

ACCESS CONTROL IN THE OPEN INFRASTRUCTURE

Pierangela Samarati samarati@dti.unimi.it
Dipartimento di Tecnologie dell'Informazione, Universita` di Milano, Italy.

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. In this paper we investigate some open issues and emerging approaches in the enforcement of access control in today's global networked infrastructure, in particular with reference to the support and management of digital identities and their support in the access control process.

1 INTRODUCTION

Nowadays, a global information infrastructure connects remote parties worldwide through the use of large scale networks, relying on application level protocols and services such as the World Wide Web. Execution of activities at various levels is increasingly based on the use of remote resources and services, and on the interaction between different, remotely located parties.

Digital identity management is becoming of paramount importance for supporting successful interaction in this scenario. Digital identity management supports privacy as it allows carrying out interaction on the Net without requiring identification of the human user and therefore the disclosure of all her properties (directly or possible through correlation with the identity). The use of digital identity also provides increased convenience: it does not require users to register at every step of complex network interactions and avoids the need of costly and error-prone profile maintenance at different servers. In a context where parties may interact

disclosing different nyms or only part of their identities, innovative models must be developed that identify under which conditions a party can trust others for their security and privacy. In fact, traditional assumptions for establishing and enforcing access control regulations do not hold anymore, since the traditional separation between authentication and access control does not apply, and alternative access control solutions should be devised.

Credential-based access control supports the use of certificates (or credentials) establishing a digital identity, and which represent statements certified by given entities, instead of requiring explicit authentication of the user to the server. 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 being a specific kind of them). In this paper we discuss the different issues that must be addressed for a successful development and use of digital identities and for the development of credential-based access control solutions.

2 DIGITAL IDENTITY MANAGEMENT

A digital identity is the electronic representation of the personal information of an individual or organization. The term digital identity is usually used to refer to two (non-disjoint) concepts: nyms and partial identities. Nyms can be used to give a user a different identity under which operate at any interaction. A partial identity is any subset of the properties (e.g., name, age, credit-card, employment, etc.) associated with a user. Partial identities may or may not be named (i.e., may or may not be related to the human identity of the user). To ensure successful identity management, a digital identity solution should support at least the following basic requirements.

- *Reliability and dependability.* Identity theft is one the fastest growing electronic crime and it is expected to accelerate. Digital identity must offer protection against forgery and related attacks. While their main goal is to protect and preserve individual users' anonymity, digital identities should fully guarantee other parties (e.g., suppliers and brokers in the framework of an e-business transaction) that the identity can be relied upon, and therefore obligations to the digital identity deriving from such a transaction will eventually be met by someone.
- *Controlled information disclosure.* Users must be given control on what identity to use in specific circumstances. Control must also be given with respect to possible replication and misuses of the identity information a party reveals in a transaction.

- *Mobility support.* The mobile computing infrastructure can keep track of an individual's physical location. In addition, mobile computing bears some peculiarity such as limited bandwidth and limited display size

Many concerns about dependability and multiple information management have emerged among the actors supposed to benefit from an identity management system: individuals (employees, partners and customers) and e-businesses. It is important, therefore, to understand open R&D issues that should be solved by future research. The most important of such information items is, of course, the one holding the digital identity itself. While many issues are likely to concur in influencing future digital identities' structure and semantics, ensuring dependability and, above all, controlled disclosure of personal information will undoubtedly play an important role. The major issues to be investigated include the following [2].

- *Identities life cycle management.* Effective identity management solutions require careful design of the full digital identity *life cycle*. Currently, the digital identities' lifecycle is modeled as a sequential multi-phase process, going from creation to termination phase throughout updating and maintenance. Such a sequential process, however, does not meet the requirements posed by multiple dependable identities. A novel digital identity-oriented life cycle must be defined as a structured *asynchronous* process that enables co-instantiation and joint evolution of all information items needed to support an individual in her/his different interactions with organizations and including provisioning, revocation, and profile management.
- *Digital identities representation.* Important issues to be solved concern the definition of how identity information should be represented and exchanged. The identity management service must support vocabulary definition for identity attributes as well as for control structures used in the protocol itself.
- *Cross-domain identity communication.* This research topic pertains a number of innovative features that are needed to enable reliable communication of multiple user identities among different domains. For instance, consider the scenario where a user provides an identity consisting only of her frequent flier number to an on line travel agency; in turn, the travel agency gives this information to an airline that finally grants to the user the permission to execute some actions. Regulating who should decide which information should compose an

identity and assessing how much can each partner trust assertions provided at each step is an important problem yet to be solved.

- *Architectural patterns for multiple identities' management.* Today's e-business architectures are based on the older (centralized)PKI concept and need to be adapted to the more modern concept of trust management based on digital identities. Centralized identity management techniques delegate identities' provisioning, maintenance, and revocation to a *trusted third party* (TTP), who may also be in charge of keeping track of the link between each user's multiple digital identities and her/his single physical one. While centralized architectures for identity management are robust and usually simpler to design and implement, they do not fit all digital identity application requirements (e.g., mobile hand-held devices need to store digital identities on board). Also, TTPs manage digital identities for large numbers of users. Digital identity management systems must scale to support the data volumes produced by large user populations. Therefore, new architectural patterns must be developed and tested. Hybrid solutions look particularly promising to achieve robustness and efficiency.
- *Identity administration.* Maintaining multiple identities as separate and independent named sets of attributes or credentials obviously poses huge management problems. Therefore, digital identity solutions should provide protocols, tools, and techniques for fast and reliable update of credentials, as well as for efficient view computation on profiles in order to extract identities. Like database views, profile-based digital identities can be *materialized* or *virtual* according to application needs. Research is needed on how to achieve seamless and scalable view computation over profiles, selectively using partial encryption or transformations of profile data. Alternative approaches rely on *hidden attributes* that can be selectively disclosed when required. Selective disclosure can be implemented by means of multiple keys encryption and lends itself to user-side execution, while dynamic view computation can only be executed on a server. Identity management solutions should also be integrated with personalization solutions in order to allow the reuse of profiles.
- *Anonymity support and dependability.* Digital identity management must provide users the ability of remain anonymous if identity information is not required in a transaction. Anonymity does not imply that no information at all be release, but requires that information released be non identifiable. Information, while anonymous, must be proved reliable.

- *Controlled dissemination of authenticated information.* Digital identity technology must operate in an environment with well-defined trust models underlying interaction between all parties involved. Users must be given control on which identity to use and on which attributes to disclose to their counterparts.
- *Trust management.* In a federated identity management system, users may obtain credentials in many different ways. Entities trying to verify such credentials often have no direct means of assessing their trustworthiness. From the user's point of view, one of the main problems with today's Web systems is the requirement to have a different password for each system to which he/she requires access rights. Password management addresses this problem; current solutions enable users to access multiple distributed resources with *single sign-on* (SSO) facility - but the system relies on other partners to trust the authentication process used to approve the identity's credentials. Trust management in a privacy-enhancing environment for supporting digital-identities means defining methods for receiving reliable evidence about credentials and for assessing their degree of trustworthiness. In principle, *strong authentication* techniques such as tokens, smart cards, digital certificates or biometrics may be used to guarantee trust in a digital identity. The current trend is toward the provision of Internet-based *trust services*, which deal with various aspects of trust and are held accountable for the services they provide. Another aspect of confidence and trust is linked to the capability to evaluate and assess the security level of components, systems and services used to authenticate a user or to relay her authentication information. Scalable and robust solutions for trust management and sharing need to be provided, and several research issues need to be solved.

3 CREDENTIAL BASED ACCESS CONTROL

The consideration of digital identities requires also the development of new means for establishing and enforcing access control policies. The development of a credential-based (or credential-supportive) access control requires the investigation of several issues, including [1].

- *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.

- *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 context. 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. 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.

4 CONCLUSION

Today's globally networked 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 remote resources and services, and on the interaction between different, remotely located parties that may know little about each other. In addition, parties may want to keep control on the different identity properties to be released to others, possibly using different identities in different contexts. This paper has illustrated the main open problems to be addressed for managing digital identities and, in particular, their considerations in the development of novel access control solutions.

5 REFERENCES

- [1] Bonatti, P. and Samarati, P (2002), "A unified framework for regulating access and information release on the web" *Journal of Computer Security*, 10(3):241—272.
- [2] Damiani, E., De Capitani di Vimercati, S. and Samarati, P.(November-December 2003), "Managing multiple and dependable identities." *IEEE Internet Computing*.