社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
NS2001-148, CQ2001-60, TM2001-38 (2001-11)

ダブルチェック型文書管理システム

原田 篤史[1]　西垣 正勝[2]　曽我 正和[3]　田窪 昭夫[4]

[1]静岡大学大学院情報学研究科　〒432-8011 浜松市城北 3-5-1
[2]静岡大学情報学部　〒432-8011 浜松市城北 3-5-1
[3]岩手県立大学ソフトウェア情報学部　〒020-0173 岩手県滝沢村滝沢字巣子 152-52
[4]三菱電機情報システム製作所　〒247-8520 鎌倉市上町屋 325
E-mail: nisigaki@cs.inf.shizuoka.ac.jp

あらまし　ネットワークに接続して稼動するシステムにおいて，部外者による攻撃はユーザ認証やファイヤウォールなどを適切に用意することにより，そのセキュリティを強固にすることはできる．しかし，部内者からの攻撃を防ぐことは困難である．部内者はそのシステムやデータにアクセスすることができる正規ユーザであるため，万一，彼らが悪意を持っていた場合にはシステムが悪用され内部のデータが改竄・消去されてしまう．また，悪意は無いにしても操作ミスや過失により同様の結果が起こる危険がある．本稿では，このような部内者の操作ミスや破壊行為からデータを守る方法について述べる．

キーワード 文書管理，データベース，改竄検出，ワンタイムライトオンリー，ヒステリシス署名

A Document management system with supervised check

Atsushi HARADA[1], Masakatsu NISHIGAKI[2], Masakazu SOGA[3] and Akio TAKUBO[4]

[1]Graduate school of Information, Shizuoka University,
3-5-1 Johoku, Hamamatsu, 432-8011, Japan
[2]Faculty of Information, Shizuoka University,
3-5-1 Johoku, Hamamatsu, 432-8011, Japan
[3]Faculty of Software and Information, Iwate Prefectural University,
Sugo 152-52, Takizawa, Iwate, 020-0173, Japan
[4]Mitsubishi Electric Corp.
325 Kamimachiya, Kamakura, 247-8520, Japan
E-mail: nisigaki@cs.inf.shizuoka.ac.jp

Abstract　For a system working with a network, the security against the outsiders could be kept by user authentication and/or firewall. However, it is difficult to defend data from attacks by the insiders. The insiders are legal users of the system and permitted to access data of it. Therefore, if they are malicious or make a mistake, the data can be destroyed. This paper proposes a scheme to protect the data from the insiders.

Key words　document management, database, fraud detection, write-once, hysteresis signature

1．はじめに

昨今のインターネットの急激な発展に伴い，組織内外をネットワークで結合して業務の高速化・円滑化を図ることが当然となった．それによって，業務上重要なデータがネットワーク上に存在することも珍しくなくなり，十分なセキュリティの確保が求められている．ネットワークを介して稼動するシステムにおいて，部外者による攻撃はユーザ認証やファイヤウォールなどを適切に用意して対処することにより，そのセキュリティを強固にすることはできる．しかし，部内者からの攻撃を防ぐことは困難である．なぜなら，部内者はそのシステムやデータへのアクセスが許された正規ユーザであるため，万一，彼らが悪意を持っていた場合にはシステムが悪用され内部のデータが改竄・消去されてしまう．また，悪意は無いにしても操作ミスや過失により同様の結果が起こる危険がある．そのため，部外者はもちろんのこと，部内者によるデータの改竄や破壊も防止できるシステムの構築は非常に重要である．

我々がワンタイムライトオンリー型データ管理方式の一形態として提案した電子カルテ管理システム[1]では，データに修正（追記）を行う際に，データそのものを上書き更新するのではなく，修正差分情報を付加していくことで電子カルテにワンタイムライトオンリーの性質を与えていた．本稿では，このシステムを通常のデータ（ワンタイムライトオンリー型以外のデータ）管理に応用することにより，部内者が過失により，または，故意にデータを破壊することを防止できるシステムを提案する．

2．ダブルチェック型文書管理システム

2.1 システム構成

図1は，本提案システムの構成を表している．システムはデータベースを持ち，文書を保存・管理する．システムに登録している複数のユーザおよびスーパバイザが，ネットワークを介してシステムに接続しており，システムが管理する文書へのアクセスが可能である．ユーザは，文書の閲覧，新規作成，修正（追記）を行う権限を持つ正規ユーザである．スーパバイザは，ユーザによって作成されたデータを定期的に検閲し，承認する立場にある正規ユーザである．ユーザによる文書の修正は，修正前の文書を修正後の文書で上書き更新するのではなく，修正前の文書に修正差分データを付加する形で行われる．すなわち，文書に対する修正は追記の形でしか行えない．スーパバイザは，ユーザが記述した新規作成文書および修正差分データを定期的に検閲する．問題が無いと判断した場合は，修正差分データを反映させた修正後の文書によって修正前の文書を上書き更新する．逆に問題があると判断した場合には，問題のある修正差分データを破棄するなどして文書を不当な修正・変更から守ることが可能である．システムによって行われる署名検証やデータの順序性の確認など構造的なチェックに加えて，スーパバイザによって意味的な検閲が行われるダブルチェック機能により，ユーザの悪意あるいは不注意による文書データの破壊が防止される．なお，本稿では簡単のため文書という言葉を用いているが，これは特にテキスト等のドキュメントファイルを指すのではなく，様々なデータファイルを意味するものである．

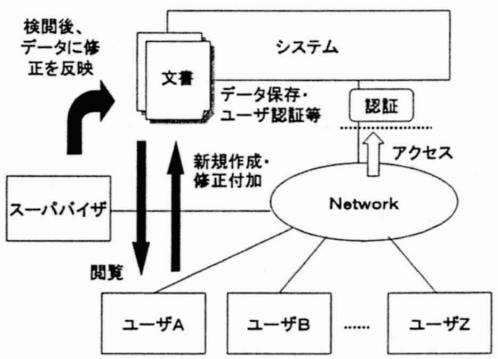

図1　システム構成

システムと各ユーザおよびスーパバイザは，公開鍵暗号の秘密鍵と公開鍵のペアを持っているものとする．システムは自身に登録している全てのユーザおよびスーパバイザの公開鍵を管理する．以下，システムの秘密鍵を Q，公開鍵を P とし，ユーザ X の秘密鍵を q_x，公開鍵を p_x，スーパバイザの秘密鍵を Q_S，公開鍵を P_S とする．システムと各ユーザおよびスーパバイザは，自分の秘密鍵を用いて任意のデータにディジタル署名を施すことができる．各ユーザおよびスーパバイザの署名は，端末（例えばユーザのパソコン内）にて行われる．以下，データ D を秘密鍵 q_x で署名したものを $q_x(D)$と記す．

ユーザおよびスーパバイザがシステムにアクセスする際には，必ずシステムによるユーザ認証およびアクセス制御が行われる．すわなち，システムに登録されていない者はもちろん，システムに登録されているユーザおよびスーパバイザも権限の無い文書にはアクセスできない．なお，本システムにおいてはユーザ認証の方式は特に問わない．パスワード，バイオメトリクスなど必要に応じた方式を採用すればよいが，システムにおける認証方式は十分なセキュリティとフレキシビリティを有するものを用いることを前提としている．

2.2 文書データ

本システムが管理するn番目の文書（以下，文書nとする）を$D_n^{v,x}$と表記する．ここでv，xは文書の現在のバージョンを表す情報であり，$D_n^{v,x}$と共にシステムが保持する．バージョン情報v,xは，vがスーパバイザによる上書きの回数を表し，xがユーザによる修正の回数を表す数値である．

ユーザが文書nを読み出し，これに修正（追記）を行ったとする．その際にユーザが記述した修正差分データを$I_n^{v,x}$と表記する（文書$D_n^{v,x}$に対する修正差分データは$I_n^{v,x+1}$となる）．ユーザは修正差分データ$I_n^{v,x+1}$に署名を付し，システムに送信する．システムは署名検証後，文書$D_n^{v,x}$に$I_n^{v,x+1}$を付加したデータを，新たな文書$D_n^{v,x+1}$としてデータベースに登録する．同時にxをインクリメントしておく．文書の新規作成の場合は，ユーザによって記述されるデータは$I_n^{0,0}$となる．ユーザは$I_n^{0,0}$に署名を付し，システムに送信する．システムは署名検証後，データ$I_n^{0,0}$を文書$D_n^{0,0}$としてデータベースに登録する．

一方，スーパバイザは定期的に文書$D_n^{v,x}$を検閲する．不審な点が無ければ，ユーザの修正を反映させたデータ$I_n^{v+1,0}$を作成して署名を付し，システムに送信する．システムは署名検証後，送られてきたデータ$I_n^{v+1,0}$を文書$D_n^{v+1,0}$として登録する．この時に初めて旧バージョンの文書$D_n^{v,x}$のデータは破棄される．同時にvをインクリメントし，xを0にリセットする．以降，本稿ではスーパバイザによる文書データの上書きを「修正反映」と呼ぶ．

2.3 文書データに対する処理

文書の新規作成・閲覧・修正（追記）・修正反映の詳細な処理について述べる．図2は，以下の説明においてシステムが保持するデータの変化を表している．なお，図中のSysはシステムを，SVはスーパバイザを表す．

#1 ユーザAが文書nを新規作成するとする．ユーザAはデータ$I_n^{0,0}$を作成し，自らの秘密鍵q_Aを用いて$I_n^{0,0}$に署名をする．その後，データ$I_n^{0,0}$と署名$q_A(I_n^{0,0})$とをシステムに送信して，文書の新規作成を要求する．

#2 システムはユーザAの認証を行い，認証をパスすればデータを受け取る．システムは受け取ったデータと署名の整合性をユーザAの公開鍵p_Aを用いて検証する．その結果が正当であれば，新規作成要求を認める．システムは，受け取った署名$q_A(I_n^{0,0})$に対して更に自らの秘密鍵Qを用いて署名をする．そしてこれら全体を結合した$I_n^{0,0} \| q_A(I_n^{0,0}) \| Q(q_A(I_n^{0,0}))$を新規文書$D_n^{0,0}$としてデータベースに保存する．ここで$\|$はデータの連結を表す．以降，$Q(q_A(I_n^{0,0}))$を$S_n^{0,0}$と記す．

#3 ユーザBが文書nを読む際には，システムに文書nの閲覧を要求する．システムはユーザBを認証し，かつ，文書nへのアクセス権を持つことを確認したのち，文書nの最新バージョン$D_n^{0,0}$をユーザBに送信する．その後，ユーザBが$D_n^{0,0}$を読み，更に$D_n^{0,0}$に対して修正を行いたいとする．ユーザBは修正差分データ$I_n^{0,1}$を作成する．そして，$D_n^{0,0}$中のシステムの署名$S_n^{0,0}$（$=Q(q_A(I_n^{0,0}))$）と$I_n^{0,1}$とを結合したデータ$S_n^{0,0} \| I_n^{0,1}$に署名をする．その後，データ$I_n^{0,1}$と署名$q_B(S_n^{0,0} \| I_n^{0,1})$とを，文書nに対する修正要求と共にシステムへ送信する．

#4 システムはユーザBの認証後にデータを受け取る．そしてユーザBの署名を検証し，データ$I_n^{0,1}$が確かにユーザBによるものであること，および，$D_n^{0,0}$に対する修正であることを確認する．問題が無ければ，修正要求を認める．システムは，ユーザBの署名$q_B(S_n^{0,0} \| I_n^{0,1})$に対して更に署名$Q(q_B(S_n^{0,0} \| I_n^{0,1}))$を生成する（以降，$Q(q_B(S_n^{0,0} \| I_n^{0,1}))$を$S_n^{1,0}$と記す）．そして，$D_n^{0,0} \| I_n^{0,1} \| q_B(S_n^{0,0} \| I_n^{0,1}) \| S_n^{0,1}$を文書$D_n^{0,1}$としてデータベースに登録する．

#5 さらに文書$D_n^{0,1}$に対してユーザAが修正を加えるとする．#3,4と同様の手順を行うことにより，システムが所持する文書nは，$D_n^{0,1} \| I_n^{0,2} \| q_A(S_n^{0,1} \| I_n^{0,2}) \| Q(q_B(S_n^{0,1} \| I_n^{0,2}))$の形の$D_n^{0,2}$となる．

#6 ここで，スーパバイザが$D_n^{0,2}$を検閲するとする．スーパバイザはシステムに文書nの閲覧を要求し，許可されると文書nの最新バージョンである$D_n^{0,2}$を受け取る．スーパバイザは，データ$I_n^{0,0}$，$I_n^{0,1}$，$I_n^{0,2}$の内容を読み，各データに不審な点が無いか検証する．問題が無ければ，データ$I_n^{0,0}$に修正差分データ$I_n^{0,1}$，$I_n^{0,2}$を反映させて新たなデータ$I_n^{1,0}$を作成し，自分の秘密鍵Q_Sを用いて$I_n^{1,0}$に署名をする．その後，データ$I_n^{1,0}$と署名$Q_S(I_n^{1,0})$とを，修正反映要求と共にシステムへ送信する．

#7 システムはスーパバイザの認証を行い，パスすればデータを受け取る．そして，受け取ったデータと署名の整合性を検証して，正当であれば修正反映を認める．システムは受け取ったスーパバイザの

署名 $Q_S(I_n^{1,0})$に対して，更に署名を付し $Q(Q_S(I_n^{1,0}))$ を生成する．そして，$I_n^{1,0} \| Q_S(I_n^{1,0}) \| Q(Q_S(I_n^{1,0}))$ を文書 $D_n^{1,0}$ として登録する．この時点で旧バージョンである文書 $D_n^{0,2}$ はデータベースから削除される．

#8　以降，文書 $D_n^{1,0}$ に修正および修正反映を行いたい場合は，#3 から同様の手順を繰り返す．

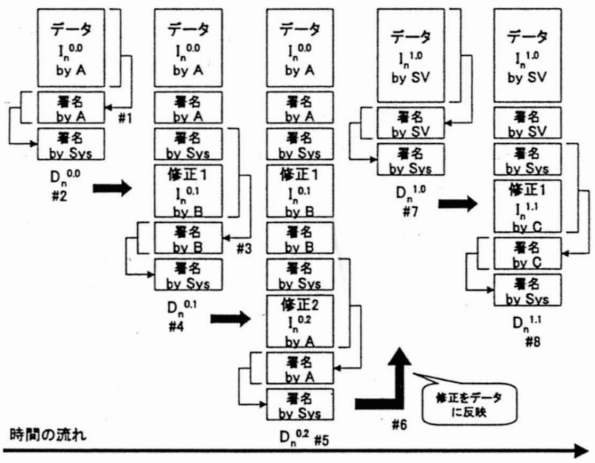

図2　システムが保持する文書の変化

3．考察

3.1　本システムの効果

本システムでは，文書に対する修正（追記）の際に修正前の文書に修正差分データを付加する方式とすることにより，修正履歴がデータとして残るようになっている．かつ，文書中の全てのデータは追記単位でデータ作成者およびシステムの署名によって二重に封印されている．従って，一度システムに登録されたデータを改竄することは，そのデータを記述した者にすら不可能である．すなわち，スーパバイザが修正反映を行うまでの間，文書はワンタイムライトオンリー性を持っており，文書に対する修正履歴は完全に保存される．スーパバイザはユーザによる文書の修正履歴を定期的に検閲し，不審な点が認められた場合には異常が発生する前の文書へと書き戻すことが可能であり，部内者による文書データの破壊に対抗することができる．その際，修正履歴を検証することで異常な修正が行われた経緯の調査や，その修正を行ったユーザの特定が可能であり，業務の品質向上を図ることも期待できる．

3.2　署名方式

本システムの文書データは追記単位でデータ作成者およびシステムによって署名を付与される．その際，署名対象となるデータは，ユーザにおいては自らが記入した修正差分データ $I_n^{v,x}$ とその直前のシステムによる署名データのみであり，システムにおいては修正差分データを記入したユーザの署名データのみである．従って，一度の署名生成の処理コストや署名サイズを抑えることができる．

上記のように生成された署名は，階層的に文書全体をカバーしている．すなわち，文書に付された署名を新しい順に，階層的に検証を繰り返すことで文書全体の改竄の有無を確認することが可能である．すなわち，本署名方式は，ヒステリシス署名[2]の一種と捉えることもできる．

本システムが管理する文書を改竄して，かつそれを検出されないようにするには，システムの秘密鍵に加えて改竄部分以降に付加されているデータを作成した全ユーザの秘密鍵を盗み出して，改竄部分以降の署名を全て偽造しなければならない．文書に対して多くの修正（追記）が加えられているほど，その文書を改竄する際の手間と難度が増すことになる．

3.3　本システムの応用

本稿では，文書の検閲を行う立場にあるスーパバイザを置いて説明したが，実際にはユーザがスーパバイザを兼任しても構わない．

対象システムがデータベースなどの場合は，データの検索が頻繁に起こる．その際，検索要求が来るたびに毎回，データと修正情報から最新データを形成して検索する方式では効率が悪い．このような場合には，データを常に最新情報に上書きすることにし，修正前のデータに戻すための情報を付加していく方式とすればよい．

本システムを拡張して，文書のバージョン管理に特化したシステムを構築することも可能であろう．例えば，本システムにおけるスーパバイザによる修正反映を文書のバージョン確定とみなし，最新バージョン v として上書き保存すると共に，旧バージョン v-1 に書き戻すための差分データを別に保存しておく方式が考えられる．

4．本システムの拡張

4.1　署名交差による改竄困難性向上

3.2 節で述べたように，本システムにおいては文書に多くの修正（追記）が加えられているほど，その文書を改竄する際の手間と難度が増すことになる．しかし，文書に対する修正が頻繁に行われるとは限らない．そして，スーパバイザによる検閲後の修正反映が行われた時点で，文書は上書き更新され過去の修正差分データは削除される．よって，付加された修正差分データが少ない文書も

多くなると思われる．

そこで，異なった文書間においても署名を交換（文献[3]ではこれを「署名交差」と呼んでいる）して利用し合うように拡張することで，文書の改竄をより困難にし，システム全体の信頼性を向上させることができる．図3は，システムが文書2に対する署名生成の際に，文書1の署名データも利用することによって，文書1の署名が文書2の署名に取り込まれる例を示している．

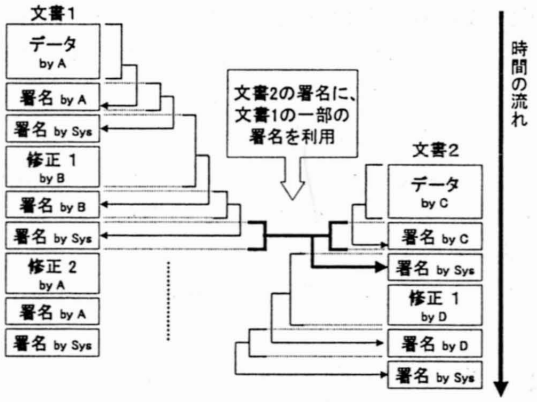

図3　署名の交差

図3の例において，文書1の「修正 1」部分を完全に（改竄が後から露見しないように）改竄したい場合，文書 1 中の「修正 1」以降に付加された署名をすべて偽造することはもちろん，更に文書2における新規データに対する「署名 by Sys」およびそれ以降の署名も全て偽造しなくてはならない．このように，文書1の修正差分データの数が少ない場合でも文書の改竄を困難にすることができる．図3は2文書間の署名交差の例だが，同様の仕組みで図4のように多数の文書間で幾重にも署名の交差を行うようにすると，1 つの文書を改竄するために同時に多数の文書の署名を改竄しなくてはならず，文書の改竄困難性は飛躍的に増大する．

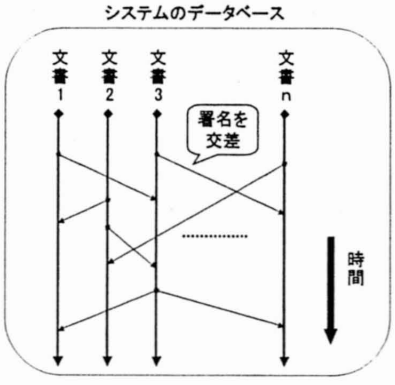

図4　多数の文書間で署名を交差する方式

図4は1つシステム内の異なる文書の署名を交差させる例であるが，更にこれを拡張して，異なるシステム間の文書の署名を交差させる方法も考えられる（図5）．

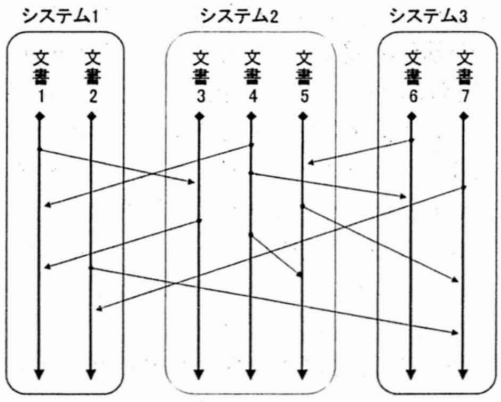

図5　複数のシステム間における署名交差

複数システム間で署名交差を行う利点は，文書の改竄困難性の向上にとどまらない．例えば図5において，システム1および文書2の作成・追記に関わった全てのデータ作成者の秘密鍵が漏洩することにより，文書2が信頼できなくなったとする．そのような場合でも，システム3内の文書7が信頼できるならば，文書2から文書7に署名が渡された時点（文献[3]ではこれを信頼ポイントと呼んでいる）までの文書2の信頼性を，文書7に渡された署名データを利用して確認することができる．

4.2　文書の復旧

本システムにおいては，文書を完全に（改竄が後から露見しないように）改竄するには，システムの秘密鍵の他に改竄部分以降の修正差分データを作成した全ユーザの秘密鍵が必要になる．従って，システムがクラッカーの侵入を許すか，あるいはシステムの管理者が悪意を持っていたとしても文書を完全に改竄することはできない．しかし，システムが悪意を持つ者の手に落ちた場合，システム内の文書データを消去されるという脅威が残っている．そこで，破壊された文書を復旧する手段が不可欠となる．文書の復旧を可能にするために，図5のシステムに改良を加え，図6のようなシステムを構築するとよいと思われる．

図6において，システム1は自身の管理する文書1のバックアップデータをn-1個のデータ片2〜nに分割し，それぞれに署名を付す．そして，各データ片と対応する署名とをシステム2〜nへ配布する．システム2〜nは受け取ったデータ片と署名を自身が管理する文書 2〜n の

末尾に付加して，更に署名を施す．何らかの理由で文書1の破損が認められた場合には，システム2～nはそれぞれに分配されたデータ片をシステム1に提出する．システム1は送られてきたデータ片を結合してバックアップデータを復元し，文書1を復旧することができる．秘密分散の技法[5]を適切に用いることで，各データ片から文書1の情報が漏洩することを防ぐことや，n-1個すべてのデータ片が集まらなくてもバックアップデータを復元できるようにすることも可能であると思われる．

分配されたバックアップデータ片2～nは，それぞれが文書2～nに対する追記データという形になっている．従って，本バックアップ方式は，単なるデータの多重化[4]にとどまらず，自文書の他文書への（追記データとしての）埋め込み，および，署名交差の側面もあわせ持っている．本バックアップ方式によって，複数のシステムが同時にクラッキングされない限り，文書データの復旧が可能となる．

バックアップデータ配布の時期は任意であるが，データ量やデータの重要性を考慮すると，スーパバイザによって文書の修正反映が行われた直後がよいだろう．ここで，スーパバイザによって文書の修正反映が行われると，その文書が今まで保持していた他文書のバックアップデータ片は削除されてしまうことになる．従って，文書iの修正反映が行われた際には，システムiは文書iのバックアップデータ片を他のシステムに送信すると共に，今まで文書iが保持していた他文書のバックアップデータ片を他システムから送信し直してもらう必要がある．こうすることで，修正反映直後の文書iにはその時点で多数のバックアップデータ片が追記データとして付加されることになる．すなわち，修正反映直後の文書データに対する改竄困難性も高まる．

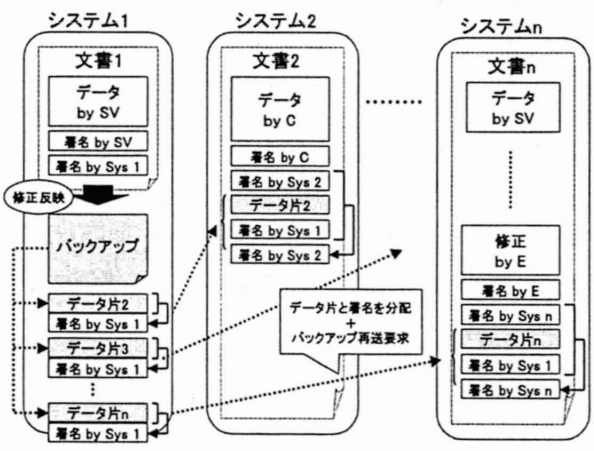

図6　文書のバックアップ

5．まとめ

組織の部外者はもちろん，部内者が過失により，または故意に組織内のデータを破壊することを防止できるシステムを提案した．本システムが管理する文書中のデータは，必ず追記単位でデータ作成者およびシステムの署名が二重に付されており，データを記述した本人にさえ後から改竄することはできない．スーパバイザによって文書に修正反映が行われるまでは，文書はワンタイムライオンリー性を持つので修正履歴が完全に残っており，不正な修正が発見された場合は，その修正を却下することで文書を守ることができる．

本システムを拡張することにより，更なる改竄困難性の向上や，システムの管理者による不正なデータ破壊をも防止することができる．また，データのバージョン管理を行うシステムへの応用も可能であると思われる．

参考文献

[1] 原田篤史，西垣正勝，曽我正和，田窪昭夫：電子カルテの管理方式，コンピュータセキュリティシンポジウム 2001 論文集，2001年10月．

[2] 松本勉，岩村充，佐々木良一，松本武：暗号ブレイク対応電子署名アリバイ実現機構（その1）－コンセプトと概要－，情報処理学会研究報告，CSEC-2000-8，pp.13-17，2000年3月．

[3] 州崎誠一，宮崎邦彦，宝木和夫，松本勉：暗号ブレイク対応電子署名アリバイ実現機構（その2）－詳細方式－，情報処理学会研究報告，CSEC-2000-8，pp.19-24，2000年3月．

[4] 南谷崇：フォールトトレラントコンピュータ，オーム社，1991年．

[5] A.Shamir :How to share a secret, Commun. ACM, 22, 11, pp.612-613, 1979.