Comparing Access-Control Technologies:
A Study of Keys and Smartphones

Lujo Bauer Lorrie Cranor Robert W. Reeder Michael K. Reiter Kami Vaniea

February 28, 2007
CMU-CyLab-07-005

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Comparing Access-Control Technologies: A Study of Keys and Smartphones

Lujo Bauer Lorrie Cranor Robert W. Reeder Michael K. Reiter Kami Vaniea

Carnegie Mellon University
{lbauer, lorrie, reeder,reiter, kami}@cmu.edu

Abstract

Significant effort has been invested in developing expressive and flexible access-control languages and systems. However, little work has been done to evaluate these theoretically interesting 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. In this paper we report on a 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 Grey, a smartphone-based distributed access-control system. We show quantitatively that Grey allowed our users to implement their ideal policies more accurately and securely than they could with keys, and describe where each system fell short. As part of this evaluation we identify conditions that users commonly required in their desired policies and explain how these conditions can or cannot be implemented with keys and Grey. Our results and experience can serve to inform the designers of access-control systems about which features these systems should include if they are to successfully meet users' needs.

1 Introduction

Access-control systems can support different kinds of security policies depending on the characteristics of their design. An access-control system is effective if the policies it supports 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 come by. 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. As a result, designers of access-control systems are left to speculate about what functionality users need. 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.

We have developed Grey [3], an access-control system built on a relatively new paradigm of *distributed* access control. In the Grey system, smartphones can be used to access resources such as office doors. To access a resource, a smartphone sends a proof, which includes signed credentials from resource owners and other participants, demonstrating that the phone owner is allowed to access the resource. A computer associated with the resource allows access (e.g., by unlocking a door) if the proof is valid. Access-control policy is distributed in that no single device or computer involved in the system stores the entire policy governing a resource. Credentials spread across many devices form the policy as a whole. The great advantage of such a system is that it enables resource owners to delegate access to their resources without the intervention of a central authority. Policy can be set by any user in the system, and policy can be set proactively, i.e., before

any request for access, or reactively, i.e., in response to a request for access. Grey allows for logging all accesses to a resource.[1]

We have deployed Grey to 29 users in our building, 8 of whom control Grey-enabled office doors, and all of whom previously used physical keys to access floor resources. We are thus in the rare position of having a new access-control technology that has actually been deployed and is in use, and for which users have volunteered to make their access-control policies and usage data available to us with both the new technology and the old (physical keys). We have collected data on Grey users' ideal, physical key, and Grey policies. Ideal policies, elicited from interviews with Grey users over the course of several months, are the policies they would like to have in place independent of the restrictions of any particular access-control technology. Key policies are the policies our users implemented using physical keys, before or in parallel with their use of Grey. Grey policies are the policies our users actually implemented with Grey. These three sources of policy data enable us to determine how closely Grey policies implement ideal policies and how closely key policies implement ideal policies. We are thus able 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. 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. However, under more realistic assumptions, the erroneous accesses allowed in key policies shot up by more than a factor of four, while still erroneously denying access in three times as many cases as Grey did. In addition, we find that Grey policies are closer to ideal policies for multiple reasons. First, Grey delegations can be created and distributed easily and at the moment they are needed, while keys must be distributed before the moment of access, and are typically made available to all who might ever conceivably need them, as well as some who should not have them. Second, Grey supports logging accesses, which is a common requirement in users' ideal policies, while keys do not.

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 resources;
2. We detail a metric and methodology for comparing the accuracy of implemented policies; and
3. We show that an end-user-based access-control system outperforms keys in overall security and effectiveness of implementing users' desired policies, and identify the features of that system that account for these improvements.

2 Grey

Grey [3] 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 a centralized administrator, Grey enables each user to delegate her authority to others, at her discretion. In this way, access-control policy is managed by end users in a distributed fashion.

[1] Some access-control scenarios require that policy is specified centrally and cannot be extended by participants, which nullifies the advantages of using a system like Grey. Here, however, we focus on a decentralized system in which individual owners can delegate access to their resources.

Figure 1: Screenshots showing (a) Bob's phone asking Alice for help, (b) Alice being prompted to reactively allow access, and (c) Alice proactively extending her policy.

To access a resource (e.g., an office door or computer login), a Grey user causes her phone to contact the resource via Bluetooth and send it a set of credentials and a 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; the proof of access is likewise in formal logic. The statement of policy that must be proved is conveyed by the resource to the phone at the beginning of each access attempt, and includes a nonce to prevent replay attacks.

Our focus here, however, is not on the technologies that underlie Grey, but rather on the modes of creating policy and accessing resources that it enables (modes that could conceivably 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, either 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 a principal with authority over the resource extends her policy to allow the access. The policies that users can create include granting another user (1) one-time access to a resource, (2) all authority the grantor possesses, and (3) the authority to access a resource for a period of time. Groups and roles are also supported. For example, a user can aggregate several resources under one name (e.g., "lab doors"), and then in one stroke delegate access to all of them. Similarly, users can create groups that include several principals (e.g., "my students"), making it possible to give all members access to a resource in one step. For each credential, the user who creates it can specify the time interval during which the credential will be valid.

A simple 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 textbook. 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 this 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 for him. 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 undoubtedly 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". Furthermore, she creates credentials that allow any member of this group to access her office and her lab. When the credentials are created, Alice instructs her phone to automatically distribute them among her students, so that the credentials are available for use when they are next needed. Figure 1 contains screenshots of some of the interfaces that Alice and Bob would have used in

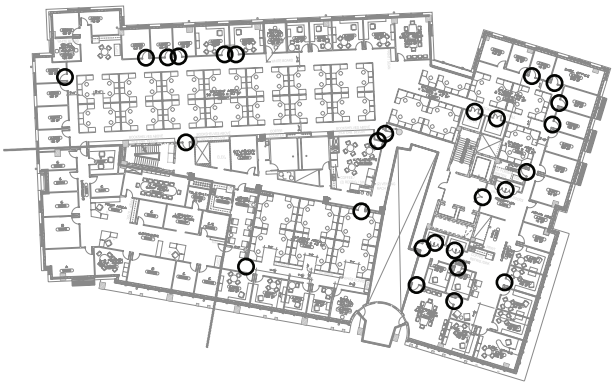

Figure 2: Floor plan of the 2nd floor of the office building, with Grey-enabled doors circled. Eight of the circled doors (we do not identify them specifically to protect the identities of study participants) and three Grey-enabled doors on another floor were used in the study.

performing these tasks.

3 Methodology

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

3.1 Environment

The study was conducted in a university office building. 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 was Bluetooth-equipped, so that Grey-enabled phones could interact with it. The Grey phones and doors were set up to log all Grey-related activity.

The second floor of the building (shown in Figure 2) includes a large work-space that is locked after 6pm 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.

Eight offices inside the common space were Grey-enabled. On the first floor (not shown in Figure 2) was a large machine room which was also Grey-enabled. The occupants of all eight offices were Grey users and the machine room was accessed primarily by Grey users. These are the nine resources that are used in this study.

3.2 Users

In January 2006 we began distributing Nokia N70 smartphones with Grey software installed. The users who received Grey phones were selected from faculty, staff, and students who either 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 Grey users to maximize the usefulness of Grey. We initially handed out smartphones to only a few users. As the system became more stable and usable we increased the number of users incrementally. At the time of this writing 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 six months.

The 29 Grey users participating in the study include nine computer science and engineering faculty members, eleven computer science and engineering graduate students, seven technical staff members and two administrative staff members. Twenty-four are male and five are female. To preserve privacy we refer to Grey users by fictitious names in this paper.

Each Grey 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 eight resource owners and 28 resource users. All but one of the resource owners were also resource users; the one exception was a user who was not able to carry a phone and therefore could not use it to access resources, although he was able to create policies via a proxy for the resource he controlled. The ten 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.

3.3 Procedure

We collected data for our study by interviewing users and by logging their use of Grey.

Initial interview Before giving a Grey phone to a user we conducted an initial interview that explored how the user managed her physical security in the office setting. We began each initial interview by asking about the different resources (doors, computers, storage closets) in the work-space, asking for each resource who else might have need to use it and how that person obtained access to the resource. We took advantage of the work structure of the university and asked about other relevant people who might need access to resources. For instance, an instructor might be asked how she passed homeworks and tests to her teaching assistants. A student might be 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.

Each user was then given a Grey phone and basic instruction on how to use it, including how to open a door and how to request access from another person. We also informed each user that if she became too frustrated at any time or if Grey failed to work it was acceptable to unlock a Grey-enabled door with a key.

Regular interviews After one month each user was interviewed again with the goal of understanding her initial use of Grey. This interview explored the user's use or lack of use of Grey's features as well as problems she encountered. We also asked how and why each user made use of Grey's delegation capabilities.

For the remainder of the study we interviewed each user every 4 to 8 weeks, depending on user availability and activity. Since access-control policy modification is a task that occurs rarely we used log data to schedule interviews shortly after users participated in any type of delegation activity. During these interviews we asked about changes to their access-control policy and the reasons for them. If users failed to re-implement a part of their key-based policies in Grey we made a point to ask about it. Conversely, if a user implemented something in Grey that did not exist in her original policy we asked about that, too. We also attempted to determine how each user's interactions with and attitudes about Grey changed over time.

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.

Principal	An individual who is being granted or denied access.
Access rule	A rule specifying a principal, a resource, and a condition such that the principal can access the resource provided that the condition is met.
Access-control policy	The collection of all access rules for a single resource.
Witness	A person trusted by the owner to observe an access, to ensure that the principal performs only the actions they are allowed to perform.
Trusted person	An individual who is trusted by a resource owner to make access-control decisions on their behalf; often a *trusted person* is also a *witness*.
Hidden key	A physical key which has been hidden and can only be used by those who know where it is.
Notification	A message sent immediately from the resource to the resource owner upon an access.

Figure 3: Terminology.

3.4 Analysis

We created a representation of each resource owner's ideal policy, using information from both implemented policies and data from our interviews. During the interviews we took special care to explore not only why the user had chosen to grant access to certain people but why some users, who would benefit from access, were denied it. We used the combination of the key-based policy, the policy implemented in Grey, and the user's explanations of both policies and her desired policy to derive the *access rules*, as defined in Figure 3, that comprise the ideal policy. Both the implemented and the ideal policies have an access rule for each principal/resource pair in our study (i.e., not only when access is explicitly granted).

To evaluate the accuracy of the policies implemented by physical keys and Grey, we compared the access rules in those policies to the access rules of the ideal policies. Due to the characteristics and limitations of keys and Grey as access-control mechanisms, the conditions under which accesses were allowed differed between the policies implemented by the two mechanisms and the ideal policies. We used the conditions to determine how well the implemented policies matched the ideal policies.

The set of conditions for rules in the ideal policies (Figure 4) was also generated using interview data. The first condition for ideal access (I1) allows the principal 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 principal 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). The conditions for keys and Grey follow a similar pattern. Conditions K1 and G1 allow the principal 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 principal 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 (the resource owner or a trusted 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). The conditions are discussed in more detail in Sections 4–6.

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,

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

the trivially satisfied condition). In addition, we made several assumptions involving implications between the conditions:

- **Ask owner via Grey (G3)** $\Rightarrow$ **Logged (I2)**: Asking the owner using Grey sufficiently logs the access.
- **Ask owner via Grey (G3)** $\Rightarrow$ **Owner notified (I3)**: Asking the owner notifies her of the access.
- **Ask owner who contacts witness (K4, G4)** $\Rightarrow$ **Logged (I2)**: Asking the owner directly sufficiently logs the access in both key and Grey implementations.
- **Ask owner who contacts witness (K4, G4)** $\Rightarrow$ **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 defined to be a principal/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 defined to be a principal/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 so may permit accesses that the implemented policy denies (rejects).

Note that 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 denote principal/resource policies, and not particular occurrences of that principal 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 evaluated in the system.

4 Ideal Policies

The ideal policies of the users in our study—the policies they would implement if not constrained by the limitations of a particular access-control system—contained 243 access rules. In this section we describe

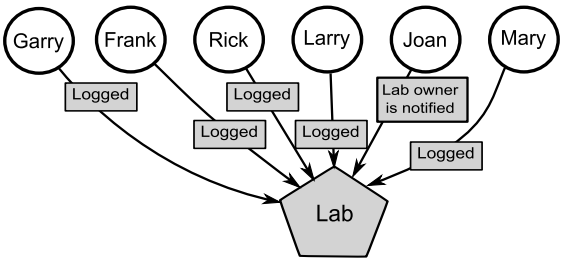

Figure 5: A representative portion of Mark's ideal policy. This policy consists of principals (circles), resources (pentagons), access rules (arrows), and conditions (boxes on lines).

Ideal access condition	*Number of occurrences*
I1. True (can access anytime)	19
I2. Logged	10
I3. Owner notified	3
I4. Owner gives real-time approval	4
I5. Owner gives real-time approval and witness present	7
I6. Trusted person gives real-time approval and is present	1
I7. False (no access)	199

Figure 6: The number of occurrences in ideal policies of each type of ideal condition.

the set of conditions that resource owners commonly required when specifying access rules and we discuss the scenarios that gave rise to each condition.

The number of access rules that depended on each type of condition is summarized in Figure 6. For illustration, we depict a representative portion of Mark's ideal policy in Figure 5.

True (access always allowed) Nineteen access rules granted unconditional access; that is, the condition under which access was granted was trivially (always) true, and the principals listed in the access rules had unconstrained access to the resource. Five of the eight resource owners created at least one rule with this condition.

Logged Ten access rules required that the access be logged. The intention of such rules was to allow access at any time, but only if a record of the access was made available to the resource owner. Only 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.

Mark's requirement for accountability was similar: since his lab houses expensive equipment he wanted all accesses to the lab logged. If the access-control system protecting his lab did not support logging, Mark was willing to give the six people with critical access needs access to his lab without logging, but he unconditionally denied access to others.

Owner notified Three access rules required that a notification message 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.

Mark explained how only a small number of his students had a good reason to enter the lab under normal conditions. However, if there was an emergency or if one of the servers stopped functioning in the middle of the night 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.

Owner gives real-time approval Four access rules required that the resource owner approve accesses on a case-by-case basis. This condition was used in scenarios in which the resource owner wished in general to deny access, except under circumstances that he deems exceptional. Furthermore, the resource owner wished to defer the decision about whether a particular circumstance is exceptional until it arises.

All four access rules that required this condition were part of Pat's policy concerning his office. Pat didn't have any shared resources in his office that other people would need and consequently saw no reason to give anyone access. However, occasionally packages with expensive equipment arrived for him when he was not there to receive them. In these cases, 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 Seven access rules required not only that the resource owner give approval on a case-by-case basis but that a third party witness the access. 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 didn't get along. When the TA needed to get into Ryan's office to retrieve tests or other papers related to his class, Ryan would have his secretary let the TA in and remain present to ensure that the TA did nothing else.

Donald, an administrator, had a similar problem: he did not like anyone going into his office to get software, printing supplies, or other resources when he was not present because they might unintentionally get the wrong item. If someone really needed an item from his office while he was gone, he would ask another administrator to let the person in and observe what they took. This way he was always certain who took what item.

Trusted person gives real-time approval and is present One access rule allowed a trusted person to make access-control decisions on the resource owner's behalf. The access rule also required that the third party witness the actual access.

The one use of this condition in our study occurred between Lisa, her secretary Emma, and a trusted person, Paul. Lisa had not gotten along well with her previous secretary and required Emma to ask Paul whenever she needed to get into 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 people and groups outside the scope of our study. A perfect example involved Mark's lab, which is shared among several faculty, all of whom were authorized to make decisions on Mark's behalf. Mark explained:

> On the lab we have a laminated sheet with contact info. ... it says here's six people that are responsible for the room, here is their office, mobile, or home phone numbers. If the room is on fire call these guys right away.

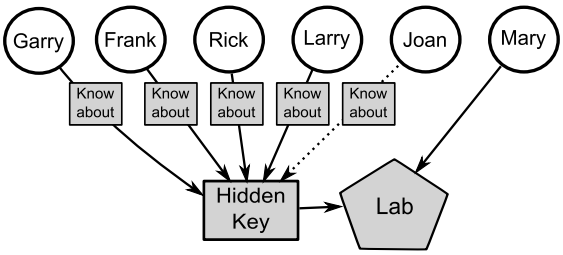

Figure 7: Mark's physical key policy. Four users know about the hidden key and one user (dotted line) could easily learn about the hidden key. The last user, Mary, has direct access to the lab.

Hidden keys assumption	*False accepts*	*False rejects*
Only authorized people know about hidden keys	8	11
Each person knows of at most one hidden key	64	8
All people know about all hidden keys	168	3

Figure 8: Counts of false accepts and false rejects for key policies under three different assumptions about who has knowledge of the location of hidden keys (see text for explanation of these assumptions). False accept and false reject counts are a measure of how well key policies matched ideal policies.

False (access always denied) One hundred thirty-six of the access rules in users' ideal policies unconditionally denied access.

5 Physical Key Policies

Figure 8 shows counts of the false accepts and false rejects we observed in key policies. The rows of the table correspond to three different assumptions we made regarding *hidden keys*. Hiding keys was a common practice in which a resource owner placed a key in a public space but only made its location known to a select group of principals. Use of a hidden key is evident in Mark's key policy, illustrated in Figure 7. Although hidden keys solve some of the problems associated with distributing keys, hiding keys is a very insecure practice. Unauthorized principals can easily find out the location of hidden keys by learning of their location from others, observing others retrieving them, or serendipitously coming across them. Thus, we counted false accepts and false rejects in key policies using three different assumptions about hidden keys. In our most conservative assumption about how hidden keys are used, we assume no unauthorized principals learn of their location. In a moderate assumption, we assume any given principal knows of at most one hidden key. In the most liberal assumption, we assume all principals know about all hidden keys.[2] We learned from interviews that, in fact, many unauthorized users knew the locations of hidden keys. In fact, our moderate assumption is probably the most realistic. The rows of Figure 8 show the false accept and false reject counts for key policies under these three different assumptions.

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

[2] Actually, since some hidden keys were hidden in offices, rather than in public spaces, our most liberal assumption is that all principals know of all hidden keys they are able to access; thus keys hidden in public spaces are known to all, but keys hidden in offices are only known to those able to access the office.

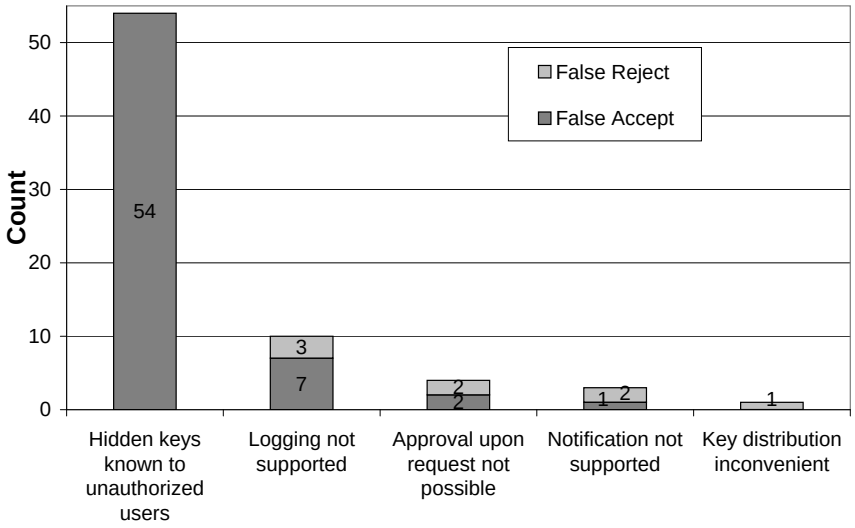

Figure 9: Counts of false accepts and false rejects by cause, under the moderate assumption that users know of no more than one hidden key to which they are not supposed to have access. Causes are listed in descending order of frequency of occurrence.

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

Figure 9 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 of policy discrepancies below.

Hidden keys Depending on the assumptions about how many unauthorized users were able to learn the location of hidden keys, there could have been zero to 160 false accepts due to hidden keys. Under our moderate assumption, which we believe to be the most realistic, there would be 54 false accepts due to hidden keys, as depicted in Figure 9.

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 principals to have access to a resource, or when an owner frequently allows access to new principals (as in a university, where new students arrive every year), a hidden key is often used to simplify key distribution.

Thomas is part of a large research center that is made up of twenty faculty, forty-two graduate students and twelve staff members. The research center maintains more than four lab spaces in three buildings, some of which require multiple keys to enter. Distributing and maintaining the 500 or more keys required to give each person access to each resource is impractical, especially since not every person has a regular need to access each resource. Thomas's research group solves the problem by providing keys only to those who need them and maintaining a set of hidden keys for use by everyone else.

However, hidden keys introduce a number of problems. Although they are easy to distribute, 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, as our various assumptions imply, it is quite easy for unauthorized users to learn the location of and gain access to hidden keys. Finally, hidden keys can be lost or stolen, thereby not only revoking access to authorized users, 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. In seven of these cases, access was granted without the logging condition being fulfilled, thus leading to false accepts; in three 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, who would let the student in. When asked why he didn't give his students keys, 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.

Approval upon request not possible There were four cases in which an owner would have granted access to resources upon request, 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 cases as false rejects, unless one of the relevant principals may have had unauthorized knowledge of a hidden key. Thus, under the moderate assumption, two of these false rejects became false accepts (because the principal was allowed access without fulfilling the desired condition), and under the liberal assumption, all four of these false rejects became false accepts.

All four false rejects were for Pat's office, to which only he had access. Since his office contained nothing normally needed by other people, he saw no reason to give anyone else access. When asked if there was any reason why someone else would need access, and if so, how they would get in, he replied that they would probably have to call him. 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.

Notification not supported There were three cases in which resource owners desired to allow access if they were notified that the access had taken place. Since keys do not support notification, these three cases inevitably led to discrepancies with the ideal policy. In two cases, no key access was granted, leading to false rejects; in one case, key access was granted via a hidden key, leading to a false accept. Under the liberal assumption about hidden keys, the false rejects became false accepts because the relevant principals could gain unauthorized access to a hidden key.

While Pat gives no one regular access to his office, there are at least two other faculty who are responsible for Pat's lab and might have a legitimate need to access his office in an emergency. Currently, Pat assumes that if the emergency was really large enough, or he couldn't be reached, they could just get campus security to open the office. However, in the ideal case he would like to give these faculty the ability to access his office at any time provided that he is notified.

Key distribution inconvenient Distributing a key entails the overhead of making the key, handing it to the person to whom it is to be given, and keeping track of who has the key. The inconvenience of distributing

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

Figure 10: Counts of false accepts and false rejects for Grey policies under two different assumptions about what constitutes the Grey policy. Since Grey allows nearly instantaneous delegation, Grey users in some cases chose to defer giving delegations until they were needed. The first row of this table shows false accepts and false rejects counting deferred delegations as false rejects. The second row shows false accepts and false rejects counting deferred delegations as though they had actually been given.

keys led to one case in which the owner's ideal policy called for access to be allowed, but the owner did not grant key access.

When asked why she hadn't given Lisa a key to her office, Emma responded that she wouldn't mind giving Lisa a key if Lisa ever asked, but that it really wasn't necessary. As a secretary, Emma works for several faculty members, of which Lisa requires the least time.

> Of my 100% I uh spend 60% on [one supported faculty], 30% on [another supported faculty] and only 10 on Lisa. . . If she needs to sign something I just put it under her door.

Emma also told us that getting keys to her office made for her work-study students took nearly two months. In the meantime, she was forced to leave her office door unlocked and provide them with a set of hidden keys so that they could do their jobs.

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

6 Grey Policies

Grey policies matched ideal policies quite closely. As Figure 10 shows, we observed no false accepts in Grey 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 assigning access to certain principal/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 three false rejects in Grey policies, all related to the fact that 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) Grey users could trivially implement anytime access by issuing a Grey credential to the relevant principal. Issuing credentials in Grey is easy and convenient; 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:

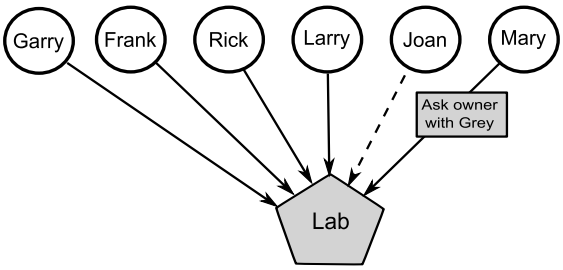

Figure 11: A representative example of Mark's Grey policy. Four people have a Grey delegation to the lab and Joan will be given a delegation when she asks (dotted line). Mary isn't given a delegation but if she asks the owner he will give her a temporary one.

> 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].

Karl also appreciated the convenience of being able to delegate easily. Tim is a student in charge of stocking soda, and Karl has a small refrigerator in his office which he likes to have stocked with soda. With Grey, delegating access to his office was sufficiently easy that Karl gave a delegation to Tim, although he had not bothered to give him a key. Karl describes his reasoning:

> [Delegation with Grey] was easy. Getting a key for him would not have been easy. I don't know, it just sorta came up. Now that [I] am using Grey, [I] can do this kinda thing.

Logged Grey-enabled doors log all Grey accesses, so Grey implements logging. In 10 cases where the ideal policy called for accesses to be logged, owners issued Grey credentials; since Grey accesses are logged, the logging condition in the ideal policy was fulfilled.

Mark gave six people access to his lab using Grey. He wanted all of them to be logged so that if anything went wrong or went missing he could hold the appropriate person responsible. During the course of our study Mark, asked us four different times to pull up 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 11.

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

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

Pat didn't give anyone access to his office using Grey, but he told us about a time when it was extremely useful to remotely grant 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 [to let the delivery guys into another room]. The request came in and I said, "You are in there one time buddy," and that is all he needed. He needed to get in here one time, get the key and get out.

Owner gives real-time approval and witness present In seven 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 (via Grey or, e.g., by phone), who would then contact a trusted person and ask them to serve as a witness. It was possible to fulfill these conditions using Grey because owners could choose any trusted person at the time of requested access and delegate to the trusted person the right to access the relevant resource (if they had not done so already).

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 In one case in which the ideal policy called for a trusted person (but not necessarily the owner) to give approval upon request and to serve as a witness, the owner issued a Grey certificate to a trusted person, who could then give access to the relevant principal upon request. This only ever 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 principals did not all have Grey phones and so the case was beyond the scope of our study. Eric only gives out access to his office using Grey because Grey supports logging, which Eric considers to be vital. Consequently, he has only given access to the small number of his students who have Grey phones. If any of his other students need access, Eric expects that they will ask a student who does have access to let them in.

False (no access) In 199 cases in which access was not allowed in the ideal policy, it was not granted through Grey.

7 Discussion

We discuss our results in two parts. First, we highlight findings from users' ideal policies and their implications for the design of access-control systems. Second, we discuss the reasons Grey policies more closely matched ideal policies than did key policies, how Grey might be extended to match ideal policies even more closely, and why features of Grey we thought would be used were not heavily used in this study.

7.1 Ideal policies and implications for design

Our primary finding with regard to users' ideal policies is that the conditions users desire to place on access to their resources fall into the seven categories discussed in Section 4:

1. Anytime access;
2. Anytime access, as long as access is logged;
3. Anytime access, as long as the owner is notified;

4. Access upon request, if approved by the owner;
5. Access upon request, if approved by the owner and observed by a person the owner trusts;
6. Access upon request, if approved and observed by a person the owner trusts;
7. No access.

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, notification, and real-time approval upon request were three desired conditions on access that are not supported by keys, but can be supported by a digital access-control system. Two other desired conditions, approval by a trusted person and presence of a trusted witness, 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, but can be made more practical and implemented more accurately by a system that does not require physical tokens to be passed around to delegate authority and enforces the presence of a "witness" through technological means (e.g., by activating a camera).

Our data on ideal policies and other interview data shed light on users' needs for several interesting features of Grey. We discuss three of these below: transitive delegation, policies with groups and roles, and on-demand delegation.

Transitive delegation Users commonly desired the ability to delegate control over their resources to others, such as administrative assistants and other trusted persons. When an owner can grant access to a principal, and that principal 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.

However, it is not always appropriate for delegations to be transitive. Some users brought up situations which fell outside the scope of our study but in which they wished they could delegate access to open a door, but not 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.

Groups and roles File access-control systems usually provide some capability for policy authors to arrange principals in groups or roles. We assumed that users in our study would also benefit from the ability to form groups of principals and to then assign privileges to groups rather than individuals. Our interview data support this assumption, in that users do think of groups of principals, such as "PhD students," "delivery people," "visitors," and so forth.

On-demand delegation We found that with Grey, users relied upon the ability to delegate access to resources when needed and upon request, an ability we call reactive delegation. Based on our users' desire to grant real-time access upon request, it seems important for an access-control system to support reactive delegation. It could even be argued, as some privacy policy researchers have [9], that reactive delegation's counterpart, proactive delegation (sometimes referred to as "configuration"), is virtually unnecessary. However, our data contradict 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.

7.2 Grey policies

The results of comparing Grey and key policies to ideal policies indicate that Grey policies match ideal policies much more closely than do key policies. Figure 12 shows a comparison of key and Grey policies

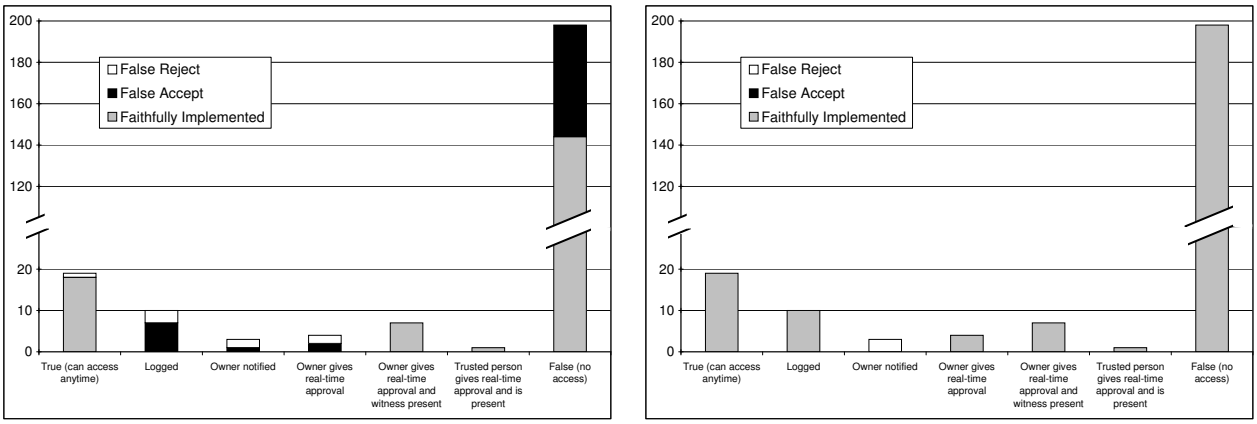

Figure 12: Graphs showing, for Grey policies (right) and key policies (left), counts of faithful implementations, false rejects, and false accepts by corresponding ideal policy condition. The graphs show that Grey policies yielded only three false rejects and no false accepts, while key policies yielded 8 false rejects and 64 false accepts. This data corresponds to our moderate assumption about who knows about hidden keys and to our assumption that deferred Grey delegations count as implemented access rules (see text for explanation of assumptions).

according to each of the 243 access rules implemented in our study. Each bar in the two bar graphs in Figure 12 corresponds to one of the seven ideal policy conditions we found. Each bar is subdivided to indicate faithful implementations of the ideal condition, false rejects, and false accepts. The bar graph on the left side of the figure shows key policy data, under our moderate assumption of who has knowledge of hidden keys, while the bar graph on the right side of the figure shows Grey policy data, under our assumption that deferred Grey delegations do count as implemented access rules. A quick visual inspection of the graph of Grey policy data shows that virtually all access rules were faithfully implemented in Grey; Grey policies only yielded 3 false rejects and no false accepts. On the other hand, key policies yielded 8 false rejects and 64 false accepts.

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 upon request, and the ability to log accesses. Furthermore, Grey's easy policy creation mechanism makes it superior in convenience to keys, which are expensive to make, distribute, and keep track of.

While our results show that Grey vastly outperforms keys as an access-control technology, there are improvements that could be made to Grey to better match users' ideal policies. First of all, it is evident from our ideal policy data that users would like notification functionality. While it would certainly be possible to implement notification in Grey, we have not yet done so because we did not anticipate a need for notification. That this study revealed such a need is a testament to the value of obtaining real-world data. Second, as noted in Section 7.1 above, Grey delegations are always transitive, but some users expressed a desire for non-transitive delegations. Like notification, non-transitive delegations could be implemented, but it took real-world data to make the need for non-transitive delegations apparent to us. Third, while Grey's easy, at-a-distance policy creation features enabled users in our study to implement policies requiring a trusted witness, Grey could make it even easier to enforce the presence of a witness. Currently, trusted witnesses are given access to a resource. This caused no problems in our study, because all trusted witnesses were allowed to have access to the relevant resource themselves in the ideal policies. However, Grey could enable an owner to delegate witness-only authority, without authority to access a resource alone, by requiring that multiple certificates be presented at a door simultaneously.

One Grey feature that, to our surprise, users in this study rarely used was the ability to create groups

of principals. Our interview data suggests that Grey users do think of other principals in terms of groups or roles, but that they simply did not use Grey's grouping feature. We have two possible explanations for this apparent discrepancy. First, it may be the case that the phone-based user interface for creating groups is too difficult to use. We are currently developing a desktop-based user interface for viewing and creating policies that will include group-creation features. We expect that the desktop-based user interface to be much easier to use, and that a better interface may encourage Grey users to create groups. Second, 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 them individually.

8 Related Work

The two technologies that we have compared and contrasted in this paper, namely physical keys and Grey, were chosen in part because of their wide disparity in adaptability to new access-control policies. That said, there are numerous other access-control mechanisms that could be considered in a study such as ours, e.g., proximity cards and swipe cards for physical resources, or passwords, RSA SecureID tokens,[3] 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, it seems that the limitations of these technologies (particularly, the lack of a user interface on the access token) would make it difficult for them to implement the kinds of reactive policies that our users desired. In addition, there are several proposed distributed systems that use portable devices to control access to physical spaces [4, 15]. However, as far as we know none of these systems have been implemented.

An example of an implemented access-control technology that supports dynamic delegation is file access control, and there have been several usability studies of this type of technology. Cao et al. showed that standard ACL interfaces had a high failure rate, despite users expressing confidence that they had manipulated the ACLs accurately [5]. 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 [8, 12]. Users also have difficulty understanding how different policies interact; for example, when a group is granted access to a resource but an individual who is a member of that group is denied access [12]. These studies look at how users build and manipulate access-control polices in a single session but they don't consider how these policies are managed and changed over time. They also do not consider what user needs are not covered by the chosen policy language. Similarly, there have been some distributed access-control file systems that have been implemented (e.g., [13]), but there is little discussion of the policies created by users of these systems.

The security community has designed and formally discussed many access-control policy languages (e.g., [1, 10, 11, 2]), each supporting a different set of policies. However, we are unaware of published research on the usability of these languages or 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 access-control needs of 28 commercial and government organizations and identified seven access-control approaches. One approach they discuss is the *discretionary access control* (DAC) approach, in which access is assigned to individuals and groups, who in turn may delegate that access to others. The authors note that DAC is well suited for organizations where end-users often have rapidly changing information access needs and must be able to specify access-control policy for resources they control. Although DAC is usually implemented through access-control lists (ACLs), they authors point out that when these ACLs are centrally administered they "can become clumsy and difficult to maintain." They also note that the DAC approach is not suitable for organizations concerned with maintaining tight

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

controls on access rights [7]. 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. They are often frustrated by current access-control mechanisms that are difficult to use and not well-suited to users' workflow [14].

Systems that allow end users to configure privacy settings may be thought of as access-control systems, as they involve policies that govern access to a user or to a user's personal information. Researchers have examined the usability of various approaches to end-user privacy configuration. For example, Cranor has developed and evaluated privacy user agents that warn users about web sites that do not match a user's specified privacy preferences. She notes that user privacy preferences tend to be complex and nuanced, and that users have little experience articulating these preferences [6]. Lederer et al. identified pitfalls common to privacy interaction design: obscuring potential and actual information flow, emphasizing configuration over action, lacking coarse-grained control, and inhibiting established practice. Some of their lessons on how to avoid these pitfalls are applicable to access-control-system design. For example, while in privacy interaction design there is a need for users to understand what information is disclosed to whom, in access-control systems users need to understand who has access to what. Likewise the authors identify the need for designs that support rather than inhibit established social practice [9].

9 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 case study of access-control policies, both ideal ones and as implemented via two technologies, namely physical keys and the Grey system. Moreover, we have developed a methodology for 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 itself, illucidate 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 with ease, in order to extend access-control policies. The failure of physical keys to implement the latter is among the primary reasons for the emergence of hidden keys on the floor, an instance of "security by obscurity" that breaks down as knowledge of the hidden key leaks.

The results of this study help us to prioritize further developments in the Grey deployment, i.e., to focus on those policies that users want but that we do not yet support. For example, desire for a trusted witness might be satisfied through the deployment of cameras in offices that are turned on when a "witness" is needed. It is our hope that the results of our study can similarly aid others in the development of 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 *Proceedings of the 6th ACM Conference on Computer and Communications Security*, Singapore, Nov. 1999.

[3] 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*, volume 3650, pages 431–445, Sept. 2005.

[4] A. Beaufour and P. Bonnet. Personal servers as digital keys. In *Proc. 2nd IEEE International Conference of Pervasive Computing and Communications*, Mar. 2004.

[5] X. Cao and L. Iverson. Intentional access management: Making access control usable for end-users. In *Sumposium On Usable Privacy and Security (SOUPS)*, 2006.

[6] L. F. Cranor. Privacy policies and privacy preferences. In L. F. Cranor and S. Garfinkel, editors, *Privacy and Usability*, pages 447–471. O'Reilly, Sebastopol, CA, 2005.

[7] 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.

[8] A. Kapadia, G. Sampemane, and R. H. Campbell. Know why your access was denied: regulating feedback for usable security. In *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 52–61, 2004.

[9] 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, November 2004.

[10] N. Li and J. C. Mitchell. Understanding SPKI/SDSI using first-order logic. *International Journal of Information Security*, 2004.

[11] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130, Oakland, CA, May 2002.

[12] 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.

[13] S. Miltchev, V. Prevelakis, S. Ioannidis, J. Ioannidis, A. Keromytis, and J. Smith. Secure and flexible global file sharing. In *USENIX Technical Annual Conference, Freenix Track*, 2003.

[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, New York, NY, USA, 2006. ACM Press.

[15] F. Zhu, M. W. Mutka, and L. M. Ni. The master key: A private authentication approach for pervasive computing environments. In *Fourth IEEE International Conference on Pervasive Computing and Communications (PerCom'06)*, pages 212–221, 2006.