(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7583000号
(P7583000)

(45)発行日　令和6年11月13日(2024.11.13)

(24)登録日　令和6年11月5日(2024.11.5)

(51)国際特許分類			FI		
G06F	*21/60*	*(2013.01)*	G06F	21/60	320
G06F	*11/10*	*(2006.01)*	G06F	11/10	604
G11B	*20/10*	*(2006.01)*	G11B	20/10	301Z
G11B	*20/18*	*(2006.01)*	G11B	20/10	H
			G11B	20/18	512C

請求項の数　21　（全35頁）　最終頁に続く

(21)出願番号　特願2022-141346(P2022-141346)
(22)出願日　令和4年9月6日(2022.9.6)
(65)公開番号　特開2024-36837(P2024-36837A)
(43)公開日　令和6年3月18日(2024.3.18)
審査請求日　令和5年5月22日(2023.5.22)

(73)特許権者　524132520
日立ヴァンタラ株式会社
神奈川県横浜市戸塚区吉田町２９２番地
(74)代理人　110001678
藤央弁理士法人
(72)発明者　水島 永雅
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
(72)発明者　吉井 義裕
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
(72)発明者　岡田 尚也
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
審査官　辻 勇貴

最終頁に続く

(54)【発明の名称】　計算機及びデータ処理方法

(57)【特許請求の範囲】

【請求項１】

複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機であって、

プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを備え、

前記プロセッサは、

複数のレジスタを有し、

前記ユーザデータの書き込みリクエストを受信し、

前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納し、

前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行し、

２以上の第１データ保全性フィールドを第１レジスタに格納し、

前記第１レジスタに格納された前記２以上の前記第１データ保全性フィールドの各々に、前記暗号化データブロックの格納位置を示す第１アドレスを設定する設定処理を並列に実行し、

前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、前記第１アドレスが設定された第１データ保全性フィールドを付加し、

前記暗号化ユーザデータを前記記憶媒体に格納する、

ことを特徴とする計算機。

【請求項２】
請求項１に記載の計算機であって、
前記第１データ保全性フィールドには、前記データブロックの書き込みの世代を表す第１世代番号を設定することができ、
前記プロセッサは、前記第１レジスタに格納された前記２以上の前記第１データ保全性フィールドの各々に、前記暗号化データブロックの前記第１世代番号を設定する設定処理を並列に実行することを特徴とする計算機。
【請求項３】
請求項１に記載の計算機であって、
前記第１データ保全性フィールドは第１誤り符号を含むことを特徴とする計算機。
【請求項４】
請求項２に記載の計算機であって、
前記プロセッサは、
前記暗号化ユーザデータの読み出しリクエストを受信し、
前記記憶媒体から、前記読み出しリクエストによって指定された前記暗号化ユーザデータを読み出し、前記メモリに格納し、
前記複数のレジスタを用いて、前記暗号化ユーザデータを復号する復号処理を実行し、
第２レジスタに２以上の第２データ保全性フィールドを格納し、
前記第２レジスタに格納された前記２以上の前記第２データ保全性フィールドの各々に、前記暗号化データブロックの格納位置を示す第２アドレスを設定する設定処理を並列に実行し、
復号された前記ユーザデータに含まれる前記複数のデータブロックの各々に、前記第２アドレスが設定された第２データ保全性フィールドを付加し、前記メモリに格納する、
ことを特徴とする計算機。
【請求項５】
請求項４に記載の計算機であって、
前記第２データ保全性フィールドには、前記データブロックの書き込みの世代を表す第２世代番号を設定することができ、
前記プロセッサは、前記第２レジスタに格納された前記２以上の前記第２データ保全性フィールドの各々に、前記データブロックの前記第２世代番号を設定する設定処理を並列に実行することを特徴とする計算機。
【請求項６】
請求項５に記載の計算機であって、
前記プロセッサは、
ＳＩＭＤ命令を用いて、前記第１レジスタに格納された前記２以上の前記第１データ保全性フィールドの各々に前記第１アドレスを設定する設定処理を並列で実行し、
前記ＳＩＭＤ命令を用いて、前記第１レジスタに格納された前記２以上の前記第１データ保全性フィールドの各々に前記第１世代番号を設定する設定処理を並列で実行し、
前記ＳＩＭＤ命令を用いて、前記第２レジスタに格納された前記２以上の前記第２データ保全性フィールドの各々に前記第２アドレスを設定する設定処理を並列で実行し、
前記ＳＩＭＤ命令を用いて、前記第２レジスタに格納された前記２以上の前記第２データ保全性フィールドの各々に前記第２世代番号を設定する設定処理を並列で実行することを特徴とする計算機。
【請求項７】
請求項４に記載の計算機であって、
前記第２データ保全性フィールドは第１誤り符号を含むことを特徴とする計算機。
【請求項８】
複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機であって、
プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒

体、及び前記プロセッサに接続される接続インタフェースを備え、
前記プロセッサは、
複数のレジスタを有し、
前記ユーザデータの書き込みリクエストを受信し、
前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納し、
前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行し、
前記複数のレジスタを用いて、前記ユーザデータの破損を検出するための第１検査処理を実行し、
前記暗号化ユーザデータを前記記憶媒体に格納し、
前記ユーザデータに含まれる前記複数のデータブロックの各々には、第１誤り符号を含む第１データ保全性フィールドが付加され、
前記暗号化処理では、
所定の数の前記データブロックから、前記データブロックの一部である部分データを読み出し、第１レジスタに格納する第１処理と、
前記第１レジスタに格納される前記部分データを暗号化した暗号化部分データを第２レジスタに格納する第２処理と、
前記第１レジスタに格納される前記部分データを用いて、前記第１誤り符号を生成するための第１演算を実行し、前記第１演算の結果を第３レジスタに格納する第３処理と、
が繰り返し実行され、
前記第１検査処理は、
第４レジスタに、所定の数の前記データブロックの前記第１データ保全性フィールドを読み出す処理と、
前記第３レジスタに格納される前記第１誤り符号と、前記第４レジスタに格納される前記第１データ保全性フィールドに含まれる前記第１誤り符号とを比較することによって、前記メモリから読み出された前記ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とする計算機。
【請求項９】
請求項８に記載の計算機であって、
前記第１データ保全性フィールドは、前記データブロックの格納位置を示す第１アドレスを含み、
前記第１検査処理は、
第５レジスタに、所定の数の前記データブロックの前記第１アドレスを読み出す処理と、
前記第５レジスタに格納される前記第１アドレスと、前記第４レジスタに格納される前記第１データ保全性フィールドに含まれる前記第１アドレスとを比較することによって、前記メモリから読み出された前記ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とする計算機。
【請求項１０】
請求項９に記載の計算機であって、
前記プロセッサは、
前記暗号化ユーザデータの読み出しリクエストを受信し、
前記記憶媒体から、前記読み出しリクエストによって指定された前記暗号化ユーザデータを読み出し、前記メモリに格納し、
前記複数のレジスタを用いて、前記暗号化ユーザデータを復号する復号処理を実行し、
前記複数のレジスタを用いて、前記暗号化ユーザデータの破損を検出するための第２検査処理を実行し、
前記ユーザデータを前記メモリに格納し、
前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々には、第２誤り符号を含む第２データ保全性フィールドが付加され、
前記復号処理では、

所定の数の前記暗号化データブロックから、前記暗号化データブロックの一部である部分暗号化データを読み出し、第６レジスタに格納する第８処理と、
前記第６レジスタに格納される前記部分暗号化データを復号した前記部分データを第７レジスタに格納する第９処理と、
前記第６レジスタに格納される前記暗号化部分データを用いて、前記第２誤り符号を生成するための第２演算を実行し、前記第２演算の結果を第８レジスタに格納する第１０処理と、
が繰り返し実行され、
前記第２検査処理は、
第９レジスタに、所定の数の前記暗号化データブロックの前記第２データ保全性フィールドを読み出す処理と、
前記第８レジスタに格納される前記第２誤り符号と、前記第９レジスタに格納される前記第１データ保全性フィールドに含まれる前記第２誤り符号とを比較することによって、前記記憶媒体から読み出された前記暗号化ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とする計算機。
【請求項１１】
請求項１０に記載の計算機であって、
前記第２データ保全性フィールドは、前記暗号化データブロックの格納位置を示す第２アドレスを含み、
前記第２検査処理は、
第１０レジスタに、所定の数の前記暗号化データブロックに対応する前記第２アドレスを読み出す処理と、
前記第９レジスタに格納される前記第２データ保全性フィールドに含まれる前記第２アドレスと、前記第１０レジスタに格納される前記第２アドレスとを比較することによって、前記記憶媒体から読み出された前記暗号化ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とする計算機。
【請求項１２】
請求項１１に記載の計算機であって、
前記第２データ保全性フィールドは、前記データブロックの書き込みの世代を表す第２世代番号を含み、
前記第２検査処理は、
第１１レジスタに、所定の数の前記暗号化データブロックに対応する前記第２世代番号を読み出す処理と、
前記第９レジスタに格納される前記第１データ保全性フィールドに含まれる前記第２世代番号と、前記第１１レジスタに格納される前記第２世代番号とを比較することによって、前記記憶媒体から読み出された前記暗号化ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とする計算機。
【請求項１３】
請求項１２に記載の計算機であって、
前記プロセッサは、ＳＩＭＤ命令を用いて、前記複数のデータブロック及び前記複数の暗号化データブロックに対する処理を並列で実行することを特徴とする計算機。
【請求項１４】
複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機が実行するデータ処理方法であって、
前記計算機は、プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを有し、
前記プロセッサは、複数のレジスタを有し、

前記データ処理方法は、
前記プロセッサが、前記ユーザデータの書き込みリクエストを受信するステップと、
前記プロセッサが、前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納するステップと、
前記プロセッサが、前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行するステップと、
前記プロセッサが、前記複数のレジスタを用いて、前記ユーザデータの破損を検出するための検査処理を実行するステップと、
前記プロセッサが、前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、第１誤り符号を含む第１データ保全性フィールドを付加するステップと、
前記プロセッサが、前記暗号化ユーザデータを前記記憶媒体に格納するステップと、
を含み、
前記暗号化処理では、
所定の数の前記データブロックから、前記データブロックの一部である部分データを読み出し、第１レジスタに格納する第１処理と、
前記第１レジスタに格納される前記部分データを暗号化した暗号化部分データを第２レジスタに格納する第２処理と、
前記第２レジスタに格納される前記暗号化部分データを用いて、前記第１誤り符号を算出するための第１演算を実行し、前記第１演算の結果を第３レジスタに格納する第３処理と、
が繰り返し実行され、
前記検査処理は、
第４レジスタに、所定の数の前記データブロックの前記第１データ保全性フィールドを読み出す処理と、
前記第３レジスタに格納される前記第１誤り符号と、前記第４レジスタに格納される前記第１データ保全性フィールドに含まれる前記第１誤り符号とを比較することによって、前記メモリから読み出された前記ユーザデータが破損しているか否かを判定する処理と、
を含むことを特徴とするデータ処理方法。
【請求項１５】
複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機であって、
プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを備え、
前記プロセッサは、
複数のレジスタを有し、
前記ユーザデータの書き込みリクエストを受信し、
前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納し、
前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行し、
前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、第１誤り符号を含む第１データ保全性フィールドを付加し、
前記暗号化ユーザデータを前記記憶媒体に格納し、
前記暗号化処理では、
所定の数の前記データブロックから、前記データブロックの一部である部分データを読み出し、第１レジスタに格納する第１処理と、
前記第１レジスタに格納される前記部分データを暗号化した暗号化部分データを第２レジスタに格納する第２処理と、
前記第２レジスタに格納される前記暗号化部分データを用いて、前記第１誤り符号を算出するための第１演算を実行し、前記第１演算の結果を第３レジスタに格納する第３処理と、

が繰り返し実行され、
前記第１データ保全性フィールドには、前記暗号化データブロックの格納位置を示す第１アドレスを設定することができ、
前記プロセッサは、
第４レジスタに、所定の数の前記暗号化データブロックの前記第１誤り符号を読み出し、
前記第４レジスタに格納される前記第１誤り符号に、当該第１誤り符号が付加される前記暗号化データブロックの前記第１アドレスを追加することによって、前記第１データ保全性フィールドを生成し、
前記第１データ保全性フィールドには、前記データブロックの書き込みの世代を表す第１世代番号を設定することができ、
前記プロセッサは、前記第４レジスタに格納される前記第１誤り符号に、当該第１誤り符号が付加される前記暗号化データブロックの前記第１世代番号を追加することによって、前記第１データ保全性フィールドを生成することを特徴とする計算機。
【請求項１６】
請求項１５に記載の計算機であって、
前記プロセッサは、
前記暗号化ユーザデータの読み出しリクエストを受信し、
前記記憶媒体から、前記読み出しリクエストによって指定された前記暗号化ユーザデータを読み出し、前記メモリに格納し、
前記複数のレジスタを用いて、前記暗号化ユーザデータを復号する復号処理を実行し、
復号された前記ユーザデータに含まれる前記複数のデータブロックの各々に、第２誤り符号を含む第２データ保全性フィールドを付加し、前記メモリに格納し、
前記復号処理では、
所定の数の前記暗号化データブロックから、前記暗号化データブロックの一部である部分暗号化データを読み出し、第５レジスタに格納する第４処理と、
前記第５レジスタに格納される前記部分暗号化データを復号した前記部分データを第６レジスタに格納する第５処理と、
前記第６レジスタに格納される前記部分データを用いて、前記第２誤り符号を算出するための第２演算を実行し、前記第２演算の結果を第７レジスタに格納する第８処理と、
が繰り返し実行されることを特徴とする計算機。
【請求項１７】
請求項１６に記載の計算機であって、
前記第２データ保全性フィールドには、前記暗号化データブロックの格納位置を示す第２アドレスを設定することができ、
前記プロセッサは、
第８レジスタに、所定の数の前記データブロックの前記第２誤り符号を読み出し、
前記第８レジスタに格納される前記第２誤り符号に、当該第２誤り符号が付加される前記データブロックの前記第２アドレスを追加することによって、前記第２データ保全性フィールドを生成することを特徴とする計算機。
【請求項１８】
請求項１７に記載の計算機であって、
前記第２データ保全性フィールドには、前記データブロックの書き込みの世代を表す第２世代番号を設定することができ、
前記プロセッサは、前記第８レジスタに格納される前記第２誤り符号に、当該第２誤り符号が付加される前記データブロックの前記第２世代番号を追加することによって、前記第２データ保全性フィールドを生成することを特徴とする計算機。
【請求項１９】
請求項１８に記載の計算機であって、
前記プロセッサは、ＳＩＭＤ命令を用いて、前記複数のデータブロック及び前記複数の暗号化データブロックに対する処理を並列で実行することを特徴とする計算機。

【請求項２０】
複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機が実行するデータ処理方法であって、
前記計算機は、プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを備え、
前記プロセッサは、複数のレジスタを有し、
前記データ処理方法は、
前記プロセッサが、前記ユーザデータの書き込みリクエストを受信するステップと、
前記プロセッサが、前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納するステップと、
前記プロセッサが、前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行するステップと、
前記プロセッサが、２以上の第１データ保全性フィールドを第１レジスタに格納するステップと、
前記プロセッサが、前記第１レジスタに格納された前記２以上の前記第１データ保全性フィールドの各々に、前記暗号化データブロックの格納位置を示す第１アドレスを設定する設定処理を並列に実行するステップと、
前記プロセッサが、前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、前記第１アドレスが設定された第１データ保全性フィールドを付加するステップと、
前記プロセッサが、前記暗号化ユーザデータを前記記憶媒体に格納するステップと、
を含むことを特徴とするデータ処理方法。
【請求項２１】
複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機が実行するデータ処理方法であって、
プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを有し、
前記プロセッサは、複数のレジスタを有し、
前記データ処理方法は、
前記プロセッサが、前記ユーザデータの書き込みリクエストを受信するステップと、
前記プロセッサが、前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納するステップと、
前記プロセッサが、前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行するステップと、
前記プロセッサが、前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、第１誤り符号を含む第１データ保全性フィールドを付加するステップと、
前記プロセッサが、前記暗号化ユーザデータを前記記憶媒体に格納するステップと、
を含み、
前記暗号化処理では、
所定の数の前記データブロックから、前記データブロックの一部である部分データを読み出し、第１レジスタに格納する第１処理と、
前記第１レジスタに格納される前記部分データを暗号化した暗号化部分データを第２レジスタに格納する第２処理と、
前記第２レジスタに格納される前記暗号化部分データを用いて、前記第１誤り符号を算出するための第１演算を実行し、前記第１演算の結果を第３レジスタに格納する第３処理と、
が繰り返し実行され、
前記第１データ保全性フィールドには、前記暗号化データブロックの格納位置を示す第１アドレスを設定することができ、
前記プロセッサは、

第４レジスタに、所定の数の前記暗号化データブロックの前記第１誤り符号を読み出し、前記第４レジスタに格納される前記第１誤り符号に、当該第１誤り符号が付加される前記暗号化データブロックの前記第１アドレスを追加することによって、前記第１データ保全性フィールドを生成し、

前記第１データ保全性フィールドには、前記データブロックの書き込みの世代を表す第１世代番号を設定することができ、

前記プロセッサは、前記第４レジスタに格納される前記第１誤り符号に、当該第１誤り符号が付加される前記暗号化データブロックの前記第１世代番号を追加することによって、前記第１データ保全性フィールドを生成することを特徴とするデータ処理方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ユーザデータの暗号化及び復号の技術に関する。

【背景技術】

【０００２】

内蔵されている記憶媒体からユーザデータが不正に読み出されないように、ユーザデータを暗号化して保存する機能を有するストレージシステムがある。前述の機能を有するストレージシステムは、ライトコマンドに応じて、外部のホストコンピュータから入力されたユーザデータを暗号化し、暗号化されたユーザデータを記憶媒体に保存する。また、前述の機能を有するストレージシステムは、リードコマンドに応じて、記憶媒体に保存された暗号化されたユーザデータを復号し、ユーザデータを出力する。

【０００３】

ストレージシステムがデータの暗号化及び復号を支援するハードウェアを有していない場合、ストレージシステムが備えるＣＰＵが、複数のＣＰＵ命令で構成されたプログラムコードにしたがって、ライト処理に伴うユーザデータの暗号化及びリード処理に伴う暗号化ユーザデータの復号を実行する。ストレージシステムのＣＰＵの暗号化及び復号の処理性能が高いほうが、ユーザデータの読み書きのレスポンスが高くなる。

【０００４】

暗号化及び復号の処理性能を向上するため、暗号専用命令をサポートしているＣＰＵがある。ＡＥＳ－ＮＩと呼ばれる命令セットは暗号専用命令の一例である。

【０００５】

特許文献１は、ＣＰＵが暗号専用命令を毎サイクルディスパッチすることによって、異なるデータブロックに対する独立した暗号化／復号を並列的に行い、暗号化／復号のスループットを向上させる技術を開示している（例えば、図４を参照）。

【０００６】

特許文献２は、ＣＰＵが、平文データと鍵ストリームの排他的論理和を取って暗号文データを生成するストリーム暗号処理にＳＩＭＤ命令（１回の命令で複数データに同じ処理を行うことができる命令）を適用し、その性能を倍増させる技術を開示している（ＴＡＢＬＥ３を参照）。

【０００７】

情報処理装置としての信頼性を向上させるために、ホストコンピュータから受信したユーザデータの処理において、ユーザデータを誤って壊していないか、別のユーザデータと取り違えていないかを確認できるように、ユーザデータにＤＩＦ（データ保全性フィールド）と呼ばれる管理情報を付加する機能を有するストレージシステムも知られている。前述の機能を有するストレージシステムは、ライト処理及びリード処理において、ＤＩＦの生成及び検査等を行う。ストレージシステムがＤＩＦの生成及び検査を支援するハードウェアを有していない場合、ストレージシステムが備えるＣＰＵが、複数のＣＰＵ命令で構成されたプログラムコードに従ってＤＩＦの生成及び検査を実行する。

【先行技術文献】

【特許文献】

【０００８】
【文献】米国特許８１９４８５４号明細書
【文献】米国特許１０３２０５５８号明細書
【発明の概要】
【発明が解決しようとする課題】
【０００９】
ユーザデータを暗号化して保存する機能及びユーザデータにＤＩＦを付加する機能を有するストレージシステムにおいて、ＣＰＵが実行するユーザデータの暗号化の処理性能を高めるための詳細な方法については従来の技術には開示されていない。
【課題を解決するための手段】
【００１０】
本願において開示される発明の代表的な一例を示せば以下の通りである。すなわち、複数のデータブロックを含むユーザデータの書き込み及び読み出しを行う計算機であって、プロセッサ、前記プロセッサに接続されるメモリ、前記プロセッサに接続される記憶媒体、及び前記プロセッサに接続される接続インタフェースを備え、前記プロセッサは、複数のレジスタを有し、前記ユーザデータの書き込みリクエストを受信し、前記書き込みリクエストに従って書き込まれる前記ユーザデータを前記メモリに格納し、前記複数のレジスタを用いて、複数の暗号化データブロックを含む暗号化ユーザデータを生成する暗号化処理を実行し、前記暗号化ユーザデータに含まれる前記複数の暗号化データブロックの各々に、第１誤り符号を含む第１データ保全性フィールドを付加し、前記暗号化ユーザデータを前記記憶媒体に格納する。前記暗号化処理では、所定の数の前記データブロックから、前記データブロックの一部である部分データを読み出し、第１レジスタに格納する第１処理と、前記第１レジスタに格納される前記部分データを暗号化した暗号化部分データを第２レジスタに格納する第２処理と、前記第２レジスタに格納される前記暗号化部分データを用いて、前記第１誤り符号を算出するための第１演算を実行し、前記第１演算の結果を第３レジスタに格納する第３処理と、が繰り返し実行される。
【発明の効果】
【００１１】
本発明の一形態によれば、暗号化処理の処理性能を向上させることができる。上記した以外の課題、構成及び効果は、以下の実施例の説明により明らかにされる。
【図面の簡単な説明】
【００１２】
【図１】実施例１の計算機システムの構成例を示す図である。
【図２Ａ】実施例１のストレージシステムが扱うデータのフォーマットを示す図である。
【図２Ｂ】実施例１のストレージシステムが扱うデータのフォーマットを示す図である。
【図２Ｃ】実施例１のストレージシステムが扱うデータのフォーマットを示す図である。
【図３Ａ】実施例１のＷＳＣの仕様を説明する図である。
【図３Ｂ】実施例１のＷＳＣの仕様を説明する図である。
【図３Ｃ】実施例１のＷＳＣの仕様を説明する図である。
【図４】実施例１のストレージシステムが実行するライト処理の一例を説明するフローチャートである。
【図５】実施例１のストレージシステムが実行するリード処理の一例を説明するフローチャートである。
【図６Ａ】従来のＸＴＳ暗号化／復号を説明する図である。
【図６Ｂ】従来のＸＴＳ暗号化／復号を説明する図である。
【図７Ａ】実施例１のストレージシステムにおけるＸＴＳ暗号化／復号を説明する図である。
【図７Ｂ】実施例１のストレージシステムにおけるＸＴＳ暗号化／復号を説明する図である。
【図７Ｃ】実施例１のストレージシステムにおけるＸＴＳ暗号化／復号を説明する図であ

る。
【図８】実施例１のストレージシステムが実行するＶｅｃｔｏｒ命令を用いたＸＴＳ暗号化／復号における入力データの構造を示す図である。
【図９】実施例１のストレージシステムが実行するＶｅｃｔｏｒ命令を用いたＸＴＳ暗号化／復号における出力データの構造を示す図である。
【図１０Ａ】実施例１のストレージシステムにおけるＣＲＣの算出方法を説明する図である。
【図１０Ｂ】実施例１のストレージシステムにおけるＣＲＣの算出方法を説明する図である。
【図１０Ｃ】実施例１のストレージシステムにおけるＣＲＣの算出方法を説明する図である。
【図１１Ａ】実施例１のストレージシステムにおけるＣＲＣ計算の並列方法を示す図である。
【図１１Ｂ】実施例１のストレージシステムにおけるＣＲＣ計算の並列方法を示す図である。
【図１２Ａ】実施例１のストレージシステムにおける４つのデータブロックのＤＩＦの並列的な生成方法を説明する図である。
【図１２Ｂ】実施例１のストレージシステムにおける４つのデータブロックのＤＩＦの並列的な生成方法を説明する図である。
【図１３Ａ】実施例１のストレージシステムにおける４つのデータブロックのＤＩＦの並列的な検査方法を説明する図である。
【図１３Ｂ】実施例１のストレージシステムにおける４つのデータブロックのＤＩＦの並列的な検査方法を説明する図である。
【図１３Ｃ】実施例１のストレージシステムにおける４つのデータブロックのＤＩＦの並列的な検査方法を説明する図である。
【発明を実施するための形態】
【００１３】
本発明の一実施例について、図面を参照して説明する。ただし、本発明は以下に示す実施例の記載内容に限定して解釈されるものではない。本発明の思想ないし趣旨から逸脱しない範囲で、その具体的構成を変更し得ることは当業者であれば容易に理解される。
【００１４】
以下に説明する発明の構成において、同一又は類似する構成又は機能には同一の符号を付し、重複する説明は省略する。
【００１５】
本明細書等における「第１」、「第２」、「第３」等の表記は、構成要素を識別するために付するものであり、必ずしも、数又は順序を限定するものではない。
【実施例１】
【００１６】
図１は、実施例１の計算機システムの構成例を示す図である。
【００１７】
実施例１に係る計算機システム１００は、ホスト計算機１１０及びストレージシステム１２０を有する。ホスト計算機１１０及びストレージシステム１２０は、直接又はネットワークを介して互いに接続される。
【００１８】
ストレージシステム１２０は、データの読み書きを行う計算機の一例である。ストレージシステム１２０は、ストレージコントローラ１３０及びストレージデバイス１４０を有する。ストレージデバイス１４０は、例えば、ＨＤＤ（Ｈａｒｄ　Ｄｉｓｋ　Ｄｒｉｖｅ）及びＳＳＤ（Ｓｏｌｉｄ　Ｓｔａｔｅ　Ｄｒｉｖｅ）等である。ストレージデバイス１４０は、ストレージコントローラ１３０に搭載されてもよい。
【００１９】

ストレージコントローラ１３０は、１つ以上のフロントエンドインタフェース（ＦＥＩ／Ｆ）１３１、１つ以上のバックエンドインタフェース（ＢＥ　Ｉ／Ｆ）１３２、ＣＰＵ１３３、及びＤＲＡＭ（Ｄｙｎａｍｉｏ　Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）１３４を有する。ＤＲＡＭ１３４は、バイト単位で読み書きが可能な揮発性メモリ（メモリデバイス）である。

【００２０】

図１では、ストレージシステム１２０のハードウェア要素を一つずつ図示しているが、冗長化、高性能化、又は大容量化等を実現するために、各構成要素は複数存在してもよい。

【００２１】

実施例１に係るストレージコントローラ１３０は、ホスト計算機１１０に論理ボリュームを提供する。提供方法としては、以下のような方法がある。

【００２２】

（提供方法１）複数のストレージデバイス１４０から一つないし複数の論理ボリューム（実体的な論理ボリューム）を形成して、ホスト計算機１１０に提供する。

【００２３】

（提供方法２）ストレージコントローラ１３０は、シンプロビジョニング技術によって形成される論理ボリュームをホスト計算機１１０に提供する。当該論理ボリュームは仮想的なボリュームであって、実際の記憶領域は動的に割り当てられる。

【００２４】

ホスト計算機１１０は、提供された論理ボリューム及び論理ボリューム内の位置（論理ブロック番号。ＬＢＡと略記されることもある）を指定したＩ／Ｏコマンド（ライトコマンド又はリードコマンド）を発行することによって、論理ボリュームに対するユーザデータのライト／リードを要求する。

【００２５】

ＦＥ　Ｉ／Ｆ１３１は、ホスト計算機１１０と通信するためのインタフェースデバイスである。ＦＥ　Ｉ／Ｆ１３１はホスト計算機１１０から受信したユーザデータにＤＩＦ（データ保全性フィールド）と呼ばれる情報を付加する機能を有する。また、ＦＥ　Ｉ／Ｆ１３１は、ホスト計算機１１０へ送信するデータに付加されたＤＩＦを検査し、除去する機能を有する。ＤＩＦは、例えばＴ１０－ＤＩＦである。Ｔ１０－ＤＩＦは、米国規格協会（ＡＮＳＩ）によって標準化されている。

【００２６】

ＢＥ　Ｉ／Ｆ１３２は、ストレージデバイス１４０と通信するためのインタフェースデバイスである。ＢＥ　Ｉ／Ｆ１３２は、例えばＳＡＳ、ＳＡＴＡ、ＮＶＭｅ、及びＦｉｂｒｅ　Ｃｈａｎｎｅｌ等である。

【００２７】

ＣＰＵ１３３は、ライト処理／リード処理で用いる複数のレジスタを有し、Ｉ／Ｏコマンド（ライトコマンド）に応じてユーザデータのライト処理を実行し、Ｉ／Ｏコマンド（リードコマンド）に応じてユーザデータのリード処理を実行する。ライト処理では、ホスト計算機１１０から受信したユーザデータが暗号化され、暗号化ユーザデータがストレージデバイス１４０に永続的に格納される。リード処理では、ストレージデバイス１４０に格納された暗号化ユーザデータが、ユーザデータに復号され、ホスト計算機１１０に送信される。これらライト／リード処理の内容は以降に詳しく説明する。

【００２８】

ＣＰＵ１３３は、例えば、Ｉｎｔｅｌ（Ｉｎｔｅｌは登録商標、以下同じ。）社がサーバ向けに開発したマイクロプロセッサである、第３世代Ｘｅｏｎ（Ｘｅｏｎは登録商標、以下同じ。）スケーラブルプロセッサ（コードネーム：ＩｃｅＬａｋｅ－ＳＰ）である。

【００２９】

ＤＲＡＭ１３４は、ＣＰＵ１３３が実行するプログラム、及びプログラムが扱うデータを格納する。また、ＤＲＡＭ１３４はキャッシュ領域を含む。キャッシュ領域には、ホスト計算機１１０から受信したＩ／Ｏコマンド（ライトコマンド）に従って入力されたユー

ザデータ、ストレージデバイス１４０に書き込まれる暗号化ユーザデータ、ストレージデバイス１４０から読み出された暗号化ユーザデータ、ホスト計算機１１０から受信したＩ／Ｏコマンド（リードコマンド）に従って出力されるユーザデータがキャッシュされる。

【００３０】

なお、ＣＰＵ１３３は、Ｉ／Ｏコマンド（ライトコマンド）とともに入力されたユーザデータをキャッシュ領域にキャッシュする場合、ユーザデータを多重化して、格納する。ＣＰＵ１３３は、キャッシュされたユーザデータの破損が検出された場合、多重化されたユーザデータを用いてライト処理等を行う。これによって、ユーザデータの損失を防止することができる。

【００３１】

また、ストレージシステム１２０は、ストレージデバイス１４０の故障による暗号化ユーザデータの損失を防ぐために、ＲＡＩＤ（Ｒｅｄｕｎｄａｎｔ Ａｒｒａｙｓ ｏｆ Ｉｎｄｅｐｅｎｄｅｎｔ Ｄｉｓｋｓ）技術に基づく、冗長化を施してから暗号化ユーザデータを格納する。具体的には、Ｎ台（Ｎは２以上の整数）のストレージデバイス１４０が搭載されている場合、ＣＰＵ１３３は、書き出すデータを（Ｎ－１）に均等に配分して各ストレージデバイス１４０に記録し、各ストレージデバイス１４０に書き込むデータの排他的論理和によって生成されたパリティを１つのストレージデバイス１４０に格納する。これによって、１つのストレージデバイス１４０が故障しても、データ回復が可能となる。例えば、Ｎ＝４のとき、ＣＰＵ１３３は、３つのストレージデバイス１４０に同じサイズのデータＤ１、Ｄ２、Ｄ３を記録して、１つのストレージデバイス１４０にデータＤ１、Ｄ２、Ｄ３の排他的論理和から生成されるパリティＰ（Ｐ＝Ｄ１＋Ｄ２＋Ｄ３：＋は排他的論理和を示す）を記録する。例えば、Ｄ２が記録されるストレージデバイス１４０が故障した場合、ＣＰＵ１３３は、パリティＰ、データＤ１、及びデータＤ３の排他的論理和によってデータＤ２を回復する。

【００３２】

図２Ａ、図２Ｂ、及び図２Ｃは、実施例１のストレージシステム１２０が扱うデータのフォーマットを示す図である。

【００３３】

図２Ａに示すデータフォーマット２００は、ホスト計算機１１０から送信されたライトコマンドとともに送信されるユーザデータ、又は、リードコマンドによってストレージシステム１２０からホスト計算機１１０に送信されるユーザデータのフォーマットの一例である。

【００３４】

ユーザデータは、１つ以上のユーザデータブロック２０１から構成される。ユーザデータブロック２０１のサイズは、例えば、５１２バイトである。図２に示すユーザデータは４つのユーザデータブロック２０１－１、２０１－２、２０１－３、２０１－４から構成された２０４８バイトのデータである。例えば、ホスト計算機１１０が８１９２バイトのユーザデータをライト又はリードする場合、ユーザデータは１６個のユーザデータブロック２０１から構成される。

【００３５】

図２Ｂに示すデータフォーマット２１０は、ストレージシステム１２０がＤＲＡＭ１３４にキャッシュするユーザデータのフォーマットの一例である。

【００３６】

ストレージシステム１２０のＦＥ Ｉ／Ｆ１３１は、ユーザデータを構成する各ユーザデータブロック２０１の末尾にＤＩＦ２０２を付加する。図２Ｂでは、ユーザデータブロック２０１－１、２０１－２、２０１－３、２０１－４の各々に、ＤＩＦ２０２－１、２０２－２、２０２－３、２０２－４が付加されている。

【００３７】

ＤＩＦ２０２はＴ１０－ＤＩＦの規格に従っており、２バイトのＣＲＣ（Ｃｙｃｌｉｃ Ｒｅｄｕｎｄａｎｃｙ Ｃｈｅｃｋ）２３１、２バイトのＡＴＡＧ（Ａｐｐｌｉｃａｔｉ

ｏｎ　Ｔａｇ）（０）２３２、２バイトのＡＴＡＧ（１）２３３、４バイトのＲＴＡＧ（Ｒｅｆｅｒｅｎｃｅ　Ｔａｇ）２３４から構成される。

【００３８】

ＣＲＣ２３１は、ＤＩＦ２０２の前に位置する５１２バイトのユーザデータブロック２０１から算出される１６ビットの誤り検出符号である。ＲＴＡＧ２３４は、ＤＩＦ２０２の前に位置する５１２バイトのユーザデータブロック２０１に対応づけられたアドレスである。ＲＴＡＧ２３４に設定するアドレスは、ユーザデータの先頭のユーザデータブロック２０１から終端のユーザデータブロック２０１に向かって昇順に設定される。例えば、ユーザデータブロック２０１－１のアドレスがＫ（Ｋは整数）である場合、ユーザデータブロック２０１－２のアドレスはＫ＋１、ユーザデータブロック２０１－３のアドレスはＫ＋２、ユーザデータブロック２０１－４のアドレスはＫ＋３である。ＡＴＡＧ（０）２３２及びＡＴＡ（１）２３３は任意の用途に用いることができる。ユーザデータをＤＲＡＭ１３４にキャッシュしている場合、ＡＴＡＧ（０）２３２及びＡＴＡＧ（１）２３３は使用されないため、ストレージシステム１２０はＡＴＡＧ（０）２３２及びＡＴＡＧ（１）２３３の各々に００ｈを設定する。

【００３９】

ストレージシステム１２０は、ＤＲＡＭ１３４に格納されるユーザデータの各ユーザデータブロック２０１にＤＩＦ２０２を付加することによって、ライト処理／リード処理におけるエラーの発生を検出できる。具体的には、ストレージシステム１２０は、ＣＲＣ２３１を検査することによって、ユーザデータブロック２０１の破損を検出できる。ストレージシステム１２０は、ＲＴＡＧ２３４を検査することによって、ユーザデータブロック２０１の順序が入れ替わりを検出できる。ＦＥ　Ｉ／Ｆ１３１がホスト計算機１１０にユーザデータを送信する場合に行うＤＩＦ検査は、前述の検査である。

【００４０】

図２Ｃに示すデータフォーマット２２０は、ストレージシステム１２０がストレージデバイス１４０に保存する暗号化ユーザデータのフォーマットの一例である。図２Ｃに示す暗号化ユーザデータは、ユーザデータブロック２０１－１、２０１－２、２０１－３、２０１－４を暗号化した暗号化ユーザデータブロック２０３－１、２０３－２、２０３－３、２０３－４から構成される。

【００４１】

ストレージシステム１２０は、ユーザデータを構成する各ユーザデータブロック２０１をＡＥＳ（Ａｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）アルゴリズムのＸＴＳ（ＸＥＸ ｅｎｃｒｙｐｔｉｏｎ ｍｏｄｅ ｗｉｔｈ Ｔｗｅａｋ ａｎｄ ｃｉｐｈｅｒｔｅｘｔ　Ｓｔｅａｌｉｎｇ）モードを用いて、暗号化ユーザデータブロック２０３に暗号化する。

【００４２】

ＡＥＳは、米国国立標準技術研究所（ＮＩＳＴ）が標準暗号として定めた共通鍵暗号アルゴリズムである。ＡＥＳ鍵のサイズは、例えば２５６ビットである。ただし、ＡＳＥ鍵のサイズは２５６ビットでなくてもよい。ＸＴＳモードは、標準化文章ＩＥＥＥ１６１９で制定されているストレージデバイス向けのブロック暗号方式である。以下の説明では、ＡＥＳアルゴリズムのＸＴＳモードによる暗号化をＸＴＳ暗号化、当該モードによる復号をＸＴＳ復号と記載する。

【００４３】

ストレージシステム１２０は、ユーザデータのキャッシュと同様に、各暗号化ユーザデータブロック２０３にＤＩＦ２０４を付加する。

【００４４】

ＤＩＦ２０４は、ＤＩＦ２０２と同様にＴ１０－ＤＩＦの規格に従っており、２バイトのＣＲＣ２４１、２バイトのＡＴＡＧ（０）２４２、２バイトのＡＴＡＧ（１）２４３、４バイトのＲＴＡＧ２４４から構成される。

【００４５】

ＣＲＣ２４１は、ＤＩＦ２０４の前に位置する５１２バイトの暗号化ユーザデータブロック２０３から算出される１６ビットの誤り検出符号である。ＲＴＡＧ２４４は、ＤＩＦ２０４の前に位置する５１２バイトの暗号化ユーザデータブロック２０３に対応づけられたアドレスである。ＲＴＡＧ２４４に設定するアドレスは、暗号化ユーザデータの先頭の暗号化ユーザデータブロック２０３から終端の暗号化ユーザデータブロック２０３に向かって昇順に設定される。ストレージシステム１２０は、暗号化ユーザデータをストレージデバイス１４０に保存する場合、ＡＴＡＧ（０）２４２にＷＳＣ（ライトシーケンスコード）と呼ばれる符号を設定する。ＷＳＣの仕様については図３を用いて説明する。ＡＴＡＧ（１）２４３は使用しないため、００ｈが設定される。

【００４６】

ストレージシステム１２０は、ストレージデバイス１４０に保存される暗号化ユーザデータの各暗号化ユーザデータブロック２０３にＤＩＦ２０４を付加することによって、書き込みエラー又は読み出しエラーの発生を検出できる。具体的には、ストレージシステム１２０は、ＣＲＣ２４１を検査することによって、読み出した暗号化ユーザデータブロック２０３の破損を検出できる。ストレージシステム１２０は、ＲＴＡＧ２４４を検査することによって、ストレージデバイス１４０に読み出しを指示したアドレスの誤りを検出できる。ストレージデバイス１４０への暗号化ユーザデータの書き込みのエラーは、ＡＴＡＧ（０）２４２に設定されたＷＳＣを検査することによって検出できる。

【００４７】

以下の説明では、ユーザデータブロック２０１及び暗号化ユーザデータブロック２０３を区別しない場合、データブロックと記載する。

【００４８】

図３Ａ、図３Ｂ、及び図３Ｃは、実施例１のＷＳＣの仕様を説明する図である。

【００４９】

図３Ａに示すデータフォーマット３００は、ＷＳＣのデータフォーマットの一例である。ＷＳＣは、ＡＴＡＧ０（２４２）に格納される１バイト（８ビット）の符号であり、１ビットのＨＥＡＤフラグ３０１、１ビットのＴＡＩＬフラグ３０２、６ビットのＳＱＮ（シーケンス番号）３０３から構成される。

【００５０】

ＨＥＡＤフラグ３０１は、ストレージデバイス１４０に書き込む暗号化ユーザデータブロック２０３が先頭であるか否かを示す値を格納するフラグである。ＴＡＩＬフラグ３０２は、ストレージデバイス１４０に書き込む暗号化ユーザデータブロック２０３が終端であるか否かを示す値を格納するフラグである。

【００５１】

ストレージデバイス１４０に書き込む暗号化ユーザデータブロック２０３が先頭である場合、ＨＥＡＤフラグ３０１に１が設定され、ストレージデバイス１４０に書き込む暗号化ユーザデータブロック２０３が終端である場合、ＴＡＩＬフラグ３０２に１が設定される。先頭及び終端のいずれにも該当しない暗号化ユーザデータブロック２０３のＨＥＡＤフラグ３０１及びＴＡＩＬフラグ３０２には０が設定される。１つの暗号化ユーザデータブロック２０３から構成される暗号化ユーザデータを書き込む場合、ＨＥＡＤフラグ３０１及びＴＡＩＬフラグ３０２には１が設定される。

【００５２】

ＳＱＮ３０３はストレージデバイス１４０に書き込まれる暗号化ユーザデータの世代番号を格納する。世代番号は、書き込むたびに異なる値を用いる。

【００５３】

図３Ｂは、ストレージデバイス１４０のアドレス０～９に暗号化ユーザデータブロック２０３を書き込む場合のＷＳＣの設定例を示す。

【００５４】

状態１では、１０個の暗号化ユーザデータブロック２０３から構成される暗号化ユーザデータが書き込まれる。ここではＳＱＮとして０ｘ０Ａが指定されているものとする。

【００５５】
アドレス０に書き込まれた暗号化ユーザデータブロック２０３は先頭であるためＨＥＡＤ３０１は１となり、ＷＳＣは０ｘ８Ａとなる。アドレス９に書き込まれた暗号化ユーザデータブロック２０３は終端であるためＴＡＩＬ３０２が１となり、ＷＳＣは０ｘ４Ａとなる。他のアドレスに書き込まれた暗号化ユーザデータブロック２０３のＷＳＣは０ｘ０Ａとなる。
【００５６】
状態２では、アドレス３～７に対して５つの暗号化ユーザデータブロック２０３から構成される暗号化ユーザデータが書き込まれる。ここではＳＱＮには０ｘ０Ｂが指定されているものとする。
【００５７】
５つのアドレスに対してデータが上書きされるためＷＳＣが更新される。アドレス３に書き込まれた暗号化ユーザデータブロック２０３は先頭であるためＨＥＡＤ３０１は１となり、ＷＳＣは０ｘ８Ｂとなる。アドレス７に書き込まれた暗号化ユーザデータブロック２３０は終端であるためＴＡＩＬ３０２は１となり、ＷＳＣは０ｘ４Ｂとなる。アドレス４～６に書き込まれた暗号化ユーザデータブロック２０３のＷＳＣは０ｘ０Ｂとなる。
【００５８】
図３Ｃに示す表３２０は、ＷＳＣに基づいて書き込みエラーの有無を検査するためのルールの一例を示す。ストレージシステム１２０は、ストレージデバイス１４０に暗号化ユーザデータを書き込む場合、ＷＳＣを読み出して、表３２０に示したルールに従って、書き込みエラーの有無を検査する。
【００５９】
表３２０には、以下のようなルールが定義されている。
（Ｒ１）自ブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが１／０又は１／１の場合、検査は行われない。
（Ｒ２）自ブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／０又は０／１、かつ、前のブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／１である場合、検査は行われない。
（Ｒ３）自ブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／０又は０／１、かつ、前のブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／０の場合、ストレージシステム１２０は、自ブロックのＷＳＣのＳＱＮが前ブロックのＷＳＣのＳＱＮと一致するか否かを検査する。
（Ｒ４）自ブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／０又は０／１、かつ、その前のブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが１／０の場合、ストレージシステム１２０は、自ブロックのＷＳＣのＳＱＮが前ブロックのＷＳＣのＳＱＮと一致するか否かを検査する。
（Ｒ５）自ブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが０／０又は０／１、かつ、前のブロックのＷＳＣのＨＥＡＤ／ＴＡＩＬが１／１の場合、検査は行われない。
【００６０】
（Ｒ３）、（Ｒ４）は、自ブロックが先頭でなく、前ブロックが終端でなかった場合、暗号化ユーザデータブロック２０３のＳＱＮの一致を検査することを表す。これによって、複数ブロックで構成されたユーザデータの書き込みが間欠なく成功しているか否かが確認できる。例えば、図３Ｂの矩形３１１、３１２、３１３に含まれるアドレスのＷＳＣは（Ｒ３）、（Ｒ４）の条件に合致するため検査対象となる。アドレス３～７に対する５つのブロックの書き込みにおいて、アドレス５への書き込みが失敗していた場合、アドレス５のＷＳＣは更新されず０ｘ０Ａのままとなる。このエラーはアドレス５のＷＳＣ検査において、アドレス５のＳＱＮ（０ｘ０Ａ）がアドレス４のＳＱＮ（０ｘ０Ｂ）と不一致となることで検出可能である。アドレス３～７に対する５つのブロックの書き込みにおいて、アドレス３への書き込みが失敗していた場合、アドレス３のＷＳＣは更新されず０ｘ０Ａのままとなる。このエラーはアドレス４のＷＳＣ検査において、ＳＱＮ（０ｘ０Ｂ）がアドレス３のＳＱＮ（０ｘ０Ａ）と不一致となることで検出可能である。
【００６１】
（Ｒ１）、（Ｒ２）、（Ｒ５）は、自ブロックが先頭である場合、又は、前のブロック

が終端である場合、自ブロックのＷＳＣのＳＱＮと、前のブロックのＷＳＣのＳＱＮとが一致している必要がないため、検査を行わないことを表す。

【００６２】

図４は、実施例１のストレージシステム１２０が実行するライト処理の一例を説明するフローチャートである。以下の説明では、ＸＴＳ暗号化されていないユーザデータブロック２０１を平文ブロック、ＸＴＳ暗号化によって暗号化ユーザデータブロック２０３を暗号文ブロックと記載する。

【００６３】

ストレージシステム１２０は、ホスト計算機１１０からライトコマンドとともにユーザデータを受信する（ステップＳ４０１）。

【００６４】

ストレージシステム１２０は、ユーザデータを構成する平文ブロックにＤＩＦ２０２を付加する（ステップＳ４０２）。具体的には、ＦＥ　Ｉ／Ｆ１３１は、平文ブロックから１６ビットの誤り検出符号を算出してＣＲＣ２３１として設定し、また、ブロックアドレスをＴＲＡＧ２３４として設定する。

【００６５】

ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０２を付加したユーザデータをＤＲＡＭ１３４に格納する（ステップＳ４０３）。このとき、ストレージシステム１２０は、ユーザデータの損失を防ぐために、ユーザデータを多重してＤＲＡＭ１３４に格納する。

【００６６】

ストレージシステム１２０のＣＰＵ１３３は、ＤＲＡＭ１３４から平文ブロックの一部を読み出し、レジスタにロードする（ステップＳ４０４）。

【００６７】

ストレージシステム１２０のＣＰＵ１３３は、レジスタにロードされたデータを用いて平文ブロックのＣＲＣを算出するための演算（ＣＲＣ計算）を実行し、ＸＴＳ暗号化によってデータを暗号化する（ステップＳ４０５）。これによって、暗号文ブロックの一部が生成される。暗号化されたデータはレジスタに格納され、また、ＤＲＡＭ１３４にも格納される。

【００６８】

ストレージシステム１２０のＣＰＵ１３３は、レジスタに格納される暗号化されたデータを用いて、暗号文ブロックのＣＲＣを算出するためのＣＲＣ計算を実行する（ステップＳ４０６）。

【００６９】

ストレージシステム１２０のＣＰＵ１３３は、全ての平文ブロックの暗号化が完了したか否かを判定する（ステップＳ４０７）。

【００７０】

全ての平文ブロックの暗号化が完了していない場合、ストレージシステム１２０のＣＰＵ１３３は、ステップＳ４０４に戻り同様の処理を実行する。このように、ストレージシステム１２０は、平文ブロックの先頭から終端に向かってロード元ポイントを順次移動させ、逐次的にデータを暗号化する。

【００７１】

全ての平文ブロックの暗号化が完了した場合、平文ブロック及び暗号文ブロックのそれぞれのＣＲＣが算出され、また、暗号化ユーザデータが生成された状態となっている。

【００７２】

ストレージシステム１２０のＣＰＵ１３３は、算出された平文ブロックのＣＲＣと、ＤＩＦ２０２に含まれるＣＲＣ２３１とを比較することによって、ユーザデータが破損しているか否かを判定する（ステップＳ４０８）。具体的には、各ＣＲＣが一致しているか否かが判定される。少なくとも一つのＣＲＣが不一致の場合、ユーザデータが破損していると判定される。

【００７３】
ＣＲＣの検査によってユーザデータの破損が検出された場合、ストレージシステム１２０のＣＰＵ１３３は、多重化されたユーザデータを用いてユーザデータを回復し（ステップＳ４１０）、その後、ステップＳ４０４に戻る。
【００７４】
ＣＲＣの検査によってユーザデータの破損が検出されなかった場合、ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０２に含まれるＲＴＡＧ２３４に基づいて、ユーザデータが破損しているか否かを判定する（ステップＳ４０９）。具体的には、アドレスが昇順になっているか否かが判定される。アドレスが昇順になっていない場合、ユーザデータが破損していると判定される。
【００７５】
ＲＴＡＧ２３４の検査によってユーザデータの破損が検出された場合、ストレージシステム１２０のＣＰＵ１３３は、多重化されたユーザデータを用いてユーザデータを回復し（ステップＳ４１０）、その後、ステップＳ４０４に戻る。
【００７６】
ＲＴＡＧ２３４の検査によってユーザデータの破損が検出されなかった場合、ストレージシステム１２０のＣＰＵ１３３は、各暗号文ブロックにＤＩＦ２０４を付加する（ステップＳ４１１）。具体的には、暗号文ブロックから算出された１６ビットの誤り検出符号がＣＲＣ２４１に設定され、図３で説明したＷＳＣがＡＴＡＧ０（２４２）に設定され、暗号文ブロックのアドレスがＲＴＡＧ２３４に設定される。
【００７７】
ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０４が付加された暗号化ユーザデータをストレージデバイス１４０にライトする（ステップＳ４１２）。この時、ＣＰＵ１３３は、Ｎ－１等分に分割された暗号化ユーザデータと、ＲＡＩＤ技術によって生成されたパリティとを複数のストレージデバイス１４０にライトする。以上がライト処理の説明である。
【００７８】
図５は、実施例１のストレージシステム１２０が実行するリード処理の一例を説明するフローチャートである。
【００７９】
ストレージシステム１２０のＣＰＵ１３３は、ホスト計算機１１０からリードコマンドを受信した場合、ストレージデバイス１４０からＤＩＦ２０４が付加された暗号化ユーザデータを読み出し、ＤＲＡＭ１３４に格納する（ステップＳ５０１）。
【００８０】
ストレージシステム１２０のＣＰＵ１３３は、ＤＲＡＭ１３４から暗号文ブロックの一部をレジスタにロードする（ステップＳ５０２）。
【００８１】
ストレージシステム１２０のＣＰＵ１３３は、レジスタにロードされたデータを用いて暗号文ブロックのＣＲＣを算出するための演算（ＣＲＣ計算）を実行し、ＸＴＳ復号によってデータを復号する（ステップＳ５０３）。これによって、平文ブロックの一部が生成される。復号されたデータはレジスタに格納され、また、ＤＲＡＭ１３４にも格納される。
【００８２】
ストレージシステム１２０のＣＰＵ１３３は、レジスタに格納される復号されたデータを用いて、平文ブロックのＣＲＣを算出するための演算（ＣＲＣ計算）を実行する（ステップＳ５０４）。
【００８３】
ストレージシステム１２０のＣＰＵ１３３は、全ての暗号文ブロックの復号が完了したか否かを判定する（ステップＳ５０５）。
【００８４】
全ての暗号文ブロックの復号が完了していない場合、ストレージシステム１２０のＣＰ

Ｕ１３３は、ステップＳ５０２に戻り同様の処理を実行する。このように、暗号文ブロックの先頭から終端に向かってロード元ポイントを順次移動させ、逐次的にデータを復号する。

【００８５】

全ての暗号文ブロックの復号が完了した場合、暗号文ブロック及び平文ブロックのそれぞれのＣＲＣが算出され、また、ユーザデータが生成された状態となっている。

【００８６】

ストレージシステム１２０のＣＰＵ１３３は、算出された暗号文ブロックのＣＲＣと、ＤＩＦ２０４に含まれるＣＲＣ２４１とを比較することによって、暗号化ユーザデータが破損しているか否かを判定する（ステップＳ５０６）。すなわち、暗号化ユーザデータのリード又はライトに失敗しているか否かが判定される。具体的には、各ＣＲＣが一致しているか否かが判定される。少なくとも１つのＣＲＣが不一致である場合、暗号化ユーザデータが破損していると判定される。

【００８７】

ＣＲＣの検査によって暗号化ユーザデータの破損が検出された場合、ストレージシステム１２０のＣＰＵ１３３は、ストレージシステム１２０からパリティを読み出し、暗号化ユーザデータを回復し（ステップＳ５０９）、その後、ステップＳ５０２に戻る。

【００８８】

ＣＲＣの検査によって暗号化ユーザデータの破損が検出されなかった場合、ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０４に含まれるＷＳＣに基づいて、暗号化ユーザデータが破損しているか否かを判定する（ステップＳ５０７）。すなわち、暗号化ユーザデータのリード又はライトに失敗しているか否かが判定される。具体的には、図３Ｃに示した表３２０に従った検査が行われる。

【００８９】

ＷＳＣの検査によって暗号化ユーザデータの破損が検出された場合、ストレージシステム１２０のＣＰＵ１３３は、ストレージシステム１２０からパリティを読み出し、暗号化ユーザデータを回復し（ステップＳ５０９）、その後、ステップＳ５０２に戻る。

【００９０】

ＷＳＣの検査によって暗号化ユーザデータの破損が検出されなかった場合、ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０４に含まれるＲＴＡＧ２３４に基づいて、暗号化ユーザデータが破損しているか否かを判定する（ステップＳ５０８）。すなわち、暗号化ユーザデータのリード又はライトに失敗しているか否かが判定される。具体的には、アドレスが昇順になっているか否かが判定される。アドレスが昇順になっていない場合、暗号化ユーザデータが破損していると判定される。

【００９１】

ＲＴＡＧ２３４の検査によって暗号化ユーザデータの破損が検出された場合、ストレージシステム１２０のＣＰＵ１３３は、ストレージシステム１２０からパリティを読み出し、暗号化ユーザデータを回復し（ステップＳ５０９）、その後、ステップＳ５０２に戻る。

【００９２】

ＲＴＡＧ２３４の検査によってユーザデータの破損が検出されなかった場合、ストレージシステム１２０のＣＰＵ１３３は、各平文ブロックにＤＩＦ２０２を付加する（ステップＳ５１０）。具体的には、平文ブロックから算出された１６ビットの誤り検出符号がＣＲＣ２３１に設定され、平文ブロックのアドレスがＲＴＡＧ２３４に設定される。

【００９３】

ストレージシステム１２０のＣＰＵ１３３は、ＤＩＦ２０２が付加されたユーザデータをＤＲＡＭ１３４に格納する（ステップＳ５１１）。

【００９４】

ストレージシステム１２０は、ホスト計算機１１０にユーザデータを送信する（ステップＳ５１２）。この時、ＦＥ　Ｉ／Ｆ１３１は、ＤＩＦの検査を行って、ユーザデータからＤＩＦを除去する。以上がリード処理である。

【００９５】
図６Ａ及び図６Ｂは、従来のＸＴＳ暗号化／復号を説明する図である。
【００９６】
図６Ａは、ＸＴＳ暗号化の処理手順を示す。ＸＴＳ暗号化では、５１２バイトの平文ブロックが１６バイトずつ暗号化される。まず、（Ｅ１）１６バイトのＴｗｅａｋと平文ブロックの一部との排他的論理和が算出される。次に、（Ｅ２）ＡＥＳ暗号化アルゴリズムに基づく暗号化が行われる。最後に、（Ｅ３）暗号化の結果とＴｗｅａｋとの排他的論理和が算出され、１６バイトの暗号文要素が生成される。（Ｅ１）、（Ｅ２）、（Ｅ３）の処理のサイクルを３２回実行することによって５１２バイトの暗号文ブロックが生成される。
【００９７】
Ｔｗｅａｋは１６バイトのデータを暗号化する毎に変更される。また、Ｔｗｅａｋは５１２バイトのブロック毎に異なる値である。
【００９８】
（Ｅ２）のＡＥＳアルゴリズムでは１５個の暗号化ラウンド鍵（１２８ビット）を用いて暗号化が行われる。これらのラウンド鍵は、規定された拡張処理（Ｋｅｙ　Ｅｘｐａｎｓｉｏｎ）を２５６ビット暗号鍵に施して生成される。最初に、（Ｅ２－１）ＡｄｄＲｏｕｎｄＫｅｙ処理が実行され、次に、（Ｅ２－２）ＳｕｂＢｙｔｅｓ、ＳｈｉｆｔＲｏｗｓ、ＭｉｘＣｏｌｕｍｎｓ、ＡｄｄＲｏｕｎｄＫｅｙの４つの処理が１３回繰り返され、最後に、（Ｅ２－３）ＳｕｂＢｙｔｅｓ、ＳｈｉｆｔＲｏｗｓ、ＡｄｄＲｏｕｎｄＫｅｙの３つの処理が実行される。１５回実行されるＡｄｄＲｏｕｎｄＫｅｙ処理では、暗号化ラウンド鍵０～１４が使用される。
【００９９】
前述した各処理は、以下のようなＣＰＵ命令に対応させることができる。データ及びＴｗｅａｋの排他的論理和はＸＯＲ命令に対応する。ＡｄｄＲｏｕｎｄＫｅｙ処理は、ＣＰＵ命令として、前段の処理結果と暗号化ラウンド鍵との排他的論理和（ＸＯＲ）に対応する。（Ｅ２－２）の４つの処理は、ＡＥＳＥＮＣ命令に対応する。つまり、ＡＥＳＥＮＣ命令は１３回実行する必要がある。（Ｅ２－３）の３つの処理は、ＡＥＳＥＮＣＬＡＳＴ命令に対応する。ＡＥＳＥＮＣ命令とＡＥＳＥＮＣＬＡＳＴ命令は、Ｉｎｔｅｌ社が２０１０年頃以降に製品化しているマイクロプロセッサにおいて、ＡＥＳ－ＮＩ（ＡＥＳ　Ｎｅｗ　Ｉｎｓｔｒｕｃｔｉｏｎｓ）と呼ばれる暗号専用命令セットの一部として実装されている。
【０１００】
図６Ｂは、ＸＴＳ復号の処理手順を示す。ＸＴＳ復号では、５１２バイトの暗号文ブロックが１６バイトずつ復号される。まず、（Ｄ１）１６バイトのＴｗｅａｋと暗号化ブロックの一部との排他的論理和が算出される。次に、（Ｄ２）ＡＥＳアルゴリズムに基づく復号が行われる。最後に、（Ｄ３）復号の結果とＴｗｅａｋとの排他的論理和が算出され、１６バイトの平文要素が生成される。（Ｄ１）、（Ｄ２）、（Ｄ３）を３２回実行することによって５１２バイトの平文ブロックが生成される。
【０１０１】
Ｔｗｅａｋは１６バイトのデータを復号する毎に変更される。また、Ｔｗｅａｋは５１２バイトのブロック毎に異なる値である。
【０１０２】
（Ｄ２）のＡＥＳ復号アルゴリズムでは１５個の復号ラウンド鍵（１２８ビット）を用いて復号が行われる。これらのラウンド鍵は、規定された拡張処理（Ｋｅｙ　Ｅｘｐａｎｓｉｏｎ）を２５６ビット暗号鍵に施して生成する。２５６ビット暗号鍵は暗号化で使用した暗号鍵と同一である。最初に、（Ｄ２－１）ＡｄｄＲｏｕｎｄＫｅｙ処理が実行され、次に、（Ｄ２－２）ＩｎｖＳｈｉｆｔＲｏｗｓ、ＩｎｖＳｕｂＢｙｔｅｓ、ＡｄｄＲｏｕｎｄＫｅｙ、ＩｎｖＭｉｘＣｏｌｕｍｎｓの４つの処理が１３回繰り返され、最後に、（Ｄ２－３）ＩｎｖＳｈｉｆｔＲｏｗｓ、ＩｎｖＳｕｂＢｙｔｅｓ、ＡｄｄＲｏｕｎｄＫ

ｅｙの３つの処理が実行される。１５回実行されるＡｄｄＲｏｕｎｄＫｅｙ処理では、復号ラウンド鍵０～１４が使用される。

【０１０３】

前述した各処理は、以下のようなＣＰＵ命令に対応させることができる。データ及びＴｗｅａｋの排他的論理和はＸＯＲ命令に対応する。ＡｄｄＲｏｕｎｄＫｅｙ処理は、前段の処理結果と復号ラウンド鍵との排他的論理和（ＸＯＲ）に対応する。（Ｄ２－２）の４つの処理は、ＡＥＳＤＥＣ命令に対応する。つまり、ＡＥＳＤＥＣ命令は１３回実行する必要がある。（Ｄ２－３）の３つの処理は、ＡＥＳＤＥＣＬＡＳＴ命令に対応する。ＡＥＳＤＥＣ命令とＡＥＳＤＥＣＬＡＳＴ命令は、上記と同様にＡＥＳ－ＮＩの一部として実装されている。

【０１０４】

図７Ａ、図７Ｂ、及び図７Ｃは、実施例１のストレージシステム１２０におけるＸＴＳ暗号化／復号を説明する図である。

【０１０５】

ＣＰＵ１３３は、第３世代Ｘｅｏｎスケーラブルプロセッサ等であり、ＡＥＳ－ＮＩをサポートし、さらにその中のＡＥＳＥＮＣ命令、ＡＥＳＥＮＣＬＡＳＴ命令、ＡＥＳＤＥＣ命令、ＡＥＳＤＥＣＬＡＳＴ命令のそれぞれをＶｅｃｔｏｒ化した命令（ＶＡＥＳＥＮＣ命令、ＶＡＥＳＥＮＣＬＡＳＴ命令、ＶＡＥＳＤＥＣ命令、ＶＡＥＳＤＥＣＬＡＳＴ命令）もサポートしている。

【０１０６】

Ｖｅｃｒｏｒ命令は、１回の実行において、複数データに対して同じ演算を並列に実施することができる命令である。ＳＩＭＤ（Ｓｉｎｇｌｅ Ｉｎｓｔｒｕｃｔｉｏｎ／Ｍｕｌｔｉｐｌｅ Ｄａｔａ）命令とも呼ばれる。Ｖｅｃｔｏｒ命令ではＺｍｍレジスタと呼ばれる５１２ビット（６４バイト）レジスタを使用し、１２８ビット（１６バイト）の４つのデータに対して同じ演算を実行する。演算結果もまたＺｍｍレジスタに格納される。ＣＰＵ１３３は、３２個のＺｍｍレジスタ（Ｚｍｍ０レジスタ～Ｚｍｍ３１レジスタ）を利用する。ＣＰＵ１３３は、マイクロプロセッサに一般的な演算（加算、減算、乗算、排他的論理和、シフト、比較など）を行う命令についても、Ｖｅｃｔｏｒ化した命令をサポートしている。

【０１０７】

ＣＰＵ１３３は、これらのＶｅｃｔｏｒ命令を用いてＸＴＳ暗号化／復号を並列的に実行する。具体的には、４つのデータブロック（５１２バイト）を並列で暗号化／復号する。

【０１０８】

図７Ａは、ＣＰＵ１３３による、４つの平文ブロックの並列的なＸＴＳ暗号化を示す。

【０１０９】

ＣＰＵ１３３は、ＤＲＡＭ１３４に格納された４つの平文ブロック（５１２バイト）の各々から１６バイトのデータをＺｍｍレジスタにロードする。Ｚｍｍレジスタには合計で６４バイトのデータが格納される。

【０１１０】

ＣＰＵ１３３は、（Ｐ１）Ｖｅｃｔｏｒ ＸＯＲ命令を実行することによって、Ｚｍｍレジスタに格納されるデータと、各平文ブロックの４つのＴｗｅａｋを並べた値との排他的論理和を算出する。次に、ＣＰＵ１３３は、（Ｐ２）Ｖｅｃｔｏｒ ＸＯＲ命令、１３回のＶＡＥＳＥＮＣ命令、及びＶＡＥＳＥＮＣＬＡＳＴ命令を実行することによって、４つの平文ブロックのＸＴＳ暗号化を並列で行う。最後に、ＣＰＵ１３３は、（Ｐ３）（Ｐ２）の演算結果と、４つのＴｗｅａｋを並べた値との排他的論理和を算出する。これによって、４つの１６バイトの暗号文要素が同時に生成される。

【０１１１】

暗号文要素はＺｍｍレジスタから逐次ＤＲＡＭ１３４に格納される。（Ｐ１）、（Ｐ２）、（Ｐ３）の処理のサイクルを複数回実行することによって、４つの暗号文ブロック（５１２バイト）が生成される。なお、（Ｐ２）で用いる暗号化ラウンド鍵０～１４は各平

文ブロックで共通である。
【０１１２】
図７Ｂは、ＣＰＵ１３３による、４つの暗号文ブロックの並列的なＸＴＳ復号を示す。
【０１１３】
ＣＰＵ１３３は、ＤＲＡＭ１３４に格納された４つの暗号文ブロック（５１２バイト）の各々から１６バイトのデータをＺｍｍレジスタにロードする。Ｚｍｍレジスタには合計で６４バイトのデータが格納される。
【０１１４】
ＣＰＵ１３３は、（Ｐ４）Ｖｅｃｔｏｒ　ＸＯＲ命令を実行することによって、Ｚｍｍレジスタに格納されるデータと、各暗号文ブロックの４つのＴｗｅａｋを並べた値との排他的論理和を算出する。次に、ＣＰＵ１３３は、（Ｐ５）Ｖｅｃｔｏｒ　ＸＯＲ命令、１３回のＶＡＥＳＤＥＣ命令、及びＶＡＥＳＤＥＣＬＡＳＴ命令を実行することによって、４つの暗号文ブロックのＸＴＳ復号を並列で行う。最後に、ＣＰＵ１３３は、（Ｐ６）（Ｐ５）の演算結果と、４つのＴｗｅａｋを並べた値との排他的論理和を算出する。これによって、４つの１６バイトの平文要素が同時に生成される。
【０１１５】
平文要素はＺｍｍレジスタから逐次ＤＲＡＭ１３４に格納される。（Ｐ４）、（Ｐ５）、（Ｐ６）の処理のサイクルを複数回実行することによって、４つの平文ブロック（５１２バイト）が生成される。なお、（Ｐ５）で用いる復号ラウンド鍵０～１４は各暗号文ブロックで共通である。
【０１１６】
図７Ｃは、Ｖｅｃｔｏｒ化した上記４種のＡＥＳ命令の実行順序を最適化する方法を示す。
【０１１７】
図７ＣのＶＡＥＳ＊は、ＶＡＥＳＥＮＣ、ＶＡＥＳＥＮＣＬＡＳＴ、ＶＡＥＳＤＥＣ、ＶＡＥＳＤＥＣＬＡＳＴに置き換えられる。
【０１１８】
これらの命令で用いる演算回路はそれぞれ４段のパイプライン構造７３１となっており、処理完了までのサイクル数は４である。パイプライン回路は、データがＮ段目から（Ｎ＋１）段目に移行した時に、Ｎ段目は次のデータを受け入れ可能である。そのため、ＶＡＥＳ＊命令を４つ並べて実行すると、パイプラインには４サイクル連続でデータを供給できるため、処理効率が向上する。この考え方に基づき、ＣＰＵ１３３は、図７Ａに示す暗号化及び図７Ｂに示す復号において、４つのデータブロックからそれぞれ１６バイトをＺｍｍレジスタにロードする場合に、４つのＺｍｍレジスタを確保して１６バイト×４のロードを４回分先行して行う。そして、ＣＰＵ１３３は、４つのＺｍｍレジスタに対する４つのＶＡＥＳ＊命令を連続で実行し、パイプライン処理の効率を向上させる。以下、４連続のＶＡＥＳ＊命令で処理されるデータのまとまりをＧｒｏｕｐと呼ぶ。
【０１１９】
図８は、実施例１のストレージシステム１２０が実行するＶｅｃｔｏｒ命令を用いたＸＴＳ暗号化／復号における入力データの構造を示す図である。
【０１２０】
４つのデータブロックのＸＴＳ暗号化／復号の並列実行においては、図７Ｃで示すＧｏｕｒｐは Ｇｏｕｒｐ０～７の８個から構成される。すなわち、「４つのＺｍｍレジスタに対して４つのＶＡＥＳ＊命令を連続で実行する」手順は、４つのデータブロックのＸＴＳ暗号化／復号の並列処理の１回の実行において８回実行される。
【０１２１】
１つのＧｒｏｕｐを構成する４つの６４バイトデータ（１６バイト×４）はそれぞれＺｍｍ０レジスタ～Ｚｍｍ３レジスタにロードされる。例えば、Ｚｍｍ０レジスタにロードされるデータは破線８１０で囲った６４バイトデータである。なお、ロード先のＺｍｍレジスタの番号は一例であって、他のレジスタ番号でもよい。

【０１２２】
ＤＲＡＭ１３４に格納されるユーザデータ又は暗号化ユーザデータの各データブロックの終端にはそれぞれ８バイトのＤＩＦが付加されている。４つのデータブロックのＤＩＦの検査は並列で実行されるため、１つのＺｍｍレジスタにＤＩＦがロードされる。図８では、Ｚｍｍ８レジスタに４つのデータブロックに付加されたＤＩＦがロードされている。
【０１２３】
Ｚｍｍレジスタは１６バイトのデータを４つ保持可能である。ただし、ＤＩＦは８バイトであるため、８バイトの間隔をあけてＺｍｍレジスタにロードされる。なお、ＤＩＦロード先のＺｍｍレジスタの番号は一例であって、他のレジスタ番号でもよい。
【０１２４】
ＸＴＳ暗号化／復号処理に同時に入力される４つのデータブロックのアドレスは昇順で連続しており、４Ｎ、４Ｎ＋１、４Ｎ＋２、４Ｎ＋３（Ｎは整数）である。例えば、ライト／リードするユーザデータのサイズが１２８Ｋバイト（２５６ブロック）の場合、４つのデータブロックのＸＴＳ暗号化／復号の並列処理を６４回行う必要がある。
【０１２５】
図９は、実施例１のストレージシステム１２０が実行するＶｅｃｔｏｒ命令を用いたＸＴＳ暗号化／復号における出力データの構造を示す図である。
【０１２６】
図９のＧｏｕｒｐ０～７は、図８のＧｏｕｒｐ０～７のデータをＸＴＳ暗号化／復号した結果をＤＲＡＭ１３４に格納する場合の出力位置に対応する。ＸＴＳ暗号化／復号の結果の出力先の位置関係は入力元のそれと同じである。すなわち、入力ブロック先頭位置がＸ、出力ブロック先頭位置がＹの時、入力ブロック先頭からＰ（Ｐは０以上）だけ離れたＸ＋Ｐの位置にあるデータをＸＴＳ暗号化／復号した結果はＹ＋Ｐの位置に出力される。
【０１２７】
１つのＧｒｏｕｐをＸＴＳ暗号化／復号した結果はＺｍｍ４～７の４つのレジスタに保持されており、４つの６４バイトのデータ（１６バイト×４）はこの規則に従った位置に格納される。例えば、破線９１０で囲った６４バイトのデータは、図８の破線８１０で囲った６４バイトのデータのＸＴＳ暗号化／復号の結果であり、Ｚｍｍ４レジスタに格納される。なお、これら格納元のＺｍｍレジスタの番号は一例であって、他のレジスタ番号でもよい。
【０１２８】
ＤＲＡＭ１３４に格納されるユーザデータ又は暗号化ユーザデータの各データブロックの終端にはそれぞれ８バイトのＤＩＦが格納される。４つのブロックのＤＩＦの生成は並列で実行されため、１つのＺｍｍレジスタからＤＩＦが読み出される。図９では、Ｚｍｍ９からＤＩＦが読み出されている。
【０１２９】
Ｚｍｍ９レジスタに格納されるＤＩＦは８バイトの間隔をあけて格納される。なお、ＤＩＦの格納元のＺｍｍレジスタの番号は一例であって、他のレジスタ番号でもよい。
【０１３０】
ＸＴＳ暗号化／復号処理の結果が同時に出力される４つのデータブロックのアドレスは、図８と同様に昇順で連続しており、４Ｎ、４Ｎ＋１、４Ｎ＋２、４Ｎ＋３（Ｎは整数）である。
【０１３１】
以上、図７～９を用いて説明した、４つデータブロックのＸＴＳ暗号化／復号の並列処理は、図４のステップＳ４０５及び図５のステップＳ５０３において適用される。
【０１３２】
図１０Ａ、図１０Ｂ、及び図１０Ｃは、実施例１のストレージシステム１２０におけるＣＲＣの算出方法を説明する図である。
【０１３３】
まず、図１０Ａを参照しながら、ＣＲＣ計算の理論を説明する。一般に、バイナリ多項

式Ｍ（ｘ）に対応する任意長のバイナリデータＭのＣＲＣは式（１）で定義される。

【０１３４】

【数１】

$$\mathrm{CRC}\big(\mathrm{M(x)}\big) = \mathrm{x}^{\deg(\mathrm{P(x)})} \cdot \mathrm{M(x)} \bmod \mathrm{P(x)} \ldots (1)$$

【０１３５】

ここで、ｄｅｇは多項式の次数を表し、Ｐ（ｘ）はＣＲＣアルゴリズムを定義する多項式を表し、記号「・」はキャリーレス乗算を表す。１６ビットＣＲＣアルゴリズムの場合、Ｐ（ｘ）は次数１６の多項式である。ＣＲＣは、ガロア体ＧＦ（２）上で定義された大きな次数の多項式Ｍ（ｘ）をＣＲＣ多項式Ｐ（ｘ）で割った時の剰余多項式として計算できる。

【０１３６】

図１０Ａは、ストレージシステム１２０において典型的な５１２バイト（４０９６ビット）のブロックから１６ビットのＣＲＣを算出する方法を示している。キャリーレス乗算を効率的に実行できるＣＰＵ命令があれば、バイナリ多項式の除算は効率的に実行できる。

【０１３７】

図１０Ｂ及び図１０Ｃは、最大で６４ビット×６４ビットのキャリーレス乗算が実行可能なＰＣＬＭＵＬＱＤＱ命令を用いたＣＲＣ計算の方法を示している。ＰＣＬＭＵＬＱＤＱ命令は、Ｉｎｔｅｌ社が２０１０年頃以降に製品化しているマイクロプロセッサに搭載されている。

【０１３８】

図１０Ｂは、５１２バイト（４０９６ビット）のバイナリデータのＣＲＣ計算を、１２８ビットのバイナリデータのＣＲＣ計算に短縮する方法を示している。１回の短縮処理によってバイナリデータのビット数は１２８だけ減る。ｋ回の短縮処理を施した後のバイナリ多項式Ｍｋ（ｘ）と、さらに１回の短縮処理を施して得られる多項式Ｍｋ＋１（ｘ）との関係は式（２）で与えられる。

【０１３９】

【数２】

$$\left.\begin{aligned} & M_k(x)\ mod\ P(x) \\ & \quad = \left\{H(x) \cdot \left[x^{(T+64)}\ mod\ P(x)\right]\right\} + \{L(x) \cdot [x^T\ mod\ P(x)]\} + G(x)\ mod\ P(x) \\ & \quad = M_{k+1}(x)\ mod\ P(x) \end{aligned}\right\} \ldots (2)$$

【０１４０】

ここで、Ｍ０（ｘ）＝Ｍ（ｘ）である。Ｈ（ｘ）はＭｋ（ｘ）が示すバイナリデータの上位６４ビットからなる多項式を表し、Ｌ（ｘ）はＭｋ（ｘ）が示すバイナリデータの上位６４ビットに続く６４ビットからなる多項式を表す。Ｇｋ（ｘ）はＭｋ（ｘ）が示すバイナリデータの上位１２８ビットを除いた残りデータからなる多項式を表す。Ｔは当該残りデータのビット数を表す。記号「＋」はビット単位の排他的論理和を表す。

【０１４１】

ＰＣＬＭＵＬＱＤＱ命令によって、Ｈ（ｘ）とＬ（ｘ）が示す６４ビット値にそれぞれ定数を乗じて２つの１２８ビット値を求め、それらとＧｋ（ｘ）が示す残りデータとの排他的論理和を算出することによって、ＣＲＣ計算の対象となるバイナリデータのビット数を１２８だけ減らすことができる。この短縮処理を３１回行うことで、５１２バイト（４０９６ビット）のバイナリデータのＣＲＣ計算は、Ｍ３１（ｘ）が示す１２８ビットのバイナリデータのＣＲＣ計算に帰着する。

【０１４２】

図１０Ｃは、Ｍ３１（ｘ）が示す１２８ビットのバイナリデータのＣＲＣ計算（図１０Ｂの最終計算）を行う方法を示している。

【０１４３】
まず、ＣＰＵ１３３は、ＰＣＬＭＵＬＱＤＱ命令によって、上位６４ビット値に定数を乗じて８０ビット値（１０３１）を求め、それと下位６４ビット値との排他的論理和を算出することによって、ＣＲＣ計算の対象データのビット数を８０に減らす。次に、ＣＰＵ１３３は、ＰＣＬＭＵＬＱＤＱ命令によって、その上位３２ビット値に定数を乗じて４８ビット値（１０３２）を求め、それと８０ビット値（１０３１）の下位４８ビットとの排他的論理和を算出することによって、ＣＲＣ計算の対象データのビット数を４８に減らす。その結果をＲとする。最後に、ＣＰＵ１３３は、Ｂａｒｒｅｔｔ　Ｒｅｄｕｃｔｉｏｎと呼ばれるアルゴリズムに基づいて、４８ビットのＲから１６ビットのＣＲＣを算出する。具体的には、ＣＰＵ１３３が、式（３）、（４）、（５）、（６）を順で演算を行う。

【０１４４】
【数３】
$$\mu = Floor\left(x^{48}/P(x)\right) \ldots (3)$$

【０１４５】
【数４】
$$T1(x) = Floor(R(x)/x^{16}) \cdot \mu \ldots (4)$$

【０１４６】
【数５】
$$T2(x) = Floor(T1(x)/x^{32}) \cdot P(x) \ldots (5)$$

【０１４７】
【数６】
$$C(x) = R(x) + T2(x)\ mod\ x^{16} \ldots (6)$$

【０１４８】
ここで、Ｒ（ｘ）はＲからなる多項式を表し、Ｆｌｏｏｒは多項式からｘの０次以上の項を残す操作を表し、記号「＋」はビット単位の排他的論理和を表す。Ｃ（ｘ）が示す１６ビット値が求めるＣＲＣである。このように、Ｂａｒｒｅｔｔ　Ｒｅｄｕｃｔｉｏｎでも、ＰＣＬＭＵＬＱＤＱ命令が適用可能なキャリーレス乗算が２回使用される。

【０１４９】
ＣＰＵ１３３は、ＰＣＬＭＵＬＱＤＱ命令をサポートしており、さらに、それをＶｅｃｔｏｒ化したＶＰＣＬＭＵＬＱＤＱ命令もサポートしている。

【０１５０】
ＣＰＵ１３３は、ＶＰＣＬＭＵＬＱＤＱ命令を用いて４つのデータブロックのＣＲＣ計算を並列で実行する。具体的には、４つの５１２バイトのデータブロックのＣＲＣ計算を並列で実行する。

【０１５１】
図１１Ａ及び図１１Ｂは、実施例１のストレージシステム１２０におけるＣＲＣ計算の並列方法を示す図である。この方法は、図４のステップＳ４０５とステップＳ４０６、図５のステップＳ５０３とステップＳ５０４におけるＣＲＣ計算に適用される。

【０１５２】
図１１Ａは、ＣＰＵ１３３による４つのデータブロックのＣＲＣ計算の並列処理における、バイナリデータの短縮処理（図１０Ｂの処理全体に相当）の方法を示す。

【０１５３】
Ｚｍｍレジスタ１１１１は、図７～９を用いて説明した４つのデータブロックのＸＴＳ暗号化／復号の並列処理において、暗号化／復号の入力データ又は暗号化／復号の出力デ

ータの構成要素である、４つのデータブロックに対応する１６バイトのデータを保持するレジスタである。具体的には、ＸＴＳ暗号化／復号の入力データの場合はＺｍｍ０レジスタ、Ｚｍｍ１レジスタ、Ｚｍｍ２レジスタ、又はＺｍｍ３レジスタを示し、ＸＴＳ暗号化／復号の出力データの場合はＺｍｍ４レジスタ、Ｚｍｍ５レジスタ、Ｚｍｍ６レジスタ、又はＺｍｍ７レジスタを示す。

【０１５４】

本発明は、Ｚｍｍレジスタ１１１１がＸＴＳ暗号化／復号のために保持しているデータを用いて、ＣＲＣ計算の対象データの短縮処理を並列で行う。Ｚｍｍレジスタ１１１１の１６バイトのデータは、図１０ＢのＨ（ｘ）とＬ（ｘ）が示す２つの６４ビット値を並べた１２８ビット値に相当する。

【０１５５】

Ｚｍｍレジスタ１１１２は、短縮処理の途中、又は短縮処理後に得られる、４つのデータブロックの１２８ビットのバイナリデータを保持するレジスタである。具体的には、ＸＴＳ暗号化／復号の入力データのＣＲＣ計算の場合はＺｍｍ１０レジスタを示し、ＸＴＳ暗号化／復号の出力データのＣＲＣ計算の場合はＺｍｍ９レジスタを示す。

【０１５６】

まず、ＣＰＵ１３３は、（ＰＰ１）ＶＰＣＬＭＵＬＱＤＱ命令を実行することによって、Ｚｍｍレジスタ１１１１の１６バイトのデータの上位６４ビットに同じ定数を乗じる。演算結果として１２８ビットのデータが４つ出力される。次に、ＣＰＵ１３３は、（ＰＰ２）ＶＰＣＬＭＵＬＱＤＱ命令を実行することによって、Ｚｍｍレジスタの各１６バイトのデータの下位６４ビットに同じ定数を乗じる。演算結果として１２８ビットのデーが４つ出力される。ＣＰＵ１３３は、（ＰＰ３）これら２つの結果とＺｍｍレジスタ１１１２との排他的論理和を算出し、その結果を再びＺｍｍレジスタ１１１２に格納する。（ＰＰ１）～（ＰＰ３）の処理が１回の短縮処理に対応する。

【０１５７】

ＣＰＵ１３３は、４つのデータブロックのＸＴＳ暗号化／復号の並列処理において、Ｚｍｍレジスタ１１１１に次の１６バイトのデータが格納された場合、（ＰＰ１）～（ＰＰ３）で示す短縮処理を繰り返す。ＣＰＵ１３３は、短縮処理を３１回実行し、Ｚｍｍレジスタ１１１１にブロック最後の１６バイトのデータ（つまり、Ｇｒｏｕｐ７の４つ目）が格納された場合、Ｚｍｍレジスタ１１１１の値とＺｍｍレジスタ１１１２との排他的論理和を算出し、その結果を再びＺｍｍレジスタ１１１２に格納する。

【０１５８】

図１１Ｂは、ＣＰＵ１３３による４ブロックデータのＣＲＣ計算の並列処理における、最終計算（図１０Ｂの最終処理）の方法を示す。

【０１５９】

Ｚｍｍレジスタ１１１２は、４つのデータブロックの、最終計算に用いる１２８ビットのバイナリデータを保持している。本発明はＣＲＣ計算の最終計算を、このＺｍｍレジスタ１１１２に保持したデータを利用して並列で行う。ここで並列の最終計算１１２１は、図１０Ｂの最終計算に用いるＣＰＵ命令を、対応するＶｅｃｔｏｒ命令に置き換えて実装する。すなわち、ＸＯＲ命令はＶｅｃｔｏｒ　ＸＯＲ命令、ＰＣＬＭＵＬＱＤＱ命令はＶＰＣＬＭＵＬＱＤＱ命令に置き換える。Ｂａｒｒｅｔｔ　Ｒｅｄｕｃｔｉｏｎを構成する各命令も同様に、それに対応するＶｅｃｔｏｒ命令に置き換える。最終的なＣＲＣ計算の結果は、Ｚｍｍレジスタ１１１２に格納する。具体的には、ＸＴＳ暗号化／復号の入力データのＣＲＣ計算の場合はＺｍｍ１０レジスタを示し、ＸＴＳ暗号化／復号の出力データのＣＲＣ計算の場合はＺｍｍ９レジスタを示す。４つのＣＲＣ計算の結果が格納される位置は、Ｚｍｍレジスタ１１１２の各１６バイトの先頭１６ビットである。なお、ＣＲＣ計算の完了時点で、各１６バイトの残り部分（１１２ビット）は未使用である。

【０１６０】

図１２Ａ及び図１２Ｂは、実施例１のストレージシステム１２０における４つのデータブロックのＤＩＦの並列的な生成方法を説明する図である。

【０１６１】
図１２Ａは、図１１に示す方法により、ＸＴＳ暗号化／復号の出力データから並列的に計算された４つのＣＲＣを保持するＺｍｍ９レジスタに対して、ＣＰＵ１３３が各ＣＲＣの後続部に１バイト（８ビット）のＷＳＣを並列的に付加する方法を示す。
【０１６２】
付加するＷＳＣは、図３を用いて説明したように、先頭ブロックのＨＥＡＤ３０１は１、非先頭ブロックのＨＥＡＤ３０１は０とし、終端ブロックのＴＡＩＬ３０２は１、非終端ブロックのＴＡＩＬ３０２は０とし、すべてのデータブロックのＳＱＮは共通の６ビット値とする。この設定規則に基づくＷＳＣを含む５１２ビットパタン（ただし、ＡＴＡＧ（１）２３３、２４３の各８ビットには００ｈが設定される。）を保持するＺｍｍ１１レジスタを用意し、これをＺｍｍ９レジスタの各ＣＲＣの後の８ビット部分のみに重ね書きする。重ね書きには、例えば、ビットマスク付きＭＯＶ命令をＶｅｃｔｏｒ化した命令を使用する。その結果、Ｚｍｍ９レジスタはＣＲＣとＷＳＣの組を４つ格納される。
【０１６３】
例えば、ライト／リードするユーザデータのサイズが１２８Ｋバイト（２５６ブロック）の場合、ＷＳＣの付加の並列処理は６４回行う必要がある。１回目の付加で、先頭ブロックのＤＩＦのＷＳＣのＨＥＡＤ３０１には１が設定され、６４回目の付加で、終端ブロックのＤＩＦのＷＳＣのＴＡＩＬ３０２には１が設定される。その他のＨＥＡＤ３０１とＴＡＩＬ３０２には０が設定される。
【０１６４】
図１２Ｂは、ＣＰＵ１３３がＺｍｍ９レジスタに対して、各ＡＴＡＧ（１）２３３、２４３の後続部に４バイトのＲＴＡＧ２３４、２４４を並列的に付加する方法を示す。
【０１６５】
付加するＲＴＡＧ２３４、２４４は、昇順に並んだ４つのデータブロックのアドレスである。この設定規則に基づくＲＴＡＧ２３４、２４４を含む５１２ビットパタンを格納するＺｍｍ１２レジスタを用意し、これをＺｍｍ９レジスタの各ＡＴＡＧ（１）２３３、２３４の後の４バイト部分のみに重ね書きする。重ね書きには、例えば、ビットマスク付きＭＯＶ命令をＶｅｃｔｏｒ化した命令を使用する。その結果、Ｚｍｍ９レジスタにはＤＩＦ（つまり、ＣＲＣ、ＷＳＣ、ＲＴＡＧの組）が４つ格納される。これは図９において、付加する４つのＤＩＦを保持するＺｍｍ９レジスタに相当する。このＺｍｍ９レジスタからＤＲＡＭ１３４への格納命令によって、４つの出力データブロックの各終端にＤＩＦが付加される。
【０１６６】
以上に述べた、並列的なＤＩＦ生成の方法は、図４のＳ４１１及び図５のＳ５０９に適用する。
【０１６７】
図１３Ａ、図１３Ｂ、及び図１３Ｃは、実施例１のストレージシステム１２０における４つのデータブロックのＤＩＦの並列的な検査方法を説明する図である。
【０１６８】
図１３Ａ、図１３Ｂ、及び図１３Ｃにおいて、Ｚｍｍ８レジスタは、図８でＸＴＳ暗号化／復号の入力データの各データブロックに後続するＤＩＦ（合わせて４つ）を格納するＺｍｍレジスタである。
【０１６９】
図１３Ａは、Ｚｍｍ８レジスタに対を用いた、４つのＣＲＣの並列的な検査方法を示す。
【０１７０】
図１１に示す方法により、Ｚｍｍ１０レジスタは、ＸＴＳ暗号化／復号の入力データから並列的に計算された４つのＣＲＣを保持する。これらの各ＣＲＣが、Ｚｍｍ８レジスタに格納される各ＣＲＣと一致するか否かを並列に検査する。当該検査には、例えば、ビットマスク付きＣＯＭＰＡＲＥ命令をＶｅｃｔｏｒ化した命令が使用される。その結果、４つの比較結果がすべて一致となればＣＲＣ検査は成功である。この検査方法は、図４のス

テップＳ４０８及び図５のステップＳ５０６に適用する。

【０１７１】

図１３Ｂは、Ｚｍｍ８レジスタを用いた４つのＷＳＣの並列的な検査方法を示す。

【０１７２】

Ｚｍｍ１３レジスタは、下位４８バイトにはＺｍｍ８レジスタの内容を１６バイト右方向にシフトしたもの、上位１６バイトには前回検査した第４のブロック（つまり、アドレスが４Ｎ＋３）のＤＩＦを保持する。これらの各ＷＳＣのＳＱＮが、Ｚｍｍ８レジスタのＷＳＣのＳＱＮと一致するか否かを並列に検査する。当該検査には、例えば、ビットマスク付きＣＯＭＰＡＲＥ命令をＶｅｃｔｏｒ化した命令を使用される。

【０１７３】

ただし、図３Ｃの表３２０に示す（Ｒ３）、（Ｒ４）の条件に合わないＷＳＣのＳＱＮの検査は行われない。この場合、例えば、ＣＯＭＰＡＲＥ命令のマスクパラメタを部分的にゼロにする。

【０１７４】

ＷＳＣの検査後に、Ｚｍｍ８レジスタ上の、第４のブロックのＤＩＦを含む下位１６バイトを次回のＷＳＣ検査のためにレジスタに退避する。その結果、有効なＳＱＮ比較（最大で４つ）の結果がすべて一致すればＷＳＣ検査は成功である。このように、Ｚｍｍ１３レジスタに、１ブロック分シフトしたＷＳＣを保持することにより、１回のＶｅｃｔｏｒ ＣＯＭＰＡＲＥ命令を実行することによって、自ＷＳＣと前ＷＳＣとのＳＱＮ比較を４つ同時に実行できる。この検査方法は、図５のステップＳ５０７に適用する。

【０１７５】

例えば、ライト／リードするユーザデータのサイズが１２８Ｋバイト（２５６ブロック）の場合、４つの暗号化データブロックのＷＳＣの並列検査は６４回行う必要がある。Ｚｍｍ８レジスタ上の、第４のブロックのＤＩＦを含む下位１６バイトは、１回目～６３回目の検査で退避され、２回目～６４回目の検査でＺｍｍ１３レジスタの上位１６バイトに転送される。１回目の検査でＺｍｍ１３レジスタの上位１６バイトには何も転送しないが、最初のＷＳＣは検査されないため、この場所には無効な値が含まれていてよい。

【０１７６】

図１３Ｃは、Ｚｍｍ８レジスタを用いた４つのＲＴＡＧの並列的な検査方法を示す。

【０１７７】

Ｚｍｍ１４レジスタは、昇順に並んだ４つのアドレスを保持する。これらの各アドレスが、Ｚｍｍ８レジスタ上の対応するＲＴＡＧと一致するかを並列に検査する。当該検査には、例えば、ビットマスク付きＣＯＭＰＡＲＥ命令をＶｅｃｔｏｒ化した命令が使用される。その結果、４つのアドレスがすべて一致となればＲＴＡＧ検査は成功である。この検査方法は、図４のステップＳ４０９及び図５のステップＳ５０８に適用する。

【０１７８】

図１２と図１３において、ＤＩＦの検査／生成に用いられるＺｍｍレジスタの番号（Ｚｍｍ１１レジスタ、Ｚｍｍ１２レジスタ、Ｚｍｍ１３レジスタ、Ｚｍｍ１４レジスタ）は一例であって、他のレジスタ番号でもよい。

【０１７９】

以上、本発明を適用したストレージシステム１２０のＣＰＵ１３３は、ライト／リード処理において、５１２ビットのＺｍｍレジスタとＶｅｃｔｏｒ命令とを用い、ＸＴＳ暗号化／復号、ＣＲＣ計算、ＤＩＦ生成、及びＤＩＦ検査を、並列で実行できる。また、上に述べたＺｍｍレジスタの使い方に従えば、ＺｍｍレジスタとＤＲＡＭ１３４との間のロード／格納の回数は必要最小限に抑えられ、データ演算サイクル以外のオーバヘッドサイクルが最小化される。

【０１８０】

なお、ライト／リードするユーザデータのサイズが、２０４８Ｕバイト（Ｕは整数）の場合、４ブロック並列のライト／リード処理をＵ回行う必要があるが、ステップＳ４０１のデータ受信、ステップＳ４１２のデータライト、ステップＳ５０１のデータリード、ス

テップＳ５１２のデータ送信は、それぞれＵ回に分けて実施せず、１回にまとめて実施してもよい。

【０１８１】

なお、本発明は、４つの処理データを保持する５１２ビットのＺｍｍレジスタの代わりに、２つの処理データを保持する２５６ビットのＹｍｍレジスタを用いてもよい。その場合、上に述べた方法と同様に、Ｙｍｍレジスタに対するＶｅｃｔｏｒ命令を適用して、ＸＴＳ暗号化／復号、ＣＲＣ計算、ＤＩＦ生成、ＤＩＦ検査を、すべて２ブロック並列で処理することができる。

【０１８２】

本発明は、ホスト計算機１１０から受信したユーザデータを暗号化してストレージデバイスに格納するライト処理、及びストレージデバイスに格納された暗号化ユーザデータを復号してホスト計算機１１０に送信するリード処理を、従来よりも高速に実行することができるという効果を奏する。

【０１８３】

以上、本発明の一実施例を説明したが、これは本発明の説明のための例示であって、本発明の範囲をこの実施例にのみ限定する趣旨ではない。すなわち、本発明は、他の種々の形態でも実施する事が可能である。

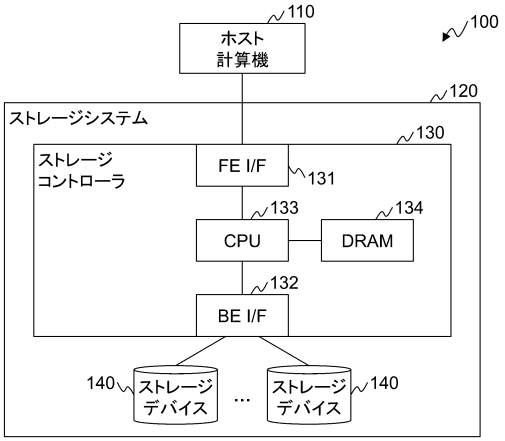
【符号の説明】

【０１８４】

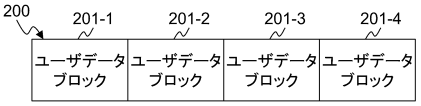
１１０　ホスト計算機
１２０　ストレージシステム
１３０　ストレージコントローラ
１３１　ＦＥ　Ｉ／Ｆ
１３２　ＢＥ　Ｉ／Ｆ
１３３　ＣＰＵ
１３４　ＤＲＡＭ
１４０　ストレージデバイス

【図面】

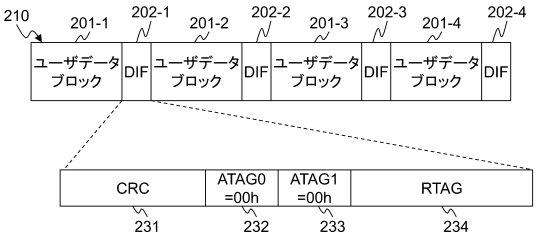
【図１】



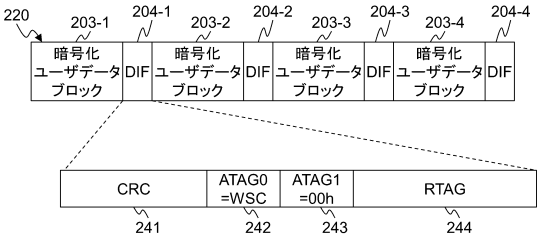
【図２Ａ】



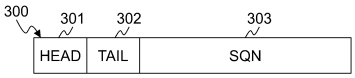
【図２Ｂ】



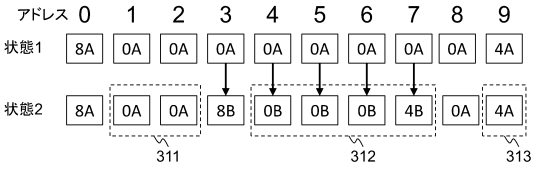
【図２Ｃ】



【図３Ａ】



【図３Ｂ】

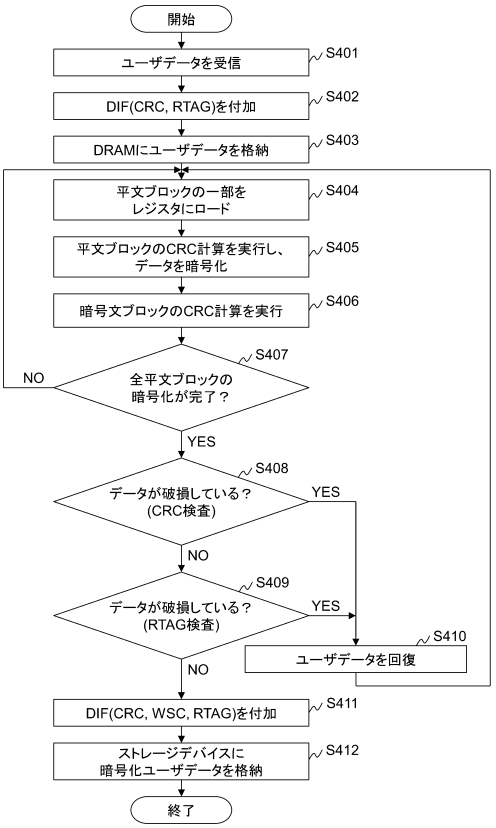


【図３Ｃ】

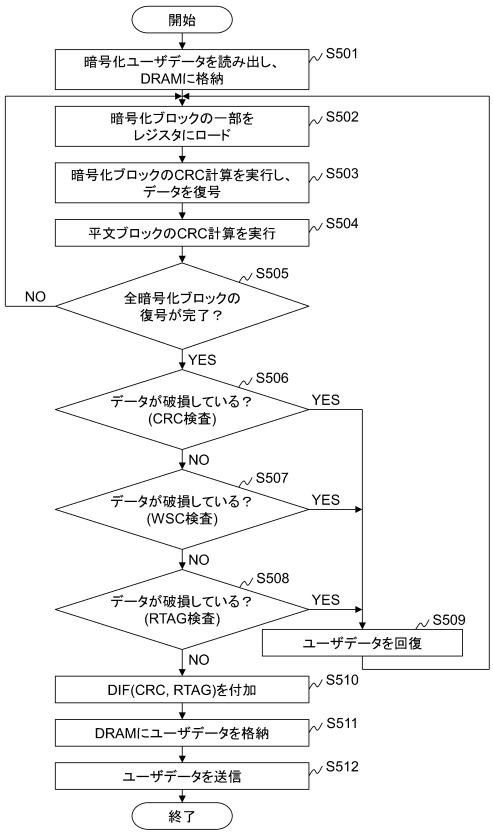
320

#	自WSC	前WSC	検査
(R1)	H/T=1/0 or 1/1	ー	なし
(R2)	H/T=0/0 or 0/1	H/T=0/1	なし
(R3)		H/T=0/0	前WSCとSQNが一致？
(R4)		H/T=1/0	前WSCとSQNが一致？
(R5)		H/T=1/1	なし

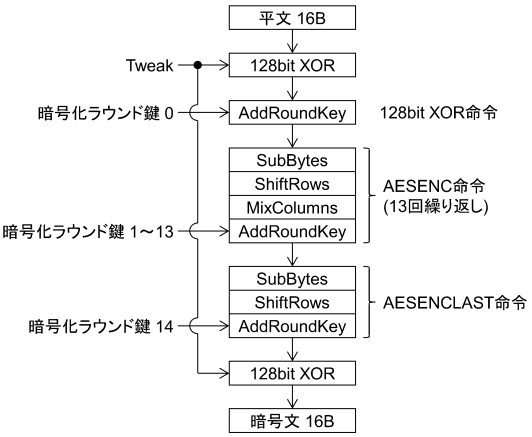
【図４】



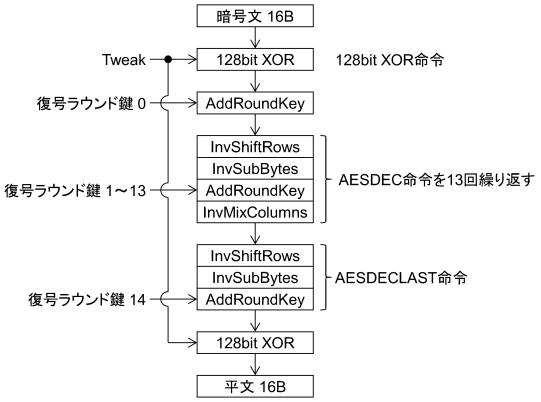
【図５】



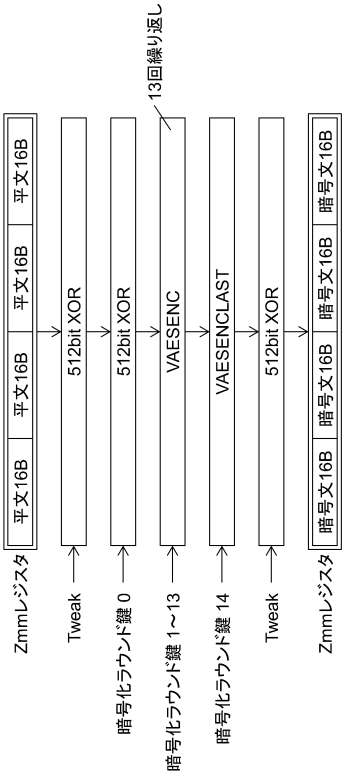
【図６Ａ】



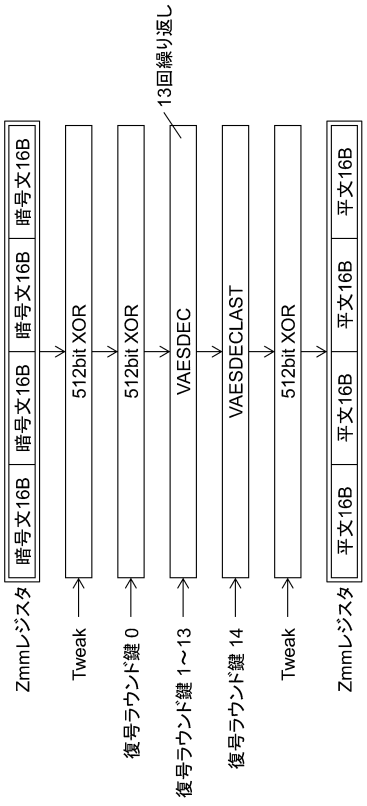
【図６Ｂ】



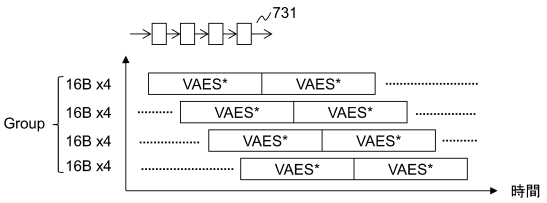
【図７Ａ】



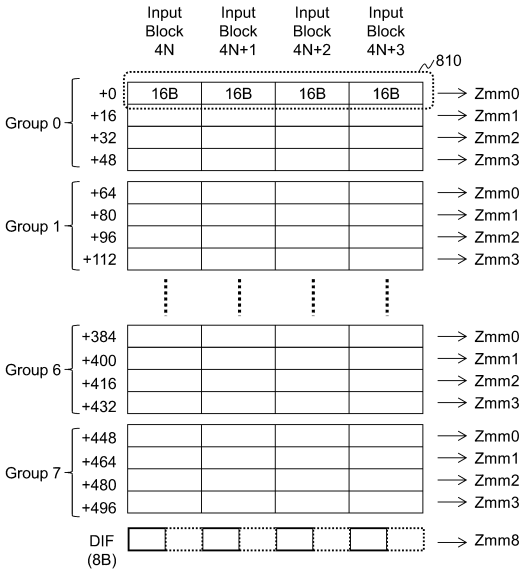
【図７Ｂ】



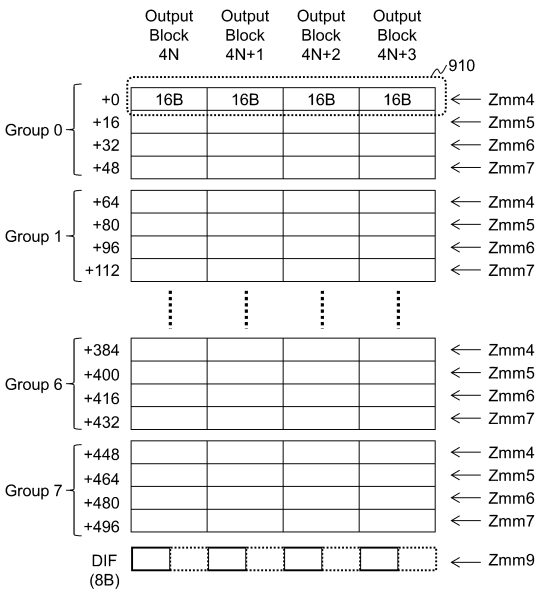
【図７Ｃ】



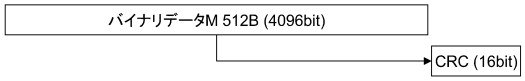
【図８】



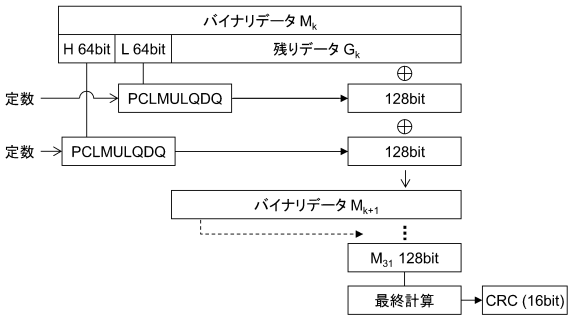
【図９】



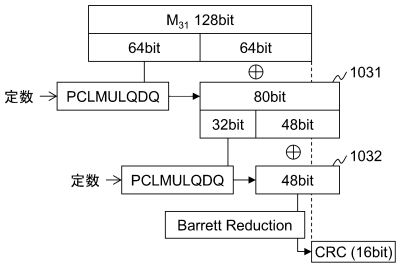
【図１０Ａ】



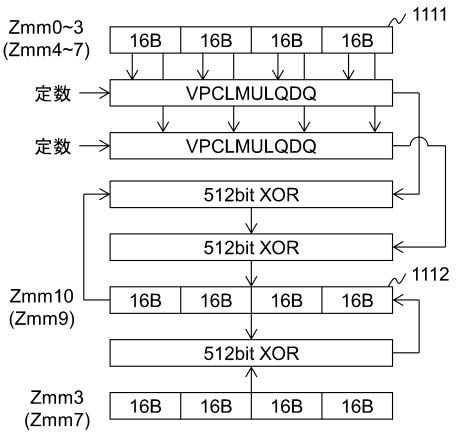
【図１０Ｂ】



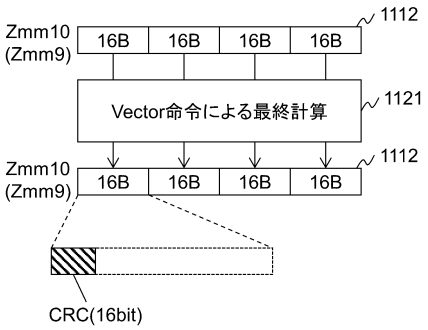
【図１０Ｃ】



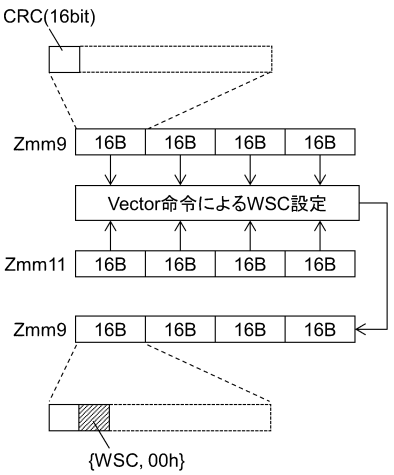
【図１１Ａ】



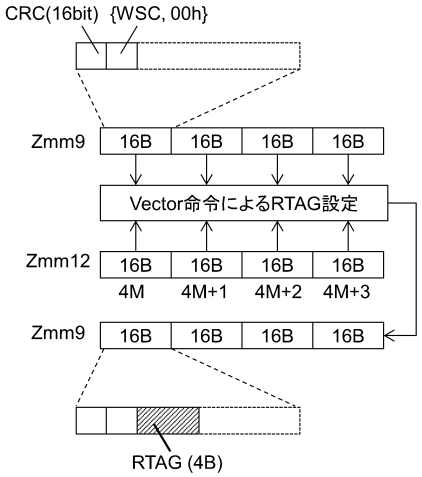
【図１１Ｂ】



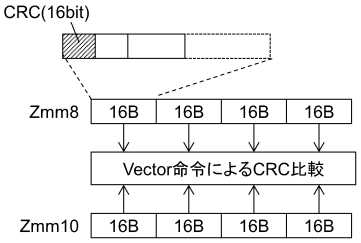
【図１２Ａ】



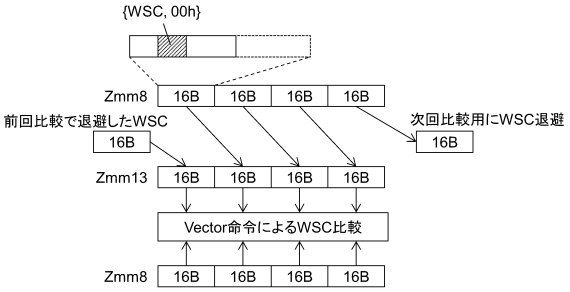
【図１２Ｂ】



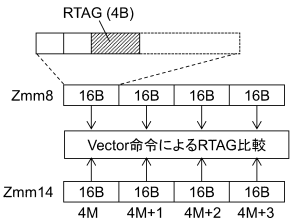
【図１３Ａ】



【図１３Ｂ】



【図１３Ｃ】



フロントページの続き

(51)国際特許分類

FI		
G11B	20/18	572B
G11B	20/18	572F
G11B	20/18	532B

(56)参考文献 特開２０２１－１３２３４２（ＪＰ，Ａ）
米国特許出願公開第２０２１／０１５００７４（ＵＳ，Ａ１）
岡田 尚也，"ストレージシステム適用に向けたプロセッサPCIe性能の評価"，ＦＩＴ２０１６ 第１５回情報科学技術フォーラム 講演論文集 第１分冊，2016年
Keith Holt，"End-to-End Data Protection Justification"，T10/03-224r0，T10 Technical Committee，2003年06月01日，［2024年5月17日検索］，インターネット ＜URL：https://www.t10.org/ftp/t10/document.03/03-224r0.pdf＞

(58)調査した分野 (Int.Cl.，ＤＢ名)

Ｇ０６Ｆ ２１／６０
Ｇ０６Ｆ １１／１０
Ｇ１１Ｂ ２０／１０
Ｇ１１Ｂ ２０／１８