arχves: 公開鍵暗号を用いた研究記録管理・公開・検証システム構築の試み

arχves: Attempt to Construct Archive, Publication and Verification System for Research Logs using Public Key Cryptosystem

加藤 未来 †, 小林 聡 ‡

Miku KATO†, Satoshi KOBAYASHI‡

skoba@ipc.shimane-u.ac.jp‡

島根大学 総合理工学部 †

島根大学 総合情報処理センター ‡

Interdisciplinary Faculty of Science and Engneering, Shimane University†

General Information Processing Center, Shimane University‡

概要

昨今，研究成果や論文の捏造の問題が社会問題となったことは記憶に新しい．改竄が困難な手法によって研究記録が保存されているならば，少なくとも研究記録の検証に関しては大いに助けになるであろう．この際，研究記録の真正性と非改竄性の保証をいかに行なうかが課題となる．本研究では，公開鍵暗号と，文書作成者以外が検印を行なう検印モデルを用いた，研究記録の管理・公開・検証システムの構築を試みた．検印の実現には，文書サーバにおけるユーザの電子署名に加え，検印サーバによる電子署名を用いた．また，本システムは，研究記録を扱うため，情報の公開範囲の柔軟な制御を実現した．

キーワード

研究記録, 公開鍵暗号, タイムスタンプ, 検印モデル, 公開範囲制御

1 はじめに

昨今，研究成果や論文の捏造の問題が社会問題となったことは記憶に新しい．これらの問題の発生を技術的に食い止める事は困難であるが，改竄が困難な手法によって研究記録が保存されているならば，少なくとも研究記録の検証に関しては大いに助けになるであろう．この際，研究記録の真正性と非改竄性の保証をいかに行なうかが課題となる．このような課題に対して，原田らは電子カルテを例に，手法の提案を行なった [1]．また，陳らは XML 文書の時刻認証を伴なった管理を試みている [2]．同様なサービスとして，SNS/blog 機能に電子文書へのタイムスタンプ付加機能を加えた，SNS/blog サービスも存在する[*1]．しかし，これは日本電子公証機構のサービスを利用しているため，比較的高額なサービスとなっている．タイムスタンプを活用したサービスは，DVCS(Data Validation & Certification Server Protocol) や TAP(Trusted Archival Protocol) などに分類される．DVCS は，電子文書のアーカイブは行なわないことを原則としており，対して TAP はアーカイブ

[*1] Synest LaboNote (http://www.labonote.jp)

も行なうことを原則としている．宇根らは各種タイムスタンプによる可用性・安全性に関して報告している [3]．

また，今日，World Wide Web に代表されるインターネット環境は，多くの人にとって重要なコミュニケーション・メディア/コミュニケーション・ツールとなっている．そのようなネットワーク環境を利用し，論文の公刊前に，関連した研究を行なう研究者同士が研究の進め方や成果について，互いに意見交換が可能な環境が提供されることは研究を進める上で大きな利点があると考えられる．

記録，コミュニケーションと並んで，コンピュータが果すことができる重要な役割として，ストレージ機能がある．研究資料や実験試料，あるいはプログラムなどを手軽に公開・参照できる機能が実現できれば，研究者間での研究資料や実験試料などの流通も盛んになり，研究自体が活性化するであろう．

ただし，コミュニケーション機能においても，ストレージ機能においても，研究の遂行途中においては，一般には秘匿したい，あるいは秘匿しなければならない情報あるいはデータも存在する．そのため，研究記録の保存においては情報あるいはデータを公開する範囲を柔軟に制御できなければならない．現在，World Wide Web を用いた SNS/blog サービスにおいても，記事の公開範囲はある程度の制御が可能である．しかし，概ねその設定の自由度は低い．Masui らや江渡らは，QuickML[4] や qwikWeb[5] により，情報にアクセスできる者を柔軟に変更可能することが可能なシステムの構築と運用を試みた．また，情報の公開範囲を柔軟に制御可能なシステムとして，SNS/blog を基盤として，永田らは Enzin を [6]，高井らは ACS を開発し [7]，運用実験を行なった．高田らは，公開 Web-DB と Web-DB 管理システムを分離可能であり，かつきめ細かなアクセス制御が可能な Web-DB 管理システムを構築している [8]．また，一般の SNS/blog にも，記事ごとに公開範囲の設定を可能とするサービスが登場しており[*2]，公開範囲の柔軟な制御の需要が伺える．

本研究では上述のような研究記録などの管理を主目的として，公開範囲を柔軟に制御可能かつ，記録などの真正性および非改竄性を可能な限り保証する，研究記録などの管理・公開・検証を行なうシステム，arχves[*3]の構築を試みた．特に，安価かつ，DVCS でもなくまた TAP でもない，第三のあり方を模索した．

[*2] Media Wagon (株式会社エイミー http://mw.aimy.jp/)

[*3] arχves: Archive system for Research logs by Kato=Shimanuki/Kobayasi Satoshi, and VErification System → ARKSVES. KS を x に，さらに x を類字形の χ に置き換え．

2 モデル

企業における，記録の真正性および非改竄性を保証する方法として，部下が書いた記録 (日報など) に，上司が検印を捺すことが広く行なわれている．このような，記述者とは異なる者が検印あるいはそれに類する行為/操作を行なうモデルを，本論文では「検印モデル」と呼ぶ．検印モデルの例としては，原田らは，公開鍵暗号を用い，記述者が自分自身の秘密鍵で電子署名をすると共に，文書管理システムに持たせた秘密鍵を使った電子署名により検印をするモデルを提案している [1]．本論文においても，この検印モデルを採用した．ただし，原田らの提案においては，文書管理システムが検印を行うが，本論文で述べるシステムにおいては，記事やデータに検印を行なうサーバ (以下「検印サーバ」) と，文書を保管するサーバ (以下「文書サーバ」)」は異なることを想定している．文書サーバは，ユーザが通常，記事を記述する際に用いるサーバとした．

これはタイムスタンプ技術 [9] と類似しているが，文書作成者の真正性を，ユーザ自身の電子署名という形で明示的に示している点が異なる．

本システムは，研究室 ～ 学科程度の規模を単位としての記録の保存を想定しており，過度に分散せず，過度に集約せず，適度に分散しつつネットワークを構成するシステムを想定している．また，検印については，大学規模の組織を越えて，相互に検印を行なうモデルを想定している．このようにシステムを分散することで，導入および管理コストの低減を期待している．

このような，システムの利用サイトが相互に検印を行なう手法により，真正性や非改竄性の保証の強度は，企業が提供するサービスに比べて幾分低くなると思われる．しかし，企業が提供するタイムスタンプ・サービスは高価であったり，利便性に欠ける部分がある．そこで，本論文で示すシステムと，企業が提供するサービスとは使い分け/棲み分けができると考えられる．

また，本システムは，基本的に研究記録を適宜入力していくという利用状況を想定しているため，blog の機能を基本としている．また，研究支援を想定し，研究資料あるいは研究試料 (以下「データ」) のストレージ機能も実装している．

3 機能

3.1 使用したツール等

本システムで使用したツール等を表 1 に示す．

表-1　利用ツール等

開発言語:	Ruby ver. 1.8.6 [*4]
フレームワーク:	Ruby on Rails ver. 1.2.6 [*5]
公開鍵暗号ソフト:	GnuPG ver. 1.4.9 [*6]
ウェブサーバ:	Apache ver. 2.0.6 [*7]
SSL ライブラリ:	OpenSSL ver.0.9.8 [*8]
DBMS:	MySQL ver. 5.0.27 [*9]

表-2　アクセス権限

	管理者	ユーザ	ゲスト	一般
TOP ページ	○	○	○	○
blog リスト	○	○	○	○
グループリスト	○	○	○	
ユーザリスト	○	○	○	
アカウントメニュー	○	○	△	
管理者メニュー	○			

△: アカウント情報の閲覧のみ可能

表-3　メニュー項目

アカウントメニュー	管理者メニュー
アカウント情報の閲覧	ユーザ管理
アカウント情報の変更	グループ管理
管理グループの変更	blog 記事管理
blog タイトルの変更	コメント管理
カテゴリの作成・変更	カテゴリ，blog 名管理
blog 記事の公開範囲の変更	
添付ファイルの削除	

Ruby on Rails は，本システムの基幹となる blog 機能の構築を中心に活用した [10]．電子署名の付与および検証については，GnuPG を活用している．

3.2　利用者範疇

システム利用者は，本システム上に何らかのアカウントを持っている．このシステム利用者はシステム内での権限によって，表 2 に示す,「ゲスト」,「ユーザ」,「管理者」の 3 つに分類される．特にアカウントメニューおよび管理者メニューの項目を表 3 に示す．

「ゲスト」とは，システム内の blog 機能に対する記事の投稿やグループの作成は行なえないが，グループに所属し，記事の閲覧は可能な利用者である．ゲストは，本システムにアクセスした際，認証を通過することにより，

*4 http://www.ruby-lang.org/ja/
*5 http://www.rubyonrails.com/
*6 http://www.gnupg.org/
*7 http://www.apache.org/
*8 http://www.openssl.org/
*9 http://www.mysql.com/

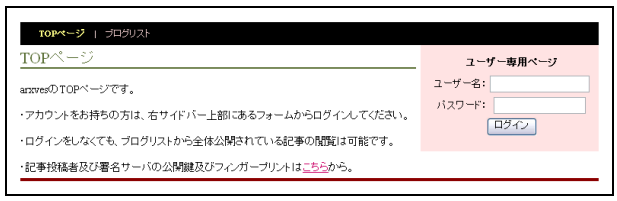

図-1　トップ画面

図-2　記事一覧表示

当該ゲストあるいは当該ゲストが属するグループに対して閲覧が許可されている記事やデータの閲覧が可能となる．

「ユーザー」とは，システム内に自身の blog を作成し，記事を投稿することができる利用者である．また，グループの新規作成や，自身のアカウント情報を修正することができる．ユーザは，自分が書いた記事の各々について，その記事を閲覧可能なグループを複数個指定できる[*10] [*11]．例外として，公開範囲を指定しないことも可能であり，そのような記事の閲覧については，ゲストとしての認証を受けずに可能である．

「管理者」とは，システムの管理者であり，管理者メニューにアクセスできる唯一の利用者である．

本システムにおいては，全てのシステム利用者は，1 つ以上のグループに属している．なお，グループに関する情報の編集は，当該グループを登録したユーザ，あるいは管理者のみが可能である．

3.3　記事の投稿

本システムにアクセスすると，図 1 のようなトップ画面になる．ここでユーザとして認証を受け，ログインすると，図 2 のような，記事の一覧表示画面となる．ここで，投稿ページにジャンプすると，図 3 のような投稿画面になる．

タイトル，本文，カテゴリ，公開範囲の指定を行ない，

*10 グループの作成や，閲覧可能なグループの設定は，管理者によっても可能である．
*11 先述の labonote では，blog 全体の公開範囲は設定可能であるが，個別の記事についての制御はできない．

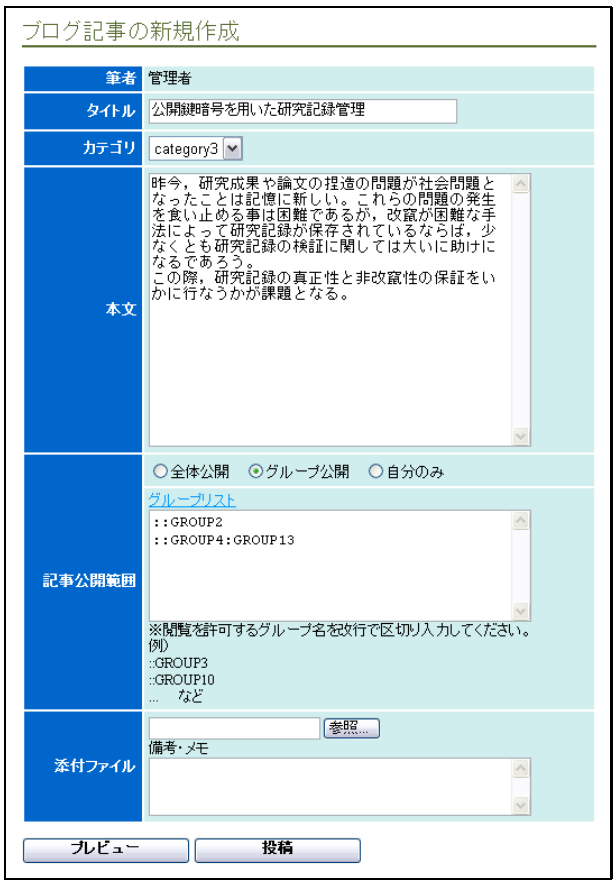

図-3　投稿画面

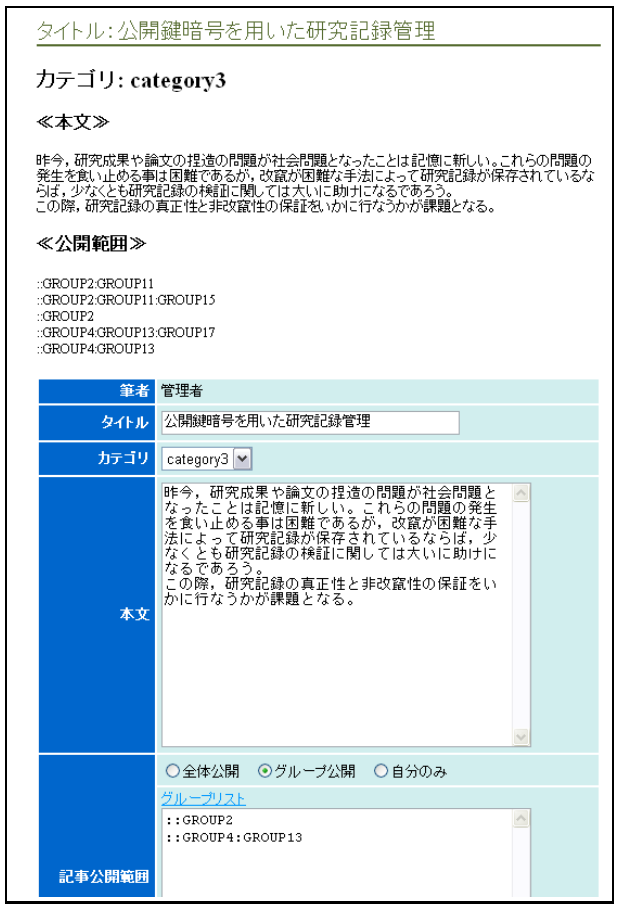

図-4　プレビュー画面

記事を記入した後,「プレビュー」ボタンを押すと，投稿はされず，図 4 に示す，署名前のテキストのチェックおよび公開範囲の確認ができる．なお，カテゴリと公開範囲は，記事の投稿後にも修正可能である．

最後に,「投稿」ボタンをクリックすることで，自動的に各ユーザの秘密鍵による署名後，署名付き記事などが検印サーバに送られ，検印サーバの秘密鍵による署名が行われる．その後，2 重の署名が施された記事などは検印サーバから送り返され，文書サーバに保管される．なお，検印サーバは，自身が行なった署名を，記事 ID および時刻とともに記録している．

また，各記事には，ファイルを添付することが出来る．このファイルについても，記事と同様に二重の電子署名が付加される．添付ファイルのダウンロードは，二重に署名されたままのファイルと，システムが自動的に復号処理を行なったファイルの両方を，記事の詳細画面からダウンロードできるようになっている (図 5)．

なお，現在，投稿後の記事の修正・再編集は認めていないない．記事の削除については，管理者のみが行なえる．添付データの削除は投稿者が行なえるが，削除理由を明記した上で削除可能とする．また添付データが存在していた記録が残るようにしている (図 6)．これは，研究記

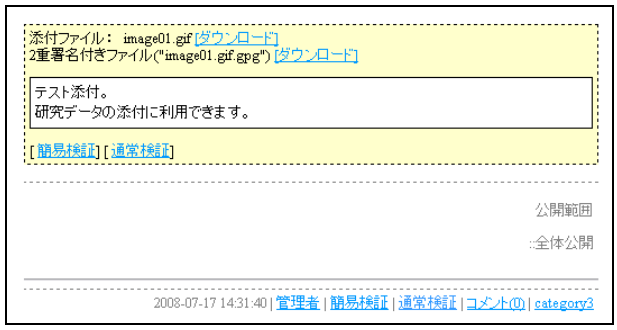

図-5　データのダウンロード画面

録の非改竄性を考慮してのことである．ただし，現在，記事などの管理には DBMS を利用しているが，DBMS を直接操作することにより，記事の修正や削除は可能である．

3.4　検証

各記事の詳細ページおよびデータには，図 7 に示すように,「簡易検証」ボタンおよび「通常検証」ボタンが設置され，ゲストはそれらをクリックすることで簡便な検証を行なえる．なお,「簡易検証」は文書サーバのみで，ユーザの公開鍵と検印サーバの公開鍵を用いて行なう検証である.「通常検証」は，検印サーバに記録されている

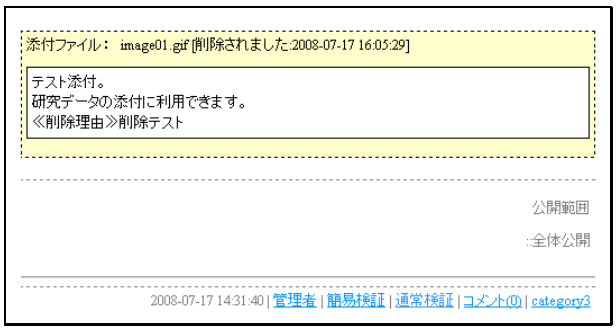

図-6　データが削除された場合

図-7　記事詳細表示

署名との照合も含めて検証を行なう方法である．検証を行なった画面例を図 8 に示す．また，改竄が行なわれた場合の画面例を図 9 に示す．

しかし，本システムはコンピュータ・プログラムとして実現されている以上，スクリプトを書き換えることにより，あたかも検証を行なったように見せかけることも可能である．そのため，少なくともゲストあるいはその他の一般の利用者が独自でも検証を行なえるよう，ユーザおよび検印サーバの公開鍵は，図 10 に示すように，画面上から取得可能としている．

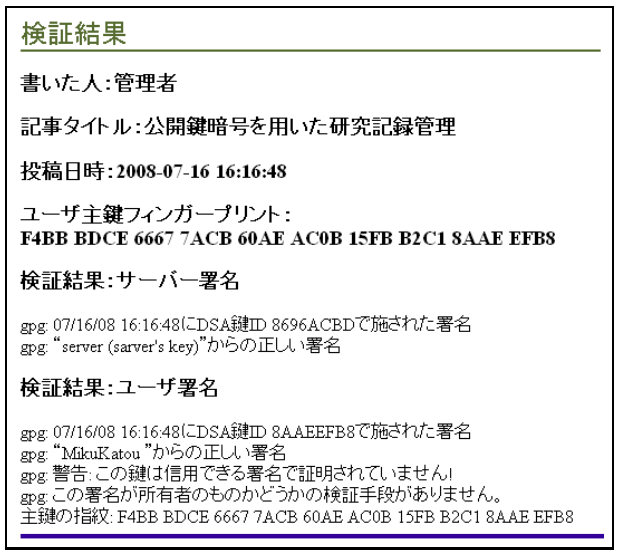

図-8　記事の簡易検証

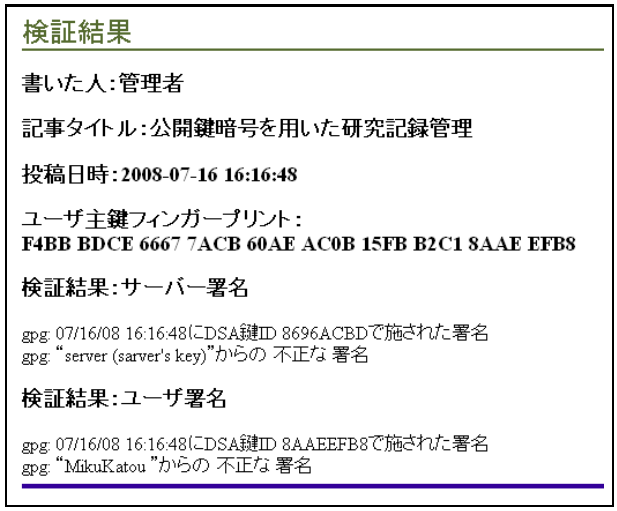

図-9　改竄された記事の検証

3.5　公開範囲の設定

公開範囲は大きく，「全体公開」，「指定グループ」，「自身のみ」の 3 種類に分けられる．「全体公開」は，ゲストとしての認証を受けなくても閲覧可能な公開範囲である．「自身のみ」は，記事を書いたユーザ自身のみが閲覧可能な公開範囲である．「指定グループ」による公開範囲は，グループを単位として指定する．その際の記述例を図 11 に示す．なお，公開範囲は，記事を書いたユーザ，あるいは管理者のみが変更可能である．

この例では分かり難いが，グループは現実の何らかの組織の階層を反映して定義されることを想定している．例えば，「::島根大学:総合理工学部:数理・情報システム学科:情報分野:小林聡研」などである．

ここで，上位のグループに対して閲覧許可が出ている場合，それ以下のグループも，その記事を閲覧可能とし

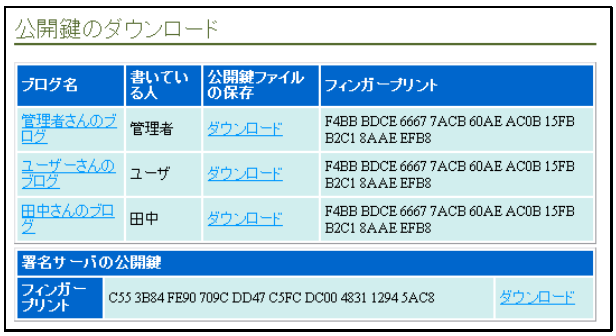

図-10　公開鍵のダウンロード

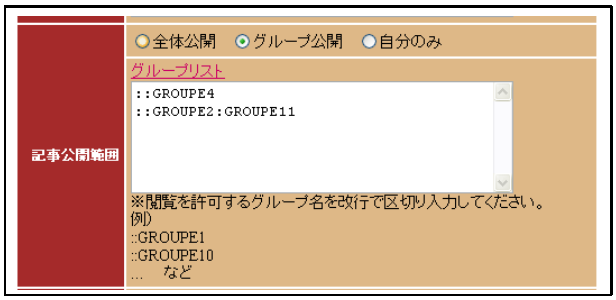

図-11　公開範囲指定

図-12　グループリスト

ている．つまり，ある記事について「::島根大学:総合理工学部」の閲覧許可が出ている場合，「::島根大学:総合理工学部:数理・情報システム学科」に属しているゲストも，「::島根大学:総合理工学部: 数理・情報システム学科:情報分野」に属しているゲストも，当該記事の閲覧が可能になっている．

グループの一覧は，図 12 に示すグループリストによって確認できる．

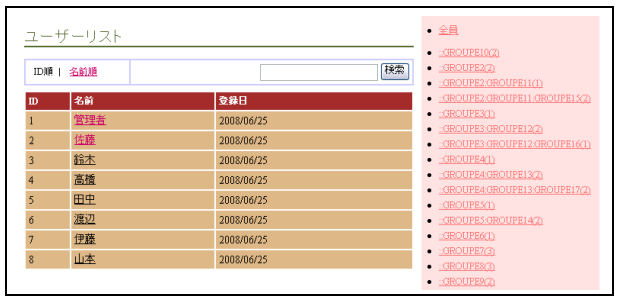

図-13　ユーザリスト

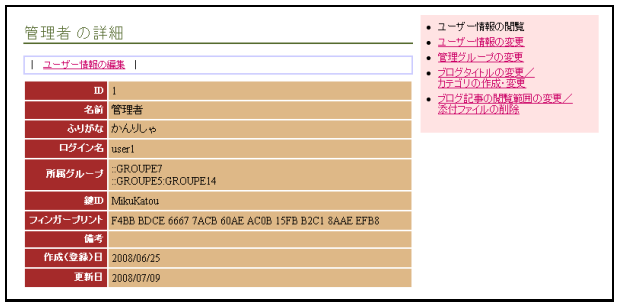

図-14　アカウント管理画面

3.6　システム利用者の管理

システム利用者の登録，修正，削除は，図 13 に示すようなユーザリストからユーザを選択し，図 14 に示すアカウント管理画面を用いて，管理者が行なう．システム利用者の，管理者，ユーザ，ゲストという権限の指定は，管理者のみが設定できる．

ただし，ユーザ自身の情報に限り，ハンドル名，パスワード，鍵情報，所属グループ，備考については修正可能としている．

3.7　グループの管理

グループの新規作成は，ユーザか管理者であれば誰でも作成可能となっている．グループの管理は，管理者およびグループを作成したユーザが原則として行なう．この二者を「グループ管理者」と呼ぶ．グループ管理者は，グループの編集，削除，参加者の変更を行なえる．管理者は，上記に加えてグループ作成者の変更を行なえる．

4　比較

4.1　真正性と非改竄性について

4.1.1　文書の真正性及び非改竄性の保証

真正性と非改竄性を保証する既存の技術として，電子文書署名サービス (タイムスタンプ・サービス) がある．

電子文書署名サービスは，一般的に，作成した電子文書などのハッシュ値を，サービスを提供するサーバに送り，受信したサーバがその受信日時を記録して電子署名を行なうものである．

本稿では本人が書いたという真正性の確保も含めた比較のため，サーバには電子文書などと共に，作成者本人の署名も含めて送り，比較検証を可能としている．

4.1.2 非改竄性の保証の強度・検証効率

電子文書署名サービスは，文書作成者とは関連のない第三者である企業が，作成された文書から得られるハッシュ値に基づいて非改竄性の保証を行なっており，サービスによっては新聞などでスーパーハッシュ値を公開しているものもある．企業がサービスを提供するため，サーバ側への侵入や，両者の秘密鍵を入手することは非常に困難であり，強度は高い．

また，検証は，必要とされた時に，サービス提供者に依頼することでその回答を得られる．

一方，本システムも作成者と作成者とは別の第三者による署名によってその非改竄性の保証を行なっているが，現段階では現実的に完全な第三者への署名サーバの管理の依頼は難しいと考えられる．そのため，電子文書署名サービスに対して強度は劣ると考えられる．強度を上げるためには，ユーザ及び検印サーバの秘密鍵の管理法の改善が課題となる．

検証については，基本的に記事の閲覧ページから 1 クリックで行なうことが可能となっている．

4.2 導入・利用のコスト

電子文書署名サービスは，2005 年の e-文書法制定以来，需要が高まり，各社で比較的手軽なサービスが提供されるようになっている．なかには，契約会社がデータの受け取りからタイムスタンプ付与までの作業をシステム化し，月額定額で契約するといったサービスもある．しかし，サービスに応じて，多くは 1 スタンプあたり 10 円 ~2000 円の課金を行なうシステムであることが多く[*12]，手軽に利用できるとは言い難い．

本システムも，導入までに技術的・金銭的・人的なコストを必要とするが，全て企業に委託する方法と比べると，金銭的なコストは低くなる．また，一度設置した後は，署名や検証までの作業は全てシステム側がバックグラウンドで行なえる．

また現在広く利用されている blog と似たシステムを基本としているため，操作は通常の blog での記事の投稿の作業と同様で済み，利用者は署名に関して意識せずに，限定的ではあるが，真正性の保証された電子記録の保存作業が行なえる．

表-4　公開範囲の指定方法と変更

システム	公開範囲の指定方法	変更
Enzin	ユーザやグループなどの意味を持たせたアイコンのドラッグ&ドロップによる指定	可
ACS	チェックボックスによる指定	不可
本システム	テキストエリアへのグループ名の記入	可

表-5　公開範囲の設定方法

Enzin:
- 自分のみ
- メンバ
- グループ (2 人以上のメンバの集合)
- インターネット全体
- または上記の組み合わせ
- グループは個人で自由に作成可能

ACS:
- 自分のみ
- グループ (一人以上のメンバからなる)
- 一般公開
- パブリックリリース (RSS を利用し，一般公開よりもさらに積極的に情報発信)
- グループの場合は複数のグループを選択可能
- グループは個人で自由に作成可能

本システム:
- 自分のみ
- グループ (一人以上の利用者からなる)
- 全体公開 (詳細は「3.5 公開範囲の設定」)
- グループはユーザであれば作成が可能
- グループの階層化可能
- グループはユーザ同士で共有
- グループの管理はグループ作成者と管理者が行なう

4.3 公開範囲の柔軟な制御

4.3.1 公開範囲指定のインタフェース

公開範囲を指定する際のインタフェースを，Enzin，ACS，本システムで比較すると，表 4 のようになる．表 4 に示される通り，投稿後の公開範囲の設定の変更の可否はあるが，いずれもその公開段階での自由度はほぼ同じと言える．

Enzin は直感的に操作できるインターフェースであり，ACS もクリック操作のみでグループなどの指定が可能であることに対し，本システムは若干分かりにくく手間がかかる．これは本稿ではグループがユーザ全体で共通であるため，グループ数が多くなることが予想され，そのためテキストエリアでの指定とした．

4.3.2 公開範囲の設定段階

同様に，Enzin，ACS，本システムにおける，公開範囲の設定を比較すると，表 5 のようになる．

Enzin および ACS は，一般的なコミュニケーション

[*12] 安価なものもあるが，1 万円程からのプリペイド方式であるなど，利便性に欠ける面がある．

ツールである SNS/blog を基盤として，一般のコミュニケーションの支援を目的として作られている．対して，本システムでは，多くの SNS/blog の機能に含まれる blog を基本としたシステムであるが，研究支援として利用することを前提としている．そのため，ある程度，記事を公開する対象や内容・目的が限定され，グループの数は多くなったとしても，グループ間に何らかの関係が存在することが予想される．そこで，特に実在する組織の階層構造を反映することを想定し，グループを階層的に定義可能とし，グループ間の関連が分かり易いグループ設定と公開範囲の指定を可能とした．

5 考察

本システムの大きな課題として，セキュリティの問題がある．

真正性の保証については，ユーザの秘密鍵を IC カードや USB メモリなどの外部記憶メディアに保管し，必要記事を投稿する時にのみそれを参照することも課題としてあげられる．また，本システムにおいて，なりすましが行なわれる可能性もある．しかし，ユーザの秘密鍵を IC カードや USB メモリなどの外部記憶メディアに保管することにより，なりすまして記事を投稿することは出来なくなる．

非改竄性の証明の強度を高めるためには，リンキングやヒステリシス署名，履歴交差などの技術の導入の検討も必要であろう [11]．また電子署名の長期利用に関しては，秘密鍵の危殆化などの問題もあり，評価・検討が必要である [12, 13]．

現状では，検印サーバでの署名のために，記事やデータそのものを通信している．そのため，情報漏洩の危険性がある．現在は，SSL での通信を行なうことで対処しているが，情報漏洩の対策として十分か否かは不明である．記事やデータのハッシュ値のみの通信に限ることも含めて検討を要する．

また，公開鍵の正当性の確保のため，信用の輪 (Web of Trust) も含めた PKI の利用の検討も必要であろう．

ユーザ・インタフェースに関連した課題としては以下のようなものが挙げられよう．

柔軟な公開範囲の設定については，本システムではグループの階層化を取り入れたが，既存のグループの上位に新たに階層を作成することはできない．また，異なる上位グループを持つグループ同士であっても，そのグループの構成員の間には何らかの関連が有る場合も現実にはある．そのような関連を持たせる事も含め，今後さらに自由度の高いグループ管理を可能としていくことも課題となる．

本システム，または本システムと同種のシステムが複数稼動した場合，ゲストが各ユーザの blog を閲覧するたびに，個々にログインを要求されるのは煩雑である．そこで，本システムあるいは同種のシステム間において，認証の delegate 機能も必要であろう．

研究支援という観点からは，記事中での表やグラフ，数式や化学式への対応も不可欠である[*13]．このような問題については，後藤らによるシステムや [14]，Jipsen の仕事 [15] が利用/応用可能であろう．また漢字における異体字や国字に対しての対応も検討していきたい[*14]．

また，記事へのグループ指定その他において，Ajax 技術などを用いた，ユーザ・インタフェースの改善も今後の課題である．

6 まとめ

本研究では研究記録などの管理を主目的として，公開範囲を柔軟に制御可能かつ，記録などの真正性および非改竄性を可能な限り保証し，検証の機能も持つシステムとして arχves の構築を試みた．結果として，ユーザおよび検印サーバの秘密鍵の管理など，セキュリティ上の課題はあるものの，一定の機能を持つシステムを実現できた．本システムにより，安価に，DVCS でも TAP でもない，電子署名付き電子文書管理の第三のあり方を実現できた．

今後は，本システムの改良とともに，本システム上に蓄積される記録の知的処理に関して研究を行なう予定である．

参考文献

[1] 原田 篤史, 西垣 正勝, 曽我 正和, 田窪 昭夫, 中村 逸一, “ライトワンス文書管理システム”, 情報処理学会論文誌, vol. 44, no. 8, pp. 2093-2105, 2003.

[2] 陳 明強, 吉川 正俊, “時刻認証付き XML 文書のデータベースによる管理について”, 電子情報通信学会第 16 回データ工学ワークショップ (DEWS2005), 5A-i10, 2005.

[3] 宇根 正志, 松本 勉, “可用性および安全性の観点からみた各タイムスタンプ方式間の関係”, 情報処理学会論文誌, vol. 43, no. 8, pp. 2644–2658, 2002.

[4] Toshiyuki Masui, Satoru Takabayashi, “Instant Group Communication with QuickML”, *Proc. ACM Conference on Supporting Group Work (Group '03)*, pp.268–273, 2003.

[*13] HTML タグを直接書くことで，限定的ではあるが可能．

[*14] 誤字/嘘字への対応は不確定．

[5] 江渡 浩一郎, 高林 哲, 増井 俊之, “quikWeb: メーリングリストと Wiki を統合したコミュニケーション・システム”, 情報処理学会研究報告, 2004-HI-111, pp. 5-11, 2004.

[6] 永田 周一, 安村 通晃, “Enzin: 情報の公開範囲を手軽に変更できるコミュニケーションツール”, 情報処理学会論文誌, vol. 48, no. 3, pp. 1134-1143, 2007.

[7] 高井 一輝, 河口 信夫, “ACS: 多様な人間関係を表現可能なソーシャルネットワーキングシステム”, 情報処理学会論文誌, vol. 48, no. 7, pp. 2328-2339, 2007.

[8] 高田 良宏, 笠原 禎也, 毛利 信浩, 松平 拓也, “多様なアクセス制限に対応した自然科学データベースシステムの開発”, 学術情報処理研究, no. 11, pp.50–59, 2007.

[9] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, “Internet X.509 Public Key Infrastructure Time-Stamp Protocol (RFC3161)”, ftp://ftp.rfc-editor.org/in-notes/rfc3161.txt

[10] 黒田 努, 佐藤 和人, “基礎 Ruby on Rails”, インプレスジャパン, 2007.

[11] 洲崎 誠一, 松本 勉, “電子署名アリバイ実現機構 — ヒステリシス署名と履歴交差”, 情報処理学会論文誌, vol. 43, no.8, pp2381–2393, 2002.

[12] 宮崎 邦彦, 吉浦 裕, 岩村 充, 松本 勉, 佐々木 良一, “第三者機関への依存度に基づく長期利用向け電子署名技術評価手法の提案”, 情報処理学会論文誌, vol. 44, no. 8, pp. 1955–1969, 2003.

[13] 小森 旭, 花岡 悟一郎, 松浦 幹太, 須藤 修, “署名鍵漏洩問題における電子証拠生成技術について”, 電子情報通信学会 「暗号と情報セキュリティシンポジウム」予稿集, pp. 983–988, 2003.

[14] 後藤 洋信, 坂本 雅洋, 江見 圭司, “数式表示可能なウェブ上でのコミュニケーションシステムの構築”, 情報処理学会 研究報告 2008-CE-94, pp.1–8, 2008.

[15] Peter Jipsen, “ASCII MathML”, http://www1.chapman.edu/ jipsen/asciimath.xml