From Authentication to Authorization

Somesh Jha
Computer Sciences Department
University of Wisconsin
Madison, WI 53717

February 7, 2001

1 Deficiencies of Public Key Infrastructures (PKIs)

PKIs focus on establishing a link between an entity and a key. Hence, PKIs provide a mechanism for *entity identification* by providing a link between an entity and a key. However, the real requirement in many systems is *authorization*, i.e., is a certain entity authorized to perform a specific action. For example, imagine a program for course registration at a university. Suppose a student, John Doe, wants to register for a graduate course in Computer Science. In order for a student to register for this course, he/she should be a graduate student in Computer Science or a undergraduate student with the requisite standing, e.g. a senior with GPA above 3.5. Therefore, the registration program needs to know that John Doe has the required credentials. In the PKI setting, John Doe will present a certificate (signed by a trusted certificate authority) which binds his identity to a key. This establishes that John Doe is interacting with the registration program. The program then consults a database to verify that John Doe has the required credentials. In this scenario, the PKI approach has the following disadvantages:

- *Authorization decisions are adhoc*
  The authorization decision is made by the program in a rather adhoc manner. This is because PKIs only enable entity identification. However, the real question is whether John Doe has the required credentials to register for the graduate course. In other words, PKIs simply provide entity identification and leave the authorization decisions to the program.

- *Privacy concerns*
  In certain contexts, because of privacy concerns an entity might want to present just enough credentials to perform a certain action. In the PKI setting, the complete identity of the entity is exposed.

There seems to be a disconnect between the capabilities that PKIs provide and the actual requirements of a software system. Our aim is to address this disconnect. Many deficiencies of PKIs are discussed by Clarke [Cla01]. We will address this disconnect along several dimensions.

2 Issues related to authorization

2.1 Languages for expressing authorization policies

Several languages and logics have been proposed for expressing authorization policies in distributed systems [BFIK99, LFG99, RL96, Li00]. A calculus for access control in distributed systems appears in [ABLP93]. Typical problem addressed in these frameworks is *compliance checking* [BFS98], i.e., given a request r, a set of credentials C, and a policy P, should the request r be granted? However, there are several restrictions in existing frameworks and problems relevant to authorization have not been addressed.

- In many situations, *credential extraction* is also required. For example, assume that a job J is going to migrate to a remote host H. Before moving to the host H, job J would like to know what credentials does the host require so that it can perform required actions on the host. We believe that the credential extraction problem has received scant attention in the literature. An initial investigation of the credential extraction problem appears in [SWW00].

- Some authorization languages require *monotonicity restrictions* [BFS98], i.e., an agent cannot revoke statements. However, in a realistic setting negative statements or revocation is a necessity. We will explore relaxing the “monotonicity restriction” in existing authorization languages.

2.2 Efficient enforcement of policies

Assume that the required policies have been expressed in a suitable authorization language. How can these policies be efficiently enforced in a distributed heterogenous environment? The problem is further complicated by dependencies between policies of different hosts. For example, host A might allow an agent z to read a certain file if host C believes that z has certain characteristics. Therefore, enforcement of a certain policy might require communication with other hosts. Distributed enforcement of authorization policies has not received adequate attention. Abstractly speaking, this is the distributed version of the compliance checking problem described earlier. We plan to address compliance checking in a distributed environment.

2.3 Cross-administrative authentication

2.4 Usability

References

[ABLP93] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(3), September 1993.

[BFIK99] M. Blaze, J. Feigenbaum, J. Ioannidis, and A.D. Keromytis. The role of trust management in distributed systems. In *Secure Internet Programming*, volume 1603 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1999.

[BFS98] M. Blaze, J. Feigenbaum, and M. Strauss. Compliance-checking in the policymaker trust managment system. In *Proceedings of Financial Crypto*, volume 1465 of *Lecture Notes in Computer Science*, pages 254–274. Springer-Verlag, Berlin, 1998.

[Cla01] R. Clarke. Conventional public key infrastructure: An artefact ill-fitted to the needs of the information society. In *Proceedings of European Conference on Information Systems (ECIS)*, June 2001. http://www.anu.edu.au/people/Roger.Clarke/II/PKIMisFit.html.

[LFG99] N. Li, J. Feigenbaum, and B.N. Grosof. A logic-based knowledge representation for authorization and delegation. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, July 1999.

[Li00] N. Li. *Delegation Logic: A Logic-based Approach to Distributed Authorization*. PhD thesis, New York University, September 2000.

[RL96] R.L. Rivest and B. Lampson. SDSI-a simple distributed security infrastructure. 1996. http://theory.lcs.mit.edu/ cis/sdsi/sdsi11.html.

[SWW00] K.E. Seamons, W. Winsborough, and M. Winslett. Internet credential acceptance policies. 2000. http://www.transarc.com/ winsboro/papers/CAP.html.