8G-03 医用電子文書管理システムの提案

藤川真樹* 八巻睦子* 中村雅一*

綜合警備保障株式会社

1 はじめに

近年、医療データベースの構築と医療事務の効率化を目的として、医用文書の電子化および診療録・医用画像等の電子媒体への保存が進められており、そのための共通規格やガイドラインが策定されている [1, 2, 3, 4]。電子カルテ (Medical Markup Language) [5, 6] は、その代表的な例である。

一方、電子化された医用文書は、第三者のなりすましによる改ざん (確定された情報に対する書き換え、消去) の脅威にさらされがちである。このため電子カルテを取扱っている医療機関の中には、確定済みの電子カルテに対して商用 TSP(タイムスタンプ・プロバイダ) からタイムスタンプを取得しているところもある [7, 10]。

さて、医用文書のうちカルテや看護日誌にはよく「追記」が発生する。「追記」とは、ある患者に対する一連の処置 (診断→処置指示→処置終了) が確定するまでの間、当該文書に患者の容体や手術後の経過等を記載することである。この追記は、外来患者よりも入院患者の方が回数は多く、入院患者でも ICU で集中治療を受けている患者の場合はより多くなる。

このような医用文書に対して、追記が発生するたびに商用 TSP を利用した場合、文書 1 枚あたりのセキュリティ・コストは非常に高くなる。このため、コストやインフラ整備の問題から医用文書の電子化が進んでいない中小規模の病院が、現状の電子カルテシステムや商用 TSP をそのまま利用することは難しく、結果として中小規模病院における医用文書の電子化が遅れるものと思われる。

そこで本稿では、このような問題を解決するために、中小規模病院向けの電子文書管理システムを提案する。これは、カルテや看護日誌等のように追記の多い電子医用文書を改ざんから保護するとともに、これらの文書に対して商用 TSP を利用しながら低コストでタイムスタンプを取得するシステムである。

* **A Proposal of The Medical Electronic Document Safekeeping System** Masaki Fujikawa, Mutsuko Yamaki, Masaichi Nakamura: Sogo Keibi Hosho Co.,Ltd. Technical Research Laboratories. 3-22, Kinshi, Sumida-Ku, Tokyo 130-0013 JAPAN

2 提案するシステム

2.1 コンセプト

我々は、以下のコンセプトをもとにシステムを構築している。

◎ 厚生省の定める共通規格やガイドライン [1, 2, 3, 4] に沿ったシステムであること。

◎ カルテや看護日誌を含むすべての電子医用文書にも対応できること。

◎ 第三者のなりすましによる、確定済みの情報に対する虚偽入力、書き換え、消去を、商用 TSP を利用しながら低コストで防止できること。(なお、正当な利用者による不正の意を持ったカルテ等の捏造は違法行為であるため、本システムの対象外とする。)

◎ 「単純なタイムスタンプ・プロトコル」、「リンキング・プロトコル」、「分散プロトコル」等のいずれのタイムスタンプ・プロトコル [8] にも対応できること。

◎ MML(Medical Markup Language) に対応できるなど、システム間のデータ互換性を確保し、相互利用性にすぐれたシステムとすること。

2.2 システム構成

本システムでは、**Fig.1** に示すように病院内に LAN を設置し、電子文書を取扱うデータセンタ (DC) および DC にアクセスできる専用端末 Term_$j(j = 1, 2, .., n)$ を LAN に接続する。DC は耐タンパ機能および電子文書のバックアップ機能を備えており、Term_j は耐タンパ機能および指紋照合装置等の本人認証機能を備えているとする。

DC は FW(Fire Wall) 経由でインターネットに接続されている。また商用 TSP もインターネットに接続しており、タイムスタンプ・リクエストやタイムスタンプの送受信はすべてインターネットを介して行うものとする。なお、商用 TSP は信頼できるものとする。

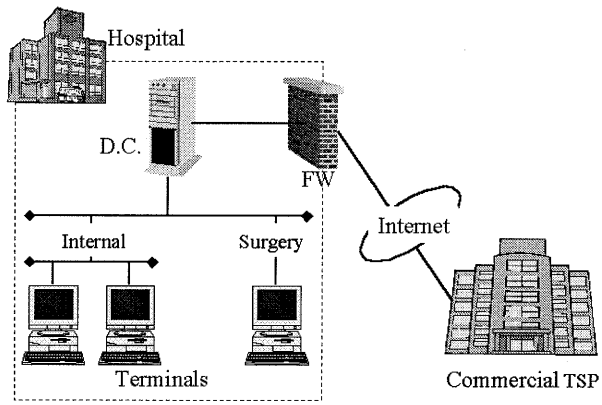

Fig.1 Composition of the system

2.3 電子文書のフォーマット

本システムではすべての医用文書を取り扱えることをコンセプトとしているが、ここでは説明のため看護日誌 (看護記録) を例に取り、電子文書のフォーマットを定義する。

Fig.2 に示すように、電子文書 M をいくつかの属性情報 $Att\_i_t(i = 1, 2, .., n, t = time)$ に分割する。
看護日誌の場合、患者の属性や個別的な情報を記録する「基礎 (個人) 情報」、患者の個別的なケアの計画を記載する「看護計画」、患者の心身の機能・能力を妨げるような事項を記載する「問題リスト」、患者の治療・処置・ケア・看護実践を記載する「経過記録」、患者の経過および情報を簡単にまとめる「看護サマリー」に分割することができる。

これらの属性情報は、記載内容に応じてさらに細かい属性情報に分割できるようにする。看護日誌では、上述の「経過記録」を文章で記録する「叙述的記録」と一覧表で記録する「フローシート」に分割することができる。

さらに、それぞれの属性情報に患者を特定できる ID を付与し、患者と属性情報とを結び付けられるようにする。

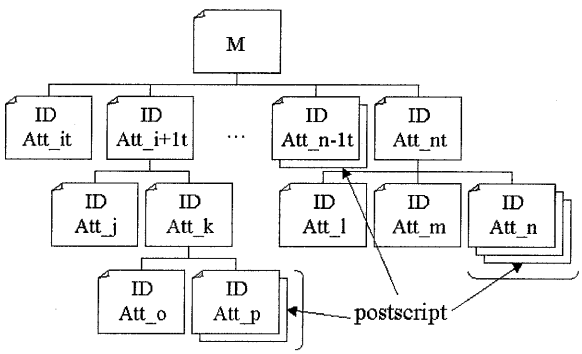

Fig.2 Electronic document format

2.4 電子文書の起票

さて、Term_j によって患者 A の看護日誌 $Diary\_A$ を時刻 t に起票したとする。説明のため、属性情報 Att_i_t の係数 i は 1 から 5 までとする。

$$Diary\_A_t \quad = \quad (Att\_1_t, Att\_2_t, Att\_4_t, Att\_5_t, Info_t)$$

ここでは、Att_1_t,Att_2_t, Att_4_t,Att_5_tが起票された属性情報である。また $Info_t$ は起票者を識別する情報であり、氏名や所属等がこれにあたる。

次に DC は、$Diary\_A_t$ と初期値 (ランダムデータ $random$) からハッシュ値 h_A_t を計算する。

$$h\_A_t \quad = \quad H(Diary\_A_t, random)$$

h_A_tを、起票後の「最新のハッシュ値」と呼ぶことにする。DC は、h_A_t と $Diary\_A_t$, $random$ を一旦 DC 内に保存し、追記の発生またはタイムスタンプ・リクエストを生成するまでの間待機する。

2.5 電子文書の追記

2.5.1 属性情報の更新

Term_j によって患者 A の看護日誌 $Diary\_A_t$ に対し、時刻 $t+1$ に追記 (属性情報の更新) を行ったとする。ここでは、 Att_4_{t+1} , Att_5_{t+1} が更新された属性情報である。

$Diary\_A_{t+1}$

$= (Att\_1_t, Att\_2_t, Att\_4_{t+1}, Att\_5_{t+1}, Info_{t+1})$

$Info_{t+1}$ は追記者を識別する情報であり、氏名や所属等がこれにあたる。

次に DC は、$Diary\_A_{t+1}$ とこれまでの最新のハッシュ値 h_A_t からハッシュ値 h_A_{t+1} を計算する。h_A_t を連鎖させることにより、あとで更新の事実を正しく確認できると同時に、確定後の電子文書に対する改ざんを発生させにくくさせている。

$$h\_A_{t+1} \quad = \quad H(Diary\_A_{t+1}, h\_A_t)$$

h_A_{t+1}を、追記後の「最新のハッシュ値」と呼ぶことにする。DC は、h_A_{t+1} と $Diary\_A_{t+1}$ を DC 内に保存し、次の追記の発生またはタイムスタンプ・リクエストを生成するまでの間待機する。

2.5.2 属性情報の追加

Term_j によって患者 A の看護日誌 $Diary\_A_{t+1}$ に対し、時刻 $t+2$ に追記 (属性の追加) を行ったとする。ここでは、Att_3_{t+2} が追加された属性情報である。

$Diary\_A_{t+2} =$
$(Att\_1_t, Att\_2_t, Att\_3_{t+2}, Att\_4_{t+1}, Att\_5_{t+1}, Info_{t+2})$

$Info_{t+2}$ は追記者を識別する情報である。次に DC は、$Diary\_A_{t+2}$ とこれまでの最新のハッシュ値 h_A_{t+1} からハッシュ値 h_A_{t+2} を計算する。h_A_{t+1}を連鎖させることにより、あとで追加の事実を正しく確認できると同時に、確定後の電子文書に対する改ざんを発生させにくくさせている。

$$h\_A_{t+2} \quad = \quad H(Diary\_A_{t+2}, h\_A_{t+1})$$

h_A_{t+2}を、追記後の「最新のハッシュ値」と呼ぶことにする。DC は、h_A_{t+2} と $Diary\_A_{t+2}$ を DC 内に保存し、次の追記の発生またはタイムスタンプ・リクエストを生成するまでの間待機する。

2.5.3 属性情報の削除

通常、一度記載した看護日誌の内容を削除することはないため、本システムでも属性情報の削除は行わないことにする。

但し、属性情報の記載内容に不備がありそれを訂正する際には、当該属性情報に対し「属性情報の更新」処理を行うことにする。これにより、訂正前と訂正後の2つの属性情報を DC 内に保管できるため、あとで訂正の事実を正しく確認することができる。

2.6 タイムスタンプの取得

ここからは、タイムスタンプ取得までの一連の流れについて述べる。

DC は、あらかじめタイムスタンプ・リクエストを生成するための時間間隔 $Tspan$ を設定する。そして、$Tspan$ 内に追記のあった看護日誌から生成した「最新のハッシュ値」を連結させ、ハッシュ値 RH_Tspan を生成する。

ここでは、**Fig.3** に示すように $Tspan$ 内に 3 人の患者の看護日誌 $Diary\_A$、$Diary\_B$、$Diary_C$がそれぞれ l, m, n 回追記され、それによって最新のハッシュ値 h_Al, h_Bm, h_Cn が生成されたとする。

$$RH\_Tspan \quad = \quad H(h\_Al, h\_Bm, h\_Cn)$$

作成された RH_Tspan は、タイムスタンプ・リクエストを構成する要素になる。次に DC は、利用している商用 TSP の要求するタイムスタンプ・リクエストのプロトコルに応じて、タイムスタンプ・リクエスト TSRを生成し、商用 TSP に送信する。

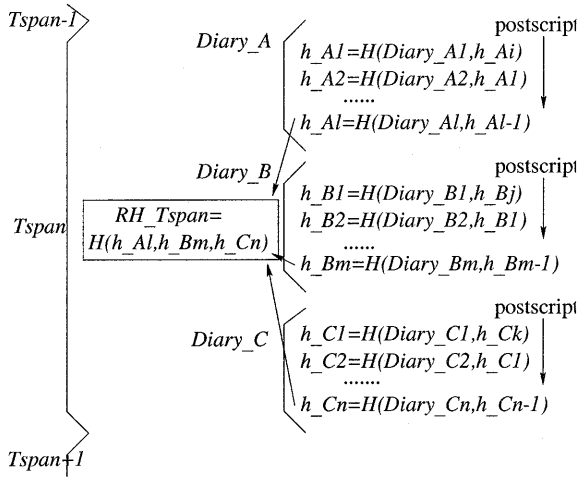

Fig.3 Timestamp request

商用 TSP では、DC から受け取ったタイムスタンプ・リクエスト TSR からタイムスタンプ TS を作成し、自 TSP 内に保存するとともに DC に送信する。TS を受信した DC は、TS を DC 内に保存する。

ここで試みに、**Fig.4** に示すようなツリー構造のリンキング・プロトコル [9] を用いてタイムスタンプを生成してみる。

＜前提＞

- 商用 TSP は、あらかじめリンク情報を生成するためのラウンドを設定する。ここでは1ラウンド中に最大8個のハッシュ値を受付可能とする。
- 説明のため、商用 TSP は受信した TSR を第 i ラウンドの H_1 として組み込むものとする。

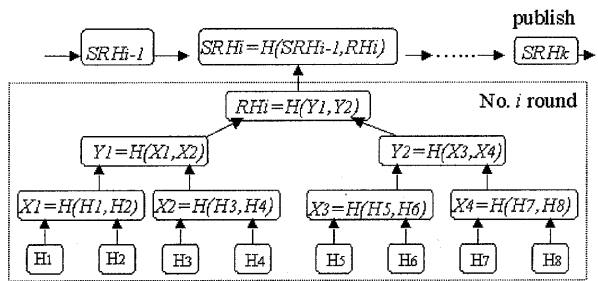

Fig.4 Timestamp(Linking Protocol)

＜タイムスタンプの生成＞

1. 第 i ラウンドが終了した後、第 i ラウンドで受け取ったハッシュ値 $(H_1, ..., H_8)$ を用いて、ハッシュ値

$$X_1 = H(H_1, H_2), X_2 = H(H_3, H_4),$$
$$X_3 = H(H_5, H_6), X_4 = H(H_7, H_8),$$

を計算する。

2. (X_1, X_2, X_3, X_4) を用いて次の計算を行う。

$$Y_1 = H(X_1, X_2), Y_2 = H(X_3, X_4)$$

3. $RH_i = H(Y_1, Y_2)$ を計算する。

4. 第 $i-1$ ラウンドの SRH_{i-1} と RH_i を結合・ハッシュ化し、第 i ラウンドの SRH_i を生成する。なお TSP はこれらのデータを保管する。

5. ハッシュ値 $H_1(= TSR)$ に対するタイムスタンプとして (H_2, X_2, Y_2, RH_i) を DC に送付する。

6. 定期的に、SRH とその時刻情報を公開する。

2.7 完全性の検証

本システムでは、DC によるセルフチェック機能を設け、自らの管理する電子文書に対して、完全性の検証を行わせるようにする。また、DC とは別に検証者 (検証システム) を立てて、タイムスタンプの検証および DC の監査ができるようにする。この場合 DC は検証者に対し、検証に必要な情報を提示できるものとする。

タイムスタンプの検証は、利用する商用 TSP のタイムスタンプ・プロトコルに依存する。このため検証者 (検証システム) は、利用するプロトコルに応じた検証方法を取る必要がある。

ここで試みに、ツリー構造のリンキング・プロトコル [9] を用いた場合の DC および検証者 (検証システム) による完全性の検証を行う。

＜ DC によるセルフチェック＞

1. DC は任意の時間間隔 $Tspan_x$ に着目し、**Fig.5** に示すように $Tspan_x$ において i 回追記のあった患者 X の看護日誌から再度「最新のハッシュ値」h_X_i' を生成し、DC に保存している h_X_i と比較する。これにより、$Tspan_x$ における患者 X の追記看護日誌に関して偽造・改ざんの有無を検証することができる。

2. 1. で生成した患者ごとの最新のハッシュ値を連結させて、**Fig.6** に示すように RH_Tspan_x' を生成し、DC に保存している RH_Tspan_x と比較する。これにより、$Tspan_x$ における、病院全体の追記のあった看護日誌に関して偽造・改ざんの有無を検証することができる。

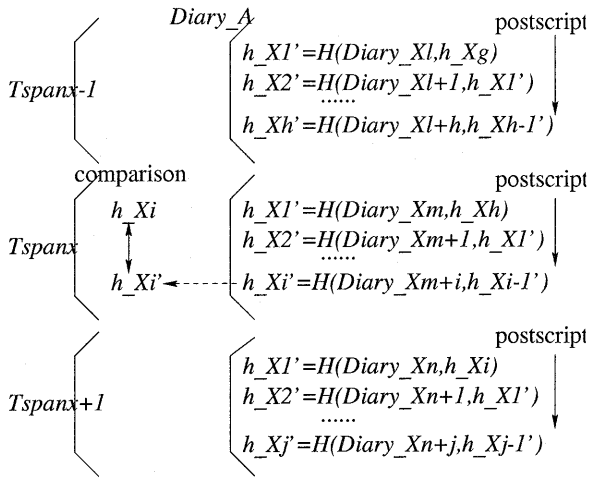

Fig.5 Self-Inspection by DC

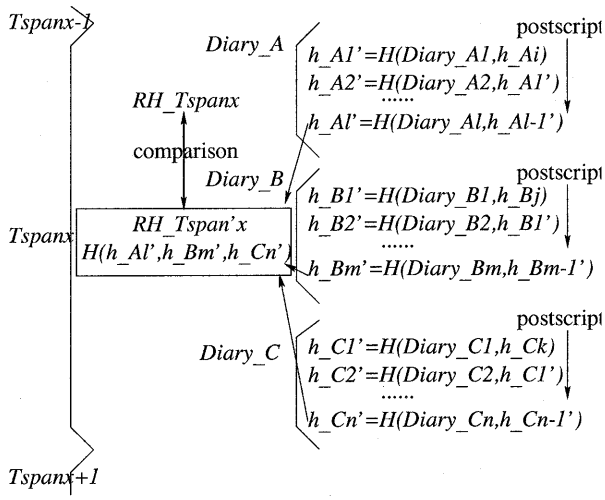

Fig.6 Timestamp inspection

＜検証者 (システム) による検証＞

1. はじめに、任意の $Tspan_x$ を対象に発行されたタイムスタンプ (H_2, X_2, Y_2, RH_i) と DC が保存しているタイムスタンプ・リクエスト $H_1(= RH\_Tspan_x)$ から RH_i' を生成し、タイムスタンプの一部である RH_i と比較する。

2. TSP が保管している SRH_{i-1} と生成した RH_i から SRH_i を生成する。そして、TSP が保管している $(RH_{i+1}, ..., RH_k)$ から SRH_k を生成し、公開されている SRH_k と比較して一致することを確認する。これにより、病院全体の追記のあった看護日誌が $Tspan_x$ において存在していたことを検証することができる。

以上により、DC が管理している電子文書の偽造・改ざんの有無、および当該文書が $Tspan_x$ において存在していたことを検証することができる。

3 考察

3.1 セキュリティ・コスト

中小規模病院において、患者一人の看護日誌に支払えるセキュリティ・コストは、筆者らがヒアリングした限りでは売上の数%以下が妥当であると考える。

さて、ベッド数400床の中小規模病院において、患者一人当たり1日に3回看護日誌の追記が発生すると仮定した場合、1ヵ月の病院全体の総追記回数は36000回となる。

試みにSurety社のシステム [10, 11] を用いて、患者一人の看護日誌に対するセキュリティ・コストを計算してみる。Surety社では、タイムスタンプ・リクエストの多い顧客からのコスト削減の要求に対して、タイムスタンプ・リクエスト生成サーバを顧客側に設置し、そこから自TSPにタイムスタンプ・リクエストを送信させている。このサーバは、**Fig.7** に示すようなツリー構造のリンキング・プロトコルにより、ラウンド内に受信したいくつかのタイムスタンプ・リクエストから1つのタイムスタンプ・リクエストを生成している。

ここでひと月に発生するラウンド数を $Round$、1ラウンドに受信できるリクエスト数を Rとし、タイムスタンプ1回の利用料金を y円とすると患者一人の看護日誌に対するコスト $cost$ は、

$$cost = \frac{y(36000)}{400 \cdot R \cdot Round}$$

となる(パディング・データの挿入を考慮していないことに注意)。セキュリティ・コストを数%以下に抑えるためには、Rと $Round$ の2つのパラメータを設定する必要がある。

一方、本システムにおける患者一人の看護日誌に対するコストは、1ヵ月の $Tspan$ の数を $Total$とした場合、

$$cost = \frac{y \cdot Total}{400}$$

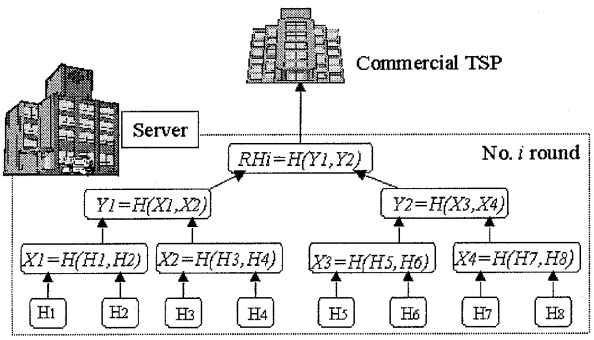

Fig.7 Timestamp request server

となり、追記回数に影響を受けないだけでなく、1つのパラメータ $Total$の設定だけでセキュリティ・コストを調整することができる。

3.2 ハッシュ関数危殆時の対応

ある時点 X_dayにおいて、本システムで使用しているハッシュ関数 H_old が危殆化した場合、X_dayまでに発行されたタイムスタンプも同時に危殆化する。また、DC内で「最新のハッシュ値」を生成する過程において、ハッシュ値を連鎖させる途中に偽の追記看護日誌を挿入される恐れもある。

本システムでは、ハッシュ関数危殆時の対応として、**Fig.8** に示す処理を行う。

1. 危殆化していない安全なハッシュ関数 H_safe を用いてすべての患者の看護日誌から再度「最新のハッシュ値」h_Xnewを生成する。

2. 各患者の看護日誌の「最新のハッシュ値」を連結させて RHを生成し、これを用いて新しいタイムスタンプの発行を依頼する。

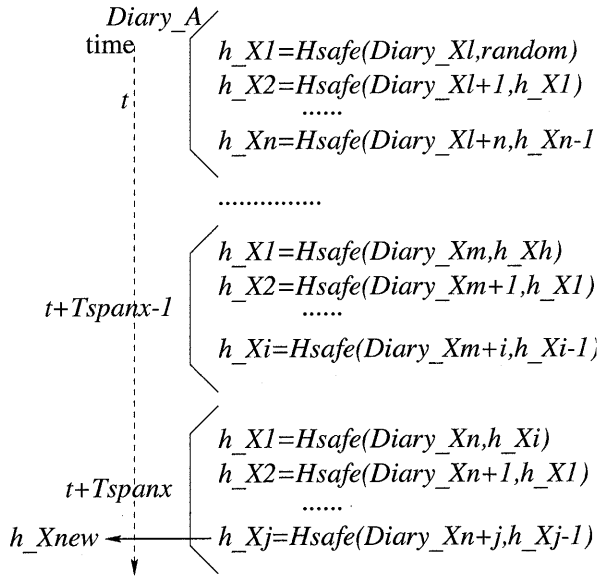

Fig.8 Remake of hash value

3.3 暗号アルゴリズムの更新方法

前節で述べたシステムを構築する際に問題となるのが、どのようにして危殆化していない安全なハッシュ関数 H_safe を低コストで効率よくシステムに組み込むか、ということである。

最近では、このような問題に対する解決方法がいくつか提案されている [12, 13] が、本システムに対して

も Fig.9 に示すようなオープンネットワークを介した暗号アルゴリズムの更新方法を組み込んでおく必要がある。

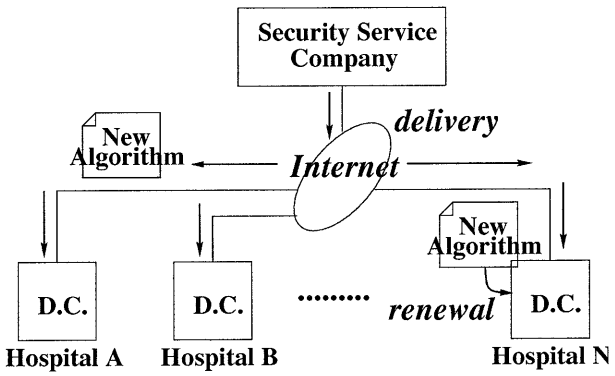

Fig.9 New hash algorithm delivery system

3.4 保存期間を過ぎた電子文書の取扱い

厚生省令第 15 号によると、看護日誌は「療養の給付の担当に関する記録」とされており、保存期間については完結の日から 3 年間と定められているが、保存期間を過ぎた看護日誌の取扱いについては法的な義務付けはなく、各医療機関の裁量に任されているのが現状である。このため本システムでも、保存期間を過ぎた電子文書の廃棄または保存については、各医療機関の方針にゆだねるものとする。ただし、本システムに対して電子文書の廃棄や長期的保存に関する技術的検討を加える必要がある。

4 まとめ

本稿では、中小規模病院に対する医用文書の電子化の促進を目的として、医用電子文書管理システムを提案した。病院で使用されるカルテや看護日誌等の医用文書は追記が多く発生する。本システムは、追記の多い電子文書を改ざんから保護すると同時に、これらの文書に対して商用 TSP を利用しながら低コストでタイムスタンプを取得するものである。

また本システムでは、DC にセルフチェック機能を設け、DC が管理している電子文書に対して独自に偽造・改ざんの有無を検証できるようにするとともに、DC とは別に検証者 (システム) を立てて、商用 TSP が発行したタイムスタンプの検証と DC の監査ができるようにした。

なお、本システムの実用化に際しては、技術的検討を行いつつ、中小規模病院に対する更なるヒアリングを実施する必要がある。

謝辞

本システムの構築にあたり、ヒアリングに御協力頂いた日揮株式会社井用重孝氏、沼田広一氏に感謝致します。

参考文献

[1] “法令に保存義務が規定されている診療録及び診療諸記録の電子媒体による保存に関するガイドライン”, 医情開第 24 号 (1999.11)

[2] “診療録等の電子媒体による保存について”, 健政発第 517 号, 医薬発第 587 号, 保発第 82 号 (1999.4)

[3] “「診療録等の記載方法等について」の一部改正等について”, 保険発第 62 号 (1999.4)

[4] “救急救命処置録の電子媒体による保存について”, 指第 32 号 (1999.4)

[5] 社団法人 日本医療情報学会 電子カルテ研究会, http://www.seagaia.org/sgmeeting/default.html

[6] “MML(Medical Markup Language) ver2.21 final ”, http://www.medxml.net/SG.html

[7] BML Co.,Ltd, http://www.bml.co.jp/

[8] 宇根, 松浦, 田倉：“デジタルタイムスタンプ技術の現状と課題”, 日銀金融研究所 (2000.4)

[9] H.Massias and J.J.Quisquater：“Time and Cryptography” Belgian Project TIMESEC Technical Report WP1(March 1997)

[10] NTT データ電子文書証明サービス (Secure Seal), http://210.144.76.11/index2.html

[11] Surety.com Digital Notary Service, http://www.surety.com/index-nn.html

[12] 山田, 宮地, 双紙：“オープンネットワークにおける安全な暗号方式の更新に関する考察”, 情報処理学会論文誌,Vol.41,No.8,pp.2102/2109(2000)

[13] 栃窪, 岡田, 遠藤, 岡本：“リニューアル可能な暗号認証システムの検討”, 情報処理学会論文誌,Vol.41, No.8,pp.2121/2128(2000)

[14] 宇根, 松本：“連鎖型タイムスタンプの検証に用いられる情報の管理”, 情報処理学会 CSS2000 論文集,pp.25/30(2000)

[15] 第 20 回医療情報学連合大会 (第 1 回日本医療情報学会学術大会) 予稿集.