Enriching Access Control to Support Credential-Based Specifications

Pierangela Samarati
Dipartimento di Tecnologie dell'Informazione
Università di Milano
samarati@dti.unimi.it

Abstract: Accessing information over the Internet has become an essential requirement in modern economy, and unknown parties can come together on the Net and interact for the purpose of acquiring or offering services. The open and dynamic nature of such scenario requires the development of new ways of enforcing access control. A promising direction is represented by the use of digital certificates, or credentials. To this purpose, new credential-based access control languages, models, and mechanisms need to be investigated.

1 Introduction

Today's Globally Internetworked Infrastructure connects remote parties through the use of large scale networks, such as the World Wide Web. Execution of activities at various levels is based on the use of remote resources and services, and on the interaction between different, remotely located, parties that may know little about each other. In such a scenario, traditional assumptions for establishing and enforcing access control regulations do not hold anymore. For instance, a server may receive requests not just from the local community of users, but also from remote, previously unknown users. The server may not be able to authenticate these users or to specify authorizations for them (with respect to their identity). The traditional separation between *authentication* and *access control* cannot be applied in this context, and alternative access control solutions should be devised. Early approaches departing from this assumption proposed associating authorizations with keys rather than with users' identities. This family of *trust management systems* (e.g., PolicyMaker [BFL96], Keynote [BFIK98], REFEREE [CFL$^+$97]) use credentials to describe specific delegation of trusts among keys and to bind public keys to authorizations. While these approaches provide an interesting framework for reasoning about trust between unknown parties, assigning authorizations to keys may result limiting and may make authorization specifications difficult to manage.

An alternative promising approach is represented by the use of *digital certificates* (or *credentials*), representing statements certified by given entities (e.g., certification authorities), which can be used to establish properties of their holder (such as identity, accreditation, or authorizations) [HFPS99]. Credential-based access control makes the access decision

of whether or not a party may execute an access dependent on properties that the party may have, and can prove by presenting one or more certificates (authorization certificates in [BFL96] being a specific kind of them). The development and effective use of credential based-access controls requires tackling several problems related to credential management and disclosure strategies, delegation and revocation of credentials, Establishment of credential chains, and protection against attacks. Several researchers have addressed the problem of establishing a Public Key Infrastructure (which is at the basis of credential-management); managing credentials and credential chains and developing strategies for automated trust negotiation, that is for determining the credentials to be required and released when interacting with other parties. Winslett's group, probably the first to investigate credential-based access control [SWW97, WCJS97a, WSJ00], has been working on strategies that can be applied in the *trust negotiation* process. As a matter of fact, credentials grant parties different choices with respect to what to release (or ask) the counterpart and when to do it. For instance, [WSJ00] distinguishes between *eager* and *parsimonious* credential release strategies. Parties applying the first strategy turn over all their credentials if the release policy for them is satisfied, without waiting for the credentials to be requested. Parsimonious parties only release credentials upon explicit request by the server (avoiding unnecessary releases). Yu et al. [YMW00] present a prudent negotiation strategy to the goal of establishing trust among parties, while avoiding disclosure of irrelevant credentials. [YS01] presents a family of possible credential disclosure strategies investigating their interoperability. Li et al. [LWM01] propose an algorithm to locate and retrieve credentials that are not available locally. [ABFK98, BK02] investigate credential-based access control in the context of mediated information systems, where autonomous sources can be brought together by mediators.

The successful use of credentials for enforcing access control, and the consequent application of all the different trust management strategies that can be though of, requires a fundamental problem to be solved: *parties must be able to state and enforce access rules based on credentials and communicate them to their counterpart*. The resolution of this problem requires the development of new access control (authorization) languages and systems.

2 Credential-based access control

The development of credential-based (or credential-supportive) access control requires the investigation of several issues, which we outline next.

- *Ontologies.* Due to the openness of the scenario and the richness and variety of security requirements and credential-based properties that may need to be considered, it is important to provide parties with a means to understand each other with respect to the properties they enjoy (or request the counterpart to enjoy). Therefore, common languages, dictionaries, and ontologies must be developed [CGM99].
- *Client-side restrictions.* The traditional distinction of client and server becomes loose as every party can behave as either a client or a server depending on the con-

text. Also, while it is true that for each specific interaction there can be a clear distinction in such a role, one assumption does not hold anymore: it is not only the server that establishes regulations. In traditional access control systems, clients need only to supply their identity (together with a proof for it), and servers need to support an access control system (i.e., include a system for stating and enforcing rules regulating access to their resources). Emerging scenarios require such ability to be supported by clients as well. Indeed, a client may—like a server—require the counterpart to fulfill some requirements. For instance, a client may be willing to release an AAA membership number only to servers supplying a credential stating that the travel agent is approved by AAA.

- *Credential-based access control rules.* Flexible and expressive languages able to express and reason about credentials need to be developed. Simple 'tuple-like' authorizations are obviously not sufficient anymore and richer languages are needed. Recent approaches toward flexible and multi-policy languages could be applied, but their consideration in the open credential-based scenario requires enriching the language to accommodate explicit reference and reasoning about certificates and party's properties. One major challenge in the development of the access control language is to find a proper balance between expressiveness and simplicity. The language must be very powerful and allow dynamic binding to properties. Recent approaches to provide richer form of access control are based on the use of logic-based languages (e.g., [JSSS01]). However, the logic-based paradigm is often seen in a reluctant way by many users unfamiliar with the concepts. For the access control language to be widely and effectively used by the general public, simplicity and easy management are a must, and languages attempting a trade-off between expressiveness and simplicity (possibly hiding the logic-based complication within the implementation while requiring users to provide simple declarative specifications) need to be investigated.

- *Access control evaluation and outcome.* The open nature of the scenario where credential-based access control operates changes the access control process itself. As a matter of fact, one of the reasons to move toward credential-based access control is that parties may be unknown to each other. On the one side, the server may not have all the information it needs in order to decide whether or not an access should be granted (and exploits certificates to take the decision). On the other side, however, the requester may not know which certificates she needs to present to a (possibly just encountered) server in order to get access. Assuming that the requester can hand over all the credentials it has is simply inconceivable: the requester will want the ability to send to the counterpart only *just what is needed* to get the access. All this requires a new way of enforcing the access control process, which cannot be assumed anymore to operate with a given prior knowledge and return a "yes/no" access decision. Rather, the access control process should be able to operate without a priori knowledge of the party requesting access and return the information of the requisites that it requires be satisfied for the access to be allowed. The access control decision is therefore a more complex process and completing a service may require communicating information not related to the access itself, but related to additional

restrictions on its execution, introducing possible forms of negotiation investigated in the automatic trust management strategies.

- *Policy communication.* Since access control does not return a definite access decision, but it returns the information about which conditions need to be satisfied for the access to be granted, the problem of communicating such conditions to the counterpart arises. To fix the ideas, let us see the problem from the point of view of the server (the client's point of view is symmetrical). The naive way to formulate a credential request—that is, giving the client a list with all the possible sets of credentials that would enable the service—is not feasible, due to the large number of possible alternatives. In particular the precise nature of the credentials might not be known in advance (as it happens with chains of credentials), and in the presence of compound credential requests such as "*one* ID *and one* membership certificate from a federated association", there may be a combinatorial explosion of alternatives, as each individual request can potentially be fulfilled in many possible ways. Similar considerations apply to the requirements formulated for *classes* of services and inherited by their subclasses and instances; combinatorial explosion of inherited alternative requirements should be avoided, and the mechanism of rule attachment is a way of achieving this goal. However, the server cannot simply send its rules to the client. Some rules may query the server's private state information. For instance, a server may require a digitally signed guarantee to specific customers (who appear blacklisted for bad credit in some database it has access to); the server should simply ask this signed document, it should not tell the customer that she appears blacklisted. Clearly, the server should not send its private information to the client; it should then evaluate state predicates at its side. A further complication arises if the communicated conditions need to satisfy a *sufficient* criteria for the access (i.e., the client wants to be ensured that providing the requested credentials it will indeed be granted access), in which case all the server private information affecting the client's ability to access should be evaluated before communicating the requirements. In principle, other parts of the state, such as the client's preferences need not be hidden from the client, but they should be evaluated anyway before sending the rules to the client. The reason is that the client is not expected to bother with profile information submitted during previous interactions; profiles are maintained by the server precisely for the purpose of making client-server interactions more concise and less redundant.

A first attempt to provide a uniform framework for credential-based access control specification and enforcement was presented by Bonatti and Samarati in [BS02]. They propose a uniform framework for regulating service access and information disclosure in an open, distributed network system like the Web. Like in previous proposals, access regulations are specified as logical rules, where some predicates are explicitly identified. Besides credentials, the proposal also allows to reason about declarations (i.e., unsigned statements) and user-profiles that the server can maintain and exploit for taking the access decision. Communication of requisites to be satisfied by the requester is based on a filtering and renaming process applied on the server's policy, which exploits partial evaluation techniques in logic programs. The filtering process allows the server to communicate to the client the requi-

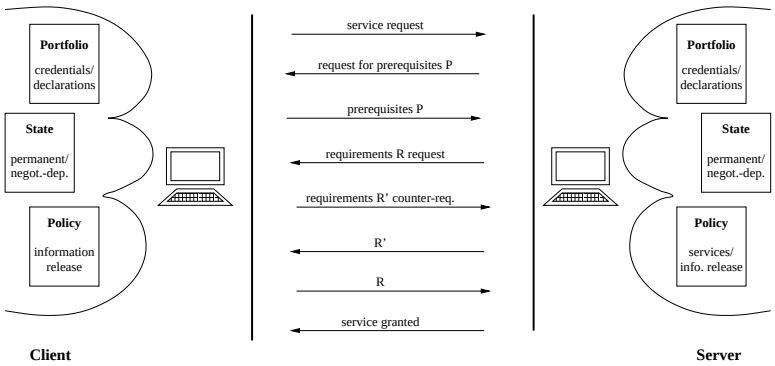

Figure 1: Client/server interplay in [BS02]

sites for an access, without disclosing possible sensitive information on which the access decision is taken. The proposal allows also clients to control the release of their credentials, possibly making counter-requests to the server, and releasing certain credentials only if their counter-requests are satisfied (see Figure 1). Client-server interplay is limited to two interactions to allow clients to apply a parsimonious strategy (i.e., minimizing the set of information and credentials released) when deciding which set credentials/declarations release among possible alternative choices they may have.

The proposal in [BS02], focused on the specification language and its communication, is complementary and couples well with related work previously discussed (e.g., [WCJS97b] on automated trust authentication and credential chain discovery). However, the work in [BS02] is only a first step and can be seen as a starting point that opens to the investigation of effective answers to the issues discussed above.

References

[ABFK98] C. Altenschmidt, J. Biskup, U. Flegel, and Y. Karabulut. Secure Mediation: Requirements and Design. In *Proc. of the 12th IFIP WG11.3 Working Conference on Database and Application Security*, Chalkidiki, Greece, July 1998.

[BFIK98] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The Role of Trust Management in Distributed Systems Security. In *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*. Springer Verlag – LNCS State-of-the-Art series, 1998.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized Trust Management. In *Proc. of 1996 IEEE Symposium on Security and Privacy*, pages 164–173, Oakland, CA, May 1996.

[BK02] J. Biskup and Y. Karabulut. A Hybrid PKI Model with an Application for Secure Mediation. In *Proc. of the 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security*, King's College, Cambridge, UK, July 2002.

[BS02] P. Bonatti and P. Samarati. A Unified Framework for Regulating Access and Information Release on the Web. *Journal of Computer Security*, 2002. (to appear).

[CFL+97] Y-H. Chu, Joan Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: trust management for Web applications. *World Wide Web Journal*, 2(3):706–734, 1997.

[CGM99] Chen-Chuan K. Chang and Hector Garcia-Molina. Mind Your Vocabulary: Query Mapping Across Heterogeneous Information Sources. In *Proc. of the 1999 ACM-SIGMOD*, pages 335–346, 1999.

[HFPS99] R. Housley, W. Ford, W. Polk, and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, rfc 2459 edition, January 1999. http://www.ietf.org/rfc/rfc2459.txt.

[JSSS01] S. Jajodia, P. Samarati, M.L. Sapino, and V.S. Subrahmanian. Flexible Support for Multiple Access Control Policies. *ACM Transactions on Database Systems*, 26(2):18–28, June 2001.

[LWM01] N. Li, W.H. Winsborough, and J.C. Mitchell. Distributed Credential Chain Discovery in Trust Management. In *Proc. of the Eighth ACM Conference on Computer and Communications Security*, Philadephia, PA (USA), 2001.

[SWW97] K. E. Seamons, W. Winsborough, and M. Winslett. Internet Credential Acceptance Policies. In *Proceedings of the Workshop on Logic Programming for Internet Applications*, Leuven, Belgium, July 1997.

[WCJS97a] M. Winslett, N. Ching, V. Jones, and I. Slepchin. Assuring Security and Privacy for Digital Library Transactions on the Web: Client and Server Security Policies. In *Proceedings of ADL '97 — Forum on Research and Tech. Advances in Digital Libraries*, Washington, DC, May 1997.

[WCJS97b] M. Winslett, N. Ching, V. Jones, and I. Slepchin. Using Digital Credentials on the World-Wide Web. *Journal of Computer Security*, 1997.

[WSJ00] W. Winsborough, K. E. Seamons, and V. Jones. Automated Trust Negotiation. In *Proc. of the DARPA Information Survivability Conf. & Exposition*, Hilton Head Island, SC, USA, January 25-27 2000. IEEE-CS.

[YMW00] T. Yu, X. Ma, and M. Winslett. An Efficient Complete Strategy for Automated Trust Negotiation over the Internet. In *Proceedings of 7th ACM Computer and Communication Security*, Athens, Greece, November 2000.

[YS01] T. Yu and M. Winslett K.E. Seamons. Interoperable Strategies in Automated Trust Negotiation. In *Proc. of the Eighth ACM Conference on Computer and Communications Security*, Philadephia, PA (USA), 2001.