平成 22 年度

卒業論文

# 研究記録管理･公開･検証システム arχves の Web 認証基盤への対応

指導教員　小林　聡　准教授

島根大学　総合理工学部

数理･情報システム学科　情報工学コース

S063078　藤坂　達也

# 目次

## 1. はじめに

近年の情報技術の発展に伴い、電子媒体に研究記録などを保存する機会が多くなってきた。電子媒体に保存された研究記録はあとで変更することが容易であり、改ざんが行われる危険性が高い。改ざんという行為そのものを防止することは技術的に困難であるが、改ざんの有無を検知することは可能である。発覚しないから改ざんを行うのであって、改ざんしたことが後ほど発覚する環境に研究記録が保存されているなら、改ざんを行おうと考える人は少なくなると推測される。この際、研究記録の非改竄性をどのように保証するかが課題となる。

このような課題に対して、原田らは電子カルテシステムがハードウェア的技術によって非改竄性を保証しているが、ソフトウェアの観点からも非改竄性を保証する仕組みが必要だと考え、電子署名とタイプスタンプを用いたシステムモデルを提案した。このモデルは検印モデルと呼ばれ、文書作成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行う形式である[1]。また、陳らは XML 文書の時刻認証を伴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。このような、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わないことを原則としており、対して TAP はアーカイブも行うことを原則としている。また宇根らは DVCS や TAP で用いられる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]。

タイムスタンプとは、電子データの作成・更新などが行われた時刻を証明するための情報である。時刻情報の偽造防止のために、電子データのハッシュ値と時刻の組み合わせにタイムスタンプ局が署名を行っている。タイムスタンプが用いる時刻情報には時刻配信局が配信する情報が利用されている。タイムスタンプによって、ある電子データがタイムスタンプの示す時刻以前に存在したことが証明されるが、文書作成者が正規の人物であるかを保証することはできない。そのため、文書作成者によるディジタル署名が必要になってくる。

また、e-mail や 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の手段となってきている。そのようなネットワーク環境を利用し、関連した研究を行う者同士が、論文の公刊前に研究の進め方や研究・実験の成果について意見交換が可能な環境が提供されれば、研究を進める上で大きな利点があると考えられる。

そして記録、コミュニケーションと並ぶ、コンピュータが果た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となり、研究自体が活性化するであろう。ただし、記録機能においても、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、あるいは秘匿する必要のある情報も存在する。そのため、情報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(Social

Network Service)などのサービスにおいて、記事の公開範囲はある程度の制御が可能ではあるが、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とするシステムの構築と運用を試みた。また、SNS を基盤としたシステムとして、永田らは Enzin を[7]、高井らは ACS を開発し[6]、運用実験を行った。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。

ar χ ves では、投稿された研究記録に対して公開鍵暗号を用いてディジタル署名を四重に行っている。一つ目の署名は投稿者本人による署名（ユーザ署名）であり、投稿した人物が本人であることを証明するためのものである。二つ目の署名は第三者の署名（サーバ署名）であり、これは投稿記録のある時点での存在を証明している。先のタイムスタンプシステムを用いるときと署名の付け方は同一である。この二つの署名によって、研究記録の非改竄性の保証が行われている。三つ目の署名は確認者による署名（確認者署名）であり、研究記録の確認作業を行った人物が確認者自身であることを証明する。四つ目の署名は第三者による署名（サーバ署名）であり、研究記録をある時点で確認者が内容確認したことを証明する。三つめと四つ目の署名によって、後述する研究記録としての知的財産的価値が高められている。

現在、米国では特許取得の際、先発明主義がとられている。先発明主義とは、最初に発明した発明者に特許権を与える制度である。この制度において、発明日の立証にはラボノートによる研究記録の記載が重要になる。しかし、ラボノートが特許取得の際の証拠として使用できる条件として、記録の改変が不可能な事及び、第三者による確認と署名などが挙げられている。このような条件を満たすために、ar χ ves には三つ目と四つ目の署名が行われている。

本研究では以前から指摘されていた、検印サーバを介して文書サーバへアクセスする際の ID・パスワードの盗難を防止することと、複数回のログイン作業を省略し快適な利用環境を提供するため、Web 認証基盤への対応を行った。

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

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

# 1． システムの特徴

## 1.1. 概要

本システムは、研究記録などの管理を主目的とし、公開範囲を柔軟に制御可能、かつ記録の正真性及び非改竄性を可能な限り保証することを目的としたシステムである。

データの正真性と非改竄性の保証を可能とするため、記事データ投稿時に、記事データに対して二重署名の処理を行い、平易な操作で検証を行えるようしている。また、電子情報の簡便な記録・保存と扱いやすさのために、基本的な操作は近年広く普及している SNS/Blog をベースとしている。

そして、柔軟な公開範囲の実装の為、一般の SNS/Blog のように Blog コンテンツ全体の公開範囲を指定する方式ではなく、投稿される記事ごとに公開範囲を指定可能としている。

特徴を要約すると以下となる。

◆ 正真性と非改竄性の保証

◆ 確認者による確認署名

◆ 研究記録の公開範囲の細かな設定が可能

◆ 数式表現に対応

◆ Ruby on Rails による実装

## 1.2. 利用ツール

Ruby on Rails は本システムの基幹となる SNS/Blog 機能の構築を中心に活用し、電子署名の付与及び検証は、全て GnuPG を活用している。

表１． 利用ツール一覧

| | |
|---|---|
| 開発言語 | Ruby ver. 1.8.7 |
| フレームワーク | Ruby on Rails ver. 2.2.2 |
| 公開鍵暗号ソフト | GnuPG ver. 1.4.9 |
| データベース | MySQL ver. 5.0.27 |
| ウェブサーバ | Apache ver. 2.0.6 |
| | Mongrel Web Server ver. 1.1.5 |
| SSL ライブラリ | OpenSSL ver. 0.9.8 |

## 1.1. 正真性と非改竄性の保証

企業における、記録の正真性及び、非改竄性を保証する方法として、部下の作成した記録に対し、上司が検印を捺すことが広く行われている。このような、記録の記述者とは異なる者による検印、あるいはそれに類似の行為/操作を行うモデルが原田らによって提案されており、そのようなモデルをここでは「検印モデル」と呼ぶ。本システムは上記の検印モデルを元に、記事の正真性と非改竄性の保証を行う。記事やデータに検印を行うサーバと文書を保管するサーバの異なる 2 つのサーバを用意し、公開鍵暗号を用い、記述者が自身の秘密鍵で電子署名をするとともに、検印サーバの秘密鍵で電子署名を行うことで、検印モデルを実現している。二重署名された記事データの構成内容の例を図1に、二重署名の処理の流れを表2に示す。

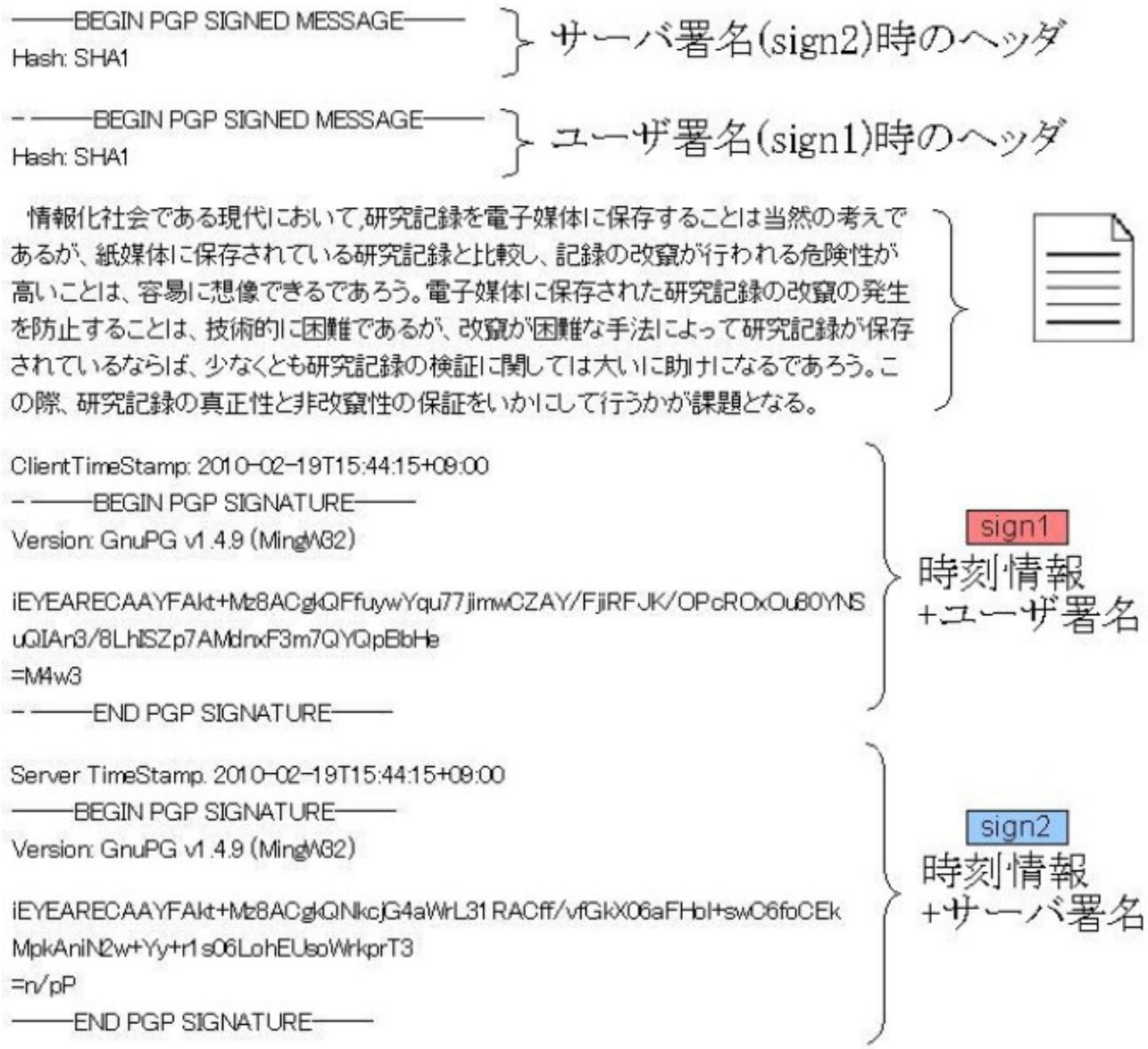
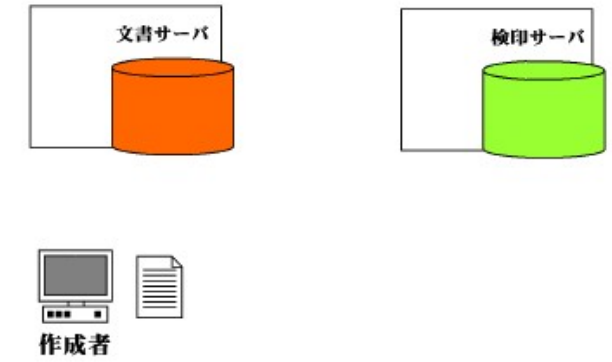
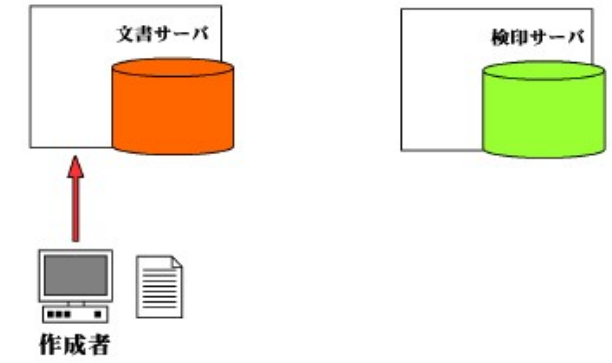
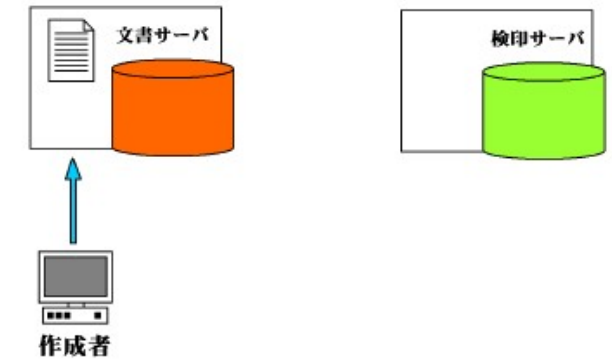
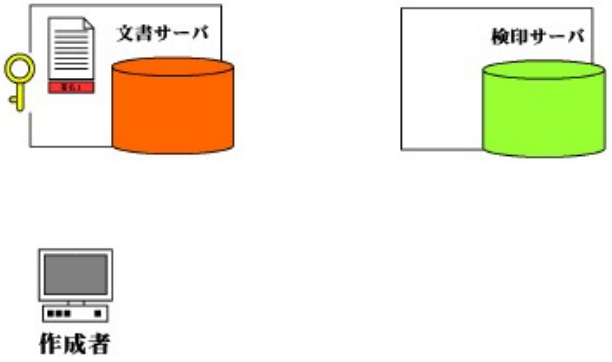
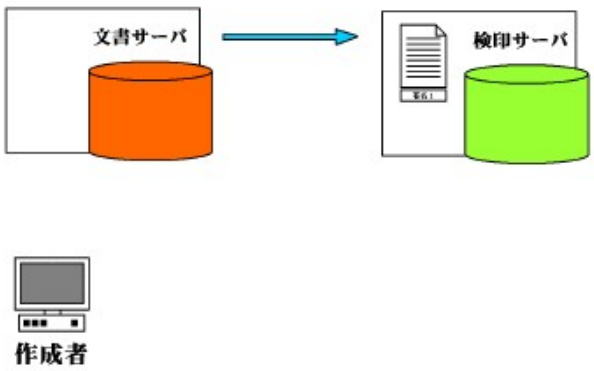
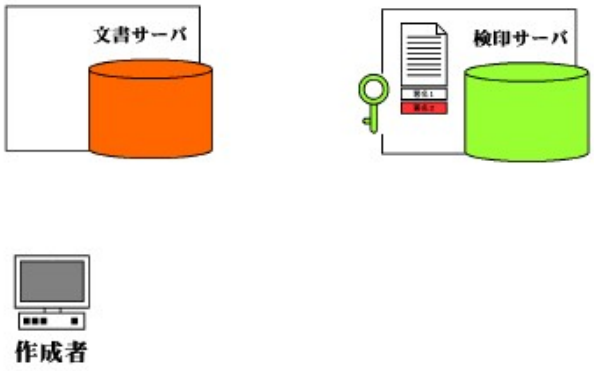
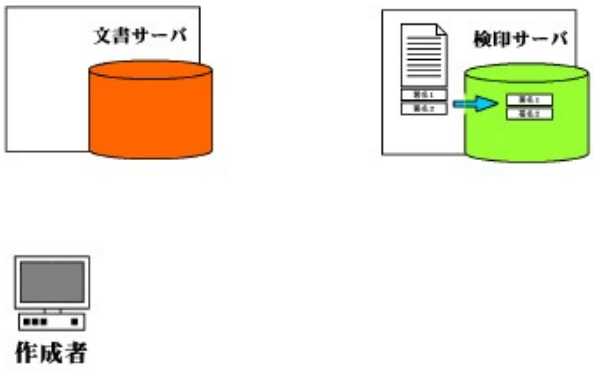
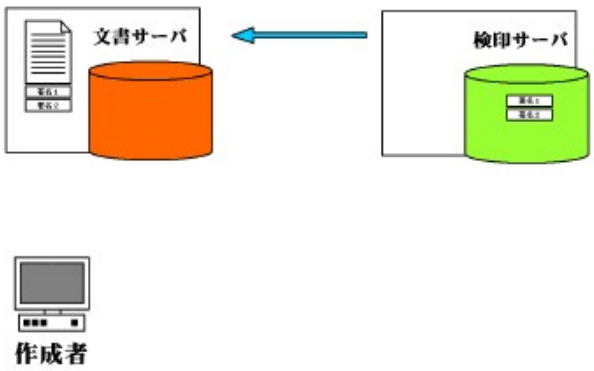
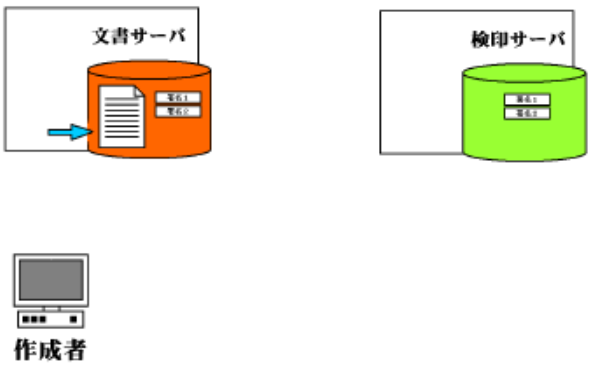



図１．二重署名された記事データの構成

表２．二重署名処理の流れ

| | |
|---|---|
|  | 文書サーバ：研究記録を保管するサーバ<br>検印サーバ：第三者による署名を行うサーバ |
|  | 作成者は Web ブラウザから文書サーバにログインする |
|  | 作成者は、Web ブラウザ上で研究記録を書き、文書サーバに研究記録を送信する |
|  | 文書サーバは、作成者の秘密鍵で研究記録に署名する |

| | |
|---|---|
|  | 文書サーバは、研究記録と署名１を検印サーバに送信する |
|  | 検印サーバは、検印サーバの秘密鍵で研究記録＋署名１に署名を行う |
|  | 検印サーバは、署名部をデータベースに保存する |
|  | 検印サーバは、研究記録と署名部を文書サーバに送信する |

| | |
|---|---|
|  | 文書サーバは、研究記録と署名部をデータベースに保存する |

## 2.4. 確認者による署名機能

本システムは二重署名処理によって、研究記録に対しサーバを第三者とした署名を行っている。しかし、検印サーバによる署名は機械的に付せられた署名であり、研究記録の内容の確認を行っているわけではない。そこで、本システムに確認者による署名を行う機能を付加することによって、研究記録の知的財産としての価値を高めることができる。

企業、大学、研究機関などで使用されているラボノートには、本人の署名と共に、確認者の署名を行う必要がある。岡崎、隅蔵編による「理系なら知っておきたいラボノートの書き方」[12]によると、ラボノートに要求されることとして、第三者による確認と署名が必要であるとされている。

また、確認者に求められる条件として以下を満たす必要がある。

条件1：共同開発者でないこと

条件2：ラボノートに記載された内容を理解することができること

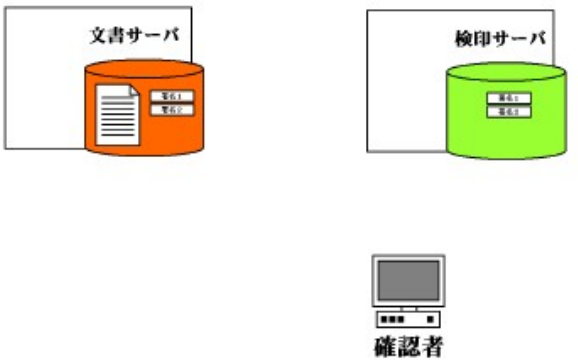
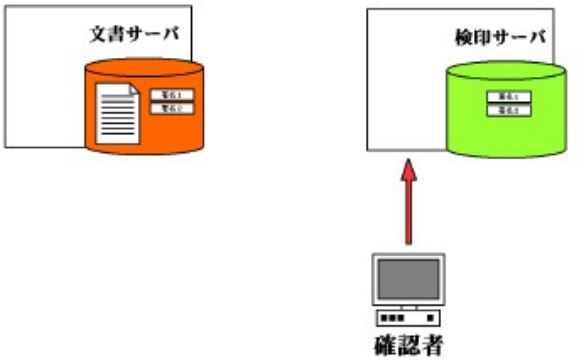
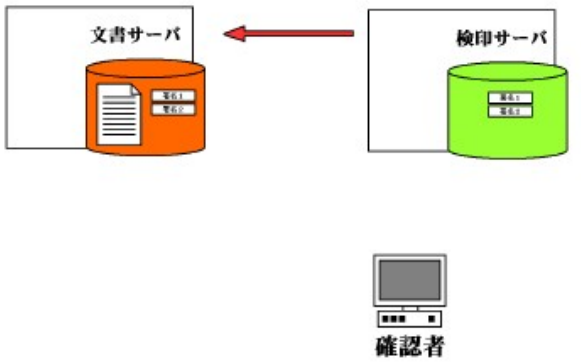
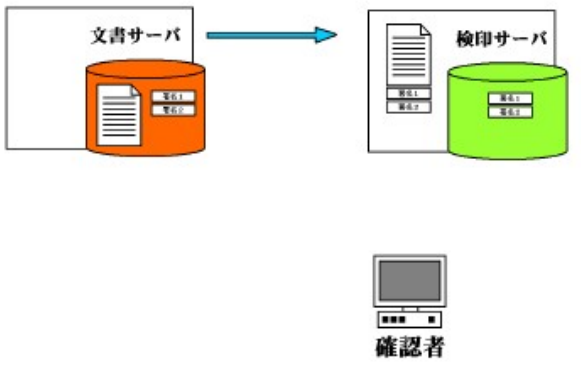
このような条件がある理由は次のように推測される。まず、条件1は確認者が共同開発者であった場合、当該研究者と利害が一致するため、ノートの日付、内容などについて、偽造、改竄を行う可能性が他の人をと比較して高いためである。条件2はラボノートの内容を確認する人は、機械的に署名をするだけでは不足であり、ノートに記載された実験内容を理解し署名を行う必要があるためである。

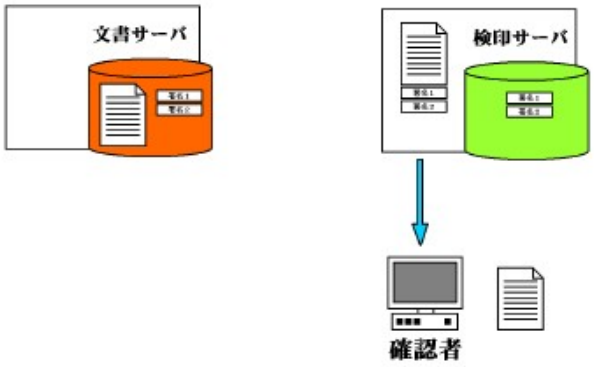
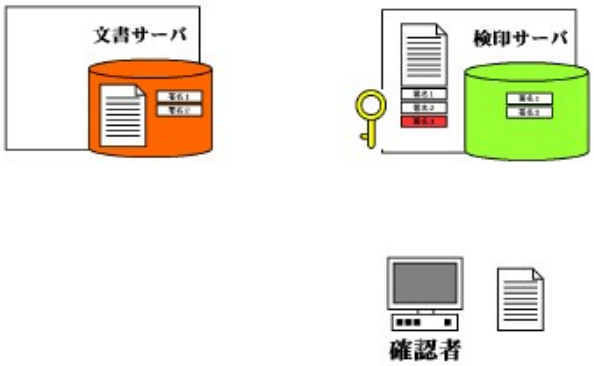
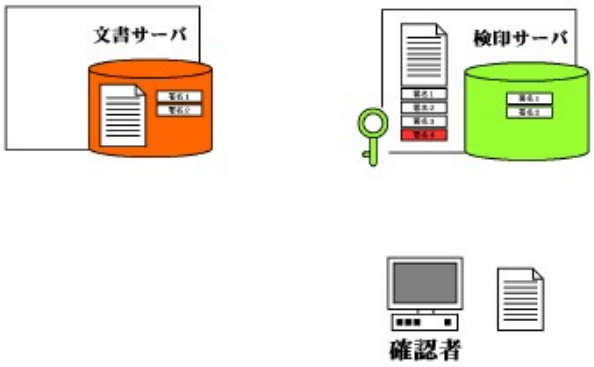
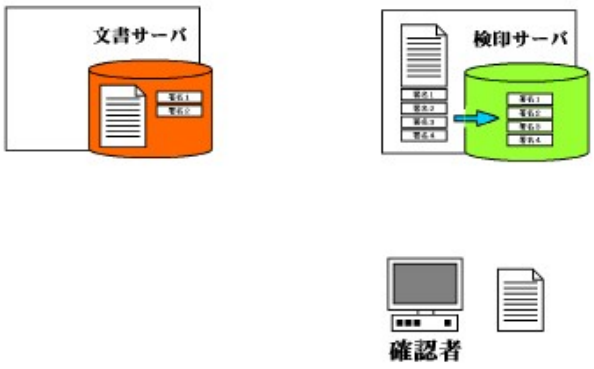
本システムにおいて、確認者による署名に関する処理の流れを表3のようになっている。

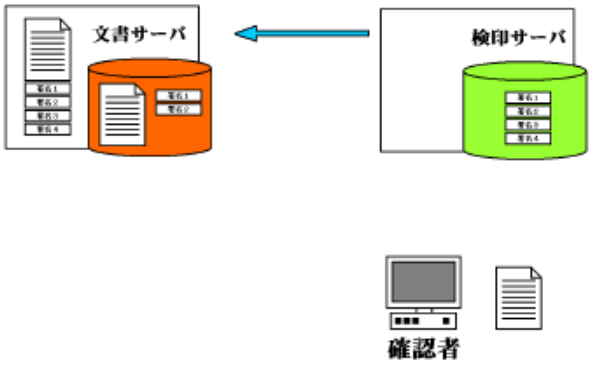
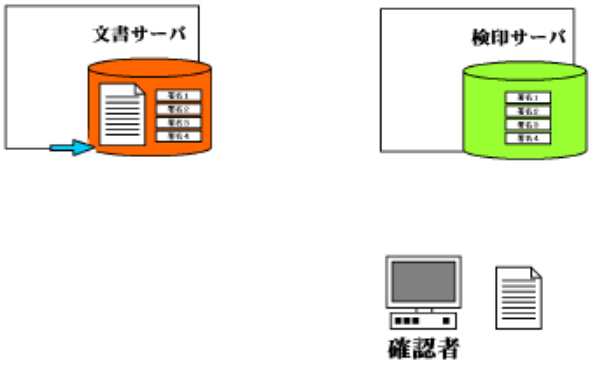
確認者が検印サーバを介して二重署名処理が終了した記事を閲覧すると、図2のように表示される。

確認者による署名処理が行われ、四重署名が付加された記事の内容は図3のように表示される。

表３．確認者による署名処理の流れ

| | |
|---|---|
|  | 文書サーバ：研究記録を保管するサーバ<br>検印サーバ：第三者による署名を行うサーバ |
|  | 確認者は Web ブラウザから検印サーバにログインする |
|  | 検印サーバを介し文書サーバにログインする |
|  | 検印サーバを介して文書サーバから研究記録を取り寄せる |

| | |
|---|---|
|  | 確認者が Web ブラウザ上で研究記録の内容を確認する |
|  | 検印サーバが確認者の秘密鍵で研究記録と署名部に署名を行う |
|  | 検印サーバが検印サーバの秘密鍵で研究記録と署名部に署名を行う |
|  | 検印サーバが署名部をデータベースに保存する |

| | |
|---|---|
|  | 検印サーバは研究記録と署名部を文書サーバに送信する |
|  | 検印サーバは研究記録と署名部をデータベースに保存する |

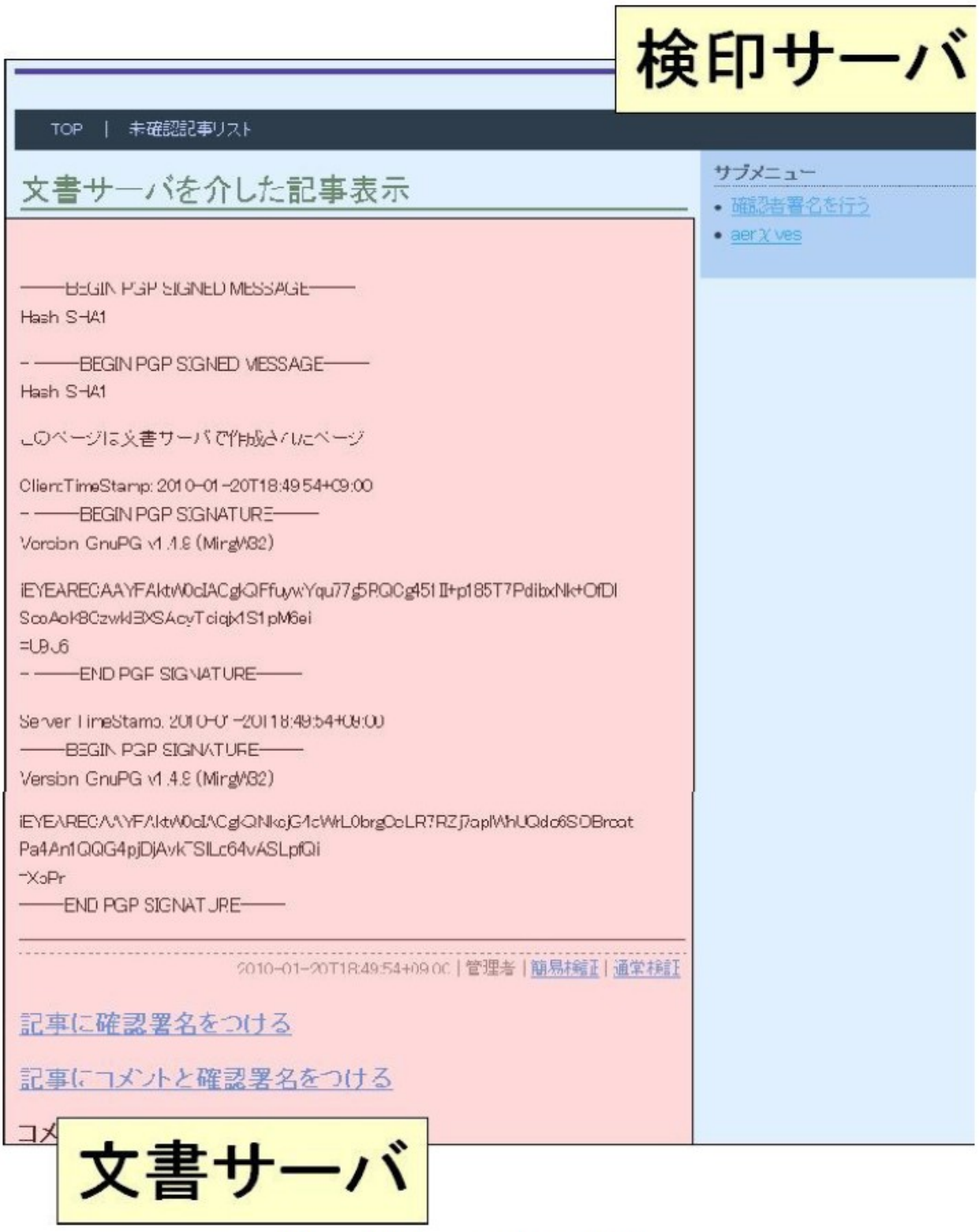



図２．検印サーバを介した記事表示

## 記事投稿例


-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

- -----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

- - -----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

- - - -----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1


arXivの特徴まとめ

・SNS/ブログベース
・記事ごとに柔軟な公開範囲設定が可能
・デジタル署名を用いたデータへの四重署名処理
・確認者による記事への署名機能
・数式表現に対応


ClientTimeStamp: 2010-02-01T13:43:25+09:00
- - - -----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (MingW32)

iEYEARECAAYFAktmWHQACgkQFfuywYqu77j/EgCg5NaHMfFMZGbEyRUiZrBZFZ+H
WWEAni66Iq5ArlKAHHinC5N4cw9trUt
=TmAy
- - - -----END PGP SIGNATURE-----

Server TimeStamp. 2010-02-01T13:43:26+09:00
- - -----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (MingW32)

iEUEARECAAYFAktmWH4ACgkQNkcjG4aWrL1ezQCYXdKFncogzSZ2gJFrHG1Uj3wW
2wCgIICDhyV/HRG8LF9BwC6Qhb+cjJs=
=KGAd
- - -----END PGP SIGNATURE-----

ConfirmerTimeStamp: 2010-02-01T13:51:00+09:00
- -----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (MingW32)

iEYEARECAAYFAktmXbQACgkQFfuywYqu77i10wCgp8xGJZy2E8sHDeDXh3/zJjIQ
VT0AoNGaf6cGJqrSuooXcxDD4B/jY730
=HDip
- -----END PGP SIGNATURE-----

Server TimeStamp#2. 2010-02-01 13:51:00
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (MingW32)

iEYEARECAAYFAktmXbQACgkQNkcjG4aWrL3pKQCfRb2JDXGRa27f8vbc9A5ENqtM
JLAAnRFGwXcO6HD/QHYSNJZbQrjmqLQq
=cz2i
-----END PGP SIGNATURE-----

図３．四重署名処理済み記事本文

## 2.5. 検証機能

前節の研究記録に対して行った、二重のディジタル署名によって、投稿された研究記録の正真性および非改竄性の保証が行われているが、それを確かめるためには署名の検証作業を行う必要がある。

また、電子商取引推進協議会、認証・公証 WG [15]のガイドラインでは、ディジタル署名の有効性を長期的に維持する為の用件として、次の 4 つを挙げている。

① 署名検証時に、署名再検証に必要な情報を明確にしておくこと

② 署名検証時の時刻を明確にしておくこと

③ 署名再検証時に必要な情報を改竄検出可能な状態にすること

④ 署名再検証時に必要な情報を保存すること

①は、検証を行う際に確かにその署名が有効であることを示す情報（あるいは失効情報）を明示しておくこと。そして、署名の検証が確かに保証されていることを確認した日時を明確にする。②は、その時点まで確かにデータが保証されていることを記録し、③再検証を正しく行えるように改竄の検出を可能にして、④署名の再検証に必要な情報を保存し、第三者でも再検証を行えるようにしておくということである。これらの用件を満たすことがディジタル署名の有効性の長期的な維持には必要であるとされている。

このように、本当にその署名が確かであることを確認する、あるいは長期的な有効性の維持には「署名の検証」が必要となってくる。本システムでは、その検証を簡単に行う事ができる機能を実装している。

各研究記録には、「簡易検証」機能および「通常検証」機能が用意され、閲覧者はそれらのリンクをクリックすることで簡便な検証を行うことができる。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う検証である。通常検証は、検印サーバに記録されている署名データとの照合も含めて検証を行う。検証を行なった画面例を図4に示す。また、研究記録の内容のみを書き換えられた場合の検証画面例を図5に示す。

## 検証結果

投稿者:管理者

検証記事タイトル:arχvesについて

投稿日時:2010-02-19 15:44:15

ユーザ主鍵フィンガープリント:
F4BB BDCE 6667 7ACB 60AE AC0B 15FB B2C1 8AAE EFB8

検証結果:確認者サーバー署名

検証結果:確認者署名

検証結果:サーバー署名

gpg: 02/19/10 15:44:15にDSA鍵ID 8696ACBDで施された署名
gpg: “server (sarver's key)”からの正しい署名

検証結果:ユーザ署名

gpg: 02/19/10 15:44:15にDSA鍵ID 8AAEEFB8で施された署名
gpg: “MikuKatou ”からの正しい署名
gpg: 警告: この鍵は信用できる署名で証明されていません!
gpg: この署名が所有者のものかどうかの検証手段がありません。
主鍵の指紋: F4BB BDCE 6667 7ACB 60AE AC0B 15FB B2C1 8AAE EFB8

検印サーバ側の署名部との照合結果

検印サーバ側の署名部と一致しています。
・全ての署名が一致しています。

図４．検証結果（非改竄時）

## 検証結果

投稿者:管理者

検証記事タイトル:arχvesについて

投稿日時:2010-02-19 15:44:15

ユーザ主鍵フィンガープリント:
F4BB BDCE 6667 7ACB 60AE AC0B 15FB B2C1 8AAE EFB8

検証結果:確認者サーバー署名

検証結果:確認者署名

検証結果:サーバー署名

gpg: 02/19/10 15:44:15にDSA鍵ID 8696ACBDで施された署名
gpg: “server (sarver's key)”からの 不正な 署名

検証結果:ユーザ署名

gpg: 02/19/10 15:44:15にDSA鍵ID 8AAEEFB8で施された署名
gpg: “MikuKatou ”からの 不正な 署名

検印サーバ側の署名部との照合結果

検印サーバ側の署名部と一致しています。
・全ての署名が一致しています。

図５．検証結果（改竄時）

## 2.6. SNS／Blog ベース

本システムは、電子情報の簡便な記録・保存と扱いやすさの為、基本的な操作は近年広く普及している SNS/Blog をベースに実装している[19]。本システムにアクセスすると、図6のようなトップページコンテンツが表示される。ここでユーザとして認証を行い、ログインを行うと、図7のような画面が表示される。

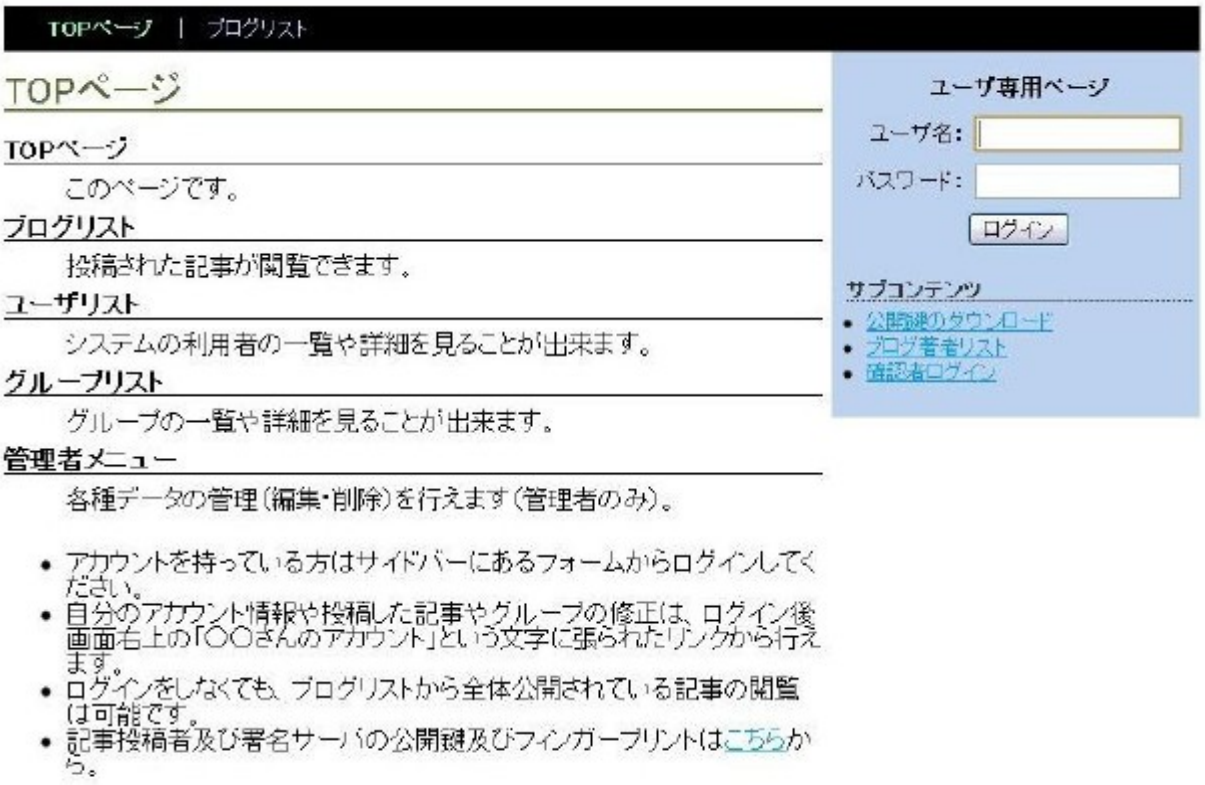



図６．未ログイン時トップページ

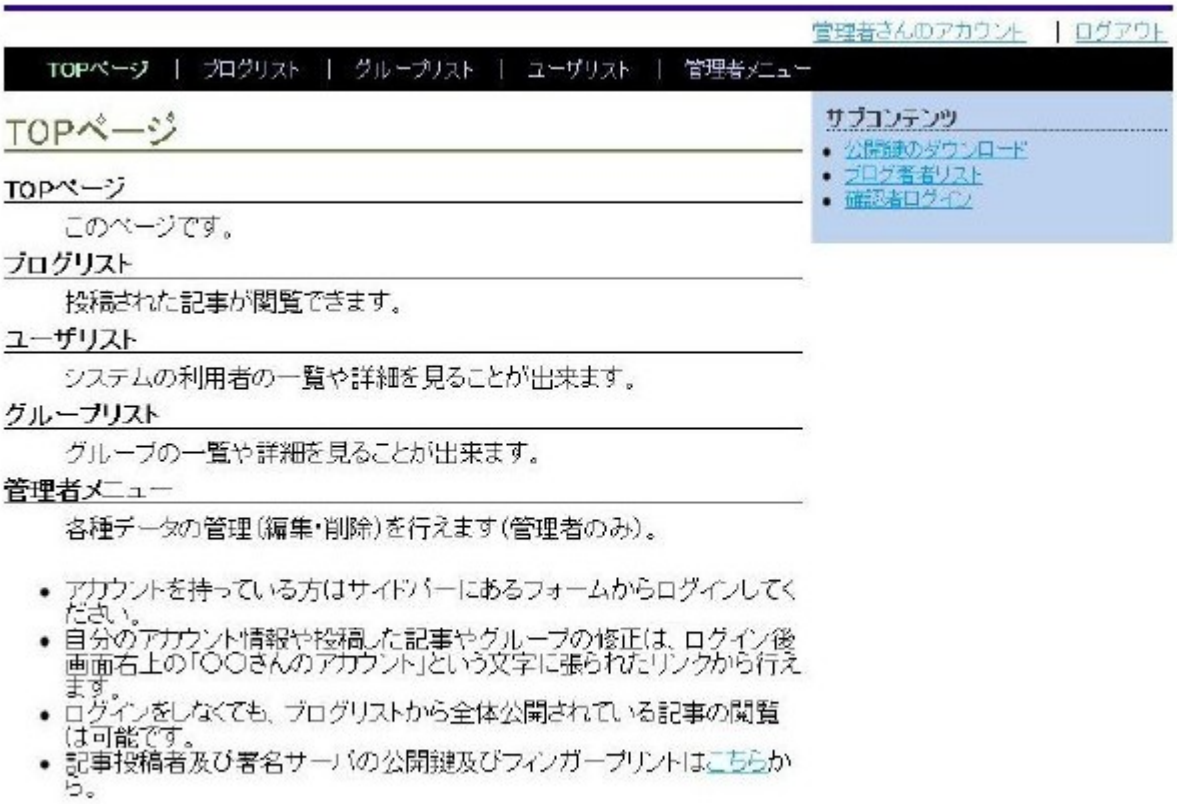



図７．ログイン後のトップページ

ページは大きくメニューバー、メインページ、サイドバーで構成されている。メニューバーには各コンテンツへのリンク、メインページには各コンテンツの内容、サイドバーにはコンテンツ内のサブコンテンツ/関連ページへのリンクが表示される。

また、ログインすることで、メニューバー上部にアカウント名が表示され、各アカウントメニューへのリンク及びログアウト用のリンクが表示される。メニューバー及びサイドバーは、ログインしたアカウントの権限ごとに異なったコンテンツが表示される。

## 3. Web 認証基盤への対応

### 3.1. Web 認証基盤とは

Web 認証基盤とは、Web 認証を1つのサーバで行うシステムであり、そのサーバを認証サーバと呼ぶ。 認証サーバは複数のシステムから利用される。そのため、認証サーバのセキュリティは厳重に行う必要がある。

そして、Web 認証基盤を用いることでシングルサインオンを実現できる。シングルサインオンとは一度ユーザ認証を行えば、複数のサーバでのユーザ認証をパスできることである。そのため、複数のサーバにアクセスが必要なシステムにおいて、1回のログインで各機能を利用可能となる。システム利用において、別のサーバにアクセスするたびにログインし直すことを回避でき、快適な利用環境を提供することができる。さらに、シングルサインオンでは同一の認証サーバを利用する際には、同一の ID とパスワードを利用する。そのため、新たなサイトに会員登録するたびに、ID とパスワードを考えなくてよいため、ユーザが覚える必要のあるパスワードの総数が減少する。結果として、より複雑なパスワードを設定しやすく、ユーザ認証におけるセキュリティレベルを高めることになる。

本システムにおいても、確認者による確認署名機能の利用において、検印サーバを介して文書サーバへログインすることが必須であったが、Web 認証基盤の導入によって二度目のログインを省略できる。さらに検印サーバを介して文書サーバへアクセスする際に ID とパスワード盗難の危険があったが、この危険も Web 認証基盤の導入によって回避できる。

Web 認証基盤の実現方法としては、OpenIDとShibbolethがある。Shibbolethは国立情報学研究所が主導であることを考えると、本研究でもShibbolethを用いるのが妥当である。しかし、ShibbolethはOpenIDに比べ導入に際して難しい点があるため、今回はOpenIDを用いることにした。その理由としては、次の3点がある。

・OpenIDの方が資料が多い

・Shibbolethに関する資料は日本語のものが少ない

・Shibbolethは独自に認証サーバをたてる必要があるが、

OpenIDでは既存の認証サーバ（Google appsなど）を利用できること

以下にOpenIDとShibbolethについて簡単に紹介する。

### 3.2. OpenID とは

OpenID とは、Web 認証基盤を実現する方法のひとつである。1つの ID を異なる対応サイトで利用でき、その際に利用される ID を OpenID と呼び、 ユーザ固有の URL 形式の ID を用いる。対応サイトでは、利用者が認証サービスを行うサイトを選べるので、信頼のおけないサイトに個人情報（E-mail アドレスなど）を登録することを回避できる。認証サーバは認証の結果と必要最小限の情

報のみをサービス側に通知するため、サービス側は ID/ パスワードを保有・管理しない。

*3　OpenID　　日本語サイト: http://www.openid.ne.jp

英語サイト　:http://openid.net

### 3.3. Shibbolethとは

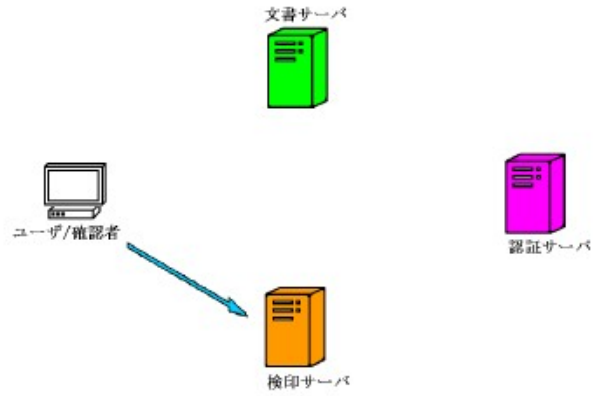
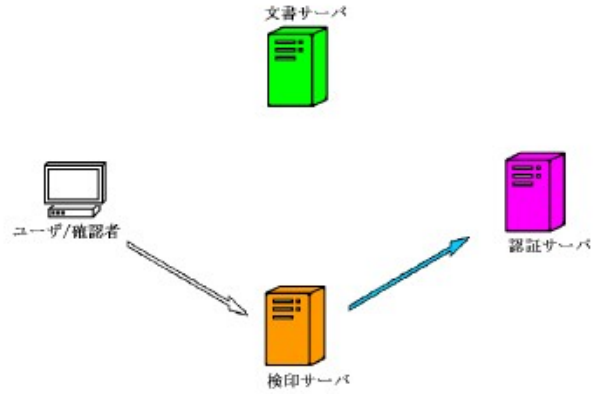
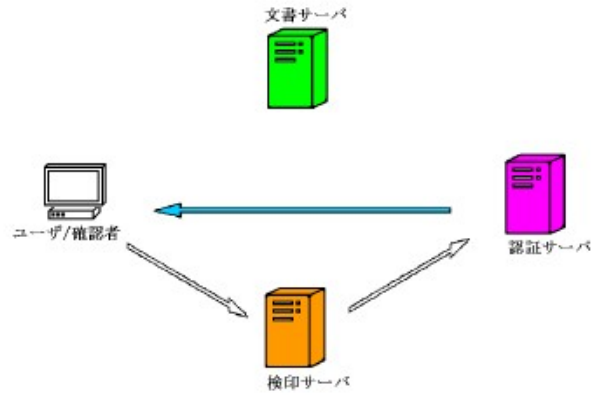
Shibbolethとは、Web 認証基盤を実現する方法のひとつである。実現できることは OpenID と同様である。Shibboleth は米国の EDUCAUSE 標準の認証システムとして、採用されている。EDUCAUSE とは情報技術の活用によって教育の向上を目標としている米国の高等教育機関である。日本国内においては、国立情報学研究所が学認*4 として支援している。学認とは、定められた規定（ポリシー）を信頼しあうことで相互に認証連携を実現し、学術リソースを利用・提供する組織から構成された連合体である。また、Shibboleth を用いた国立情報学研究所主導の Web 認証基盤を学認と呼ぶ。学認においては、各所属大学において認証を実施する。そして、学内でのシングルサインオンを実現可能で、かつ学外の公開サービスへのシングルサインオンも実現可能である。

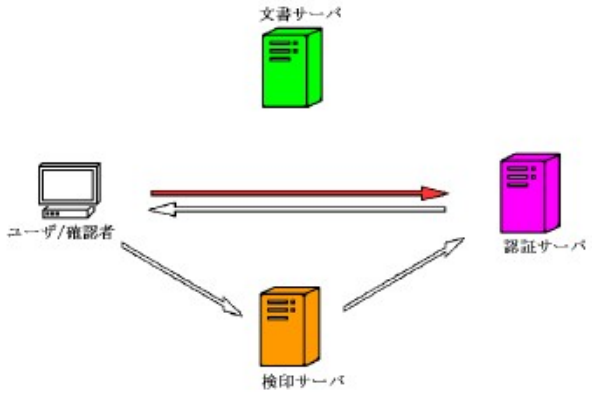
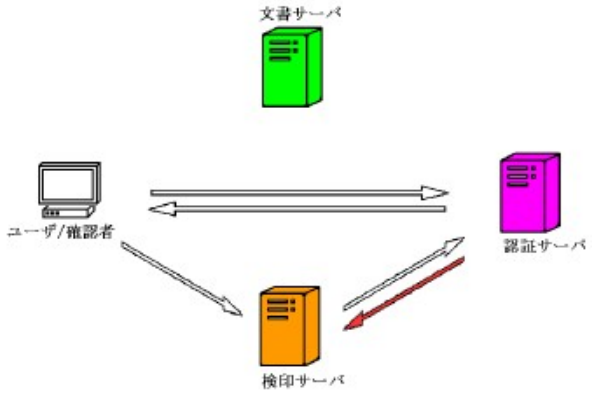
*4 学認（https://www.gakunin.jp ）

### 3.4. ログインの流れ

Web 認証基盤を導入した本システムのログインの主な処理の流れを表 4 に示す。

表４．ログインの流れ

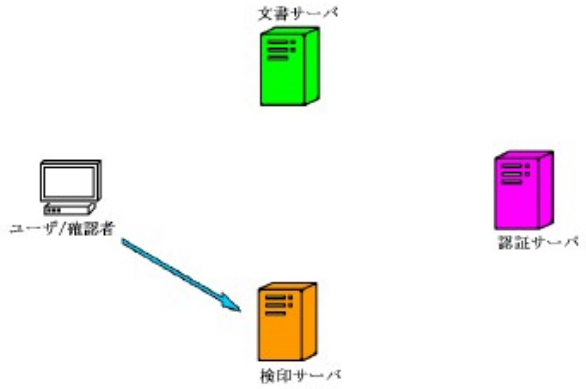
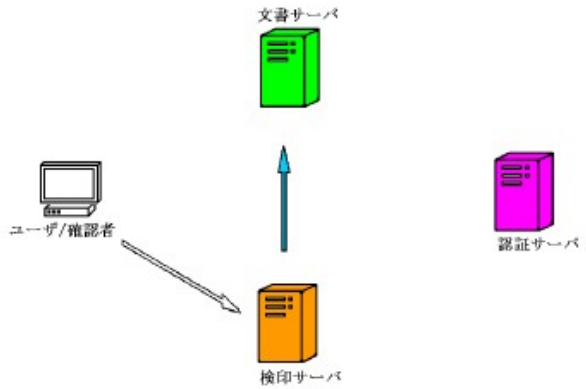
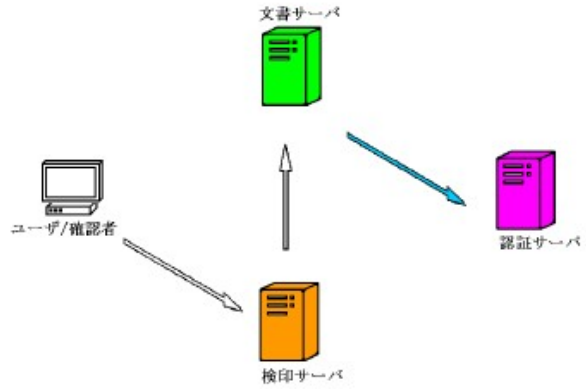
| | |
|---|---|
|  | 利用したい認証サーバの URL 形式のユーザＩＤを検印サーバに送信する（検印サーバが提示するページに入力する）<br><br>※ユーザは事前に認証サーバに ID とパスワードを登録しておく |
|  | 検印サーバは認証サーバに認証依頼を行う |
|  | 認証サーバはユーザに対し、ID とパスワードを要求する |

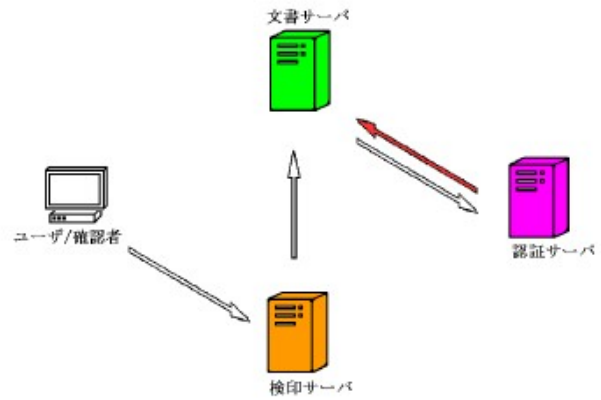
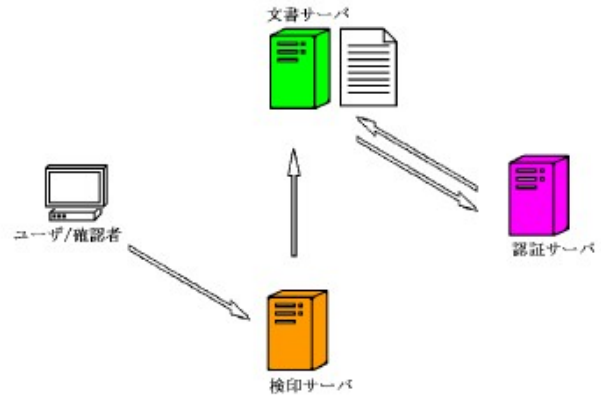
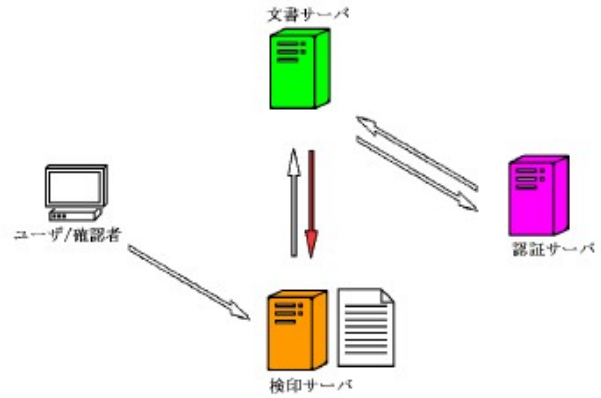
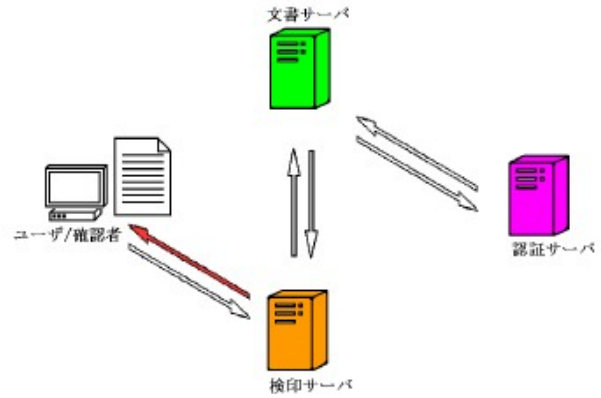
| | |
|---|---|
|  | ユーザは、認証サーバが提示するページに ID とパスワードを入力する。 |
|  | 認証サーバはユーザが入力した ID とパスワードが正しいかどうか検証を行い、認証結果のみを検印サーバに送信する。<br>このとき、認証 OK の場合は検印サーバにセッション ID も送信する |

### 3.5. 研究記録確認の流れ

Web 認証基盤を導入した本システムの研究記録確認の主な処理の流れを表5に示す。

表５．研究記録確認の流れ

| | |
|---|---|
|  | ユーザ／確認者は、検印サーバに対して研究記録を要求する |
|  | 検印サーバは、文書サーバに研究記録を要求する。その際、セッション ID（ログインの処理の際に、認証サーバから入手したもの）を文書サーバに送信する |
|  | 文書サーバは、認証サーバにセッション ID が正当なものであるかの検証を依頼する |

| | |
|---|---|
|  | 認証サーバはセッション ID の検証を行い、その結果を文書サーバに送信する |
|  | 認証サーバから送られてきた認証結果が OK な場合、文書サーバは、文書サーバのデータベースから指定された研究記録を取り出す |
|  | 文書サーバは取り出した研究記録を検印サーバに送信する |
|  | 検印サーバは、確認用のメニューと合成し、ユーザに研究記録を提示する。そして、ユーザ／確認者が内容を確認し、” 確認しました”ボタンを押すことによって確認者による確認署名の処理が完了となる。 |

## 4. 今後の課題

### 4.1. セキュリティに関する課題

本システムではいくつかのセキュリティに関する課題が存在する。正真性の保障について、本システムでは記事投稿時に使用する各ユーザの秘密鍵は DB による保管を行っている。そのため、ユーザの秘密鍵を IC カード及び USB メモリ等の外部記憶メディアに保管し、記事投稿時に限りそれを参照することにより、正真性の保障を行うことができるであろう。また、本システムにおいて、成りすましが行われる可能性がある。この問題に対しても、ユーザの秘密鍵を IC カード、USB メモリ等に保管することにより、成りすましによる記事の投稿を防ぐことができる。

非改竄性の証明の強度向上のため、リンキングやヒステリシス署名、履歴交差などの技術の検討も必要であろう[3、13]。関連して、現在 GnuPG に依存したシステムである為、多様な署名方式に対応可能とすることにより、利便性とともに非改竄性の向上が期待できる[14]。また、現在 GnuPG 使用時、ハッシュ関数 sha-1 を用いているが、暗号アルゴリズムの 2010 年問題を考慮し、sha-2 等のハッシュ関数に変更する必要がある。

電子署名の長期利用に関して、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[15、16]。同様に一定期間ごとの検証記録を保存することにより、どの時点まで正真性が保持されていたのかを証拠として残すことも今後必要となってくるであろう[17]。

現状では、検印サーバの署名及び確認者署名処理時、サーバ間通信において、記事やデータそのものによる通信を行っている。そのため、情報漏えいの危険性が挙げられる。現在は通信時に SSL を使用することで対処を行っているが、情報漏えい対策としては、十分とは言いがたい。記事やデータのハッシュ値のみによる通信に限る事も含めて、今後検討を要する。また、文書サーバと検印サーバ間の通信時にエラーが起きた場合に、DB への保存を破棄するなどの対処も必要である。

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。

### 4.2. ユーザインタフェースに関する課題

研究支援という観点から、研究記録中での画像や表・グラフ、化学式への対応が不可欠である。このような問題については、Jipsen の ASCIIsvg[18]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。

現在は記事投稿の際にグループを全て表示し指定することなどに対しての、DHTML および Ajax 技術などを用いたユーザインタフェースへの改善も今後の課題である。

## 5. おわりに

本研究では、検印サーバを介して文書サーバへアクセスする際の ID・パスワードの盗難を防止することができ、部分的ではあるがシステムのセキュリティ向上を行えた。

今後は、4章で取り上げた本システムの課題に対応し、携帯情報端末との連携を行う。本システムによって蓄積される研究記録を対象として自然言語処理・知的処理を行う。

## 6. 謝辞

本研究にあたり、最後まで熱心なご指導を頂きました小林聡准教授に、心より御礼申し上げます。

## 7. 参考文献


[1]原田　篤史,西垣　正勝, 曽我　正和, 田窪　昭夫,「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), pp268-273,2003

[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]加藤　未来,「GnuPG を用いた研究記録管理・公開・検証システム構築」, 島根大学卒業論文,2008.

[10]加藤　未来, 小林　聡,「ar χ ves: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」, 学術情報処理研究, No.12, pp.43-51, 2008.

[11]島貫稚華,「研究記録管理・公開・検証システム ar χ ves の数式およびグラフへの対応」,島根大学卒業論文,2009.

[12]岡崎、隅蔵編,「理系なら知っておきたいラボノートの書き方」,洋土社,2006

[13]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

[14]山田　竜也,宮地　充子,双紙　正和,「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.4, No.8,pp 2102-2109,2000

[15]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,vol.44,no.8,pp1955-1969,2003

[16]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」予行集,pp.983-988,2003

[17] 電子商取引推進協議会,認証・公証 WG,「電子署名文書長期保存に関するガイドライン」,2002

[18]Peter Jipsen,"ASCII MathML",http://www1.chapman.edu/~jipsen/asciimath.html

[19]黒田　努,佐藤　和人　共著,株式会社オイアクス　監修,「基礎 Ruby on Rails」,インプレスジャパン

[20] 加藤康,「研究記録管理・公開・検証システム ar χ ves への確認者署名機能の付加」,島根大学卒業論文,2010.