

SECURE MEDIATION: REQUIREMENTS AND DESIGN

Joachim Biskup

Ulrich Flegel
and Yücel Karabulut

Abstract: In this paper¹ we discuss the security requirements for mediation, and present our approach towards satisfying them, with an emphasis on confidentiality and authenticity. Furthermore we outline the design of the basic security mechanisms for mediators. Our basic approach suitably combines the concepts of credentials, for authentic authorization with some kind of anonymity, and of asymmetric encryption, for confidentiality, and it can be extended to include additional mechanisms like digital signatures and fingerprints. Additionally it adopts the model of role based security policies because of its application orientation and of its potentials to integrate and unify various policies.

1 INTRODUCTION

Recent trends in information technologies led to vastly improved communication facilities like the Internet, an explosion of on-line multimedia information providers, and challenging new demands of users. Whenever a user looks for a piece of information, he may aim at identifying promising sources, which can be quite heterogeneous and autonomous, and then retrieving and integrating the required data. And whatever data a source has to offer, it may aim at

¹This work was partially supported by the *Ministerium für Wissenschaft und Forschung des Landes Nordrhein-Westfalen* within the joint project "*Virtuelle Wissensfabrik*" (*The Virtual Knowledge Factory*).

supporting a wide range of potential clients, which in general are unknown in advance. According to these trends, various forms of interoperable information systems have been developed. While federated database systems [26, 18] have already come into existence, increasingly ambitious further demands have evolved and resulted in the paradigm of mediated information systems [31, 33]. Some current projects on mediation are TSIMMIS [29], HERMES [27], Information Manifold [17], SIMS [1], AURORA [34], DISCO [28], Squirrel [13], DIOM [19], Garlic [7], OBSERVER [20], InfoSleuth [2], and MMM [3, 4].

In mediated information systems a client, seeking for information, and various and autonomous sources, holding potentially useful data, are brought together by a third kind of independent components, called mediators. Mediation is required to deal with the heterogeneity and the autonomy of the sources, not only from the functional point of view but also with respect to all aspects of security. This includes confidentiality and authenticity, as well as integrity, anonymity, non-repudiation and availability.

Previous work on security of interoperable information systems has mainly been done for federated databases [15, 10, 8], where the emphasis laid on resolving heterogeneity. According to the structure of federated databases, the security mechanisms were identity rather than credential based. There also appeared some contributions to security in mediated systems [6, 32, 14].

The concept of credentials has been advocated by Chaum [9] for supporting privacy in networked systems. Since then it has been adopted for various purposes in interoperable systems, for electronic payment and marketplaces as well as for middleware systems like CORBA [21]. Further work includes [25, 5, 11].

The model of role based security policies [24, 16, 12] has been successfully used before, and in particular studied for integrating various policies.

2 REQUIREMENTS OF FEDERATED AND MEDIATED SYSTEMS

While both federated database management systems and mediated information systems are used to integrate various autonomous information sources, several differences between both approaches may be identified. Most interesting in the context of this paper are differences related to and affecting security issues. We first observe that many of the differences result from differing motivations of participants of federated and mediated environments, respectively.

In a federated system the federation establishment is stimulated by the members of the information source organization in order to support a closed group of client users. In many cases the client group is part of the organization which supports its interactions by means of the federation. Furthermore there exist dependencies between clients and information sources due to their identical organizational origin. The information sources act in a common interest anchored in the organization they belong to. This network of dependencies probably has had significant impact on specific security architectures as often found in federated systems. As mentioned above, the information sources directly belong to a holding organization and therefore are trusted. Most federated systems do not

authenticate information sources in a client-verifiable way. The methods used to integrate information sources at the federation layer often involve administrative interaction, which takes place before client queries can be accepted. This suggests, that the set of information source layer members is rather static than dynamic. Under the assumption of a closed world the static nature of this approach leads to temporary loss of service when one or more information sources, which hold information relevant to the client query, are unavailable.

Opposed to information sources the federation clients are not trusted and require proper authentication and authorization. Because clients normally are members of the federation's organization, there is a closed group of registered users. New users are assigned predefined roles but some systems also support anonymous client accesses.

The motivation to integrate information sources using mediators is quite different. Clients demand systems enabling them to effectively work with heterogeneous information sources. This demand stimulates information sources to supply their information on an ad-hoc basis, in particular for purchase. Information sources are likely to meet the client's requirements and to cooperate with mediators. Like in a marketplace of supply and demand there exist different motivations for cooperation in each layer. Generally clients, information sources and mediators are independent of each other. Information sources exist in competitive and non-competitive relationships with other information sources. Obviously there is no base for mutual trust between the three layers of a mediated system. While information sources probably will have cooperation contracts with mediators, it can be assumed that spontaneous clients are unknown beforehand. Clients thus cannot be registered in a static way before queries can be accepted. Even the group of information sources probably won't be as stable as found in federated systems due to the lack of organizational associations in mediated systems. Though one or more information sources may be temporarily unavailable a client query can be satisfied. There apparently is no useful assumption of a closed world in mediated systems due to their dynamics. Other new requirements relate to non-repudiation issues (e.g. origin, affirmative authorization) for traded items.

We believe that mediated systems are more suitable to model dynamics and low trust of interacting parties. A mediator's top-down design paradigm allows for a stable presentation schema of integrated information at varying degrees of source fluctuation, whereas a bottom-up approach requires a redesign of the global integration schema each time a local schema changes or is added. Mediators strive for tolerance with respect to information provider failures and offer service to ad-hoc clients which have not registered with the service beforehand. The latter forbids employment of merely identity based identification approaches as traditionally used in federated systems. This paper shows a possible approach to achieve secure mediation while considering our trust model and high dynamics in a certain mediation scenario as presented in the following sections.

3 SCENARIOS

Basically there are two extreme scenarios for mediator security handling imaginable: *simple forwarding* or *complete mediation* of security information. Both scenarios feature benefits and drawbacks, of which some will be outlined here. Based on this analysis we will postulate a hybrid scenario that will be taken as a motivation for our approach.

In the *simple forwarding* scenario security requirements and their fulfillment travel back and forth between clients and information sources, while being forwarded completely unmodified and uncomplemented by the mediator. That is, all three layer participants can authenticate each other. When identities are authenticated, it is difficult to allow anonymous clients. On the other hand information sources know their clients and may use fine grained authentication, authorization and accountability. The sources inform clients about their security requirements via the mediator. In this scenario the mediator does not complement or modify these specifications such that the clients are completely aware of each used source's security requirements. A client may profit from a wealth of detail, but it is the client software's business to present a general view of a query's security requirements. Since in this scenario the mediator does not provide an integration layer for source policies, it is the client software's duty to do that. Consequently, the necessity to trust the mediator is limited to forwarding security information properly and privacy preserving.

A mediator which provides *complete mediation* of security information retrieves security requirements from information sources and integrates them. It presents the clients a coherent view of security requirements for a given query. This layer of abstraction conforms with the presentation of external objects. On the other hand an isolation of clients and sources is artificially created. While this allows for anonymous clients, it has severe drawbacks on the granularity of authentication, authorization and accountability at the sources. Sources cannot destine query results for specific clients and the latter cannot directly determine the origin of results. Obviously in this scenario clients as well as information sources need to trust the mediator.

Our approach is based on a compromise of both of the above scenarios to get the best of both worlds. It is one of a mediator's design goals to integrate information and we seek to achieve integration for security information, too. On the other hand we think that it is necessary to provide information sources and clients with sufficient information to establish a secure relationship via the mediator. Further it is a goal to protect the client's privacy and to minimize the necessary trust towards the mediator. In our hybrid scenario the mediator integrates source requirements and lets the clients choose, how much information to divulge about themselves. Only minimal necessary information about the clients then is sent to the sources. Subsequently, they can authenticate, authorize and audit the clients and appropriately protect results. The mediator cannot use results in a fraudulent way but is still able to integrate results of different sources.

4 DESIGN OF SECURE MEDIATION

4.1 Fundamental requirements of secure querying

The following security requirements for querying are considered: 1. Any source wishes or is even legally obliged to autonomously follow a security policy with respect to confidentiality which ensures that requested information is delivered to appropriate clients only. In order to achieve this goal, clients have to provide evidence that they are eligible for requested information, and sources have to maintain mechanisms to inspect such evidence and to decide whether and which information is returned. Furthermore, a source has to ensure that information is actually delivered to only that client which provided the inspected evidence. 2. The policy with respect to confidentiality as stated above should be at least compatible with additional viewpoints concerning authenticity, anonymity, integrity and availability. 3. And any client wishes that shown evidences cannot be misused.

Surely, these fundamental requirements should be met for the simple case that a client directly addresses a source, as well as when both the client's request and the source's delivery are mediated.

4.2 Basic informational environment

We assume that there are trusted third parties (TTPs), trusted by all participants of a transaction, that offer at least the following services:

- A TTP signs a *certificate* of the rough form
(identity(address), public (encryption) key, public (verification) key),
thereby assuring that the *participant* specified by the first component is the owner of the keys.
- A TTP signs a *credential* of the rough form
(attribute, public (encryption) key, public (verification) key),
thereby assuring the *attribute* specified by the first component is enjoyed by the owner of the matching secret keys for decryption and signing.

Our basic protocols employ *attributes* contained in credentials, when shown to a source, as evidence that the owner of the matching secret keys might be eligible for some requested information. That is, a source decides on the basis of the presented attributes, whether and which information is returned. It is important to observe, that the source does not care how it has got knowledge of the credentials, whether directly from the owner of the matching secret decryption key or otherwise.

For the basic protocols we always only need the public *key for encryption* in credentials, as sketched in the following. Suppose that a participant wants to ensure that some returned data contain meaningful information only for the supposed owner of the matching secret decryption key. Then the participant

takes care that the delivered data is the ciphertext of the plaintext which contains the information under consideration, where the encryption is done with the public key. The other keys are merely provided as a precaution for more advanced protocols.

4.3 Secure direct querying

Given this basic informational environment, we can specify the basic protocols. We distinguish a preparatory phase and a query phase.

In the *preparatory phase*, clients and sources do not yet interact. A *client*, wishing to request information later on, assembles credentials with his attributes supposed to provide evidence of his eligibility. And a source, entitled to answer queries later on, defines a *security policy* with respect to confidentiality which relates sets of attributes to the amounts of information allowed for delivery. More precisely a security policy is abstracted to be specified in the following form:

- As input, the policy accepts some set of credentials belonging to a unique owner. For instance, this is the case if all occurring public keys are the same. It is important to observe that it is not necessary for the source to know the identity of that owner.
- Only based on the set of attributes shown by the credentials, the policy states which kind of information is allowed to be delivered to the owner.

The protocol for the *query phase* is outlined as follows.

Protocol for secure query answering.

1. The client sends a request (identity(address), query, set of credentials) to the source.
2. The source verifies each credential, checks whether the set of credentials is acceptable, i.e. belong to a unique owner, and determines the associated set of attributes.
3. The source evaluates the query under the restriction that only such information is generated that, on the basis of the associated set of attributes, is allowed to be delivered.
4. The result of the restricted query evaluation is considered as plaintext and encrypted with (some of) the public key(s) occurring in the shown credentials.
5. The resulting ciphertext is sent back to the client.

This protocol satisfies the fundamental security requirements, as stated in Section 4.1:

- a) Clients and sources exclusively have to trust the TTPs that signed credentials.
- b) Sources are supposed to have an interest in checking eligibility. Thus, for instance, data subjects, the data of which is stored in a source, have to trust the source with respect to checking eligibility appropriately.
- c) Eligibility is supposed to be definable in terms of attributes.
- d) Since attributes

are shown in the form of credentials that do not contain a field for the identity of the owner, a client can stay anonymous as far as the source cannot infer the identity from its knowledge about the attributes and the connection data. e) Even in an untrusted network, only the unique owner of the verified credentials can recover the plaintext and thus gain the requested information. However, in some situations, we should have to take care of possible plaintext attacks. These situations are given if an attacker is himself eligible for another user's eligible request. A possible countermeasure would be to employ nondeterministic encryption, i.e. adding some random data to the plaintext before encrypting it. f) If any participant misused somebody else's credentials, the correct owner could be erroneously or maliciously blamed for the request. However, in a dispute about the sender of the request nobody can exhibit any essential evidence pro or contra the blame. If there is an interest in documenting the sender of a request, the protocol must be extended by appropriately signing the request. g) Further concerns about authenticity or requirements on integrity could be dealt with by additional actions that are based on appropriate signing. These actions would be founded on the certificates offered by the basic informational environment. h) In particular, if a source is concerned about a client redistributing received data without the source's approval, the source can fingerprint the delivered copies of the data. i) There are no specific provisions or additional obstacles to availability.

4.4 Secure mediation

We now extend the approach presented in Section 4.3 for the case of mediation. To begin with, we ignore security requirements for the moment and just state a rough abstract protocol for mediated query answering, see also Figure 1

Protocol for mediated query answering.

Request phase.

- a) A client C sends a global query q to a mediator M .
- b) The mediator M decomposes the query q into a set of subqueries q_S , where the subquery q_S is supposed to be appropriate for some source S .
- c) The mediator sends the subquery q_S to the source S , for each relevant source S .

Delivery phase.

- d) Each relevant source S evaluates its subquery q_S and produces a local answer consisting of data d_S .
- e) Each relevant source S sends its local answer d_S back to the mediator M .
- f) The mediator M integrates the received local answers d_S into a global answer d .

Now taking care of the fundamental security requirements we can easily combine the basic protocols for secure query answering with the protocol for mediation. In the straightforward case the protocols for the preparatory phase

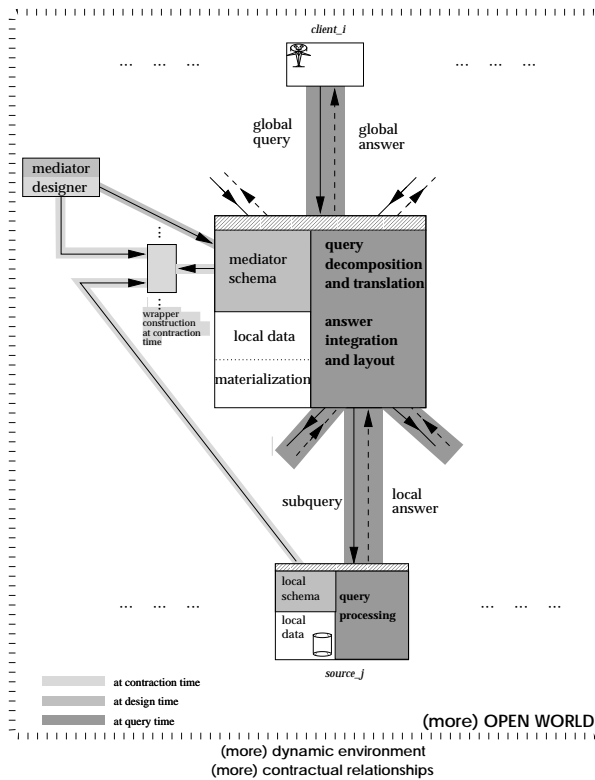


Figure 1 Design of mediator-based information integration

remain unchanged. And the mediation protocol is modified by integrating the basic protocol for the query phase as follows. In step a) the client includes a set of credentials into the request for query q . In step c) the mediator just forwards the received set of credentials to each of the relevant sources. In step d) each relevant source performs the security actions of step 2) of the basic protocol, and query evaluation is restricted as stated in step 3) of the basic protocol. Finally, in step e), each relevant source first encrypts its local answer according to step 4) of the basic protocol before sending it back to the mediator.

It can be checked that the fundamental security requirements are invariantly satisfied as for the simple case. There are only some minor restrictions. We note two aspects. Now another participant, the mediator, acquires knowledge about the client's credentials and thus could give raise to false blames about the sender of a request. But this possibility does not introduce a substantially new problem. And a mediator may compromise the integrity of data, if no additional actions are taken. There is also an improvement with respect to a client's wish to stay anonymous with respect to a source, since in general there is no need for a direct connection between the client and a source.

However, there are important observations dealt with in section 4.5. Firstly, the functional requirements on the mediator may be seriously affected if in the integration step f) the expected operations for integrating local answers can not be performed on the ciphertexts. And secondly, step b) can be greatly improved by facilities of the mediator to assist a client in the management of credentials.

4.5 *Advanced secure mediation*

Layered mediation. So far we treated the case that there is only one layer of mediation. However, we have argued that mediation does not essentially affect the fundamental security properties of direct querying, and thus we could use our approach also for mediation across several layers.

Referencing and using public encryption keys. In Section 4.2 we simply assumed credentials to contain the public encryption key of the attribute's owner. And in Section 4.3 we showed a straightforward way how a source can employ such an encryption key. These features allow some useful variations. Firstly, there is no essential need for including the public encryption key in the credential. In place of the key itself it is sufficient to equip the credential with information on how to retrieve the key of the attribute's unique owner. And secondly, confidentiality of delivered information can also be ensured as follows: The source encrypts the answer with any session key using any encryption method, and it encrypts only the session key with the public key. Then both the ciphertext and the encrypted session key are returned to the client.

Mediated management of credentials. In Section 4.4 we presented a modified mediation protocol, in which the mediator during step c) just forwards the received set of credentials to each of the relevant sources. There is room for a lot of important improvements which, basically, assist a client in managing his credentials. The most important issues to be addressed are: A client may wish to present a minimal set of credentials to each of the relevant sources. He may also require that, if there is any choice to answer his global query, the mediator should decompose the query in such a way that subqueries are sent to sources with minimal credential requirements. More generally, a client would like to be assisted in revealing as few of its attributes as possible. On the other hand, a client may specify a wanted level of quality with respect to the global answer to his query. This goal requires that the mediator takes best advantage of all available credentials. More generally, a client would like to be assisted to achieve a maximal level quality of the answer. Obviously, in general there will be a tradeoff between minimizing the use of credentials and maximizing the quality of information. Accordingly, a client would like to be assisted in balancing the conflicting goals. Even more generally, a client would like to negotiate with the mediator which set of credentials he is willing to submit. Additionally, due to heterogeneity, the formats of the credentials currently at the client's disposal may not be accepted by some of the possible

sources. In this case, the client would like to be assisted in getting reformatted credentials from some of the TTPs. For these and similar tasks, the mediator has to be able to resolve all kinds of heterogeneity among the security policies of the sources. Thus the characteristic services of mediators with respect to pure query answering should be extended to dealing with security policies as well. Moreover, the mediator, having its own mediator schema and its own local data and possibly also materialized data from previous queries, could have its own mediator security policy. Surely, such a mediator security policy must be suitable to integrate appropriate views on the various security policies of the sources. For this purpose, the mediator security policy should be considered as part of an extended mediator schema, and accordingly it should be declared during the preparatory phase. Furthermore, whenever a source is contracted to participate in the mediated information system, an appropriate security wrapper has to be constructed from the given mediator policy and the source policy.

Apparently all these and other related issues could be treated in many different ways. We argue that exploiting features of object orientation and of role based evaluation control are most promising. *Object orientation* is used for a unified view of all parts of the information system, and for providing appropriate granularities of controlled units. *Role based* control is selected for being application oriented and for its potential to integrate and unify various policies. Finally *evaluation control* is meant to combine aspects of access control, to be exercised mainly when invoking an operation, and of information flow control, to be exercised mainly when returning the result of a (nearly) completed operation. Proposing a specific object oriented role based evaluation control model is beyond the scope of the present paper.

Integration of local answers - functionality versus confidentiality. As already observed in Section 4.4, during the delivery phase of the protocol for mediated and secure query answering we are faced with the problem that the following requirements may be conflicting: Firstly, the mediator has to integrate and possibly materialize the local answers, sent back to the mediator by the sources to be finally delivered to the client. And secondly, the mediator should not be able to break the security policies of the sources. In particular, ideally the mediator should not gain meaningful information from the partial answers.

We discuss several solutions to this problem. They vary in two parameters: the achieved functionality for integration and the required trust in the mediator necessary to keep partial answers confidential.

Pessimistic solutions. These solutions follow the specification as given in Section 4.4. Here the mediator operates on the ciphertexts only, and thus no trust in the mediator is necessary. Without any provisions, the achieved functionality for integration will be rather low. Essentially, the mediator can only annotate and forward the local answers. The functionality for integration can be improved if the mediator causes all sources to uniformly use a *privacy*

homomorphism [23, 30] for encrypting their local answers. Such a privacy homomorphism allows a subset of typical database manipulations on ciphertexts to be carried out as if they were executed on plaintexts. In order to employ a privacy homomorphism the mediator instructs all relevant sources to use an appropriate encryption method and the same session key. As discussed before, there are no essential limitations in doing so. Of course, in this situation the encryption method should be asymmetric because otherwise we could not guarantee confidentiality among the sources.

Optimistic solutions. These solutions allow the mediator to observe the local answers as plaintexts. Then the mediator can operate on local answers without any restrictions, but sources and clients have to put their trust on the mediator, at least to some extent. Surely, once the mediator has observed plaintext answers, the sources cannot technically enforce correct usage of the information gained. The best they can achieve is to bind the mediator to fixed obligations. Later on they can try to somehow supervise the behaviour of the mediator, and to blame the mediator for detected misuse. The following modification of the delivery phase is suitable for this purpose.

In step e) of the protocol, before sending local answers back to the mediator, the source performs the following actions: It *fingerprints* the copy of the data to be delivered such that later on that copy can be identified as devoted to the specific mediator [22]. It attaches binding *approvals* to the data. An approval of form $\langle \textit{distribute}, oid, S, M, C \rangle$ roughly states that "source S allows mediator M to distribute the content of (the object identified by) oid to client C ", and it is digitally signed by the source. And it *encrypts* the data using a public encryption key of the *mediator*.

In forthcoming disputes, the source can use the fingerprints to prove that the mediator has been delivered the data, and the mediator can use the approvals to prove that it has been allowed to further distribute the data.

However, not all possible problems are solved. Whenever later on the source claims that some further participant illegally holds a copy of the delivered data, then that copy may originate either from the mediator or the client who has issued the global query. The last case is also a problem without mediation. The new problem of mediation is to discriminate between misbehaviour of the client and misbehaviour of the mediator.

At the expense of additional security overhead all problems could be resolved with the same techniques sketched above, namely fingerprinting and approvals, now specific for the client (instead of specific for the mediator).

Materialization of local answers. Once the mediator has got local answers from the sources, it could materialize that data in order to reuse it for further queries. Obviously, on the one side materialization raises new variants of the old problems concerning functionality, confidentiality, trust and claim of origin. But on the other side it could increase the overall efficiency of the mediator. A

full treatment of all details of the interdependence of efficiency and security in the context of materialization is beyond the scope of this paper.

5 CONCLUSION

This paper has discussed the requirements for secure mediation, and it has presented the overall design and various advanced features for meeting them. A more detailed analysis and further topics for multimedia applications as well as promising areas of additional research and system development are sketched in the preproceedings and will be treated in more depth elsewhere.

Acknowledgments

We wish to thank Gerrit Bleumer and the anonymous reviewers for their valuable suggestions about draft versions of this paper.

References

- [1] Y. Arens, C. A. Knoblock, and W. Shen. Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems*, 6(2-3), 1996.
- [2] R. J. Bayardo et al. InfoSleuth: Agent-based semantic integration of information in open and dynamic environments. In *SIGMOD'97*, pages 195–206, Tucson, AZ, USA, May 1997.
- [3] J. Biskup, J. Freitag, Y. Karabulut, and B. Sprick. A mediator for multimedia systems. In *Proc. 3rd Int. Workshop on Multimedia Information Systems*, pages 145–153, Como, Italia, Sept. 1997.
- [4] J. Biskup, J. Freitag, Y. Karabulut, and B. Sprick. Query evaluation in an object-oriented multimedia mediator. In *4th Int. Conf. on Object-Oriented Information Systems*, pages 31–43, Brisbane, Australia, Nov. 1997, Springer-Verlag.
- [5] G. Bleumer and M. Schunter. Privacy oriented clearing for the German health-care system. In R. Anderson, editor, *Personal Information security, engineering and ethics*, pages 175–194. Springer-Verlag, 1997.
- [6] K. S. Candan, S. Jajodia, and V. S. Subrahmanian. Secure mediated databases. In S. Y. W. Su, editor, *12th Int. Conf. on Data Eng.*, pages 28–37, New Orleans, Louisiana, USA, February, March 1996. IEEE Computer Society Press.
- [7] M. J. Carey et al. Towards heterogeneous multimedia information systems: The Garlic approach. In *Proc. of the 5th Int. Workshop on Research Issues in Data Engineering(RIDE): Distributed Object Management*, pages 123–130, L. A., CA, 1995.
- [8] S. Castano, S. D. C. di Vimercati, and M. Fugini. Automated derivation of global authorizations for database federations. *Journal of Computer Security*, 5:271–301, 1997.

- [9] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, Oct. 1985.
- [10] S. D. C. di Vimercati and P. Samarati. Access control in federated systems. In *ACM New Security Paradigm Workshop*, pages 87–99, Lake Arrowhead, CA, 1996.
- [11] C. M. Ellison et al. Simple public key certification. Internet draft, work in progress. <http://www.ietf.org/ids.by.wg/spki.html>, Mar. 1998.
- [12] W. Essmayr, G. Pernul, and A. M. Tjoa. The security API of IRO-DB. In S. Katsikas, editor, *Proc. Joint IFIP TC 6 and TC 11 Working Conf. on Communications and Multimedia Security*, Athen, Greece, Sept. 1997. Chapman & Hall.
- [13] R. Hull and G. Zhou. A framework for supporting data integration using the materialized and virtual approaches. In *ACM SIGMOD'96*, pages 481–492. Montreal, Canada, June 1996.
- [14] S. Jajodia, P. Samarati, V. Subrahmanian, and E. Bertino. A unified framework for enforcing multiple access control policies. In *SIGMOD'97*, pages 474–485, Tucson, AZ, May 1997.
- [15] D. Jonscher and K. R. Dittrich. An approach for building secure database federations. In *Proc. of the 20th Int. Conf. on Very Large Databases*, pages 24–35, 1994.
- [16] D. Jonscher and K. R. Dittrich. Argos—A configurable access control system for interoperable environments. In *Proc. of the 9th Annual IFIP WG 11.3 Working Conf. on Database Security*, pages 43–60, Rensselaerville, NY, Aug. 1995.
- [17] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proc. of 22nd Int. Conf. on Very Large Data Bases VLDB'96*, pages 251–262, Mumbai (Bombay), India, Sept. 1996. Morgan Kaufmann.
- [18] W. Litwin, L. Mark, and N. Roussopoulos. Interoperability of multiple autonomous databases. *ACM Comput. Surv.*, 22(3):267–293, Sept. 1990.
- [19] L. Liu and C. Pu. Distributed interoperable object model and its application to large-scale interoperable database systems. In *Proc. of ACM Int. Conf. on Information and Knowledge Management (CIKM'95)*, 1995.
- [20] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Observer: an approach for query processing in global information systems based on interoperation accross pre-existing ontologies. In *1st IFCIS Int. Conf. on Cooperative Information Systems (CoopIS'96)*, Brussels, Belgium, June 1996.
- [21] Object Management Group. The CORBA security specification. <http://www.acl.lanl.gov/cgi-bin/doclist.pl>, 1996.
- [22] B. Pfitzmann and M. Waidner. Anonymous fingerprinting. In *Euro-Crypt'97*, LNCS, Berlin, 1997. Springer-Verlag.

- [23] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In R. DeMillo et al., editors, *Foundations of Secure Computation*, pages 169–177. Academic Press, NY, 1978.
- [24] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 2:38–47, 1996.
- [25] K. E. Seamons, W. Winsborough, and M. Winslett. Internet credential acceptance policies. In *Proc. of the Workshop on Logic Programming for Internet Applications*, Leuven, Belgium, July 1997.
- [26] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, Sept. 1990.
- [27] V. S. Subrahmanian, S. Adali, A. Brink, R. Emery, et al. HERMES: Heterogeneous reasoning and mediator system. Submitted for publication. <http://www.cs.umd.edu/projects/hermes/>.
- [28] A. Tomasic, L. Raschid, and P. Valduriez. Scaling heterogeneous databases and the design of DISCO. In *Proc. of the Int. Conf. on Distributed Computer Systems*, Hong Kong, 1995.
- [29] J. D. Ullman. Information integration using logical views. In *Proc. of the 6th Int. Conf. on Database Theory, ICDT'97*, LNCS, pages 19–40, Delphi, Greece, 1997. Springer-Verlag, Berlin.
- [30] N. R. Wagner, P. S. Putter, and M. R. Cain. Encrypted database design: Specialized approaches. In *IEEE Symposium on Security and Privacy*, pages 148–153, 1986.
- [31] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, 1992.
- [32] G. Wiederhold, M. Bilello, and C. Donahue. Web implementation of a security mediator for medical databases. In T. Y. Lin and S. Qian, editors, *Proc. of the 11th Annual IFIP WG 11.3 Working Conf. on Database Security*, pages 60–72, Lake Tahoe, CA, 1997. IFIP, Chapman & Hall.
- [33] G. Wiederhold and M. Genesereth. The conceptual basis for mediation. *IEEE Expert, Intelligent Systems and their Applications*, 12(5):38–47, Sept.–Oct. 1997.
- [34] L. L. Yang, T. Özsu, and L. Liu. Accessing heterogeneous data through homogenization and integration mediators. In *2nd IFCIS Conf. on Cooperative Information Systems (CoopIS-97)*, Charleston, SC, June 1997.

Index

Anonymity, 2, 5
Authenticity, 2, 5
Authorization, 3–4
Confidentiality, 2, 5, 10
Credential, 2, 5, 9
Federated database, 2

Fingerprint, 11
Materialization, 11
Mediation, 2, 4, 7
Mediator, 3
Multimedia, 12
Public encryption, 9
Role, 2, 10