信頼性の検証が可能な認証システムの設計と運用

平塚 紘一郎 †　大垣内 多徳 ‡,†　田中 光也 †
福井大学 総合情報処理センター †
福井大学 医学部附属病院 医療情報部 ‡

概 要

多くの組織において、組織内の認証システムを「単一パスワード」もしくは「シングルサインオン」システムに代表される統合認証システムへと変更を行っている。これらの統合認証システムは、利用しているシステムのセキュリティ上、非常に重要な役割を果たしているものの、登録されているデータの正当性については、運用者の信頼性によっており、客観的な第三者による評価は行われていないものと考えられる。しかしながら、電子文書法が施行されたとにより現実的となった、電子文書として作成された文書の真正性や責任の所在を担保するために認証システムを利用する場合、裁判等の場面で、記録された ID の登録手続きの正当性の証明を要求される可能性があるものと考えられる。今回、我々は認証システムへのデータ登録について、電子署名およびタイムスタンプを用いることで、第三者による検証可能なシステムを考案し、運用を開始したので報告する。

An implementation of authentication system which 3rd party could validate the reliability

Kouichirou HIRATSUKA[†], Tatoku OGAITO[‡,†]　Mitsuya TANAKA[†]
Center for the Computing and Network Services, University of Fukui[†]
Department of Medical Informatics, University of Fukui Hospital[‡]

Abstract

Many organizations begin to convert their authentication system to universal authentication system, which is represented in “Password synchronization system” or “single sign on system”. The universal authentication system is one of the most critical pieces of the security of the systems, which rely on the authentication system. But in many cases, the reliability of the authentication system highly depends on the reliability of the system administrators and the inspections by 3rd parties are not made.

Since the digital archives law has been in force, one would uses the authentication systems to guarantee the genuineness of documents and/or the responsibilities to the ones. When questionable matter occurs in the “uid” data themselves, the administrator will be required to proof the justifiability of the registration procedure.

We have implemented a system which 3rd party could validate the registration procedure to the authentication system by using digital signature and time stamping protocol and started the operation. The report on the implementation and the running will be made.

1　はじめに

現在までに、コンピュータを利用したシステムが多数考案/構築されており、組織内活動や生活の様々な場面で利用されている。このようなコンピュータシステムのうち、個々の利用者を識別してサービスを行う必要がある場合は、利用者を特定するために認証が行われる。コンピュータシステムでは、あらかじめシステムに登録してある利用者番号 (id) とパスワード組み合わせを提示することで認証を行うことが多いが、多数のシステムの利用者になると、この id とパスワードの組合わせが多くなり、利便性を損なう事が増えてきた。このため、LDAP 等を利用した統一認証システムを構築して複数のシステムでの認証を同一 id、パスワードの組合わせで行うことで、利便性を回復する試みが広く行われている。

このような統一認証システムを利用する場合、従来は各システムの管理者が行っていた利用者登録について、各システムでは行わないことになる。多くの場合、これはコストの低減につながるため歓迎されることであるが、利用登録の正当性については統一認証システムを全面的に信頼することに等しい。

すなわち統一認証システムは、複数のシステムのセキュリティ上、重要な位置を占めている事となる。それにも関わらず、統一認証システムの運用について外部評価を行うような試みは報告されておらず、多くの場合、運用者の自覚や責任感により、その機能が維持されているものと考えられる。

一方、電子的に作成される文書において、作製者を特定するために認証システムにより認証を受けた id を利用する事は自然な発想である。しかし、これらの電子文書が電子文書法で規定される法的な文書であった場合、その id に関連する登録手続きについて、第三者からも理解される客観的な評価を求められる事態が起こり得る。

このような事態に対応するために、我々は、福井大学総合情報処理センター認証システムに、第三者による登録データの正当性の評価が可能な枠組みを導入し、運用を開始した。

2　データ作成者と運用者の分離

認証システムは、クライアントから提供されるデータを登録されたデータと照合することで、認証の可否を判断し、結果をクライアントに回答する。この時、考慮すべき重要なことは照合を行うデータベースに登録されているデータが本来登録されるべきデータであるかどうかを、認証システムの利用者はもちろん、認証システム自身も判断できないという点である。データベースに登録されたデータは、その登録内容や手続きが正当であるか否かを問わず認証要求に対して照合が行われ、内容が一致すれば認証されるのである。

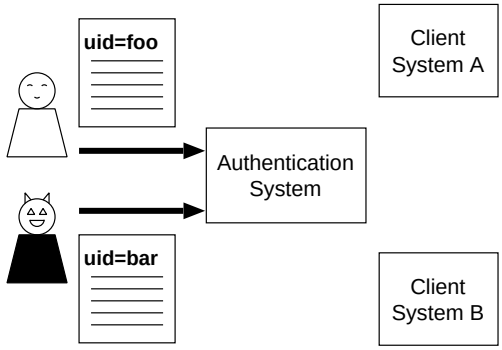

図 1: 認証データベースへの登録

すなわち、図 1 において、正規に登録された `uid = foo` と、不正に登録された `uid = bar` は認証システムのクライアントからは区別できない事になる。

小規模な組織を除けば、組織構成員を把握している人事部門と、情報システムを運用する部門は異なっているのが一般的であると考えられる。認証システムがサイバー空間での組織構成員の証明を行っている場合でも、その運用は専門的知識が必要等の理由で情報部門により行われている場合が多い。このような運用形態の場合、認証システムを運用している情報部門においては、その登録内容について全体を把握することは非常に困難であり、前述のような問題を見過ごす可能性が高い。我々はシステムの運用者とデータ作成者の権限を分離することで、この問題に対応することが可能であると考え、システムの設計を行った。

具体的には、認証システムへのデータ登録においては、関連するすべてのシステムが信頼すべき認証局により発行された公開鍵証明書を用いて検証できる、電子署名が行われたデータのみを有効とし、それ以外のデータに関しては認証データベースに登録できないこととした。

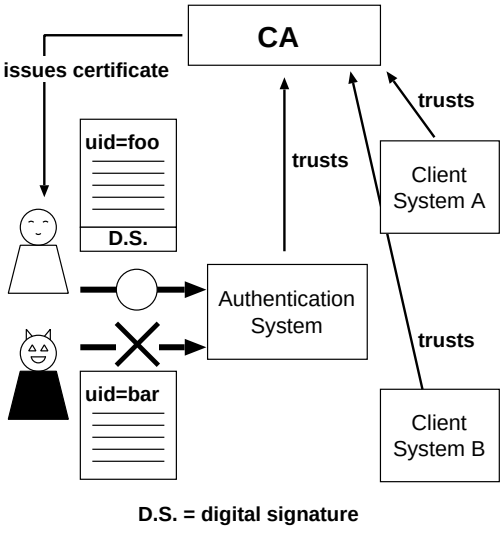

図 2: 電子署名を伴う認証データベースへの登録

この際に必要となる証明書を発行する認証局としては、対象とした認証システムが大学内に対する認証サービスを行うことを念頭に置いて、総合情報処理センターが運用する認証局を用いることとした。

3　データ投入の時刻の記録

認証データベースへの登録に用いられたデータの作製者を特定できるだけでは検証作業としては不十分である。電子記録に残されたアカウントの正当性が問題となるような事態においては、認証に用いられる個人識別子が、どのユーザに対して与えられていたかを検証する必要が生じる。これは、認証に用いられる個人識別子が再利用される場合にだけ問題になるのではなく、利用者の他の識別子 (姓名、身分、職種等) が変更される場合においても必要となる可能性が高い。そのため、実際に投入された変更データを記録した上で、その log に対して、RFC 3161 に基づくタイムスタンプを取得することで、データ投入時刻の記録を行う事とした。

タイムスタンプを発行するタイムスタンプサーバとしては、以下の理由により総合情報処理センターが運用するタイムスタンプサーバを用いることとした。

1. 公開鍵証明書を発行する認証局が、総合情報処理センター運営の認証局であること。すなわち、実在性の担保が自組織内で行われているのであるから、時刻の担保も同等で十分であると考えた。

2. 時刻の精度としてはそれほど厳密である必要がないこと。すなわち、新規アカウントの申請や登録内容の変更申請を利用者が行った後、実際にデータベースへの反映が行われ、その結果を利用者へ伝達することを考慮した場合、現状では日単位の精度で十分であると考えられる。

3. 商用タイムスタンプサーバを利用する場合のコストの問題。タイムスタンプをアカウント管理作業を行うごとに利用するため、継続的にタイムスタンプ発行費用を負担する必要がある。費用負担を行ってまで外部タイムスタンプサーバを利用した場合、現在のところタイムスタンプの重要性が理解されていないこともあり、学内他組織への説明が困難であると考えられた。

4　実装

認証システムに用いるプロトコルとしては、LDAP を採用し、Sun Java System Directory Server(SJSDS) をサーバとして用いている。ただし SJSDS 本体に、検証機能を組み込むことは困難であったため、SJSDS とともに動作する独立したプログラムとして設計・構築を行った。そのため、投入されたデータそのものを得ることができず、SJSDS が出力する `audit.log` を用いて代用している。

次に、各場面ごとに用いるプログラムの動作について述べる

4.1　データ作成時処理

データ作製時には、作製者自身を明らかにする手続きを行う。具体的には LDIF の各エントリごとに署名者 DN と、そのエントリに対する電子署名を追加する。スキーマ拡張を行うことで、署名者と電子署名を記録するための属性を用意した。たとえば図 3 のような LDIF が与えられた場合、`uFukuiSigner` および `uFukuiSignature` という属性を各エントリに追加し、最後に全体のチェックサム等を追加し、図 4 のように変更する。

```
dn: cn=foo,ou=people,o=University of Fukui,c=JP
changetype: add
cn: foo
uid: foo
sn:  Smith
givenName: John
employeeType: parttime

dn: cn=bar,ou=people,o=University of Fukui,c=JP
changetype: modify
replace: employeeType
employeeType: permanent
-
replace: sn
sn: Raymond
-
```

図 3: 通常の LDIF

```
dn: cn=foo,ou=people,o=University of Fukui,c=JP
changetype: add
cn: foo
uid: foo
sn:  Smith
givenName: John
employeeType: parttime
uFukuiSigner: cn=baz,ou=people,o=University of
 Fukui,c=JP
```

```
uFukuiSignature: 7e07cb80163e9065922281ef5e786
 cfbf1a56979

dn: cn=bar,ou=people,o=University of Fukui,c=JP
changetype: modify
replace: employeeType
employeeType: permanent
-
replace: sn
sn: Raymond
-
replace: uFukuiSigner
uFukuiSigner: cn=baz,ou=people,o=University of
 Fukui,c=JP
-
replace: uFukuiSignature
uFukuiSignature: a8b87686da534b4c3c0ba94af720c
 4311275950a
-

Cksum: dcff4434e28e12bdd1d00eb2b0bcf19d
uFukuiSigner: cn=baz,ou=people,o=University of
 Fukui,c=JP
uFukuiSignature: fec028cec0f82960749b09bff265c
 588c98aba4c
```

図 4: 作製者を明らかとした LDIF

図 4 の最後のブロックは LDIF 形式に似せてあるが、認証データベースに登録されるデータではない。各エントリに加えられた署名データを含んだ全体のチェックサムおよび電子署名である。これらのデータは、後述するデータ投入前およびデータ投入後の場面で利用される。

各エントリごとに署名を追加したのは、SJSDS の `audit.log` の出力が同時に複数の LDIF が投入された場合、エントリ単位でしかまとまって出力されない事が判明したためである。

4.2 データ投入時処理

前節で作成された LDIF は、任意の経路で認証システム運用者の元に届けられ、運用者によって、認証システムへの登録手続きが行われる。この場面では運用者によって起動されるプログラムと、`audit.log` を監視することで実際に投入されたデータの検証を行うプログラムが協調して、次のように動作する。(図 5 参照)

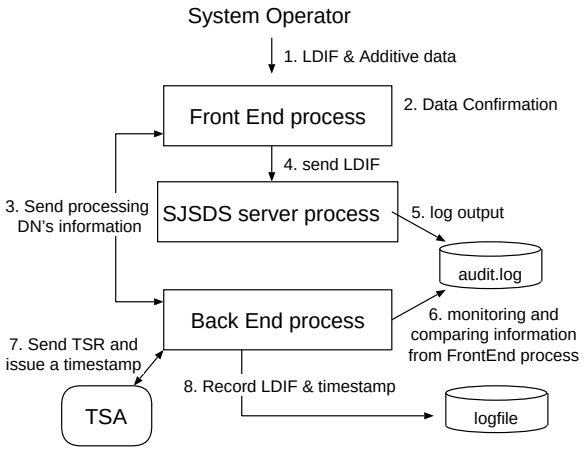

図 5: データ投入時のプログラムの振舞

1. システム運用者が送付を受けたデータを FrontEnd プロセス (以下 FE) に対して入力する。
2. FE は、LDIF の最後に追加されたブロックを用いて、LDIF 作製者が作成したデータと異なっていないかの検証を行う。
3. 誤りがなければ、FE は入力を各エントリごとに分解し、エントリの dn 情報を BackEnd プロセス (以下 BE) に送信する。 BE は、audit.log が監視できていることを確認し、受信した dn に関する情報が出現するのを待つ準備ができた段階で、FE にデータ投入許可を送信する。
4. FE はデータ投入許可を受信後、LDAP プロトコルを用いて SJSDS サーバプロセスに認証データの変更を行う。
5. SJSDS サーバは、処理が成功した場合、`audit.log` へ処理を行った LDIF を出力する。
6. BE は `audit.log` の出力から元の LDIF エントリを再構成し、含まれる署名者情報および署名を用いて、データ作成時から変更されていないことを確認する。
7. 再構成された LDIF に対するタイムスタンプを時間認証局 (Time Stamp Authority: TSAq) から取得
8. 再構成された LDIF およびそれに対するタイムスタンプをログファイルに記録

9. データが終了するまで 3 から 8 を繰り返す。

署名者情報および署名は、データ登録後も各エントリの属性として記録され続けるが、一般に記録された値は、その時点の当該エントリの内容とは関係がないことには注意しなければならない。

また、FE と BE が SJSDS を通すことなく通信を行うことで、作製された LDIF の一部分が欠落した場合に検出が可能であるようにしている。

なお、BE は常に `audit.log` を監視しており、不適切な出力が発見された場合、管理者への通知等を行う機能も合わせて持っている。

4.3 監査時処理

認証システムに登録されたデータに疑義が生じた場合等の監査が必要となった場合は、次のような手続きで検証が可能である。

1. タイムスタンプとともに記録された logfile から該当 dn についての変更記録を、初期登録時からすべて抜き出す。
2. 各々の変更記録について、入力データに対するデータ作製者の電子署名が一致することを確認する。
3. それらのデータに対して記録されたタイムスタンプを検証することで、変更が行われた日時を特定する。
4. 初期登録データにすべての変更を施したものが、現状のデータと一致しているかを確認する。

現状では、具体的にこのプロセスを実装したプログラムは存在しないため、必要な事態が生じた場合は、上記のプロセスを手作業で行わなければならない。

5 運用と課題

2007 年 4 月より本システムを利用して、総合情報処理センター内のサービスを行い、データ作成および投入場面については動作している。監査については 2007 年 8 月現在行われておらず、今後実際に検証を行うことで正しく動作していることを確認する予定である。

設計段階を含め、これまでに判明した問題について以下に議論を行う。

5.1 設計上の問題

本システムでは、認証システムに組み込むのではなく、別プログラムとして設計を行った。すなわち、認証システムの「運用ルール」として、本来使用する `ldapmodify` 等を原則利用しないようにしている。

したがって、運用ルールが守られなかった場合は、未署名データが投入されるため、監査時に問題となる可能性が考えられる。これに対応するために、暫定処置として LDAP の特権ユーザの利用について、そのパスワードを知る管理者を厳しく制限しているが、今後これが十分な処置であるかどうかの検討が必要である。

5.2 LDAP プロトコルの問題

LDAP には transaction が規定されていない。そのため、処理が途中でエラーとなり終了した場合、問題になる可能性がある。

第一に、入力 LDIF の後ろ部分が、故意または事故により入力されなかった場合と区別がつかないため、認証システムに対する不正が行われた事象と判断される可能性がある。

第二に入力データ作製者は LDIF 全体が投入されるものと仮定してデータを作成している (例: YYYY 年 4 月 1 日現在のデータ) ものと考えられるため、一部の処理のみが行われた状態は、認証システム全体の整合性が保たれないこととなる。

このような事態が発生した場合には、データ作製者とシステム運用者とで認証データベースの内容の点検を行う必要がある。

5.3 署名の必要性の検討

本システムでは設計段階から、**登録者自身による登録データの変更は署名を必要としない**事としていた。これは、本人によるパスワードの変更を考慮したためである。この原則の元、設計・運用を行ったが、次のような問題が考えられている。

- 「どのホストに対する利用権限を与えるか」等の総合情報処理センター運用場面でのみ利用する属性の変更にも電子署名を伴ったデータ変更が必要なのか？
- 登録者自身による変更については署名を必要としないとしたが、悪意を持った登録者が姓名等を勝手に変更することが許されるのであろうか？

第 1 の問題に対しては、センター内部でのみ利用する属性については、署名無しで変更可能であるように Backend Process が不正な記録と判断しないようにして対応した。

第 2 の問題に対しては、総合情報処理センター配下の特定のマシンからのみ変更要求を受け付けるように制限を行う事で対応している。その上で総合情報処理センター職員の立ち会いのもとでのみ、基本情報を変更可能なシステムを構築中である。

5.4 公開鍵証明書との関連

公開鍵基盤における公開鍵と持ち主とのつながりを保証するために発行される公開鍵証明書は、サイバー空間における身分証明書であり、本来、その発行時には身元審査等を行い、発行対象が間違いがないことを確認する。

そのため、公開鍵証明書を発行するための入力データとして、認証システムのデータを用いることは本末転倒である。しかしながら、本稿で述べた検証機能を応用し、データ作製者を人事部門等の実空間における身分証明書の発行部署に制限することで、登録内容の正当性を担保し、公開鍵証明書の発行における入力データとして利用することも可能となり、業務の軽減に貢献できる可能性があると考えられる。

6 まとめ

福井大学総合情報処理センターでは、大学全体にたいしてサービスが可能な認証システムを構築した。その際に、認証システムに登録されたデータに対する責任者や、その変更がいつ行われたかを追跡可能であるような機能を追加した。このことで、認証サービスを利用するクライアントシステムに対して、その登録手続きを客観的に示すことが可能となった。

これにより、電子的記録の作製者の特定に認証システムを利用しても、その個人識別子の登録手続きが明らかとできるため、電子文書法等に関連した事態が発生した場合、客観的な説明が可能となるものと考えられる。

今後は、認証システム自身に本機能を実装する事を計画するとともに、手続きを標準化し、本機能がひろく利用されるよう提案していくものである。

参考文献

[1] RFC 2251: Lightweight Directory Access Protocol (v3)

[2] RFC 3161: Internet X.509 Public Key Infrastructure - Time-Stamp Protocol (TSP)

[3] ITU-T Recommendation X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks