Construction and Use Examples of Private Electronic Notary Service in Educational Institutions

Masanori NAKAKUNI
Research Institute for Information Technology
Kyushu University
6-10-1 Hakozaki, Higashi-ku Fukuoka 812-8581
JAPAN
nakakuni@cc.kyushu-u.ac.jp http://nlab.tv

Eisuke ITO
Research Institute for Information Technology
Kyushu University
6-10-1 Hakozaki, Higashi-ku Fukuoka 812-8581
JAPAN
itou@cc.kyushu-u.ac.jp

Yoshiaki KASAHARA
Research Institute for Information Technology
Kyushu University
6-10-1 Hakozaki, Higashi-ku Fukuoka 812-8581
JAPAN
kasahara@nc.kyushu-u.ac.jp

Sozo INOUE
Library
Kyushu University
6-10-1 Hakozaki, Higashi-ku Fukuoka 812-8581
JAPAN
sozo@lib.kyushu-u.ac.jp

Hiroshi DOZONO
Faculty of Science and Engineering
Saga University
1-Honjyo Saga 840-8502
JAPAN
hiro@dna.ec.saga-u.ac.jp

Abstract: - People have many documents. For example, a variety of documents are prepared and used in public institutions. As the internet becomes widely available in recent years, paper documents are being replaced with electronic data, which are often distributed in the form of electronic data without being printed out. Similarly, in educational institutions, increasing number of documents are distributed in the form of electronic data. Such data are distributed through various routes and means, and prone to the risk of alteration in the process. Data may be protected against alteration, but it is difficult to completely prevent data alteration in the distribution process. Data can be generated with electronic signature that allows for the identification of data creator and possible alterations by third parties. This method is, however, not valid if the data becomes separated from the electronic signature,

making the validation of data creator or data alterations difficult or impossible. In this paper, we describe the invention of a system that, even in cases where data is separated form the electronic signature, enables easy identification of possible data alterations by the electronic signature management. And we describe here an exploratory construction of private electronic notary service in university. We also add a review on the utilization method of private electronic notary service in universities.

Key-Words: - Notary Service, Electronic Signature, e-Learning, Digital Data Management, PKI

1 Introduction

As the internet becomes widely available in recent years, academic educational environment has changed significantly. For example, as e-Learning system becomes more widely used, lecture materials are being switched from papers to electronic media. Recently, some universities are making attempts to distribute educational video data through their websites. In addition, changes are also seen in the receiver side of education. In a conventional system, students at University A were able to take only those lectures given at University A. In these days, introduction of credit transfer system has enabled flexible curriculum that allows students to take lectures of other universities pre-approved in interuniversity exchange program. University alliance, that is to say that the students of University A can take lectures of University B and vice versa, is becoming more common. Moreover, lectures are not limited only to students at pre-approved universities within the interuniversity exchange program, but also available to residents in the neighboring communities.

Lecture materials will be distributed even wider than ever if the lectures are to be taken not only by the university constituents but also by people outside the university. This means that the materials are more prone to alteration in the distribution process. Part of lecture materials may get dropped or miswritten in the process of passing from person to person. The causes are various including accidents and intentions, but the alterations of lecture materials without permission of the creators are likely to be disadvantageous to the creators and the viewers as well. For example, if a lecture material gets altered in a way the content is contradictory to the fact, the creator as an academic staff of the university and the university itself might lose credibility. Therefore, the creator of lecture materials must take measures against such problems if he/she is to distribute the materials in a form of easily-copiable electronic data.

2 Data management

2.1 Conventional data management and its problems

In conventional data management, anti-alteration protection was the main focus in the preventive measures against data alteration. However, prevention of data alteration is quite difficult. Even a non-editable ODF file may be forged into a closely-resembled PDF file, which is almost as same as alteration. Video data, created with non-editable setting similar to PDF, may also be converted to an editable format by using video capturing software [1, 2]. That is, even those data with highly-protective functions can be easily altered by counterfeiting because the contents are always displayed to the viewers.

2.2 Data management method in universities

In educational institutions including universities, strict management is often necessary in order to prevent alteration of data distributed by themselves. It calls for a system that ensures that the data was created by faculties or staff of the university, and that the data has not been altered. One way to do this is to utilize an assurance system similar to notary service [3, 4, 5]. In this case though, strict assurance by third party is not required and the university itself will give the assurance. Then the cost can be cut down through self-management of the notary service by the university, eliminating external commission expense. Also, all that is required is to provide a method which allows the viewers to check the possible presence of data alteration. A private electronic notary system as simple as this can be constructed and operated at low cost. Our proposed system was named PENS from the initials of “Private Electronic Notary System”.

3 Data management by electronic signatures

3.1 Advantages of electronic signatures

The above-described requirements can be satisfied by the attachment of electronic signature [6] to data. A brief summary is as follows.

- Validation of data creator
  - Data creator can be identified.
- Verification of data alteration
  - Possible presence of data alteration can be checked.

3.2 Common electronic signature methods

One of the common electronic signature methods is to embed an electronic signature in the data itself. In this case, the software that handles data must support electronic signature embedding. Software may be self-tailored to support the signature embedding, but it is often difficult in cases where the technical specifications for the software are not open. Even in cases where the technical specifications are open, the specifications may undergo changes through version upgrades, which may necessitate changes of electronic signature method and complicate the adjustment process for the continuous support of new specifications.

Another method of electronic signature is PKI (Public Key Infrastructure) [7, 8, 9, 10]. PKI uses secret key for the attachment of electronic signature, and public key for the verification of electronic signature. In this method, electronic signature is attached in a separate form without being embedded in the data, thereby giving an advantage that the electronic signature can be applied to any formats of data. However, if data gets distributed separately, data verification or assurance is no longer possible. In order to prevent such events, electronic signature method has to be evaluated based on the premise of separate distributions of data and electronic signature.

3.3 PENS Electronic signature method

In the PENS electronic signature method, the data creator generates a message digest from the data by using hash function [11, 12, 13], and then register the message digest and the related information into the PENS database. PENS is therefore similar to PKI, in the way that data and the corresponding electronic signature are separated, giving an advantage that the electronic signature can be easily applied to any formats of data. The PENS database will be managed by the university. And the message digests and the related information registered in the database will be released to the viewers. The viewers will, in accordance with prescribed procedures, compare and verify the message digest of the obtained data with that registered in the database, and thereby obtain information on the data creators and possible alterations.

3.4 Manipulations of PENS electronic signatures

There are three kinds of manipulations for PENS electronic signatures; “Attachment of electronic signature”, “Invalidation of electronic signature” and “Verification of electronic signature”.

Procedures for the attachment of electronic signature is shown in Figure 1. In order to attach an electronic signature, the data creator will access PENS through a web browser. The user will be prompted to enter his/her user ID and password for PENS electronic signature attachment, and the user must be pre-approved for the signature attachment. User ID will be provided to faculties, staff, and others that were approved by the university. First, the user will enter his/her PENS user ID and password to login. Next, the user will upload data to which an electronic signature is to be attached. Finally, comments may be added as needed. Attachment of electronic signature is completed through the procedures above.

Procedures for the invalidation of electronic signature is shown in Figure 2. Invalidation of previous electronic signature is necessary upon data revision. In the process of invalidation, the previous signature attachment history is kept in record while the signature invalidation history is added. Procedures are similar to those for electronic signature attachment. The user will login to PENS, upload data of which the electronic signature is to be invalidated, and add comments as needed. Invalidation of electronic signature is completed through the procedures above.

Procedures for the verification of electronic signature is shown in Figure 3. User ID and password are not required for the verification of PENS electronics signature. That is, anyone can use the verification function. The user is only required to upload data which is to be verified. If the data has a PENS electronic signature attached, information of the creator’s name, etc. will be displayed on the web

browser, which allows verification of the data creator and possible alterations.

3.5 Information registered in PENS database

The following information concerning signature-attached data are registered in the PENS database.

- Message digest
- Registered date in PENS
- Name of the data creator
- Department/division to which the creator belongs
- E-mail address of the creator

In addition to the information above, other information may be added as needed. Data itself is not saved in the PENS database.

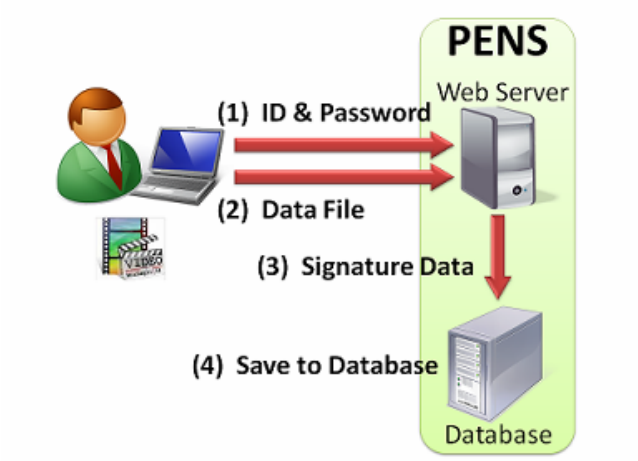

Figure 1 Procedures for the attachment of electronic signature

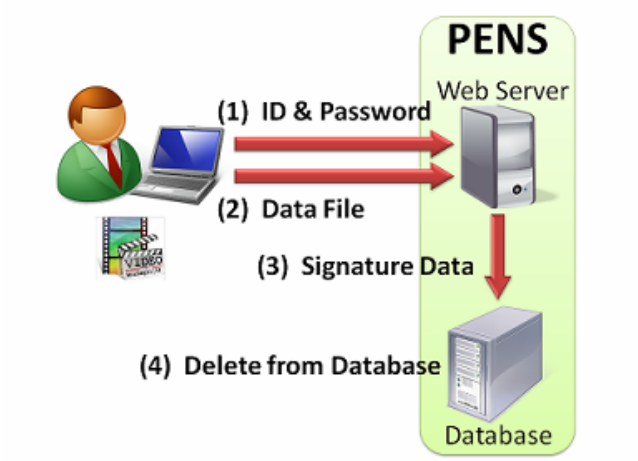

Figure 2 Procedures for the invalidation of electronic signature

4 Construction of a prototype PENS

4.1 System construction

System architecture of the prototype is shown in Table 1. The system architecture is also shown in Figure 4.

4.2 PENS operation screens

PENS operation screens are depicted below. Figure 5 shows the PENS login screen. Figure 6 shows the screen for the attachment of electronic signature. Figure 7 shows the screen for the invalidation of electronic signature. Figure 8 shows the screen for the verification of electronic signature. Figure 9 shows the screen for the result of verification.

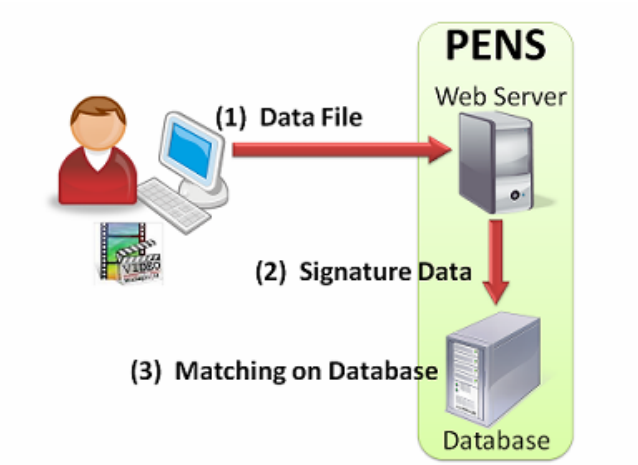

Figure 3 Procedures for the verification of electronic signature

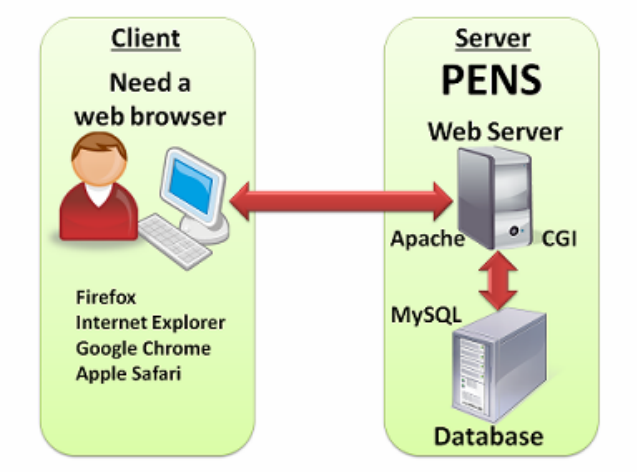

Figure 4 The system architecture of PENS

Table 1 System architecture of the prototype

Hardware	
CPU	Intel Xeon
Memory	2GByte
HDD	250GByte
Software	
OS	CentOS 5.1
Web Server	Apache 2.2.3
DB	MySQL 5.0
Development Language	Perl 5.8.8
Hash Function	MD5

4.3 Handling of large files in PENS

As mentioned above, in PENS, data is uploaded onto the server for electronic signature attachment, and the server performs processing task for creating message digest according to the uploaded data. Similarly, invalidation and verification of electronic signature also requires uploading of relevant data to PENS server. Data upload to PENS server is time consuming if the data size is several hundred megabytes or larger. In addition, network load also becomes heavy due to the increased data transmission through the network. Moreover, creation of message digest for larger file causes a heavy load on PENS server. Large-data upload is not a significant problem if it is infrequent, such as a few times a day. But if large-data uploads count up to several dozen times a day, or converge in a short period of time, PENS server and the related network may experience overload and suffer adverse impacts. And the system will face a greater risk of various failures. In order to avoid such risk, appropriate measures have to be taken to distribute load on PENS server and network. We considered a method in which message digest is created on each client terminal and the resulting message digest alone is sent to PENS server. This means that a part of processing tasks on PENS server is allocated to terminal computers on the user side. In this method, PENS server load is distributed and network load is also reduced by omitting data upload to PENS server. Hence, the above-mentioned problem can be solved. We developed PENS client that realizes this method.

4.4 PENS client

Detailed description of PENS client is given below. As mentioned above, PENS client is a software that enables a local computer to create message digest and to send the resulting message digest alone to PENS server for the attachment of electronic signature. The development environment and system requirements for the PENS client are shown in Table 2.

Table 2 The development environment and system requirements for the PENS client

Development environment	
OS	Microsoft Windows Vista
Programming Language	Hot Soup Processor (HSP) Version: 3.1
System requirements for PENS client	
OS	Microsoft Windows 2000, 2003, XP, Tablet PC Edition, Vista
CPU	Intel 1.3 GHz processor
RAM	128MB of RAM (256MB recommended)
HDD	5MB of available hard disk space

Host Soup Processor (HSP) [14] is a programming language that was developed by a Japanese, Onitama, and are provided free of charge. Currently, development is underway in the scope of an open source implementation. HSP supports a variety of plugins to add extended commands to be used in programs, and most of these plugins are provided free of charge. Grammar of the programming language is similar to that of BASIC. Increasing number of primary and middle school students in Japan use HSP due to its grammatical simplicity. One of the notable features of HSP is that highly

functional software can be developed with short and simple codes.

Next, start up screen of PENS client is shown in Figure 10. Basic flow of operation procedures is similar to those described in 3.4. First, specify a URL for accessing PENS. ID and password are also required in case of attachment or invalidation of electronic signature. Next, specify the relevant data pass, and select from the pull-down menu an operation to be performed - electronic signature attachment, invalidation, or verification. In case of electronic signature attachment, enter your name, affiliation, and other information such as data description at the same time. Lastly, click "Run" button, and the program will initiate the creation of message digest, communicate with PENS server, and perform the specified operation of signature. Communications between PENS server and client are secured with SSL data encryption using the https protocol.

Figure 5 The PENS login screen

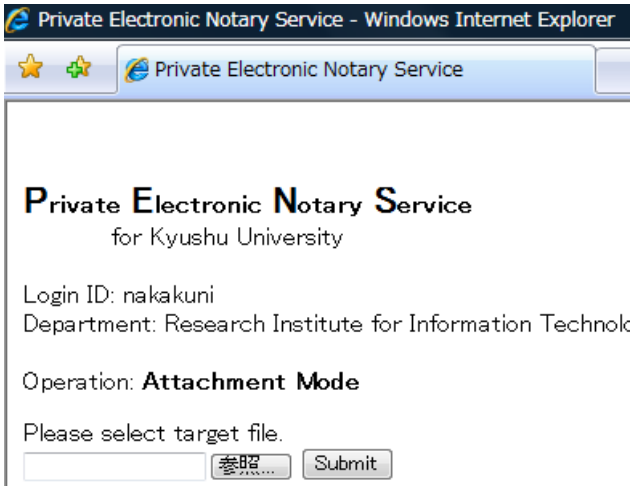

Figure 6 The screen for the attachment

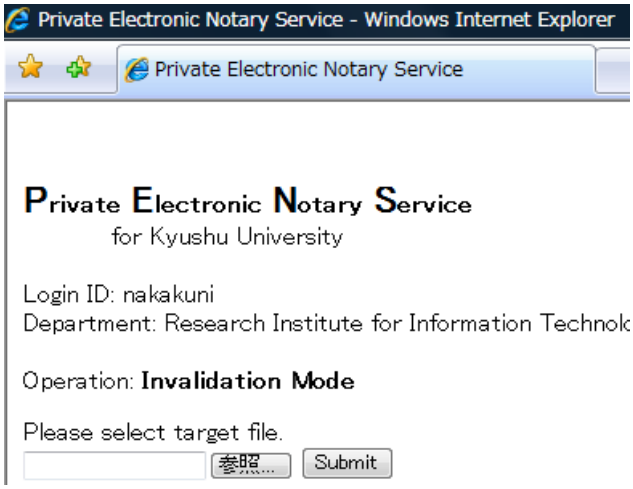

Figure 7 The screen for the invalidation

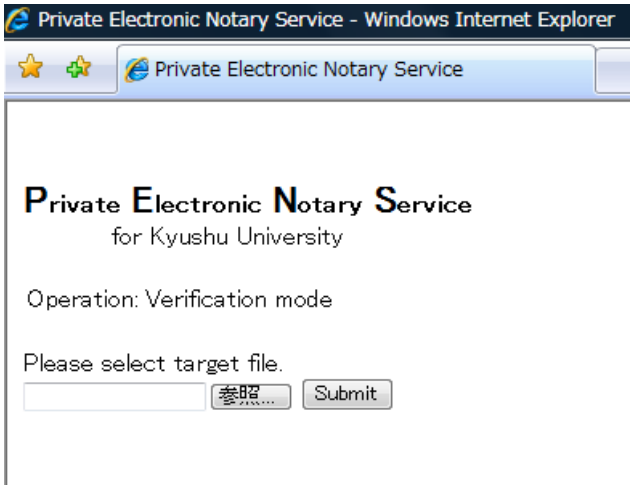

Figure 8 The screen for the verification

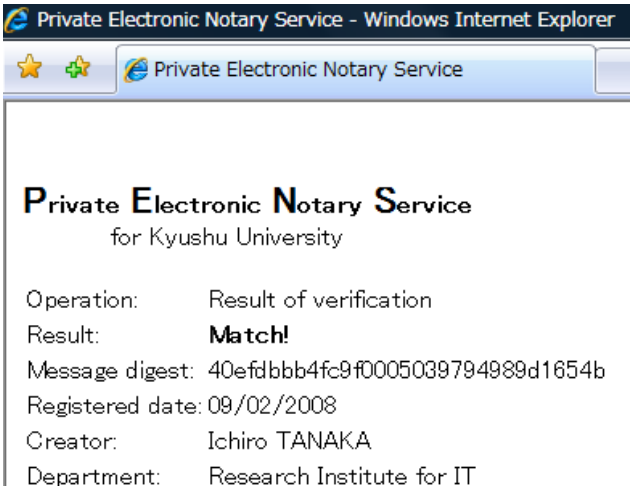

Figure 9 The screen for the result of verification

4.5 Technical advantages and disadvantages in PENS

Advantages and disadvantages in the technical aspect of PENS are listed below.

- Advantages
  - Electronic signature can be applied to any formats of data.
  - Data can be easily verified and assured even in cases where data becomes separated from the electronic signature.
  - Simple and compact system enables easy operation.
  - Simple architecture enables easy manipulation of the system.
  - Costs of system introduction and operation are low.
- Disadvantage
  - Data alterations cannot be prevented.

5 Example of PENS usage

5.1 Electronic signature attachment to notifications

Most organizations, including but not limited to educational institutions such as universities and colleges, frequently issue various notifications internally and externally. Such notifications may be posted on website or sent to specified individuals via e-mail in the form of electronic data. It is difficult to verify the provider of notification in either case. In other words, those who view a notification cannot easily confirm the information provider. The issue may be addressed by PENS. The provider's identity can be confirmed by PENS for any notification issued with PENS electronic signature, regardless of distribution routes and means, provided no modifications are made. PENS is highly versatile because its electronic signature attachment is supported by most file formats including simple text format. In addition, PENS will play a more significant role when the notification has higher priority.

5.2 Electronic signature attachment to software

In educational institutions including universities, software programs are developed as part of research and education activities, and those programs are often distributed to general public in order to widely publicize the results. Software may be distributed through websites that are operated by university, or personal websites of the developers. Specialized websites for software distribution, such as SOURCEFORGE.NET [15], may be used, and there are various other ways of distribution. It's also possible that software is passed from user to user via e-mail. In these ways, software may be distributed through various routes and means, and therefore the programs and appended documents are prone to the risk of alteration in the process. Even if the developer's name is clearly stated in a document contained in the software, users cannot easily confirm the developer. Therefore, many users are afraid of running software on their computers without proper identification of the developer. It is out of fear that computer virus infections and spyware activities may cause damages. The issue may be addressed by PENS. The developer's identity can be confirmed by PENS for any software provided with PENS electronic signature, regardless of distribution routes and means, provided no modifications are made. Verification method of PENS electronic signature can be included in an appended document, so that users can easily confirm the developer's identity by themselves. Distributing software with electronic signature attachment will provide a sense of security to users by properly identifying the developer.

5.3 Electronic signature attachment to video data

Recently, it is becoming common that universities create video data to put on their websites for public access. In some cases, electronic signature is attached in order to clarify the creator. An increasing number of universities and colleges also distribute video data on YouTube [16]. However, video data

with electronic signature is not necessarily compatible with YouTube. The reason is as follows. Data is automatically and compulsorily converted to FLV format [17] in the uploading process of YouTube. At this step, some video data in a particular format with electronic signature attachment may not be converted to FLV. Or the video data may be converted but the electronic signature is erased. For example, in case an electronic watermark is embedded in the video data, the watermark may become invalid by quality degradation due to the conversion to FLV format. Therefore, video data distributed through YouTube may not contain electronic signature. On the other hand, PENS can solve the problem. First, video data will be uploaded to YouTube. Then the video data in FLV format will be downloaded from YouTube and saved to a local computer. Electronic signature can then be attached to the downloaded video data using PENS.

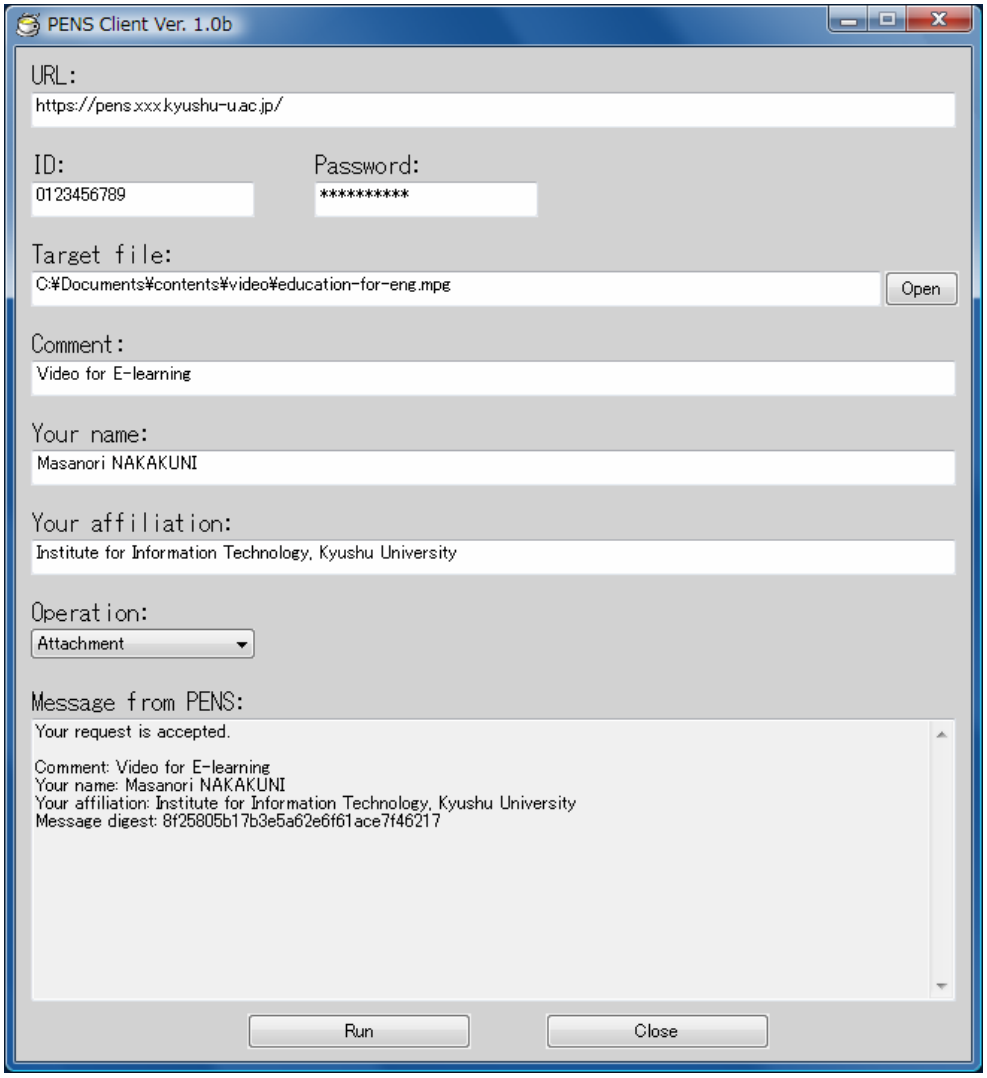

Figure 10 PENS client

6 Conclusions

In this paper, we presented the invention of an electronic signature system PENS, which was based on the premise of separate distributions of data and electronic signatures. Construction of a prototype system was also described. PENS has features that enable the data creators to easily attach electronic signatures to their data and the viewers to easily verify the data creators and possible alterations, although data alterations themselves cannot be prevented. When distributing electronic documents including lecture materials, the university will provide to the viewers information on the data creators and alterations, but no other advanced functions are necessary. In this way, PENS has a simple architecture and simple functions, and still capable of contributing to the maintenance of the university's credibility. However, there is one prerequisite that none of the PENS users operates with malicious intention. Although it is difficult to eliminate all malicious users, the first measure will be to prevent unauthorized use by masquerading users who stole user IDs and passwords. Current PENS uses only user IDs and passwords for the user authentification, which does not provide a robust security. In order to improve the situation, it is necessary to incorporate other authentification methods such as biometric identification [18, 19, 20, 21, 22, 23, 24] and IC card identification [25, 26, 27]. From the reason described above, reinforcement of the user authentification function is our main focus in the future.

We are also conducting a study on single sign-on system [28, 29, 30, 31, 32], with a view of linked operation of PENS and single sign-on system. We are particularly interested in a single sign-on system called Shibboleth [33, 34]. Shibboleth was developed by the Internet2 [35] Shibboleth Development Team, and is widely distributed as open source software. Main feature of single sign-on and the merit of its introduction are as follows. When logging in to services that are operated under a single sign-on system, user ID and password will be prompted once for the first service, and if the login state is maintained, logins to other services are smooth without repeated entering of ID and password. This is the main feature of single sign-on. Repetitive typing of ID and password can be saved by this feature. Reducing the typing tasks for ID and password also reduces the frequency of ID and password data transmission through network, resulting in lower risk of ID/password eavesdropping. In this way, single sign-on enhances the convenience and safety at the same time. We are working on a method to enhance the convenience and safety of PENS, in linked operation with a single sign-on system.

For the purpose of enhancing PENS convenience, we are also considering the implementation of a function that enables extensive searching and verification of electronic signatures by linking multiple PENS constructed at each organization. The flow is depicted in Figure 11. With this function, any electronic signature, of which the relevant PENS organization is unknown, can be verified by accessing any of the linked PENS. If no information is retrieved from the accessed PENS, the server will access other PENS and send a query to provide the user with the query result. Implementation of this function will significantly improve the convenience of PENS.

Our future objective is to further develop PENS by implementing the function as described above.

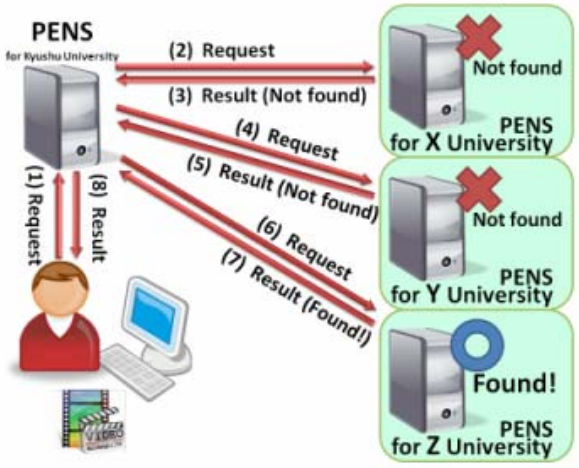

Figure 11 Linking multiple PENS constructed at each organization

References:

[1] RipTiger.com, RipTiger, http://www.riptiger.com/.

[2] dvdshrink & ATyC Group, DVD Shrink, http://www.dvdshrink.org/.

[3] Yoshida, A., Security of electronic patient recording system (EPRS) in a clinic, *Japan*

Journal of Medical Informatics, Volume 17, Issue 3 SUPPL., 1997, pp.317-318.

[4] Ruotsalainen, P., Manning, B., A notary archive model for secure preservation and distribution of electrically signed patient documents, *International Journal of Medical Informatics*, Volume 76, Issue 5-6, 2007, pp.449-453.

[5] Ota, H., Watanabe, Y., Nakao, K., Sugaya, F., Proposal and evaluation of a digital document protection scheme to specify whether the malicious entity is the sender or receiver, *Electronics & communications in Japan. Part 3, Fundamental electronic science*, Volume 88, Issue 8, 2005, pp.18-27.

[6] TechTarget, What is digital signature?, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci211953,00.html.

[7] TechTarget, What is PKI?, http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci214299,00.html.

[8] mozilla.org, Open Source PKI Projects, http://www.mozilla.org/projects/security/pki/.

[9] Stefan Kelm, The PKI page, http://www.pki-page.org/.

[10] Park, K.-W., Lim, S.S., Park, K.H.: Computationally efficient PKI-based single sign-on protocol, PKASSO for mobile devices, IEEE Transactions on Computers 57 (6), 2008, pp. 821-834.

[11] TechTarget, What is hashing?, http://searchsqlserver.techtarget.com/sDefinition/0,,sid87_gci212230,00.html.

[12] Akashi Satoh and Tadanobu Inoue: ASIC-hardware-focused comparison for hash functions MD5, RIPEMD-160, and SHS, *International Conference on Information Technology: Coding and Computing, ITCC 1*, 2005, pp.532-537.

[13] Shaohua Tang: Hash-function based efficient key assignment for hierarchical access control, *WSEAS Transactions on Computers*, Issue 9, Volume 5, 2006, pp. 1931-1935.

[14] ONION software: Hot Soup Processor Official Homepage (in Japanese), http://www.onionsoft.net/hsp/.

[15] SourceForge, Inc.: SOURCEFORGE.net, http://sourceforge.net/.

[16] YouTube, LLC, YouTube - Broadcast Yourself., http://jp.youtube.com/.

[17] Adobe, FLV/F4V Technology Center, http://www.adobe.com/devnet/flv/.

[18] Hiroshi Dozono and Masanori Nakakuni et.al, The Analysis of Pen Inputs of Handwritten Symbols using Self Organizing Maps and its Application to User Authentication, *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN 2006)*, 2006, pp.4884-4889.

[19] Hiroshi Dozono and Masanori Nakakuni et.al, The Analysis of Key Stroke Timings using Self Organizing Maps and its Application to Authentication, *Proceedings of the 2006 International Conference on Security and Management (SAM'06)*, 2006, pp.100-105.

[20] Masanori Nakakuni, Hiroshi Dozono, et.al, Application of Self Organizing Maps for the Integrated Authentication using Keystroke Timings and Handwritten Symbols, *WSEAS TRANSACTIONS on INFORMATIONSCIENCE & APPLICATIONS*, Issue 2, Volume 4, 2007, pp.413-420.

[21] Hiroshi Dozono, Masanori Nakakuni, et.al, An Integration Method of Multi-Modal Biometrics Using Supervised Pareto Learning Self Organizing Maps, *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN 2008)*, 2008, CD-ROM.

[22] Masanori Nakakuni, Hiroshi Dozono, Ito Eisuke and Yoshiaki Kasahara, A Method of Automatic User Authentication by Fulltime Monitoringof Keystroke Timings, *Proceedings of the 2008 International Conference on Security and Management (SAM'08)*, 2007, CD-ROM.

[23] FENGHUA WANG, JIUQIANG HAN, XIANGHUA YAO: Iris Recognition Based on Multialgorithmic Fusion, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, Issue 12, Volume 4, 2007, pp.1415-1422.

[24] Hiroshi Dozono, Daishi Takata, Masanori Nakakuni and Yoshio Noguchi: The Analysis of Pen Pressures of Handwritten Symbols on PDA Touch Panel Using Self Organizing Maps, *The 2005 International Conference on Biometric Authentication (BIOAU'05)*, 2005, pp. 440-445.

[25] Hiroto Yasuura: Digitally named world: Challenges for new social infrastructures, *Proceedings of the 2005 IEEE International Workshop on VLSI Design and Video Technology (IWVDVT 2005)*, 2005, pp.21.

[26] Haeryong Park, Kilsoo Chun and Seungho Ahn: The security requirement for off-line E-cash system based on IC Card, *Proceedings of the*

International Conference on Parallel and Distributed Systems (ICPADS), 2005, pp.260-264.
[27] Israsena P.: Securing ubiquitous and low-cost RFID using tiny encryption algorithm, *2006 1st International Symposium on Wireless Pervasive Computing*, 2006, pp.1-4.
[28] SYLVIA ENCHEVA and SHARIL TUMIN: Preventing Conflict Situations During Authorization, *WSEAS TRANSACTIONS on COMPUTERS*, Issue 4, Volume 7, 2008, pp.265-272.
[29] The Open Group: Single Sign-on, http://www.opengroup.org/security/sso/.
[30] Gang Zhao, Dong Zheng, Kefei Chen: Design of single sign-on, Proceedings of the E-Commerce Technology for Dynamic E-Business, *IEEE International Conference*, 2004, pp. 253-256.
[31] Thomas Gross: Security Analysis of the SAML Single Sign-on Browser/Artifact Profile, *Proceedings of Computer Security Applications Conference*, 2003, pp. 298-307.
[32] Google, Inc.: SAML Single Sign-On (SSO) Service for Google Apps, http://code.google.com/apis/apps/sso/saml_reference_implementation.html.
[33] The Shibboleth Development Team: Shibboleth (A Project of the Internet2 Middleware Initiative), http://shibboleth.internet2.edu/.
[34] Toshihiro Takagi, Takaaki Komura, Shuichi Miyazaki and Yasuo Okabe: Privacy Oriented Attribute Exchange in Shibboleth Using Magic Protocols, *2008 International Symposium on Applications and the Internet (SAINT2008)*, 2008, pp.293-296.
[35] Internet2: Internet2, http://www.internet2.edu/.