、第１の認証コードが記憶された記憶ユニットとを有するホスト装置に接続されるセキュリティ装置であって、

前記第１の認証コードが記憶された記憶ユニットと、

乱数を生成する乱数生成ユニットと、

データを暗号化または復号化する暗号化ユニットと、

前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
前記ホスト装置から、前記第１のスクランブルキーから取得した前記乱数に従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信し、前記スクランブルデータと前記乱数とを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信する。

【発明の効果】

【００１１】

第１の側面によれば、ホスト装置に暗号化手段を設けることなく、ホスト装置とセキュリティ装置間の配線に流れるデータの情報漏洩を困難にする。

【図面の簡単な説明】

【００１２】

【図１】本実施の形態例における組み込みシステムの半導体装置の一例を表す図である。
【図２】本実施の形態例におけるホストチップとセキュリティチップを表す例図である。
【図３】第１の実施の形態例におけるチップ間の処理の一例を表す第１の図である。
【図４】第１の実施の形態例におけるチップ間の処理の一例を表す第２の図である。
【図５】第１の実施の形態例におけるチップ間の処理の別の例を表す第１の図である。
【図６】第１の実施の形態例におけるチップ間の処理の別の例を表す第２の図である。
【図７】第２の実施の形態例におけるチップ間の処理の一例を表す第１の図である。
【図８】第２の実施の形態例におけるチップ間の処理の一例を表す第２の図である。
【図９】第３の実施の形態例におけるチップ間の処理の一例を表す第１の図である。
【図１０】第３の実施の形態例におけるチップ間の処理の一例を表す第２の図である。

【発明を実施するための形態】

【００１３】

以下、図面にしたがって本発明の実施の形態について説明する。ただし、本発明の技術的範囲はこれらの実施の形態に限定されず、特許請求の範囲に記載された事項とその均等物まで及ぶものである。

【００１４】

図１は、本実施の形態例における組み込みシステムの半導体装置１００の一例を表す図である。同図の半導体装置１００は、例えば、ホストチップ１０、セキュリティチップ２０、その他のユニット３０、外部ストレージ４０、通信ユニット５０を有し、バス線によって接続されている。また、セキュリティチップ２０は、ホストチップ１０とバス線やシリアル線で接続されており、ホストチップ１０から転送されたデータについて、ＲＳＡやＡＥＳ等を用いて暗号化処理及び復号化処理を行う。

【００１５】

例えば、ホストチップ１０で処理されたデータはセキュリティチップ２０に転送され、暗号化データに変換後、外部ストレージ４０や通信ユニット５０、その他のユニット３０に転送される。また、外部の機器等から通信ユニット５０を介して受信された暗号化データや、外部ストレージ４０に格納された暗号化データが、セキュリティチップ２０に転送されて復号化データに変換後、ホストチップ１０に転送され処理が行われる。

【００１６】

図２は、本実施の形態例におけるホストチップ１０とセキュリティチップ２０の構成の一例を表す図である。本実施の形態例におけるホストチップ１０は、例えば、プロセッサ１１、認証コードＡ１、認証コードＡ２、プロセッサ１１の演算処理に用いられるメモリ１２を有する。一方、セキュリティチップ２０は、例えば、コントローラ２１、メモリ２２、認証コードＡ１、認証コードＡ２、乱数生成ユニット２３、暗号化ユニット２４を有する。セキュリティチップ２０の乱数生成ユニット２３は真性乱数または擬似乱数を生成

し、暗号化ユニット２４は、データの高度な暗号化処理及び復号化処理を行う。

【００１７】

また、図２のホストチップ１０のプロセッサ１１及び、セキュリティチップ２０のコントローラ２１は、四則演算や論理演算等の演算機能を有する。また、ホストチップ１０及びセキュリティチップ２０は認証コードＡ１、Ａ２を共有する。認証コードＡ１、Ａ２は、例えば、約１６桁の数値であり、例えば、外部から入力されてもよい。

【００１８】

続いて、まず、第１の実施の形態例の処理について説明する。なお、認証コードＡ２は、第１、第２の実施の形態例では使用されず、第３の実施の形態例でのみ使用される。

【００１９】

［第１の実施の形態例］

図３、図４は、第１の実施の形態例におけるホストチップ１０とセキュリティチップ２０間の処理の一例を表す図である。同図は、ホストチップ１０からセキュリティチップ２０に対して、非暗号化データが送信される場合のデータのスクランブル化について説明する図である。具体的に、非暗号化データはスクランブル化されてホストチップ１０からセキュリティチップ２０に送信され、セキュリティチップ２０によって復元された後、暗号化処理されると共にスクランブル化されホストチップ１０に送信される。本実施の形態例において、ホストチップ１０とセキュリティチップ２０とは、同一の認証コードＡ１を保持している。

【００２０】

ここで、まず、データのスクランブル化演算について説明する。本実施の形態例では、スクランブル演算として排他的論理和（以下、ＸＯＲ）演算を使用する。ＸＯＲ演算は、ホストチップ１０のプロセッサ１１、及び、セキュリティチップ２０のコントローラ２１に基本的に備えられる演算処理機能に含まれる。ＸＯＲ演算によるスクランブル演算では、対象の値がキーとなる値に従ってスクランブル化される。具体的に、対象の値Ａはキーである値ＸとＸＯＲ演算されることによりスクランブル化され、値Ｂが生成される。そして、値Ｂが、再度、値ＸとＸＯＲ演算されることによって値Ａが復元される。つまり、データ（値Ａ）とキー（値Ｘ）とのＸＯＲ演算の結果、生成されたスクランブルデータ（値Ｂ）が、再度、キー（値Ｘ）に従ってＸＯＲ演算されることにより、元のデータ（値Ａ）が復元される。

【００２１】

ＸＯＲ演算によるスクランブル化では、スクランブルデータから元のデータを復元するためには、キーが必要となる。しかし、第３者はキーを知らないため、スクランブル化データに基づいて元のデータを復元することができない。なお、スクランブル化演算は、ＸＯＲ演算に限定されるものではない。スクランブル演算は、元のデータを復元可能な演算であれば、例えば、四則演算等の他の演算によって行われてもよいし、ＸＯＲ演算や四則演算等の組合せによって行われてもよい。

【００２２】

図３に戻り、初めに、ホストチップ１０は、セキュリティチップ２０に処理の開始を知らせる初期化コマンドを発行する（ａ）。セキュリティチップ２０は、初期化コマンドを受信すると、乱数Ｒ１を生成する（ｂ）。続いて、セキュリティチップ２０の乱数生成ユニット２３は、乱数Ｒ１と認証コードＡ１とをＸＯＲ演算してスクランブルキーＳ１を生成する（ｃ）。そして、セキュリティチップ２０は、初期化処理の結果と共に、生成したスクランブルキーＳ１をホストチップ１０に送信する（ｄ）。

【００２３】

続いて、ホストチップ１０では、スクランブルキーＳ１を受信すると、スクランブルキーＳ１を、予め保持している認証コードＡ１に従ってＸＯＲ演算し、乱数Ｒ１を取得する（ｅ）。これにより、認証コードＡ１に加えて乱数Ｒ１が、ホストチップ１０とセキュリティチップ２０間で共有される。そこで、ホストチップ１０は、暗号化対象のデータＤ１と共有した乱数Ｒ１とをＸＯＲ演算してスクランブル化データＸ１を生成し（ｆ）、暗号

化命令と共にセキュリティチップ２０に送信する（ｇ）。
【００２４】
セキュリティチップ２０は、暗号化命令とスクランブル化データＸ１とを受信すると、スクランブル化データＸ１を乱数R1に従ってＸＯＲ演算して、データＤ１を取得する（ｈ）。そこで、セキュリティチップ２０の暗号化ユニット２４は、データＤ１を暗号化処理し暗号化データＥ１を生成する（ｉ）。続いて、セキュリティチップ２０は、暗号化データＥ１と乱数R1とをＸＯＲ演算し、スクランブル化暗号データＹ１を生成する（ｊ）。
【００２５】
続いて、図４に移り、セキュリティチップ２０は、次の新たなスクランブル化用乱数である乱数Ｒ２を生成する（ｋ）。そして、セキュリティチップ２０は、生成した乱数Ｒ２と乱数R1とをＸＯＲ演算し、スクランブルキーＳ２を生成する（ｌ）。続いて、セキュリティチップ２０は、図３の処理ｊで生成したスクランブル化暗号データＹ１と共に、スクランブルキーＳ２をホストチップ１０に送信する（ｍ）。
【００２６】
ホストチップ１０は、スクランブル化暗号データＹ１とスクランブルキーＳ２とを受信すると、スクランブル化暗号データＹ１を乱数R1に従ってＸＯＲ演算して、暗号化データＥ１を取得する（ｎ）。また、ホストチップ１０は、受信したスクランブルキーＳ２を乱数Ｒ１に従ってＸＯＲ演算し、乱数Ｒ２を取得する（ｏ）。これにより、暗号化対象のデータＤ１に対応する暗号化データＥ１がホストチップ１０で取得されると共に、新たな乱数Ｒ２がホストチップ１０とセキュリティチップ２０間で共有される。
【００２７】
そして、暗号化する対象のデータＤ２がさらにある場合、同様にして、ホストチップ１０は、暗号化対象のデータＤ２と、新たな乱数Ｒ２とをＸＯＲ演算し、スクランブル化データＸ２を生成し（ｐ）、セキュリティチップ２０に送信する（ｑ）。セキュリティチップ２０は、受信したスクランブル化データＸ２を乱数Ｒ２に従ってＸＯＲ演算し、暗号化対象のデータＤ２を取得する（ｒ）。前回の暗号化対象データＤ１と同様にして、データＤ２は暗号化ユニット２４によって暗号化データＥ２に変換される。そして、暗号化データＥ２は、乱数Ｒ２とＸＯＲ演算され、スクランブル化暗号データＹ２が生成される。そして、新たなスクランブル化用乱数である乱数Ｒ３（図示せず）が生成され、スクランブル化暗号データＹ２と共に、ホストチップ１０に送信される。
【００２８】
このようにして、セキュリティチップ２０は、ホストチップ１０とセキュリティチップ２０間で共有した乱数に従ってデータをスクランブル化し、送受信されるデータの漏洩を防ぐ。また、乱数は、一定数の送受信セッション毎に変更されると共に、前回のセッションの乱数（初回は認証コードＡ１）に従ってスクランブル化されてホストチップ１０に送信され、ホストチップ１０で当該乱数に従って復元されることによって共有される。
【００２９】
ところで、上述した図３、図４の例では、ホストチップ１０からセキュリティチップ２０に暗号化対象のデータが送信される場合を例示したが、次に、ホストチップ１０からセキュリティチップ２０に復号化対象のデータが送信される場合について説明する。
【００３０】
図５、図６は、ホストチップ１０からセキュリティチップ２０に暗号化されたデータが送信され、セキュリティチップ２０で復号化される場合のデータのスクランブル化について説明する図である。図３、図４と同様にして、ホストチップ１０及びセキュリティチップ２０は、同一の認証コードＡ１を保持している。
【００３１】
図５において、図３と同様にして、ホストチップ１０とセキュリティチップ２０間で乱数Ｒ１を共有すると、ホストチップ１０は、暗号化データＥ１と乱数Ｒ１とをＸＯＲ演算し、スクランブル化暗号データＹ１を生成し（ｆ）、復号化命令と共にセキュリティチッ

プ２０に送信する（ｇ）。そして、セキュリティチップ２０では、受信したスクランブル化暗号データＹ１を、乱数Ｒ１に従ってＸＯＲ演算して、暗号化データＥ１を取得する（ｈ）。そして、セキュリティチップ２０の暗号化ユニット２４は、暗号化データＥ１を復号化してデータＤ１を生成する。

【００３２】

続いて、セキュリティチップ２０は、データＤ１のスクランブル化処理の前に、新たな乱数Ｒ２を生成する（ｊ）。そして、セキュリティチップ２０は、（図３のように）乱数Ｒ１ではなく、乱数Ｒ２とデータＤ１とをＸＯＲ演算して、スクランブル化データＸ１を生成する（ｋ）。復号化された非暗号化データ（例えば、Ｄ１）は、暗号化データ（例えば、Ｅ１）に対して、データが解析された場合のリスクが高いためである。そのため、セキュリティチップは、非暗号化データを、既にスクランブル化に使用された乱数Ｒ１ではなく、新たに生成されスクランブル化に使用されていない乱数Ｒ２に従ってスクランブル化する。これにより、非暗号化データの復元がより困難になる。ただし、この例に限定されるものではなく、処理ｋにおいて、セキュリティチップ２０は、非暗号化データＤ１を、乱数Ｒ１に従ってスクランブル化してもよい。

【００３３】

図６に移り、続いて、セキュリティチップ２０は、乱数１と乱数２とをＸＯＲ演算して、スクランブルキーＳ２を生成し（ｌ）、スクランブル化データＸ１と共に、ホストチップ１０に送信する（ｍ）。ホストチップ１０は、スクランブルキーＳ２を、乱数Ｒ１に従ってＸＯＲ演算し、乱数Ｒ２を取得する（ｏ）。そして、ホストチップ１０は、スクランブル化データＸ１を、乱数Ｒ２に従ってＸＯＲ演算し、データＤ１を取得する（ｎ）。これにより、暗号化データＥ１が復号化処理されたデータＤ１がホストチップ１０で取得されると共に、ホストチップ１０とセキュリティチップ２０間で乱数Ｒ２が共有される。そして、復号化する対象の暗号化データＥ２がさらにある場合、ホストチップ１０は、暗号化データＥ２と乱数Ｒ２とをＸＯＲ演算し、スクランブル化暗号データＹ２を生成する（ｐ）。この後の処理は、図５、図６の処理ｇ～処理ｎと同様である。

【００３４】

以上のように、本実施の形態例におけるセキュリティチップ（セキュリティ装置）２０は、データを暗号化する場合、乱数生成ユニット２３が生成した乱数と認証コードとをスクランブル演算してスクランブルキーを生成してホストチップ（ホスト装置）１０に送信し、ホストチップ１０から、スクランブルキーから取得した乱数に従って暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信する。そして、セキュリティチップ２０は、スクランブルデータと乱数とをスクランブル演算して暗号化対象データを生成し、暗号化対象データを暗号化ユニット２４によって暗号化して暗号化データを生成し、ホストチップ１０に送信する。

【００３５】

一方、本実施の形態例におけるセキュリティチップ２０は、データを復号化する場合、乱数生成ユニット２３が生成した乱数と認証コードとをスクランブル演算してスクランブルキーを生成して前記ホストチップ１０に送信すると共に、暗号化データを暗号化ユニット２４によって復号化して復号化データを生成し、復号化データと乱数とをスクランブル演算してスクランブルデータを生成しホストチップ１０に送信する。そして、セキュリティチップ２０は、ホストチップ１０のプロセッサ１１に、スクランブルキーから取得された乱数に従ってスクランブルデータをスクランブル演算させ、スクランブルデータから復号化データを取得させる。

【００３６】

また、ホストチップ１０では、乱数は、初回は認証コードに２回目以降の各送受信セッションでは前回のセッションの乱数に従って、スクランブルキーがスクランブル演算されることによって取得される。２回目以降の各送受信セッションでは、セキュリティチップ２０は、今回の送受信セッションの乱数と前回の送受信セッションの乱数とのスクランブル演算によってスクランブルキーを生成してホストチップ１０に送信し、ホストチップ１

０では、当該スクランブルキーが、前回の送受信セッションの乱数に従ってスクランブル演算されることにより、今回の送受信セッションの乱数が取得される。

【００３７】

これにより、本実施の形態例におけるセキュリティチップ２０は、ホストチップ１０に高度な暗号化処理を行う暗号化ユニット２４を要することなく、ホストチップ１０との間のデータをスクランブル化して、データの漏洩を防ぐことができる。また、セキュリティチップ２０は、データのスクランブル化に高度な暗号化処理を用いないため、システムの負荷を抑え、スクランブル化による通信速度の低下を防ぐことができる。

【００３８】

また、データのスクランブル化は、ホストチップ１０及びセキュリティチップ２０で共有された乱数に従って行われる。そのため、第３者は、スクランブルデータが乱数に従ってスクランブル化されていることを認識していた場合であっても、乱数の値を知らないためスクランブル化データを復元することができない。また、当該乱数は、セキュリティチップ２０で前回の乱数（初回は、認証コード）に従ってスクランブル化されてホストチップ１０に送信され、ホストチップ１０で前回の乱数（初回は、認証コード）に従って復元されることにより共有される。このため、第３者が、スクランブルキーを取得した場合でも、前回の乱数（初回は認証コード）を検知しない限り、スクランブルキーから乱数の値を復元することができない。このように、本実施の形態例におけるセキュリティチップ２０は、仮に、第３者がデータのスクランブル化手順を認識していた場合であっても、スクランブル化データに基づくデータの復元を困難にする。

【００３９】

また、本実施の形態例におけるセキュリティチップ２０によると、乱数を生成する乱数生成ユニット２３はセキュリティチップ２０のみに備えられていればよく、ホストチップ１０は乱数生成ユニット２３を有している必要がない。つまり、本実施の形態例におけるホストチップ１０は、予め備えられている演算機能に加えてメモリ１２等に認証コードを有していればよく、新たに暗号化ユニット２４及び乱数生成ユニット２３を備える必要がない。従って、ホストチップ１０を有する既存システムの回路基板に、本実施の形態例におけるセキュリティチップ２０が搭載されることによって、当該ホストチップ１０とセキュリティチップ２０間のデータのスクランブル化が可能になり、データの漏洩が防止される。

【００４０】

さらに、本実施の形態例において、データのスクランブル化に使用される乱数は一定数の送受信セッション毎に変更される。これにより、仮に第３者によってある乱数が解析された場合であっても、一定数のセッションの後、データをスクランブル化する乱数は変更される。このため、本実施の形態例におけるセキュリティチップ２０は、スクランブルデータの継続した解析を困難にし、データの漏洩範囲を最小限にすることができる。なお、本実施の形態例において、乱数の変更頻度は１回の送受信セッション毎であるが、この頻度に限定されるものではない。乱数の変更頻度は、複数回の送受信セッション毎であってもよいし、１回の送信または受信処理毎であってもよい。

【００４１】

ところで、本実施の形態例では、暗号化された暗号化データＥ１、Ｅ２についてもスクランブル化の対象とされる。しかしながら、暗号化データは、仮にスクランブル化された状態から復元された場合であっても、暗号化されていることによりデータ内容が検知されるリスクは低い。そのため、暗号化データは、必ずしもスクランブル化の対象にされる必要はなく、少なくとも復元された場合のリスクの高い非暗号化のデータについてのみ、スクランブル化の対象にされればよい。

【００４２】

また、本実施の形態例におけるセキュリティチップ２０は、乱数は一定の送受信セッション毎に変更するものの、認証コードについては変更しない。セキュリティチップ２０及びホストチップ１０において、例えば、一時的に生成され使用される乱数はＲＡＭ等のメ

モリに格納され、認証コードについては不揮発性メモリ等のメモリに格納される。不揮発性メモリでは、電源供給が不安定な場合や書き込み限度回数を有する場合、更新時にエラーやデータ破損が発生することがある。

【００４３】

しかし、本実施の形態例におけるセキュリティチップ２０は認証コードについては変更しないことから、認証コードの変更に係る不揮発性メモリ更新時における認証コードの破損及び損失の懸念を回避することができる。そして、例え電源供給が不安定であることに起因してＲＡＭ等に格納された乱数が失われた場合であっても、セキュリティチップ２０は、新たな初期化コマンド（図３、図５のａ）に応答して、認証コードと新たに生成する乱数に従って、データのスクランブル化処理を再開することができる。

【００４４】

［第２の実施の形態例］

第２の実施の形態例におけるセキュリティチップ２０は、新たに生成した乱数の共有時におけるスクランブル化において、前回のセッションの乱数ではなく認証コードに従って、乱数をスクランブル化する。以下、図に従って説明する。

【００４５】

図７、図８は、第２の実施の形態例におけるホストチップ１０とセキュリティチップ２０の処理の一例を表す図である。同７、図８は、図３、図４と同様に、ホストチップ１０からセキュリティチップ２０に非暗号化データが送信される場合のデータのスクランブル化について説明する図である。第１の実施の形態例と同様にして、ホストチップ１０及びセキュリティチップ２０は、同一の認証コードＡ１を保持している。

【００４６】

図７の処理は、第１の実施の形態例における図３の処理と同様である。ホストチップ１０とセキュリティチップ２０間で乱数Ｒ１が共有され、ホストチップ１０から、暗号化対象のデータＤ１がスクランブル化されたスクランブル化データＸ１が、セキュリティチップ２０に送信される（ｇ）。セキュリティチップ２０は、スクランブル化データＸ１から暗号化対象のデータＤ１を取得し暗号化する。暗号化された暗号化データＥ１は、乱数Ｒ1に従ってスクランブル化される（ｊ）。

【００４７】

続いて、図８に移り、セキュリティチップ２０は、次の新たなスクランブル化用乱数である乱数Ｒ２を生成し、スクランブル化する（ｌ）。ただし、本実施の形態例では、セキュリティチップ２０は、生成した乱数Ｒ２を、乱数Ｒ1ではなく認証コードＡ１に従ってＸＯＲ演算し、スクランブルキーＳ２を生成する。スクランブルキーＳ２は、スクランブル化暗号データＹ１と共に、ホストチップ１０に送信される（ｍ）。

【００４８】

そして、ホストチップ１０は、スクランブル化暗号データＹ１を乱数Ｒ1に従ってＸＯＲ演算して暗号化データＥ１を取得すると共に（ｎ）、スクランブルキーＳ２を、乱数Ｒ１ではなく認証コードＡ１に従ってＸＯＲ演算し、乱数Ｒ２を取得する（ｏ）。これにより、暗号化対象のデータＤ１の暗号化データＥ１がホストチップ１０で取得されると共に、乱数Ｒ２がホストチップ１０とセキュリティチップ２０間で共有される。この後の処理は、図４の処理ｐから処理ｒと同様である。

【００４９】

なお、図７、図８の例では、ホストチップ１０からセキュリティチップ２０に暗号化対象のデータが送信される場合を例示したが、ホストチップ１０からセキュリティチップ２０に復号化対象のデータが送信される場合についても同様である。この場合についても、セキュリティチップ２０で新たに生成された乱数は認証コードＡ１に従ってスクランブル化されてホストチップ１０に送信され、ホストチップ１０で認証コードＡ１に従って復元されることによって共有される。そして、セキュリティチップ２０で復号化されたデータは、乱数に従ってスクランブル化されてホストチップ１０送信され、ホストチップ１０で乱数に従ってスクランブル演算されることにより復元される。

【００５０】
以上のように、本実施の形態例におけるセキュリティチップ２０は、データを暗号化する場合、乱数生成ユニット２３が生成した乱数と認証コードとをスクランブル演算してスクランブルキーを生成してホストチップ１０に送信し、ホストチップ１０から、スクランブルキーから取得した乱数に従って暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信する。そして、セキュリティチップ２０は、スクランブルデータと乱数とをスクランブル演算して暗号化対象データを生成し、暗号化対象データを暗号化ユニット２４によって暗号化して暗号化データを生成し、ホストチップ１０に送信する。

【００５１】
一方、本実施の形態例におけるセキュリティチップ２０は、データを復号化する場合、乱数生成ユニット２３が生成した乱数と認証コードとをスクランブル演算してスクランブルキーを生成して前記ホストチップ１０に送信すると共に、暗号化データを暗号化ユニット２４によって復号化して復号化データを生成し、復号化データと乱数とをスクランブル演算してスクランブルデータを生成しホストチップ１０に送信する。そして、セキュリティチップ２０は、ホストチップ１０のプロセッサ１１に、スクランブルキーから取得された乱数に従ってスクランブルデータをスクランブル演算させ、スクランブルデータから復号化データを取得させる。

【００５２】
また、ホストチップ１０では、乱数は、スクランブルキーが認証コードに従ってスクランブル演算されることによって取得される。

【００５３】
これにより、本実施の形態例におけるセキュリティチップ２０は、ホストチップ１０に暗号化ユニット２４を要することなく、ホストチップ１０との間のデータをスクランブル化して、データの漏洩を防ぐことができる。また、セキュリティチップ２０は、データのスクランブル化に高度な暗号処理を用いないため、システムの負荷を抑え、スクランブル化による通信速度の低下を防ぐことができる。

【００５４】
また、データのスクランブル化は、ホストチップ１０及びセキュリティチップ２０間で共有された乱数に従って行われる。そのため、第３者は、スクランブルデータが乱数に従ってスクランブル化されていることを認識していた場合であっても、乱数の値を知らないためスクランブル化データを復元することができない。そして、本実施の形態例において、当該乱数は、セキュリティチップ２０で認証コードＡ１に従ってスクランブル化されホストチップ１０に送信され、ホストチップ１０で認証コードＡ１に従って復元されることにより共有される。このため、第３者が、スクランブルキーを取得した場合でも、認証コードＡ１を検知しない限り、スクランブルキーから乱数の値を復元することができない。このように、本実施の形態例におけるセキュリティチップ２０は、仮に、第３者がスクランブル化手順を認識していた場合であっても、スクランブル化データに基づくデータの復元を困難にする。

【００５５】
また、本実施の形態例におけるセキュリティチップ２０によると、乱数を生成する乱数生成ユニット２３はセキュリティチップ２０のみに備えられていればよく、ホストチップ１０は乱数生成ユニット２３を有している必要がない。従って、ホストチップ１０を有する既存システムの回路基板に、本実施の形態例におけるセキュリティチップ２０が搭載されることによって、当該ホストチップ１０とセキュリティチップ２０間のデータのスクランブル化が可能になり、データの漏洩が防止される。

【００５６】
さらに、データのスクランブル化に使用される乱数は一定数の送受信セッション毎に変更される。そのため、本実施の形態例におけるセキュリティチップ２０は、スクランブルデータの継続した解析を困難にし、データの漏洩範囲を最小限にすることができる。

【００５７】
［第３の実施の形態例］
第３の実施の形態例におけるセキュリティチップ２０は、２つの認証コードを有する（以下、第１の認証コード、第２の認証コードと称する）。第１の認証コードは、第２の実施の形態例と同様に、新たに生成した乱数の共有時におけるスクランブル化において使用される。そして、第２の認証コードは、データをスクランブル化するための第２のスクランブルキー生成のために使用される。以下、図に従って説明する。
【００５８】
図９、図１０は、第３の実施の形態例におけるホストチップ１０とセキュリティチップ２０の処理の一例を表す図である。同９、図１０は、第１の実施の形態例における図３、図４と同様にして、ホストチップ１０からセキュリティチップ２０に非暗号化データが送信される場合のデータのスクランブル化について説明する図である。本実施の形態例では、ホストチップ１０及びセキュリティチップ２０は、第１の認証コードＡ１、第２の認証コードＡ２を予め共有する。
【００５９】
図９において、セキュリティチップ２０は、初期化コマンドを受信すると（ａ）、乱数Ｒ１を生成する（ｂ）。続いて、セキュリティチップ２０は、乱数Ｒ１と第１の認証コードＡ１とをＸＯＲ演算して第１のスクランブルキーＳ１１を生成する（ｃ）。そして、セキュリティチップ２０は、初期化処理の結果と共に、生成した第１のスクランブルキーＳ１１をホストチップ１０に送信する（ｄ）。ここまでの処理は、第１、２の実施の形態例と同様である。
【００６０】
続いて、本実施の形態例におけるセキュリティチップ２０は、さらに、生成した乱数Ｒ１と第２の認証コードＡ２とをＸＯＲ演算して第２のスクランブルキーＳ２１を生成する（ｄ）。一方、ホストチップ１０では、第１、２の実施の形態例と同様に、第１のスクランブルキーＳ１１を、予め保持している認証コードＡ１に従ってＸＯＲ演算し、乱数Ｒ１を取得する（ｆ）。そして、さらに、本実施の形態例におけるホストチップ１０は、取得した乱数Ｒ１と第２の認証コードＡ２とをＸＯＲ演算し、第２のスクランブルキーＳ２１を生成する（ｇ）。これにより、第１、第２の認証コードＡ１、Ａ２に加えて、乱数Ｒ１及び第２のスクランブルキーＳ２１が、ホストチップ１０とセキュリティチップ２０間で共有される。
【００６１】
そして、本実施の形態例におけるホストチップ１０は、暗号化対象のデータＤ１と第２のスクランブルキーＳ２１とをＸＯＲ演算してスクランブル化データＸ１を生成し（ｈ）、暗号化命令と共にセキュリティチップ２０に送信する（ｉ）。セキュリティチップ２０は、スクランブル化データＸ１とを受信すると、スクランブル化データＸ１を、生成しておいた第２のスクランブルキーＳ２１に従ってＸＯＲ演算して、暗号化対象のデータＤ１を取得する（ｊ）。そして、データＤ１は、暗号化ユニット２４によって暗号化処理され、暗号化データＥ１が生成される（ｋ）。続いて、セキュリティチップ２０は、暗号化データＥ１と第２のスクランブルキーＳ２１とをＸＯＲ演算し、スクランブル化暗号データＹ１を生成する（ｌ）。
【００６２】
図１０に移り、続いて、セキュリティチップ２０は、新たな乱数Ｒ２を生成し（ｍ）、当該乱数Ｒ２と第１の認証コードＡ１とをＸＯＲ演算して、新しい第１のスクランブルキーＳ１２を生成する（ｎ）。続いて、セキュリティチップ２０は、図９の処理ｌで生成したスクランブル化暗号データＹ１と共に、第１のスクランブルキーＳ１２をホストチップ１０に送信する（ｏ）。そして、セキュリティチップ２０は、乱数Ｒ２と第２の認証コードＡ２とをＸＯＲ演算して、第２のスクランブルキーＳ２２を生成しておく（ｐ）。
【００６３】
一方、ホストチップ１０は、スクランブル化暗号データＹ１と第１のスクランブルキー

Ｓ１２とを受信すると、スクランブル化暗号データＹ１を、共有済みの第２のスクランブルキーＳ２１に従ってＸＯＲ演算し、暗号化データＥ１を取得する（ｑ）。また、ホストチップ１０は、受信した第１のスクランブルキーＳ１２を第１の認証コードＡ１に従ってＸＯＲ演算し、乱数Ｒ２を取得する（ｒ）。これにより、暗号化対象のデータＤ１に対応する暗号化データＥ１がホストチップ１０で取得されると共に、乱数Ｒ２がホストチップ１０とセキュリティチップ２０間で共有される。

【００６４】

そして、暗号化する対象のデータＤ２がさらにある場合、同様にして、ホストチップ１０は、乱数Ｒ２と第２の認証コードＡ２とをＸＯＲ演算して第２のスクランブルキーＳ２２を生成する（ｓ）。そして、ホストチップ１０は、暗号化対象のデータＤ２とスクランブルキーＳ２２とをＸＯＲ演算してスクランブル化データＸ２を生成し（図示せず）、セキュリティチップ２０に送信する（図示せず）。セキュリティチップ２０は、受信したスクランブル化データＸ２を、処理ｋで生成した第２のスクランブルキーＳ２２に従ってＸＯＲ演算し、暗号化対象のデータＤ２を取得する。この後の処理は図９、図１０の処理ｋ～処理ｑと同様である。

【００６５】

なお、図９、図１０の例では、ホストチップ１０からセキュリティチップ２０に暗号化対象のデータが送信される場合を例示したが、ホストチップ１０からセキュリティチップ２０に復号化対象のデータが送信される場合についても同様である。この場合についても、セキュリティチップ２０で新たに生成された乱数は認証コードＡ１に従ってスクランブル化されてホストチップ１０に送信され、ホストチップ１０で認証コードＡ１に従って復元されることによって共有される。そして、セキュリティチップ２０で復号化されたデータは、乱数がさらに第２の認証コードＡ２に従ってスクランブル化されて生成された第２のスクランブルキーに従ってスクランブル化され、ホストチップ１０送信される。ホストチップ１０では、スクランブル化されたデータが第２のスクランブルキーに従ってスクランブル演算されることにより、復号化データが復元される。

【００６６】

以上のように、本実施の形態例におけるセキュリティチップ２０は、データを暗号化する場合、乱数生成ユニット２３が生成した乱数と第１の認証コードとをスクランブル演算して第１のスクランブルキーを生成してホストチップ１０に送信すると共に、乱数と第２の認証コードとをスクランブル演算して第２のスクランブルキーを生成する。また、セキュリティチップ２０は、ホストチップ１０から、第１のスクランブルキーから取得された乱数と第２の認証コードとがスクランブル演算されて生成された第２のスクランブルキーに従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信する。そして、セキュリティチップ２０は、スクランブルデータと第２のスクランブルキーとをスクランブル演算して暗号化対象データを生成し、暗号化対象データを暗号化ユニット２４によって暗号化して暗号化データを生成し、ホストチップ１０に送信する。

【００６７】

一方、本実施の形態例におけるセキュリティチップ２０は、データを復号化する場合、乱数生成ユニット２３が生成した乱数と第１の認証コードとをスクランブル演算して第１のスクランブルキーを生成してホストチップ１０に送信すると共に、乱数と第２の認証コードとをスクランブル演算して第２のスクランブルキーを生成する。また、セキュリティチップ２０は、暗号化データを暗号化ユニットによって復号化して復号化データを生成し、復号化データと第２のスクランブルキーとをスクランブル演算してスクランブルデータを生成しホストチップ１０に送信する。そして、セキュリティチップ２０は、ホストチップ１０のプロセッサ１１に、第１のスクランブルキーから取得された乱数と第２の認証コードとがスクランブル演算されて生成された第２のスクランブルキーに従って、スクランブルデータをスクランブル演算させ、スクランブルデータから復号化データを取得させる。

【００６８】

また、ホストチップ１０では、乱数は、第１のスクランブルキーが、第１の認証コードに従ってスクランブル演算されることによって取得される。

【００６９】

これにより、本実施の形態例におけるセキュリティチップ２０は、ホストチップ１０に暗号化ユニット２４を要することなく、ホストチップ１０との間のデータをスクランブル化して、データの漏洩を防ぐことができる。また、セキュリティチップ２０は、データのスクランブル化に高度な暗号処理を用いないため、システムの負荷を抑え、スクランブル化による通信速度の低下を防ぐことができる。

【００７０】

また、本実施の形態例において、データのスクランブル化は、ホストチップ１０及びセキュリティチップ２０間で共有された乱数がさらに第２の認証コードに従ってスクランブル化され生成された第２のスクランブルキーに従って行われる。そのため、第３者は、乱数に加えて第２の認証コードを知らなければ、スクランブル化データを復元することができない。つまり、仮に、乱数が復元されてしまった場合でも、第２の認証コードを知らない第３者は、スクランブル化データを復元することができない。これにより、第３者によるスクランブルデータの復元がより困難になる。このように、本実施の形態例におけるセキュリティチップ２０は、ホストチップ１０とセキュリティチップ２０間のデータ漏洩をより効果的に防止することができる。

【００７１】

また、本実施の形態例におけるセキュリティチップ２０によると、乱数を生成する乱数生成ユニット２３はセキュリティチップ２０のみに備えられていればよく、ホストチップ１０は乱数生成ユニット２３を有している必要がない。従って、ホストチップ１０を有する既存システムの回路基板に、本実施の形態例におけるセキュリティチップ２０が搭載されることによって、当該ホストチップ１０とセキュリティチップ２０間のデータのスクランブル化が可能になり、データの漏洩が防止される。

【００７２】

さらに、データのスクランブル化に使用される第２のスクランブルキーの元になる乱数は、一定数の送受信セッション毎に変更される。そのため、本実施の形態例におけるセキュリティチップ２０は、スクランブルデータの継続した解析を困難にし、データの漏洩範囲を最小限にすることができる。

【００７３】

このようにして、本実施の形態例におけるセキュリティチップ２０は、２つの認証コードを有し、第１の認証コードを使用して共有した乱数と第２の認証コードとに基づくスクランブルキーに従って、データをスクランブル化する。これにより、本実施の形態例におけるセキュリティチップ２０は、ホストチップ１０にハードウェアの変更を要することなく、システム全体の負荷を抑えながら、セキュリティ強度の高いデータのスクランブル化を実現することができる。これにより、セキュリティチップ２０は、よりデータの漏洩を効果的に防ぐことができる。

【００７４】

なお、本実施の形態例において、第１のスクランブルキーは、共有対象の乱数が、第１の認証コードに従ってスクランブル演算されることによって生成される。しかしながら、第１のスクランブルキーは、共有対象の乱数が、第１の認証コードではなく前回の送受信セッションの乱数に従って、スクランブル演算されることによって生成されてもよい。

【００７５】

この場合、２回目以降の各送受信セッションでは、セキュリティチップ２０は、今回の送受信セッションの乱数と、前回の送受信セッションの乱数とのスクランブル演算によって第１のスクランブルキーを生成し、ホストチップ１０は、第１スクランブルキーから、前回の送受信セッションの乱数によるスクランブル演算によって、今回の送受信セッションの乱数を取得する。これにより、仮に第１の認証コードが第３者に知られてしまった場合でも、前回の送受信セッションの乱数を知らない第３者は、第１のスクランブルキーか

ら今回の送受信セッションの乱数を復元することができない。
【００７６】
以上、ホストチップ１０とセキュリティチップ２０間のデータのスクランブル化について述べた。ただし、本実施の形態例におけるデータのスクランブル化は、ホストチップ１０とセキュリティチップ２０間におけるデータのスクランブル化に限定されるものではなく、他の２者間（例えば、サーバとクライアント等）における通信のデータのスクランブル化にも適用可能である。
【００７７】
以上の実施の形態をまとめると、次の付記のとおりである。
【００７８】
（付記１）
スクランブル演算機能を有するプロセッサと、第１の認証コードが記憶された記憶ユニットとを有するホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
前記ホスト装置から、前記第１のスクランブルキーから取得した前記乱数に従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信し、前記スクランブルデータと前記乱数とを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティ装置。
【００７９】
（付記２）
スクランブル演算機能を有するプロセッサと、第１の認証コードと第２の認証コードとが記憶された記憶ユニットとを有するホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードと前記第２の認証コードとが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
前記ホスト装置から、前記第１のスクランブルキーから取得された前記乱数と前記第２の認証コードとが前記スクランブル演算されて生成された第２のスクランブルキーに従って、暗号化対象データがスクランブル演算されて生成されたスクランブルデータを受信し、前記スクランブルデータと前記第２のスクランブルキーとを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティ装置。
【００８０】
（付記３）
スクランブル演算機能を有するプロセッサと、第１の認証コードが記憶された記憶ユニットとを有するホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードが記憶された記憶ユニットと、

乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記乱数とを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置の前記プロセッサに、前記第１のスクランブルキーから取得された前記乱数に従って前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得させるセキュリティ装置。

【００８１】

（付記４）
スクランブル演算機能を有するプロセッサと、第１の認証コードと第２の認証コードとが記憶された記憶ユニットとを有するホスト装置に接続されるセキュリティ装置であって、
前記第１の認証コードと前記第２の認証コードとが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記第２のスクランブルキーとを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置の前記プロセッサに、前記第１のスクランブルキーから取得された前記乱数と前記第２の認証コードとが前記スクランブル演算されて生成された前記第２のスクランブルキーに従って、前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得させるセキュリティ装置。

【００８２】

（付記５）
付記１乃至４のいずれかにおいて、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の認証コードによる前記スクランブル演算によって前記乱数が取得されるセキュリティ装置。

【００８３】

（付記６）
付記１乃至４のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前回の送受信セッションの乱数との前記スクランブル演算によって生成し、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記前回の送受信セッションの乱数による前記スクランブル演算によって、前記今回の送受信セッションの乱数が取得されるセキュリティ装置。

【００８４】

（付記７）
付記１乃至４のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前記第１の認証コードとの前記スクランブル演算によって生成し、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の認証コードによる前記スクランブル演算によって、前記今回の送受信セッションの乱数が取得されるセキュリティ装置。

【００８５】
（付記８）
付記１乃至４のいずれかにおいて、
前記スクランブル演算は、排他的論理和演算を含むセキュリティ装置。

【００８６】
（付記９）
ホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードが記憶された記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数に従って暗号化対象データをスクランブル演算しスクランブルデータを生成して前記セキュリティ装置に送信し、
前記コントローラは、受信した前記スクランブルデータと前記乱数とを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティシステム。

【００８７】
（付記１０）
ホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードと第２の認証コードとが記憶された記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードと第２の認証コードとが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱

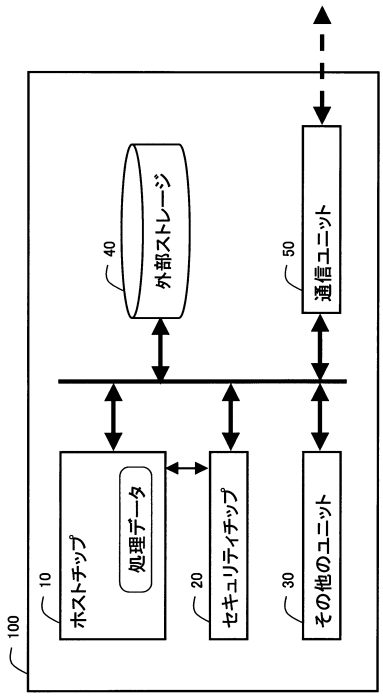
数と前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数をさらに前記第２の認証コードと前記スクランブル演算して生成した第２のスクランブルキーに従って、暗号化対象データをスクランブル演算しスクランブルデータを生成して前記セキュリティ装置に送信し、
前記コントローラは、受信した前記スクランブルデータと前記第２のスクランブルキーとを前記スクランブル演算して前記暗号化対象データを生成し、当該暗号化対象データを前記暗号化ユニットによって暗号化して暗号化データを生成し、前記ホスト装置に送信するセキュリティシステム。

【００８８】
（付記１１）
ホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードが記憶された記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記乱数とを前記スクランブル演算してスクランブルデータを生成し前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得された前記乱数に従って前記スクランブルデータをスクランブル演算して前記スクランブルデータから前記復号化データを取得するセキュリティシステム。

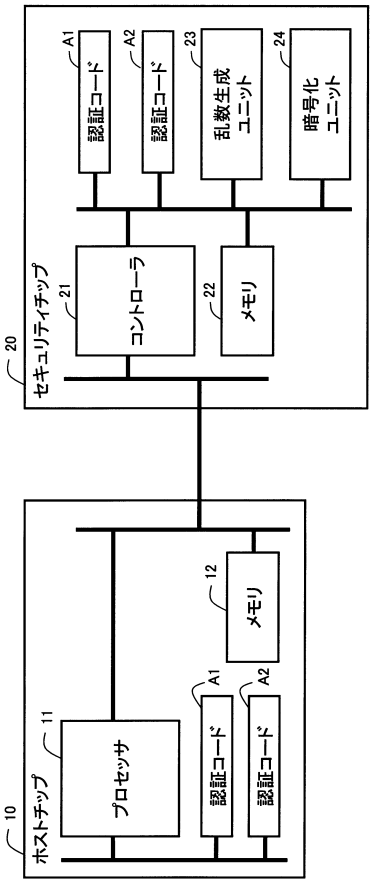
【００８９】
（付記１２）
ホスト装置と、前記ホスト装置に接続されるセキュリティ装置とを有するセキュリティシステムであって、
前記ホスト装置は、
スクランブル演算機能を有するプロセッサと、
第１の認証コードと第２の認証コードとが記憶された記憶ユニットとを有し、
前記セキュリティ装置は、
前記第１の認証コードと第２の認証コードとが記憶された記憶ユニットと、
乱数を生成する乱数生成ユニットと、
データを暗号化または復号化する暗号化ユニットと、
前記スクランブル演算機能を有するコントローラとを有し、
前記コントローラは、
前記乱数生成ユニットが生成した前記乱数と前記第１の認証コードとを前記スクランブル演算して第１のスクランブルキーを生成して前記ホスト装置に送信すると共に、前記乱数と前記第２の認証コードとを前記スクランブル演算して第２のスクランブルキーを生成し、
暗号化データを前記暗号化ユニットによって復号化して復号化データを生成し、当該復号化データと前記第２のスクランブルキーとを前記スクランブル演算してスクランブルデ

ータを生成し前記ホスト装置に送信し、
前記ホスト装置は、前記プロセッサによって、前記第１のスクランブルキーから取得した前記乱数と前記第２の認証コードとを前記スクランブル演算して生成した前記第２のスクランブルキー従って、前記スクランブルデータをスクランブル演算させ、前記スクランブルデータから前記復号化データを取得するセキュリティシステム。
【００９０】
（付記１３）
付記９乃至１２のいずれかにおいて、
前記ホスト装置では、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の認証コードによるスクランブル演算によって前記乱数が取得されるセキュリティシステム。
【００９１】
（付記１４）
付記９乃至１２のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前回の送受信セッションの乱数との前記スクランブル演算によって生成し、
前記ホスト装置は、前記プロセッサにより、前記第１のスクランブルキーから、前記前回の送受信セッションの乱数による前記スクランブル演算によって、前記今回の送受信セッションの乱数を取得するセキュリティシステム。
【００９２】
（付記１５）
付記９乃至１２のいずれかにおいて、
前記乱数は前記スクランブルデータの送受信セッション毎に変更され、
２回目以降の各送受信セッションにおいて、
前記コントローラは、前記第１のスクランブルキーを、今回の送受信セッションの前記乱数と、前記第１の認証コードとの前記スクランブル演算によって生成し、
前記ホスト装置は、前記プロセッサにより、前記第１のスクランブルキーから、前記第１の認証コードによる前記スクランブル演算によって、前記今回の送受信セッションの乱数を取得するセキュリティシステム。
【符号の説明】
【００９３】
１０：ホストチップ、１１：プロセッサ、Ａ１：第１の認証コード、Ａ２：第２の認証コード、１２：メモリ、２０：セキュリティチップ、２１：コントローラ、２２：メモリ、２３：乱数生成ユニット、２４：暗号化ユニット

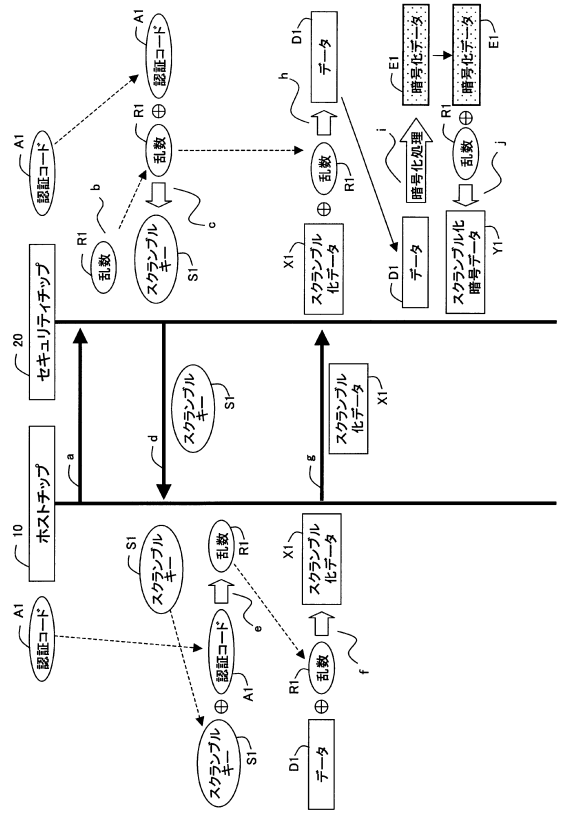
【図１】


100
ホストチップ
10
処理データ
セキュリティチップ
20
その他のユニット
30
外部ストレージ
40
通信ユニット
50

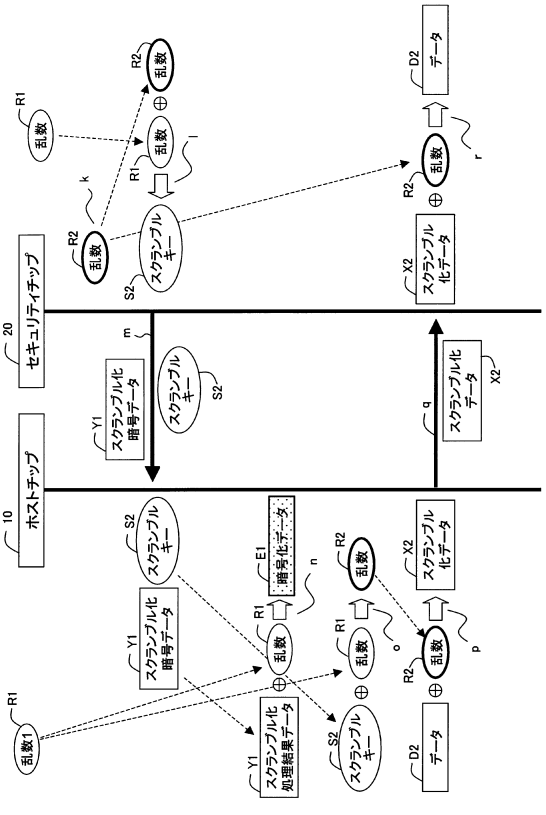
【図２】


ホストチップ
10
プロセッサ
11
メモリ
12
認証コード
A1
認証コード
A2
セキュリティチップ
20
コントローラ
21
メモリ
22
認証コード
A1
認証コード
A2
乱数生成ユニット
23
暗号化ユニット
24

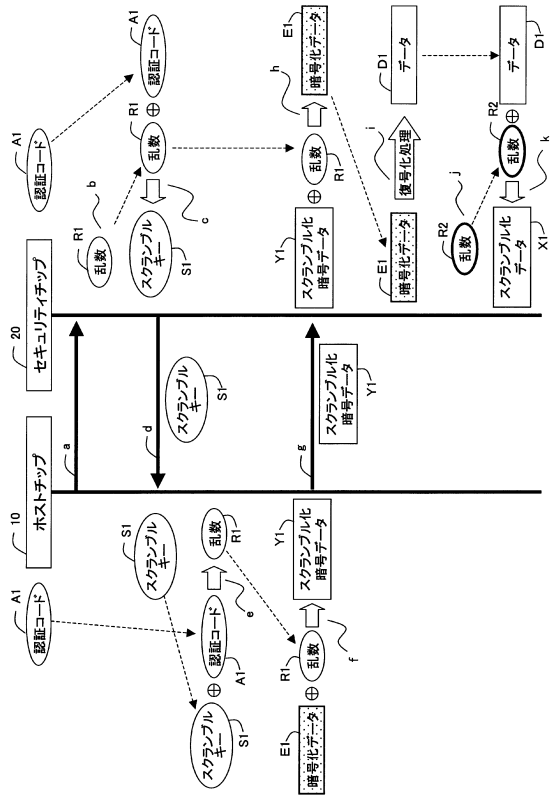
【図３】


ホストチップ
10
セキュリティチップ
20
認証コード
A1
乱数
R1
スクランブルキー
S1
スクランブル化データ
X1
データ
D1
暗号化処理
スクランブル化暗号データ
Y1
暗号化データ
E1

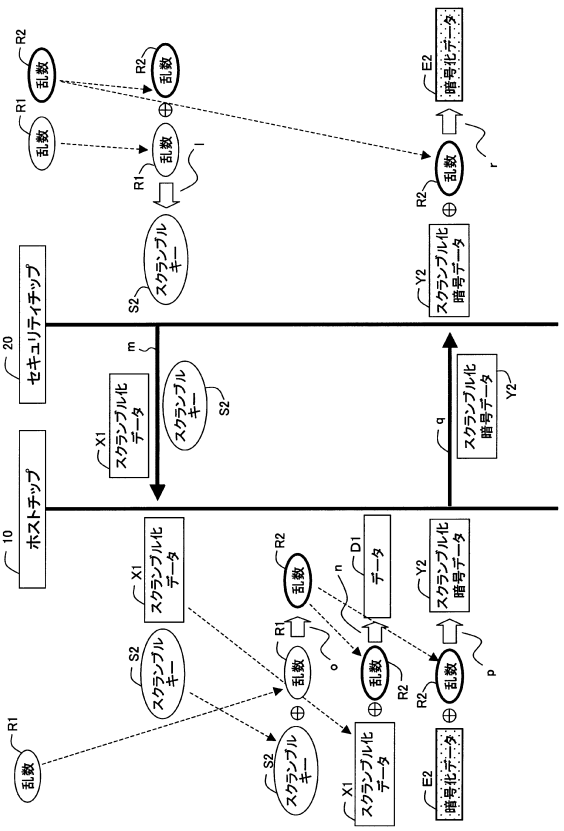
【図４】


ホストチップ
10
セキュリティチップ
20
乱数
R1
R2
スクランブルキー
S1
S2
スクランブル化データ
X1
X2
スクランブル化暗号データ
Y1
暗号化データ
E1
スクランブル化処理結果データ
データ
D2

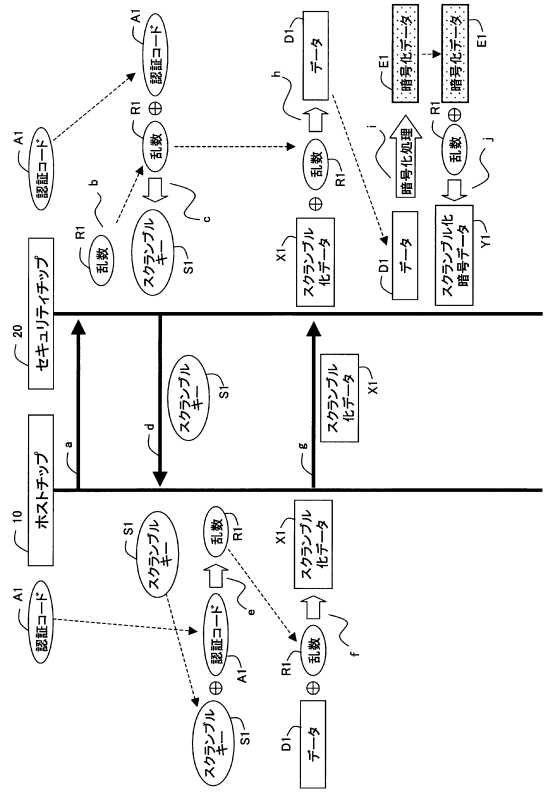
【図５】


ホストチップ
セキュリティチップ
認証コード
乱数
スクランブルキー
スクランブル化暗号データ
暗号化データ
復号化処理
データ
スクランブル化データ

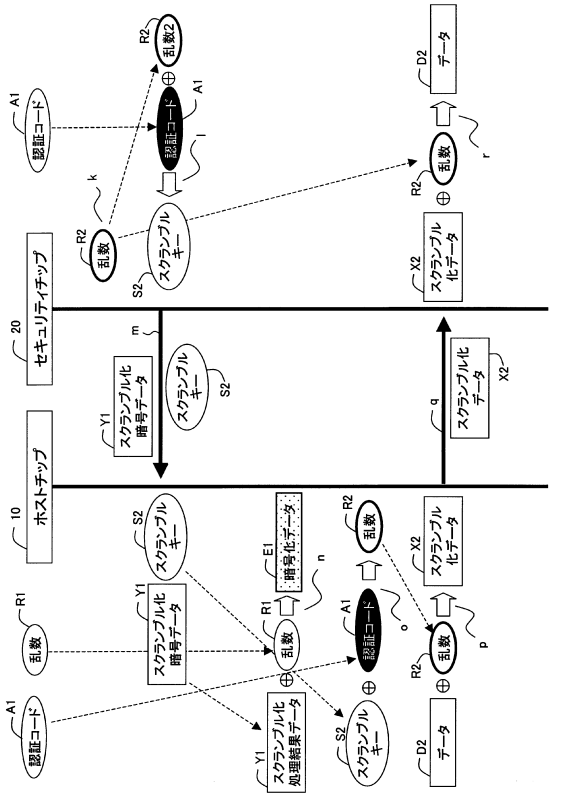
【図６】


ホストチップ
セキュリティチップ
乱数
スクランブルキー
スクランブル化データ
スクランブル化暗号データ
データ

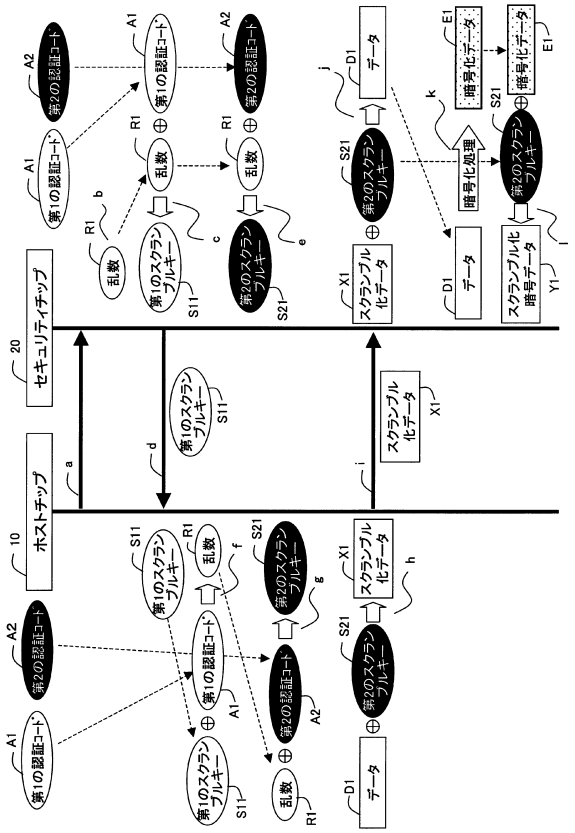
【図７】


ホストチップ
セキュリティチップ
認証コード
乱数
スクランブルキー
スクランブル化データ
暗号化処理
暗号化データ
データ

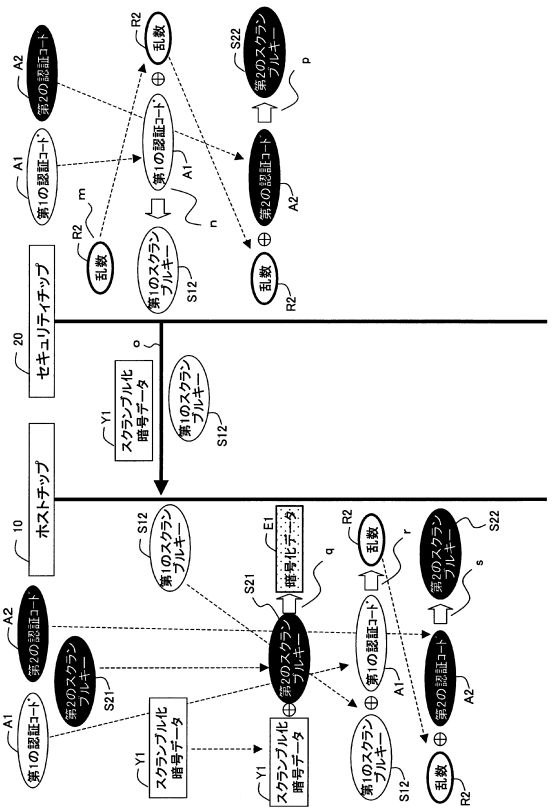
【図８】


ホストチップ
セキュリティチップ
認証コード
乱数
乱数2
スクランブルキー
スクランブル化データ
スクランブル化暗号データ
暗号化データ
認証結果データ
データ

【図９】


ホストチップ
セキュリティチップ
第1の認証コード
第2の認証コード
乱数
第1のスクランブルキー
第2のスクランブルキー
スクランブル化データ
暗号化処理
暗号化データ
データ

【図１０】


ホストチップ
セキュリティチップ
第1の認証コード
第2の認証コード
乱数
第1のスクランブルキー
第2のスクランブルキー
スクランブル化暗号データ
暗号化データ

フロントページの続き

(56)参考文献 特開２０１０－０１６５２６（ＪＰ，Ａ）
特開２００６－２７７４１１（ＪＰ，Ａ）
特開平０８－３３５０４０（ＪＰ，Ａ）
特開２００５－２９５１６４（ＪＰ，Ａ）
特開２００２－０２４９１４（ＪＰ，Ａ）
特開２００１－００５７３１（ＪＰ，Ａ）
特開平０９－２７０７８４（ＪＰ，Ａ）
岡本　栄司，“明るい情報化社会の実現をめざす暗号技術▲１▼　情報に対する脅威と対策”，ｂｉｔ，日本，共立出版株式会社，１９９１年　7月　1日，Ｖｏｌ．２３、Ｎｏ．８，ｐ．６０－６６

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／１４
Ｈ０４Ｌ　９／０８