CHAPTER 1

Protecting Information when Access is Granted for Collaboration

Keynote 1

Gio Wiederhold

Dept. of Computer Science, Stanford University, Stanford CA 94305; gio@cs.stanford.edu

Abstract: There are settings where we have to collaborate with individuals and organizations who, while not being enemies, should not be fully trusted. Collaborators must be authorized to access information systems that contain information that they should be able to receive. However, these systems typically also contain information that should be withheld. Collaborations can be rapidly created, requiring dynamic alterations to security policies. Classifying data to cover all current and possible access privileges is both awkward and costly, and always unreliable.

An alternative approach to protection, complementing basic access control, is to provide filtering of results. Filtering of contents is also costly, but provides a number of benefits not obtainable with access control alone. The most important one is that the complexity of setting up and maintaining specific, isolated information cells for every combination of access rights held by collaborators is avoided. New classes of external collaborators can be added without requiring a reorganization of the entire information structure. There is no overhead for internal use, i.e., for participants that are wholly trusted. Finally, since documents contents rather than their labels are being checked, cases of misfiled information will not cause inappropriate release.

The approach used in the TIHI/SAW projects at Stanford uses simple rules to drive filtering primitives. The filters run on a modest, but dedicated computer managed by the organization's security officer. The rules implement the institution's security policy and must balance manual effort and complexity. By not relying on the database systems and network facilities, and their administrators a better functional allocation of responsibilities ensues.

1. Introduction

There are data sources that are primarily intended for external access, such as public web pages, reports, bibliographies, etc. These are organized according to external access criteria. If some of the information is not intended to be public, then security provisions are put into place. In more complex settings several mandatory layers will exist, and protection may require discretionary access control as well. When we deal with collaborators more discretionary partitions will be needed, and those are likely to overlap, creating combinatorial cells. Examples in the medical domain include external research groups, various public health agencies, pharmaceutical companies performing drug surveillance, as well as third-party payors.

These collaborators are increasingly important in our complex enterprises, and cannot be viewed as enemies. They may be part of a supply chain, they may provide essential, complementary information, or may supply specialized services, beyond our own capabilities. However, their roles are often specific, so that they do not fit neatly into existing security categories. The problem of managing security becomes impossibly complex.

Today, security provisions for computing focus on controlling access. At least five technological requirements interact in the process, and all of these are well recognized in the literature:

1. Secure Communication, c.f. [He:97].
2. Perimeter Control, c.f. [CheswickB:94].
3. Reliable Authentication, c.f. [CastanoFMS:95]
4. Authorization to information cells, c.f. [GriffithW:76]
5. Partitioning of the information into cells, c.f. [LuniewskiEa:93]

The fifth requirement, often under-emphasized, is that there must be a highly reliable categorization of all information to those cells [Oracle:99]. It is in the management of that categorization where many failures occur, and where collaborative requirements raise the greatest problem.

Relying on access control makes the assumption that all these five conditions are fulfilled. Unfortunately, there are many situations where the last one, namely perfect partitioning of the information into cells for disjoint access is not realistic. A corporation may have large, existing databases which were established before external access needed to be considered. In modern environments access will be needed for off-site staff, corporate salespersons, vendors which have contract relationships, government inspectors, and an ever-increasing number of collaborators. The trend to outsourcing of tasks that used to be internal exacerbates the problem. Reorganizing corporate databases to deal with developing needs for external

access is costly and disruptive, since it will affect existing users and their application. It is not surprising that security concerns were the cited as the prime reason for lack of progress in establishing *virtual enterprises* [HardwickS:96].

We encountered the problem initially in the manufacturing area, where security concerns caused the interchange of manufacturing data to a subcontractor to take many weeks, although they had installed compatible CAD systems and high-speed datalinks. All drawings had to be printed, inspected by a security specialist, verified, and edited if the design contained information inappropriate for the subcontractor. The edited drawings could then be copied and shipped to the contractor, who had to scan the paper copies into their systems. The source of the problem is, of course, that the design engineer makes drawings of the equipment to be built, with justifications, finishing information, and explicit and implicit performance criteria. The drawings are not initially produced to satisfy the capabilities of an unknown subcontractor.

Our actual initial application domain was actually in healthcare. Medical records are needed for many purposes: diagnosis, care delivery, drug supplies, infection control, room assignments, billing, insurance claims, validation of proper care, research, and public health records. Patient care demands that the record be accessible in a comprehensive form and up-to-date [Rindfleisch:97]. Historical information is important for disease management, but not for many billing tasks. It is obviously impossible to split the record into access categories that match every dimension of access. Even if that would be possible, the cost and risks to the internal operations in a hospital or clinic would be prohibitive. Expecting a physician to carry out this task is unrealistic, and, if required, would greatly increase the cost of healthcare.

Partitioning of the information into cells

The process of assigning categories to information involves every person who creates, enters, or maintains information. When there are few cells, these originators can understand what is at stake, and will perform the categorization function adequately, although error in filing will still occur. When there are many cells, the categorization task becomes onerous and error prone. When new coalitions are created, and new collaborators must share existing information system, the categorization task becomes impossible.

A solution to excessive partitioning might be to assign accessors combinations of access rights. This approach appears to invert the multi-level security approach. Numerous research results deal with multi-level

security within a computer system [LuntEa:90]. Several simple implementations are commercially available, but have not found broad acceptance, likely because of a high perceived cost/benefit ratio [Elseviers:94]. These systems do not accommodate very many distinct cells, and mainly support mandatory security levels [KeefeTT:89]. Leaks due to inference are still possible, and research into methods to cope with this issue is progressing [Hinke:88]. However few cases of exploiting these weaknesses have been documented [Neuman:00].

However, break-ins still occur. Most of them are initiated via legitimate access paths, since the information in our systems must be shared with customers and collaborators. In that case the first three technologies provide no protection, and the burden falls on the mappings and the categorization of the information. Once users are permitted into the system, protection becomes more difficult.

A complementary technology

The solution we provide to this dilemma is *result checking* [WiederholdBSQ:96]. In addition to the conventional tasks of access control the results of any information requests are filtered before releasing them to the requestor. We also check a large number of parameters about the release. This task mimics the manual function of a security officer when checking the briefcases of collaborating participants leaving a secure meeting, on exiting the secure facility. Note that checking of result contents is not performed in standard security processing. Multi-level secure systems may check for unwanted inferences when results are composed from data at distinct levels, but rely on level designations and record keys. Note that result checking need not depend on the sources of the result, so that it remains robust with respect to information categorization, software errors, and misfiling of data.

2. Filtering System Architecture

We incorporate result checking in a *security mediator* workstation, to be managed by a security officer. The *security mediator* system interposes security checking between external accessors and the data resources to be protected, as shown in Fig.1. It carries out functions of authentication and access control, to the extent that such services are not, or not reliably, provided by network and database services. Physically a security mediator is designed to operate on a distinct workstation, owned and operated by the enterprise security officer (S.O.). It is positioned as a pass gate within the enterprise firewall, if there is such a firewall. In our initial commercial

installation the security mediator also provided traditional firewall functions, by limiting the IP addresses of requestors [WiederholdBD:98].

Figure 1. Functions provided by a TIHI/SAW Security Mediator

The mediator system and the source databases are expected to reside on different machines. Thus, since all queries that arrive from the external world, and their results, are processed by the security mediator, the databases behind a firewall need not be secure unless there are further internal requirements. When combined with an integrating mediator, a security mediator can also serve multiple data resources behind a firewall [Ullman:96]. Combining the results of a query requiring multiple sources prior to result checking improves the scope of result validation.

The supporting database systems can still implement their view-based protection facilities [GriffithsW:76]. These need not be fully trusted, but their mechanisms add efficiency.

Operation

Within the workstation is a rule-base system which investigates queries coming in and results to be transmitted to the external world. Any request and any result, which cannot be vetted by the rule system, is displayed to the security officer, for manual handling. The security officer decides to approve, edit, or reject the information. An associated logging subsystem provides an audit trail for all information that enters or leaves the domain.

The log provides input to the security officer to aid in evolving the rule set, and increasing the effectiveness of the system.

The software of our security mediator is composed of modules that perform the following tasks

1. Optionally (if there is no firewall): Authentication of the requestor
2. Determination of authorization type (clique) for the requestor
3. Processing of a request for information (pre-processing) using the policy rules
4. If the request is dubious: interaction with the security officer
5. Communication to internal databases (submission of certified request)
6. Communication from internal databases (retrieval of unfiltered results)
7. Processing of results (post-processing) using the policy rules
8. If the result is dubious: interaction with the security officer
9. Writing query, origin, actions, and results into a log file
10. Transmission of vetted information to the requestor

Item 7, the post-processing of the results obtained from the databases, possibly integrated, is the critical additional function. Such processing is potentially quite costly, since it has to deal thoroughly with a wide variety of data. Applying such filters selectively, specifically for the problems raised in collaborations, as well as the capabilities of modem computers and text-processing algorithms, makes use of the technology feasible. A rule-based system is used in TIHI to control the filtering, allowing the security policies to be set so that a reasonable balance of cost to benefit is achieved. It will be described in the next section.

Having rules, however is optional. Without rules the mediator system will operate in fully paranoid mode. Each query and each result will be submitted to the security officer. The security officer will view the contents on-line, and approved, edit, or reject the material. Adding rules enables automation. The extent of automation depends the coverage of the rule-set. A reasonable goal is the automatic processing of say, 90% of queries and 95% responses.

Unusual requests, perhaps issued because of a new coalition, assigned to a new clique, will initially not have applicable rules, but can be immediately processed by the security officer. In time, simple rules can be entered to reduce the load on the officer.

Traditional systems, based on access control to precisely defined cells, require a long time to before the data are set up, and when the effort is great, may never be automated. In many situations we are aware of, security mechanisms are ignored when requests for information are deemed to be important, but cannot be served by existing methods. Keeping the security officer in control allows any needed bypassing to be handled formally. This capability recognizes that in a dynamic, interactive world there will always be cases that are not foreseen or situations the rules are too stringent.

Keeping the management of exceptions within the system greatly reduces confusion, errors, and liabilities.

Even when operating automatically, the security mediator remains under the control of the enterprise since the rules are modifiable by the security officer at all times. In addition, logs are accessible to the officer, who can keep track of the transactions. If some rules are found to be too liberal, policy can be tightened. If rules are too stringent, as evidenced by an excessive load on the security officer, they can be relaxed or elaborated.

3. The Rule System

The rules system is composed of the rules themselves, an interpreter for the rules, and primitives, which are invoked by the rules. The rules embody the security policy of the enterprise. They are hence not preset into the software of the security mediator.

In order to automate the process of controlling access and ensuring the security of information, the security officer enters rules into the system. These rules are trigger analyses of requests, their results, and a number of associated parameters. The interpreting software uses these rules to determine the validity of every request and make the decisions pertaining to the disposition of the results. Auxiliary functions help the security officer enter appropriate rules and update them as the security needs of the organization change.

The rules are simple, short and comprehensive. They are stored in a database local to the security mediator system with all edit rights restricted to the security officer. Some rules may overlap; in which case the most restrictive rule automatically applies. The rules may pertain to requestors, cliques of requestors having certain roles, sessions, databases tables or any combinations of these.

Rules are selected based on the authorization clique determined for the requestor. All the applicable rules will be checked for every request issued by the requestor in every session. All rules will be enforced for every requestor and the request will be forwarded to the source databases only if it passes all tests. Any request not fully vetted is posted immediately to the log and sent the security officer. The failure message is directed to the security officer and not to the requestor, so that the requestors in such cases will not see the failure and its cause. This prevents that the requestor could interpret failure patterns and make meaningful inferences, or rephrase the request to try to bypass the filter [KeefeTT:89].

The novel aspect of our approach is that security mediator checks outgoing results as well. This is crucial since, from the security-point-of-view, requests are inclusive, not exclusive selectors of content and may

retrieve unexpected information. In helpful, user-friendly information systems getting more than asked for is considered beneficial, but from a security point-of-view being generous is risky. Thus, even when the request has been validated, the results are also subject to screening by a set of rules. As before, all rules are enforced for every requestor and the results are accessible only if they pass all tests. Again, if the results violate a rule, a failure message is logged and sent to the security officer but not to the requestor.

Primitives

The rules invoke executable primitive functions which operate on requests, data, the log, and other information sources. As new security functions and technologies appear, or if specialized needs arise, new primitives can be inserted in the security mediator for subsequent rule invocation. In fact, we do not expect to be the source of all primitives. We do hope that all primitives will be sufficiently simple that their correct function can be verified.

Primitives that have been used include:

- Assignment of a requestor to a clique
- Limit access for clique to certain database table segments or columns
- Limit request to statistical (average, median, ...) information
- Provide number of data instances (database rows) used in a statistical result
- Provide number of tables used (joins) for result for further checking
- Limit number of requests per session
- Limit number of sessions per period
- Limit requests by requestor per period
- Block requests from all but listed sites
- Block delivery of results to all but listed sites
- Block receipt of requests by local time at request site
- Block delivery of results by local time at delivery site
- Constrain request to data which is keyed to requestor name
- Constrain request to data which is keyed to request site name
- Filter all result terms through a clique-specific good-word dictionary
- Disallow results containing terms in a clique-specific bad-word dictionary
- Convert text by replacing identifies with non-identifying surrogates [Sweeney:96]
- Convert text by replacing objectionable terms with surrogates
- Randomize responses for legal protection [Leiss:82]
- Extract text out of x-ray images (for further filtering) [WangWL:98]

- Notify the security officer immediately of failure reports
- Place failure reports only in the log

Not all primitives will have a role in all applications.

Primitives can vary greatly in cost of application, although modern technology helps. Checking for terms in results is costly in principle, but modern spell-checkers show that it can be done fairly fast. For this task we create clique-specific dictionaries, by initially processing a substantial amount of approved results. In initial use the security officer will still get false failure reports, due to innocent terms that are not yet in the dictionary. Those will be incrementally added, so that in time the incidence of such failures will be minimal.

For example, we have in use a dictionary for ophtamology, to allow authenticated researchers in that field to have access to patient data. That dictionary does not include terms that would signal, say HIV infection or pregnancies, information which the patients would not like to see released to unknown research groups. Also, all proper names, places of employment, etc. are effectively filtered.

Figure 2. Extract from a report to the Security Officer

Several of these primitives are designed to help control inference problems in statistical database queries [AdamW:89]. While neither we, nor any feasible system can prevent leaks due to inference, we believe that careful management can make reduce the probability [Hinke:88]. Furthermore, providing the tools for analysis, as logging all accesses will reduce the practical threat [Hinke:88], [Sweeney:97]. The primitive to enforce dynamic limits on access frequencies will often have to refer to the

log, so that efficient access to the log, for instance by maintaining a large write-through cache for the log, will be important. Here again the function of traditional database support and security mediation diverges, since database transaction are best isolated, where as inference control requires history maintenance.

4. Logging

Throughout, the failures, as well as the request text and source, and actions taken by the security officer, are logged by the system for audit purposes. Having a security log that is distinct from the database log is important since:

- A database system logs all transactions, not just external requests, and is hence confusingly voluminous
- Most database systems do not log attempted and failed requests fully, because they appear not to have affected the databases
- Reasons for failure of requests in database logs are implicit, and do not give the rules that caused them.

We provide user-friendly utilities to scan the security log by time, by requestor, by clique, and by data source. Offending terms in results are marked.

No system, except one that provides complete isolation, can be 100% foolproof. The provision of security is, unfortunately, a cat-and-mouse game, where new threats and new technologies keep arising. Logging provides the feedback, which converts a static approach to a dynamic and stable system, which can maintain an adequate level of protection. Logs will have to be inspected regularly to achieve stability.

Bypassing of the entire system and hence the log remains a threat. Removal of information on portable media is easy. Only a few enterprises can afford to place controls on all personnel leaving daily for home, lunch, or competitive employment. However, having an effective and adaptable security filter removes the excuse that information had to be downloaded and shipped out because the system was too stringent for legitimate purposes. Some enterprises are considering limiting internal workstations to be diskless. It is unclear how effective this approach will be outside of small, highly secure domains in an enterprise. Such a domain will then have to be protected with its own firewall and a security mediator as well, because collaboration between the general and highly secure internal domains must be enabled.

5. Current State and Further Work

Our initial demonstrations have been in the healthcare domain, and a commercial version of TIHI is now in use to protect records of genomic analyses in a pharmaceutical company. As the expectations for protection of the privacy of patient data are being solidified into governmental regulations we expect that our approach will gain popularity [Braithwaite:96]. Today the healthcare establishment still hopes that commercial encryption tools will be adequate for the protection of medical records, since the complexity of managing access requirements has not yet been faced [RindKSSCB:97]. Expenditures for security in medical enterprises are minimal [NRC:97]. Funding of adequate provisions in an industry under heavy economic pressures, populated with many individuals who do not attach much value to the privacy of others, will remain a source of stress.

Non-textual contents

Identifying information is routinely deleted from medical records that are disseminated for research and education. However, here a gap existed as well: X-ray, MRI, and similar images accompany many records, and these also include information identifying the patient. We have developed software that recognizes such text using wavelet-based decomposition and analysis, extracts it, and can submit to the filtering system developed in TIHI. Information, which is determined to be benign, can be retained, and other text is effectively removed by omitting high-frequency components in the affected areas [WangWL:98].

We have also investigated our original motivating application area, namely manufacturing information. Here the simple web-based interfaces that are effective for the customer and the security officer interfaces in health care are not adequate. We have demonstrated interfaces for the general viewing and editing of design drawings and any attached textual information. In drawings significant text may be incorporated in the drawings themselves. When delivering an edited drawing electronically, we also have to assure that there is no hidden information. Many design formats allow *undo* operations, which would allow apparently deleted information to reappear.

Before moving to substantial automation for collaboration in manufacturing, we will have to understand the parameters for reliable filtering of such information better. However, as pointed out initially, even a fully manual security mediator will provide a substantial benefit to enterprises that are trying to institute shared efforts rapidly.

6. Conclusions

Security mediation provides an architectural function as well as a specific service. Architecturally, expanding the role of a gateway in the firewall from a passive filter to an active pass gate service allows concentration of the responsibility for security to a single node, owned by the security officer. Assigning responsibilities for security to database or network personnel, who have primary responsibilities of making data and communication available, will conflict with security concerns and is unwise. These people are promoted to their positions because they have a helpful attitude and know how to overcome problems of system failures and inadequacies. This attitude is inherently in conflict with corporate and legal concerns for the protection of data.

Existing services, as constraining views over databases, encryption for transmission in networks, password management in operating systems can be managed via the security mediator node.

The specific, novel service presented here, result checking, complements traditional access control. We have received a patent to cover the concept. Checking results is especially relevant in systems with many types of users, including external collaborators, and complex information structures. In such settings the requirement that systems that are limited to access-control impose, namely that all data are correctly partitioned and filed is not achievable in practice. Result checking does not address all issues of security of course, as protection from erroneous or malicious updates, although it is likely that such attacks will be preceded by processes that extract information. A side-effect of result checking that it provides a level of intrusion detection.

The rule-based approach allows balancing of the need for preserving data security and privacy and for making data available. Data, which is too tightly controlled, reduces the benefits of sharable information in collaborative settings. Rules, which are too liberal, can violate security and expectation of privacy. Having a balanced policy will require directions from management. Having a single focus for execution of the policy in electronic transmission will improve the consistency of the application of the policy.

Result filtering does not solve all problems, in security, of course. They rely still on a minimum level of reliability in the supporting systems. They cannot compensate when information is missing or not found because of misidentification. In general, a security mediator cannot protect from inadvertent or intentional denial of information by a mismanaged database system.

Acknowledgements

Research leading to security mediators was supported by an NSF HPCC challenge grant and by DARPA ITO via Arpa order E017, as a subcontract via SRI International. Steve Dawson was the PI at SRI. The commercial transition was performed by Maggie Johnson, Chris Donahue, and Jerry Cain under contracts with SST (www.2ST.com). Work on editing and filtering graphics is due to Jahnvi Akalla and James Z. Wang, This paper is a version of a broader paper being submitted to an IEEE Transaction.

References

- [CastanoFMS:95] S.Castano, M.G. Fugini, G.Martella, and P. Samarati: *Database Security*; Addison Wesley Publishing Company - ACM Press, 1995 pp. 456
- [CheswickB:94].William R.Cheswick and Steven M. Bellovin: *Stalking the Wily Hacker*; Addison-Wesley, 1994.
- [Didriksen:97] Tor Didriksen: "Rule-based Database Access control – A practical Approach"; *Proc. 2nd ACM workshop on Rule-based Access Control*, 1997, pp.143-151.
- [Elseviers:94] Elseviers Advanced Technology Publications: Trusted Oracle 7; *Computer Fraud and Technology Bulletin*, March 1994.
- [GriffithsW:76] Patricia P. Griffiths and Bradford W. Wade: "An Authorization Mechanism for a Relational Database System"; *ACM Trans. on Database Systems*, Vol.1 No.3, Sept.1976, pp.242-255.
- [HardwickS:96] M. Hardwick, D.L. Spooner, T. Rando, and KC Morris: "Sharing Manufacturing Information In Virtual Enterprises"; *Comm. ACM*, Vol.39 no.2, pp.46-54, February 1996.
- [He:97] J. He: "Performance and Manageability Design in an Enterprise Network Security System"; *IEEE Enterprise Networking Miniconference 1997 (ENM-97)*, IEEE, 1997.
- [Hinke:88] T. Hinke: "Inference Aggregation Detection in Database management Systems"; *Proc. IEEE Symposium on Security and Privacy*, Oakland CA, April 1988.
- [JohnsonSV:95?] Johnson DR, Sayjdari FF, Van Tassel JP.: Missi security policy: A formal approach. Technical Report R2SPO-TR001, National Security Agency Central Service, July 1995.
- [KeefeTT:89] T. Keefe, B.Thuraisingham, and W.Tsai: "Secure Query Processing Strategies"; *IEEE Computer*, Vol.22 No.3, March 1989, pp.63-70.
- [LandwehrHM:84] Carl E. Landwehr, C.L. Heitmyer, and J.McLean: "A Security Model for Military Message Systems"; *ACM Trans. on Computer Systems*, Vol.2 No.3, Aug. 1984, pp. 198-222.
- [LuniewskiEa:93] Luniewski, A. et al. "Information organization using Rufus" SIGMOD '93, *ACM SIGMOD Record*, June 1993, vol.22, no.2 p. 560-1
- [LuntEa:90] Therea Lunt et al.: "The SeaView Security Model"; *IEEE Trans. on Software Eng.*, Vol.16 No.6, 1990, pp.593-607.

- [Neuman:00] Peter Neumann: Illustrative Risks to the Public in the Use of Computer Systems and Related Technology"; SRI International, May 2000, <http://www.csl.ri.com/neumann/illustrative.html>.
- [Oracle:99] *Oracle 8i Fine-grained Access Control*, Oracle corporation, February 1999.
- [QianW:97] Qian, XiaoLei and Gio Wiederhold: "Protecting Collaboration"; abstract for *IEEE Information Survivability Workshop, ISW'97*, Feb. 1997, San Diego.
- [RindKSSCB:97] David M. Rind, Isaac S. Kohane, Peter Szolovits, Charles Safran, Henry C. Chueh, and G. Octo Barnett: "Maintaining the Confidentiality of Medical Records Shared over the Internet and the World Wide Web"; *Annals of Internal Medicine* 15 July 1997. 127:138-141.
- [Rindfleisch:97] Thomas C. Rindfleisch: Privacy, Information Technology, and Health Care; *Comm. ACM*; Vol.40 No. 8, Aug.1997, pp.92-100.
- [SchaeferS:95] M. Schaefer, G. Smith: "Assured discretionary access control for trusted RDBMS"; in *Proceedings of the Ninth IFIP WG 11.3 Working Conference on Database Security*, 1995:275-289.
- [Seligman:99] Len Seligman, Paul Lehner, Ken Smith, Chris Elsaesser, and David Mattox: "Decision-Centric Information Monitoring"; *Jour. of Intelligent Information Systems (JIIS)*, Vol.14, No.1.; also at http://www.mitre.org/pubs/edge/june_99/dcim.doc
- [Sweeney:96] Latanya Sweeney: "Replacing personally-identifying information in medical records, the SCRUB system" Cimino, JJ, ed. *Proceedings, Journal of the American Medical Informatics Association*, Washington, DC: Hanley & Belfus, 1996, Pp.333-337.
- [Sweeney:97] Latanya Sweeney: "Guaranteeing anonymity when sharing medical data, the DATAFLY system"; *Proceedings, Journal of the American Medical Informatics Association*, Washington DC, Hanley & Belfus, 1997.
- [Ullman:97?] Jeffrey Ullman: Information Integration Using Logical Views; *International Conference on Database Theory (ICDT '97)* Delphi, Greece, ACM and IEEE Computer Society, 1997.
- [WangWL:98] James Z. Wang, Gio Wiederhold and Jia Li: Wavelet-based Progressive Transmission and Security Filtering for Medical Image Distribution"; in Stephen Wong (ed.): *Medical Image Databases*; Kluwer publishers, 1998, pp.303- 324.
- [WiederholdBC:98] Gio Wiederhold, Michel Bilello, and Chris Donahue: "Web Implementation of a Security Mediator for Medical Databases"; in T.Y. Lin and Shelly Qian: *Database Security XI, Status and Prospects*, IFIP / Chapman & Hall, 1998, pp.60-72.
- [WiederholdBSQ:96] Gio Wiederhold, Michel Bilello, Vatsala Sarathy, and XiaoLei Qian: A Security Mediator for Health Care Information"; *Journal of the AMIA Security Mediator for Health Care Information*"; *Journal of the AMIA* issue containing the Proceedings of the 1996 AMIA Conference, Oct. 1996, pp.120-124.
- [WiederholdEa:96] Gio Wiederhold, Michel Bilello, Vatsala Sarathy, and XiaoLei Qian: Protecting Collaboration; (postscript); presented and published at the National Information Systems Security Conference, 21 Oct. 1996; as *Proceedings of the NISSC'96*, Baltimore MD, Oct. 1996, pp.561-569.