A User Study of Policy Creation in a Flexible Access-Control System

Lujo Bauer[†] lbauer@cmu.edu
Lorrie Faith Cranor[†] lorrie@cmu.edu
Robert W. Reeder[†] reeder@cmu.edu
Michael K. Reiter[†‡] reiter@cs.unc.edu
Kami Vaniea[†] kami@cmu.edu

[†]Carnegie Mellon University, Pittsburgh, PA, USA
[‡]University of North Carolina, Chapel Hill, NC, USA

ABSTRACT

Significant effort has been invested in developing expressive and flexible access-control languages and systems. However, little has been done to evaluate these systems in practical situations with real users, and few attempts have been made to discover and analyze the access-control policies that users actually want to implement. We report on a user study in which we derive the ideal access policies desired by a group of users for physical security in an office environment. We compare these ideal policies to the policies the users actually implemented with keys and with a smartphone-based distributed access-control system. We develop a methodology that allows us to show quantitatively that the smartphone system allowed our users to implement their ideal policies more accurately and securely than they could with keys, and we describe where each system fell short.

Author Keywords

Access control, policy creation, smartphones, discretionary access control, distributed access control.

ACM Classification Keywords

D.4.6 Security and protection, H.1.2 User/Machine systems, H.5.2 User Interfaces, H.5.3 Group and Organization Interfaces, K.4.3 Organizational Impacts, K.6.5 Authentication

INTRODUCTION

Access-control systems are used to permit or deny use of physical or electronic resources (e.g., office doors, file cabinets, or computer systems). Access-control systems can support different kinds of security policies depending on the characteristics of their design. For an access-control system to be effective, the policies it supports must match those that its users want or require. Thus, to thoroughly evaluate an access-control system, it is necessary to have real-world data about both users' "ideal" policies and those they actually implement with the system. Unfortunately, real-world policy data is hard to obtain. Even when the logistical challenges of collecting data can be met, people and organizations are reluctant to share sensitive data about their security policies and practices. Thus, designers have created a wide variety of access-control mechanisms, policy languages, and systems, but often have little understanding of which are really effective in practice. Moreover, it is unclear whether some features of these languages and systems contribute to or undermine the effectiveness and security of a system. This difficulty becomes especially acute as new technology enables the development of access-control systems that allow greater flexibility and have more features than their legacy counterparts, but at the cost of increased complexity and user involvement.

In this paper we describe the evaluation of one such system, targeted at access control for physical space in an office environment. Our evaluation focuses on the impact of the following functionality on the effectiveness and security of the system: the ability to delegate access between users; and the ability to delegate access on demand, from any location and at any time, up to and including the moment an access is attempted.

Specifically, we studied over the course of 11 months the deployment of Grey [4], a smartphone-based system used by 29 users to control access to locked physical spaces in an office environment. We have collected comprehensive usage logs and approximately 30 hours of interview data on users' ideal policies and those implemented with physical keys and with Grey. These three sets of policy data enable us to evaluate Grey policies both in absolute terms and relative to key policies, and we are able to determine which features of Grey are actually useful and used in practice.

Our results show that Grey policies are significantly closer to users' ideal policies than are key policies. Also, despite its potentially greater permissiveness, use of Grey resulted in fewer accesses being allowed overall. In our data, Grey policies never erroneously allowed access, and erroneously denied access rarely. Key policies, under the most generous assumptions about how securely keys are handled in practice, erroneously allowed access in a moderate number of cases and erroneously denied access in three times as many cases as Grey did. We find that Grey policies are closer to ideal policies for multiple reasons. First, Grey policies can be created and distributed at the moment they are needed, while keys must be distributed in advance. Second, Grey supports logging accesses, which is a common requirement in users' ideal policies, while keys do not.

We chose physical keys as a basis for comparison with Grey because we wanted to compare Grey with another deployed discretionary access-control system where users had the ability to manage access to their own resources. Keys were a natural choice for this comparison as the Grey-enabled doors used in our study were already equipped with traditional key locks. While our study focuses on two specific access-control technologies, we believe that our methodologies can provide guidance on how other solutions might be evaluated against one another, and that our results suggest factors that are important when developing other access-control technologies in order to meet the needs of users. More specifically, our three primary contributions are as follows:

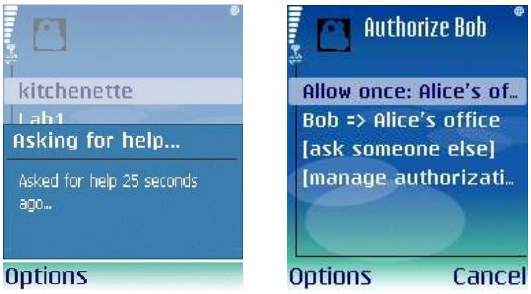
1. We document a collection of ideal policy data, which shows what conditions users want to place on access to their physical resources;
2. We develop a metric and methodology for quantitatively comparing the accuracy of implemented policies; and
3. We present a case study in which a smartphone-based discretionary access-control system outperforms keys in overall security and effectiveness of implementing users' desired policies, and identify the features that account for these improvements.

GREY

Grey [4] is a distributed access-control system that uses off-the-shelf smartphones to allow users to access and manage resources. Unlike a system where all access-control policy is managed by an administrator, Grey enables each end user to delegate her authority to others, at her discretion.

To unlock an office door, a Grey user causes her phone to contact the door via Bluetooth and send it a set of credentials and a formal proof that the credentials imply that this access should be granted. Each credential is a digitally signed certificate that includes a statement, in formal logic, of the authority it represents, itself a form of policy. The statement of policy that must be proved is conveyed by the door to the phone at the beginning of each access, and includes a random number, or nonce, to prevent replay attacks.

Our focus here is not on the underlying technology, but rather on the policy-creation and resource-access modes that Grey enables (modes that could also be supported by systems with entirely different technical underpinnings). Most importantly, by allowing users to create and modify policies via their smartphones, Grey enables policy to be created at the time and place of the users' choosing. Policies can be created proactively through a wizard interface or by associating access rights with entries in a Grey address book, or reactively in response to an attempted access that cannot succeed unless an authorized user extends her policy to allow the access. The policies that users can create include *delegating* to another user (1) one-time access, (2) all authority the grantor possesses, and (3) the authority to access a door for a period of time. For each credential, the user who creates it can specify the time interval during which the credential will be valid. A user can also create a group of doors (e.g., “lab doors”) or users (e.g., “my students”), and grant access to the entire group in one step.



Asking for help...

Asked for help 25 seconds ago...

Figure 1. Screen shots showing (a) Bob's phone asking Alice for help, (b) Alice being prompted to reactively allow access.

An example illustrates how this functionality is used in practice. Alice, a professor, is Bob's advisor. While Alice is travelling, Bob attempts to access her office to borrow a book. Since it doesn't yet contain sufficient credentials to let him into Alice's office, Bob's phone suggests that contacting Alice might help. Bob allows this, causing Alice's phone to inform her of Bob's access attempt and prompt her to modify her policy to allow the access. Alice instructs her phone to create a one-time delegation that will allow Bob access in this one instance. Her phone sends the appropriate credentials to Bob's phone, which is then able to open the door. This is an example of *reactive* policy creation, as Alice modified her policy in response to an access attempt. Later, realizing that Bob and her other students will need to borrow books again, Alice *proactively* creates credentials making each of her students a member of a new group that she calls “Alice's students.” She also creates credentials that allow any member of this group to access her office and lab, and instructs her phone to automatically distribute these credentials to her students so that they are available for use when next needed. Figure 1 contains screen shots of some of the interfaces that Alice and Bob use to perform these tasks.

Our goal for the present study was to evaluate the policies that users create with Grey regardless of whether they are created reactively or proactively. The details of the Grey user interface are relevant to this study to the extent that they support or hamper users in the creation of their desired policies. In a previous study we discuss usability issues with some of the Grey user interfaces [3]. We expect to address these issues in future work.

METHODOLOGY

We lay the groundwork for our results by describing the context in which our study took place and developing a methodology for comparing implemented policies to ideal policies.

Environment

We conducted our study in an office building on our university campus in which Grey has been deployed. Each of over three dozen doors was outfitted with an electric strike connected to an embedded computer that could unlock the strike. Each embedded computer interacts with Grey-enabled phones via Bluetooth. The second floor of the building includes a large workspace that is locked after 6 P.M. and on weekends. Inside the perimeter is a common area with cubicles for students and staff. Along the perimeter of the common area are offices, labs and storage rooms used primarily by faculty and staff. We studied the policies associated with nine of the Grey-enabled resources: eight offices inside the common area and a large machine room on the first floor.

Users

In January 2006 we began distributing Nokia N70 smartphones with Grey software installed to faculty, staff, and students who had a desk in the office building or had a regular need to access Grey-protected areas. We tried to select users who were in working groups with other participants. At the time of our evaluation all 29 study participants had been using Grey for at least three months and all participants with offices had been using it for at least eight months.

The 29 users who participated in the study included 9 computer science and engineering faculty members, 11 computer science and engineering graduate students, 7 technical staff members, and 2 administrative staff members; 24 were male and 5 were female. Most of our study participants were highly technical users. While users' technical abilities may have impacted their abilities to specify sophisticated access control policies, in a previous study we found that even highly technical Grey users had trouble learning to use the more complicated and less user-friendly Grey features [3]. Nonetheless, it would be useful to study less technically-savvy users in future studies. To preserve privacy we refer to study participants by fictitious names.

Each user could be classified as a *resource owner*, i.e., the primary owner and policy creator for an office or lab; a *resource user* who accesses resources controlled by others; or both. Our study included 8 resource owners and 28 resource users. All but one of the resource owners were also resource users. The 10 participants who either helped develop or had an uncommonly deep understanding of Grey were counted only as resource users even if they owned resources—the policies they created were excluded from the study to avoid biasing the results.

Procedure

We collected data by interviewing users and by logging their use of Grey. We recorded approximately 30 hours of interviews and logged 19,750 Grey access attempts.

Initial interview. Before giving a Grey phone to a user, we interviewed her to explore how she managed her physical security in the office setting. We began each interview by asking about the different resources (doors, computers, storage closets) in the workspace, asking for each resource who else might have need to use it and how that person obtained access to the resource. For instance, an instructor was asked how she passed homeworks and tests to her teaching assistants. A student was asked if he ever needed access to his advisor's office and, if so, how he obtained it. We also asked participants to show us their key chains and asked what resource each key opened and how the key had been obtained.

Regular interviews. We interviewed each user again after one month, and then every four to eight weeks, depending on user availability and activity. In particular, we scheduled interviews shortly after users created delegations, to ensure that they would remember the reasons and context behind the delegation. The purpose of these interviews was to determine what access-control policies the user wanted to create, what policies the user was actually creating with keys and Grey, her reasons for creating the policies, and what Grey features the user used or did not use. We generated questions for these interviews using a set of basic questions and follow-up questions to prompt users for further explanation, and questions that inquired about data in the Grey logs; for example, we might ask why a resource owner gave access to a specific user with Grey, or what conditions the owner would have liked to put on access when handing out a key.

Logs. Both the doors and the phones logged all Grey-related activity. Doors recorded every time they were contacted by a phone, whether or not the attempted access succeeded, and what credentials were used as part of a successful access. Phones logged all interaction with the user, including how users traversed menus and what methods were used for making delegations and accessing doors. Events such as delegations or multiple failed access attempts were flagged and used to plan interviews.

Analysis

To evaluate the accuracy of the policies implemented by physical keys and Grey, we compared those policies with resource owners' ideal policies—the policies they would implement if not constrained by the limitations of a particular access-control system. Eliciting ideal policies from users was difficult, as users were aware of the limitations of the deployed access-control systems when they discussed their policies. However, our interviews with resource owners were designed to provide information on who they *wanted* to provide access to and under what conditions, allowing us to construct representations of ideal policies. We also asked resource owners to whom and under what conditions they *actually* provided access to create representations of physical key policies. We created a representation of each resource owner's Grey policy from Grey log data combined with interview data. Each policy representation consisted of a set of *access rules*, each specifying a user, a resource, and a condition that had to be met in order for the user to be able to access the resource. We refer to the collection of all access rules for a single resource as an *access-control policy.* Each of our 9 resources has a corresponding access-control policy with 27 rules—one for each resource user, excluding the resource owner—for a total of 244 access rules. In this study we examined only the policies created by resource owners, not those that might be created subsequently by the resource users to whom the resource owners delegate access.

Due to the characteristics and limitations of keys and Grey as access-control mechanisms, the conditions under which accesses were allowed differed between the ideal, key, and Grey policies. We used the conditions to determine how well the implemented policies matched the ideal policies, measuring false accepts and false rejects associated with each of the implemented policies.

Ideal Access Conditions
I1. True (can access anytime)
I2. Logged
I3. Owner notified
I4. Owner gives real-time approval
I5. Owner gives real-time approval and witness present
I6. Trusted person gives real-time approval and is present
I7. False (no access)

Physical Key Access Conditions
K1. True (has a key)
K2. Ask trusted person with key access
K3. Know location of hidden key
K4. Ask owner who contacts witness
K5. False (no access)

Grey Access Conditions
G1. True (has Grey access)
G2. Ask trusted person with Grey access
G3. Ask owner via Grey
G4. Ask owner who contacts witness
G5. False (no access)

Figure 2. Conditions for access rules in ideal policies, as well as in actual policies implemented with physical keys or Grey.

IDEAL POLICIES

We used the interview data to identify a set of seven conditions that resource owners required in their ideal access rules, as shown in Figure 2. The first condition for ideal access (I1) allows the user to access the resource directly with no limitations. More stringent conditions require that the access be logged (I2) or that the owner be notified (I3). Three other ideal access conditions require someone else to let the user in. In interviews this other person was always the owner (I4)—who might also require a witness to the access (I5)—or a trusted person (I6). The most stringent condition is permitting no access (I7). In this section we describe these conditions and discuss some scenarios that gave rise to each condition.

True (access always allowed) (I1). In 19 access rules, the condition under which access was granted was trivially (always) true, and the users listed in the access rules had unconstrained access to the resource. Of the 8 resource owners, 5 created at least one rule with this condition.

Logged (I2). Access logging was required by 10 access rules. The intention was to allow access at any time, but only if a record of the access was made available to the resource owner. Two resource owners made use of this condition.

Eric's policy specified logging for all of his students and his secretary. He viewed logging as very important because it would force his students to be accountable for what they do. He refused to give his students any access to his office unless it was logged, even though giving access would have been mutually beneficial.

Owner notified (I3). In 3 access rules, a notification message was required to be be sent to the resource owner for the access to be granted. This message could be an email or a text message but had to be sent immediately after the access.

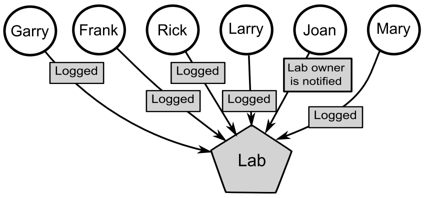


Figure 3. A representative portion of Mark's ideal policy. This policy consists of resource users (circles), resources (pentagons), access rules (arrows), and conditions (boxes on lines).

Mark explained that only a few of his students normally had reason to enter the lab. However, if there was an emergency or if one of the servers stopped functioning in the middle of the night, then he wanted any of his students to be able to get in to fix the problem. He trusted his students to make good decisions about what constitutes such an emergency but wanted to be notified of the access. A representative portion of Mark's ideal policy is illustrated in Figure 3.

Owner gives real-time approval (I4). Resource owners were required to approve accesses on a case-by-case basis in 4 access rules. This condition was used in scenarios in which the resource owner wished in general to deny access, except under circumstances that he deemed exceptional at the time they arose.

All 4 access rules that required this condition were part of Pat's policy about his office. Pat did not have any shared resources in his office and consequently saw no reason to give anyone access. However, on the occasions when packages with expensive equipment arrived for him when he was not there, Pat was willing to give access to one of his students so the student could put the packages in his office.

Owner gives real-time approval and witness present (I5). The resource owner was required to give approval on a case-by-case basis and a third party witness the access for 7 of the access rules. This condition was typically used when it was difficult to envision how a technological solution could enforce the desired policy.

Ryan told us how he once had to deal with a teaching assistant (TA) with whom he did not get along. When the TA needed to get into Ryan's office to retrieve tests or other papers, Ryan would have his secretary let the TA in and remain present to ensure that the TA did nothing else.

Trusted person gives real-time approval and is present (I6). A trusted person was allowed to make access-control decisions on the resource owner's behalf in only 1 access rule. The rule also required that the third party witness the actual access.

The one use of this condition in our study was by Lisa, who required her secretary, Emma, to ask Paul whenever she needed access to Lisa's office. Emma rarely needed to get into Lisa's office so asking Paul each time was not too inconvenient, and Lisa felt that having Paul witness each access made Emma more accountable.

Although this condition appeared only once among our users, it was commonly used when referring to users outside of our study. One example involved Mark's lab, which is maintained by several staff members. According to Mark's policy, he trusts these staff members to allow others access to the lab provided they are present when the access occurs.

False (access always denied) (I7). Users' ideal policies contained 200 rules that unconditionally denied access.

Additional conditions. The conditions described above were the only ones observed among our study participants. However, there were additional conditions used in reference to users and places outside the scope of this study. Specifically, resource owners desired to restrict access to a specific time interval each day, or to allow access only if the owner is not present or when the door is not expressly locked.

POLICIES IMPLEMENTED WITH KEYS AND GREY

In this section we describe our methodology for measuring false accepts and false rejects, and detail the false accept and false reject rates associated with the implemented policies.

The conditions for keys and Grey follow a similar pattern as the ideal conditions, as shown in Figure 2. Conditions K1 and G1 allow the user access to the resource with no conditions, and conditions K2 and G2 allow access via a trusted person. Condition K3 is unique in requiring knowledge of a secret in order to gain access, specifically of the location of a hidden key for the resource. Conditions K4, G3, and G4 require the user to ask the owner (and that a witness be present in conditions K4 and G4). In the case of Grey, conditions requiring the involvement of another person do not necessarily imply that person must be physically present where the access occurs, though in the case of physical keys they do. Again, the most stringent conditions are those prohibiting access (K5, G5).

After determining these sets of conditions, we compared individual access rules based on which conditions imply others. Specifically, the notation $A \Rightarrow B$ should be read as: the condition A is at least as stringent as condition B, and so if condition A is met, then so is B. Note that False (no access, the most stringent condition possible) implies any condition, and any condition implies True (unconditional access, the trivially satisfied condition). In addition, we made several assumptions involving implications between the conditions:

- **Ask owner via Grey (G3) ⇒ Logged (I2)**: Asking the owner using Grey sufficiently logs the access.
- **Ask owner via Grey (G3) ⇒ Owner notified (I3)**: Asking the owner notifies her of the access.
- **Ask owner who contacts witness (K4, G4) ⇒ Logged (I2)**: Asking the owner sufficiently logs the access in both key and Grey implementations.
- **Ask owner who contacts witness (K4, G4) ⇒ Owner notified (I3)**: Asking the owner notifies her of the access for both key and Grey implementations.

Aside from these, we made no assumptions about relationships between conditions.

Using these assumptions, we counted false accepts and false rejects in the implemented policies, where

- A *false accept* is a user/resource pair for which *Implemented* $\not\Rightarrow$ *Ideal*, i.e., for which the condition in the implemented access rule is not at least as stringent as the condition in the ideal access rule, and hence could result in accesses being allowed, or accepted, without the ideal policy being satisfied.
- A *false reject* is a user/resource pair for which *Ideal* $\not\Rightarrow$ *Implemented*, i.e., for which the ideal policy is not at least as stringent as the implemented policy, and may permit accesses that the implemented policy denies (rejects).

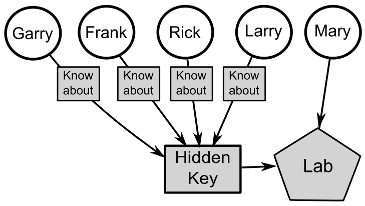


Figure 4. Mark's physical key policy. Four resource users were given access to the lab via a hidden key. One user (dotted line) does not have access under the optimistic assumption about hidden keys, but does under the moderate assumption (because she has access to the hidden key even though Mark did not intend to give her access to the lab). The last user, Mary, has direct access.

An access rule could conceivably be counted as both a false accept and a false reject, if *Implemented* $\not\Rightarrow$ *Ideal* and *Ideal* $\not\Rightarrow$ *Implemented*. For example, if the two conditions are *Owner notified* (I3) and *Ask trusted person with key access* (K2), then an access might be granted without the owner being notified (false accept) and an access may be refused if no trusted person is available, even if the owner had been notified (false reject). However, we encountered no such situations in this study.

We emphasize that false accepts and false rejects describe user/resource policies, and not particular occurrences of that user attempting to access that resource. In this way, our measures of false accepts and false rejects are independent of the frequency with which any particular policy is applied in the system.

Physical Key Policies

The false accept and false reject rate in key policies were heavily influenced by the common practice of hiding keys—a resource owner would hide a key or set of keys but make its location known only to a select group of users. Although we were surprised at the prevalence of hidden keys in an office space, hiding keys (most often car and home keys) is such a common practice that the online auction site eBay has a special category for key-hiding devices. Key hiders are also available for offices, though these tend to be in the form of combination lock boxes.[1]

Hidden keys solve some of the problems associated with distributing keys, but can easily be discovered by unauthorized users, e.g., by observing others retrieving the keys or by coming across the keys serendipitously. Thus, we counted false accepts and false rejects in key policies using three different assumptions about hidden keys that approximate how many people know about them. The importance of using different assumptions about hidden keys is illustrated by

[1] For example, http://www.nokey.com/comlocbox.html.

Hidden keys assumption	*False accepts*	*False rejects*
Optimistic	7	12
Moderate	64	8
Pessimistic	169	3

Figure 5. Counts of false accepts and rejects for key policies under three assumptions about knowledge of the location of hidden keys.

Mark's key policy, shown in Figure 4.

In our three assumptions we explicitly distinguish between a user's ability to *enter* a room and the key policy *giving her access* to the room. In all cases we assume that if one room encloses another then entering the second room requires the ability to enter the first. Furthermore, a user can enter a room if the key policy allows the user to access it and the room is not enclosed by another. In our environment, all users who were given access to an enclosed room were also given access to the enclosing room. We stress, however, that entering a room need not imply having access to that room in the key policy, depending on which of our three assumptions we make about uses of hidden keys:

Optimistic assumption: Users will respect the key policy. More precisely: If a user can enter room X, and room X contains a set of hidden keys with a key to room Y, and the key policy allows her access to room Y, and she can enter Y's enclosing room, then she can enter room Y.

Moderate assumption: A user can use any hidden key located in a space to which she has access by the key policy. More precisely: If a user can enter room X, and the key policy gives her access to X, and X contains a set of hidden keys with a key to room Y, and she can enter Y's enclosure, then she can enter Y.

Pessimistic assumption: Users will use any hidden key they can find. More precisely: If a user can enter room X, and X contains a set of hidden keys with a key to room Y, and she can enter Y's enclosure, then she can enter Y.

We learned from interviews that, in fact, many unauthorized users knew the locations of hidden keys. Although all users didn't know about all hidden keys in the areas to which they had access, as our moderate assumption states, many also knew of hidden keys that could be used to access even more hidden keys. Hence, we believe our moderate assumption is a conservative approximation of the real knowledge of the users in our study. Figure 5 shows the false accept and false reject counts for key policies under each assumption.

We observed five causes for discrepancies between ideal policies and key policies:

1. Hidden keys were available to unauthorized users.
2. Logging (I2) was not supported.
3. Notification (I3) was not supported.
4. Approval upon request (I4) when the owner is not physically present at the resource was not possible.
5. Key distribution was inconvenient.

Figure 6 shows counts of false accepts and false rejects by the five causes under the moderate assumption about knowledge of hidden keys. We discuss each of the causes below.

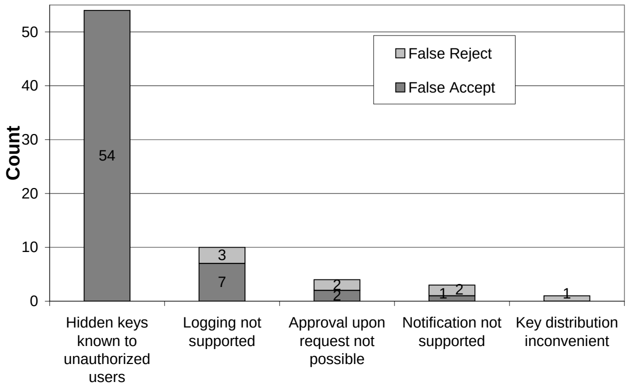


Figure 6. Counts of key policies' false accepts and rejects by cause, under the moderate assumption about knowledge of hidden keys.

Hidden keys. Depending on the assumptions about how many unauthorized users were able to learn the location of hidden keys, there were 0 to 160 false accepts due to hidden keys. Under our moderate assumption, which we believe to be the most realistic, there were 54 false accepts due to hidden keys. (Of the false rejects under the optimistic assumption, 4 become false accepts or true accepts as we assume more widespread knowledge of hidden keys. The 54 false accepts under the moderate assumption include only those access rules that were true rejects under the optimistic assumption, but become false accepts under the moderate.)

Owners use hidden keys to address the inconveniences of key distribution. It is much easier to convey the location of a hidden key to a person than to make a new key and give it to that person. Thus, when an owner's ideal policy calls for multiple users to have access to a resource, or when an owner frequently allows access to new users (as in a university, where new students arrive every year), a hidden key is often used to simplify key distribution.

However, hidden keys introduce a number of problems. They are hard to revoke from any subset of users, changing the hiding place revokes everyone's access to the key, and the new hiding place needs to be disseminated to those still allowed access to the key. Furthermore, it is easy for unauthorized users to learn the location of and gain access to hidden keys. In our results, we have counted false accepts pertaining only to users participating in our study; the number of false accepts due to hidden keys would be much higher if we counted non-participants in the office building who could access the hidden keys. Finally, hidden keys can be lost or stolen, thereby not only revoking access to authorized user, but also raising the possibility that the resource has been irrevocably compromised (at least until the lock is changed).

Logging not supported. There were 10 cases in which a resource owner desired to allow access if it was logged (I2). In 7 of these cases, access was granted without the logging condition being fulfilled, thus leading to false accepts; in 3 cases no access was granted, leading to false rejects.

Eric's ideal policy gives his students access to his office, but Eric chose not to give them access using keys. Instead, if one of them needed access, he would contact his secretary,

Deferred delegation assumption	*False accepts*	*False rejects*
Deferred delegations counted as false rejects	0	13
Deferred delegations counted as given	0	3

Figure 7. Counts of false accepts and false rejects for Grey policies under two assumptions about what constitutes the Grey policy.

who would let the student in (K2). He explained that he was only willing to give his students access to his office if they knew they could be held accountable for their actions. Mark also wanted all accesses to his lab logged, but for him it was more important that his staff and students gain access than it was that they be logged. Thus, Mark distributed keys, even though the logging condition would not be satisfied.

Notification not supported. In 3 cases resource owners desired to allow access if they were notified (I3). Since keys do not support notification, this led to discrepancies with the ideal policy. In 2 cases, no key access was granted, leading to false rejects; in 1 case, key access was granted via a hidden key, leading to a false accept. Under the pessimistic assumption about hidden keys, the false rejects became false accepts because the relevant users could gain unauthorized access to a hidden key.

Approval upon request not possible. There were 4 cases in which an owner would have granted access upon request (I4), but was not willing to distribute keys. Since the owner presumably would not be present for some requests and keys cannot be shared at a distance, we counted these as false rejects unless one of the relevant users had unauthorized knowledge of a hidden key. Thus, under the moderate assumption, 2 of these false rejects became false accepts (because the user was allowed access without fulfilling the desired condition), and under the pessimistic assumption, all 4 of these false rejects became false accepts.

All 4 false rejects were for Pat's office, to which only he had access. When asked if there was any reason that someone else would need access, he explained:

> I have a copy of our passwords for the, uh, the lab, so if there was an emergency . . . it is possible that someone might want to come in and look at the root password archive or something like that. It's pretty rare, but it's possible, though. And you can imagine if that happened after hours, they would wanta, we gotta get in and get that. So they might call me.

Key distribution inconvenient. Distributing a key entails the overhead of making the key, physically giving it to someone, and keeping track of who has it. The inconvenience of distributing keys led to a case in which the owner's ideal policy called for access to be allowed, but the owner did not grant key access. Emma told us that getting keys for her work-study students took nearly two months. In the meantime, she left her office door unlocked and provided them with hidden keys to other resources (K3) so that they could do their jobs.

Keeping track of who has what key can be problematic since keys can be given away or lost. In an initial interview, Brian told us that he had given away one of his keys to a friend who he thought needed it more. During this study three users permanently gave their keys to another person; two couldn't even remember to whom the key had been given.

Grey Policies

Grey policies matched ideal policies quite closely. As Figure 7 shows, we observed no false accepts in the policies. False reject counts depended on an assumption about what constituted the Grey policy. Because Grey allows for granting access at the time it is needed, some owners deferred creating policies for certain user/resource pairs until it was requested. Under the conservative assumption that deferred delegations do not count as implemented policy, we observed 13 false rejects in Grey policies. Under the more liberal assumption that deferred delegations were implemented policy, we observed only 3 false rejects, all because Grey does not support notification. In the remainder of this section, we discuss, for each of the seven ideal policy conditions, how users implemented their policies with Grey.

True (can access anytime) (I1). Users could trivially implement anytime access by issuing a credential to the relevant user (G1). Under the assumption that deferred delegations count as implemented policy, we did not observe any cases in which access was desired but not issued. Under the assumption that deferred delegations do not count as implemented policy, we observed 10 false rejects, i.e., in 10 access rules in which the ideal policy called for anytime access, the owner deferred delegating that access.

Fred explained that his advisor initially granted him a temporary delegation to the advisor's office before deciding to give Fred a longer-term delegation:

> We have, like, a weekly meeting, and normally he is running late, so one week he just let me in for, like, a one-time authorization. Then after that, like, OK, it's obvious that I'm going to be going in there frequently so he just gave me authorization to [his office].

Logged (I2). Grey-enabled doors log all accesses, and so logging was implemented in the 10 cases where ideal policy called for accesses to be logged.

Mark gave 6 other participants access to his lab. He wanted all accesses to be logged so that if anything went wrong he could hold the appropriate person responsible. During the course of our study, Mark asked us 4 times for the logs for the lab. When asked about it, he said that they were just small things he wanted to check up on. For example, one time a piece of equipment had moved and no one knew who had moved it. The policy Mark implemented in Grey is illustrated in Figure 8.

Owner notified (I3). Grey does not support notification. In 3 cases in which the ideal policy called for notification, owners stated that they were willing to grant access to the relevant users if contacted at the time of the request (G3). This was a compromise, because the ideal policy merely required that the owner be notified of the access, but not that the owner be required to intervene to grant access. Thus, these 3 cases counted as false rejects.

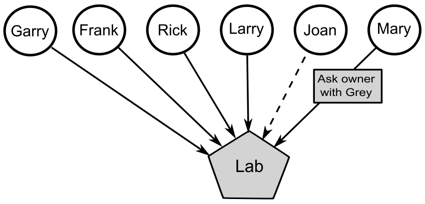


Figure 8. A representative example of Mark's Grey policy. Four people have a Grey delegation to the lab and Joan will be given a delegation when she asks (dotted line). Mary will be given a temporary delegation on demand.

Owner gives real-time approval (I4). Grey supports approval upon request (G3), so when the ideal policy called for the owner to give real-time approval, this condition was implemented easily in Grey. We observed 4 cases in which approval upon request was required by the ideal policy and implemented accurately in Grey.

Pat did not give anyone access to his office using Grey, but he told us about a time when it was extremely useful to remotely grant temporary access to one of his staff:

> I was at the airport at 5:30 in the morning, and we actually had some stuff being delivered at 5:30 in the morning, and Ethan was here, and I wasn't going to get here 'til six. And true to their word, they were here at 5:30 and he needed to get a key out of my room. The request came in and I said, "You are in there one time buddy," and that is all he needed.

Owner gives real-time approval and witness present (I5). In 7 cases where the ideal policy called for the owner to give approval upon request and for a witness to be present, owners implemented a Grey policy that required users to contact the owners, who would then contact a trusted person and ask them to serve as a witness (G4). It was possible to fulfill these conditions using Grey because owners could choose any trusted person at the time of requested access and (if they had not done so already) delegate to the trusted person access to the relevant resource.

When we asked Donald if he would be willing to use Grey to remotely let someone into his office, he replied that he would rather call a trusted witness to let the person in. Even if the person only wanted something simple like printer supplies, Donald felt more comfortable if someone trusted was there to make sure that printer supplies was what was taken.

Trusted person gives real-time approval and is present (I6). In 1 case in which the ideal policy called for a trusted person (but not necessarily the owner) to approve a request and serve as a witness, the owner issued a Grey certificate to a trusted person, who could then give access to the relevant user upon request (G2). This happened between Lisa, her secretary Emma, and Paul, and it closely mirrors the corresponding ideal policy and key implementation.

A more interesting case involved Eric's office, but was not counted in the results we report because the relevant users did not all have Grey phones. Eric gives out access to his office only using Grey because Grey supports logging, which Eric considers to be vital. Consequently, he has given access to only those of his students who have Grey phones. If any of his other students need access, Eric expects that they will ask a student who has access to let them in.

False (no access) (I7). In 200 cases in which access was not allowed in the ideal policy, it was not granted through Grey.

DISCUSSION

Our data shows what conditions users want to set in their ideal access control policies and highlights those aspects of a flexible access-control system like Grey that allow users to implement policies that match their ideal policies. We found that the conditions users desire to place on access to their resources fall into the seven categories discussed in the Ideal Policies section. This list of seven conditions may not be complete, but it at least serves as a minimal set of the conditions users are likely to want to set in their policies.

Notably, logging (I2), notification (I3), and real-time approval upon request (I4) were desired conditions that are not supported by keys, but can easily be supported by a digital access-control system such as Grey. (Grey does not currently support notification, but could be extended to do so, e.g., using SMS messages.) Two other desired conditions, approval by a trusted person (I6) and presence of a trusted witness (I5), require functionality to delegate authority to a trusted person to make access-control decisions on the owner's behalf or to delegate the authority to serve as a witness. Policies that include these conditions can be approximated by keys (K2, K4), but can be made more practical and implemented more accurately by a system that does not require the exchange of physical tokens to delegate authority and that enforces the presence of a "witness" through technological means (e.g., by activating a camera).

Our data shows that Grey's support for the conditions users desired allowed users to implement policies with Grey that more closely matched their ideal policies than did the policies they implemented with keys. Figure 9 compares key and Grey policies according to each of the 244 access rules implemented in our study. Each pair of bars in the graph corresponds to one of the seven ideal policy conditions. The left bar in each pair shows key policy data, under our moderate assumption of who has knowledge of hidden keys, while the right bar in each pair shows Grey policy data, under our assumption that deferred Grey delegations do count as implemented access rules. Each bar is subdivided to indicate faithful implementations of the ideal condition, false rejects, and false accepts. Virtually all access rules were faithfully implemented in Grey; Grey policies yielded only 3 false rejects and no false accepts, while key policies yielded 8 false rejects and 64 false accepts.

To measure the "permissiveness" of policies in each system, we count the total number of rules that allowed access in each system. We initially thought that allowing easy delegation, as does Grey, might lead to excessively permissive policies. In fact, we found the opposite to be true: 19 of 244 Grey rules allowed access, while 25 of 244 key rules allowed access under our most optimistic hidden-keys assumption. Under our more realistic assumption about hidden keys, 68 key rules allowed access. Although Grey is the more flexible

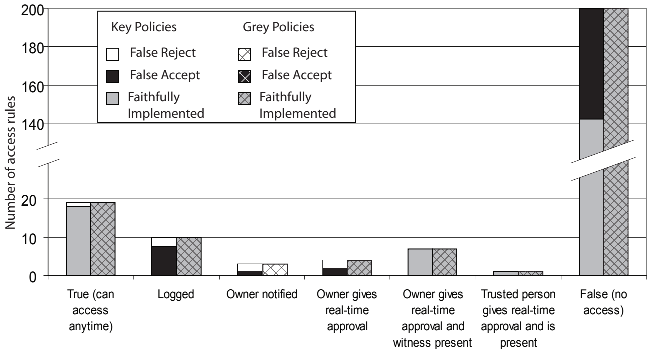


Figure 9. Counts of faithful implementations, false rejects, and false accepts by ideal policy condition, for both key policies under the moderate assumption (left bar in each pair) and Grey policies (right bar).

system, it led to more restrictive policies, because users did not need to circumvent the system in order to implement the policies they wanted.

The reasons for Grey's superior performance are clear from our ideal policy data. Users' ideal policies called for features Grey supports, particularly the ability to create policy from anywhere in real-time and upon request, and the ability to log accesses. Furthermore, Grey's easy policy creation mechanism makes it more convenient than keys, which are expensive to make, distribute, and keep track of.

Our study also revealed lessons that pertain to other features of Grey, as described below.

Transitive delegation. Users commonly desired to delegate control over their resources to others, such as administrative assistants. When an owner can grant access to a user, and that user can then grant that same access to others, we call this delegation *transitive*. We found transitivity to be a practical and highly desired property that was used in many of our users' access-control policies. Transitive delegation enables users to implement the ideal policy condition that requires delegation of authority to a trusted person. However, it is not always appropriate for delegations to be transitive. Some users, in situations that fell outside the scope of our study, wished they could delegate access to open a door without delegating the ability to pass on this access to others. It is important, then, that an access-control system support both transitive and non-transitive delegation.

Arbitrary grouping granularity. Grey allows users to dynamically form arbitrary groups of users and resources. This is in contrast to keys, which at best provide rigid sets of resources through master and submaster keys, and have no notion of groups of users. We assumed that users would benefit from the ability to form groups of users and resources, and to then assign privileges to groups rather than individuals. Our interview data supports this assumption, in that users do think in terms of groups, such as "PhD students," "delivery people," and "visitors." However, users in this study rarely created groups. Users may have found the phone-based user interface for creating groups too difficult to use. A desktop-based user interface we are currently developing will make group creation easier and may encourage users to create groups. In addition, this study only involved 29 users; Grey's group creation feature might be exercised in a larger-scale deployment, where the number of users might be too great to set policies for each user individually.

Reactive delegation. We found that Grey users relied upon reactive delegation, the ability to delegate access to resources when needed and upon request. Two of the ideal policy conditions required the ability to give real-time approval upon request, and it thus seems important for an access-control system to support reactive delegation. It could even be argued, as some privacy policy researchers have [10], that reactive delegation's counterpart, proactive delegation (sometimes referred to as "configuration"), is virtually unnecessary. However, our data contradicts this; users used both Grey's proactive and reactive delegation capabilities, and we conclude that access-control systems should provide both. Further study will perhaps shed more light on when users wish to use each kind of delegation.

It is important to note that this study focuses on the needs of resource owners, which are not necessarily aligned with the needs and interests of resource users, trusted witnesses, or other people who may interact with an access-control system. For example, logging may be objectionable to some resource users, and those called upon frequently to be trusted witnesses may find this role burdensome. Indeed, these features may not only impact the way users interact with the access-control system, but also affect social interactions in an organization.

RELATED WORK

The two access-control technologies that we examined, physical keys and Grey, make for an interesting comparison because they offer very different levels of flexibility. That said, numerous other access-control mechanisms could be considered in a study such as ours, e.g., proximity cards and swipe cards for physical resources, or passwords, RSA SecureID tokens,[2] and smart cards for electronic resources. We are unaware of any published studies of these technologies on the axes we consider here, in particular with attention to the accuracy of the policies that people implement with them. However, the limitations of these technologies (particularly, the lack of a user interface on the access token) would make it difficult to use them to implement the kinds of reactive policies that our users desired. In addition, several proposed distributed systems use portable devices to control access to physical spaces [6, 15]; however, as far as we know, none of these has been implemented.

An implemented access-control technology that supports dynamic delegation is file access control. Cao et al. showed that standard access-control list (ACL) interfaces had a high failure rate, despite users expressing confidence that they had manipulated the ACLs accurately [7]. Other studies showed that low levels of feedback in many access-control systems make it difficult for users to understand what is wrong with a policy and what needs to be changed [9, 13]. These studies look at how users build and manipulate access-control polices in a single session but don't consider if user's nees are met or how policies are managed and changed over time.

[2] `http://www.rsasecurity.com/node.asp?id=1156`

The security community has designed and formally discussed many access-control policy languages (e.g., [1, 2, 11, 12]), each supporting a different set of policies. However, we are unaware of published research on the ability of these languages to meet access-control needs in practice.

A few studies have surveyed needs for access-control systems from an organizational or end-user perspective. Ferraiolo et al. studied the needs of 28 organizations and identified seven access-control approaches, including *discretionary access control* (DAC), in which access is assigned to individuals and groups, who in turn may delegate that access to others. The authors note that DAC, which is usually implemented through ACLs, is well suited for organizations where end-users have rapidly changing information access needs, but that when the ACLs are centrally administered they "can become clumsy and difficult to maintain." They also note that DAC is not suitable for organizations concerned with maintaining tight controls on access rights [8]. Whalen et al. conducted an online survey on end-user experiences with sharing and access control. They found that users have dynamic access-control needs that vary with task and are often frustrated by current access-control mechanisms that are difficult to use and not well-suited to users' workflow [14].

Finally, related works have discussed the usability and social impacts of Grey [3] and its underlying algorithms [5].

CONCLUSION

The dearth of access-control policy information, either ideal or as implemented, is a barrier to development of advanced access-control technologies. In this paper we have detailed a real-world user study of access-control policies, both ideal ones and as implemented via two technologies, physical keys and the Grey system. We have developed a methodology for quantitatively evaluating these implemented access-control policies against the policies that users would ideally like to have, so that we can account for their false accepts (implemented policies allowing accesses that ideally would be prevented) and false rejects (implemented policies that reject accesses that would ideally be allowed).

The results of our study, aside from demonstrating the utility of our methodology, elucidate several reasons why Grey implemented users' ideal access-control policies more accurately than keys did. Among these reasons are that Grey supports access logging, and that delegations can be created and distributed when needed. The failure of physical keys to implement the latter is among the main reasons for the use of hidden keys, an instance of "security by obscurity" that breaks down as knowledge of the hidden key leaks. Our results also help us to prioritize further developments of Grey, e.g., to focus on those policies that users want but that we do not yet support. We hope that the results of our study can similarly aid others in developing access-control technologies to better support users' policy goals.

ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation Cyber Trust Grants CNS-0433540 and CNS-0627513, and U.S. Army Research Office contract no. DAAD19-02-1-0389 ("Perpetually Available and Secure Information Systems") to Carnegie Mellon University's CyLab.

REFERENCES

1. M. Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1–2):3–21, Oct. 1998.
2. A. W. Appel and E. W. Felten. Proof-carrying authentication. In *6th ACM Conference on Computer and Communications Security*, Nov. 1999.
3. L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. Lessons learned from the deployment of a smartphone-based access-control system. In *Symposium On Usable Privacy and Security*, July 2007.
4. L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar. Device-enabled authorization in the Grey system. In *Proceedings of the 8th Information Security Conference*, Sept. 2005.
5. L. Bauer, S. Garriss, and M. K. Reiter. Efficient proving for practical distributed access-control systems. In *Computer Security—ESORICS 2007: 12th European Symposium on Research in Computer Security*, Sept. 2007.
6. A. Beaufour and P. Bonnet. Personal servers as digital keys. In *2nd IEEE International Conference of Pervasive Computing and Communications*, Mar. 2004.
7. X. Cao and L. Iverson. Intentional access management: Making access control usable for end-users. In *Symposium On Usable Privacy and Security*, 2006.
8. D. F. Ferraiolo, D. M. Gilbert, and N. Lynch. An examination of federal and commercial access control policy needs. In *16th National Computer Security Conference*, pages 107–116, 1993.
9. A. Kapadia, G. Sampemane, and R. H. Campbell. KNOW Why your access was denied: regulating feedback for usable security. In *11th ACM Conference on Computer and Communications Security*, 2004.
10. S. Lederer, J. I. Hong, A. K. Dey, and J. A. Landay. Personal privacy through understanding and action: Five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454, 2004.
11. N. Li and J. C. Mitchell. Understanding SPKI/SDSI using first-order logic. *International Journal of Information Security*, 2004.
12. N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *IEEE Symposium on Security and Privacy*, May 2002.
13. R. A. Maxion and R. W. Reeder. Improving user-interface dependability through mitigation of human error. *International Journal of Human-Computer Studies*, 63(1-2), 2005.
14. T. Whalen, D. Smetters, and E. F. Churchill. User experiences with sharing and access control. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1517–1522, 2006.
15. F. Zhu, M. W. Mutka, and L. M. Ni. The master key: A private authentication approach for pervasive computing environments. In *4th IEEE International Conference on Pervasive Computing and Communications*, pages 212–221, 2006.