

Technical Report NAS-TR-0042-2006  
Understanding Equivalence in High-Level and Information Flow Policy

Patrick McDaniel

## 1 Introduction

Information flow policies (labels and lattices) are not stated in terms that administrators and developers articulate security goals (natural language). This raises an important question central to the proposed investigation, “how do you translate higher level policies into information flow implementation?” Our approach is to develop models and algorithms that enable this translation.

Consider the following simple access control policy: *Any principal of the group `ifPlayers` can read a sensitive account balance from a local file owned by them.* We assert that principal `Alice` is authenticated by password and is a member of the group `ifPlayers`. One possible implementation of this policy using information flow would proceed using delegation; that is, `ifPlayers` delegates to `Alice` (and the other members) through the principal hierarchy. To enforce the access rights, the program implements an `openLocalFile` function whose input is labeled (can only be called with data of equal or higher sensitivity than) `ifPlayers` and returns a file object labeled with the calling principal.

The high-level policy is implemented by the information flow enforcement. The principal hierarchy ensures the group rights are enforced; no one other than the group members can call the function that reads the local file because they cannot produce input data of the correct label. The returned balance object is labeled with the sensitivity of the calling principal. `Jif` guarantees that no principal other than the caller (and the principals it delegates to) can access the object, and thus nobody can else read it (even other members of the group). The compilation of the code is a formal witness to this isolation.

Such information flow policy enforcement is not a panacea; it cannot combat poor operational practices, bad cryptography, or bad policy. However, as in this case, it can prove that the code written to implement that policy does implement that policy. What remains is discovering how to formalize and automate the mapping from high-level policies to low-level policy implementations.

## 2 Testing Policy Equivalence

The above examples raise an important question, “how do you know that an information flow implementation is equivalent to the high-level policy?” A key first task in this work is to develop a system for evaluating exactly this property of *policy equivalence*. In the context of this work, we initially restrict ourselves to a definition of a high-level policy as a formulation of authentication (requirements) and access control policy and restrict information flow policy to confidentiality. Broadly speaking, policy has been used in different network contexts as a vehicle for representing, among many others, authorization [1, 2, 3, 4, 5, 6, 7, 8, 9], peer or group session security [10, 11, 12, 13, 14, 15], quality of service guarantees [16], and network configuration [17, 18]. Access control policies, such as those formulated in an access control matrix by Lampson [19] and later by Graham and Denning [20], specify discretionary access control by stating the rights subjects have on a particular object [21, 22, 23, 24, 25, 26, 27, 28, 29].

### Policy Statements

[user] <b>when</b> [constraint]	indicates the [user] is authenticated using the [constraint] method.
[user] <b>reads</b> [object]	states that the [user] can read the [object].
[delegator] <b>delegates to</b> [user]	indicates [user] may assume all rights of the [delegator].

### Balance Policy

```
Alice when password
ifPlayers reads balance
ifPlayers delegates to Alice
```

Figure 1: A high-level policy specification and formulation of the balance policy

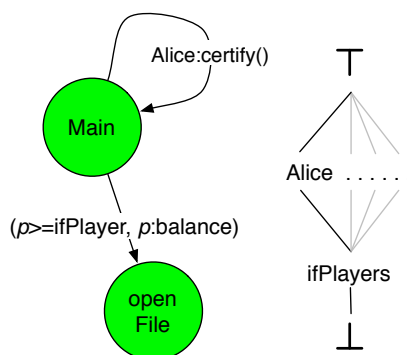


Figure 2: A low-level policy implementation.

Similar formalisms have been given for other kinds of isolation and detection policy [15, 30]. Several have considered correctness policy enforcement based on formal models [31, 32, 33, 34, 35, 36, 37, 38]. However, these efforts are largely limited to single systems and few have provided meaningful enforcement guarantees.

One of the central issues in the construction and use of any policy is the analysis of *safety* [39, 40, 41]. In the context of access control policy, safety determines if a policy system can reach a state in which a particular right is granted to a subject who does not initially have that right. Safety has been studied in many policy systems [42, 43, 27, 41, 44, 45, 46, 6]; it remains undecidable for many useful policy systems. Li and others have extended this analysis into a broad spectrum of *security analysis*, where other system properties are scrutinized [40, 47], e.g., availability. Closely related to policy equivalence, there is a considerable body of literature studying the comparative expressive power of access control systems [48, 49, 50, 51, 52, 53, 54]. Recently, Tripunitara and Li introduced a unified state transition-based modeling framework for comparing access control schemes [29]. The approaches that follow are guided in spirit by these works, and will exploit and extend their theoretical machinery as available and possible.

We define policy equivalence in a manner similar to prior works in expressive comparison: two policies are equivalent if for all possible system states they provide the same access. Equivalence is a syllogism of *soundness* and *completeness*, where soundness states that all accesses rejected by the high-level policy should be rejected by the low-level policy and completeness property requires that all accesses allowed by the high-level should be allowed by the low-level policy.

We now informally define an expository approach to test equivalence for limited confidentiality pol-

icy models. Illustrated in Figure 1, a model of high-level policy consists of access control statements, delegation, and authorization requirements/constraints (credentials/process needed to assume a principal identity). The figure also illustrates a simple language syntax (with obvious semantics) its use in the balance policy.

Our vastly simplified model of an information flow policy is built upon the principal hierarchy, the program call graph, the required principal identity assumption constraints, and the return value labels on functions themselves. In this model, the implementation is represented by a graph (isomorphic to the call graph) where each edge is annotated with the tuple  $(caller, output\ data : level)$ . At each function in the code there exists an opportunity to assume another principal identity, we create a loop edge at the node labeled with the governing constraint. An information flow implementation of the balance policy is illustrated in figure 2. The semantics of this policy are straightforward: the simplified information flow controlled program ensures that every function is called by a principal with at least `caller` level (via input labels and as dictated by the principal hierarchy), and that the function returns an object of type `p` with label `level`. This model is realistic: one can easily construct a Jif program governed in this way. Because Jif performs all flow analysis within a function, we need only model the function’s external behavior.

For each principal and set of satisfied constraints, there exist a set of data items to which they will have access known as the *access set*. In the case of the high-level policy, this set can be quickly identified by computing the transitive closure of the delegations of the principal identities satisfied by the constraints. Because the constraints above are monotonic, the access sets for a program state in which multiple constraints are satisfied is simply the union of the access sets of the individual executions.

We now compute the access sets of the information flow policy via simulated execution as follows: begin at the main node with the public ( $\perp$ ) principal with an empty access set for each principal identity. The algorithm then simulates the recursive traversal of all arcs that are labeled with principal identity. This process is repeated until all possible transitions are exhausted. Note that there may appear to be a potential for infinite recursion due to cycles in the graph and alternating principal identity assumption. This is not an issue because the total access cannot increase on successive visits under the same identity.

For principal  $p$  and satisfied constraints  $c$ , denote the access set of the high-level policy as  $H(p, c)$  and the policy implementation as  $L(p, c)$ . We formulate an equivalence test as:

$$\forall u_i \in U, c_j \in C \cup \emptyset \quad | \quad H(u_i, c_j) = L(u_i, c_j)$$

Note again that because of the monotonicity of the constraints, we need only check each constraint access individually, rather than all possible subsets of  $C \cup \emptyset$ .

## References

- [1] T. Woo and S. Lam. Authorization in Distributed Systems; A New Approach. *Journal of Computer Security*, 2(2-3):107–136, 1993.
- [2] M. Blaze, J. Feigenbaum, and Jack Lacy. Decentralized Trust Management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, November 1996. Los Alamitos.
- [3] Y. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: Trust Management for Web Applications. In *Proceedings of Financial Cryptography ’98*, volume 1465, pages 254–274, Anguilla, British West Indies, February 1998.
- [4] M. Blaze, J. Feigenbaum, John Ioannidis, and A. Keromytis. The Role of Trust Management in Distributed Systems Security. In *Secure Internet Programming: Issues in Distributed and Mobile*

*Object Systems*, volume 1603, pages 185–210. Springer-Verlag Lecture Notes in Computer Science State-of-the-Art series, November 1999. New York, NY.

- [5] Trevor Jim. SD3: A Trust Management System with Certified Evaluation. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, May 2001.
- [6] L. Cholvy and F. Cuppens. Analyzing Consistency of Security Policies. In *1997 IEEE Symposium on Security and Privacy*, pages 103–112. IEEE, May 1997. Oakland, CA.
- [7] Sushil Jajodia, P. Samarati, and V. Subrahmanian. A Logical Language for Expressing Authorizations. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 31–42, Oakland, CA, March 1997. IEEE.
- [8] T. Woo and S. Lam. Designing a Distributed Authorization Service. In *Proceedings of INFOCOM '98*, San Francisco, March 1998. IEEE.
- [9] T. Ryutov and C. Neuman. Representation and Evaluation of Security Policies for Distributed System Services. In *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 172–183, Hilton Head, South Carolina, January 2000. DARPA.
- [10] H. Harney and C. Muckenhirn. Group Key Management Protocol (GKMP) Architecture. *Internet Engineering Task Force*, July 1997. RFC 2094.
- [11] Y. Amir and J. Stanton. The Spread Wide Area Group Communication System. Technical Report CNDS-98-4, The Center for Networking and Distributed Systems, The Johns Hopkins University, 1998.
- [12] Patrick McDaniel, Atul Prakash, and Peter Honeyman. Antigone: A Flexible Framework for Secure Group Communication. In *Proceedings of the 8th USENIX Security Symposium*, pages 99–114, August 1999. Washington, DC.
- [13] J. Zao, L. Sanchez, M. Condell, C. Lynn, M. Fredette, P. Helinek, P. Krishnan, A. Jackson, D. Mankins, M. Shepard, and S. Kent. Domain Based Internet Security Policy Management. In *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 41–53, Hilton Head, South Carolina, January 2000. DARPA.
- [14] Hugh Harney, Andrea Colegrove, and Patrick McDaniel. Principles of Policy in Secure Groups. In *Proceedings of Network and Distributed Systems Security 2001 (NDSS)*, pages 125–135. Internet Society, February 2001. San Diego, CA.
- [15] Patrick McDaniel and Atul Prakash. Methods and Limitations of Security Policy Reconciliation. *ACM Transactions on Information and System Security*, 2006. Accepted for publication.
- [16] David C. Blight and Takeo Hamada. Policy-Based Networking Architecture for QoS Interworking in IP Management. In *Proceedings of Integrated Network Management VI, Distributed Management for the Networked Millennium*, pages 811–826. IEEE, 1999.
- [17] S. Bellovin. Distributed Firewalls. *login.*, pages 39–47, Nov 1999.
- [18] Yair Bartal, Alain J. Mayer, Kobbi Nissim, and Avishai Wool. Firmato: A Novel Firewall Management Toolkit. In *IEEE Symposium on Security and Privacy*, pages 17–31, 1999.

- [19] Butler W. Lampson. Protection. *ACM Operating Systems Review*, 8(1):18–24, January 1974.
- [20] G. Scott Graham and Peter J. Denning. Protection Principles and Practice. In *Proceedings of the AFIPS Spring Joint Computer Conference*, pages 417–429. AFIPS Press, May 1972. volume 40.
- [21] Elisa Bertino, Claudio Bettini, Elena Ferrari, and Pierangela Samarati. An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning. *ACM Transactions on Database Systems*, 23(3):231–285, 1998.
- [22] Elisa Bertino, Claudio Bettini, and Pierangela Samarati. A Temporal Authorization Model. In *CCS '94: Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 126–135, New York, NY, USA, 1994. ACM Press.
- [23] *A Guide to Understanding Discretionary Access Control in Trusted Systems*, first edition, September 1987.
- [24] D. D. Downs, J. R. Rub, K. C. Kung, and C. S. Jordan. Issues in Discretionary Access Control. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 208–218, April 1985.
- [25] Teresa F. Lunt. Access Control Policies: Some Unanswered Questions. In *Proceedings IEEE Computer Security Foundations Workshop*, pages 227–245, 1988.
- [26] P. Samarati and S. De Capitani di Vimercati. Access Control: Policies, Models, and Mechanisms. In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, 2171 in Lecture Notes in Computer Science. Springer-Verlag, 2001.
- [27] J. A. Solworth and R. H. Sloan. A Layered Design of Discretionary Access Controls with Decidable Safety Properties. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy, 2004*, pages 56–67, 2004.
- [28] Jon A. Solworth and Robert H. Sloan. Security Property Based Administrative Controls. In *ESORICS*, pages 244–259, 2004.
- [29] Mahesh V. Tripunitara and Ninghui Li. Comparing the Expressive Power of Access Control Models. In *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 62–71, New York, NY, USA, 2004. ACM Press.
- [30] Hao Wang, Somehs Jha, Patrick McDaniel, and Miron Livny. Security policy reconciliation in distributed computing environments. In *Proceedings of 5th International Workshop on Policies for Distributed Systems and Networks (Policy 2004)*, pages 137–146. IEEE Computer Society Press, June 2004. Yorktown Heights, NY.
- [31] F. Corbato and V. Vyssotsky. Introduction and Overview of the Multics System. In *Proceedings of the Fall Joint Computer Conference*, volume 27, pages 185–196, Jun 1965.
- [32] W. Wulf, C. Wang, and D. Kienzle. A New Model of Security for Distributed Systems. In *Proceedings of the 1996 Workshop on New Security Paradigms*, pages 34–43, 1996.
- [33] J. Shapiro, J. Smith, and D. Farber. EROS: A Fast Capability System. In *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*, pages 170–185, 1999.

- [34] C. Hawblitzel and C. Chang. Implementing Multiple Protection Domains in Java. In *Proceedings of the USENIX Annual Technical Conference*, pages 259–270, Jun 1998.
- [35] M. Hibler, D. Anderson, and J. Lepreau. The Flask Security Architecture: System Support for Diverse Security Policies. In *Proceedings of the Eighth USENIX Security Symposium*, pages 123–139, 1999.
- [36] T. Jaeger, R. Sailer, and X. Zhang. Analyzing Integrity Protection in the SELinux Example Policy. In *Proceedings of the 12th USENIX Security Symposium*, pages 59–74, 2003.
- [37] T. Jaeger, X. Zhang, and F. Casheda. Policy Management Using Access Control Spaces. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):327–364, Aug 2003.
- [38] G. Zanin and L.V. Mancini. Towards a Formal Model for Security Policies Specification and Validation in the SELinux System. In *Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies*, pages 136–145, 2004.
- [39] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in Operating Systems. *Communications of the ACM*, 19(8):461–471, August 1976.
- [40] Bowen Alpern and Fred B. Schneider. Recognizing Safety and Liveness. *Distributed Computing*, 2(3):117–126, 1987.
- [41] Ninghui Li and Mahesh V. Tripunitara. Safety in Discretionary Access Control. In *2005 IEEE Symposium on Security and Privacy*. IEEE, MAY 2005. Oakland, CA.
- [42] Paul Ammann and Ravi S. Sandhu. Safety Analysis for the Extended Schematic Protection Model. In *IEEE Symposium on Security and Privacy*, pages 87–97, 1991.
- [43] Trent Jaeger and Jonathon E. Tidswell. Practical Safety in Flexible Access Control Models. *ACM Trans. Inf. Syst. Secur.*, 4(2):158–190, 2001.
- [44] Ninghui Li, William H. Winsborough, and John C. Mitchell. Beyond Proof-of-Compliance: Safety and Availability Analysis in Trust Management. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 123–139. IEEE Computer Society Press, May 2003.
- [45] Ravi S. Sandhu and Paul Ammann. One-Representative Safety Analysis in the Non-Monotonic Transform Model. In *Proceedings IEEE Computer Security Foundations Workshop*, pages 139–149, 1994.
- [46] W. H. Winsborough and Ninghui Li. Safety in Automated Trust Negotiation. In *Proceedings of the IEEE Symposium on Security and Privacy, 2004*, pages 147–160, 2004.
- [47] Ninghui Li, John C. Mitchell, and William H. Winsborough. Beyond Proof-of-Compliance: Security Analysis in Trust Management. To appear in *Journal of the ACM*.
- [48] R. S. Sandhu. Expressive power of the schematic protection model. *Journal of Computer Security*, 1(1):59–98, 1992.
- [49] R. S. Sandhu and S. Ganta. On testing for absence of rights in access control models. In *Proceedings of the sixth Computer Security Foundations Workshop*, pages 109–118. IEEE Computer Society Press, June 1993.

- [50] P. Ammann, R. Lipton, and R. S. Sandhu. The expressive power of multi-parent creation in monotonic access control models. *Journal of Computer Security*, 4(2-3):14–165, January 1996.
- [51] S. Ganta. *Expressive Power of Access Control Models Based on Propagation of Rights*. PhD thesis, George Mason University, 1996.
- [52] R. S. Sandhu and Q. Munawer. How to do discretionary access control using roles. In *Proceedings of the Third ACM Workshop on Role-Based Access Control (RBAC 1998)*, pages 47–54, October 1998.
- [53] S. Osborn, R. S. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security*, (2):85–106, May 2000.
- [54] A. Chander, D. Dean, and J. C. Mitchell. A state-transition model of trust management and access control. In *In Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pages 27–43. IEEE Computer Society Press, June 2001.