Multi-Application Authentication based on Multi-Agent System

Somchart Fugkeaw, Piyawit Manpanpanich, and Sekpon Juntapremjitt

Abstract— **This paper proposes an authentication approach to support multi-clients in using a multi-application based environment. The approach is primarily based on the public key infrastructure (PKI) authentication scheme and the multi-agent technique. A key pair and a certificate issued by the Certification Authority (CA) are normally kept in a smart card or a token in order to enforce two-factor authentication. Both key pair and certificate are deployed to encrypt/decrypt electronic data or transaction, or sign/authenticate the sender and the recipient. We apply the Single Sign-On (SSO) and the Multi-Agent System (MAS) concepts to facilitate the authentication and the authorization process in order to work with multi-applications and multi-clients more dynamically and efficiently. The agent system functions when each client requests to sign on and it is responsible for validating a client certificate, granting an access role to the client, and controlling a concurrent use of applications.**

Index Terms— **Authentication, Multi-application, Certification Authority, SSO, Multi-Agent System.**

I. Introduction

In general, the authentication within computing systems encompasses identity verification, message origin authentication, and message content authentication [1]. An authentication scheme by PKI is a profound technique used in most web-based applications in which the authentication is needed to verify the authenticity of clients and entities using the automated web-based information system. A variety of methods are available for performing client authentication, and these methods form the basis for access control systems [2].

Nowadays, distributed system environment may comprise many system applications to support various business purposes demanded by many clients. In such environment, a security, non-repudiation and authentication technique is critically required. The encryption and password authentication are a common technique used by most applications. However, the security level of the information and application depends on the value in the business context. The security mechanism may be developed both at network level and at application level.

Currently, the Single Sign-On concept has been adopted to supply the security system to be more feasible and efficient for managing the exposure of the number of users in distributed system environment. However, the key role for multiapplications and multiusers authentication with the high trustworthy are not fullly addressed by SSO system. Apparently, PKI is recognized as a powerful technique to satisfy the security services including confidentiality, authentication, integrity, and non-repudiation. The PKI is thus mostly adopted as a trust model for embedding in messaging environment.

In addition to the authentication by Single-Sign on concept enabling the clients to access various systems by providing a single credential only one time, an efficient mechanism for supporting multi-relying party identification and the multi-application allocation is necessary. In this paper, we present an alternative design of the multi-application authentication model based on the multi-agent technique and Single-Sign-on concept to serve the research goal.

In fact, the multi-agent system (MAS) is a technique in the artificial intelligence area focusing on the system where several agents communicate with each other. In [4], multi-agent system is defined as “a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity”.

Technically, we apply MAS concept as a mediator to (1) perform the authentication control of the relying entities having certificate; (2) verify the client role and grant the permission to the legal applications; (3) schedule client requests and allocate the application services to multi clients autonomously and dynamically.

Besides, we also point out some implementation issues related to the real world application and current status of our ongoing implementation. Finally, we outline some promising idea for extending the capability of our system model, in the future work section.

Manuscript received January 15, 2007. This work has been supported by Thai Digital ID Co., Ltd.

Somchart Fugkeaw is with the CA Operation Department, Thai Digital ID Co., Ltd., BKK 10500 Thailand, phone: (66)-2634-3230; fax: (66)-2634-3231; e-mail: somchart@ thaidigitalid.com

Piyawit Manpanpanich is with the CA Operation Department, Thai Digital ID Co., Ltd., BKK 10500 Thailand. (email: piyawit@thaidigitalid.com)

Sekpon Juntapremjitt is with the IT Security Consulting Department, Whitehat Certified Co.,Ltd., BKK 10310 Thailand (e-mail: sekpon@whitehatpro.com).

(Advance online publication: 24 May 2007)

The contributions of the paper are:

(1) a strong authentication mechanism based on PKI and two-factor authentication;
(2) an application of MAS on SSO
(3) a support of multi-user and multi-application authentication ;
(4) a trusted agent cooperation;
(5) a practical and flexible model for administration, parallel computation and resource allocation.

The rest of this paper is organized into five additional sections. Section 2 presents some works related to our research. Section 3 describes the architectural framework of multi-application authentication approach based on the MAS concept. Section 4 details the design and implementation of the proposed model. Finally, Section 5 summarizes our research work, reports our current implementation and suggests the future work.

II. Related Work

The research work related to the authentication model in the network and internet based applications have been done so far by common techniques [8 ,9, 10, 11] such as password based authentication, token based authentication, biometric based authentication, and combination of those methods.

To the best of our knowledge, there are very few works dedicated to the integration issue of authentication model and secure multi-application management. In addition, the conventional scheme of the broker authentication has such many problems as, the administration of a greater number of anonymous clients, the vulnerability about the non-repudiation of the entity, an exposure of IDs and passwords.

In [5] a public key based cryptographic protocol for secure channel protocol using a combination of public key, secret key and Diffie-Hellman key establishment protocols are proposed to support in multi-application smart cards. The research focus is to establish a secure protocol based on the PKI facilitating the use of multi-application smart cards that is beneficial to implicitly support the use of a smart card containing the key for any secure applications. However, it does not address any authentication scheme for multi-application environment.

In [15] the architecture and implementation of the security system implementing the authentication and the secure communication among agents are proposed. The approach uses the certification authority (CA) to ensure the full cooperation of agents. The paper also deals with security mechanism's activity during inaccessibility of CA and possibility of CA's reconstruction. However, it does not focus on the multi-application authentication based environment.

The work proposed in [6] really inspires our research idea. The authors propose an authentication broker model that integrates the Single Sign-On and Multi-Agent system to satisfy the security requirements including confidentiality, integrity, and non-repudiation. The system factor and protocol of agent creation are presented with respect to the authority level of the corresponding service. Nevertheless, the clear function of agents used for delegating application to the authorized clients is not provided.

Our research emphasizes the effective use of full-fledged PKI authentication scheme with two-factor authentication and SSO. We also aim to promote PKI technology in which the CA is a core entity to provide a key pair and certificate to the clients to support all security as well as authentication services. The clients in our test environment need an authorization identified by the CA and hold the keys and certificate in the form of a secure smart card or a token. Another focus is to apply the multi-agent system to control the strong authentication and flexible use of various applications by many clients.

III. A Framework of Multi-Application Authentication based on MAS Concept

A. *Overview of the Proposed Model*

Fig.1 presents the conceptual view of our proposed model.

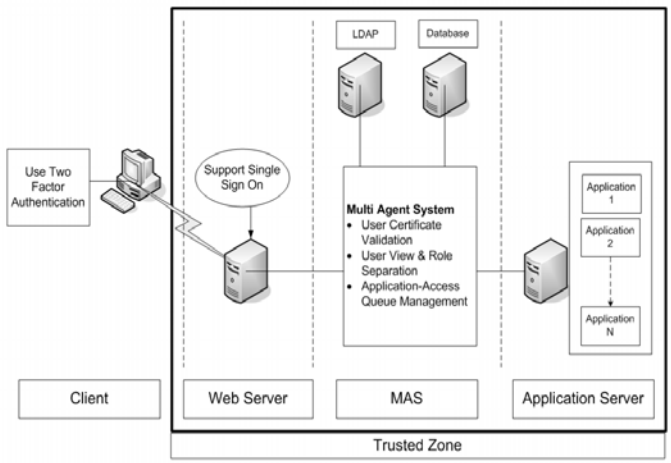

Fig.1: A Framework of Multi-Application Authentication based on MAS

The system model consists of four main parts:

(1) Client is generally a client who requests to use application(s). In our system, the clients need to authenticate themselves by using the certificate securely stored in a smart card or a token for two-factor authentication before accessing the application(s). In addition, the single sign-on is required in this process so that the clients can access several applications without necessity to be authenticated by each system individually.

(2) Web Server is responsible for accepting HTTP requests

from clients and giving responses along with the data contents, which usually are Web pages. In our system environment, web sessions are secured by SSL.

(3) Multi Agent System (MAS) is the core part of the proposed model. At the MAS server, the key pair and certificate are installed to further use for securing and authenticating the communication process among agents. Since MAS is the core trusted entity, all active agents trust all information signed by the MAS key.

There are two types of agents:

• User Agent (UA) is responsible for validating client certificates, verifying client requests, and delegating corresponding application(s) to the client. Each UA will be dead after a complete logout, or after certain idle period, which is the SSO session timeout value.

• Application Agent (AA) is mapped to a particular application and functions as the representative of an application in serving requests from UAs. Its job is to schedule the sequence of clients connecting to applications, to support the multiple application selection by clients, and to log on to the application on behalf of the client. Each AA has its own key pair and certificate.

(4) Application Server is a server provisioning application service(s).

B. Trust Model

In secure cooperation of agents, a trust model is required. We make use of the PKI as the basic technique in creating trust among agents. CA is the core of the infrastructure. AA has its own keypair and digital certificate, issued by the CA. This is not applicable to UA, which could be countless when the number of users is huge. Hence we have MAS subscribe to the CA and own the keypair and the certificate. As the UA generator, it guarantees messages for all UAs. On this ground, all messages among agents can be signed/verified and encrypted/decrypted with the basic PKI scheme. After the authentication among the trusted components is done, the user will be authorized to access the application accordingly.

C. Multi-Application Authentication process by MAS

In this section, we describe how the MAS is constructed and deployed.

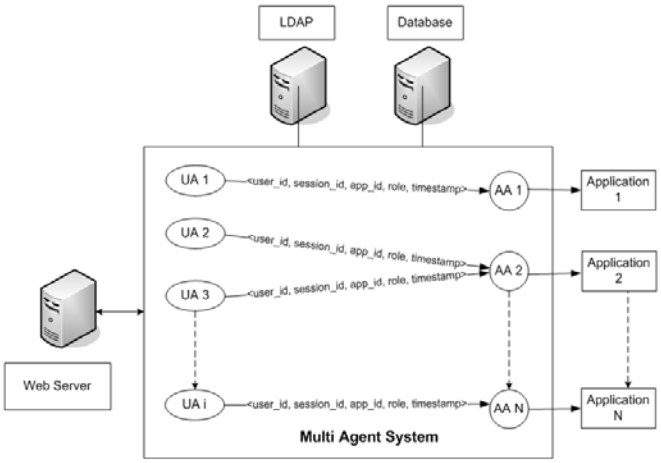

Fig.2: MAS Authentication Process

Fig.2 presents the MAS process and communication between user agent and application agent.

In the process, there are two major phases: Setup and Runtime. Only when a client signs in to the system, the Setup phase is done for client authentication and UA preparation. Upon the receipt of an application access request from the client, the Runtime phase starts for access verification and application delegation. The processes are described as follow:

Setup Phase:

[Step1] Two-Factor Authentication: Client uses the smart card or Token to authenticate himself/herself via SSL to the Web Server. This step is normally supported by SSL technology.

[Step 2] MAS Construction: After the successful two-factor authentication, Web Server requests the MAS module to generate a UA. The UA is mapped to the client for managing all of its application requests. Logically, the MAS module, a trusted core component, will generate the UA whenever the client has successfully authenticated to the system. On this ground, this newly-created UA is automatically trusted.

[Step 3] Client Certificate Validation: The UA looks up the LDAP, verifies the authenticity of the client certificate, and checks its validity against a pre-defined policy (e.g. CRL status, specific content rules)

[Step 4] Client Capability Identification: If the authenticity and validity of the client certificate is ensured, the user will be induced to the profile forming step. From the database in which the client information has been stored, the UA obtains the user information from the authorization matrix and form the capability list, which is securely stored in the UA memory. Essentially the capability list contains information about the action/role that the user can do/have on all allowed applications.

Runtime Phase:

[Step 5] Application Delegation:

Once the UA recognizes an application access request (application and action) from the client, it will verify such a request against the client capability list (and maybe some additional policies). If the user is authorized, the UA will then make a request to an appropriate AA in the scheme detailed in Step 6 to start the new session.

[Step 6] UA Message Delivery:

The message that UA sends to the AA includes <user_id, session_id, app_id, role, timestamp> where

- *user_id* is the id of client or user asking for the request

- *session_id* is the id of communication session of the request (this could be randomly generated at the beginning of the session)

- *app_id* is the id of Application which is requested by the user

- *role* is the function that the user presents himself/herself to the application, used together with user_id to obtain proper authorization. This information is optional for many applications as it could be useless.

- *timestamp* is the time that UA sends the request

Trust of the UA message is assured by the PKI technique. That is, MAS guarantees the UA message to the AA by signing it with its private key. The signed message is then encrypted with the AA's public key to ensure confidentiality. AA automatically trusts the message signed by MAS key since the MAS is a core trusted element. In addition, only legal AA can use its own private key to decrypt the message. This process is used to ensure that trust has been thoroughly created in the agent system.

[Step 7] UA Message Verification:

Upon receipt of a message, AA will acknowledged the UA and verify the trustworthiness of the message by verifying the digital signature signed by MAS in the previous step.

[Step 8] Multi-Application Control: After the process in Step 7 is done, AA will then be responsible for controlling the use of multi-application requests by several users (UAs). It manages the application access queue and does the login task on behalf of the authorized users.

As a consequence, all processes above achieve the multi-client and multi-application authentication purpose with MAS functions. However the issue of the agent, particularly AA, recovery, complex administration policies e.g. mandatory access control, quota, concurrent access constraints, priority, as well as accountability are highly required for our extended version of the proposed MAS model.

IV. IMPLEMENTATION

A. Overview of the Implementation

We initially prove our proposed idea on how the MAS supports the multi-application authentication and management. Therefore the primitive goal of our experiment is to verify that the proposed MAS module is functionally correct and feasible to support the authentication of multi-applications and multi-clients. The test scenario consists of a web server, LDAP directory, Database Oracle 9i. For the MAS module, we use Java programming for the development.

In our initial experiment, ten clients are assigned to register for the certificate and key pair, which are kept in the USB e-token issued by the certification authority in order to use the multiple web-based applications autonomously.

The clients need to perform two-factor authentication and single sign on before accessing to web applications. The clients will be allowed to get through the corresponding web application when their authenticity and application's access right are checked to be valid by the MAS engine.

Fig.3 displays the screen shot of the client authentication which is showed up when the clients connect to the web server with their tokens.

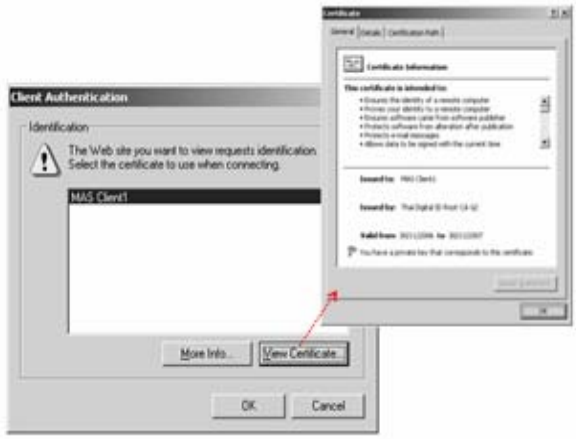

Fig.3: Client Authentication Screen

Transparent to clients, the authentication process and multi-application access management are controlled by the MAS. If the clients are successfully authenticated, they will be allowed to traverse to any applications available to them as shown in Fig. 4 (based on the capability list) and select one(s) without the need of several sign-in requests. The communication among client, web server, MAS and web application are secured by SSL protocol.

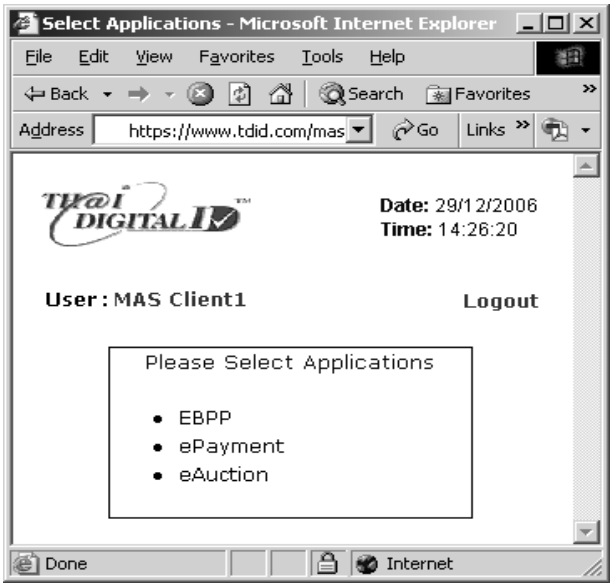

Fig.4: Application Selection Window

B. MAS Configuration Administration

To provide the effective way in configuring the MAS, the MAS configuration interface is designed and developed. Fig. 5 presents the MAS configuration screen

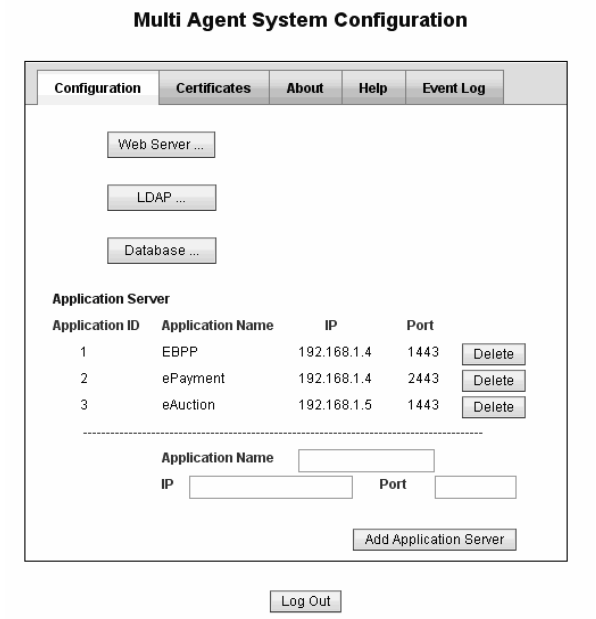

Fig.5: Multi Agent System Configuration

From this screen, the system administrator can configure the system components including web server, LDAP directory, databases, and application servers. Here, we can add any applications to the profile list and connect them to the authentication pool. In addition, we are currently implementing the key and certificate management function which co-operate with the certification authority to enable our relying parties to request for issuance, suspension, revocation, and renewal in a more feasible way.

To verify the MAS authentication process and current status of application to which the clients connect, the administrator can check all activities from the event log as shown in Fig.6.

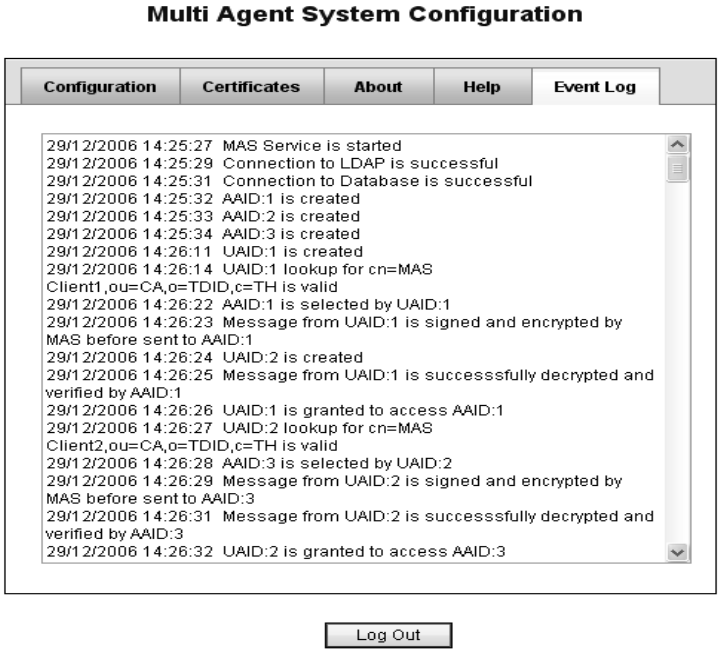

Fig.6: Event Log

According to our experiment, the result from the verification shows that MAS functionalities are correct and robust for all connections. Empirically, the communication cost outperforms the several authentication visits and accesses to multiple applications.

V. Conclusion and Future Work

We have presented the idea and implementation of how to apply MAS technique to serve the authentication service in the multi-clients and multi-application environment. The design of user agent and application agent is introduced to perform the client authentication and multi-application delegation. The combination of two-factor authentication, Single sign-on, and digital certificate are adopted to reflect the real need of current distributed applications. Therefore, client convenience is greatly increased by using our system. Also, the administrator can save the management cost since the security and authentication policy and configurations can be made easy. Finally, we present our ongoing implementation with the focus on features of MAS.

At present, we have been implementing the preventive activity-based authorization policy to serve the full authentication, authorization, and accountability. The enforcement of the activity-based policy helps identify excessive unauthorized access requests, and subsequent

actions, as defined in the authorization database. It also prevents negative consequences of the activities. For example, the preventive authorization policy could define that a user privileges will be degraded to 'guest' if it found that he/she requested for over-privilege accesses more than 10 times within 20 minutes. As the capability list is based on user activity, it is dynamic

In our future works, there are a number of issues to be addressed. Agent recovery is very important for system robustness. Complex administration policies could be extensively applied. For example, AA could do security clearance checks for mandatory access control; quota and accountability system must be established. In terms of reliability, the system needs to be tested under a high number of clients and applications. Moreover, a serious consideration, evaluation and assessment for the performance and resource consumption should be done further. For a more advanced feature of our system, an integration of several types of agents, e.g. mobile agent can be adopted for future version of a hybrid authentication model.

REFERENCES

[1] Woo, T. Y. C., and S.S. Lam, Authentication for Distributed Systems, IEEE CS Press, January 1992.

[2] Guideline on User Authentication Techniques for Computer Network Access Control, National Institute of Standards and Technology, Federal Information Processing Standards Publication 83, National Technical Information Service, Springfield, VA, September 1980.

[3] Jennings, N.R., Sycara, K. and Wooldridge, M. A Roadmap of Agent Research and Development. In: *Autonomous Agents and Multi-Agent Systems Journal*, N.R. Jennings, K. Sycara and M. Georgeff (Eds.), Kluwer Academic Publishers, Boston, 1998, Volume 1, Issue 1, pages 7-38.

[4] Durfee, E.H., Lesser, V.R. and Corkill, D.D. Trends in Cooperative Distributed Problem Solving. In: *IEEE Transactions on Knowledge and Data Engineering*, March 1989, KDE-1(1), pages 63-83.

[5] Konstantinos Markantonakis, Keith Mayes, "A Secure Channel protocol for multi-application smart cards based on public key cryptography", CMS 2004 - 8th IFIP TC-6-11 Conference on Communications and Multimedia Security, 15-18 September 2004

[6] Deok-Gyu Lee, Seo-Il Kang, Dae-Hee Seo, Im-Yeong Lee: Authentication for Single/Multi Domain in Ubiquitous Computing Using Attribute Certification. ICCSA (4) 2006

[7] Dae-Hee Seo, Im-Yeong Lee, Soo-Young Chae, and Choon-Soo Kim , Single sign-on authentication model using MAS, Proc. of IEEE Communications, Computers, and Signal Processing 2003.

[8] Guideline for The Use of Advanced Authentication Technology Alternatives National Institute of Standards and Technology, Federal Information Processing Standards Publication 90.

[9] Password Usage, National Institute of Standards and Technology, Federal Information Processing Standards Publication 112, National Technical Information Service, Springfield, VA, May 1985.

[10] Smart Card Technology: New Methods for Computer Access Control, National Institute of Standards and Technology, NIST Special Publication 500-157, National Technical Information Service, Springfield, VA, September 1988.

[11] Dray, J. F., M. E. Smid and R. Warnar, A Token Based Access Control System for Computer Networks, Proceedings - The 12th National Computer Security Conference, October 1989.

[12] William Stallings, Cryptography and Network Security: Principles and Practice, Fourth Edition;, Prentice Hall: 2005.

[13] Somchart Fugkeaw, Piyawit Manpanpanich, and Sekpon Juntapremjitt, Multi-Application Authentication based on Multi-Agent System, *Proceedings of IAENG International Conference on Communication Systems and Applications (ICCSA'07)*, HongKong, March 2007.

[14] Zhaohui Wu, Shuming Tang, Shuigang Deng, Jian Wu, Huajun Chen, Haojn Gao, DartGrid II: A Semantic Grid Platform for ITS, *IEEE Intelligent Systems*, vol.20, No.3, Jun. 2005.

[15] Petr Nova, Milan Rollo, Jiri Hodik, Tomas Vlcek: Communication Security in Multi-agent Systems, *CEEMAS 2003*, Springer-Verlag Berlin Heidelberg 2003.

[16] Wenpin Jiao, Minghui Zhou, Qianxiang Wang: Formal framework for adaptive multi-agent systems, *Proceedings IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003)*, pages 442-445, *13-16 October 2003.*

[17] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R.L. Rivest: Certificate chain discovery in SPKI/SDSI, *Journal of Computer Security*, 9(4):285-322, 2001.

[18] P. Bonatti and P. Samarati, Regulating service access and information release on the web, *Proceedings of the 7th ACM Conference on Computer and Communication Security*, pages 134-143, Athens, Greece, Nov. 2000.

[19] Q. He, K. P. Sycara, and T. Finin. Personal Security Agent: KQML-Based PKI, *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 377-384. ACM Press, 1998.

[20] J. Biskup and Y. Karabulut. A hybrid PKI model with an application for secure mediation, *16th Annual IFIP WG 11.3 Working Conference on Data and Application Security*, Cambridge, England, July 2002.