(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-114682
(P2010-114682A)
(43)公開日　平成22年5月20日(2010.5.20)

(51)Int.Cl.			F I			テーマコード（参考）
H04L	9/08	(2006.01)	H04L	9/00	601B	5B017
G06F	21/24	(2006.01)	H04L	9/00	601E	5J104
			G06F	12/14	540P	

審査請求 未請求　請求項の数 9　ＯＬ　（全 17 頁）

(21)出願番号　特願2008-285787 (P2008-285787)
(22)出願日　平成20年11月6日 (2008.11.6)

特許法第３０条第１項適用申請有り　平成２０年９月１４日　社団法人情報処理学会発行の「情報処理学会研究報告／ＩＰＳＪ　ＳＩＧ　Ｔｅｃｈｎｉｃａｌ　Ｒｅｐｏｒｔｓ（情処研報　Ｖｏｌ．２００８，Ｎｏ．８８）」に発表

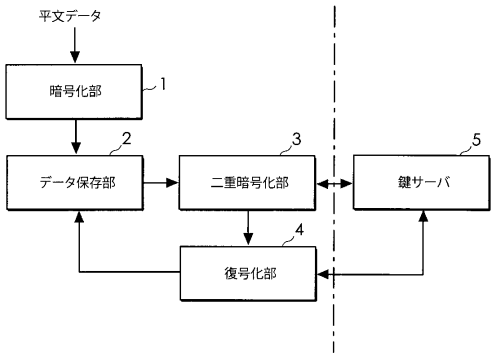
特許法第３０条第１項適用申請有り　平成２０年１０月８日　社団法人情報処理学会コンピュータセキュリティ研究会主催の「コンピュータセキュリティシンポジウム２００８（Ｃｏｍｐｕｔｅｒ　Ｓｅｃｕｒｉｔｙ　Ｓｙｍｐｏｓｉｕｍ　２００８／ＣＳＳ２００８）」において文書をもって発表

(71)出願人　503360115
独立行政法人科学技術振興機構
埼玉県川口市本町四丁目１番８号
(74)代理人　100091443
弁理士　西浦　▲嗣▼晴
(72)発明者　横田　治夫
東京都江東区越中島１－３－１６－１００５
(72)発明者　高山　一樹
千葉県千葉市若葉区東寺山町４５９

Ｆターム(参考)　5B017 AA03 BA07 CA16
5J104 AA16 EA01 EA04 EA15 EA16 EA17 EA19 JA03 JA21 MA05 NA02 NA27 NA37 PA14

(54)【発明の名称】ストレージノード用再暗号化システム及び方法

(57)【要約】
【課題】　ストレージノード上での再暗号化の際に、平文データを生成することながないストレージノード用再暗号化システムを提供する。
【解決手段】　暗号化データを新しい鍵で再暗号化する場合には、二重暗号化部３が、データ保存部２に記憶された暗号化データK_1（Ｆ）を第２の暗号鍵K_2により二重に暗号化して二重暗号化データK_1K_2（Ｆ）を作成する。そして復号化部４は、二重暗号化データK_1K_2（Ｆ）を第１の暗号鍵K_1を用いて復号して再暗号化データK_2（Ｆ）を作成し、データ保存部２に保存されている暗号化データK_1（Ｆ）を再暗号化データK_2（Ｆ）で置き換える。第１の暗号鍵K_1及び第２の暗号鍵K_2は、ネットワークＮＷを介してつながる鍵サーバ５に登録されている。
【選択図】　図３



【特許請求の範囲】
【請求項１】
平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、前記暗号化データを読み出し可能に記憶するデータ保存部とを備えたストレージノードに設けられて、前記暗号化データを第２の暗号鍵で復号できる再暗号化データに変換するストレージノード用再暗号化システムであって、
前記データ保存部に記憶された前記暗号化データを第２の暗号鍵により二重に暗号化して二重暗号化データを作成する二重暗号化部と、
前記二重暗号化データを前記第１の暗号鍵を用いて復号して前記再暗号化データを作成し、前記暗号化データを前記再暗号化データで置き換える復号化部とを備えてなるストレージノード用再暗号化システム。
【請求項２】
前記第１の暗号鍵及び第２の暗号鍵はネットワークを介してつながる鍵サーバに公開鍵により暗号化されて登録されており、
前記二重暗号化部は二重の暗号化に用いた前記第２の暗号鍵を前記鍵サーバに登録し、前記復号化部は前記鍵サーバから取得した前記第１の暗号鍵を秘密鍵を用いて復号する請求項１に記載のストレージノード用再暗号化システム。
【請求項３】
前記ストレージノードは、前記第１の暗号鍵及び前記第２の暗号鍵を第１の公開鍵により暗号化して格納するロックボックスを備えており、
前記第１の公開鍵と組となる第１の秘密鍵が第２の公開鍵により暗号化されて鍵サーバに格納されており、
前記二重暗号化部は、二重の暗号化に用いた前記第２の暗号鍵を前記第１の公開鍵により暗号化して前記ロックボックスに格納し、
前記復号化部は前記第２の公開鍵により暗号化された前記第１の秘密鍵を前記鍵サーバから取得して、前記第２の公開鍵と組になる第２の秘密鍵を用いて暗号化された前記第１の秘密鍵を復号し、前記第１の公開鍵により暗号化された前記第１の暗号鍵を前記ロックボックスから取得し、暗号化された前記第１の暗号鍵を前記第１の秘密鍵を用いて復号し、復号した前記第１の暗号鍵を用いて前記二重暗号化データを復号して前記再暗号化データを作成する請求項１に記載のストレージノード用再暗号化システム。
【請求項４】
前記第１の暗号鍵及び前記第２の暗号鍵を格納するロックボックスを備え、
前記二重暗号化部及び前記復号化部がセキュリティチップによって構成されている請求項１に記載のストレージノード用再暗号化システム。
【請求項５】
前記二重暗号化部及び前記復号化部は、ブロック単位でデータの暗号化及び復号化を行う請求項１乃至４のいずれか１項に記載のストレージノード用再暗号化システム。
【請求項６】
前記暗号化部、前記二重暗号化部及び前記復号化部は、それぞれ暗号化と復号が可逆な暗号化モードにより暗号化または復号を実行する請求項１乃至４のいずれか１項に記載のストレージノード用再暗号化システム。
【請求項７】
平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、前記暗号化データを読み出し可能に記憶するデータ保存部とを備えたストレージノードにおいて、前記暗号化データを第２の暗号鍵で復号化できる再暗号化データに変換することを内部演算装置により実行するストレージノード用再暗号化方法であって、
前記データ保存部に記憶された前記暗号化データを第２の暗号鍵により二重に暗号化して二重暗号化データを作成し、
前記二重暗号化データを前記第１の暗号鍵を用いて復号して前記再暗号化データを作成し、

前記暗号化データを前記再暗号化データで置き換えることを特徴とするストレージノード用再暗号化方法。

【請求項８】

平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、前記暗号化データを読み出し可能に記憶するデータ保存部とを備えた複数のストレージノードが、ネットワークに並列接続されて構成され、

前記ストレージノードが、前記暗号化データを第２の暗号鍵で復号できる再暗号化データに変換するストレージノード用再暗号化システムを備えているネットワークストレージにおいて、

前記ストレージノード用再暗号化システムが、前記データ保存部に記憶された前記暗号化データを前記第２の暗号鍵により二重に暗号化して二重暗号化データを作成する二重暗号化部と、

前記二重暗号化データを前記第１の暗号鍵を用いて復号して前記再暗号化データを作成し、前記暗号化データを前記再暗号化データで置き換える復号化部とを備えており、

前記ネットワークに接続された他の前記ストレージノードに、第１の公開鍵により暗号化された前記第１の暗号鍵及び第２の公開鍵により暗号化された前記第２の暗号鍵を格納するロックボックスが設けられており、

前記二重暗号化部は、前記第２の公開鍵により暗号化された前記第２の暗号鍵を前記ロックボックスに格納し、

前記復号化部は、前記ロックボックスから前記第１の公開鍵により暗号化された前記第１の暗号鍵を前記第１の公開鍵と組になる第１の秘密鍵を用いて復号し、復号した前記第１の暗号鍵を用いて前記二重暗号化データを復号して前記再暗号化データを作成するネットワークストレージ。

【請求項９】

前記暗号化部、前記二重暗号化部及び前記復号化部は、それぞれ暗号化と復号が可逆な暗号化モードにより暗号化または復号を実行する請求項８に記載のネットワークストレージ。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ストレージノードにおいて、途中で平文データに変換することなく暗号化データを再暗号化データに変換することができるストレージノード用再暗号化システム及び方法に関するものである。

【背景技術】

【０００２】

非特許文献１及び２に示されるように、情報セキュリティの重要性が増大する中、ネットワークストレージにおいても様々なデータ保護要件の考慮が必須となっている。従来、保護要件の一つである、伝送中データの機密性保持のための手法として、encrypt-on-disk方式がある。これはデータを暗号化した状態でストレージに格納する方式で、転送時のみ暗号を用いるencrypt-on-wire 方式と比較して転送性能が良い反面、ユーザのアクセス権失効（revocation）に伴い、暗号化データを新しい鍵で再暗号化しなければならない。一方、高機能な分散ストレージシステムとしての自律ディスク［非特許文献３］は、ネットワークに接続された高機能ディスクノードのクラスタにより構成される。ノードの演算処理能力を利用し、耐故障化、負荷均衡化、容量分散等の機能を自律的に実行する。encrypt-on-disk 方式を採用した高機能ストレージでは、ユーザのアクセス権失効（revocation）に伴う再暗号化処理をストレージノード上で行うことが望ましい。そうすることでクライアントの負担を軽減でき、同時にencrypt-on-disk方式の目的である伝送路の機密性も維持できる。

【０００３】

encrypt-on-disk 方式は、ネットワークストレージにおける、伝送路上のデータを悪意

あるユーザの傍受から守る暗号利用方式の１つである。encrypt-on-disk 方式ではデータを暗号化した状態でストレージノードに格納する。その為、ストレージ側でのデータ送受信時に暗号処理を必要としない。他方式としてencrypt-on-wire 方式が挙げられる。この方式ではデータを平文の状態で格納し、送受信時に暗号化及び復号処理を行う。この二方式をデータ転送性能の面で比較すると、encrypt-on-disk方式の方が優れている。これはencrypt-on-wire 方式ではストレージ側送受信時に必ず暗号処理が発生すること、セッション毎に新しい暗号鍵を生成するため、鍵生成コストがかかることが理由である。

【０００４】

複数のユーザでデータを共有しているシステムの場合、encrypt-on-disk 方式ではユーザのアクセス権失効（revocation）に伴い、対象データを新しい暗号鍵で再暗号化する必要がある。これは、アクセス権を失効されたユーザ（revoked user）は現在使われている暗号鍵を保持している可能性があり、アクセス制御リスト（ACL）等のアクセス管理手法でrevoked user のアクセス要求を拒否しても、傍受等不当なアクセスでrevoked userにデータが渡ると、情報が漏洩してしまうからである。

【０００５】

そこで発明者は、分散ストレージにおける、性能とセキュリティを両立したアクセス権失効（revocation）時再暗号化手法［BA-Rev（Backup Assist Revocation）］［非特許文献４］を提案した。この手法では、再暗号化処理をストレージノード上で処理する。そのため予めバックアップデータを新しい鍵で暗号化しておき、アクセス権失効（revocation）発生時にプライマリとすることで、アクセス回復までの時間を再暗号化処理の分短縮することができる。次に元のプライマリをバックグラウンドで再暗号化し、新しいバックアップとして設定する。

【０００６】

なお特開２００５－２５２３８４号公報［特許文献１］には、暗号化データを保管するサーバシステムにおける再暗号化技術の例が開示されている。このサーバシステムにおいては、再暗号化の際に、新たな鍵と古い鍵との差分のみを代理サーバに送り、代理サーバは鍵の差分のみで直接再暗号化する。

【非特許文献１】E. Riedel、 M. Kallahalla、 and R. Swaminathan、 “A framework for evalating storage system security、 ” FAST ’02: Proceedingsof the 1st USENIX Conference on File and StorageTechnologies、 pp. 15－30、 USENIX Association、 2002.

【非特許文献２】P. Stanton、 “Securing Data in Storage: A Review of CurrentResearch、 ” ArXiv Computer Science e-prints、 2004.

【非特許文献３】H. Yokota、 “Autonomous Disks for Advanced Database Applications、 ” Proc. of International Symposium on Database Applications in Non-Traditional Environments（DANTE ’99）、pp. 435－442、Nov. 1999.

【非特許文献４】高山一樹、小林大、横田治夫、“複製を利用したストレージ中での暗号化データの権限失効処理”、第１８回データ工学ワークショップ（DEWS2007）、Feb./Mar. 2007

【特許文献１】特開２００５－２５２３８４号公報

【発明の開示】

【発明が解決しようとする課題】

【０００７】

再暗号化処理をストレージノード上で処理する場合において、通常の再暗号化手法では処理中に平文の状態がある、または再暗号化の為に獲得可能な形で暗号鍵が保存されている等の理由より、ストレージノード上の機密性は担保することができない。なお特許文献１に示されるようなサーバシステムは、処理コストの高い公開鍵暗号を用い、かつその中の特定の暗号アルゴリズムの性質に依存するシステムであるため、サーバーシステムの技術をそのままストレージノード上での再暗号化で使用するのには適さない。

【０００８】

本発明の目的は、ストレージノード上での再暗号化の際に、平文データを生成することながないストレージノード用再暗号化システム及び再暗号化方法並びにネットワークストレージを提供することにある。

【０００９】

本発明の他の目的は、ストレージノード上に暗号鍵を保存せずに上記目的を達成することができるストレージノード用再暗号化システム及びネットワークストレージを提供することにある。

【課題を解決するための手段】

【００１０】

本発明のストレージノード用再暗号化システムは、平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、暗号化データを読み出し可能に記憶するデータ保存部とを備えて、暗号化データを第２の暗号鍵で復号できる再暗号化データに変換する。ストレージノード用再暗号化システムは、暗号化データを第２の暗号鍵で復号できる再暗号化データに変換するために、二重暗号化部と復号化部とを備えている。二重暗号化部は、データ保存部に保存された暗号化データを第２の暗号鍵により二重に暗号化して二重暗号化データを作成する。復号化部は、二重暗号化データを第１の暗号鍵を用いて復号して再暗号化データを作成し、暗号化データを再暗号化データで置き換える。

【００１１】

本発明によれば、データ保存部に記憶された暗号化データを第２の暗号鍵により二重に暗号化する際には、平文データが生成されることはない。また二重暗号化データを第１の暗号鍵を用いて復号したときには、第１の暗号鍵が解かれて第２の暗号鍵により暗号化された再暗号化データが生成されるだけで、復号の途中において、平文データが生成されることはない。したがって本発明によれば、一度も平文データを生成することなく、ストレージノード上で再暗号化することができる。

【００１２】

第１の暗号鍵及び第２の暗号鍵は、ネットワークを介してつながる鍵サーバに公開鍵により暗号化して格納するのが好ましい。二重暗号化部は二重の暗号化に用いた第２の暗号鍵を鍵サーバに登録し、復号化部は鍵サーバから取得した第１の暗号鍵を秘密鍵を用いて復号する。このようにストレージノード上に第１の暗号鍵及び第２の暗号鍵を保存せずに鍵サーバに保存すれば、共通鍵の機密性を確保することができる。

【００１３】

ストレージノードが、第１の暗号鍵及び第２の暗号鍵を第１の公開鍵により暗号化して格納するロックボックスを備えていてもよい。この場合には、第１の公開鍵と組となる第１の秘密鍵を第２の公開鍵により暗号化して鍵サーバに登録する。そして二重暗号化部は、二重の暗号化に用いた第２の暗号鍵を第１の公開鍵を用いてロックボックスに格納する。そして復号化部は、第２の公開鍵により暗号化された第１の秘密鍵を鍵サーバから取得し、第２の公開鍵と組になる第２の秘密鍵を用いて、暗号化された第１の秘密鍵を復号する。また復号化部は、第１の公開鍵により暗号化された第１の暗号鍵をロックボックスから取得し、暗号化された第１の暗号鍵を第１の秘密鍵を用いて復号し、復号した第１の暗号鍵を用いて二重暗号化データを復号して再暗号化データを作成する。

【００１４】

このように鍵サーバを利用すると、ストレージノード上に設けたロックボックスに第１の暗号鍵及び第２の暗号鍵が格納されていたとしても、第三者が第２の秘密鍵を知らない限り、第１の秘密鍵を復号できない。そのために、ロックボックスから取り出した新たな共通鍵としての第２の暗号鍵を復号することができない。したがってストレージノード上のロックボックスに第１の暗号鍵及び第２の暗号鍵を格納したとしても、機密性を確保することができる。

【００１５】

また第１の暗号鍵及び第２の暗号鍵を格納するロックボックスをストレージノード上に設けた場合において、二重暗号化部及び復号化部をセキュリティチップによって構成して

もよい。セキュリティチップを用いると、ロックボックスから読み出した第１の暗号鍵及び第２の暗号鍵をセキュリティチップ内に機密状態で展開して、二重暗号化及び復号化も機密状態で実行することができる。したがってストレージノード上のロックボックス３６に第１の暗号鍵及び第２の暗号鍵を格納したとしても、機密性を確保することができる。

【００１６】

また暗号化部、二重暗号化部及び復号化部は、それぞれ暗号化と復号が可逆な暗号化モードにより暗号化または復号化を実行する。このような暗号化モードとしては、ＯＦＢ暗号化モードを用いることができる。これにより暗号化と復号とを平文を生成しないで再暗号化を実現することができ、再暗号化処理中の攻撃者によるノード陥落によるデータの漏洩を防ぐことができる。

【００１７】

本発明のストレージノード用再暗号化方法は、平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、暗号化データを読み出し可能に記憶するデータ保存部とを備えたストレージノードにおいて、暗号化データを第２の暗号鍵で復号できる再暗号化データに変換することをコンピュータの内部演算装置により実行する。本発明の方法では、データ保存部に記憶された暗号化データを第２の暗号鍵により二重に暗号化して二重暗号化データを作成する。次に、二重暗号化データを第１の暗号鍵を用いて復号して再暗号化データを作成する。そして暗号化データを再暗号化データで置き換える。

【００１８】

なお二重暗号化部及び復号化部は、データをブロック単位で暗号化及び復号するように構成するのが好ましい。このようにすると、データ処理が容易になる。

【００１９】

また本発明のネットワークストレージは、平文データを第１の暗号鍵を用いて暗号化して暗号化データを作成する暗号化部と、暗号化データを読み出し可能に記憶するデータ保存部とを備えた複数のストレージノードが、ネットワークに並列接続されて構成される。ストレージノードは、暗号化データを第２の暗号鍵で復号できる再暗号化データに変換する前述のストレージノード用再暗号化システムを備えている。ストレージノード用再暗号化システムは、データ保存部に記憶された暗号化データを第２の暗号鍵により二重に暗号化して二重暗号化データを作成する二重暗号化部と、二重暗号化データを第１の暗号鍵を用いて復号して再暗号化データを作成し、暗号化データを再暗号化データで置き換える復号化部とを備えている。そしてネットワークに接続された他のストレージノードに、第１の公開鍵により暗号化された第１の暗号鍵及び第２の公開鍵により暗号化された第２の暗号鍵を格納するロックボックスが設けられている。二重暗号化部は、第２の公開鍵により暗号化された第２の暗号鍵をロックボックスに格納する。また復号化部は、ロックボックスから第１の公開鍵により暗号化された第１の暗号鍵を第１の公開鍵と組になる第１の秘密鍵を用いて復号し、復号した第１の暗号鍵を用いて二重暗号化データを復号して再暗号化データを作成する。このように他のストレージノードにロックボックスを設ければ、ストレージノード上に暗号鍵が保存されないので、機密性を高めることができる。また再暗号化後の共通鍵となる第２の暗号鍵は第１の公開鍵とは異なる第２の公開鍵によって暗号化されているので、より機密性を高めることができる。

【発明の効果】

【００２０】

本発明によれば、データ保存部に記憶された暗号化データを第２の暗号鍵により二重に暗号化する際には、平文データが生成されることがなく、また二重暗号化データを第１の暗号鍵を用いて復号したときにも、第１の暗号鍵が解かれて第２の暗号鍵により暗号化された再暗号化データが生成されることはあっても、復号の途中において、平文データが生成されることはないので、一度も平文データを生成することなく、ストレージノード上で再暗号化することができ、高い機密性で再暗号化をすることができる。

【発明を実施するための最良の形態】

【００２１】

以下本発明のストレージノード用再暗号化システム及び再暗号化方法並びにネットワークストレージの実施の形態を詳細に説明する。本発明の実施の形態を説明する前に、本発明の実施の形態で使用するシステム、データ構造、暗号化モードについて概略を述べる。

【００２２】

［高機能分散ストレージシステム］

発明者等は、ストレージ装置上の演算処理能力を利用してデータの管理を自律的に行うシステムとして自律ディスク［非特許文献３］を提案してきた。自律ディスクは、ネットワークに接続された高機能ディスクノード即ちストレージノードのクラスタにより構成される。この高機能ディスクの演算能力を利用し、ストレージ側で耐故障化、負荷均衡化、容量分散等の機能を自律的に実行し、ユーザによるストレージ管理の負担を軽減する。本実施の形態は、前述のencrypt-on-disk 方式を採用した、自律ディスクのような高機能ストレージシステムで用いるストレージノード用再暗号化システム及びネットワークストレージを対象とする。自律ディスクの方針に従い、アクセス権失効（revocation）に伴う再暗号化処理は、ストレージノード側で行い、クライアント側の負担を軽減する。

【００２３】

［ネットワーク構成］

本実施の形態のネットワークストレージでは、複数のストレージノードが並列にネットワーク接続される。そしてこれらのストレージノードに対し、同様にネットワークに接続されたクライアントノードからアクセスするものとする。

【００２４】

［使用する暗号技術］

暗号は、暗号化と復号に共通の鍵を用いる共通鍵暗号と、異なる対の鍵を用いる公開鍵暗号に分類できる。共通鍵暗号は、予め暗号化側と復号側で、安全な方法で鍵を共有していなければならない。一方、公開鍵と秘密鍵の対を用いる公開鍵暗号では、対の内の片方（公開鍵）を公開することができる為、鍵配布の問題はない。しかし、その反面、公開鍵暗号は一般的に共通鍵暗号の数百から数千倍の処理速度であるという問題がある。これらの特性を考慮し、一般的には共通鍵の配布を公開鍵暗号を用いて行い、その共通鍵を用いてデータのやりとりをする場合が多い。本実施の形態においても、この方式を採用した。

【００２５】

本実施の形態の一部では、データの暗号鍵を管理する構造としてロックボックスを用いる。ロックボックスとは、暗号鍵を格納し、アクセス権を持つユーザのみ鍵を取り出すことができる鍵管理構造である。図１には、非特許文献（E. Miller、D. Long、W. Freeman、and B. Reed、“Strong Security for Network-Attached Storage、”FAST’02:Proceedings of the 1st USENIX Conference on File and Storage Technologies、p. 1、Berkeley、CA、USA、2002、USENIX Association. ）におけるロックボックスであるkey objectの例を示している。ユーザＸは、公開鍵Kx^{+}と秘密鍵Kx^{-}を持ち、あるデータは共通鍵Ｋで暗号化されているものとする。key object の各行はユーザID、ユーザの公開鍵で暗号化された共通鍵$Kx^{+}(K)$、ユーザに認可されているアクセス権の種類Ｐx からなる。格納された共通鍵を獲得するには、公開鍵と対を成す、ユーザ自身のクライアントノード上に保存された秘密鍵を用いてのみ復号できるため、正しく認可されたユーザのみが共通鍵を獲得できる。本発明の一部の実施の形態では、このkey object の構造を利用して鍵を管理する。１ファイルに対し１つのkey object が対応し、ファイルを暗号化した鍵は、ストレージノード上に格納される。また本発明の一部の実施の形態では、ストレージ側で再暗号化等の暗号処理を行うため、各ストレージノードもユーザと同様に公開鍵と秘密鍵を持ち、処理対象のファイルのkey object に獲得可能な鍵を保持するものとする。

【００２６】

［ＯＦＢ暗号化モード］

ＯＦＢ（Output Feed Back）モードは、ブロック暗号の暗号化モードの一つである。本実施の形態では、この暗号化モードを使用した。処理の流れの簡略図を図２に示す。暗号文の最初のブロックは初期ベクトル（Initial Vector:IV）を暗号化して疑似乱数ビット

列を生成し、平文のブロックとの排他的論理和演算で得られる。後続のブロックは、前ブロックの疑似乱数ビット列をさらに暗号化し、同様に平文との排他的論理和演算をすることで得られる。ＯＦＢモードの特色として、暗号処理部分を平文とは独立に実行でき、実際の平文への処理は排他的論理和のみである点、ストリーム暗号的に利用される点が挙げられる。また暗号化と復号の処理が同じであるのも特徴である。ＯＦＢモードは初期ベクトルの暗号化に始まり、先頭ブロックから順次処理しなければならないため、ランダムアクセスに弱いという欠点がある。これに対し、CTR（counter）モードというランダムアクセスに強いモードがある。CTRモードでは暗号化処理の入力として、前ブロックの疑似乱数ビット列ではなく、ブロックの位置に対応した値を持つカウンタの値を入力として用いるため、途中のブロックからの処理が可能である。本発明では、CTR（counter）モードを用いて暗号化することも可能である。

【００２７】

なお以下の実施の形態において、想定される攻撃者は、対象となる分散ストレージシステムと同じネットワーク上に存在し、次の攻撃を行えるものとする。

【００２８】

・伝送路へ攻撃し、転送中のデータを傍受する。

【００２９】

・あるストレージノードに攻撃し、陥落させて内部のデータを奪取する。なお、攻撃者は格納されたファイルへのアクセス権は持たず、鍵等いかなるデータも所持していないものとする。

【００３０】

上記の攻撃を防ぐ本発明の実施の形態のストレージノード用再暗号化システムは、自律ディスク等の高機能ストレージシステムにおいて、様々な処理をストレージノード側で自律的に処理することで、ユーザやストレージ管理者の負担を軽減する。伝送データ保護方式の中ではデータ転送性能が良いencrypt-on-disk 方式を高機能ストレージに適用する場合、アクセス権失効（revocation）時の再暗号化処理もストレージ側で処理することで、同様にクライアントの負担を軽減することができる。しかしその一方で、通常の再暗号化処理では、暗号化データを復号し、その後暗号化するため、中間生成物として平文が現れるので、ストレージノード上の機密性は守られないという問題が発生する。この問題点に対し、本発明の実施の形態のストレージノード用再暗号化システムを用いると、処理中に平文が生成されることがない。

【００３１】

以下に説明する本実施の形態では、暗号化モードとして、ＯＦＢモードを用いる。ＯＦＢモードでは、暗号化は平文と疑似乱数ビット列の排他的論理和を求めることで実行される。また復号は暗号文に対して同様の処理を行うことで実行される。ここで、排他的論理和は交換則が成り立つため、再暗号化処理における暗号化と復号も可逆である。この性質を利用し、再暗号化処理を“暗号化”、“復号”の順に処理することで、平文を生成しない再暗号化を実現でき、再暗号化処理中の攻撃者による当該ストレージノードの陥落によるデータ漏洩を防ぐことが可能となる。なお、定常状態を含めて、１つのストレージノード陥落による漏洩を防ぐためには、鍵管理の考慮が必要である。再暗号化をストレージ側で行うために、共通鍵をストレージノードが利用可能な形で保存する必要があるが、この無防備な状態の鍵を暗号化ファイルと同じノードに置くと、そのノードが陥落した時点でデータが漏洩してしまうからである。対策としては、後述する一つの実施の形態のように他ストレージノードで鍵を管理する方法や、暗号化をセキュリティプラットフォームまたはセキュリティチップのような専用のハードウェアを用いて行うことで鍵を奪取不可能にする方法等が考えられる。後者の例としては、ハードディスク（HDD）に暗号化用回路を組み込んだシーゲイト（Seagate）・テクノロジー社のDriveTrust（商標）や富士通株式会社のMTZ2 CJがある。

【００３２】

図３はストレージノードに適用される本発明のストレージノード用再暗号化システムの

第１の実施の形態の構成を示すブロック図であり、図４は第１の実施の形態の再暗号化を概念的に示す図であり、図５は本発明における再暗号化の流れを示す図である。なお図５に示す鍵管理構造は、鍵サーバやロックボックスによって実現される。

【００３３】

このストレージノード用再暗号化システムは、図４のストレージノードＡ上に暗号化部１と、データ保存部２と、二重暗号化部３と復号化部４とを備えており、ネットワークで接続された鍵サーバ５もこのシステムの一部を構成している。なお本実施の形態では、ストレージノードとして前述の高機能分散ストレージシステムで使用される高機能ディスクノードを用いている。またネットワークＮＷには、前述のように複数のストレージノードが並列に接続されている。そして暗号化部１、二重暗号化部３及び復号化部４で使用する暗号化モードとしては、前述のＯＦＢ暗号化モードを採用している。

【００３４】

暗号化部１は、平文データＦを第１の暗号鍵K_1を用いて暗号化して暗号化データK_1（Ｆ）を作成する。データ保存部２には、暗号化データK_1（Ｆ）が読み出し可能に保存される。ユーザのアクセス権失効（revocation）に伴い、暗号化データを新しい鍵で再暗号化する場合には、二重暗号化部３が、データ保存部２に記憶された暗号化データK_1（Ｆ）を第２の暗号鍵K_2により二重に暗号化して二重暗号化データK_1K_2（Ｆ）を作成する。そして復号化部４は、二重暗号化データK_1K_2（Ｆ）を第１の暗号鍵K_1を用いて復号して再暗号化データK_2（Ｆ）を作成し、データ保存部２に保存されている暗号化データK_1（Ｆ）を再暗号化データK_2（Ｆ）で置き換える。本実施の形態では、第１の暗号鍵K_1及び第２の暗号鍵K_2は、ネットワークＮＷを介してつながる鍵サーバ５に登録（または管理）されている。前述のように、ユーザＸは、公開鍵Kx^+と秘密鍵Kx^-とを持っている。第１の暗号鍵K_1及び第２の暗号鍵K_2は、公開鍵Kx^+により暗号化されて、Kx^+（K_1）及びKx^+（K_2）として鍵サーバ５に登録されている。なお再暗号化が完了した後は、暗号化された第１の暗号鍵Kx^+（K_1）は鍵サーバ５から削除してもよい。また暗号化された第２の暗号鍵Kx^+（K_2）は、二重暗号化部３で二重暗号化する際に鍵サーバ５に共通鍵として登録される。ユーザＸは、ストレージノードＡの復号化部４で、秘密鍵Kx^-を用いて暗号化された第１の暗号鍵Kx^+（K_1）を復号して第１の暗号鍵K_1を取得する。復号化部４は、第１の暗号鍵K_1を用いて二重暗号化データK_1K_2（Ｆ）を復号して、第２の暗号鍵K_2で復号できる再暗号化データK_2（Ｆ）を出力する。

【００３５】

本実施の形態のストレージノード用再暗号化システムによれば、データ保存部２に保存された暗号化データK_1（Ｆ）を二重暗号化部３で第２の暗号鍵K_2により二重に暗号化する際に、平文データＦが生成されることはない。また二重暗号化データK_1K_2（Ｆ）を復号化部４で第１の暗号鍵K_1を用いて復号したときには、第１の暗号鍵K_1が解かれて第２の暗号鍵K_2により暗号化された再暗号化データK_2（Ｆ）が生成されるだけで、復号の途中において、平文データＦが生成されることはない。したがって本実施の形態によれば、一度も平文データＦを生成することなく、ストレージノード上で再暗号化することができる。また本実施の形態では、第１の暗号鍵K_1及び第２の暗号鍵K_2は、ネットワークＮＷを介してつながる鍵サーバ５に格納されているので、再暗号化データK_2（Ｆ）の復号に必要な第２の暗号鍵K_2をストレージノードに対する攻撃から守って、機密性を確保することができる。

【００３６】

図６及び図７は、本発明のストレージノード用再暗号化システムの第２の実施の形態のブロック図及び再暗号化の概念図である。なお図３及び図４に示した第１の実施の形態で用いるブロックと同様のブロックには、図３及び図４に示した符号の数に１０の数を加えた数の符号を付して説明を省略する。第２の実施の形態も図５の再暗号化の処理の流れで再暗号化を実行する。第２の実施の形態では、ストレージノードが、第１の暗号鍵K_1及び第２の暗号鍵K_2を第１の公開鍵Kx^+により暗号化して格納するロックボックス１６を備えている。再暗号化の前には、第１の暗号鍵K_1が第１の公開鍵Kx^+により暗号化され

てロックボックス１６に格納されている。そして本実施の形態では、第１の公開鍵Kx^+と組になる第１の秘密鍵Kx^-が第２の公開鍵K_{YK}^+により暗号化されて鍵サーバ１５に登録されている。二重暗号化部は、二重の暗号化に用いた第２の暗号鍵K_2を第１の公開鍵Kx^+により暗号化してロックボックス１６に格納する。そして復号化部１４は、鍵サーバ１５から取得した暗号化された第１の秘密鍵Kx^-を、第２の公開鍵K_{YK}^+と組になる第２の秘密鍵K_{YK}^-により復号する。また復号化部１４は、第１の公開鍵Kx^+により暗号化された第１の暗号鍵K_1をロックボックス１６から取得し、暗号化された第１の暗号鍵K_1を復号した第１の秘密鍵Kx^-を用いて復号する。そして復号化部１４は、復号した第１の暗号鍵K_1を用いて二重暗号化データK_1K_2（Ｆ）を復号して再暗号化データK_2（Ｆ）を作成する。

【００３７】

本実施の形態のように鍵サーバ１５を利用すると、ストレージノード上に設けたロックボックス１６に第１の暗号鍵K_1及び第２の暗号鍵K_2が格納されていたとしても、第三者が第２の秘密鍵K_{YK}^-を知らない限り、第１の秘密鍵Kx^-を復号できないために、ロックボックス１６から取り出した第２の暗号鍵K_2を復号することができない。したがってストレージノード上のロックボックス１６に第１の暗号鍵K_1及び第２の暗号鍵K_2を格納したとしても、機密性を確保することができる。

【００３８】

図８及び図９は、本発明のネットワークストレージで用いられるストレージノード用再暗号化システムの第３の実施の形態のブロック図及び再暗号化の概念図である。なお図３及び図４に示した実施の形態で用いるブロックと同様のブロックには、図３及び図４に示した符号の数に２０の数を加えた数の符号を付して説明を省略する。この実施の形態も図５の再暗号化の処理の流れで再暗号化を実行する。このネットワークストレージは、平文データを第１の暗号鍵K_1を用いて暗号化して暗号化データを作成する暗号化部２１と、暗号化データを読み出し可能に記憶するデータ保存部２２とを備えた複数のストレージノードＡ，Ｂが、ネットワークＮＷに並列接続されて構成される。ストレージノードＡは、暗号化部２１とデータ保存部２２とを含んで構成されて、暗号化データを第２の暗号鍵K_2で復号できる再暗号化データに変換するストレージノード用再暗号化システムを備えている。ストレージノード用再暗号化システムは、データ保存部２２に記憶された暗号化データを第２の暗号鍵K_2により二重に暗号化して二重暗号化データを作成する二重暗号化部２３と、二重暗号化データK_1K_2（Ｆ）を第１の暗号鍵K_1を用いて復号して再暗号化データK_2（Ｆ）を作成し、暗号化データK_1（Ｆ）を再暗号化データK_2（Ｆ）で置き換える復号化部２４とを備えている。そしてネットワークＮＷに接続された他のストレージノードＢに、第１の公開鍵Kx^+により暗号化された第１の暗号鍵K_1及び第２の公開鍵K_{XK}^+により暗号化された第２の暗号鍵K_2を格納するロックボックス２６が設けられている。二重暗号化部２３は、第２の公開鍵K_{XK}^+により暗号化された第２の暗号鍵K_2をロックボックス２６に格納する。また復号化部２４は、ロックボックス２６から第１の公開鍵Kx^+により暗号化された第１の暗号鍵K_1を第１の公開鍵Kx^+と組になる第１の秘密鍵Kx^-を用いて復号し、復号した第１の暗号鍵K_1を用いて二重暗号化データK_1K_2（Ｆ）を復号して再暗号化データK_2（Ｆ）を作成する。このように他のストレージノードＢにロックボックス２６を設ければ、ストレージノードＡ上に暗号鍵が保存されないので、機密性を高めることができる。また再暗号化後の共通鍵となる第２の暗号鍵K_2は第１の公開鍵Kx^+とは異なる第２の公開鍵K_{XK}^+によって暗号化されているので、より機密性を高めることができる。なお各ストレージノードＡ，Ｂには、他のストレージノードのためのロックボックス２６がそれぞれ設けられている。

【００３９】

図１０及び図１１は、本発明のストレージノード用再暗号化システムの第４の実施の形態のブロック図及び再暗号化の概念図である。なお図８及び図９に示した実施の形態で用いるブロックと同様のブロックには、図８及び図９に示した符号の数に１０の数を加えた数の符号を付して説明を省略する。この実施の形態では、図１２に示す再暗号化の処理の

流れで再暗号化を実行する。本実施の形態では、第１の暗号鍵K_1及び第２の暗号鍵K_2を格納するロックボックス３６をストレージノード上に設けている。ロックボックス３６に格納される第１の暗号鍵K_1及び第２の暗号鍵K_2は、公開鍵Kx^+により暗号化されて、Kx^+（K_1）及びKx^+（K_2）として格納されている。また本実施の形態では、二重暗号化部３３及び復号化部３４をセキュリティチップ３７によって構成している。セキュリティチップ３７は、ロックボックス３６から読み出した第１の暗号鍵K_1及び第２の暗号鍵K_2を機密状態で内部で展開し、二重暗号化及び復号化も機密状態で実行される。したがってストレージノードのロックボックス３６に第１の暗号鍵K_1及び第２の暗号鍵K_2を格納したとしても、機密性を確保することができる。本実施の形態では、図１２に示すように、データ保存部２に記憶された暗号化データK_1（Ｆ）をブロック単位K_1（Ｆ）［１］～K_1（Ｆ）［ｎ］に分割して、セキュリティチップ３７へブロック転送する。セキュリティチップ３７では、ブロック転送された暗号化データ（Ｆ）［１］～K_1（Ｆ）［ｎ］を、第２の嵌合鍵K_2を用いて順次二重に暗号化して、ブロック単位の二重暗号化データK_1K_2（Ｆ）［１］～K_1K_2（Ｆ）［ｎ］を作成する。そしてセキュリティチップ３７は、復号化部３４において、ロックボックスから取得し暗号化された第１の暗号鍵K_1を秘密鍵Kx^-を用いて復号し、復号した第１の暗号鍵K_1を用いてブロック単位の二重暗号化データK_1K_2［１］～K_1K_2（Ｆ）［ｎ］を復号し、ブロック単位の再暗号化データK_2（Ｆ）［１］～K_2（Ｆ）［ｎ］を作成して、データ保存部３２に転送する。データ保存部３２では、ブロック単位の再暗号化データK_2（Ｆ）［１］～K_2（Ｆ）［ｎ］を合成して、暗号化データK_1（Ｆ）を再暗号化データK_2（Ｆ）で置き換える。

【００４０】

本発明ではデータ暗号化にブロック暗号を用いる為、上記ブロック単位としてブロック暗号におけるブロックをそのまま適用することで処理が可能となる。

【００４１】

以下本発明を性能面で評価する為、ＰＣクラスタ上で動作するencrypt-on-disk方式のファイルサーバ・クライアントプログラムを作成して、実験を行った結果について説明する。

【００４２】

［実験プログラム］

実験は、図８及び図９に示した第３の実施の形態と従来の再暗号化方法（通常の再暗号化）を用いる場合について行った。そしてファイル獲得（get）の応答時間と、ストレージノード上の再暗号化処理の実行時間を測定した。通常の暗号化では、通常の再暗号化処理を行うために、二重暗号化をせずに、暗号化データを第１の暗号鍵K_1で復号した後、第２の暗号鍵K_2で暗号化することにより再暗号化処理を行う。ロックボックスは対象の暗号化ファイルと同じストレージノードに格納する。

【００４３】

暗号の利用方法や鍵管理は前述の記述に従った。図８及び図９に示した第３の実施の形態では、初期状態として、ファイルＦはユニークな第１の暗号鍵K_1、ＯＦＢモードで暗号化されてストレージノードＡに格納され、そのロックボックスは隣接するストレージノードＢに格納されている。ロックボックス２６には再暗号化処理用に、公開鍵Kx^+で暗号化された第１の暗号鍵K_1を保存している。なお、プログラムは以下の手順で処理を行うことで実際に平文を生成せず再暗号化処理を完了できることを確認できた。

【００４４】

step１：暗号化ファイルＦを格納したストレージノードで第２の暗号鍵K_2を生成し、第２の暗号鍵K_2で二重に暗号化する。この段階でファイルは二重に暗号化された状態K_1K_2（Ｆ）となる。

【００４５】

step２：ストレージノードＢのロックボックス２６に第２の暗号鍵K_2を転送し、代わりに第１の暗号鍵K_1を受信する。この時鍵交換は送信先ノードの公開鍵Kx^+で暗号化した状態で行うため、送信前後及びロックボックス２６の再構成時に秘密鍵Kx^-と公開鍵K

x^+とを用いた復号、暗号化の必要がある。

【００４６】

step３：第１の暗号鍵K_1で二重に暗号化されたファイルK_1K_2（Ｆ）の復号を行い、その後第１の暗号鍵K_1をロックボックス２６から破棄する。プログラムで実行できる処理のうち、ファイル獲得（get）は、ロックボックスに格納された対象ユーザ用の共通鍵の転送も含む。その為、ロックボックスの格納場所が異なる本実施の形態では、再暗号化処理だけでなくファイル獲得（get）の処理も流れが異なるため、以下評価の対象とする。

【００４７】

［実験環境］

実験は表１に示す構成のＰＣクラスタ（ストレージノード）上で行った。またパラメータとして表２のものを用いた。

【表１】

CPU	AMD Athlon XP-M1800+ (1.53GHz)
Memory	PC2100 DDR SDRAM 1GB
HDD	TOSHIBA MK3019GAX (30GB, 5400rpm, 2.5inch)
Network	TCP/IP + 1000BASE-T
OS	Linux 2.4.20
Java VM	Sun J2SE SDK 1.5.0_03 Server VM

【表２】

公開鍵	RSA 1024bit
共通鍵	AES 128bit
暗号化モード	OFB
パディング	NoPadding
ストレージノード数	3
ノード当たりデータサイズ	1MB×500
Zipf 母数 θ	0.7

【００４８】

上記表の記述の通り、実験では３台のストレージノードにそれぞれ１ＭＢ、５００個のファイルを暗号化して格納する。ただし総データサイズにロックボックスのサイズは含まない。また、ファイル獲得の対象となるファイルは、選択したストレージノードの全ファイルの中から、パラメータμ によって決まるZipf分布に従い偏って選ばれるものとした。Zipf分布については、D. E. Knuth著の「Sorting 及びSearching」（Addison-Wesley Publishing Company、1973）に記載されている。

【００４９】

［実験１：ファイル獲得（get）の応答時間の比較］

ファイルを格納した３つのストレージノードに対し、それぞれ１つのストレージノード（クライアントノード）からファイル獲得（get）リクエストを１０００回実行し、その

応答時間を測定した。

【００５０】

図１３は従来の通常の再暗号化（normal）と本実施の形態（RORE）の再暗号化における、３ストレージノード分の平均応答時間及び９５％信頼区間を表している。実験の結果、本実施の形態を適用した環境ではファイル獲得（get）の応答時間が平均で約４．５パーセント増加した。これは、本実施の形態では、ファイルを格納しているストレージノードＡがロックボックスを格納する隣接したストレージノードからアクセスしたユーザに対応する共通鍵（第１及び第２の暗号鍵）を取得しなければならないためであると考える。ただし鍵は小さい固定長である為、その転送の影響は非常に小さいことがわかる。

【００５１】

［実験２：再暗号化の処理時間の比較］

ファイルを格納した３つのストレージノード上で、それぞれが再暗号化処理を１０００回実行し、その実行開始から終了までの時間を測定した。ここで再暗号化対象のファイルはランダムで選択するものとする。図１４は従来の通常の再暗号化（normal）と本実施の形態（RORE）における処理時間の平均と９５％信頼区間を表している。図より、本実施の形態（RORE）の再暗号化処理時間は、通常の再暗号化（normal）に比べて約１００ミリ秒、約４７パーセント増加した。これはストレージノード間通信が必要であることに加え、共通鍵転送時に公開鍵暗号による暗号処理が必要であるためだと考える。ここでは、ストレージノードＡからストレージノードＢへの新しい鍵の送受信時、新しい鍵のロックボックスへの設定時、ストレージノードＢからストレージノードＡへの古い鍵の送受信時に公開鍵暗号での暗号処理が必要となる。実験１の結果より、鍵の転送処理自体の影響は非常に小さいことより、公開鍵暗号による処理が性能に与える影響は大きいことがわかる。

【００５２】

［実験３：再暗号化処理が他アクセスに与える影響の比較］

再暗号化処理が、同時に発生した他のアクセスに対してどの程度の影響を与えるかを実験により測定した。ストレージノードＡ、Ｂにファイルを格納した状態で、ストレージノードＡで再暗号化処理を連続で実行する。この時ストレージノードＢは、ストレージノードＡに格納されたファイルのロックボックスが格納されるノードである。すなわち、従来の通常の再暗号化（normal）ではストレージノードＡのみで再暗号化処理が実行され、本実施の形態（RORE）では、図８及び図９のように、ストレージノードＡで再暗号化処理、ストレージノードＢでロックボックスの処理が実行される状態である。この状態でストレージノードＡ、Ｂに対しファイル獲得（get）を１０００回実行し、応答時間を測定した。図１５は通常の再暗号化（normal）と本実施の形態（RORE）において、各ストレージノードに対するファイル獲得（get）の平均応答時間、９５％信頼区間を表す。結果は、再暗号化処理の無い平常状態（図のnormal:ストレージノードＢ）と比較して、ファイル獲得（get）の応答時間は通常の環境で約２９パーセント、本実施の形態（RORE）ではストレージノードＡ、Ｂ共に、約１４パーセント増加した。通常では再暗号化処理が１つのストレージノード内で行われる分、大きく他のアクセスに影響を与えるのに対し、処理が２つのストレージノードに分散される本実施の形態（RORE）では、他のアクセスへの影響も分散されることがわかる。

【００５３】

実験１の結果より、ファイル本体とロックボックスの格納ノード分離によるファイル獲得（get）の性能低下は非常に小さいことがわかり、無視できる程度であると考える。実験２より、再暗号化の処理時間は増加することが判る。その為、再暗号化の発生箇所とタイミングによっては性能低下を引き起こす可能性がある。例えばアクセス対象のファイルが再暗号化中の場合、処理が終わるまでアクセスがブロックされ、応答時間が長くなる。バックグラウンドでの再暗号化処理による性能への影響は、実験３の結果より、本実施の形態（RORE）では複数ストレージノードで影響が出る反面、１ノード当たりの性能劣化度合は小さい。QoS（Quality of Service）の観点から見ると、突出して性能劣化するストレージノードがない為、優れていると言える。

【００５４】
また本実施の形態（RORE）を適用することにより、再暗号化処理中に平文が生成されず、また１つのストレージノードに平文生成が可能となるデータ（暗号化ファイル、共通鍵）が同時に存在する期間がないため、１ストレージノード陥落時のデータ漏洩を防ぐことができる。また、あるファイルに関する、ファイルを格納しているストレージノードと対応するロックボックスを格納しているストレージノードのうち、何れかが陥落しない限り、データ漏洩は防ぐことができる。一方で、あるファイルを格納しているストレージノードとロックボックスを格納しているストレージノードが同時に陥落した場合、或いはロックボックスが格納されているストレージノードが陥落し、かつ通信中の対象ファイルが傍受された場合、そのファイルは漏洩する。しかし、再暗号化処理で平文が生成されてしまう通常の方式のように、１つのストレージノード陥落でデータが漏洩してしまうシステムと比較すると、分散ストレージシステムの構造上、本実施の形態によれば攻撃者が複数ステップを踏まなければデータが漏洩することがないため、本実施の形態のほうが従来の技術と比べて、セキュリティ面で優れているといえる。また、既存のセキュアストレージシステムのようにシステムをtrustでないと仮定した環境と比較すると、本実施の形態ではストレージ側で再暗号化処理等を行うために生の鍵を扱っている分、セキュアでないといえる。しかしその反面、再暗号化処理をユーザの手を煩わせずストレージ側で行うことで、クライアント側の負担を大きく軽減することが可能となっている。そのため、本実施の形態（RORE）を適用した高機能ストレージシステムは、多人数で共有する大容量の分散ファイルサーバ等への適用に適している。このようなシステムではデータ管理コストが大きく、ストレージ主導で管理を行い、ユーザ側の負担を軽減することが望ましい。また本実施の形態（RORE）を適用することで、アクセス権失効（revocation）時のクライアント側の負担及びシステム性能劣化を防ぎ、かつ通常の再暗号化処理を行うよりも高い機密性を維持することができる。

【００５５】
上記各実施の形態では、再暗号化処理中の暗号化処理と復号処理を可逆にできるＯＦＢ暗号化モードを利用することで、途中生成物として平文を生成しない再暗号化処理を実現することができ、再暗号化処理時のストレージノード陥落による漏洩を防ぐことができる。それに加え、暗号化ファイルの保存と鍵の管理を異なるノードで管理する、或いは暗号処理専用のハードウェア（セキュリティチップ）をストレージノードに搭載し鍵を奪取できないようにすると、１つのストレージノード内のデータで平文が奪取可能な期間が無いように処理を考慮することで、上記の１つのストレージノード陥落に対する機密性が実現可能となる。

【図面の簡単な説明】

【００５６】
【図１】ロックボックスの例を示す図である。
【図２】ＯＦＢ暗号モードを使用した処理の流れを示す図である。
【図３】ストレージノードに適用される本発明のストレージノード用再暗号化システムの第１の実施の形態の構成を示すブロック図である。
【図４】第１の実施の形態の再暗号化を概念的に示す図である。
【図５】本発明における再暗号化の流れを示す図である。
【図６】本発明のストレージノード用再暗号化システムの第２の実施の形態のブロック図である。
【図７】第２の実施の形態の再暗号化の概念図である。
【図８】本発明のネットワークストレージで用いられるストレージノード用再暗号化システムの第３の実施の形態のブロック図である。
【図９】本発明のネットワークストレージで用いられるストレージノード用再暗号化システムの第３の実施の形態の再暗号化の概念図である。
【図１０】本発明のストレージノード用再暗号化システムの第４の実施の形態のブロック図である。

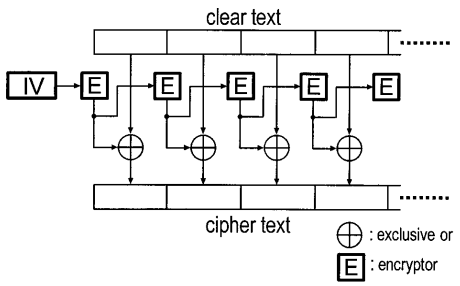
【図１１】第４の実施の形態の再暗号化の概念図である。
【図１２】第４の実施の形態における再暗号化の処理の流れを示す図である。
【図１３】ファイル獲得（get）コマンドにおける平均応答時間の比較を示す図である。
【図１４】暗号化の処理時間の比較を示す図である。
【図１５】バックグラウンドの際暗号化処理がファイル獲得（get）に与える影響の評価を示す図である。
【符号の説明】
【００５７】
１，１１，２１，３１　暗号化部
２，１２，２２，３２　制データ保存部
３，１３，２３，３３　二重暗号化部
４，１４，２４，３４　復号化部
５　鍵サーバ
１６，２６，３６　ロックボックス

【図１】

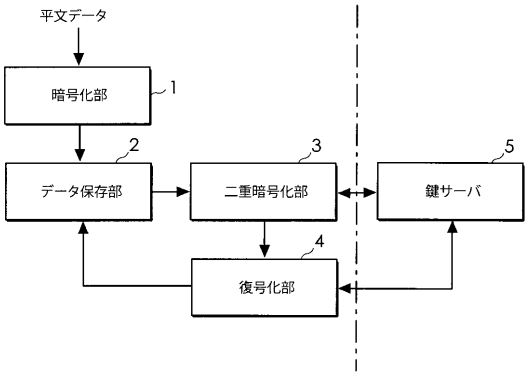
Key Oblect ID	User ID	Signature	Reference Count
User ID	Encrypted key		Permissions
A	$K_A^+(K)$		P_A
B	$K_B^+(K)$		P_B

K_A^+ : A's public key
K : secret key

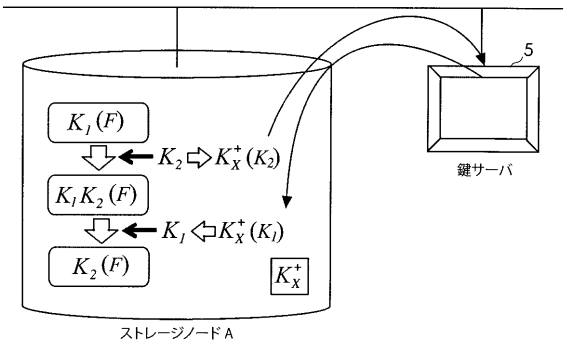
【図２】



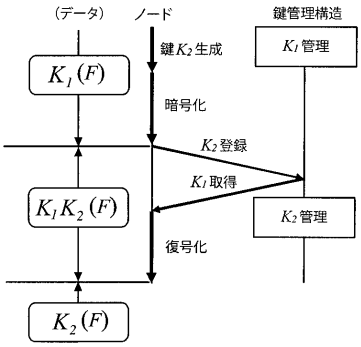
【図３】



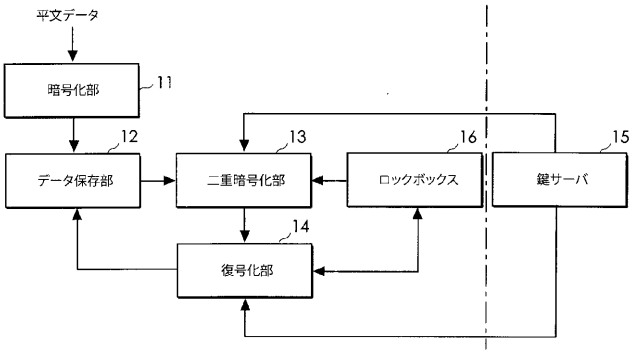
【図４】



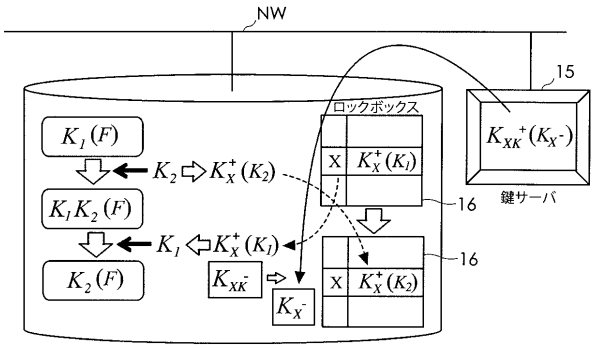
【図５】



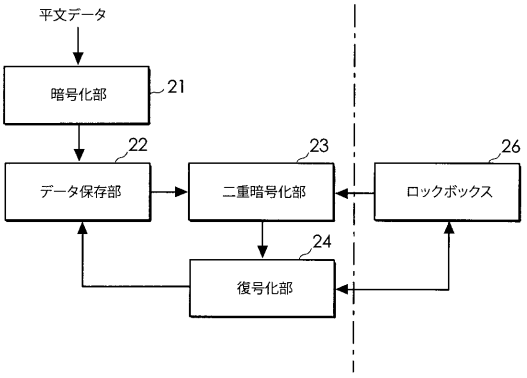
【図６】



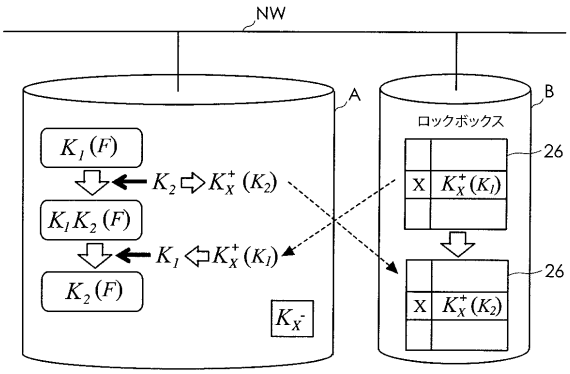
【図７】



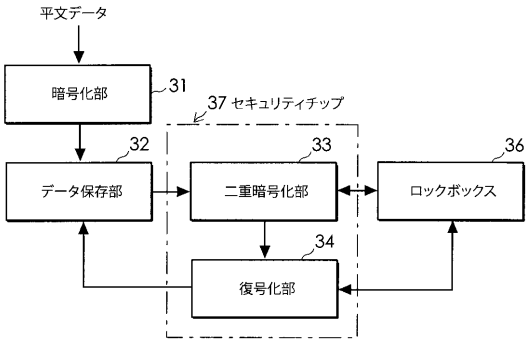
【図８】



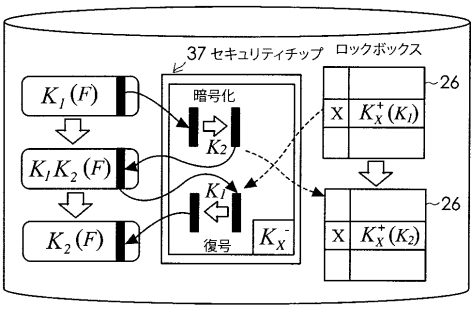
【図９】



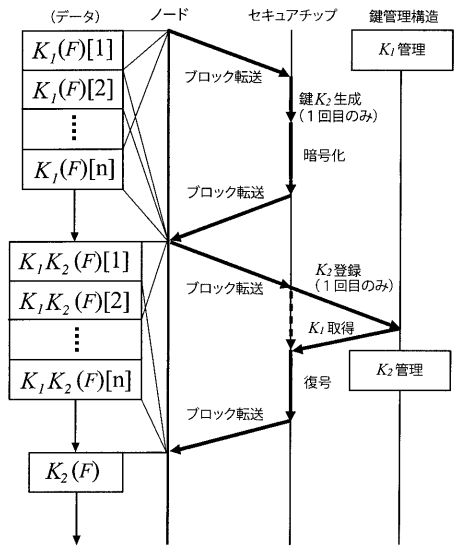
【図１０】



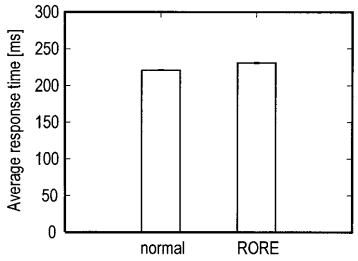
【図１１】



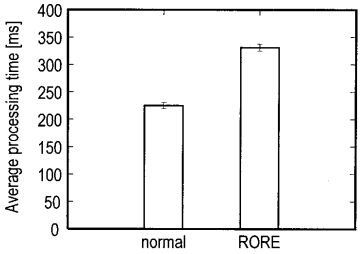
【図１２】



【図１３】



【図１４】



【図１５】