Attribute-Based, Usefully Secure Email

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Christopher P. Masone

DARTMOUTH COLLEGE
Hanover, New Hampshire
August, 2008

Examining Committee:

(chair) Sean W. Smith, Ph.D.

David F. Kotz, Ph.D.

Christopher J. Bailey-Kellogg, Ph.D.

Denise L. Anthony, Ph.D.

M. Angela Sasse, Ph.D.

Charles K. Barlowe, PhD
Dean of Graduate Studies

Abstract

A secure system that cannot be used by real users to secure real-world processes is not really secure at all. While many believe that usability and security are diametrically opposed, a growing body of research from the field of Human-Computer Interaction and Security (HCISEC) refutes this assumption. All researchers in this field agree that focusing on aligning usability and security goals can enable the design of systems that will be more secure under actual usage.

We bring to bear tools from the social sciences (economics, sociology, psychology, etc.) not only to help us better understand why deployed systems fail, but also to enable us to accurately characterize the problems that we must solve in order to build systems that will be secure in the real world. *Trust*, a critically important facet of any socio-technical secure system, is ripe for analysis using the tools provided for us by the social sciences.

There are a variety of scopes in which issues of trust in secure systems can be studied. We have chosen to focus on how humans decide to trust new correspondents. Current secure email systems—such as S/MIME and PGP/MIME—are not expressive enough to capture the real ways that trust flows in these sorts of scenarios. To solve this problem, we begin by applying concepts from social science research to a variety of such cases from interesting application domains; primarily, crisis management in the North American power grid. We have examined transcripts of telephone calls made between grid management personnel during the August 2003 North American blackout and extracted several different classes of *trust flows* from these real-world scenarios. Combining this knowledge with some design patterns from HCISEC, we develop criteria for a system that will enable humans apply these same methods of trust-building in the digital world. We then present *Attribute-Based, Usefully Secure Email* (ABUSE) and not only show that it meets our criteria, but also provide empirical evidence that real users are helped by the system.

Acknowledgments

When I was born, my family began saving money with which to send me to college. Whatever college I wanted. It was important, my parents believed, that my intellect and my drive be the only limiting factors on my education. They wanted me to have the freedom to pursue every opportunity that my mind could earn me; because of their hard work and sacrifice, I was able to. I would not have been able to complete this dissertation without the help of a myriad of people, all of whom I try to remember and thank in the next few paragraphs of text. I would not have even been able to start without the support of my mother and father.

Like many PhD students, I started graduate school with nebulous research goals. I knew that I wanted to work in systems; I liked building things, and the chalk dust in the theory labs made me sneeze. I drifted somewhat until, in Sean Smith's *Security and Privacy* topics class, I read "Why Johnny Can't Encrypt," Whitten and Tygar's seminal work in the area of security and usability. I was intrigued by this intersection of people and programs, and asked Sean if I could join his research group to study the wonderful and vexing problems that arise when the messy real world intrudes upon the clean-room of academic Computer Science. As my advisor, Sean has been unfailingly supportive, believing in me even when I didn't believe in myself. As I barreled towards my defense, his willingness to adapt to my feast-or-famine work schedule provided me with a constant source of feedback on my text, making this document far better than it would have been otherwise. I cannot tell him how much his help has meant to me.

My entire committee deserves to be acknowledged for their participation in my research, but I thank Denise Anthony and Angela Sasse especially for their help in understanding the social science research that played such an important role in my work. Furthermore, I would have been lost without Denise's help in designing my user studies and evaluating the resulting data. I also want to thank my colleagues from the Trustworthy Cyber Infrastructure for the Power Grid project, especially Matt Davis, Peter Sauer and Tom Overbye; their power grid domain knowledge proved invaluable when fleshing out the setup of my power-grid-based user study. On a broader scale, I owe a debt of gratitude to Simson Garfinkel and Alma Whitten, whose HCISEC work inspired and guided my own research. I can only hope that some future student reads my work as closely as I have read theirs.

On a more personal level, I must thank my colleague and friend Sara Sinclair for her unfailing cheer, her ever-present shoulder, her enthusiasm for white-board brainstorming, and for roping me into the most rewarding experience of my graduate school career. I thank my interns Kate Bailey and Linden Vongsathorn for allowing me to mother them incessantly, I thank my lab mates for providing me with coffee conversation and implementation advice, and I thank the College for providing me with ten years of unparalleled academic opportunity. Lastly, I thank my doctor and friend Kari Gleiser. Of all the work I have done in the six years since I started my PhD, the work I have done with her has been the most difficult, and the most valuable. I thank her for not only helping me weather the stress and despair that accompanies many courses of doctoral study, but also for the person that I have become through the time we have spent together.

This dissertation has been the most massive project I have ever undertaken, and inspiration and assistance may have come from corners that I have failed to enumerate here. To those people as well, I offer my apologies and, again, my gratitude.

This research was supported in part by the NSF (under grants CNS-0448499 and CNS-0524695) and by Cisco. The views and conclusions do not necessarily reflect those of the sponsors.

Hanover, NH
August 2008

Preface

Code for ABUSE is available upon request.
Contact information:

- The author: `Christopher.P.Masone@alum.dartmouth.org`
- His advisor, Sean Smith: `sws@cs.dartmouth.edu`

Contents

Chapter 1

Introduction

“The goal of security is not to build systems that are theoretically securable, but to build systems that are actually secure” when real people use them in real-world scenarios. [119] In other words, systems need to be not only secure, but also need to be usably secure. While many believe that usability and security are diametrically opposed, a growing body of research from the field of Human-Computer Interaction and Security (HCISEC) refutes this assumption. All researchers in this field agree that focusing on aligning usability and security goals can enable the design of systems that will be more secure under actual usage.

Security software is always deployed within some social context. [47] As it is not always feasible or desirable to excise humans from security tasks, secure systems are *socio-technical* in nature; humans and technology work together to achieve some production task while also keeping the system secure. [11] Realizing this and bringing to bear tools from the social sciences (economics, sociology, psychology, etc.) can not only help us better understand why deployed systems fail but also enable us to accurately characterize the problems that we must solve in order to build systems that will be secure in the real world. *Trust*, a critically important facet of any socio-technical secure system, is ripe for analysis using the tools provided for us by the social sciences.

There are a variety of scopes in which issues of trust in secure systems can be studied. We have chosen to focus on how humans decide to trust new correspondents. Current secure email systems—such as S/MIME and PGP/MIME—are not expressive enough to capture the real ways that trust flows in these sorts of scenarios. We agree with Ackerman; “human activity is highly flexible, nuanced, and contextualized and...computational entities such as information transfer, roles, and policies need to be similarly flexible, nuanced, and contextualized.” [1] To solve this problem, we begin by applying concepts from social science research to a variety of such cases from interesting application domains; primarily, crisis management in the North American power grid. We have examined transcripts of telephone calls made between grid management personnel during the August 2003 North American blackout and extracted several different classes of *trust flows* from these real-world scenarios. Combining this knowledge with some design patterns from HCISEC, we develop criteria for a system that will enable humans apply these same methods in the digital world. We then present *Attribute-Based, Usefully Secure Email* (ABUSE) and not only show that it meets our criteria, but also provide evidence that real users are helped by the system.

As public key cryptography and Public Key Infrastructure (PKI) are critical enabling technologies for existing trustworthy communication systems—and also for ABUSE—we discuss them in the next section of this introduction. Then, since "trust" is a heavily overloaded term across several fields, Section 1.3 will present and explicate the meaning *we* intend to use in this work. In Section 1.4, we motivate our work by presenting a set of requirements for trustworthy communication and explain how current paradigms fail to meet these criteria. We then present an overview of ABUSE, our solution to the problem of trustworthy communication between individuals who share no prior trust relationship. Finally, we provide an outline for this document and enumerate the contributions made by this thesis.

1.1 Enabling technologies

1.1.1 Public key cryptography: a building block

Public key cryptography [110] enables the secure exchange of information between multiple parties without requiring them to share any secrets a priori. If Alice wishes to enable other parties to communicate with her in secret, she first generates a pair of specially-related cryptographic keys (called a *key pair*). One of these becomes her *private key* and must be kept secret; otherwise, the system breaks down. The other key becomes Alice's *public key*, and can be distributed as widely as Alice desires. Any information enciphered with Alice's public key can only be deciphered with the associated private key. Thus, anyone wishing to communicate with Alice in secret must only encrypt their information with Alice's public key and send it to her. Provided that she has been diligent about maintaining the secrecy of her private key, only Alice will be able to read this information.

Public key cryptography can also work in the other direction; information that can be deciphered with Alice's public key can only have been enciphered with her private key. Thus, if Alice has been diligent about maintaining the secrecy of her private key, she can use public key cryptography to prove that a piece of data came from her, and only her.

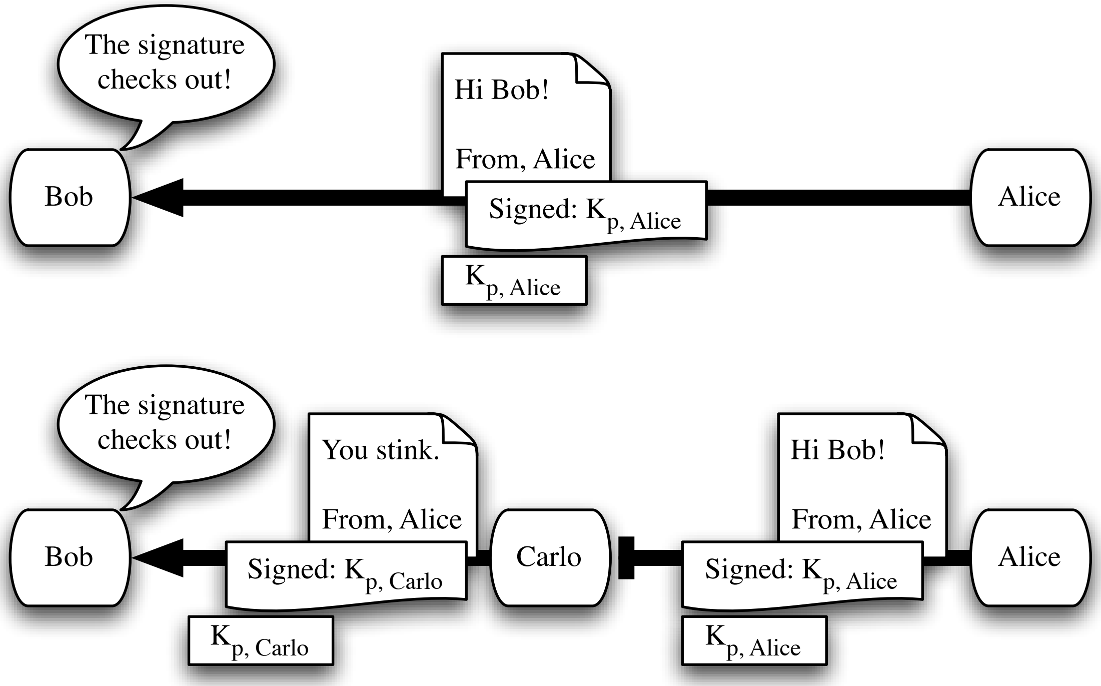


Figure 1.1: Assume that the only way Bob can learn Alice's public key is to trust her to provide it along with her signed messages. If Carlo blocks Alice's message and forwards a message of his own purporting to be from Alice, but signed with *his* key...Bob has no way of knowing this has happened!

These *digital signatures* [109] are often used to provide the basis of trustworthy messaging systems as they provide a number of important features. A valid digital signature over a message provides

- *integrity*: the data cannot have been changed since it was signed;
- *non-repudiability*: the signer cannot later assert that she did not send the message[1]; and
- a binding between the message and the signer's key pair.

These are very useful features, but we have glossed over an important detail: a signature cannot be verified without knowing which public key to use. In a trustworthy messaging context, the verifier cannot trust the sender to provide both the signature *and* the (unvouched-for) public key; attackers could simply replace messages wholesale, signed with valid-but-meaningless signatures, as shown in Figure 1.1. This is an instance of the problem known as *key distribution*. Even if this problem is solved, allowing the verifier to get

[1] unless she can prove that his private key was leaked before the signature was generated

the real sender's public key and thus determine that the signed data is appropriately bound to it, this is still not enough to enable many real-world applications. To make meaningful trust decisions about signed data, the signature verifier needs

- to know that the sender's public key is bound to some real-world information, and
- this information to tell her "what she needs to know" to come to a conclusion.

Public Key Infrastructures (PKIs) were created to address these two issues, though it is perhaps more accurate to say that PKIs were created to address the first issue by binding a key to an identity, with the tacit belief that this solved the second issue as well. We show that identity does *not* always tell the verifier "what she needs to know" in Section 1.2.1, and then begin exploring what humans *do* need to know in order to establish trust in Section 1.3. First, though, we discuss PKIs.

1.1.2 Public Key Infrastructure (PKI)

For a thorough discussion of PKI, see [115]. Here, we merely hit the high points. PKIs are commonly concerned with binding key pairs to identities. Every kind of PKI rests on some underlying trust assumptions; as long as Bob believes these assumption to be true, he can rely on the infrastructure to provide him with a binding between Alice and her key pair. So, we can say that, as long as Bob buys into the trust assumption that underpins his PKI, he can trust it to reliably provide him with Alice's public key when he needs to verify signatures on the messages that she sends. Strictly speaking, a verifiable digital signature over some message shows only that the holder of the private key K_s associated with a public key K_p signed the message. PKI is necessary to provide a mapping from "holder of the private key K_s" to "Alice". Thus, digital signatures plus a PKI have provided meaningful message integrity and non-repudiation. The introduction of a PKI has also provided a way to move from a binding between a message and a key pair to a binding between a message and the sender's identity. (It is true that providing a meaningful identifier for the identity "Alice" is a thorny one. [37] For the purposes of this discussion, which is about email systems, we consider an email address as a proxy for identity.)

There are two different models of identity mapping used by the predominant PKIs: the hierarchical X.509 model [19] and the PGP web-of-trust model [48] (SDSI/SPKI, not widely used, is discussed in Section 4.2.3).

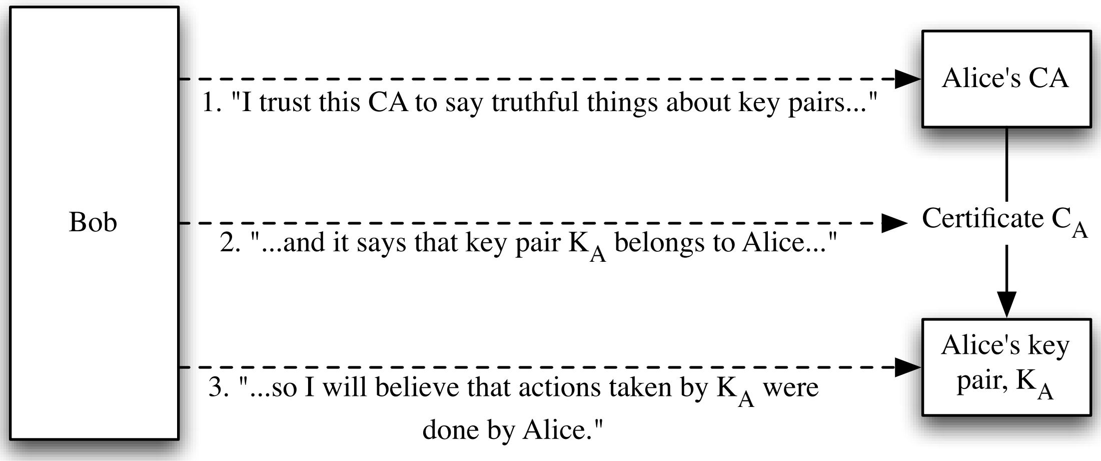

Bob
Alice's CA
1. "I trust this CA to say truthful things about key pairs..."
Certificate C_A
2. "...and it says that key pair K_A belongs to Alice..."
Alice's key pair, K_A
3. "...so I will believe that actions taken by K_A were done by Alice."

Figure 1.2: In a simple X.509 PKI, relying parties believe certificates issued by the CA they have designated as their trust root. Adapted with permission from [115, p. 251].

Hierarchical X.509 identity PKIs

The underlying trust assumption in X.509 is that there exist *trust roots*, trusted third parties who can speak authoritatively about entities in the system. In the most basic X.509 PKI, there is a single trust root, a *Certification Authority (CA)* that issues digitally signed statements—called *certificates*—binding a public key to an email address for some period of time (Figure 1.2). The CA asserts, for example, a binding between the public key K_p and the email address “alice@foo.com”, and guarantees this to be true from January 1, 2008 to January 1, 2009. Perhaps Alice did not receive the key until January 1, 2008, or perhaps “foo.com” did not wish to officially associate with Alice until that date. Perhaps her appointment at foo.com ends on January 1, 2009, or the company wishes to require her to get a new key pair before that day. Either way, the CA will not guarantee that the entity generating signatures verifiable with K_p is “alice@foo.com” outside of this validity period. There is a concept of a *Registration Authority* in X.509, whose job it is to verify for the CA that the entity asking that its key pair be bound to “alice@foo.com” is really an entity for whom the CA should be willing to perform that binding. Here at Dartmouth, for example, when we are issued our high-assurance X.509 certificates, we must appear in person at a Registration Authority and present our Dartmouth ID card.

In this simple model, Bob gets a signed message from Alice, acquires her certificate, checks that it is appropriately signed by the CA he trusts, checks that it is currently valid, and then uses her public key to verify her signature. As in Figure 1.3, Alice can even be the one who provides her certificate to Bob; she cannot fake it, as she does not have access to the CA’s private key.

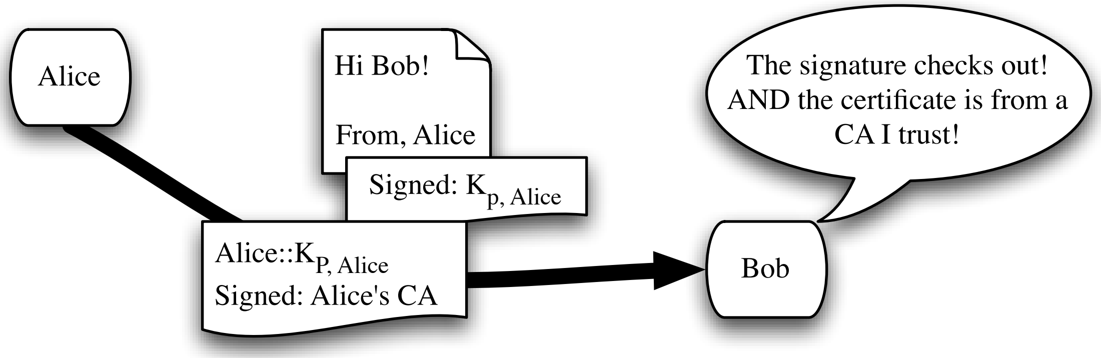


Figure 1.3: Alice signs mail to Bob with the key bound to her by her CA. Bob trusts Alice's CA, so he knows that the key she provides (in her certificate) is ok to use for signature verification.

Real life is, of course, not this simple. First, there is the fact that we have many CAs in the world, and users need to decide which ones to trust. Generally, some organization (perhaps the one Bob works for, or the one from whom he downloaded his email software) has decided for Bob which CAs are trustworthy, and he (perhaps unknowingly) chooses to believe key bindings based on signed assertions from these authorities. The trust placed by Bob in his X.509 PKI is generally all *institutional*[2]. Bob could also develop a *process-based*[3] trust relationship over time with a new CA, deciding to add it to his list of trust roots after many positive experiences with certificates issued by that authority. Any trust that Bob extends to users subordinate to this new root would still be institutional, however, as he is relying on the formal certification process of the CA in order to define the authority-to-user relationship.

In addition to the issue of multiple trust roots, there is also the issue of multi-level hierarchies, summarized in Figure 1.5. Perhaps Alice's company, Foo.com, has gone multinational and wishes to manage certificates for American employees and Canadian employees separately. It could set up two CAs, but then any software that uses foo.com certificates would need to be configured to use both as possible trust roots. Instead, it can have a single root CA issue certificates to two subordinate CAs, one for America and one for Canada. These CAs use the key pairs certified by the foo.com parent CA to issue credentials to end users. When Alice signs up with the Foo.com American CA, she gets back not just her certificate, but also the certificate of the CA itself. This *certificate chain* connects Alice all the way back up to the original trust root. Bob, who has only the top-level Foo.com CA installed as a trust root in his client, can still validate Alice's certificate as long as she provides him with the whole chain she got back from the Foo.com American CA.

[2] We discuss *institutional* trust in Section 1.3.

[3] We discuss *process-based* trust in Section 1.3 as well.

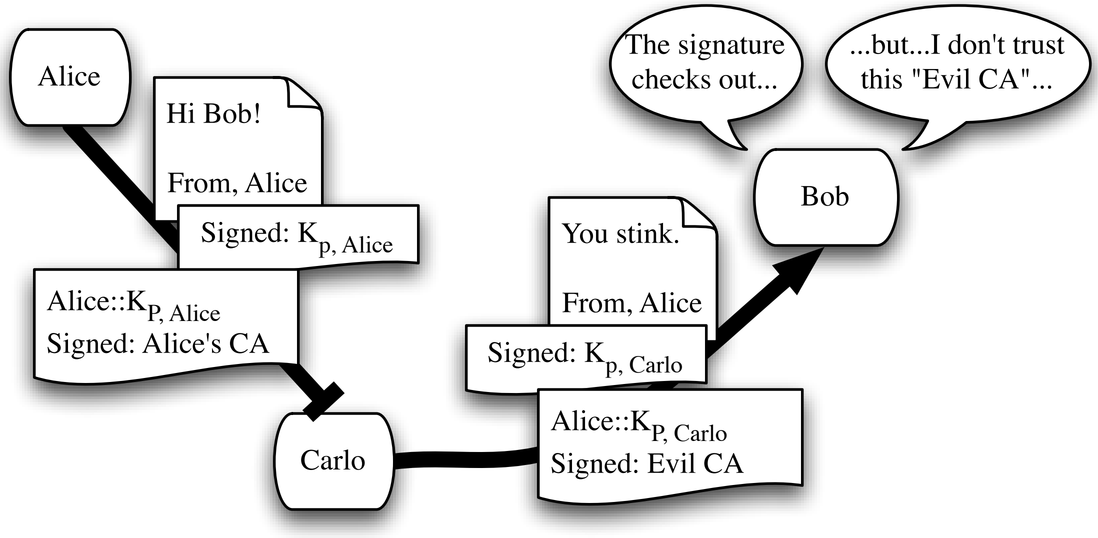


Figure 1.4: Carlo blocks Alice's mail, as in Figure 1.1. When he sends his mail to Bob, he tries again to fool Bob into happily verifying his signature with the provided key. With PKI in place, Bob sees that the key is not signed off on by the CA he trusts—Alice's CA—and so he ignores the message.

Multi-level hierarchies work well for distributing CA responsibilities inside a single organization. However, there exist cases in which many organizations wish to be able to leverage each other's certificates, but are not willing to subordinate themselves to some higher-level CA. They may choose, in this case, to create a bridged PKI. In this hub-and-spoke model, explained in Figure 1.6, all parties can accept each other's certificates without having to tell their users to trust any CA besides their own. Some existing bridges include the Federal Bridge Certification Authority [39]; the CertiPath bridge [14] that links aerospace PKIs; SAFE [102], the bridge for the pharmaceutical industry; and the Higher Education Bridge CA (HEBCA) here at Dartmouth.

We have not yet addressed the issue of what happens when a CA wishes to break a binding between a key pair and an identity. This is called *revocation* and it is rife with problems. [37, 53, 115] There are a bevy of approaches to the issue [19, 43, 81], each with their own tradeoffs. As the revocation problem is orthogonal to our work, we do not explore this any further. We do discuss, however, the impact of revocation on ABUSE in Section 5.5.1.

These are the issues surrounding the hierarchical X.509 model of a PKI that are germane to the rest of this research. There are many other interesting issues in this space that are explicated in [115] and many other places, but we will not go into them here.

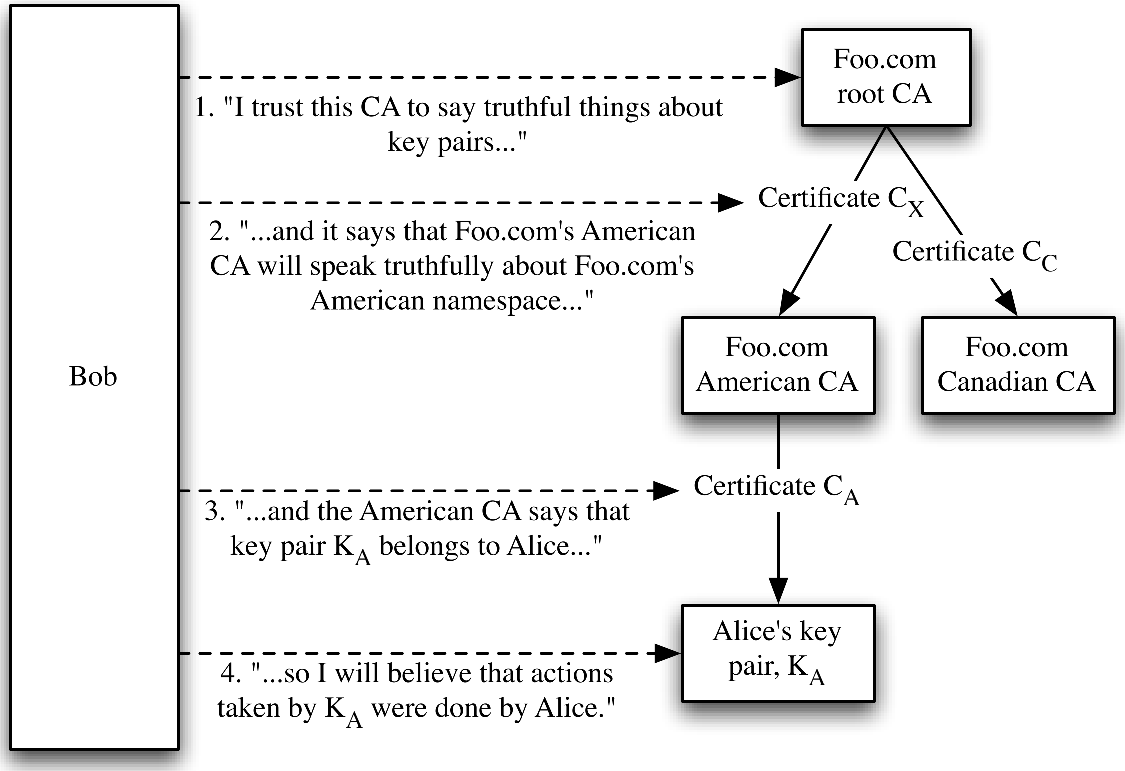

Bob
Foo.com root CA
1. "I trust this CA to say truthful things about key pairs..."
Certificate C_X
Certificate C_C
2. "...and it says that Foo.com's American CA will speak truthfully about Foo.com's American namespace..."
Foo.com American CA
Foo.com Canadian CA
Certificate C_A
3. "...and the American CA says that key pair K_A belongs to Alice..."
Alice's key pair, K_A
4. "...so I will believe that actions taken by K_A were done by Alice."

Figure 1.5: In a multi-level X.509 PKI, Bob believes certificates that chain back to the CA he has designated as his trust root. So, even though Alice's key pair is certified by the Foo.com American CA, Bob will still believe the binding between Alice and her key. Adapted with permission from [115, p. 260].

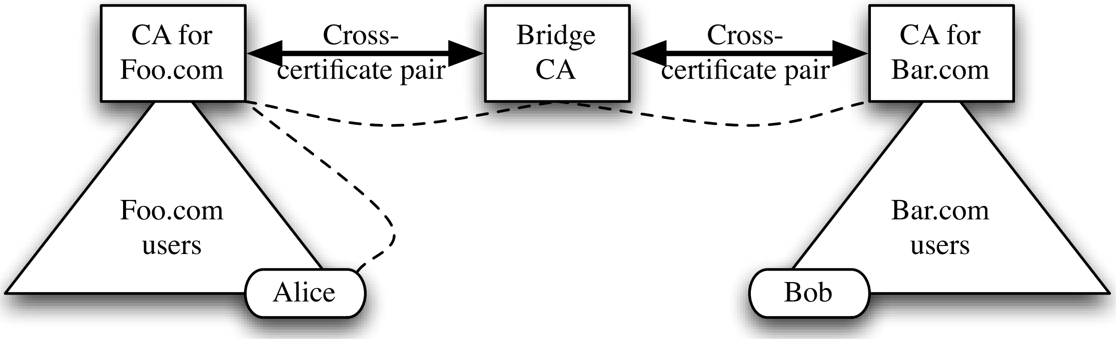


Figure 1.6: In a bridged X.509 PKI, trust roots at different organizations *cross-certify* with a go-between bridge CA. Foo.com's root issues a certificate to the Bridge CA, and vice versa. The Bar.com CA does the same. This enables Alice at Foo.com to present certificate chains to Bob that he can accept—without having to trust any CA other than his own.

The PGP web-of-trust

PGP [48] is a PKI that does not rely upon CAs. Instead, PGP requires users to publish their keys and then build a "web of trust" by signing off on keys they know to be good. For example, assume Alice feels sure that the key she has for Bob really does belong to Bob. So, Alice signs Bob's key. Cindy trusts the binding she has for Alice's key, and wishes to know if she can trust Bob's key. Because Alice has signed Bob's key, Cindy decides that she too can trust (at least somewhat) the accuracy of the binding of Bob to his key. This is the underlying trust assumption of PGP: that Cindy can make reasonable assumptions about the correctness of the binding between Bob and his key based on what Alice says. This provides more flexibility and user control than X.509, but also requires the user to perform more key management tasks, something that many users are not very good at. [125] Also, it is worth noting that the consistency of PGP's web-of-trust model has been called into serious question. [66]

1.2 Motivation

1.2.1 When identity is not enough

As we have seen, PKIs work to establish a binding between identity and a key pair. In a small organization, most users probably know each other and this will be enough to establish trust. In these settings, identity-focused technologies such as S/MIME [95, 96] or PGP/MIME [35] (both discussed in Chapter 4) would likely serve well enough as trustworthy email systems[4]. In large organizations, though, it becomes less likely that a sender and recipient knew each other prior to contact. Thus, assurance of only the sender's name and/or email address would not be enough to help the recipient make a good decision. Systems that focus only on identity are not expressive enough to allow users to specify the right properties for conclusions in human trust settings. [5]

The first class of issues arises when users expect that a name, verified by the communication system, equates to a person. In Dartmouth's Computer Science department, for instance, our grant manager shares a name with many other people in our Name Directory. They are distinguishable only by middle initial (not helpful) and department (since she is listed in a generic administrative unit, also not helpful). We don't care that mail is from

[4] If the reader is protesting that S/MIME, when used properly, may also provide users with some notion of each other's organizational affiliations, we are aware of this and will address it in Section 4.1.1.

[5] The authors, along with another colleague, previously explored expressiveness problems in trustworthy email systems in [117]. The following paragraphs are adapted from that report and material presented first in my thesis proposal.

"Joan B. Wilson"[6], we care that mail is from "the Joan Wilson who manages grants for Computer Science at Dartmouth."

A second class of issues arises when a name, verified by the communication system, does not tell the user what he needs to know. A senior colleague has preached the need for academic PKI to prevent a repeat of an incident at Yale in which someone forged mail from the Dean, canceling classes. Here at Dartmouth, we often receive "mail from the Dean" that is not from the Dean at all, but from one of the Dean's administrative assistants. Focusing on identity doesn't help at all in this case, because a signature from "John Wilson" doesn't help unless the recipients know that "John Wilson" is the new assistant to the Dean, and is allowed to speak for her on such matters.

The final class of expressiveness issues shows up when the same property does not mean the same thing in different contexts. Take the case of a colleague who moved to another university, and was asked for an extension by a student who had an athletic event in which he needed to participate. Our colleague, used to Dartmouth where coaches are faculty or staff, gave permission—pending an email from the student's coach (whom he assumed was a responsible adult) confirming the event. "John Wilson", who really was the coach, sent mail to confirm. "John Wilson" was also a student, happy to help his friend get out of work.

1.2.2 Using scenarios from the power grid

In working on this thesis, we became involved with the Trustworthy Cyber Infrastructure for the Power Grid (TCIP)[7] project. Through this involvement, we were introduced to specifics of the North American blackout of August 2003 and were able to draw a number of interesting trust scenarios from these events. They all fall into the second class of issues, in which "names are not enough."

Even within the same power company, operational decision makers sit in centralized control facilities that are geographically separated from the power generation and transmission stations. Furthermore, many different companies and management organizations need to collaborate in the event of a crisis. Thus, there is nearly always a requirement for some kind of technologically mediated communication, and a reduced likelihood that the people who run the actual equipment are personally familiar with all the people authorized to request operational changes. Additionally, we have seen these centralized control facilities, and observed their control panels annotated here and there with handwritten notes indicat-

[6] All names have been changed to some variant of "John Wilson" for anonymity, in reference to Carl Ellison's anecdote explaining why globally unique names are not usable [116]

[7] `http://www.iti.uiuc.edu/tcip/index.html`

ing the myriad of small ways in which standard procedure needs to be worked around in the cases of various facilities and pieces of equipment. Operators may need to take the central controllers at their word in situations that involve these exceptions. Moreover, deregulation has created a greater number of organizational boundaries within the industry than ever before [62], even further decreasing the probability that communicants share pre-existing trust relationships. Currently, this communication is primarily done via telephone. As was observed during the 2003 blackout, relying on control room phones for communication during emergencies can be problematic; a given individual can only be handling one call at a time, and a lack of available phones can cause a bottleneck. Migrating communication in the grid to some form of digital messaging system could alleviate the bottleneck issue, but current technologies do not provide support for the kinds of trust building we saw during the blackout.[8]

As an example, a simple case involved Alice at Company A calling Bob at Company B asking him to modify an erroneous request for an action that she had submitted. Alice and Bob shared no prior relationship, and Alice's name was unknown to Bob. However, Bob used to work with Carlo, who is now also at Company A. Once Alice can show that Carlo is on board with her request, Bob is willing to make the change for her [88, pp. 56–58]. This situation and several others are analyzed further in Chapter 2.

1.2.3 Current coping strategies are not good enough

According to informal surveys we have conducted, problems of this nature are currently worked around by phone calls, searching the institution's website, checking a company directory, or just assuming that everything is fine. Even in sensitive applications, such as in the power grid, we see users fall back to making phone calls to people that they know when they are in questionable trust scenarios [88, pp.56–58]. We also understand from our contacts in that industry that technicians in the grid frequently accept "talking the talk" and knowing the right contact information as proof of authorization. [30] The "unmotivated user" property of security [125], which states that users will give up on behaving securely if it is too difficult or annoying, leads us to believe that the last case is the most likely. Therefore, relying upon people to check up on every potentially sensitive email is not a winning strategy. Given this, it becomes clear that a system which addresses the kinds problems detailed above as a part of the normal workflow is required for trustworthy email.

[8] We are aware of Trust Management and Automated Trust Negotiation technologies. We discuss them, and why they are inappropriate for these scenarios, in Section 4.2.1 and Section 4.2.2.

1.3 Humans and their trust

In this section, we create a definition of trust that borrows from both sociological literature and and technical literature. We use this definition to analyze how trust is built in human scenarios throughout this work, so that we can apply what we learn to the design of a system that can better facilitate these kinds of use cases. We believe this analysis to be novel.

1.3.1 Our notion of trust

For the purposes of this work, we consider *trust* to be the willingness of an entity to undertake a potentially dangerous action at the behest of a second entity as a result of a set of shared expectations between the two (we built this definition from [135] and [17]). There are two components to this set of expectations:

- **Background expectations**, the assumptions defined by a "world known in common" [111], and
- **Constitutive expectations**, the parameters of the particular situation. [45]

This kind of trust, based on building a shared context, is called *calculus-based trust* [101] in the parlance of the social sciences. According to [135], there are three main ways to create this context between two parties:

- **Process:** using reputation and prior experience.
- **Characteristic:** using innate attributes, e.g. family background, gender or ethnicity.
- **Institutional:** using formal social structures, like certifications or membership in a professional organization.

We do not speak simply of one individual trusting another. When we talk about trust, we refer to one individual trusting another for the purposes of a given transaction. Alice does not simply "trust" Bob. When Bob asks Alice to do something at his behest, Alice attempts to build a shared context between the two of them respective to the action she has been asked to take. If Alice *believes* that her expectations have aligned with Bob's, she will trust him. If not, she won't. It is important to note that Alice's belief is what matters. This is how attacks can happen; Alice is tricked into believing that she has built a shared set of expectations with the sender. We discuss this further in Section 1.4.1.

It may seem that this action-centric model may leave out cases that involve Bob providing to Alice some information that may be of doubtful veracity. However, in these cases, we can say that Bob is tacitly asking Alice to take the potentially dangerous action of believing his information.[9]

1.3.2 Reliance

Over time, as process-based trust builds between Alice and Bob, their relationship becomes governed more by *reliance* than trust. Alice becomes less cognizant that she is making herself vulnerable when performing actions on Bob's behalf, because she expects him to behave in a trustworthy manner. [41] This sense of reliance becomes a part of her background expectations. Moreover, "Bob" does not have to be an individual, he can be an organization. Alice can develop reliance in Dartmouth College, for example. Alice may also develop reliance on entities that share some sort of attributes. Perhaps a string of positive experiences with PhD students in the Dartmouth Computer Science Department has led Alice to add to her background expectations a sense of reliance on people who possess a degree from that program.

1.3.3 Signals

We mentioned above that, when Alice is deciding whether or not to trust Bob, she "attempts to build a shared context" between herself and Bob. She certainly takes some information from the world around her and the situation at hand, but she must also get some sense of Bob, what properties he has, his state of being, his understanding of the situation, etc. We say that Bob *signals* this information to Alice. [98] In a standard trustworthy communication system, all the signals that Bob sends to Alice are focused on assuring her of his identity. We have demonstrated that it is necessary to signal a richer set of properties in order to support the kinds of use cases we see in the real world. The onus is upon us to ensure that we choose good signaling mechanisms, ones that are "cheap and easy to emit for trustworthy actors and costly or difficult for" attackers. [98]

1.3.4 Using this framework

Applying this framework to the myriad of methods in which humans interact through the Internet could be interesting. However, we believe it will be more instructive and useful to investigate to some concrete examples from a single application domain—email, since it has become the primary means of communication between humans in Internet settings. It is worthwhile to note, however, that the results here can generalize to any method of

[9] Portions of this section adapted from my thesis proposal.

digitally mediated person-to-person communication. Instant messaging, for instance, could be considered similarly to email. While it could be possible to extend our results to cases that involve users developing trust in non-human entities (web sites or other online services, for example), we do not consider those situations in this work.[10]

1.4 Requirements for a trustworthy email system

For the purposes of this work, we consider a *trustworthy email* to be a message that has the following properties[11]:

1. **integrity**: the message has not been undetectably tampered with in transit.
2. **sender authenticity**: the recipient can reliably determine that the purported sender of the message actually sent it.
3. **context**: the recipient can glean enough information from the message to build a shared set of expectations with the sender, as discussed in Section 1.3.

Standard email technologies, such as plain text and Multipurpose Internet Mail Extensions (MIME), provide none of these things, as the entire message can be forged. Nothing prevents an attacker from undetectably creating arbitrary header information (sender address, date, subject, etc.), attaching it to message of his choosing and sending it to any valid email address on the Internet. "Man in the Middle" attacks are also possible, in which an adversary can modify the content of a legitimate email while the message is in transit. Integrity can be achieved through the appropriate application of digital signatures, discussed in Section 1.1.1. By combining these signatures with some form of PKI, it is also possible to achieve sender authenticity. At this point, the message recipient knows the message has not been faked; Alice knows that she has received a message from some sender, and it has not been altered or replaced in transit. In the case where Alice and the sender already share a trust relationship, context and sender authenticity become conflated and boil down to a question of identity. If Alice knows Bob the sender, and what she trusts him to ask, then proving to Alice that a message came from Bob is enough context to bring their pre-existing process-based trust into play. In this case, the combination of digital signatures and PKI have bound Bob's public key to a piece of real-world information—his identity—and that is

[10] Portions of this section adapted from my thesis proposal.

[11] Issues of message secrecy and privacy are undeniably interesting, and could feasibly be included in a definition of trustworthy email. However, we consider these topics orthogonal to the issue of deciding trust as we have defined it.

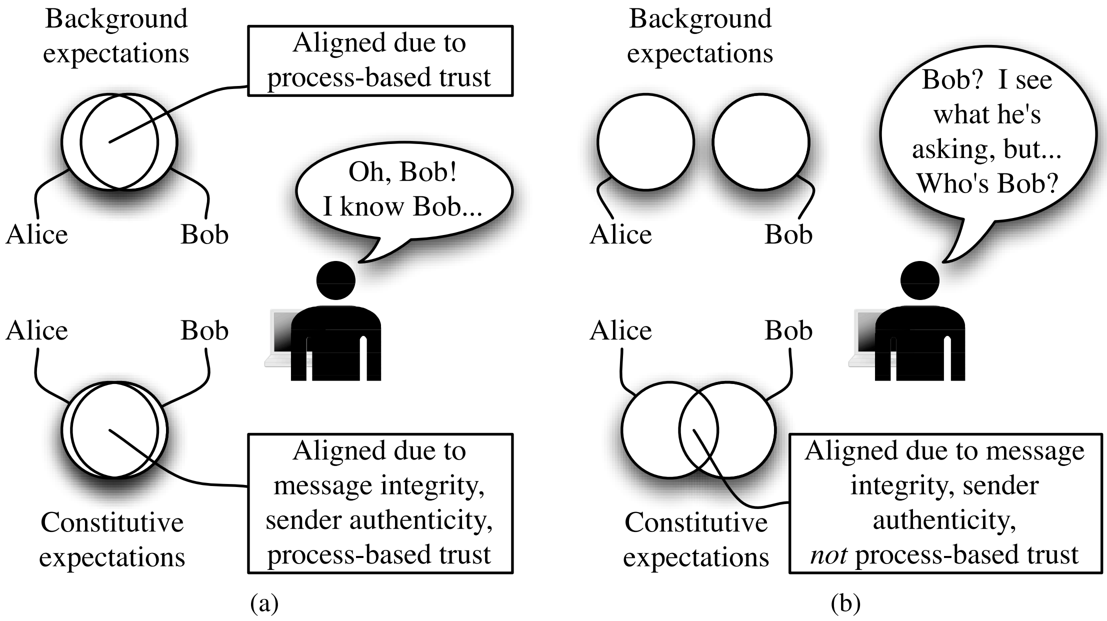

Background expectations
Aligned due to process-based trust
Oh, Bob! I know Bob...
Alice
Bob
Alice
Bob
Aligned due to message integrity, sender authenticity, process-based trust
Constitutive expectations
(a)
Background expectations
Bob? I see what he's asking, but... Who's Bob?
Alice
Bob
Alice
Bob
Aligned due to message integrity, sender authenticity, *not* process-based trust
Constitutive expectations
(b)

Figure 1.7: In (a), Bob and Alice share a trust relationship a priori. Assurance that this message is from Bob is enough context to allow Alice to determine that her background and constitutive expectations are in alignment with those of the sender. In (b), Alice and Bob share no relationship. She knows the message isn't fake, so she can establish some shared constitutive expectations with Bob, but she cannot determine whether the rest of their expectations are in alignment or not. No trust can be established in this case.

what Alice "needs to know" in order to make an appropriate trust decision[12] (Figure 1.7a). However, in cases where Alice and Bob do *not* share a process-based trust relationship a priori, such as the ones detailed in Section 1.2.1, identity ceases to be enough and context is more difficult to provide (Figure 1.7b). In these cases, the digital signature/PKI combination has once again bound Bob's key to a piece of real-world information, but it is not what Alice needs to know in order to draw the appropriate trust conclusion.

1.4.1 Attacks in a trustworthy email system

As we endeavor to enable users to trust each other in the kinds of cases discussed in the previous section, we must be careful not to concomitantly make users more vulnerable to attack. In terms of our trust framework, we can say that an attack is an attempt to convince the recipient of a message that a set of shared expectations has been established when, in fact, none exists. The attacker attempts to send false signals that ape those that might be sent by some trustworthy party. Consider the case of a simple password phishing email. Let's say Alice works at Dartmouth, which has a webmail system. Alice and Dartmouth share the background expectation that, from time to time, email will be sent out to inform users of technical issues at the College. Alice also has some background expectation that it is reasonable for Dartmouth's technical staff to ask for her login credentials; whether or not the *real* Dartmouth technical staff share this belief with Alice is immaterial. The attacker, obviously, wishes to prey upon this expectation. So, he sends an email that purports to be from some variant of "Dartmouth Computing Services" asking Alice to reply with her username and password so that some service can be provided for her webmail account—maintenance, for example.[13] The message expresses several things:

1. the sender shares Alice's background expectation that she should give up her password to Dartmouth's technical staff under some circumstances,
2. the sender represents Dartmouth's technical staff, and
3. the sender has a constitutive expectation that one of those circumstances has arisen.

Once Alice comes to believe that the sender does represent Dartmouth's technical staff, she is lost. There are many signals an attacker can send in order to convince her of this, most of

[12] provided, of course, that Alice was not confused (or tricked!) by any of the myriad of usability issues that inflict PKI-based systems. We address usability in Section 1.4.2.

[13] This example is based on a group of several phishing emails received by the Dartmouth community from March 2008 through June 2008. The first several times this occurred, enough users fell for the scam that technical staff chose to send out campus-wide warnings about the messages, in addition to taking the appropriate corrective actions. Even as recently as early June, a user's account was compromised in this way and used to send out spam.

which involve institutional trust. The attacker my include Dartmouth-specific iconography and terminology, or perhaps insert "Verified by Verisign" or "TRUSTe" badges. These signals are very cheap for attackers to emit, and so should not be relied upon, but have unfortunately been known to trick users in the past. [32] Alternatively, he can attempt to leverage process-based trust that Alice may share with Dartmouth's technical people by including the names of specific individuals in positions of influence. Our experiences in penetration testing and discussions with a security consulting firm [6] have shown that a lot of organizational information (especially about educational institutions) is accessible via the Internet. Thus, even outside attackers can falsely signal that they have a relationship with some trusted insider. Regardless, once Alice trusts that the sender speaks for the technical staff, the attacker has won. Her background expectations come into alignment with those of the attacker, and so she is willing to accept his assertion that it is time to send out her password. At that point, her constitutive expectations align with the attacker's and she is willing to reply with her password at his behest.

We have already discussed the need to provide the appropriate context for Alice as she attempts to make trust decisions. It is imperative that we choose good signals to carry this context, lest we provide attackers with another avenue by which to trick message recipients.

1.4.2 Usability: the last criterion

Many software engineers and system designers assert that security and usability are inherently at odds. The HCISEC community fundamentally disagrees with this point of view, believing that we cannot build systems that are actually secure when used by average users without aligning these two design goals. As Garfinkel puts it, "a system that is secure but not usable will not be used, while a system that is usable but not secure will eventually be hacked and rendered unusable" [47, p. 13].

This line of reasoning obviously applies to secure and trustworthy email systems. In fact, the piece of research that in many ways led to the evolution of HCISEC as a field of study unto itself was a usability evaluation of a PGP/MIME email client. [125, 127] Others have looked at usability issues in S/MIME [47, 54] and found it wanting as well. Many have gone on to propose and build academic prototypes that address these usability concerns, which we will discuss in Section 4.4, but they continue to focus on providing assurance of the identity of message senders. It is clear, however, that as we develop a more expressive system that is capable of handling the problems we have laid out thus far, we cannot lose sight of usability.

1.4.3 The Problem

Taking all this together, the problem becomes clear: build an email system that is sufficiently expressive to encompass the nuances of human trust decisions, without making users more vulnerable to attack or significantly negatively impacting usability. To summarize from the above, it must provide the following:

- **message integrity**: messages cannot be undetectably tampered with in transit.
- **sender authenticity**: the system provides assurance that the purported sender of a message actually sent it.
- **context**: the recipient can glean enough information from a message to build a shared set of expectations with the sender.
- **good signals**: the new signals used by the new system must be easy for trustworthy actors to emit and difficult or costly for attackers.
- **usability**: the system cannot introduce new usability challenges beyond those already present in existing trustworthy email systems.

1.5 ABUSE Overview

For the dual purposes of demonstrating my design philosophy and helping users manage trust in secure email, we introduce the *Attribute-Based, Usefully Secure Email (ABUSE)* system. The design, architecture and implementation of this system are discussed in Chapter 5 and Chapter 6. Rather than attempt to automatically make trust decisions for users, ABUSE is designed to help them make more well-informed trust decisions about email that they receive. The initial system is meant for deployment within a single logical domain that already has an identity PKI. The power grid, though many organizations operate and manage it, is managed collectively enough that ABUSE can still apply in that space. For example, NERC provides guidelines that specify names for and duties of roles at the different organizations that interact during crises. [83] (We provide thoughts about how to make ABUSE work across domain boundaries in Section 10.1.2) By building on top of a pre-existing secure email technology (we have chosen S/MIME) and leveraging this identity PKI, we get message integrity and sender authenticity almost for free. We focus, then, on providing recipients with the context they need to make trust decisions, without introducing new usability problems into the system.

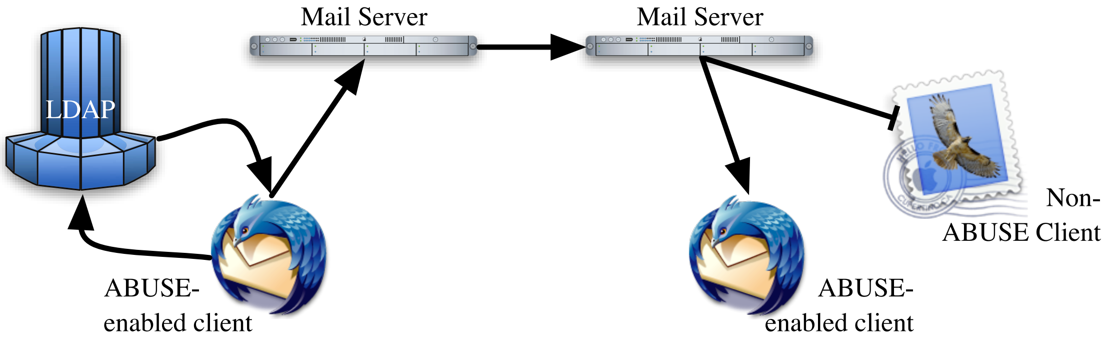


Figure 1.8: The architecture of the ABUSE system. Users make assertions about other people and leave them in the central attribute store to be picked up later. They can bind them to outgoing messages, and users with ABUSE-savvy clients will see these attributes along with the messages to which they are bound. Users without ABUSE-savvy clients see no extra information.

Instead of attempting to predetermine what kinds of context information users will need to build trust with each other, ABUSE allows users to collectively build up a set of non-repudiable assertions about each other. When Bob wishes to send a message to Alice, then, he selects some of his attributes, the ones that he believes will lead her to honor his request, and binds them to the message he signs and sends. On Alice's end, the system presents Bob's attributes to her along with his message, and she decides whether or not a shared set of expectations has been established that lead her to act on Bob's request.

In order for ABUSE to work, there must be a body of attributes for users to bind to their messages. As stated above, users create these assertions themselves. While it is true that there would need to be some bootstrapping of the attribute space at the organizational level, the strength of ABUSE is that users can use any attribute they already possess to sign off on an assertion about someone else. Properly, then, a single attribute is actually a chain of digitally signed assertions rooted at the trust root of the identity PKI that the organization already has in place.

For example, say Alice's organization has issued her an attribute stating that she is a Full-Time Employee. It is a chain of assertions with length one. Alice can take this assertion use it to sign off on some statement about Bob, perhaps saying that he is her intern. Bob winds up the subject of an attribute that is a chain of two assertions:

```
Organization → Alice (Full-Time Employee),
Alice (Full-Time Employee) → Bob (is my intern).
```

Alice creates this attribute in conjunction with a central attribute store via a protocol that we define in Chapter 5. Bob can then download his attributes from this store whenever he wishes and bind them to his outgoing messages.

This architecture is shown in Figure 1.8.

1.6 Contributions

In this work, we examine how to enable humans to build trust in third parties they do not know over digitally mediated communication channels. We are moving to a world where email and instant messaging are becoming widely used in professional settings. This has some advantages, like allowing people to handle multiple threads of conversation at the same time and providing superior auditing capabilities for organizations that require it. One drawback here is that it is much easier to impersonate a trusted individual in systems like these than in real life or over the phone. Traditional security measures are aimed at preventing these kinds of attacks. However, it is rare that the only way a human will take a risky action is on the say-so of one, particular individual. There are a variety of perfectly valid ways in which trust can flow from an authorizer with whom a user is familiar to a requester that she does not know. The first contribution we make is a discussion of a selection of these scenarios, and a characterization of what kinds of trust flows must be enabled by a system designed to enable a usefully secure communication system. The approach we take to building a usefully secure email system involves a standard identity PKI for key management with a user-managed, decentralized, non-hierarchical PKI grafted on to enable users to express their trust relationships with one another. ABUSE is the realization of this approach. The system and its various user interfaces are the second contribution, which we evaluated experimentally with user studies as well as analytically by applying Cranor's Humans-in-the-Loop framework [21] for examining secure systems that leverage users. In short, our contributions are these:

- an exploration of what it will take to build a system that allows human calculus-based trust to be migrated into the digital world,
- a set of criteria for a system that can facilitate this migration,
- ABUSE, a prototype of a system which meets these criteria,
- a discussion of ABUSE within an established framework for evaluating secure systems with humans in the loop [21],
- an experiment that uses the power grid setting to evaluate the ability of ABUSE to assist humans in building trust over digitally mediated communication channels, and
- a second experiment that qualitatively evaluates the expressive capabilities of ABUSE.

1.7 Thesis Outline

In Chapter 2, we examine cases from academia and the power grid in which people had to extend trust to unfamiliar third parties. By looking at how trust flowed among the individuals in these different real-world scenarios, we extract several different classes of "trust flows." These form the basis for some of the experiments we performed to evaluate ABUSE. Chapter 3 explores the required characteristics of a solution to the problem of trustworthy email. We move on to discuss work that has been done touching on this area in Chapter 4. Chapter 5 and Chapter 6 detail the architecture of the system we have built and the design of the user-facing portions of ABUSE. In Chapter 7 we analyze ABUSE according to Cranor's Human-in-the-loop framework [21] as a systematic way of discussing how it addresses the myriad of pitfalls that face secure systems which build humans in. After describing our experiments in Chapter 8 and Chapter 9, we provide concluding remarks and interesting directions for future work in Chapter 10. Detailed material about the experiments is available in Appendix A and Appendix B.

Chapter 2

Patterns of trust flow

In order to develop a system capable of helping users manage their trust in unfamiliar people with whom they communicate over the Internet, a system designer must first study how users build this trust in real-life scenarios. As discussed in Section 1.3, we began this process by defining trust as the willingness to undertake a dangerous action on the behalf of another party due to a set of shared expectations, and by coming to understand how humans build this context. Armed with this understanding, we sought real-life examples that we could examine to discover patterns in how sets of shared expectations are developed and aligned between people who do not already know each other.

We already had evidence of cases in the academic space [117] that required humans to trust unfamiliar people, but wanted to look beyond that to find situations that involved greater risk on the part of the relying parties. As a part of our work with the Trustworthy Cyber Infrastructure for the Power Grid (TCIP[1]) project, we discovered anecdotal evidence that operators in the power grid frequently exhibit the kind of behavior during emergencies. Furthermore, we were able to get access to transcripts of telephone calls made by power grid personnel during the August 2003 North American blackout [88] so that we could actually see this trust-building in action. From these transcripts and our academic scenarios, we were able to identify some patterns in the trust flows we saw and group the different situations accordingly. We do not argue that the groups we identify cover the set of all possible reasonable scenarios, but we do argue that, by identifying such patterns, we can identify whole classes of trust decisions that can be aided by ABUSE.

Before enumerating the grouping scheme we have created, we first introduce our notion of a *trust flow*, which will be helpful in our discussion. We believe classifying these sorts of scenarios in this manner to be novel.

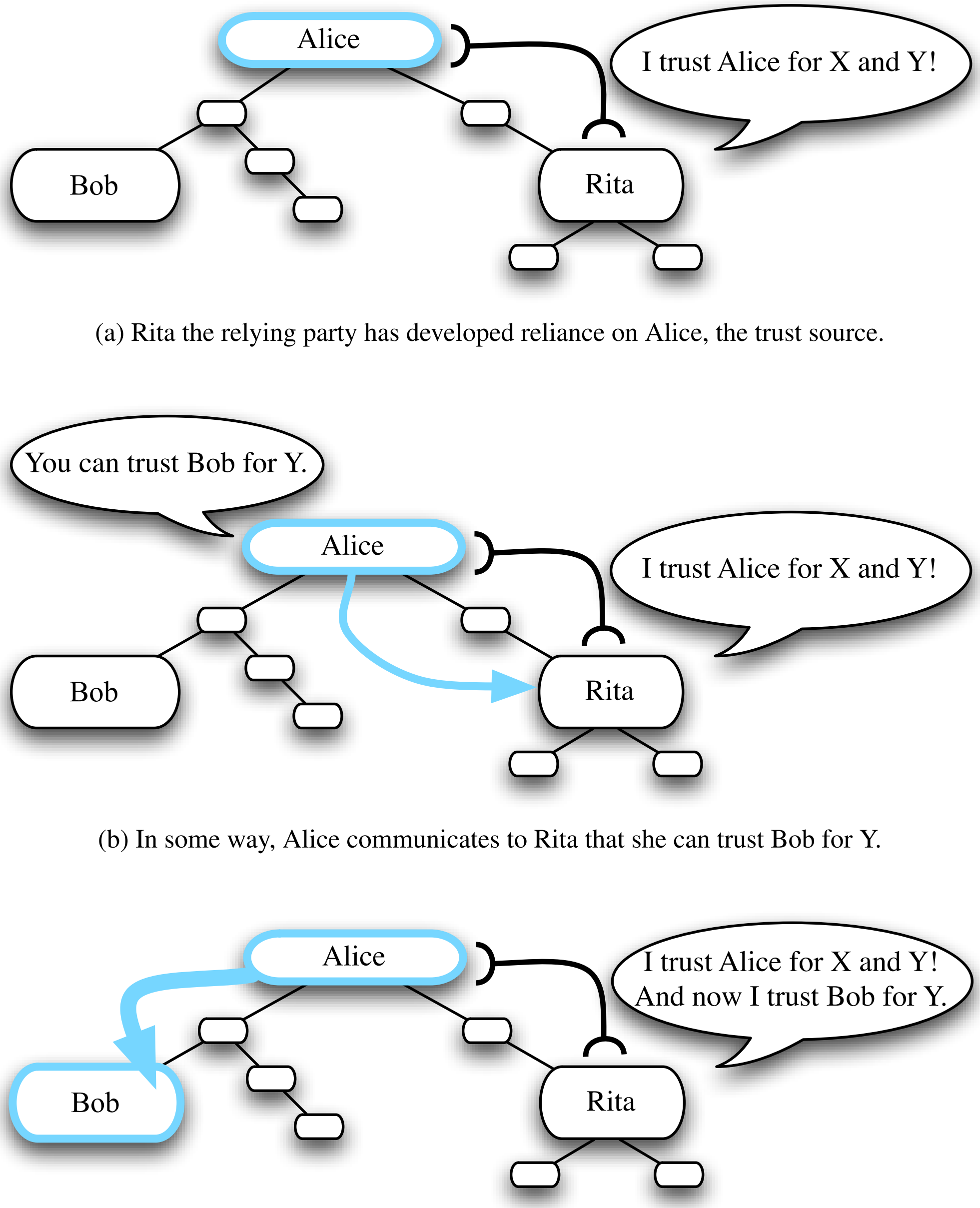
[1] `http://www.iti.uiuc.edu/tcip/index.html`

2.1 Trust flows

Referring back to our definition of trust from Section 1.3, we note that there are two principal participants involved in the decision to extend trust: the entity making a potentially dangerous request, and the entity deciding whether or not to act. We refer to these two as the *requester* and the *relying party*, respectively. In all the cases we discuss, we assume that the requester and the relying party are from the same domain, such as “power grid operators” or “academics”. Thus, they share at least some basic set of background expectations; academics and people who work with them all know what a “professor” is, for instance. We also assume that the requester and the relying party share no pre-existing trust relationship. In all the considered cases, there is always some *trust source*, an individual or entity with whom the relying party has a pre-existing relationship. In social science terms, we say the relying party has developed *reliance* on the trust source. The trust shared by these two is not all-or-nothing; a set of shared expectations governs that relationship as well. The relying party trusts the source for only a certain set of actions. Bob trusts Dartmouth to identify members of the College community, but not to designate people to check his email for him. Alice trusts Professor Smith to indicate which students are good to work with, but not which ones are funny.

In a given flow, communication from one or more trust sources begins a process that enables the relying party to develop a set of shared constitutive expectations with the eventual *trust sink*, the requester. This process may involve some intermediaries. Regardless, in a valid flow of trust, the expectations eventually shared by the relying party and the trust sink cannot expand beyond the expectations originally shared by the relying party and the trust source(s). Alice cannot validly wind up trusting that a person is funny based solely on expectations that flow from her relationship with Professor Smith, because her shared context with him does not lead her to believe that he is a good arbiter of humor. If she does, that would indicate that Alice has chosen to trust the sink for some purpose that she does not believe the source has the ability to speak about. Multiple trust sources only come into play when more than one source is necessary for trust to flow; consider the “two-man rule”, used to authenticate the launch of a nuclear weapon. [87] *Both* sources are needed to authorize the launch, so this is properly a multi-source flow—distinct from multiple single-source flows. In an email scenario, the requester might request multiple actions in a given message, and so multiple flows drawing from multiple sources might have to come into play for a single message to be completely trusted. Remember, though, that each flow applies only to the decision to trust the requester for a particular action.

Now, there are plenty of situations that seem to violate this stipulation. People fre-


Alice
I trust Alice for X and Y!
Bob
Rita
You can trust Bob for Y.
Alice
I trust Alice for X and Y!
Bob
Rita
Alice
I trust Alice for X and Y!
And now I trust Bob for Y.
Bob
Rita

(a) Rita the relying party has developed reliance on Alice, the trust source.

(b) In some way, Alice communicates to Rita that she can trust Bob for Y.

(c) Rita trusts Bob for Y, because Alice says it's ok. Rita's trust has flowed from Alice to Bob.

Figure 2.1: A very basic trust flow.

quently place trust in requesters that they should not. Sometimes, the relying party is simply mistaken; she believes that some expectation of hers is shared by the source when it is not. In attack scenarios, the relying party is being tricked into believing some shared expectations exist that do not. We discussed how this might happen back in Section 1.4.1 and don't consider it further in this chapter.

Given all of this, we can say that a valid *trust flow* exists between a source and a sink when communication initiated by the source helps the relying party to develop a set of shared constitutive expectations with the sink, leading to trust. A very basic trust flow is diagrammed in Figure 2.1.

2.2 Power Grid Background

As mentioned earlier, we found a number of motivating scenarios for our work through our involvement in the TCIP project. The power grid is a large, interconnected system, with pieces of the infrastructure owned by many different companies. Even under normal operation, "the system must be able to meet the continually changing load demand...Unlike other types of energy, electricity cannot be conveniently stored in sufficient quantities." [68, p. 8] Load (i.e. usage of electricity) is constantly in flux, and prone to unpredictable changes. Since there is no convenient way to store large quantities of excess electricity, generation assets must be available on short notice, and adequate transmission capacity must be available to get the energy where it needs to go. Companies often purchase generation and/or transmission capacity from each other to fill needs as demand shifts—especially during crises that knock out equipment in unanticipated ways. This necessitates constant coordination among the various parties involved in the grid, and emergency situations add a time-sensitive element to the process. Equipment can often handle operating outside of its rating for short periods of time when the grid is in a state of emergency, but leaving things in this state for too long can cause the grid to become *in extremis*, at which point cascading outages will occur and large portions of the grid may shut down. [68, p. 11] This is exactly what occurred during the August 2003 North American blackout.

2.2.1 The blackout

On August 14th, 2003, some high-voltage transmission lines in northern Ohio sagged too close to some ill-trimmed trees, arced, and shut down. This event led to an outage that cascaded through the state, into Michigan, crossed the border into Canada, and eventually brought down much of New York as well. The events of this day, the preconditions that set the stage for the blackout, and the factors that contributed to the continuing escalation of the problems are detailed in a 238 page report put out by the U.S.-Canada Power System Outage Task Force. [121] A detailed exploration of these events would be irrelevant to our discussion, but it is worth noting that the task force lamented a lack of "effective internal communications procedures" and a lack of "joint procedures...on when and how to coordinate" the mitigation of a particular kind of infrastructure failure. [121, p. 19] It is also worth noting that much of the cross-organizational coordination during the blackout took place over the phone, much of which was recorded at Midwest ISO (MISO). We discuss the organizational structure of the grid in the next section, but suffice to say that MISO's function is to mediate among many companies that run power generation and transmission infrastructure. So, they were a central point of contact during this crisis. In all, conversations from seven different phones at MISO were recorded, resulting in over 550 pages of transcripts that were eventually submitted as evidence during a congressional hearing on the blackout. [88]

Having heard from our partners on the TCIP project that these transcripts contained evidence of grid technicians and operators leveraging informal trust connections in order to try to stave off outages, we began culling through the calls to try to reconstruct the particular scenarios. In order to understand what we were reading, it was first necessary to develop an understanding of the different entities involved in grid management, the roles of people within those entities, and the relationships among them. Also, knowing the kinds of risks the people in these situations might have to take, and the potential consequences to them, would enable us to better evaluate which situations were actually potentially costly. That understanding is detailed here.

2.2.2 Power Grid Organizations

There are two classes of management organizations in the US power infrastructure: *operation companies* and *Regional Transmission Organizations (RTOs)*. Operation companies own generators and/or power lines and directly control these pieces of equipment. RTOs mediate among different operation companies; help maintain stability and reliability in the grid; and, to serve these ends, can request that operation companies in their area take steps they deem necessary to address problems. In dire situations, an RTO can order an operation company to take a particular action, though we noted a preference among personnel at RTOs to avoid using this authority unless absolutely necessary. [88, pp. 150–152, 221–227] Overseeing all of this are the North American Electric Reliability Corporation (NERC), an industry body, and the Federal Energy Regulatory Commission (FERC), which is an in-

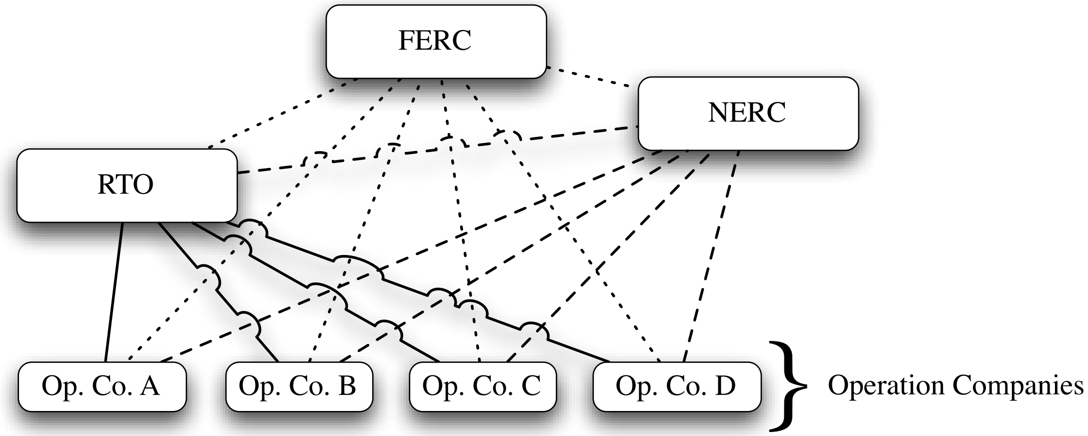


Figure 2.2: The relationships between RTOs, operation companies, NERC and FERC. NERC provides regulatory oversight, while RTOs have a more direct managerial relationship with operation companies during crises.

dependent non-industry and non-governmental organization. Neither generally takes an active role in crises, but instead create and enforce regulations designed to promote stability in the grid. As NERC is an industry body, FERC provides oversight of their behavior as well, though it also can investigate and provide regulatory guidance for RTOs and operation companies as well. The relationships are presented in Figure 2.2.

In the examples discussed in Section 2.3, we refer to two RTOs, MISO and PJM Interconnect. We also refer to several operation companies, including IP&L, Cinergy, Allegheny, AEP, and First Energy.

2.2.3 Power Jobs

There are a wide variety of jobs that employees may possess at a power company or grid management organization. The set that may be involved in coping with an emergency is more limited, and is standardized (along with some other jobs/roles) by NERC. [83] For the purposes of this discussion, there is only one RTO role that really matters, that of a *Reliability Coordinator*. The Reliability Coordinator works with generation and transmission companies to mitigate local problems in addition to cooperating with other Reliability Coordinators—at her own RTO as well as others—during wider-scale crises. There are also several jobs at operation companies that are relevant:

- **Transmission System Operator** - controls her company's power lines and other transmission infrastructure. May interface directly with Reliability Coordinators in some instances.

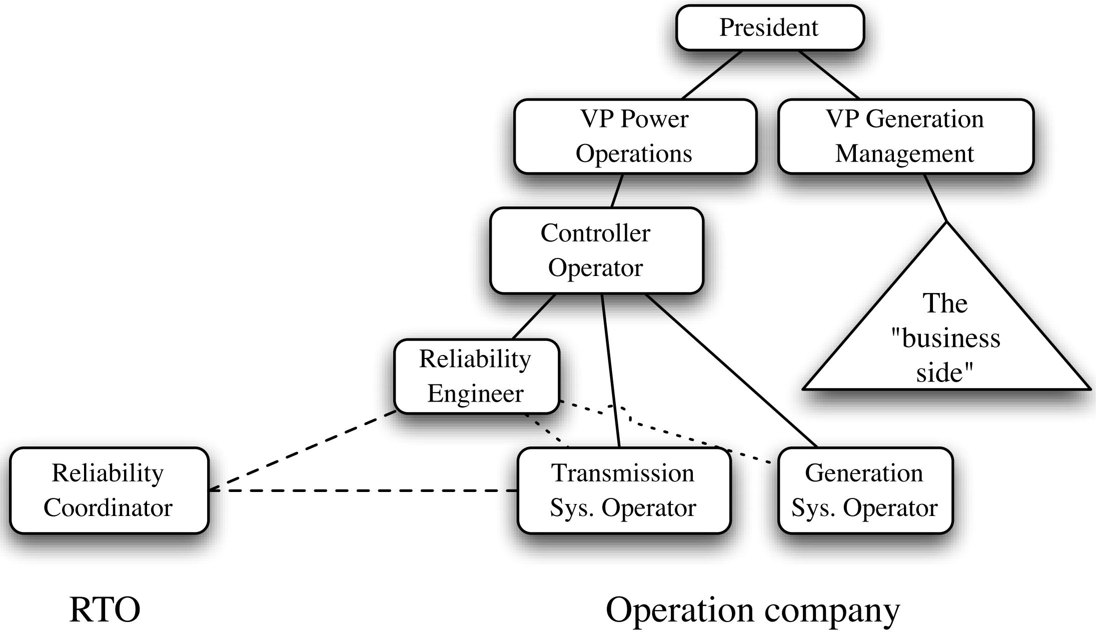


Figure 2.3: The relationships between relevant positions at operation companies and reliability coordinators at RTOs.

- **Generator System Operator** - controls his company's power generators, if they have any; some companies have transmission assets only.
- **Reliability Engineer** - works with Generator and/or Transmission System Operators at the company to maintain stability in the portion of the grid over which the company has control. Interfaces with Reliability Coordinators at RTOs during wider-scale problems.
- **Controller Operator** - manages Generator and Transmission System Operators at an operation company. They are his direct-reports, whereas a reliability engineer would work with them only during problem situations.

The relationships among these jobs is shown in Figure 2.3. Knowing now the roles of the players, we can discuss by example the patterns of trust flow that we have identified in power grid scenarios.

2.3 Trust flow patterns

Here, we introduce the trust flow patterns that we have identified from real-life scenarios:

- Role-based trust;

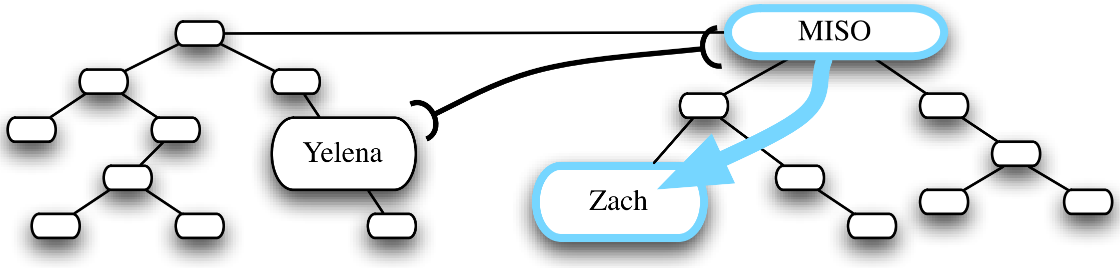


Figure 2.4: A role-based trust flow. The relying party and the source, an institution, share a pre-existing trust relationship. Trust flows directly to the requester.

- Simple delegation, which has some sub-classes;
- Coopetition; and
- Non-contemporaneous trust.

To the best of our knowledge, we are the first to codify different trust-building scenarios in this way, according to our shared-expectations model of calculus based trust and reliance (presented in Section 1.3). As we cover each class of trust-flows, we explicate how each flow develops in terms of our trust model, using real-world examples. We make use of these patterns in the experiments discussed in Chapter 8 and Chapter 9.

2.3.1 Role-based trust

Role-based trust is the simplest pattern of trust flow. The requester can be anywhere in the trust infrastructure. Trust is institutional; a trusted authority has asserted that the requester is in a given role. In this scenario, the trusted authority is the trust source. The relying party understands the meaning of the role; thus, background expectations are shared among the two individuals. Constitutive expectations are set up by the message sent by the requester. If the expectations expressed in the message align with what the relying party expects of someone in the claimed role, trust is built and action can reasonably be taken. This pattern of trust-flow is well-understood in both computer science [104] and human arenas. It can be seen throughout the blackout phone transcripts. A particular instance involves individuals at MISO and IP&L [88, p. 426]. Zach at MISO contacts Yelena, a transmission operator at IP&L. Yelena believes Zach to be in a role at MISO that is allowed to make the stated request, and so she satisfies it. MISO, then is the trust source, the Zach is the sink, and Yelena is the relying party.

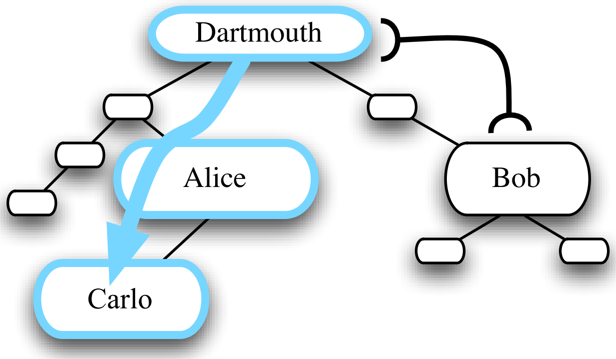


Figure 2.5: A role-based delegation trust flow. The relying party and the source, an institution, share a pre-existing trust relationship. Trust flows to the authorizer as a result of her role. The trust continues to the requester as a result of a collective understanding of the relationship between his role and that of the authorizer.

2.3.2 Simple delegation

In a simple delegation scenario, some individual that the relying party trusts (an *authorizer*) has initiated a flow to the requester. The reason that the relying party trusts the authorizer is irrelevant. Perhaps they share process-based or characteristic-based trust, making the authorizer and the source the same entity. Or, there may be a role-based trust flow from some other source to the authorizer. In any case, the authorizer is trusted by the relying party. The authorizer expresses to the relying party the constitutive expectation that he has transferred some permission of his to the requester. If background expectations, shared by all the parties involved, include the concept of delegation, then the relying party's constitutive expectations align with those of the authorizer and she becomes willing to trust the requester if he exercises that permission.

There are several slightly different styles of simple delegation, drawn from both the power grid scenarios and academic scenarios. We enumerate them here.

Role-based delegation

In the mail-from-the-dean scenario we discussed first in [117] and mentioned in Section 1.2.1, Bob the relying party trusts Alice the Dean because he knows that the College has granted to Alice the "Dean" role. Bob, then, uses role-based trust to develop a shared set of expectations with Alice the Dean. Alice has an assistant, Carlo, whose role is a descendant of hers in the tree. When Carlo sends out mail on Alice's behalf, Bob must decide whether to take the message as though it was from the Dean herself. All three share an understanding of the roles Alice and Carlo inhabit, and the relationship between them, so

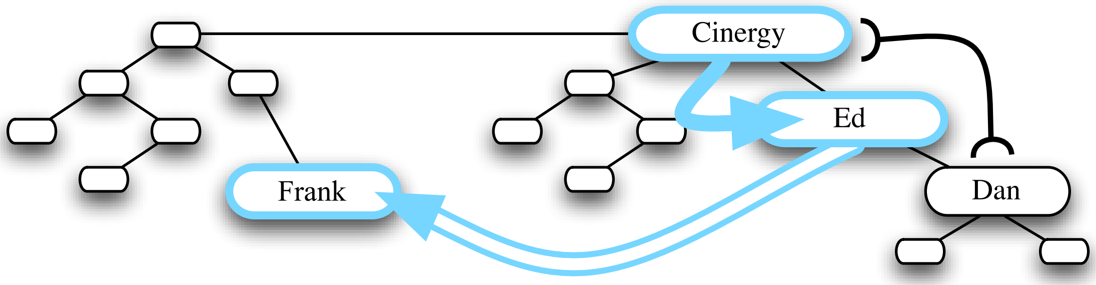


Figure 2.6: A role-sourced, arbitrary delegation trust flow. The relying party and the source, an institution, share a pre-existing trust relationship. Trust flows to the authorizer as a result of her role. She delegates some specific subset of the actions she is trusted to authorize to the requester.

Bob has enough of a shared set of background expectations with Alice to decide to trust Carlo. Here, the College is the source, Alice is the authorizer, Carlo is the sink and Bob is the relying party. This is shown in the general case in Figure 2.5.

Role-sourced arbitrary delegation

In this style of delegation, the relying party again trusts the authorizer due to role-based trust. The difference is that delegation is explicit, rather than requiring the relying party to understand the flow of trust due to a shared understanding of inter-role relationships (Figure 2.6). Again, we adapt an example from the power space [88, pp. 236–238]. The relying party is a generation systems operator at Cinergy, Dan. There is a controller operator, Ed, at his company that is not physically at the same facility as Dan. A reliability coordinator, Frank, at MISO is in contact with Ed, and they decide that they need Dan to make some change. Frank contacts Dan, with Ed on the phone, and leverages Ed's authority to get Dan to make the required change. Here, Dan trusts Ed because Ed is in a superior role at the same company. In this case, Ed sets up a constitutive expectation with Dan that Frank is allowed to request this particular change. Frank's request, then, is aligned with the collective set of shared constitutive expectations. They already share the same background expectations about their given roles, the roles of their companies when attending to a crisis, that it makes sense for a reliability coordinator to be collaborating with a controller operator to figure out what to do about some stability problem, etc. So, Cinergy is the source, Ed is the authorizer, Frank the sink and Dan is the relying party.

Friend-sourced arbitrary delegation

Here, the relying party trusts the authorizer because they share process-based trust a priori. They are friends, or at least acquaintances who trust each other on some level (Figure 2.7).

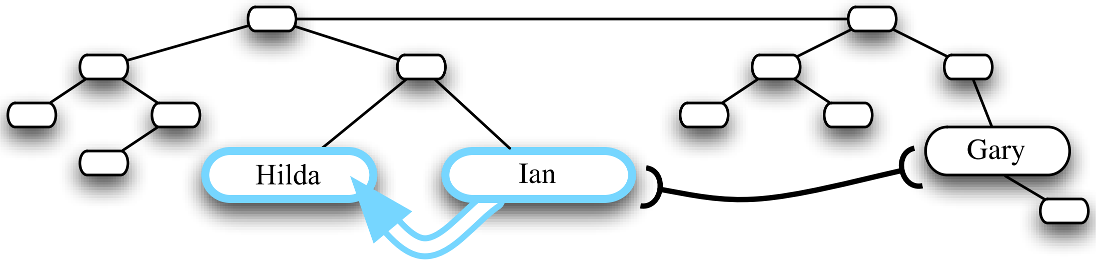


Figure 2.7: A friend-sourced, arbitrary delegation trust flow. The relying party and the source share a pre-existing trust relationship. The authorizer and the source are the same person. He delegates some specific subset of the actions she is trusted to authorize to the requester.

Delegation is, again, explicit. From the power grid, we have an example in which Gary at MISO would not trust Hilda at Allegheny without the go-ahead of his former co-worker Ian, also at Allegheny. Ian does not outrank Hilda, but Gary is willing to trust him due to their pre-existing relationship [88, pp. 56–58]. Ian is both the source AND the authorizer, Hilda is the requester, and Gary the relying party.

It is also possible for there to be trust flows that involve re-delegation, in which one or more intermediaries stand between the authorizer and the sink. In order for this to work, acceptance of the level of re-delegation happening must be either a part of the background expectations shared by the people involved, or the authorizer must make it clear as a part of the constitutive expectations that he communicates to the relying party that he allowed for re-delegation to occur.

2.3.3 Coopetition

Coopetition is a variant of delegation in that the relying party and the requester have a pre-existing institutional disinclination to trust each other prior to their attempt to build a set of shared expectations. The relying party initially expects the requester to act in a way counter to the relying party's best interest. As such, in order for the requester and the relying party to build trust, the background expectation of competition must be superseded by a constitutive expectation of cooperation. We see this in the power grid, where two companies compete for customers but must also cooperate to keep the infrastructure running along smoothly so that they can do business at all. One example of this involves two operation companies with their attendant RTOs: First Energy, an operation company overseen by MISO; and AEP, an operation company overseen by PJM Interconnect [88, pp. 150–153 , pp. 219–227].

John at AEP is concerned about a set of power lines that tie his company's area of

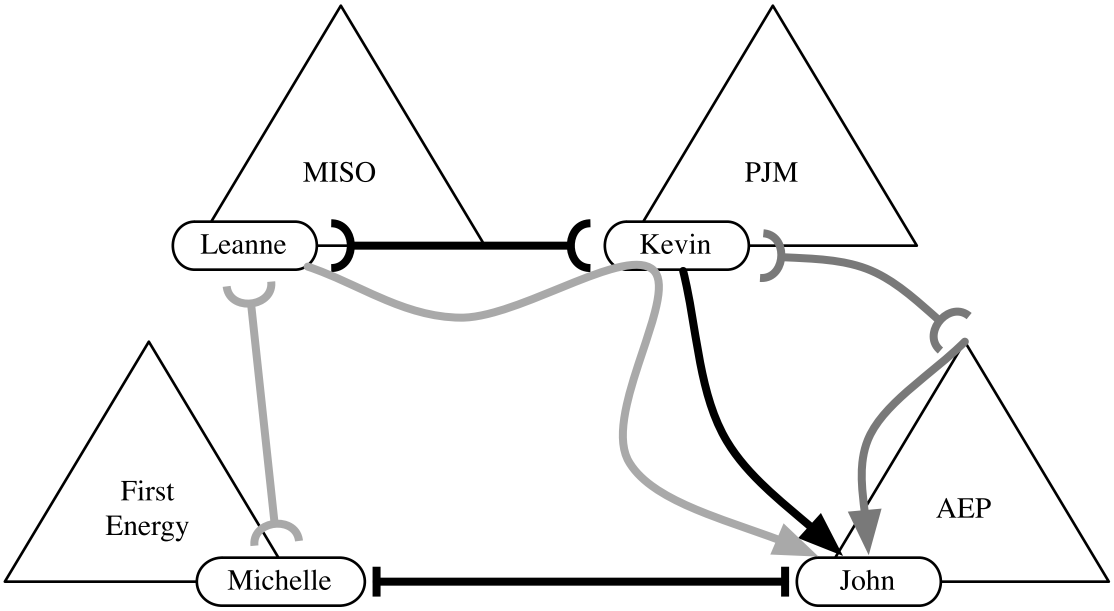


Figure 2.8: The coopetitive scenario laid out in Section 2.3.3.

control to that of First Energy. Some have overloaded already, and the others are being stressed by First Energy's behavior. John escalates up the chain to a reliability coordinator Kevin at PJM, who contacts Leanne, a reliability coordinator at MISO. Leanne then contacts Michelle, a transmission operator down at First Energy, to tell her what actions John is asking her to take. There are a number of flows occurring here, all shown in Figure 2.8. First, John builds role-based trust with Kevin so that Kevin will accept his information. Then, in order for Leanne to trust the information coming from John, there has to be some form of arbitrary delegation flow in which Leanne's trust in Kevin moves to encompass John. It is impossible to tell from the phone transcripts whether this flow is role-sourced or friend-sourced, unfortunately. Once Leanne has made this trust decision, she can initiate a coopetition flow down to John. She becomes the authorizer; Michelle trusts her for role- or process-based reasons. Keith is a re-delegatory intermediary, and John is the requester. The re-delegation is acceptable in this flow because all the parties understand the relationships among MISO, PJM, and the two operation companies. John's information expresses a constitutive expectation that his requested action makes things better for both an First Energy, which has to supersede the expectation that they shared before: that their companies would not act in each other's best interest. Michelle's expectations align with John's only because of the trust flowing down to him from Leanne.

A coopetition flow does not need to be as complex as this. A requester at a competing entity may be the sink of some arbitrary-delegation-based trust flow, as seen in Figure 2.9. The key is the need to develop a shared constitutive expectation of mutual benefit that

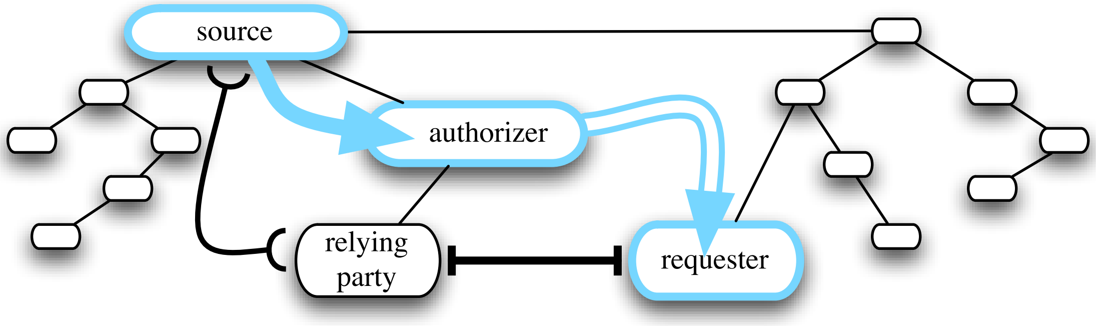


Figure 2.9: An idealized coopetitive trust flow. The relying party and the source share a pre-existing trust relationship. Initially, the requester is distrusted by the relying party; they share a negative trust relationship. Trust either flows to the authorizer due to her role, though she and the source might be the same entity. She delegates some specific subset of the actions she is trusted to authorize to the requester. The background expectations shared among all parties indicate that trust can flow if the constitutive expectations expressed by the authorizer indicate mutual benefit for the relying party and the requester.

overrides the existing background expectation of competition.

2.3.4 Non-contemporaneous trust

The flows discussed thus far involve the requester being the sink of a trust flow at the time he makes a request. However, it is also possible for a trust flow that is not currently active to enable the relying party and the requester to align their expectations anyhow. Consider the case of a colleague, who was still working at his old job while preparing for his move to Dartmouth to take a professorship. [117] He needed to convince a textbook publisher to send him answer keys for the book he would be teaching from once he began his appointment. Both parties shared the background expectation that a professor should get the answer key; both shared the expectation that a person who is about to enter that role should get the answer key. The issue surrounds getting the two to share the constitutive expectation that the requester falls into that group, that there *is going to be* a role-based flow from Dartmouth to the putative professor.

Non-contemporaneous trust can also refer to cases where there *used to be* a trust flow in place, and that should lead the relying party to align their expectations with those of the requester. To take an example from the life of the author, Chris, consider the case of authorizing new users of Dartmouth's recording studio. The Music Department controls access to this resource, and new students come to them asking to use it. The author Chris used to manage the studio, teach classes in how to use the equipment, and keep track of who was allowed in. During that time, there was a clear role-based flow to Chris. After resigning this position, the department continued to trust him to authorize new users because he retained the relevant knowledge and expertise, and was still qualified to screen new requesters for competency. Despite there not being a valid flow any longer, the department continued to trust him to perform these actions.

Any of the types of flows listed in this chapter can be used in non-contemporaneous fashion.

2.4 Conclusions

We have presented the concept of a *trust flow* within the shared-expectations model of trust that we discussed in Section 1.3. We have identified a number of patterns of trust flows that we saw repeated in several different arenas; primarily, crisis management in the power grid. The patterns are:

- role-based trust,
- several varieties of simple delegation,
- coopetition, and
- non-contemporaneous trust.

We can now discuss the design of ABUSE and the ways in which it is built to enable these different patterns of trust flow.

Chapter 3

Characteristics of a Solution

Thus far, we have identified and motivated the need for a system that enables humans to leverage their current, informal methods of building trust with unfamiliar individuals when communicating over digitally mediated channels. Though such a system would be useful in a variety of domains, the power grid scenarios we have discussed provide a compelling real-world case for our work. We have chosen email as the application for which to develop a solution, because there is good support for signed digital messaging in that space, and current technology provides some features that provide a strong jumping-off point. So, the problem then becomes an issue of integrating, into an email infrastructure, technology that enables humans to appropriately extend trust to the right message senders without making them more vulnerable to attacks. First, we consider three different approaches we could take to this problem: *algorithmic*, *heuristic*, and *user-centric*. We then discuss a set of patterns with which we should design our system in order to make it both usable and secure. By the end of this chapter, we will show that we can solve the problem we have set forth by creating a system with the following characteristics:

- the system must be expressive enough to provide contextual support for the classes of trust flows enumerated in Chapter 2,
- the new signals used to carry this information must be good (as defined in Section 1.3.3).
- the system must be designed to be usable according to guidelines from the HCISEC community.

3.1 Possible approaches

In Section 1.4.3, we laid out five features that the system we design must provide: message integrity, sender authenticity, context, good signals, and usability. As we are working on an email-based system, we can build on top of an existing technology that provides message integrity and sender authenticity, leaving only the issue of providing context, good signals and usability. The challenge lies in creating a system expressive enough to provide contextual support for all the flows discussed in Chapter 2 without exposing the user to a greater risk of attack or reducing the usability of his secure email client. We believe this allows users to make accurate trust decisions without needing to waste time trying to build up the necessary context through out-of-band channels—*and* decreases the likelihood that they will simply skip that step and make an ill-informed decision (we evaluate this belief empirically in Chapter 8). We identify three different directions from which we could attack this problem:

1. **automatic**:
   (a) **algorithmic**: the system tells the user which messages he can trust, and which he should disregard.
   (b) **heuristic**: the system estimates, according to some statistical profile of what messages are trustworthy, which messages the user should trust.
2. **user-centric**: the system provides a channel for appropriate context to get to the user, and the user decides which messages to trust.

Algorithmic and heuristic approaches are *automatic*, in that they attempt to make "the right decision" for the user. From a user's point of view, "the right decision" is "the trust decision the I would make, if I had all the pertinent and available information." Pertinent information could include both information about the sender (not only name and email address, but perhaps job title, project-team or group membership, position within organization, and more) as well as information about the message (whether the sender is asking for information, asking recipient to perform some action, giving recipient information, etc.). Each of the approaches we mentioned runs into different challenges when trying to help a user reach the right decision. We discuss each approach in turn.

3.1.1 Algorithmically deciding the email trust problem

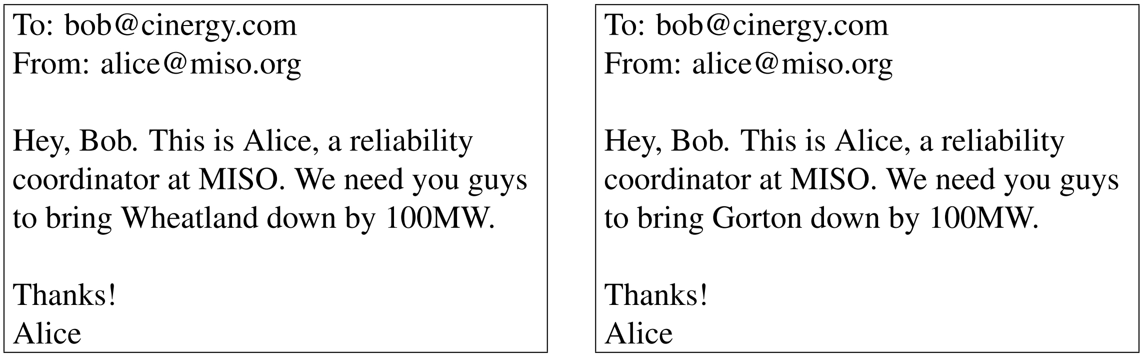
One way to enable users to make better trust decisions about incoming email is to build a system that makes decisions for the users. In this section, we consider systems that use some kind of pre-specified definition of what constitutes an allowable request, as opposed to adaptive approaches discussed in Section 3.1.2. If such an automated system could make the “right” decision every time, this would clearly be the best choice. However, there are security processes that do not lend themselves to automation. Some, for instance, “rely on human knowledge that is currently difficult for a computer to reason about or process.” [21] We consider now whether the email trust problem we have advanced is one of these cases.

To enable an automated system to decide which messages could be trusted, a user would have to be able to define a policy stating which kinds of senders are trusted to make which kinds of requests. Since email messages come in as text, this system would need to be able to reliably comprehend arbitrary text from authors it has never encountered—which is not, currently, feasible. [22] Even if this problem could be surmounted, the user would still need to be able to sit down and enumerate this policy. It is informative to consider the parallels with the Platform for Privacy Preferences (P3P). [20]

P3P is a system designed to enable users to manage the privacy of their information on the web. The idea is that websites specify their privacy policies in some language that a P3P client can parse and comprehend. Users specify *a priori* which kinds of privacy policies they are comfortable with, and the P3P client built into their browser tells them to which sites they can release their private information. To specify this policy, users would need to dictate, for every kind of private information, all the kinds of websites that are allowed to receive it. This is analogous to the problem faced in the case of deciding email trust with which we are concerned. In discussing the issue of P3P policy creation, Ackerman states that “even a cursory examination shows...an ill-formed, intractable problem...users must be able to handle essentially an infinite information space.” [1] He also points out, however, that people decide to whom to release private information every day, in a “nuanced and seamless manner,” despite constantly re-categorizing individuals in their mental trust model and the consistent appearance of exceptional cases. In short, it is not that users don’t have trust policies in their minds, merely that they are incapable of effectively enumerating them in a machine-comprehensible format. [21]

Other researchers extend the points that Ackerman has made, asserting that this issue is not limited to P3P. Secure systems which require the kind of flexibility made necessary by our desire to accommodate the trust flows of Chapter 2 do not lend themselves to approaches that require users to specify complex policies. [34,41] So, for the problem we are considering, asking users to input their own policies into a system is not a usable solution.

In an enterprise setting, one might consider domain policies, written by local administrators. System designers like us lack the specific knowledge required to craft policies appropriate for all domains, but perhaps administrators embedded in an enterprise could



To: bob@cinergy.com
From: alice@miso.org

Hey, Bob. This is Alice, a reliability coordinator at MISO. We need you guys to bring Wheatland down by 100MW.

Thanks!
Alice

(a) A message from Alice, a reliability coordinator at MISO.

To: bob@cinergy.com
From: alice@miso.org

Hey, Bob. This is Alice, a reliability coordinator at MISO. We need you guys to bring Gorton down by 100MW.

Thanks!
Alice

(b) Another message from Alice, perhaps sent in error.

Figure 3.1: Two extremely similar messages that our system should help Bob, a reliability engineer at Cinergy, differentiate between. Gorton is not a Cinergy facility. Can a junk filter tell these apart? Even if Bob would never honor the second request, if the system tells him messages like this are trustworthy, Bob will cease considering it reliable.

craft appropriate policies. The challenges are similar to those that apply to the single user case, but now these administrators are charged with defining policies that apply to broad classes of users, and also making sure that they take into account different sets of circumstances. Even if accurate policies can be generated for the organization at a given point in time, the maintenance of these policies over time is extremely costly and challenging, if it is even possible. [114] Moreover, even disregarding the issue of maintaining policy correctness as an organization evolves over time, it is not clear that the correct policy for everyday usage is still correct in crisis situations or other exceptional circumstances. Furthermore, even if policy correctness can be achieved, we still run into the need to convert email text into a format on which the policy checker could run. As we mentioned above, for the kinds of communication with which we are concerned (between user who share no trust relationship *a priori*), this is currently infeasible.

The foregoing has not stopped researchers from attempting to build policy-based systems to address trust issues in the digital space. We discuss them in Section 4.2.1.

3.1.2 Heuristically deciding the email trust problem

Given that a completely automated approach to our problem will not work, one might consider a junk-filter-style heuristic approach. Generally this approach takes in (or builds) a statistical model of "good messages" and then determines how similar to this model a new message appears to be. We are concerned with messages from unfamiliar senders in exceptional situations; while a heuristic approach may be good at flagging messages that exploit trust flows similar to ones that users have seen before, or are requests from familiar senders, these are the cases in which users need the least help. Furthermore, exceptional situations (power grid emergencies, for example) necessitate the usage of trust flows different than those used during day-to-day tasks. A heuristic system trained during normal operation is unlikely to accurately identify messages that, in extraordinary circumstances, would be reasonable for the user to trust.

To definitively analyze the efficacy of a heuristic approach, we would need a large volume of email among unfamiliar communicants that exercise the kinds of trust flows we wish to enable. We have no such data for the domains from which draw our concrete examples. Furthermore, especially in the power grid, no such data is likely to become available. However, looking at the example in Figure 3.1, it seems intuitively unlikely that a junk filter would prove reliable. This fundamental characteristic of heuristic systems creates a usability problem, and would likely lead to users ignoring or disabling the system.

3.1.3 Taking a user-centric approach

It seems clear that fully-automated approaches will not work for the problem we wish to solve. Edwards et al. suggest that more security systems should be designed to "allow their users to make more informed decisions about what the correct or incorrect choice of action may be." [34] Given Ackerman's point about users' ability to make decisions about who to trust in real life, it makes sense to leverage this ability as a part of a system that works to address the email trust issue. Thus, our goal with ABUSE becomes to allow senders to signal trust flow information to recipients in a reliable and non-spoofable fashion without impacting the usability of secure email. Usability challenges remain, but they are of the variety that we can get a handle on using design principles advanced by the HCISEC community.

3.2 Design criteria for usably secure software

We have admitted that taking a user-centric approach in ABUSE will leave us with some usability hurdles. However, we also believe that the HCISEC community has provided us with tools to help us through these problems as we build our system. In [47], Garfinkel surveys the HCISEC space, collects design principles advanced by a variety of researchers, and lays out a set of design patterns for usably secure software. Some apply across the board, and some are more specific to issues in secure and trustworthy digital messaging. We discuss the sources from which Garfinkel drew his inspiration in Section 4.4 and enumerate his applicable design patterns here.

3.2.1 General patterns

Garfinkel provides six design patterns that he applies generally to usably secure software. [47, pp. 320–322]

Least surprise/least astonishment [47, p. 320]

Garfinkel traces the Principle of Least Surprise to Saltzer and Schroeder's concept of *psychological acceptability*. [103] While the provenance of the term itself is unknown [47, p. 7], the idea is that systems should behave as the user expects them to. Or, more precisely, in order for a system to be usable, the user's mental model of how the system works must be aligned with the system's behavior. Traditionally, any mismatch here has been considered a problem of user education; if the user doesn't understand how the system works, train him until he does. HCISEC takes a different tack: ensure that your system effectively communicates its behavior to the user via its user interface. This work can involve analyzing the problem at hand using a model of human behavior, and should certainly include an iterative software design process [85] that focuses on understanding the flaws in the current version from the user's point of view. [137]

Good security now [47, p. 320]

This principle embodies Voltaire's notion that "the perfect is the enemy of the good." Lampson refers to the same idea as "striv[ing] to avoid disaster rather than to attain an optimum." [69] There is a bias in security circles against deploying systems that are imperfect. The rationale is that users will assume that these systems are perfect and get into trouble when their assumption turns out to be false. The problem with that notion is that deploying no solution does not stop users from engaging in risky behavior. We discussed some cases back in Section 1.2.1. Garfinkel points to the delay of systems that leverage public key cryptography. Widespread deployment was held off until the availability of keys that were certified by third parties could be assured. [47, Chapters 5, 6] argue that systems without keys that are so certified can, in practice, provide security and privacy guarantees that are very similar to the systems we wound up deploying.

Provide standardized security polices [47, p. 321]

As we explained above in Section 3.1.1 and Garfinkel discusses in [47, Section 9.4], security policy "construction kits" are not usable. This principle asserts that it is better to provide a simple set of not-quite-perfect security policies from which users can choose than to provide them with an overwhelming plethora of options from which to build their own.

Consistent meaningful vocabulary [47, p. 321]

When building a new system, designers must strive not to overload terminology. Digital signatures for email are a case in point. Users were already familiar with "email signatures," the chunks of text their clients could append to every outgoing message. This clash of terminology muddies any discussion of signed email with non-savvy users. Garfinkel details a wider range of problems in [47, Section 8.2] and there is at least one book devoted to the issue as well. [5] This principle also extends to symbols. If many other applications use an exclamation point inside a triangle as an indicator of an informational warning message, a new application should not use a similar icon to signify anything else.

Consistent controls and placement [47, p. 322]

Similar to issues with vocabulary and symbol usage, this principle focuses on issues surrounding interface elements that actually activate functionality. If the application being designed shares some functionality with other software with which users may be familiar, the design of this other software should be taken into account. While he acknowledges that this is difficult in application spaces that are already populated by many competing products, Garfinkel points out that—fundamentally—the questions here are no different than those faced by protocol standardization efforts.

No external burden [47, p. 322]

Frequently, new security systems (especially those designed by academic researchers) are designed without consideration of the social context in which the technology must be deployed. A system that does not interact well with existing technologies can impose a usability burden not only on the user, but on those around him. In some cases, the text of S/MIME signed messages cannot even be read by users with non-S/MIME compliant email clients.[1] Such issues can cause *push-back*; a user's associates provide negative feedback about his use of the system, making him less likely to continue usage. For this reason, systems like Domain-Key Identified Mail (DKIM) [3] and some of Garfinkel's own usably

[1] S/MIME messages signed with *opaque* signatures can behave this way. Opaque signatures are robust to SMTP servers that might reformat messages, but the text cannot be retrieved without the appropriate parsing code.

secure email prototypes [49, 50] (as well as ABUSE) use email *headers* to carry extra information; mainstream clients (Mozilla Thunderbird, Apple Mail, Microsoft Outlook and others) simply ignore unfamiliar headers, so users never even see them. Thus, push-back can be avoided.

3.2.2 Patterns specifically applicable to secure/trustworthy email

In his work, Garfinkel also provides several design patterns that apply to systems concerned with identification online, and/or that leverage public key cryptography [47, pp. 331–339]. Not all of these apply to ABUSE, but we explain them here so that we can intelligently discuss the decisions we made when choosing patterns to apply during our design phase.

Leverage existing identification [47, p. 331]

Rather than throwing out systems that already provide some guarantees in real life, build off of them. When a system of identification re-affirms some kind of relationship that users already understand, it is much easier for them to make reasonable judgments about participants in the system. For example, Dartmouth's PKI certificates re-affirm the real-world relationship between an entity that is attached to the college. A user presented with such a certificate (provided he can reliably determine that the certificate is really from Dartmouth) can then make decisions based upon his understanding of this real-world association.

Email-based identification and authorization [47, p. 332]

This pattern promotes leveraging existing identification by using the ability to receive mail at a given address to boot-strap account setup and management for other systems. This behavior is commonly used by websites that require log-in credentials for use. Following the "send S/MIME-signed email" pattern could help mitigate the risk of phishing.

Send S/MIME-signed email [47, p. 333]

The action recommended by this pattern is obvious. Garfinkel envisions this pattern being adopted first by organizations sending official communications to recipients, getting them used to seeing signed mail and distributing keys that can later be used to send encrypted mail back to the organization. There are usability issues with S/MIME that would negatively impact its utility for these purposes; we touch on some in Section 4.1. Garfinkel discusses more in [47, Chapter 6]. The perfect is the enemy of the good, however, and some of Garfinkel's remaining patterns do address some of these problems.

Create keys when needed [47, p. 334]

Cryptographic protocols that leverage public key cryptography require some method of authenticating bindings between keys and parties to communication. Systems based on X.509 use certification authorities, but it is also possible to pre-distribute public keys or use the PGP web-of-trust model. In the absence of such authentication, parties can provide their keys to each other at the point of connection establishment. If an attacker Trudy imposes herself between Adam and Bob when they begin communication, Trudy can provide both men with keys of her choosing. Thus, Trudy will be able to read all of their communication without them knowing. However, if Trudy is not present at the moment of key exchange, Adam and Bob are safe. Given that the alternative is communicating in the clear (or not communicating at all), Garfinkel argues that creating and distributing encryption keys whenever they are needed at least cuts out the threat of passive eavesdropping and is thus worthwhile.

Key Continuity Management (KCM) [47, p. 335]

X.509 and PGP attempt to provide users with a notion of a concrete identity; if the PKI glue works as intended, Bob can know that he is talking to Alice. Garfinkel advances the notion that, often, Bob doesn't care that he's talking to Alice; rather, he cares that the Alice he's talking to now is the same Alice he has been talking to all along. The *Key Continuity Management (KCM)* pattern reflects this notion, and is discussed in greater detail in Section 4.1.

Track received keys [47, p. 336]

Along with KCM, Garfinkel recommends keeping track of how familiar a received key is. The intuition is that keys used regularly over a longer period of time are more likely to be legitimate than a new key. Automatically tracking key usage and enabling users to perceive this information can help them to apply this intuition.

Track recipients [47, p. 337]

This pattern is not as aggressive as it initially sounds. Garfinkel advocates sending S/MIME signed email in an earlier pattern, and here asserts that senders should make a best-effort attempt to determine which recipients can appropriately handle such messages. Thus, they can avoid sending signed email to users who will be annoyed by or, worse, unable to read the messages due to a deficient client. This pattern plays into the "No external burden" pattern discussed on page 42.

Migrate and backup keys [47, p. 338]

If the "Create keys when needed" pattern is to be followed, users will often be unaware of the existence of important key material. Thus, to avoid blind-siding users in the event of key loss, systems must also take pains to ensure that keys are available when needed and backed up religiously.

Distinguish internal senders [47, p. 339]

In email as it exists today, users can send mail that appears to come from one domain that is actually sent through an email server in another domain. For example, Alice can send mail "From" alice@dartmouth.edu, by setting the email headers appropriately, authenticating to the SMTP server for her GMail account, and sending the mail out that way. Bob, on the other hand might send email from bob@dartmouth.edu over a properly authenticated connection to Dartmouth's own mail server. Garfinkel argues that recipients would benefit from being able to discern between these two cases.

3.2.3 Remaining, inapplicable patterns

The rest of the design patterns put forth in [47] apply more to changing overall attitudes toward behaving securely and to the specific area of protecting user privacy by providing usable disk sanitization tools. We have enumerated here all the patterns that we believe might apply to the design of ABUSE, and will discuss specifics in Chapter 5. We also apply Cranor's framework for evaluating secure systems that build humans in [21] to our work to determine that following the chosen patterns has indeed resulted in a system that accounts for the usability stumbling blocks identified by the HCISEC community.

3.3 Wrap-up

In this chapter, we have motivated the choice of a user-centric approach to the problem we have identified. While we acknowledge that usability issues exist when designing such a system, we have laid out a set of design principles from the HCISEC field that we can follow to alleviate these concerns. We discuss how they informed the design of ABUSE in Chapter 5 and Chapter 6. Choosing, as we have, to build atop an existing signed email technology, we are left with the following characteristics:

- **expressiveness**: the system must be expressive enough to provide contextual support for the classes of trust flows enumerated in Chapter 2.
- **good signals**: the new signals used to carry this information must be good (as defined in Section 1.3.3).
- **usability**: the system must be designed to be usable according to guidelines from the HCISEC community.

Having laid out these desired system characteristics, we now explore related work and discuss how it fails to meet our criteria.

Chapter 4

Related Work

In Chapter 3, we derived the characteristics required of a system that solves the email trust problem we have laid out:

- the system must be expressive enough to provide contextual support for the classes of trust flows enumerated in Chapter 2.
- the new features of the system must not cause users to become more vulnerable to attacks than non-users.
- the system must be designed to be usable according to guidelines from the HCISEC community.

We now discuss the current state of work in secure email and relevant technologies in trust to show how none of them meet these criteria. We move on to address work that has applied sociological notions of trust to computer security, akin to the trust flow work in Chapter 2. Last, we discuss work in usably secure email and mention some other tangentially related HCISEC research, as well as providing a pointer to an excellent retrospective of the growth of the field as a whole.

4.1 Secure/Trustworthy email technologies

Public key cryptography and the various flavors of PKI discussed in Section 1.1, paved the way for secure and trustworthy email. Two, S/MIME [95, 96] and PGP/MIME [35] have achieved the widest deployment over the years and become standardized. Researchers have also advanced a few systems with interesting properties (discussed in Section 4.1.3) that have not achieved significant deployment. Industry has even put out some proprietary solutions (Section 4.1.4) that provide some interesting features when deployed in a "walled

Property	S/MIME	PGP/MIME
Message integrity	Yes	Yes
Non-repudiability	Yes	Yes
Centralized trust root	Yes	No
Meaning of a Certificate	Trust root attests to binding of key to holder	Unclear [66]
Deployment	Heavy in government, industry; some academic	Some academic, individuals; perhaps industry
Trust model	Institutional	Process-based

Table 4.1: The differences between S/MIME and PGP/MIME

garden"—a closed environment in which all participants are users of the system. All of these approaches focus on providing message integrity, sender authenticity, and assurance of sender identity. Some provide enough additional context to support some of the trust flows presented earlier, but not all. Some focus on addressing usability problems in secure email. We have leveraged ideas from some of these systems, and some could be interesting to explore as a combined solution along with ABUSE.[1]

4.1.1 S/MIME

To address email security and privacy concerns, many organizations in the commercial, federal and educational sectors have deployed S/MIME [95, 96], a secure email standard that leverages an X.509 PKI [19] to provide message integrity and non-repudiation via digital signatures. [67, 84] An S/MIME signature block contains, in addition to the actual digital signature over the message body, the identity certificate of the sender. In this way, the system also provides sender authenticity and assurance of sender identity—in addition to the sender's public key. Note that S/MIME does not cover the headers of a message, which could leave some issues. For instance, an attacker might change the "From:" line in an S/MIME email header. This is likely why the S/MIME standard dictates that the From address in a message must match the email address in the credential used to sign it. Many clients do not seem to enforce this and, indeed, there are cases where strict enforcement of this policy leads to further problems!

Even in cases in which the sender is familiar to the recipient, usability issues exist. One interesting problem arises from the fact that standard S/MIME clients treat all installed trust roots as equal.[2] When an S/MIME signature is deemed valid, the client will display

[1] Portions of this section were adapted from my thesis proposal.

[2] At least, all that are allowed to identify users for the purposes of signing email. Though X.509 does technically allow a subordinate CA to be configured to only speak about certain domains, this requires much a priori policy construction that is rarely performed in a real-world deployment.

the same information to the user *regardless of which CA issued the credentials used to sign the message!* The author leveraged this quirk, along with Thawte's Freemail CA and Dartmouth's name directory, to generate what appears at first glance to be legitimately signed S/MIME email from the College's president. The Freemail CA will allow a user to get a certificate for any email address over which he can demonstrate control. Dartmouth's name directory allows users to choose any nickname, even one close to the actual name of the President. Yes, the certificate used to sign this message was not from Dartmouth's CA, but this is only evident after some extra effort (the Director of Technical Services was reportedly taken in by the ruse). The provenance of the signing certificate is not immediately obvious, and the user is likely to assume that signed mail from a Dartmouth email address uses a certificate from the Dartmouth CA. Possession of such a certificate implies a relationship with the College; obtaining a Freemail certificate for a Dartmouth email address allows an attacker to trick a target into assuming the existence of such a relationship.

In terms of the trust model presented in Section 1.3, S/MIME can do one of two things for the recipient, depending on whether she has experience with the sender. If she knows the sender a priori, S/MIME can enable the recipient to leverage her trust in an institution to assure herself of the sender's identity and thus apply her process-based trust to the incoming message. If she has little or no prior experience with the sender, then S/MIME allows the recipient to extend some measure of institutionally-based trust to the sender. This is not enough to avoid the issues discussed in Section 1.2.1, however. Membership in a subculture is being established, e.g. the "Member of the Dartmouth Community" subculture. This allows some kind of sphere to be defined in which the individual can be trusted. Standard S/MIME implementations can only establish membership in a fairly large subculture. The members of this group are not homogenous enough to clearly define an area in which all members should be trusted. If we can build a system that allows a smaller subculture to be defined ("Members of the PKI/Trust Lab", or "Sean Smith's PhD Students"), this makes it more likely that users will be able to come to useful trust conclusions.

Despite these issues, S/MIME *has* provided both message integrity and sender authenticity, as well as the sender's public key—provided that the recipient trusts the sender's CA and that the sender's private key has remained private. S/MIME, therefore, could be a good starting point for a trustworthy email system, and the public key in particular could provide a way to hook further contextual information about the sender into the message.

4.1.2 PGP/MIME

PGP has been applied to email by combining it with the MIME standard to create PGP/MIME. [35] Trust in PGP is all process-based. Previous interactions between two parties create the basis for building local trust, and then that can be extended to people with whom a party has not interacted directly. PGP provides message integrity and non-repudiation as well as binding the sender's public key to his signed messages, just like S/MIME. PGP/MIME message signatures do not cover message headers, either. One disadvantage of PGP is that it is not widely deployed in enterprises, either commercial, federal or academic [47, p. 168]. Another is that PGP/MIME clients have usability issues, as mentioned in Section 1.1.2 and explored in [125].

PGP, like S/MIME, focuses on providing message recipients with some assurance of sender identity. PGP's lack of widespread deployment makes it less appealing than S/MIME as a basis for ABUSE, though the idea that message senders are presented to recipients with some sense of how the two are connected in the "web-of-trust" informs the usage of attribute chains in ABUSE.

4.1.3 Research approaches

Key Continuity Management (KCM)

PGP and X.509 are the only two traditional PKI systems with wide deployment. Many computer scientists, however, use a kind of stealth-PKI every single day when they use `ssh`. Secure Shell (SSH) is a remote-login program that uses public key cryptography to provide encrypted communication between two computers. The first time Alice connects to a machine M using `ssh`, the program on her local device D downloads the public key reported by M (unless D is already pre-configured with a key for M), generally after presenting a warning that the download is about to occur. Thereafter, D remembers M's key and informs Alice upon subsequent logins if M's reported key changes. Keys are distributed as needed, with no assurance at all that the mapping between machine and key is valid. Alice simply makes the assumption that an attacker wasn't providing a faulty key for M the first time she logged in, and her machine throws up a warning if ever anything changes. Garfinkel migrated this concept into the email space with his work on Stream [46] and CoPilot [50], reasoning that what users wanted from their email was assurance that the Bob they're talking to now is the same Bob they were talking to before. The first time Bob emails Alice, her client remembers his public key, notifying her if his signing key ever changes. The author and some colleagues also applied this concept to interactions on the web. [78]

Assuring continuity of identity in Stream and CoPilot makes explicit the focus on process-based trust in email, and still misses the use cases we are concerned about in this work. As we mentioned in Section 3.2.1, both Stream and CoPilot use email headers to carry around extra information. ABUSE borrows this idea in order to transmit attributes along with messages without violating Garfinkel's "no external burden" design pattern.

Role-Based Messaging

To my knowledge, only Role-Based Messaging [16,133] has attempted to address the same portion of the problem space as ABUSE.

Role Based Messaging (RBM) is a system that creates role-based mail accounts. Users who have appropriate credentials (where "appropriate" is defined by policy on a per-role basis) can log into those accounts to read mail sent *to* that role and also to send signed and encrypted mail *from* that role. Mail may be encrypted to a role, not simply to a specific user. Role membership is controlled by a PERMIS [15] back end, in which X.509 ACs are used to store role membership information. Policies can be added to messages to further control what recipients can do with them. A policy governs who can assign roles to users, though the system could be set up to allow any user to grant roles to others. Also, while these "role managers" can create new roles within their organization, they will not be recognized by the system. [131] Thus, they will not have mail accounts created for them and it is not clear what utility these ad-hoc roles would have in the system, if any.

Later RBM work introduced Policy Based Management (PBM), which adds infrastructure to allow organizations to advertise what kinds of policy languages they support. [134] PBM also allows third parties to sign off on particular implementations of particular policy languages and enforcement models, so that an enterprise can prevent (again, via policy) users from sending secure messages to an external organization whose mail system may not respect message permissions set by the sender.

In addition to several other issues, RBM offers a solution for part of the email trust problem. Users could all be granted roles, and then choose the appropriate role from which to send a message that needs to be trusted. It does not appear that users could claim multiple roles at the same time, but new, combined roles could feasibly be created to handle that. As mentioned above, these new roles could not be usefully created on the fly, however. Most importantly, RBM handles only role-based trust flows. While these may be common, they are far from the only class of useful flow. Furthermore, while the authors mention "user friendliness" as a design goal in one of their early papers, correspondence indicates that usability has been very much a second tier goal in their work thus far. [132] It is also not clear how RBM would handle many people needing to be in the same role at the same time; all of MISO's reliability coordinators may have to share the same role-based email account when communicating with people who do not know them beforehand. According to the authors, they have so far implemented an RBM policy decision engine and a distributed RSA algorithm that they plan to use to avoid having a single point of compromise in their system architecture. They plan to use Mozilla Thunderbird as a client platform for RBM, but did not say much about the RBM user experience beyond that.[3]

RBM handles role-based trust flows, provided that all roles can be enumerated during system set up. Presumably, new role-based accounts could be added to the system, but only by those with the authority to reconfigure the infrastructure. This approach has all the problems we presented in Section 3.1.1, and also does not concern itself with usability.

Attribute-Based Messaging

Unlike ABUSE and RBM, Attribute-Based Messaging (ABM) [9] does not work on the behalf of message recipients. Instead, it focuses on allowing sender to address messages using attributes instead of identities. One could address a message to all Computer Science majors who are also seniors, for example. There is an assumed pre-defined set of attributes in some set of organizational databases, and policies are defined that restrict which attributes can be used by which senders. Message recipients do not see the attributes of senders; indeed, ABM is not concerned with characteristics of the sender once it has decided which attributes he is allowed to use in address construction. The problems considered in ABM are orthogonal to this work.

4.1.4 Commercial approaches

Both Lotus Notes [136] and Groove Virtual Office [80] provide some measure of context for their users. Meant to be deployed in an enterprise setting, the systems provide signed messaging with an interface that reliably indicates when the sender of a message is internal to the company. They also allow for companies to establish trust relationships at the organizational level, so that a user at Company A can tell when a sender is certified as an employee of Company B. Essentially, these systems provide a more-usable S/MIME-style experience. Users are insulated from many of the issues that a multiplicity of trust roots bring to X.509-based secure email, but at the expense of being able to interoperate with users who aren't "inside the garden".

[3] This discussion was adapted from my thesis proposal and one of my previous papers. [79]

Garfinkel provides a lengthier discussion of these systems, and some others, in [47, chapter 5]. However, none focus on the problem of providing for users adequate context for deciding whether to trust unfamiliar correspondents.

4.2 Technological approaches to trust

In this section, we look at software that has taken a technological approach to the problem of helping users decide which entities they can trust.

4.2.1 Trust-Management-Based Technologies

Trust Management (TM) deals with automatically deciding a form of trust based on attributes and policies. Essentially, Bob makes a request of Alice, bundling with his request some set of credentials that assert somethings about him. The allowable requests, allowable assertions, and allowable relationships between entities in the system are all predefined in TM systems; they must be, in order to allow for an algorithmic decision about the request to be reached. Alice's computer puts Bob's request, his credentials (in some cases, there are credential repositories that hold statements about Bob beyond those that he provided), and Alice's policy into some decision module and then determines whether the policy indicates that the credentials authorize the request. TM could be used as a part of an algorithmic email trust system of the form discussed in Section 3.1.1, but we have already discussed why such a thing would be unlikely to work in practice.

TM systems use many different methods of representing credentials. Some use statements in a logical programming language like Datalog, some use an XML format, some use s-expressions, and some create their own format. Delegation Logic (DL) [71], the Role-based Trust-management (RT) framework [72–74], and SD3 [64] all use some variant of logical programming statements to encode credentials. These are text strings, and actually fairly human readable, so it would be possible to use them to carry the assertions we want to build into ABUSE. However, each of these systems has its own grammar that defines "legal credentials", and we would need to develop or find code to handle the chosen format. Trust Policy Language (TPL) [58] is an XML-based TM language that can be reduced to Prolog, another logical programming language. Libraries that deal with XML are readily available, but libraries to deal specifically with TPL are less so. KeyNote [8] and PolicyMaker [7] both define their own credential formats, and the availability of libraries to handle these formats is unknown. REFEREE [17] defines a simple policy language based on s-expressions. [99] Some also consider SDSI/SPKI, discussed below, a TM system based on s-expressions. The credential representation here is compact, though still not very human-readable. At least one C library is available for manipulating SDSI/SPKI credentials, but one would likely need to develop code to handle REFEREE.

TM systems, with their focus on deciding trust based on policies, would dictate an algorithmic approach to the email trust problem we have laid out. To summarize the arguments from Section 3.1.1 against such an approach:

1. it would require the comprehension of arbitrary text from arbitrary senders,
2. users are incapable of effectively enumerating their personal trust policies in a machine comprehensible format,
3. administrator-defined domain policies are difficult (and expensive) to get right,
4. domain policies are even harder and more expensive to maintain over time, and
5. it is unclear that domain policies useful for the average case are still applicable in exceptional circumstances.

The potential for using a TM system credential format to carry ABUSE signed assertions is discussed in Section 4.2.5.

4.2.2 Automated trust negotiation

Following the introduction of Trust Management, researchers noted that, in some cases, the credentials themselves might be sensitive information. This led to the field of *automated trust negotiation*, in which access to both credentials AND resources are controlled by policies. Per Lee et al:

> In trust negotiation, access control decisions are made based on the attributes of the entity requesting access to a particular resource, rather than his or her identity. To determine whether an entity should be granted access to a resource, the entity and resource provider conduct a bilateral and iterative exchange of policies and credentials (used to certify attributes) to incrementally establish trust in one another. [70]

This approach to the issue of managing trust shares the same usability issues in human scenarios as TM. The arguments reviewed at the end of Section 4.2.1 apply here as well. Both technologies could be useful at a organizational level, for managing shared access to network resources or similar. For example, the GridStat[4] project has begun using automated

[4] http://www.gridstat.net

trust negotiation to control distribution of sensor information in the context of monitoring the power grid. [55] However, humans would be unlikely to be able to use an approach like these to manage trust in other people with whom they communicate over email.

4.2.3 SDSI/SPKI

Around Dartmouth's PKI-Trust lab, we sometimes refer to SDSI/SPKI [18, 36, 100] as "the Libertarian Party of the PKI world: intellectually seductive, but one cannot shake the nagging feeling that it will not work in real life." Users are globally identified only by their public key; all other naming is local. Alice can assert a binding between "Bob" and the key she believes to belong to him. Carlo, with Alice's assertion in hand, can then issue his own assertions referring to "Alice's Bob". In this way, assertions can be chained to allow people who have never even heard of Bob to make statements about him. SDSI/SPKI certificates also support *tags*, arbitrary text meant to signify individual permissions. Howell provides some formal semantics for both naming and the use of tags in SDSI/SPKI [60], which could be useful in an algorithmic-style system. As discussed in the sections on TM and ATN, we eschew the algorithmic approach for the reasons discussed in Section 3.1.1.

Like PGP, trust is all process-based in SDSI/SPKI. Unlike PGP, in which users only assert some level of confidence in a binding between an identity and a key, SDSI/SPKI users can assert whatever they want about each other. This is interesting, in that it moves the focus of building trust off of identity and onto arbitrary properties that are bound to truly globally unique identifiers. So, SDSI/SPKI-style arbitrary signed assertions could be useful in terms of enabling a trustworthy messaging system to communicate context from senders to recipients along with messages. The tradeoffs of using the specific certificate format from SDSI/SPKI for ABUSE assertions are discussed in Section 4.2.5.

4.2.4 Non-identity X.509 PKI

In recent years, a pair of X.509-based PKI technologies have arisen that focus not on binding identities to key pairs, but on binding other kinds of properties instead. Both *X.509 Attribute Certificates (ACs)* [38] and *X.509 Proxy Certificates (PCs)* [120, 123] are expressed in ASN.1, a binary format, just like regular X.509 ID certificates. So, while the format is very compact, it is not human readable, and parsing can be difficult. Both ACs and PCs allow for arbitrary assertions to be built into X.509 certificates, and signed by users. Attribute Certificates are designed to, as the name suggests, use a hierarchy of Attribute Authorities (analogous to Certificate Authorities) to issue X.509 credentials binding arbitrary attributes to identities. AC PKIs use the same kind of revocation strategies as

X.509 identity PKIs, though the designers state that a system using short-lived ACs could likely do without revocation. There is a small set of pre-defined attributes that are meant to have global meaning. Other attributes can be defined by the deployers of an AC system for whatever local purposes they desire. Trust would be institutional in such a deployment; users trust that attributes are granted to individuals based on some policy implemented by the issuing organization, and so they are willing to believe the bindings provided.

As opposed to ACs, which can be either long-lived or short-lived, Proxy Certificates are *all* meant to be short-lived, and thus no revocation infrastructure is mandated by the RFC. PCs are designed to be issued by users who wish to delegate a subset of their permissions to processes running on their behalf in grid computing environments. As PCs are not meant for human consumption, it does not make sense to apply our model of human trust to a system that deploys them.

Both ACs and PCs rely on an X.509 identity PKI already being in place. ACs need it to enable applications to go from a public key to an identity to the attributes bound to that identity. Users of PCs issue them directly using the key pair they got by participating in the identity PKI. Neither technology can solve the problem we have laid out on its own, but either could be used to carry signed assertions as a part of a larger system. We discuss the tradeoffs in the next section.

4.2.5 Choosing the right technology for signed assertions

ABUSE requires signed assertions. As there are a plethora of formats available for this, many of which are discussed in the preceding sections, it seems unnecessary to define our own. Software tools at varying levels of maturity exist for manipulating a number of these formats, and choosing the most well-supported will both simplify the software engineering tasks before us and make the behavior of the system more likely to be correct.

Trust Management systems use a wide variety of credential formats. We could use any of them for ABUSE assertions, but without the need to put these credentials into a policy decision engine, the primary benefit of using such a format becomes irrelevant. Since none of the formats mentioned in Section 4.2.1 are well-supported in commodity software, we decided against using them.

SDSI/SPKI has not seen much use outside of academic prototypes, though there is a C library for manipulating SDSI/SPKI certificates. Prior experience [52] has shown us that trying to shoehorn SDSI/SPKI into an X.509-centric world can be frustrating, however.

This leaves us with ACs and PCs. OpenSSL [90], a widely used cryptographic library, and NSS [86], the Mozilla cryptography infrastructure, support X.509 well. OpenSSL supports PCs off the shelf. AC support, on the other hand, requires some extra code to be patched into OpenSSL. Thus, Proxy Certificates are our signed assertion format of choice. They have the best support among commodity tools, and the special features provided by other formats are not useful in our system.

4.3 Approaching trust from the human side

Broader concepts of "trust" have been thoroughly studied over the years by researchers in sociology, psychology and economics. Ideas drawn from these areas have, relatively recently, been applied to HCISEC.

4.3.1 Security as a socio-technical system

Security systems cannot be considered as solely technological entities. They exist in a social context that governs their patterns of usage, their deployment, and their design. Thus, secure systems are actually *socio-technical* entities. For example, Whitten and Tygar advanced the idea that security software faced the unique problem of the "unmotivated user", saying that behaving securely is not something users are innately motivated to care about. [125] Adams and Sasse challenged this notion, pointing out that users *can* be motivated to behave securely—and not simply by punishing them when they make mistakes. [2] By creating an environment in an organization that creates positive social incentives for acting securely, one can build a secure socio-technical system. Sasse and others went on to posit design processes informed by sociological understandings of "the whole socio-technical system that is security" [105] and realized *Appropriate and Effective Guidance for Information Security (AEGIS)* in [40, 42]. AEGIS is targeted at the design of an application for a specific purpose; while our design process is informed by this work, directives such as "gather important stakeholders" cannot really be applied to ABUSE, as it is not an application, but a building block technology.

4.3.2 Sociological trust models applied to system design

Our work is concerned with a very specific corner of the trust problem: how a user decides whether to honor a request they have received from a person they do not know. Some often-studied aspects of trust that we do not consider include

1. issues that arise as two humans build trust over time,
2. issues surrounding the establishment of mutual trust, or

3. cases where the sender of a message has promised some action in return for the satisfaction of his request.

This is not to say that ABUSE could not be useful in addressing the above, merely that we did not analyze cases like this with our trust model during our design process and thus cannot speak authoritatively about how these kinds of situations fit into the ABUSE model.

In [98], Riegelsberger et al. provide a broader definition of trust than we do, defining it as "an attitude of positive expectation that one's vulnerabilities will not be exploited." [98] Our definition of trust is closer to what those authors refer to as *reliability*. Working from their definition, Riegelsberger et al. develop a model that can be applied to a variety of kinds of "computer-mediated communication", including voice and video chat, interactions with web sites, email, eCommerce, online auctions, telephone calls and more. Their model has two actors, the *trustor* and the *trustee*, who are participating in some transaction that requires them to trust one another. Consider a person buying an item from an online merchant. The merchant is the trustee, asking the buyer (the trustor) to transfer some money with the expectation that the goods will follow. When the items appear in the mail, the trustee has provided *fulfillment* of this expectation. The kinds of questions that we consider do not have a fulfillment step; Bob the trustee is recommending to Alice the trustor a course of action and intimating that she will perceive some kind of gain by pursuing it. Including fulfillment as it does, the Riegelsberger et al. model is in some ways more complicated than we need. As an example, issues surrounding incentives for the trustee to provide fulfillment are not relevant to our work.

The authors do, however, provide a discussion of an abstract trust interaction that is applicable to our work. When first communicating, the trustor and trustee exchange signals that bilaterally communicate both *contextual properties* and *intrinsic properties*, in order to initiate a trust relationship. Intrinsic properties are innate characteristics of the actors: how willing they are to take risks, how much they have internalized social norms that bias humans towards behaving honestly, how benevolent they are and so on. Contextual properties are more in line with the kinds of attributes we have considered, but also include things like prior experience shared among the two parties, concern over decreasing the potential for future interactions, and reputation within a community. These properties are broken into three categories:

1. those that express *temporal embeddedness*,

2. those that express *social embeddedness*, and

3. those that express *institutional embeddedness*.

Temporal embeddedness collectively embodies expectations relating to interactions other than the one currently taking place. These may be expectations drawn from prior experience or intentions of future transactions. In our work, we are not concerned with these kinds of properties, unless they arrive as sort of testimonials from other people. In that case, they would be evidence of social embeddedness—essentially, reputation. Institutional embeddedness maps conveniently onto the notion of institutional trust that we discussed in Section 1.3.

The model presented by Riegelsberger et al. could certainly be a useful tool for analyzing the kinds of transactions that we consider. However, the shared-expectations model that we have advanced suits our needs and does not include extra complexity. The model presented in [98] is used in [41] to devise some design principals for entire socio-technical systems that go beyond software to include business processes and organizational structure.

4.4 Usable Security

Garfinkel provides a thorough discussion of work in usability and security in [47, Chapter 2], combining his own work with design principles from Karat [65]; Zurko and Simon [137]; Whitten [126]; Yee [130]; Perrig and Song [93]; and Balfanz, Durfree and Smetters [4] to come up with a set of design patterns to use when building secure software for real humans to use. We discussed his relevant patterns in Section 3.2; we relate how we followed them in Chapter 5.

4.4.1 Secure email usability

Both Garfinkel and Whitten have developed clients for secure email that focus on usability. [47, 126] Whitten's "Lim" system was designed to help users understand the key certification portion of a PGP/MIME email system. There are also several systems that insert a transparent proxy between the user and his email server [12, 47, 49, 92] which handles all encryption and signing duties. The details are, thus, abstracted away from the user. They each have their strengths and weaknesses, and none focus on providing more context for users to make trust decisions. None focused on the problem of providing more context for users trying to make trust decisions regarding incoming messages.

Groove Virtual Office and Lotus Notes, discussed earlier, provide some usability benefits by making a system administrator responsible for key and certificate management. HushMail [61] is a web-based email system that uses PGP in a similar fashion; users do not have to deal with issues of key generation and certification, as the HushMail team handles it all behind the scenes. In all three cases, these benefits only apply if communicating parties are inside the same “walled garden”.

4.4.2 Providing extra context for the Web

Users of the Web also face trust decisions; they must decide whether to provide potentially sensitive information to websites to which they navigate. Both TrustBar [57] and Net Trust [51] seek to provide these users with more context for these trust decisions. Websites that use SSL to protect communication with users all have X.509 certificates that contain some extra pieces of information about the site; TrustBar moves this information to the forefront of the user experience. Net Trust allows users to join a social network of users that rate websites for trustworthiness, and then displays the appropriate ratings when a given site is visited. These approaches both seek to contextualize trust decisions in unfamiliar entities like ABUSE, but neither is as flexible or as expressive as our work—and they apply to websites as opposed to other humans.

4.4.3 Anti-phishing

If users came to expect signed mail containing ABUSE attributes as a matter of course, they would be less likely to believe phishing email. In this sense, ABUSE can be an anti-phishing technology, though this is not its specific goal. In addition to spam-filtering and forged-link-detection that is common in email clients like Apple Mail, Mozilla Thunderbird and Microsoft Outlook, there are also some approaches that do attempt to use a sense of sender context as an anti-phishing strategy. DomainKeys Identified Mail (DKIM) [3], Sender ID [76], and Sender Policy Framework (SPF) [128] all attempt to provide assurance that an email came from a legitimate account within the sender’s claimed domain of origin. If Alice receives mail from “bob@foo.com”, these technologies allow Alice’s email client to, at least, determine that the message was originally forwarded by servers belonging to foo.com. Thus, she can be sure that some zombie did not send this email and forge the source address. Again, this is far from as expressive or flexible as ABUSE, and supports none of the trust flows we enumerated in Chapter 2.

4.5 Conclusion

In this chapter, we have shown that other systems lack the flexibility and usability required to enable users to leverage over email the kinds of trust flows we enumerated in Chapter 2. We have, however, identified some useful technologies upon which we can build

ABUSE: S/MIME and X.509 Proxy Certificates. In the next chapter, we discuss how these technologies play into the system architecture we have designed.

Chapter 5

Attribute-Based, Usefully Secure Email (ABUSE)

In this chapter, we present Attribute-Based, Usefully Secure Email, our solution to the email trust problems we have enumerated thus far. Back in Chapter 3, we enumerated three high-level characteristics that our system must possess:

- **expressiveness**: the system must be expressive enough to provide contextual support for the classes of trust flows enumerated in Chapter 2.
- **good signals**: the new signals used to carry this information must be cheap/easy for trustworthy actors to emit, and costly/difficult for attackers.
- **usability**: the system must be designed to be usable according to guidelines from the HCISEC community.

These three characteristics lead to a number of design goals that affect both the system architecture and GUI design. First we examine the architectural goals that fall out and discuss the system we developed to fulfill them. Next, we enumerate the user interface design goals dictated by the above and present the GUIs we developed in response. We do not explore in this chapter the process by which we came to our GUI design; we put these issues off until Chapter 6 in order to avoid bogging down this chapter.

5.1 Architectural design goals

As we mentioned in Section 1.5 and justified in Section 3.1, we have chosen to build ABUSE as a user-centric system that helps people make better informed trust decisions about incoming email. Just from this, we can enumerate some simple goals:

- enable senders to bind assertions about themselves to their outgoing email, and
- reliably convey this context information to recipients, so that it may inform their judgment.

By *reliably convey*, we mean that the ABUSE architecture must guard against the malicious fabrication of attributes, and detect when attackers interfere with attributes bound to messages. These are vague goals, but we will refine them by stepping through the three characteristics provided above. First, we consider refinements stemming from our expressiveness requirement.

5.1.1 Goals dictated by the need for expressiveness

The trust flows enumerated in Chapter 2 encompass a broad range of cases. Even just naively looking at these flows and the examples that generated the, we can see that ABUSE needs to be able to express any organizational role, any permission, the delegation of a job function or single permission, re-delegation, and also handle temporal issues. It may also be useful to enable the signaling of process-based trust, in the manner of a testimonial. If Alice has experienced a satisfactory transaction with Bob in some domain, she can provide Bob with this piece of signed feedback. When, in the future, he explores transactions with Alice's colleagues he could include this testimonial to signal to them that Alice considers him at least somewhat trustworthy.

Given this range of possibly useful statements it seems unlikely that we could pre-enumerate everything users would want to assert within ABUSE. Furthermore, given that simply trying to maintain an accurate list of roles in a single enterprise can be very expensive and challenging in the real world [114], we can assume that maintaining an accurate directory of an even larger set of assertions about users would be even less tractable. Thus, we can put forth the following two design goals:

- avoid limiting the space of possible assertions, and
- avoid the need for an organization-wide "Assertion Administrator".

It is fine if we include some centralized infrastructure for the storage and distribution of these assertions, but we should not require there to be a single person or group whose job it is to certify every single statement that users wish to make about each other. Take the coopetition scenario in Section 2.3.3, for example. Getting each delegation and re-delegation in that trust flow authorized by some centralized authority would have been extremely onerous, and thus probably worked around by the users. That said, we do not wish to impose a heavy burden on end-users either. So, we can add the following goal to our list:

- minimize the administrative burden on users throughout the system.

Adding these three design goals for ABUSE should enable us to build a system expressive enough to handle the kinds of trust flows enumerated in Chapter 2. We empirically evaluate this belief in Chapter 8 and Chapter 9.

5.1.2 Goals that lead to good signals

In order to enable the building of a shared set of expectations, senders must be able to signal some of their characteristics to message recipients. We wish to design good signals that are also readily comprehensible to users. The issue of comprehensibility is one of usability and properly dealt with in the next section; here we discuss designing the signals so that they are cheap and easy for trustworthy actors to emit while remaining costly or difficult for attackers.

The first step we can take is requiring the assertions used in ABUSE to be covered by a valid digital signature. This requirement makes it at least a bit more challenging for an attacker to completely fabricate an assertion. The next issue, then is the provenance of the signing key. As we stated above, we do not want to require a centralized signatory authority for all attributes. However, avoiding the involvement of a central authority all together would, essentially, introduce into ABUSE all the problems faced by PGP. Furthermore, we have already expressed that ABUSE is meant to be deployed inside some logical organization, a setting in which an X.509 identity PKI and S/MIME can get a lot of traction on enabling trustworthy messaging between users familiar with one another. So, it makes sense for us to leverage this existing infrastructure—indeed, Garfinkel's "leverage existing identification" pattern (p. 43) recommends this course of action. We take a hybrid approach, in which some relatively small set of "top-level" assertions are made about end users directly by the organization. This set would include characteristics that are unlikely to change rapidly, such as being a "student" or an "employee" of a university. There is already a set of common attributes that should be included in institutional directories across academia [33]; this would be an ideal starting point in that space. We then enable users to chain new signed statements off of these assertions. We refer to one of these chains as an ABUSE *attribute*. In this way, we make it difficult for an attacker to cryptographically mimic a legitimate attribute; he cannot forge top-level attributes, and he must subvert or or collude with other participants in the system in order to generate a longer chain. This does not exclude the possibility of the attacker using non-cryptographic methods to mimic the signaling of an ABUSE attribute; he could still try to manipulate the GUI of the recipient's messaging client in order to send false signals. However, dealing with this class of problems is a GUI usability issue; we address these in Section 5.3.

From this discussion, we can derive a modification of one of our earlier design goals and craft one additional goal as well:

- enable senders to bind their digitally signed attributes to their outgoing email, and
- allow users to chain new assertions (about other people) off of attributes they already possess.

5.1.3 Architectural usability goals

Usability is a theme that has run throughout this entire work. The whole approach that we have used to identify and explore the problem at hand is informed by HCISEC research and principles from that community. We will point out the influence of Garfinkel's applicable patterns during the appropriate portions of this chapter.

Garfinkel's first three patterns, "Least surprise/least astonishment," "Good security now," and "Provide standard security policies" (pp. 41, 41, and 41) are guiding principles of our work. By studying how trust flows in real-life scenarios, we can design ABUSE so that it enables the same kinds of behaviors and does not "surprise" the user by forcing him to behave in ways that are unnatural. We acknowledge that taking a user-centric approach deprives us of the ability to create a system that is provably secure, but we believe that it will enable us to build something that can be used securely in the real world. ABUSE is not perfect, but it is "good security now." As we explain in our discussion of algorithmic approaches to trust in email (Section 3.1.1), the "policy kit" approach is not usable. Garfinkel proposes that a small set of good-but-not-perfect policies be provided for users. ABUSE essentially provides one policy: if the user decides a request is reasonable, based on the accompanying attributes, the user should act. If something goes awry, the organization can audit the offending employee's email and investigate what informed their trust decision.

There are also a pair of design patterns that ABUSE meets by its very nature: the secure-messaging-specific "distinguish internal senders" and "send S/MIME email" patterns (p. 45 and 43). Anyone with attributes is operating within the same trust infrastructure as the recipient, so internal senders are easily identified: they have attributes. Senders outside the infrastructure do not. Given that ABUSE is built on top of S/MIME, it should be obvious how we have applied that pattern.

The "no external burden" pattern (p. 42) is the only one that has specific consequences for the architecture of ABUSE. Applying this pattern, we can add one final design goal to our list:

- avoid push-back from users without ABUSE-savvy clients.

5.1.4 Collecting our architectural design goals

In this section we have refined a set of design goals for the ABUSE architecture from the high-level characteristics we enumerated in Chapter 3. They are:

1. enable senders to bind their digitally signed attributes to their outgoing email,
2. reliably convey this context information to recipients, so that it may inform their judgment,
3. allow users to chain new assertions (about other people) off of attributes they already possess,
4. avoid limiting the space of possible assertions,
5. avoid the need for an organization-wide "Attribute Administrator",
6. minimize the administrative burden on users throughout the system, and
7. avoid push-back from users without ABUSE-savvy clients.

We now discuss the architecture of the system and the ways in which it meets these goals.

5.2 The ABUSE architecture

ABUSE is designed to rely upon two pieces of existing infrastructure: an email system and an X.509 identity PKI. In addition to these, ABUSE requires two component pieces:

- an ABUSE-savvy email client, and
- a centralized ABUSE attribute store.

The ABUSE client participates in a number of different facets of the system: attribute *presentation*, *issuance*, *distribution* and *validation*. Attribute presentation, obviously, is a GUI question and will be discussed in Section 5.4.1. There is a GUI involved with attribute issuance (Section 5.4.3), since users are involved in this process, but there are also

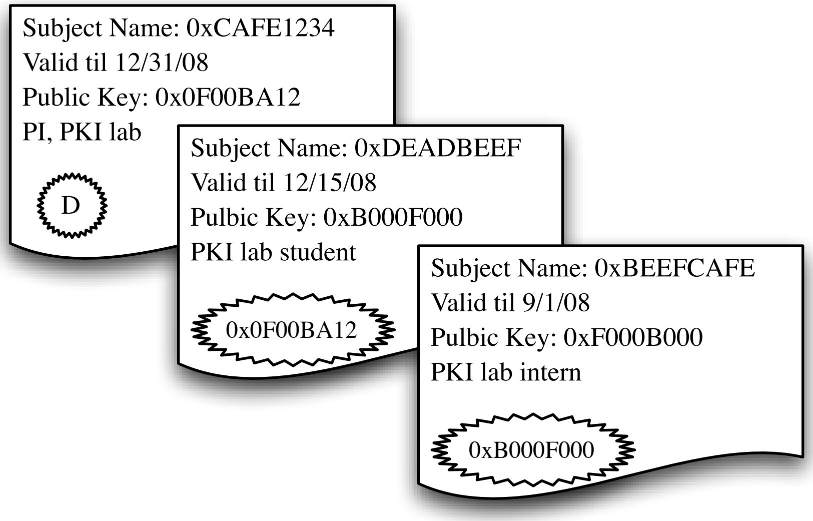


Figure 5.1: A chain of signed assertions, making up an attribute. We use X.509 Proxy Certificates as our assertion format. "D" represents a signature by Dartmouth's trust root. The ordering of the elements of this chain is unambiguously determined by the signatures on the certificates. Note the use of public keys in the place of human names; we rely on the organizational identity PKI to connect public keys to individuals, as shown in Figure 5.2.

interactions between the client and the attribute store that we will discuss here. Distribution is primarily an architectural issue, though users do choose which attributes get bound to and sent out with their messages. The UI for this selection process is shown in Section 5.4.2. The attribute store participates in issuance, distribution and, obviously attribute *storage*. Before discussing any of these processes, we first explain the structure of an attribute.

5.2.1 ABUSE attributes

As we discussed earlier in Section 1.5 and Section 5.1, an ABUSE attribute is a chain of digitally signed assertions, rooted at the CA of the organization's identity PKI (Figure 5.1). We go through a concrete example of an attribute and its component assertions in Section 5.4.1. Each assertion is an X.509 Proxy Certificate (PC), the canonical structure and usage of which are discussed in Section 4.2.4. The two aspects of the PC specification that we bend relate to *naming* and certificate *validity period*. Recall from Section 1.1.2 that a certificate is a digitally signed statement made by an *issuer*, binding a public key to a *subject* for some period of time (a *validity period*). In normal X.509 schemes, the subject and issuer of a certificate are identified by *distinguished names*, which are supposed to be globally unique. There are a plethora of issues with distinguished names [37], which led some to promote the usage of the public keys themselves as identifiers. [18, 36, 100] We

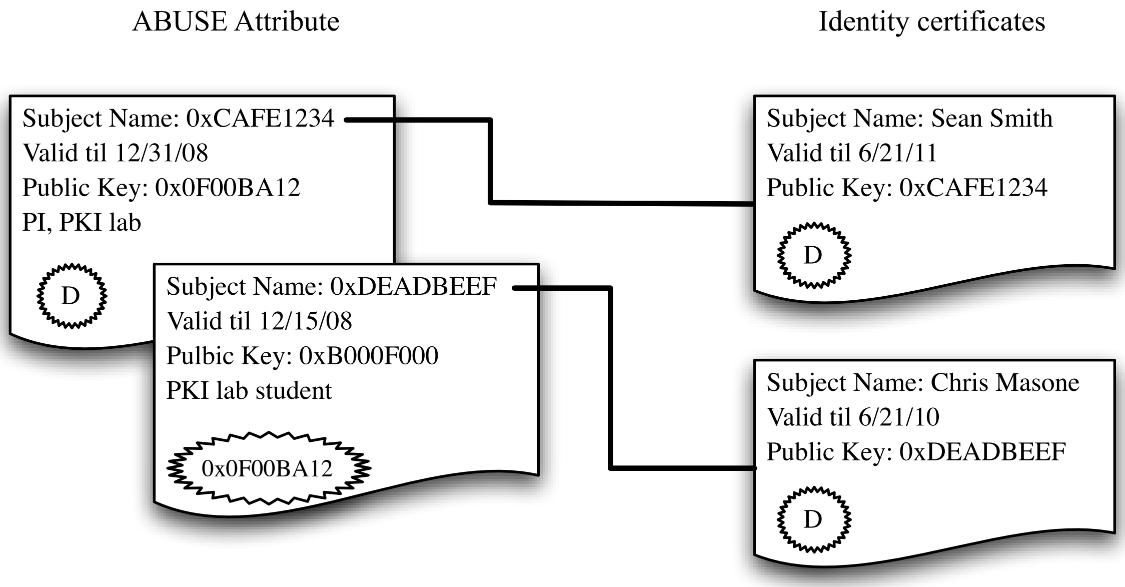


Figure 5.2: In the attribute shown on the left, subjects are identified by the public keys tied to their names by the organizational identity PKI.

use both; the identity PKI on which we rely uses distinguished names to bind human users to public keys (which is how real-world deployments work, for good or ill), and we use the public keys from this PKI to identify issuers and subjects within ABUSE (Figure 5.2). This is the first way in which we depart from the PC specification [120], which calls for distinguished names to be used as identifiers.

The PC specification also calls for certificates to have a validity period on the same scale as the running time of a computational process. Usually, this would be measured in minutes or hours, though it could be on the order of days. These short recommended validity periods are how the PC specification mitigates the risk of designing a system that does not have a revocation strategy. There is no maximum validity period, however. ABUSE, similarly, does not mandate any maximum. We expect that assertions in ABUSE will have validity periods ranging from hours to months, though we cannot know for certain without seeing the system deployed in real-life scenarios. We acknowledge that choosing to eschew revocation and relying upon expiration in ABUSE creates a tradeoff. Avoiding revocation allows us to reduce the administrative burden across the board, which helps us meet design goals five and six. It does, however, make it impossible for issuers to selectively render a particular attribute invalid prior to its expiration. We discuss the rationale behind and the consequences of this decision in Section 5.5.1.

Our approach to representing the content of an assertion stays within the PC specification. Proxy Certificates contain a *policy* field that can contain arbitrary text. Thus, we can use this field to support any assertion we desire, allowing us to meet design goal number four.

Lastly, PCs bind an assertion and a subject identifier to a key pair. The private half of this pair can be used to issue new PCs. This allows users to issue new PCs using attributes that have been granted to them by other users. Thus, we enable the creation of chains of signed assertions, as required by goal three.

5.2.2 ABUSE attribute issuance

In a true PKI, individual entities generate and control their own private keys. Whether as a stepping stone to this world or due to a institutional unwillingness to deploy real PKI tools, we have seen centralized PKI signing services begin to make inroads in enterprise environments. [63, 82] Having end entities control their own secrets and perform their own cryptographic operations is much more scalable, and also more secure; the distributed approach avoids the introduction a single point of compromise into the system. That said, while a distributed approach is superior, it is more complex to implement.

To simplify the implementation of our prototype, we did not implement attribute issuance in a distributed fashion. Instead, the centralized store plays a key role in the process, as shown in Figure 5.3. At this point, our greatest need is to build something that we can test, to verify the utility and usability of our approach to solving the email trust problem we have laid out. The rest of the ABUSE architecture is designed to be agnostic to the details of attribute issuance, which allows us to upgrade to a distributed solution at any time without modifications to the rest of the system. We outline such a distributed scheme in Section 10.1.1; Alice interacts directly with Bob in order to issue him an attribute. In that scenario, each participant performs his or her own cryptographic operations and the centralized store becomes nothing more than an attribute backup and key escrow service, a situation that is much less risky and far more scalable. Before doing any kind of wide-scale deployment, we would certainly implement this change.

In the current implementation, when Alice wishes to grant a new attribute to Bob, she first decides what she wants to say about him. Then, Alice authenticates to the attribute store with her identity certificate, downloads her current attributes from the centralized store and selects one whose authority she feels allows her to make the desired assertion. After inputting the assertion content, Alice indicates for how long she would like this attribute to be considered valid. Her client then sends this data, along with Bob's public key, over the authenticated channel to the store. The store creates a *Certificate Signing Request (CSR)*, a specially formatted data blob containing the hash of Bob's public key in the sub-

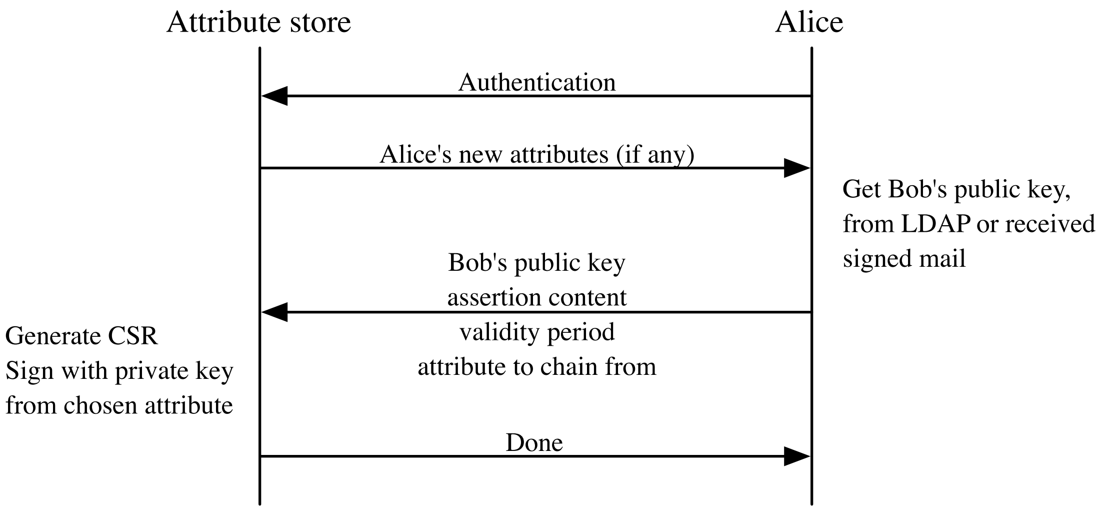


Figure 5.3: The protocol by which Alice issues a new attribute to Bob. After the process is complete, Bob acquires his new attribute the next time he authenticates to the attribute store.

ject field, the attribute content in the policy field, the validity period specified by Alice, and the public half of the key pair that has been generated for this PC. The store then signs the CSR with the private key associated with the attribute indicated by Alice. Since the store generated this key in the first place, allowing the machine to use this key for signing only upon Alice's appropriately authenticated request does not create any additional vulnerability. Only Alice can initiate the chaining of new assertions off of attributes that have been issued to her. Since the private key associated with an attribute never leaves the store, the only way an attacker could generate false attributes is by colluding with Alice, stealing the private key associated with her identity certificate, or by breaking into the machine hosting the centralized store.

Of those three weak points, the attribute store is the only one in the scope of our work. The attribute store should be appropriately hardened against compromise, and it should not run services other than the attribute store to limit the possible avenues of incursion. Running the store on a secure-hardware-based system [77] could be one useful avenue to protect the attribute keys against compromise, but this is a matter of implementation.

5.2.3 ABUSE attribute storage

All of Alice's attributes are available to her in the centralized store after she correctly authenticates using her credentials from the organizational identity PKI. This store is an LDAP directory, searchable by public key. Alice cannot get the private keys associated with her attributes; those never leave the store. Although we restrict access to Alice's attributes at the point of storage, ABUSE cannot guarantee the privacy of these assertions. Alice binds her attributes to outgoing messages and does not retain control of them thereafter. We do not provide any mechanism for Alice to delete or otherwise prevent forwarding of her attribute information; much like any text she puts into an ABUSE-enhanced message, the attributes are under the control of the recipient once they are sent.

5.2.4 ABUSE attribute distribution

As we mentioned above, Alice pulls her attributes down from the centralized store when she connects. She may cache them if she so desires, but this is an implementation detail. Until now, we have merely said that Alice can "bind" attributes of her choosing to her outgoing messages. In this section, we explain how this binding is realized, helping us meet our first design goal. One caveat here is that ABUSE is a prototype, and so there are likely ways that we could increase the efficiency of attribute distribution and validation. We have chosen to build something that works before concerning ourselves overmuch with reducing overhead.

Bundling an attribute with a message

As we mentioned earlier, an ABUSE attribute is a "chain" of X.509 Proxy Certificates. Such chains are represented as a set of individual certificates, where one is tied to the next by a digital signature as in Figure 5.1. PCs, as mentioned in Section 4.2.4, are natively formatted as binary data. In order for information to be sent along with email, it must be formatted as printable text. Fortunately, there are standardized ways to encode certificates as text, which we use to prepare attributes for transmission with an email message. Delineators are inserted between certificates and also between attributes, so that the client on the receiving end can appropriately parse the attributes for verification and display. For a message with two simple attributes, chains of length one, the header section of an outgoing message, has the following content added:

```
X-AbuseAttributes:  -----BEGIN CERTIFICATE-----
X-AbuseAttributes:  <multi-line certificate content here>
X-AbuseAttributes:  -----END CERTIFICATE-----
X-AbuseAttributes:  chaindivider
X-AbuseAttributes:  -----BEGIN CERTIFICATE-----
X-AbuseAttributes:  <multi-line certificate content here>
X-AbuseAttributes:  -----END CERTIFICATE-----
X-AbuseAttributes:  chaindivider
```

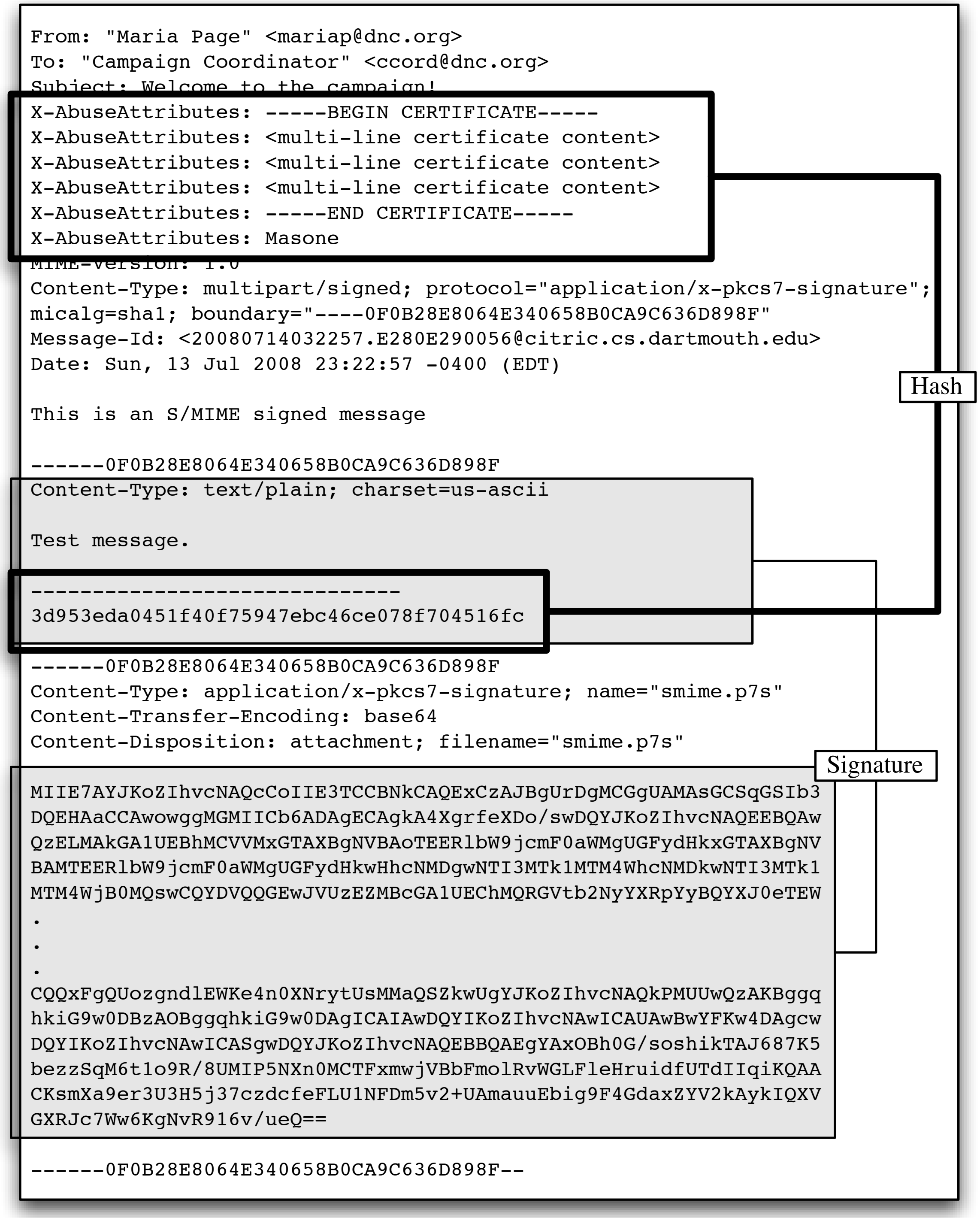
Attack type	Prevention strategy	Detection strategy
Assertion deletion Assertion insertion Assertion modification Assertion reordering	Encrypt with recipient public key	Certificate signatures render these detectable/irrelevant
Attribute deletion Attribute insertion Attribute modification Attribute reordering	Encrypt with recipient public key	Hash of attributes appended to message, covered by message signature makes these all detectable
Total attribute removal	Unpreventable by ABUSE	

Table 5.1: Possible technical attacks on the binding of attributes to messages.

Common graphical email clients (Apple Mail, Mozilla Thunderbird, Microsoft Outlook, etc) do not recognize the `X-AbuseAttributes` header name, and thus do not display this content. This is in contrast to schemes that enclose extra information as email attachments, like S/MIME and PGP/MIME. Before widespread client support for the format existed, users of PGP would experience push-back from the non-users to whom they sent mail, as the signature would be presented as a mysteriously named attachment by the recipient's email software. [47, p. 322] By using headers to carry the information we avoid this problem, thus helping achieve our seventh design goal.

Cryptographically tying attributes to a message

Digital signatures on email cover only the body of the message. We have already mentioned that this leaves subject, date, sender and other header information open to modification by attackers in Section 1.2.1. As our attributes are contained in headers, it is possible that they might be vulnerable. Attributes are, as we mentioned earlier, a chain of assertions. Each assertion is an X.509 Proxy Certificate, digitally signed by the private key associated with the previous certificate in the chain. Thus, none of the assertions in a given attribute can be removed or modified without the system detecting it during the validation process. The signatures on the certificates also allow us to determine the appropriate order of the assertions in an attribute, so attackers cannot insert single assertions or re-order the existing ones without detection. An attacker could *add* an entire attribute, though he would have to possess or create one that has been appropriately issued to the sender. He could also remove one or more attributes without detection, as long as he deletes them in their entirety. We could combat these attacks by somehow extending the signature on the message to cover our headers, but this would break compatibility with existing S/MIME clients and drastically increase push-back. What we choose to do, then, is to allow these attacks to happen but make them detectable. We do risk a small amount of push-back, but less than



```
From: "Maria Page" <mariap@dnc.org>
To: "Campaign Coordinator" <ccord@dnc.org>
Subject: Welcome to the campaign!
X-AbuseAttributes: -----BEGIN CERTIFICATE-----
X-AbuseAttributes: <multi-line certificate content>
X-AbuseAttributes: <multi-line certificate content>
X-AbuseAttributes: <multi-line certificate content>
X-AbuseAttributes: -----END CERTIFICATE-----
X-AbuseAttributes: Masone
MIME-Version: 1.0
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
micalg=sha1; boundary="----0F0B28E8064E340658B0CA9C636D898F"
Message-Id: <20080714032257.E280E290056@citric.cs.dartmouth.edu>
Date: Sun, 13 Jul 2008 23:22:57 -0400 (EDT)

This is an S/MIME signed message

------0F0B28E8064E340658B0CA9C636D898F
Content-Type: text/plain; charset=us-ascii

Test message.

------------------------------
3d953eda0451f40f75947ebc46ce078f704516fc

------0F0B28E8064E340658B0CA9C636D898F
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"

MIIE7AYJKoZIhvcNAQcCoIIE3TCCBNkCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
DQEHAaCCAwowggMGMIICb6ADAgECAgkA4XgrfeXDo/swDQYJKoZIhvcNAQEEBQAw
QzELMAkGA1UEBhMCVVMxGTAXBgNVBAoTEERlbW9jcmF0aWMgUGFydHkxGTAXBgNV
BAMTEERlbW9jcmF0aWMgUGFydHkwHhcNMDgwNTI3MTk1MTM4WhcNMDkwNTI3MTk1
MTM4WjB0MQswCQYDVQQGEwJVUzEZMBcGA1UEChMQRGVtb2NyYXRpYyBQYXJ0eTEW
.
.
.
CQQxFgQUozgndlEWKe4n0XNrytUsMMaQSZkwUgYJKoZIhvcNAQkPMUUwQzAKBggq
hkiG9w0DBzAOBggqhkiG9w0DAgICAIAwDQYIKoZIhvcNAwICAUAwBwYFKw4DAgcw
DQYIKoZIhvcNAwICASgwDQYJKoZIhvcNAQEBBQAEgYAxOBh0G/soshikTAJ687K5
bezzSqM6t1o9R/8UMIP5NXn0MCTFxmwjVBbFmolRvWGLFleHruidfUTdIIqiKQAA
CKsmXa9er3U3H5j37czdcfeFLU1NFDm5v2+UAmauuEbig9F4GdaxZYV2kAykIQXV
GXRJc7Ww6KgNvR916v/ueQ==

------0F0B28E8064E340658B0CA9C636D898F--
```

Figure 5.4: In order to protect the attributes we have placed in the header, we generate a hash and append it to the text in the message. When the email client generates an S/MIME signature over the message body, the signature covers the hash as well.

was faced by PGP/MIME and S/MIME.

In order to detect the addition or removal of whole attributes, we generate a hash of the attributes that the sender has chosen to bind to the message, append it to the message text, and then allow the client to generate a signature over the entirety of the message body as usual (Figure 5.4). Remember that any modification of the body will cause the signature on the message to be invalid, enabling the attack to be detected. This includes any attacks on this attribute hash. If the S/MIME signature is invalid, we can no longer be certain if the attributes in the headers are those that the sender meant to bind to the message, so we suppress all of them. This also allows us to detect the total removal of attributes; if there is a hash at the end of the message but no attributes in the headers, we know that something is wrong. The possible attacks on ABUSE attributes as a result of their placement in email headers and attendant mitigation strategies are shown in Table 5.1.

Having described all the facets of ABUSE in which the centralized store participates, we now move on to duties handled solely by the client.

5.2.5 Attribute validation

When a message is received, the email client does its standard S/MIME signature validation. If this fails because the signing certificate has been revoked, we ignore the attributes altogether. This allows an organization to exercise at least some coarse-grained control over user attributes; if the organization chooses to cut ties with an individual, we no longer allow that individual to claim attributes rooted in that organization's trust infrastructure. Assuming the signing certificate is not revoked, the attributes are parsed out of the header and individually validated. Currently, ABUSE does not support bridged PKIs, described in Section 1.1.2, though we discuss how to add this in Section 10.1.2. Thus, in the current prototype, the first assertion in an attribute must be signed by the organizational trust root. The validation process is shown in Figure 5.5. We take a single attribute, pull off the final assertion in the chain, and then pass it to OpenSSL [90] command-line tools to be validated against the organizational trust root. The intermediate certificates in the chain (if any) are also passed in to be used by OpenSSL as it tries to build a path of valid certificates connecting the final assertion to the trust root. We run the software in such a way that it ensures that all the signatures on the certificates are valid, flagging any certificates that are outside of their validity period. We keep track of this validity information so that we can display it to the user later, rather than declaring the attribute invalid. We also ensure that the hash of the public key in the message signing certificate matches the subject of the terminal certificate in the chain that makes up each attribute. Attributes that are invalid, or which

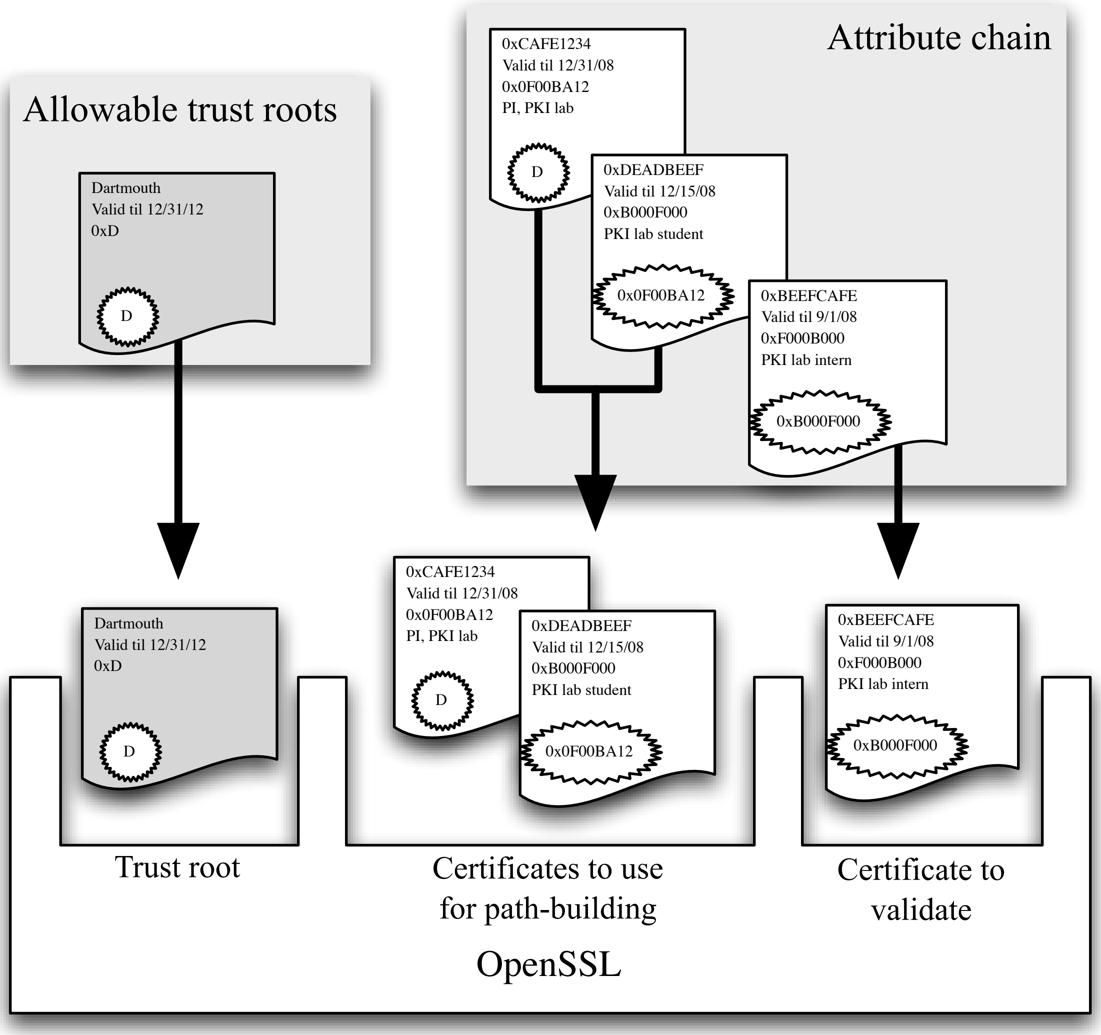


Figure 5.5: We use OpenSSL to validate ABUSE attributes. The terminal assertion in the chain is passed to OpenSSL command line tools to be validated, using the organizational CA as a trust root. The other assertions in the attribute (if any) are passed in to enable the validation tool to rebuild the entire certificate chain. Validation is only concerned with ensuring that it is possible to build a chain of correctly signed certificates connecting the trust root to the certificate being validated. It is up to the user to read the assertions contained within these certificates and decide whether the attribute makes sense.

were not issued to the signer of the message to which they are bound, are not displayed to users. Attributes containing expired (or not-yet-valid) assertions are displayed, but with some special visual cues that help users understand what this status might mean to their trust-decision. We discuss these visual cues in Section 5.4.1.

Validation screens out attributes that cannot be legitimately claimed by the sender. Fabricated attributes and attributes that were issued to someone other than the sender are blocked. Thus, validation is an important part of presenting users with appropriate context information—our second design goal.

5.2.6 Attribute presentation

Presenting an attribute to the user (in support of design goal number two) is primarily a GUI issue. However, there are a few support tasks that the infrastructure must perform in order to provide the interface with all the information it needs to appropriately convey context information to the user. For example, we wish to support the use of process-based trust in ABUSE by calling users' attention to assertions made by people with whom they are familiar. We also wish to downplay information that users have seen frequently in order to avoid *habituation*, a concept discussed at greater length in Chapter 7. Intuitively, the visual impact of our GUI will scale with the novelty of the information ABUSE has to display. Familiar attributes from familiar senders are of the least import; the message recipient already shares some trust relationship with the sender, and so ABUSE is not particularly helpful. Messages from unfamiliar senders, the case in which ABUSE is designed to be most useful, have attributes displayed prominently.

In order to support this behavior, the ABUSE client must keep track of a couple of pieces of information:

- familiarity of message senders, and
- familiarity of attributes.

Familiarity must be defined if we are to programmatically track it. A variety of familiarity metrics make sense here, and the meaning of "familiar" could be different for senders and for attributes.

Familiarity of senders

For senders, we adopt the approach Garfinkel used in his CoPilot system [50], following his "track received keys" pattern (p. 44). If the client has received signed mail from a given sender, it remembers the associated public key and, if it sees that key associated with the signature on a new message, the sender of that message is considered familiar. We could also set some kind of threshold in addition to this metric; three messages received, or perhaps require an actual exchange of multiple messages between the sender and recipient. We could introduce some kind of temporal evaluation as well; perhaps a sender becomes unfamiliar if he has not been heard from in some number of months. It might also make sense to consider senders in the user's address book as familiar, even if they have not sent signed mail before. Their identity certificate (and public key) could be queried from a database associated with the organization's identity PKI in that case.

Determining the best metric for sender familiarity is a question that should be addressed future work. For now, as mentioned above, we follow Garfinkel's "Track received keys" design pattern (p. 44).

Familiarity of attributes

Determining the "right" metric for attribute familiarity is, again, open. We choose a naive metric: if we have seen this exact chain of assertions before, the attribute is considered familiar. It might be the case that users would find it useful for ABUSE to alter its display if an attribute is expressing a familiar kind of relationship. For example, Alice has received many messages from Bob, who binds to his correspondence an attribute from Carol indicating that he is her executive assistant. When Alice receives a message from Darrell with an attribute stating that he is also Carol's executive assistant, it might be useful for ABUSE to somehow convey to Alice that this is a kind of attribute with which she is familiar, though she has not seen this particular chain before.

Tracking familiarity

The ABUSE client caches a sender's public key whenever it sees a validly signed incoming message. Attributes are cached after they are validated. This is fine the first time a message is viewed; unfamiliar senders and attributes are appropriately flagged as such. When a message is viewed again, though, the sender and attributes are already in the cache and will be flagged as familiar. While this could be a legitimate semantic decision, early testing showed that this was confusing to users. Thus, we decided that familiarity status should be "sticky"; the first message from an unfamiliar sender continues to be displayed as such. To support this behavior, we made the ABUSE client add a nonce to the headers it inserts into outgoing messages. Upon receiving a message from an unfamiliar sender, the ABUSE client caches the sender's public key and associates it with the nonce found in the headers. When re-loading messages, the client checks not only whether the sender is familiar, but also whether or not the nonce on the message being loaded matches the nonce stored along with the sender's public key. If so, the message is rendered as being from an unfamiliar sender. We explore the details of this more thoroughly in Section 5.4.1.

Preparing an attribute for presentation

In addition to determining familiarity, the client must also do some work to get the information necessary to present attributes to the user. The presentation GUI that we designed displays each attribute as an entity unto itself. Though it is possible (even probable) that multiple attributes on a given message may share some assertions near the top of the certificate chain, we did not explore visualizing the content in this way—though it may be interesting to explore this in future work. To prepare a single attribute for presentation, the client first determines and stores the name of the attribute's trust root, and then separates the chain into its component assertions. For each assertion, the client then parses out the the subject, issuer, validity period, and assertion content. Remember that subjects and issuers in ABUSE assertions are identified by hashes of their public keys from the organizational identity PKI. The client resolves these key hashes into human-readable names by looking up the appropriate identity certificates.

5.2.7 Towards expressiveness, good signals and usability

Earlier in this chapter, we laid out seven design goals that, if achieved, would help ABUSE exhibit the three characteristics that we require from a solution to the email trust problem we have laid out in this thesis: expressiveness, good signals and usability. We allow users to issue arbitrary digitally signed assertions to each other and distribute them reliably along with email while avoiding both the need for a central administrator and push-back from users without ABUSE-savvy email clients. Designing an architecture that provides these features gets us part of the way towards our goal of a user-centric system that solves the email trust problem we have laid out. Now, we must discuss the GUIs that present context information to users, allow users to issue assertions, and allow users to bind attributes to their outgoing messages.

5.3 Design goals for the ABUSE user interfaces

In support of ABUSE's overall goal, we must design user interfaces capable of ensuring that Carol, when she receives an attribute issued by Alice bound to a message from Bob, can understand what Alice meant to say about Bob. This implies goals at both ends: we must present Bob's attribute to Carol accurately and also help Alice accurately convey what she wants to say about Bob. As we mentioned in Section 5.2, this requires GUIs for attribute issuance, so Alice can accurately express what she wishes to say about Bob; attribute selection, so that Bob can accurately pick the attributes he wants to send with his message; and attribute presentation, so that what Alice said is accurately represented to Carol. When developing design goals for these user interfaces, we must again take into account the characteristics we desire ABUSE to possess: expressiveness, good signals, and usability.

5.3.1 Getting good signals all the way to the end user

We have already discussed in Section 5.2.1 and Section 5.2.2 the ways in which we make it difficult for an attacker to falsify an ABUSE attribute. We cited that, by allowing users to issue assertions to one another, attackers would need to collude with people in the system to generate convincing false attributes. This assumes that the chained nature of ABUSE attributes can be accurately expressed to users; if message recipients do not understand how the assertions in an attribute relate to one another, this opens a door for social engineering attacks. Thus, our first design goal for our user interfaces is to

- make the semantics of assertion chaining clear to users.

Now, we must address the issue of *mimicry*. If an attacker can convincingly mimic the appearance of having an attribute bound to his message, he does not need to create a false chain of certificates at all. It ceases to matter that we have made it difficult to craft a falsified attribute, because the attacker can create the appearance of having done so. We mentioned a small example of this briefly in Section 1.4.1: web-based phishing attacks that include "Verified by VeriSign" or "TRUSTe" icons in fraudulent web pages. [32] Other researchers have also noted similar problems with the padlock icons that web browser use to denote SSL-encrypted connections to web servers. [26, 129] An email client that renders HTML email (Thunderbird, Mail, Outlook, etc.) faces similar challenges; the sender of an attack email can use the formatting capabilities of HTML to craft convincing false signals. To combat mimicry, then, we must

- provide users with a way to reliably distinguish content presented by ABUSE from content delivered in the body of HTML email.

5.3.2 Making the UI expressive

The first goal enumerated above addresses expressiveness concerns as well; chaining semantics are an important part of understanding the meaning of an ABUSE attribute. Also, if people familiar to the user have issued component assertions of an attribute we are presenting, we would like to highlight this. The user may share some pre-existing trust relationship with one or more of the issuers involved in the creation of an attribute; reminding them of this will help better inform their decision. Adding a few more self-explanatory semantic issues, we come up with the following list of concerns that we must address when presenting an attribute:

- the semantics of chaining,
- the presence of familiar users in a chain of assertions,
- the meaning of expired/not-yet-valid assertions, and
- the presence of multiple attributes bound to a message.

We have said before that we allow users to make any assertions they desire. However, it is possible that providing them with suggestions could be useful. Starting with a list of common assertions could reduce the burden of attribute issuance. The size and variety of such a list would vary across domains, and navigating a sizable list could perhaps introduce new usability issues. This requires further study, which we propose in Section 10.3. Intuitively, it seems as though providing suggested attributes could be useful to users, and so it behooves us to include a facility for defining such a set within the system. A similar set up for suggesting attribute validity periods could also be useful. Again, this would be very context dependent. A university scenario could suggest periods such as “until the end of this term” or “fall semester, 2008” while a business setting might provide suggested periods aligned with financial quarters. So, we can add another goal to our list:

- provide a facility for suggested assertions and validity periods.

Practitioners must be careful not to turn this facility into a “policy kit” by providing a myriad of suggestions. As we mentioned above, work should be done to determine where the tradeoff point is between providing a helpful set of suggestions and overwhelming users with choice.

5.3.3 Designing the UI to be usable

In this section, we discuss which of Garfinkel's patterns influenced the usability design goals we set for our GUIs. We also address the patterns that we did not apply anywhere in our design and why we consider them inapplicable. Evaluation of our designs occurs later: we relate the iterative design process we used—along with our prototypes—in Chapter 6, perform a usability analysis in Chapter 7, and provide empirical evidence that users understand our GUIs in Chapter 8 and Chapter 9.

We have already mentioned in Section 5.2.6 that ABUSE follows Garfinkel's "track received keys" pattern (p. 44) by keeping track of sender and attribute familiarity. In order for users to take this information into account, we must

- express sender and attribute familiarity to users in a clear and concise fashion.

When designing the actual look and feel of our GUIs, we must apply the "consistent meaningful vocabulary" and "consistent controls and placement" patterns (p. 42 and 42). Not only must we avoid overloading any pre-existing terminology, but any that we introduce should be used to mean the same things throughout the ABUSE interfaces. For example, we should not tell users that they can "attach" or "include" attributes, because these terms have specific meanings in the email space. We should also endeavor to make our selection of control elements similar to the program into which we build our ABUSE client, as well as consistently using the same representation for the same kind of information throughout our software. Attributes should be represented similarly regardless of the context in which they appear, for instance. Thus, we add the following goals:

- avoid overloading existing terminology/use new vocabulary consistently, and
- be consistent with design and functionality of control elements

Patterns that do not apply to ABUSE

Despite being presented as "designed to advance the goal of secure messaging for all users" [47, p. 330], we consider some of Garfinkel's patterns inapplicable to ABUSE. This is, primarily, because our work is designed to be deployed inside a logical organization and to rely upon an institutional identity PKI—a choice that Garfinkel admits is reasonable where the wherewithal exists to support such a system.

- **Email-Based Identification and Authentication (p. 43)**: As we assume the existence of an identity PKI, we do not require the use of EBIA to set up identities for users in ABUSE.

- **Create keys when needed (p. 44)**: Though we create keys every time a new assertion is issued, we do not use these in cryptographic communication protocols. The only keys we use in that fashion come from the identity PKI.

- **Key continuity management (KCM, p. 44)**: KCM is designed for situations in which no identity PKI is likely to be in existence; as such, we do not employ this pattern.

- **Track recipients (p. 44)**: Any ABUSE-savvy email client is by default also S/MIME compliant. Thus, we have no need to track the message receiving capabilities of recipients.

- **Migrate and backup keys (p. 45)**: Any keys that we create are stored within the centralized attribute store; key escrow for identity keys is outside the scope of our work.

5.3.4 Collecting the GUI design goals

Summing up the design goals for our GUIs, we have the following:

1. make the semantics of assertion chaining clear to users,

2. highlight the presence of familiar users in a chain of assertions,

3. make clear the meaning of expired/not-yet-valid assertions,

4. help users notice when multiple attributes are bound to a message,

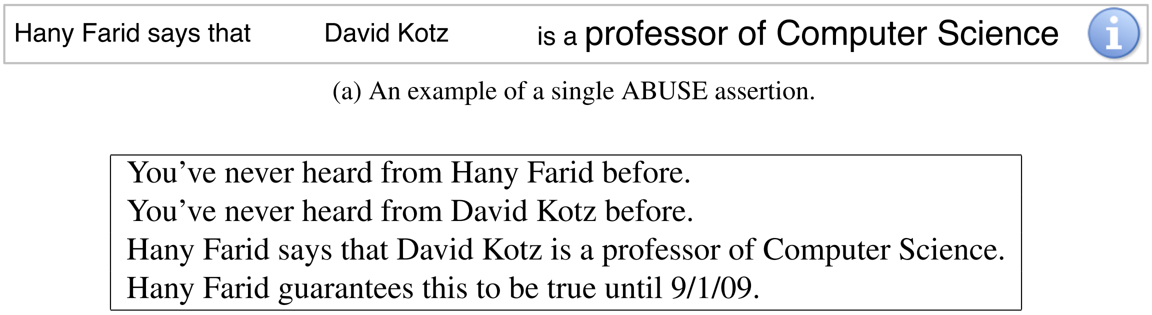
5. provide users with a way to reliably distinguish content presented by ABUSE from content delivered in the body of HTML email,

6. express sender and attribute familiarity to users in a clear and concise fashion,

7. provide a facility for suggested assertions and validity periods,

8. avoid overloading existing terminology/use new vocabulary consistently, and

9. be consistent with design and functionality of control elements.



Hany Farid says that David Kotz is a professor of Computer Science

(a) An example of a single ABUSE assertion.

You've never heard from Hany Farid before.
You've never heard from David Kotz before.
Hany Farid says that David Kotz is a professor of Computer Science.
Hany Farid guarantees this to be true until 9/1/09.

(b) Verbose description of the above assertion.

Figure 5.6: An ABUSE assertion show both in summary form, and in the long form available by clicking on the information icon shown at the right end of (a).

5.4 The ABUSE GUIs

As we mentioned in Section 5.2, the ABUSE client has GUIs for attribute presentation, issuance, and selection (which is a sub-task of distribution). As both issuance and selection require the system to present attributes to users, we first discuss the ABUSE attribute presentation GUI.

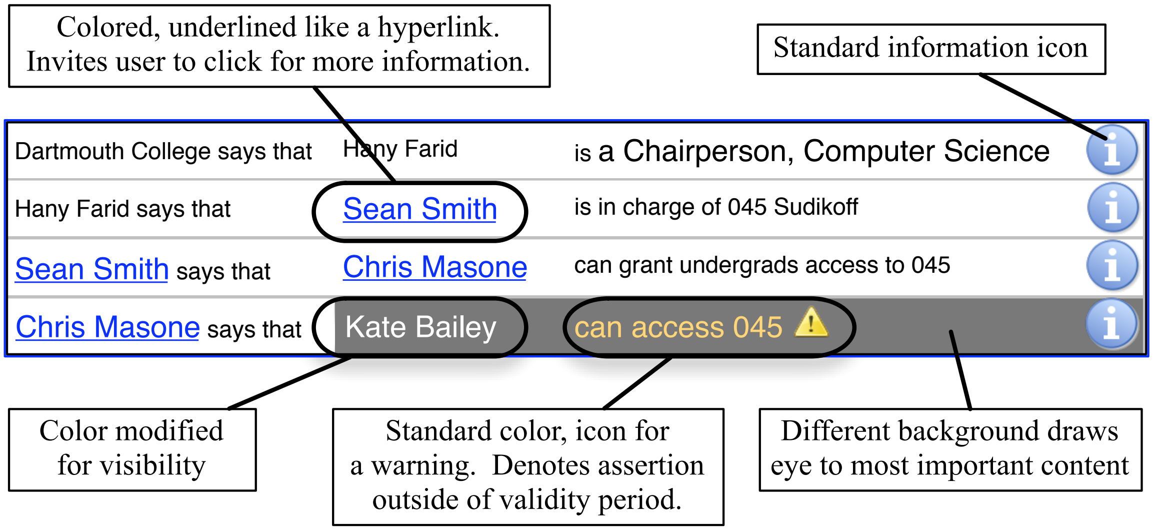
5.4.1 Attribute presentation GUI

A single assertion

Though assertions are never presented on their own, it is convenient to break one out for the purposes of explication. Figure 5.6 shows a single ABUSE assertion, issued by someone unfamiliar to someone unfamiliar. We modify attribute presentation based on the familiarity to the recipient of issuers and subjects in the chain, but will discuss this later. For now, note that we use simple, straightforward vocabulary in the summary form (Figure 5.6a). The content of this assertion is displayed in larger font to draw attention to it; as the names of the issuer and subject are unfamiliar, the content is the most important piece of context presented here. A standard information icon is placed at the end of the row of elements shown. When this icon is clicked or hovered over with the mouse, an expanded version of the information is presented. This *long form*, shown in Figure 5.6b, spells out the meaning of an assertion and provides expiry information. The validity period can also be seen by hovering the mouse over the assertion content.

A single attribute

An attribute is, essentially, displayed as a series of assertions. Figure 5.7 shows a complete attribute. Assertions are stacked in the order in which they are chained, with the first



Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science	i
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff	i
Sean Smith says that	Chris Masone	can grant undergrads access to 045	i
Chris Masone says that	Kate Bailey	can access 045 ⚠	i

Colored, underlined like a hyperlink. Invites user to click for more information.

Standard information icon

Color modified for visibility

Standard color, icon for a warning. Denotes assertion outside of validity period.

Different background draws eye to most important content

Figure 5.7: An example of an ABUSE attribute. Note the use of the standard warning icon in the last row; this denotes that the last assertion in the chain is outside of its validity period. The names of familiar individuals are large and made to appear similar to standard web links. When a name is clicked, any attributes that ABUSE has seen issued to that individual are displayed in a dialog box.

assertion (signed by the trust root) at the top. The ordering, showing the subject of each assertion as the issuer of the next, shows users how assertions are chained into an attribute, meeting our first design goal.

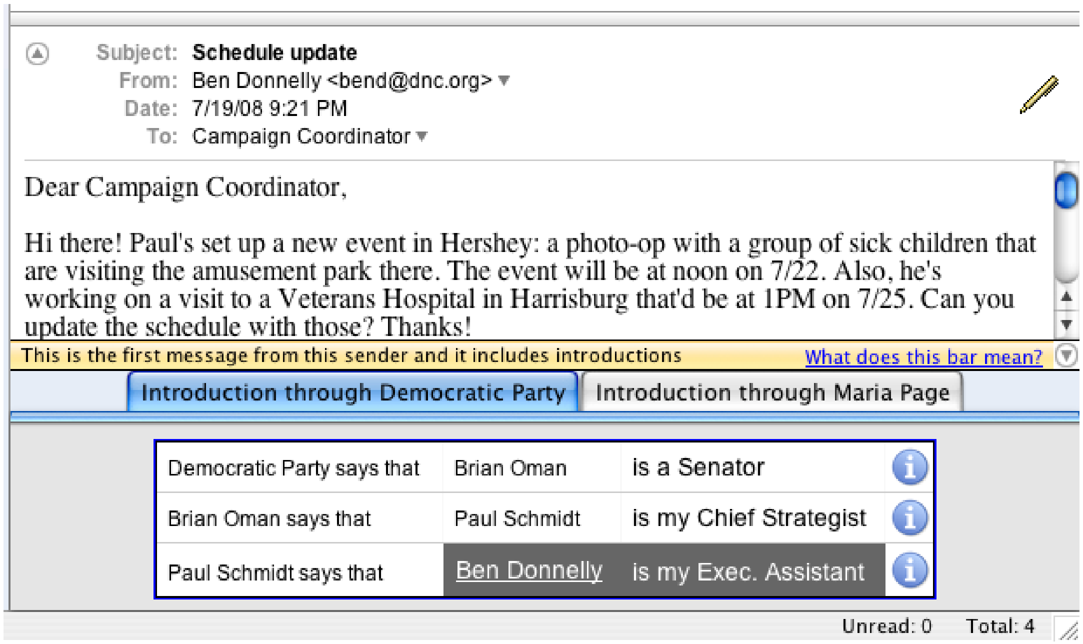
Looking now to our second design goal, familiar users are denoted by enlarging the font used to display their names. In the example shown, "Sean Smith" and "Chris Masone" are familiar to the user. This means that the system knows the user has encountered them before; thus, it likely has cached attributes issued to each man. The system enables the user to access that contextual information directly from this GUI by clicking on their names. It renders the names in the style of hyperlinks in a web page, as users are familiar with clicking on links to get more information in the web-browsing context. This is one specific way in which we worked to remain consistent with existing control metaphors: further information about some item is accessible behind a hyperlink anchored on that item. We also modify the long form (Figure 5.6b) of an assertion when a principal is familiar; instead of "you've never heard from Sean Smith before," we display the phrase "you have communicated with Sean Smith before."[1]

To meet our third goal, the accurate expression of validity period information, we use standard coloring and "warning sign" iconography to indicate to the user when there is some cause for concern. Hovering over the attribute content displays a short message

[1] If the user asks for the long form of the final assertion in a chain the text regarding the subject is subtly different. For the attribute shown in Figure 5.7, as an example, we would display "You are communicating with Kate Bailey right now."

Status	Summary message	Long message
Valid	This statement expires on [date]	[Issuer] guarantees this to be true until [date]
Expired	This statement EXPIRED on [date]	[Issuer] only guaranteed this to be true until [date]
Not yet valid	This statement is NOT VALID until [date]	[Issuer] does not guarantee this to be true until [date]

Table 5.2: Textual explanations of assertion validity status.



Subject: **Schedule update**
From: Ben Donnelly <bend@dnc.org>
Date: 7/19/08 9:21 PM
To: Campaign Coordinator

Dear Campaign Coordinator,

Hi there! Paul's set up a new event in Hershey: a photo-op with a group of sick children that are visiting the amusement park there. The event will be at noon on 7/22. Also, he's working on a visit to a Veterans Hospital in Harrisburg that'd be at 1PM on 7/25. Can you update the schedule with those? Thanks!

This is the first message from this sender and it includes introductions

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Paul Schmidt	is my Chief Strategist
Paul Schmidt says that	Ben Donnelly	is my Exec. Assistant

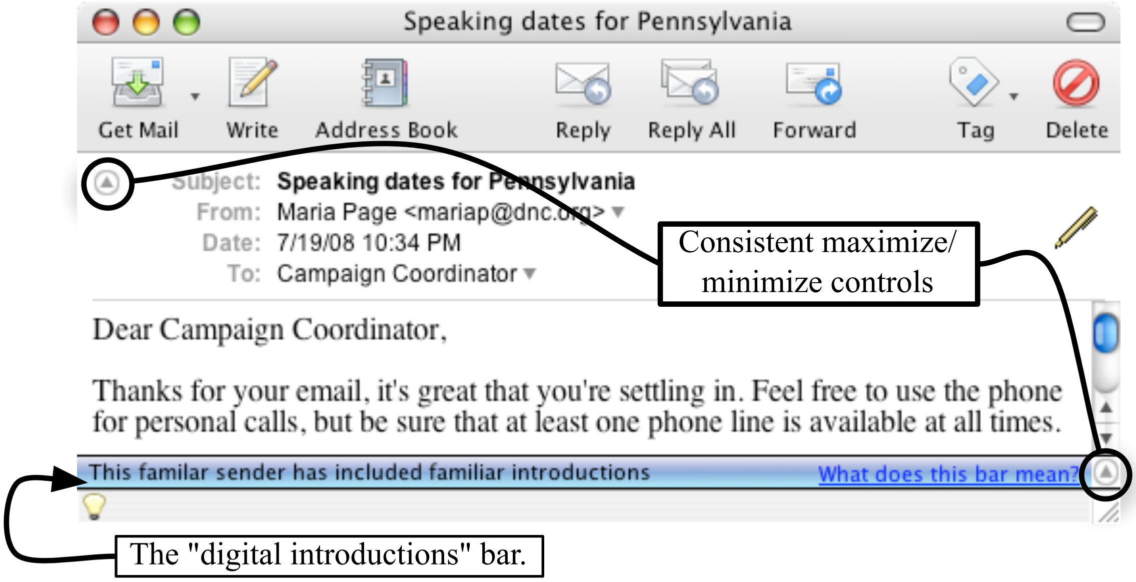
What does this bar mean?
Introduction through Democratic Party
Introduction through Maria Page
Unread: 0
Total: 4

Figure 5.8: The ABUSE prototype is built atop Mozilla Thunderbird. Here we show a pair of attributes, displayed in a part of the Thunderbird email window.

indicating the current validity status of the assertion; if the assertion is not currently valid, the text is correspondingly concerning. The long form of the assertion is similarly modified, as in Table 5.2.

Presenting attributes with email

We toyed with a number of names by which to refer to ABUSE attributes in the prototype client we built. We started with *sender attributes* in the early mockups used in Chapter 8, but some feedback from the subjects in that study led us to re-think our terminology. The term didn't capture the idea that other users are attesting to or vouching for characteristics of or permissions held by the sender. *Digital introductions*, the term upon which we settled, better captures the idea: a third party, with whom the recipient shares some existing trust



Subject: Speaking dates for Pennsylvania
From: Maria Page <mariap@dnc.org>
Date: 7/19/08 10:34 PM
To: Campaign Coordinator

Dear Campaign Coordinator,

Thanks for your email, it's great that you're settling in. Feel free to use the phone for personal calls, but be sure that at least one phone line is available at all times.

This familar sender has included familiar introductions

Figure 5.9: A familiar sender who has included introductions that the system has seen before.

relationship, is providing some reasons to consider extending trust to the sender. So, as Figure 5.8 shows, we display an attribute as an "introduction" through some issuer in the chain of assertions. If none are familiar to the recipient, we just use the trust root. Our solution for goal four, displaying multiple attributes bound to a message, is also shown in Figure 5.8. We tab the attributes, a metaphor familiar to users from web browsers such as Mozilla Firefox, Apple Safari, and the latest version of Microsoft Internet Explorer.

Another key GUI element is visible in Figure 5.8, just above the tabs. This is the *digital introductions bar*, also shown in Figure 5.9. This bar is displayed as a part of the *chrome* for every message, in order to prevent attackers from spoofing it in messages that have no attributes bound to them. *Chrome* refers to the portions of a GUI that do not blindly render user-provided content. The pane displaying the attributes themselves can be maximized and minimized by using a control consistent in appearance with one that performs the same function elsewhere in the same window (pointed out in Figure 5.9). Since the introductions bar is always present, if an attacker attempts to mimic it, the user will see two. If this is not enough to clue users in to an attack, there is existing work on how to better distinguish chrome from rendered HTML [26, 129] that we could apply. This is a potential area for further study.

As we discuss at length in Chapter 7, the prominence of the digital introductions bar and attribute pane are designed to increase as ABUSE decides that the information it has to present to the user is more and more relevant. Figure 5.8 shows a case in which an unfamiliar sender has bound as-yet-unseen attributes to his message; the bar is yellow

Attributes	Color	Pane	Introductions bar text
Present	Yellow	Open	This is the first message from this sender and it includes introductions
None	Yellow	Open	This is the first message from this sender and it includes no introductions

(a) Behavior for unfamiliar senders

Attributes	Color	Pane	Introductions bar text
Familiar	Blue	Closed	This familiar sender has included familiar introductions
Some new	Blue	Open	This familiar sender has included some new introductions
None	Blue	Open	This familiar sender has not included any introductions

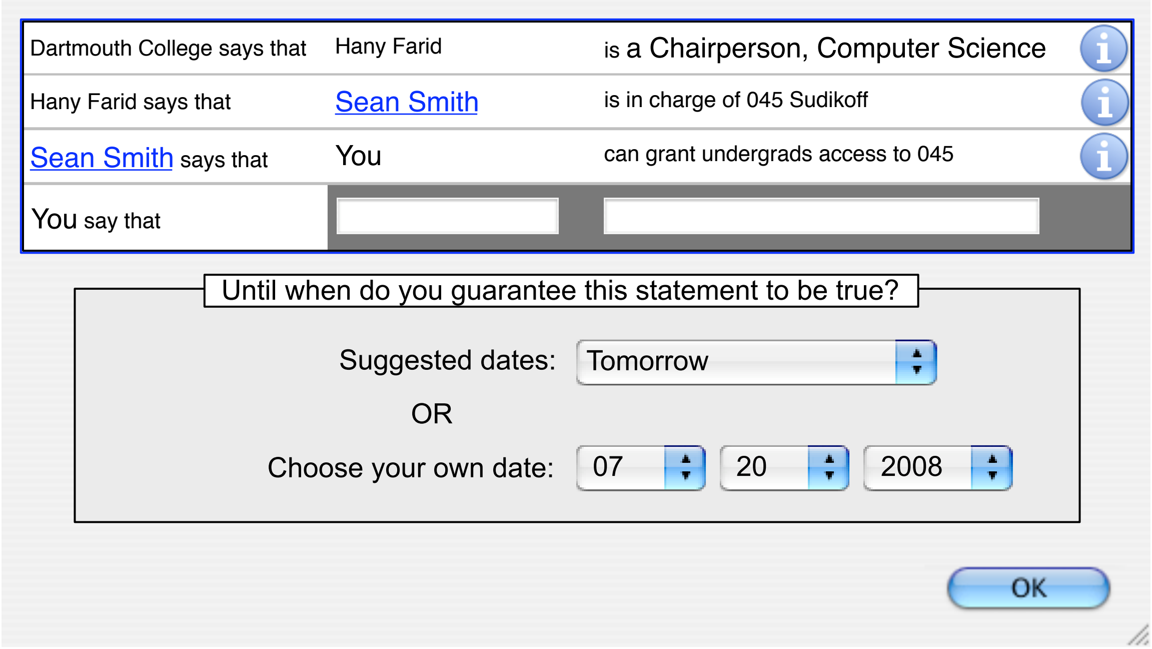
(b) Behavior for familiar senders

Table 5.3: Display options versus sender and attribute familiarity.

and the attributes automatically displayed as a result. In Figure 5.9, a familiar sender has bound to her message attributes that the user has already seen. The bar is thus blue (a neutral color) and the attributes are hidden by default. Table 5.3 details all the cases and the resulting display choices

5.4.2 ABUSE selection GUI

Selecting attributes to bind to outgoing messages is currently a simple process. Alice is the subject of the final assertion in all of her attributes, and presumably knows the person who issued that assertion. ABUSE generates a list of all such issuers, and Alice chooses among them. Assume that she chooses the issuer Bob. Alice is presented with all of her attributes in which Bob issued the final assertion, and indicates which she wishes to bind to her message. She may choose more, or she may choose to be done. It may be useful to implement some kind of search functionality; we cannot know until we see how the attribute space grows in a real-world ABUSE deployment. Regardless, this is more a matter of convenience. Similarly, enabling the user (or the organization) to set a default set of attributes could also be useful, but real-world use patterns are necessary to judge the utility of these features.



Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff
Sean Smith says that	You	can grant undergrads access to 045
You say that		

Until when do you guarantee this statement to be true?

Suggested dates: Tomorrow

OR

Choose your own date: 07 20 2008

Figure 5.10: The ABUSE attribute issuance GUI. Shown here with suggested expiration dates, but not suggested assertion values.

5.4.3 ABUSE issuance GUI

As a part of our desire to be consistent in our representation of information, the GUI for issuing attributes relies heavily on both the attribute selection and presentation GUIs. After Alice picks an attribute off of which she wishes to chain a new assertion (using, essentially, the same GUI discussed in Section 5.4.2), she is presented with Figure 5.10. The subject field auto-completes like an address field in a message composition window in a modern email client; as Alice types the name of the person to whom she wishes to issue this assertion, her address book and the organizational directory are used to help ensure that she unambiguously specifies her subject. Pursuant to our seventh design goal, we provide drop-down menus for both assertion content and validity period in the event that Alice wishes to choose a suggested value for either of these fields. The administrators of the ABUSE deployment can specify lists of suggested assertions and validity periods (XML files, downloaded by the client from the centralized attribute store) to populate these menus. If Alice eschews the content suggestions, putting the skeleton of the attribute ("Alice says that Bob...") in place helps guide Alice to frame the wording of her assertion content in a way that will make sense when viewed later.

Having detailed the entirety of our system architecture and user interface, we now address the design and implementation decisions that we were not able to fit into the flow of the preceding discussion.

5.5 Design and implementation decisions

5.5.1 Attribute revocation

The thorniest design decision we have made revolves around the issue of attribute revocation. This is unsurprising, as revocation is one of the more contentious issues in the PKI community at large. Researchers and security professionals tout a variety of approaches, including: *Certificate Revocation Lists (CRLs)* [19], on-line revocation status checking protocols [43, 81], and leaving it out altogether. [120] The X.509 identity PKI upon which ABUSE relies must have *some* strategy for dealing with this issue, or else it would be incomplete. In cases where an individual is separated from the organization, his identity certificate will be revoked, causing all his attributes to be invalidated. This "nuclear option" aside, the issue of revoking individual attributes is rife with concerns.

Given the plethora of certificates that we expect to be issued in an ABUSE deployment, it is possible that maintaining and distributing an up-to-date CRL would quickly become onerous. This brings us to on-line approaches, which are technically appealing but would not work when one is reading email offline. Enabling either approach introduces the need for users to manage revocation, which brings with it a host of new usability questions and management burden. Leaving revocation out saves us this burden, but it opens up the possibility that there will be periods of time during which a subject can claim an attribute that the issuer would like to revoke. Perhaps Alice has issued to Bob an attribute stating that he works in her office for the month of August. Part way through the month, they have a falling out and Bob is transferred. In ABUSE, Bob can still bind to his messages an attribute that states "Alice says that Bob works for me. Alice guarantees this to be true until 8/31." It may be possible to limit this risk by providing users with a maximum possible validity period, but it is unclear whether we can, as system designers, determine the "right" duration to use. Even if we consider only role-style attributes, we believe that the correct lifetime would vary drastically across deployments. Our contacts in the financial industry [28, 114] tell us that retail banks have very stable role assignments (like "teller"), but that investment banks experience a drastically greater rate of churn. The question of the "correct" validity period duration has been pondered before; we defer to the designers of SPKI, who realized that "the answer...comes from risk management. It will probably be based on expected monetary losses, at least in commercial cases." [36]

Given this set of tradeoffs, we have chosen to provide a facility for setting a maximum validity period, but not provide one by default. The file in which administrators specify suggested validity periods can also specify a maximum period, but it is not required. We have also not designed the system such that it rules out revocation. It is possible that some deployment scenarios would consider it necessary; the "top-level" attributes issued directly by the organization would be prime candidates for revocation in this case. For this prototype, we have chosen to avoid the extra usability and management burdens of revocation. We propose in Section 10.2 to study the efficacy with which short-lived certificates and well-chosen suggested validity periods can manage the inevitable risk incurred by users being able to claim attributes to which they should no longer have a right.

5.5.2 Implementation decisions

As mentioned briefly in Section 5.4, we have chosen to build the ABUSE client prototype on top of Mozilla Thunderbird. The reason for this is that Mozilla applications support an extension framework that makes it relatively easy to add functionality. As Thunderbird is fully S/MIME compliant, it has much of the functionality that we needed to build ABUSE. Proxy Certificates are not well supported by Mozilla software, however, so we use OpenSSL [90] instead for those pieces of functionality. OpenSSL supports PCs with a few simple configuration tweaks.

As mentioned in Section 5.2.3, we have implemented the central attribute store with an LDAP database (OpenLDAP [89]). One reason for this choice is that Thunderbird provides LDAP querying functionality natively; a second is that it supports client-side SSL authentication.

5.6 Wrapup

In this chapter, we have laid out the architecture and user interface of ABUSE and shown how it meets the criteria we laid out for a solution to the problem of deciding trust in unfamiliar email correspondents. The system is expressive, and we discuss in Chapter 9 how it can be used for all of the flows enumerated in Chapter 2. The signals (attributes) are difficult for an attacker to emit while being easy for legitimate senders to bind to their outgoing messages. Finally, we have explained how design patterns from the HCISEC space guided our work. We evaluate the usability of our GUIs in Chapter 7. In the next chapter, we discuss the iterative, user-focused process that we used to refine the GUI for attribute presentation, central as it is to the entire ABUSE user experience.

Chapter 6

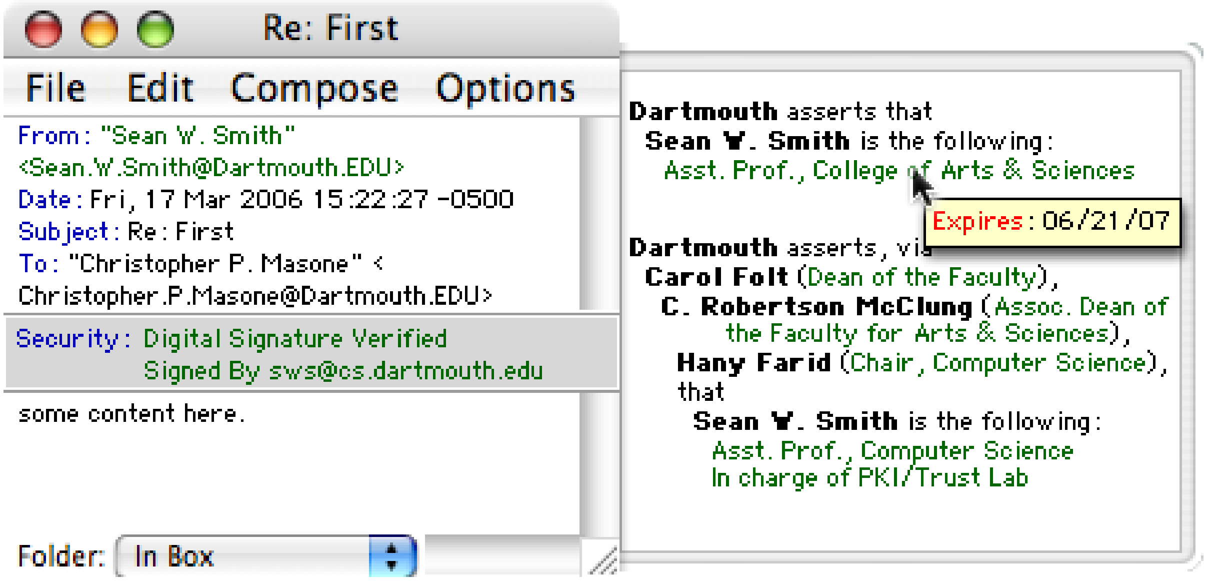
Designing the Attribute Presentation UI

All three user interfaces used in ABUSE rely on the ability to present an attribute to the user in a comprehensible fashion. We sought to optimize the psychological acceptability [103] of our interface by following an iterative design process [85], focusing at each step on understanding the flaws in each candidate representation from the user's point of view. [137] Our resulting GUI is the result of this process.

6.1 First attempts

When first we proposed ABUSE, we hoped to deploy our system here at Dartmouth in order to collect data on a large population of users. To facilitate this, our initial designs had us building ABUSE functionality into BlitzMail, an email client developed at Dartmouth that enjoys widespread among the student population. One of our early mockups is shown in Figure 6.1. Assertion chaining is shown using indentation, and the attributes are protected from mimicry by placing them in a tray-style UI element to the side of the message window. The names of subjects and issuers are bold, but there is no distinction between familiar and unfamiliar users. The content of the final assertion in the attribute is expressed clearly, but non-terminal assertions are somewhat unclear. Issuers do not appear on the same line as subjects, and the attribute content is place in parentheses next to the subject's name. Assertion validity status is indicated by the color of the text representing the assertion content. Green means valid, and amber would mean that the assertion is outside of its validity period.

For a variety of reasons, we abandoned our plans to deploy ABUSE here on campus and were thus no longer tied to the BlitzMail platform. Primarily, the issue is that the Dartmouth community is too close-knit. Many people, especially faculty and staff, know each other already. Thus, it is unlikely that users would frequently find themselves in situations where



From: "Sean W. Smith" <Sean.W.Smith@Dartmouth.EDU>
Date: Fri, 17 Mar 2006 15:22:27 -0500
Subject: Re: First
To: "Christopher P. Masone" < Christopher.P.Masone@Dartmouth.EDU>

Security: Digital Signature Verified
Signed By sws@cs.dartmouth.edu

some content here.

Folder: In Box

Dartmouth asserts that
Sean W. Smith is the following:
Asst. Prof., College of Arts & Sciences

Expires: 06/21/07

Dartmouth asserts, via
Carol Folt (Dean of the Faculty),
C. Robertson McClung (Assoc. Dean of the Faculty for Arts & Sciences),
Hany Farid (Chair, Computer Science),
that
Sean W. Smith is the following:
Asst. Prof., Computer Science
In charge of PKI/Trust Lab

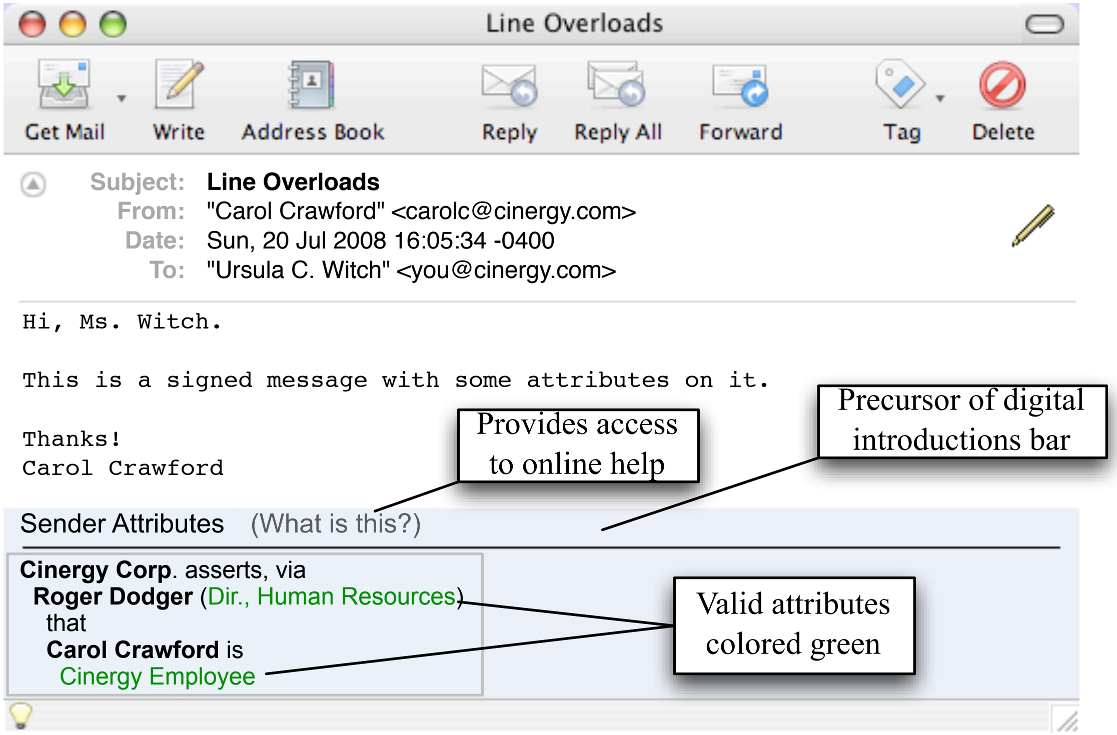
Figure 6.1: The initial UI mockup, dating back to when we planned to build ABUSE on top of Dartmouth's homegrown email client, BlitzMail. The attributes are in the tray to the right.

ABUSE would help. After abandoning BlitzMail, we chose Mozilla Thunderbird, for the reasons discussed in Section 5.5.2. The next mockup, used to carry out the experiment detailed in Chapter 8, is based on this new client platform and shown in Figure 6.2. The most obvious difference between these two mockups (aside from the change to Thunderbird from BlitzMail) is the absence of the tray element. This element is not in the Mozilla UI toolkit, so we had to move to a different display option. Note the precursor of the digital introductions bar, with a first pass at a UI element that provides in-program help for the user (the "What is this?" text element). The formatting of attribute content remains unchanged from the first mockup to this one. At this point, we had already settled on having some ABUSE GUI element be present in the email message reading window at all times; this evolved without user input into the digital introductions bar discussed in Section 5.4.1.

Before our power-grid-inspired user study, we performed a number of trial runs to test out the study interface, protocol and infrastructure. The mock-up we were using at that time was almost exactly the same as the one shown in Figure 6.2; it lacked only the on-line help functionality. In addition to smoothing out some issues with the study, we also got valuable feedback from the pre-testers about the mockup. Pursuant to these discussion, we developed a number of concerns:

1. Some users initially thought that "Hany Farid (Chair, Computer Science)" meant that Hany Farid was asserting that someone else was the Chair of Computer Science.

2. Indenting did not reliably convey the notion of chaining; some users initially per-



Subject: **Line Overloads**
From: "Carol Crawford" <carolc@cinergy.com>
Date: Sun, 20 Jul 2008 16:05:34 -0400
To: "Ursula C. Witch" <you@cinergy.com>

```
Hi, Ms. Witch.

This is a signed message with some attributes on it.

Thanks!
Carol Crawford
```

Sender Attributes (What is this?)

Cinergy Corp. asserts, via
Roger Dodger (Dir., Human Resources)
that
Carol Crawford is
Cinergy Employee

Provides access to online help

Precursor of digital introductions bar

Valid attributes colored green

Figure 6.2: The first mockup based on Mozilla Thunderbird. The tray element is gone, as it is not available in the Mozilla UI toolkit.

ceived the sequence of assertions as a simple list.

3. Our usage of “asserts, via” sometimes contributed to the perception of the chain as a list.

4. Users did not naturally mouse-over or click on UI elements about which they were confused; things needed to “look more clickable.”

5. Multiple attributes placed next to one another caused the UI to become crowded quickly.

As every pre-tester indicated that they had eventually understood the meaning of the attributes, we felt comfortable going ahead with the study after adding some on-line help. The feedback they provided, however, sent us back to the drawing board to redesign the presentation GUI from scratch.

6.2 Getting on the right track

The first decision we made upon returning to the drawing board was that we would not try to display more than one attribute at a time. We would instead use tabs to allow users to flip through a set of attributes. This freed us from the space constraints under which we had placed ourselves and allowed us to make decisions based solely on which designs tested best with our small pool of pre-testers.

Remember from Section 5.3.2 that our goals for the presentation of a single attribute were as follows:

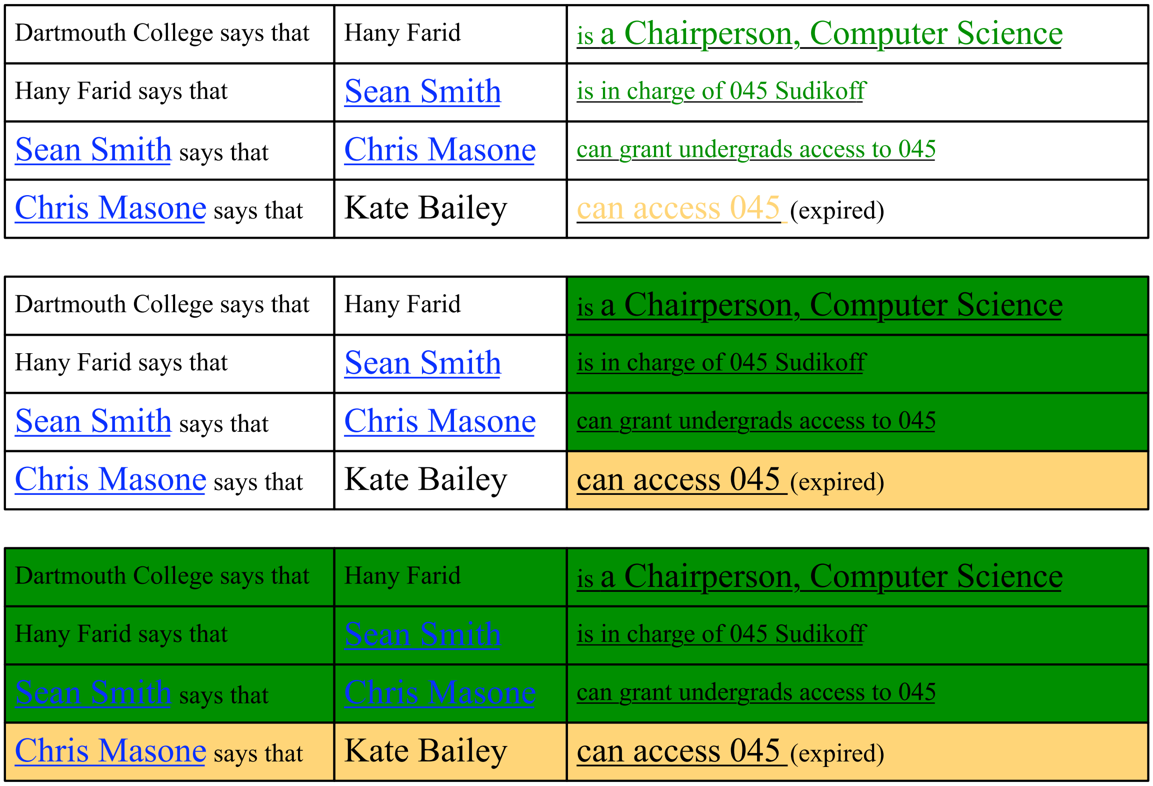
- make the semantics of assertion chaining clear to users,
- highlight the presence of familiar users in a chain of assertions,
- make clear the meaning of expired/not-yet-valid assertions,

We also wanted to provide access to lots of on-line help from this interface; there should be information about each assertion as well as any information the system has regarding any of the principals involved in the attribute. The elements providing this access should look "clickable", so that users realize they are controls and not simple labels. This thought process led us to the idea of using the visual cues used by web links; users are quite familiar with the concept of clicking on things that look like links. So, we will style the names of familiar principals as links, providing access to information about these individuals when their names are clicked. We also chose to apply similar styling to the content of each assertion, meaning to signify that more information (long-form explication, validity period) would be made available upon clicking.

In addition to making controls appear more clickable, we also wanted a way to make certain pieces of information more prominent. We settled on the idea of using varying font sizes to draw the user's attention to important text. In all of the following prototypes, there are the three cases in which we choose to enlarge text:

- when a subject or issuer is familiar to the user, we enlarge his name.
- when neither the subject nor issuer is familiar to the user, we enlarge the assertion content.
- for the last assertion in the chain, we always enlarge the attribute content.

The final assertion in an attribute is the upshot of the whole thing, after all, so it makes sense to enlarge this no matter what. These decisions, in addition to choosing to stick with color as a way of representing validity status led us to three initial prototypes, shown in Figure 6.3, which we sent to our pre-testers. We first asked them to tell us what information was being represented, and then provided them with the following questions to answer about each prototype:



Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff
Sean Smith says that	Chris Masone	can grant undergrads access to 045
Chris Masone says that	Kate Bailey	can access 045 (expired)

Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff
Sean Smith says that	Chris Masone	can grant undergrads access to 045
Chris Masone says that	Kate Bailey	can access 045 (expired)

Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff
Sean Smith says that	Chris Masone	can grant undergrads access to 045
Chris Masone says that	Kate Bailey	can access 045 (expired)

Figure 6.3: Our initial attempts at a revamped prototype. We have moved to a tabular format, to emphasize that we are expressing a series of transitive statements; "A says B about C, C says D about E" and so on. We continue to use color to indicate validity status, and have added link-styling to elements that we wish the users to perceive as clickable.

1. To what is your attention drawn?
2. Which elements would you click on?
3. What information would you expect to see when you clicked on those elements?

We were surprised that some of the testers reported that they had initially parsed the display as being organized by columns; this was the first thing we addressed in our next prototype. In general, though, they perceived the presented attribute as a sequence of transitive statements; "A says B about C, C says D about E" and so on. Our new design had already out-performed our initial attempts at expressing the semantics of chaining. Thus, we decided to stick with this tabular presentation going forward.

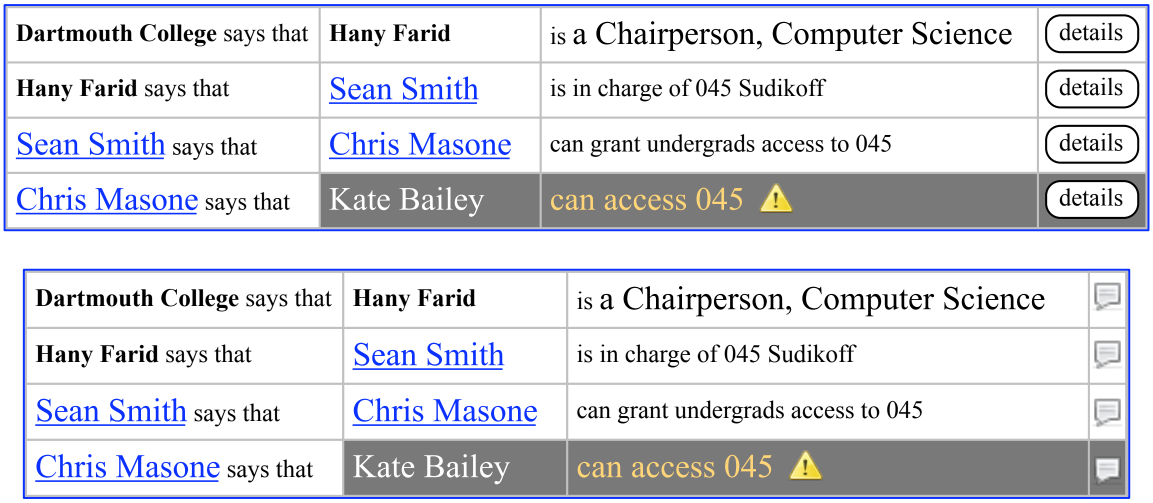
The responses to our questions were fairly uniform across both pre-testers and the first two prototypes. Few responded well to the third prototype; the over-use of color was deemed too distracting. "Sean Smith" and "Chris Masone" were the most frequently noted pieces of information, followed by "can access 045." The link-style formatting worked well in one case; users thought that if they clicked on the names, they would get more information about those people, which is exactly what we hoped to see. However, the testers indicated to us that we had erred in using the same styling on the attribute content. They believed, perfectly logically, that clicking on those links would provide insight into the meaning of the content itself; where 045 Sudikoff is, or what the duties of the Computer Science Chairperson are. They also expressed confusion as to why there were links of several different colors; the connection between color and validity status had been lost. In discussions with the testers, we came to the conclusion that there is actually no real need to color attributes that are valid; we expect attributes to be valid. We should reserve the use of color for information to which we wish to call special attention. We realized one last flaw on our own: using the text "(expired)" to denote an assertion that is outside its validity period is inaccurate; an assertion may also be not-yet-valid.

From our first round of prototypes, we found several choices that were worth keeping:

- the tabular format (though make sure users scan horizontally and not vertically),
- link-styling on the names of familiar people, and
- yellow/amber coloring on non-valid attributes, as this is an anomalous case.

We also discovered a number of problems that needed to be addressed in the next iteration:

- using too many different colors with different meanings is confusing,



Dartmouth College says that	**Hany Farid**	is a Chairperson, Computer Science	details
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff	details
Sean Smith says that	Chris Masone	can grant undergrads access to 045	details
Chris Masone says that	Kate Bailey	can access 045 ⚠	details

Dartmouth College says that	**Hany Farid**	is a Chairperson, Computer Science	💬
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff	💬
Sean Smith says that	Chris Masone	can grant undergrads access to 045	💬
Chris Masone says that	Kate Bailey	can access 045 ⚠	💬

Figure 6.4: The second set of designs. Dividing lines between cells are de-emphasized, and a dark gray background is used to draw attention to the subject and content of the final attribute.

- link-styling on assertion content did not express that the available information would be about the entirety of the assertion, and
- using the term “expired” is inaccurate and must be changed.

In addition to these changes, we also decided that we wanted to draw more attention to the subject and content of the final assertion.

6.3 Moving forward

In order to help users parse the tabular format we are using as a set of rows, we de-emphasized the dividing lines between elements, as shown in both designs in Figure 6.4. Furthermore, we need to come up with a new way to express to the user that extra information is available about each assertion—and show them where to click to see that information. In the second set of prototypes, we have added a fourth column to the table; one design places a button in each row with the label “details”, while the other uses an icon that is reminiscent of an information bubble. We have also gotten rid of the coloring on valid assertions; as we said above, this is what we expect to happen. Having more content in black makes the amber text of invalid assertions and the blue of familiar principals pop out more. In the interest of clearing out extraneous text and making error cases more visually noteworthy, we have replaced “(expired)” with a standard warning icon. The detailed validity information can be accessed by mousing over or clicking on this warning icon. As Garfinkel recommended, we chose iconography consistent with that used elsewhere; in retrospect, we should have done the same when trying to choose an icon for our information button. We did eventually, but not until our final prototype, the one discussed in Chapter 5.

In addition to making the above changes in an attempt to address the flaws our testers found in the first set of prototypes, we also added a dark gray background to the cells containing the subject and content of the final assertion in the attribute. The idea is to call extra attention to this, the most pertinent part of the attribute. We also bolded the names of unfamiliar subjects and issuers, to distinguish their names from the boilerplate text.

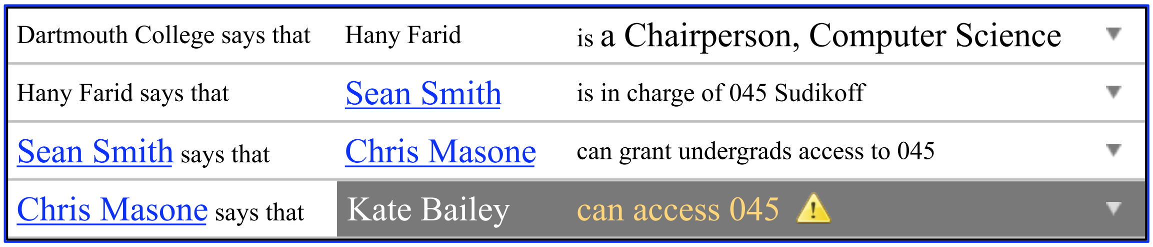
We selected a new set of pre-testers and sent them our new prototypes, with the same questions we asked the first group. Again, some indicated that they initially parsed the tabular layout as being a set of columns before reading the text and understanding the presented information. This group noted that their attention was drawn to the familiar principals and to the invalid attribute, though they did not use this terminology. They pointed out that the bottom row appeared to be important, and also said that they noticed the "Chairperson, Computer Science" assertion; clearing out the extra color and link-styling seems to have helped make that information more noticeable, which is what we wanted to see. This group indicated that they would also click on the link-styled text; they added that they would click on the "details" button, but did not recognize the information bubble icon as something that would be worth clicking on. Again, the pre-testers seemed to believe that the information button would provide further detail about the assertion content directly to its left. No one reacted positively to the bolding of sender and issuer names. From this iteration, we determined to keep the following features:

- valid attributes are written in black and are not link-styled,
- dark gray background in the final assertion,
- big text on assertions without familiar principals, and
- common warning iconography.

We decided to drop the bolded text (as it made no difference), make the table appear still more row-oriented, and continue our search for a better "more information" button.

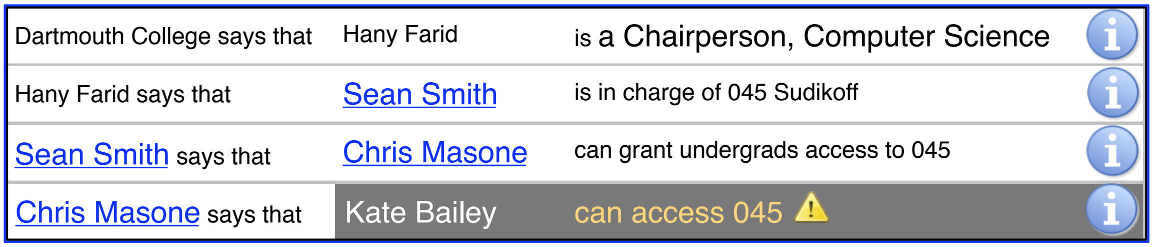
6.4 The final steps

In our third design, we nearly obliterate the cell dividing lines within rows. After this change, our third group of testers all to perceived the attribute in a row-wise fashion. We had also added a downwards triangle to the end of the row, trying to ape the appearance of



Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science	▼
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff	▼
Sean Smith says that	Chris Masone	can grant undergrads access to 045	▼
Chris Masone says that	Kate Bailey	can access 045 ⚠	▼

Figure 6.5: The penultimate design. Dividing lines between cells within a row are almost wholly erased, and we make another attempt at finding the right "more information" icon.



Dartmouth College says that	Hany Farid	is a Chairperson, Computer Science	ⓘ
Hany Farid says that	Sean Smith	is in charge of 045 Sudikoff	ⓘ
Sean Smith says that	Chris Masone	can grant undergrads access to 045	ⓘ
Chris Masone says that	Kate Bailey	can access 045 ⚠	ⓘ

Figure 6.6: The final design, which we discussed at length in Section 5.4.1.

drop-down menus and express to the user that more information would be available upon clicking. This, once again, did not work. The third group agreed with their predecessors about what appeared important and what seemed clickable. In response, we explored a number of GUIs in search of consensus on a kind of icon to use for this purpose—something we should have done in the first place. We discovered that many used a stylized letter "i" in a blue circle; putting this in our GUI seems to have given us the results we desire. We have already discussed the final design at length in Section 5.4.1.

6.5 Wrapping up

In this chapter, we have described our application of a user-centric, iterative design process to develop the GUI most central to the use of ABUSE. We now move on to analyze our user interfaces using Cranor's Human-in-the-Loop framework [21], a tool used to evaluate security systems which, like ABUSE, build humans in.

Chapter 7

Analyzing ABUSE

We have already discussed in Chapter 5 and Chapter 6 the ways in which we designed the ABUSE architecture and user interfaces according to principles laid out by the HCISEC community. Now, to analyze our work, we turn to another tool provided by usable security research: Cranor's *human-in-the-loop* framework.

7.1 Reasoning about the human in the loop

While it may seem ideal to automate every security-critical function in every piece of software, this is sometimes simply infeasible—or even undesirable. [34,41] There exist tasks at which humans are simply better (noting "suspicious behavior", for example), and at other times decisions rely on a level of human context that is difficult to capture and encode. Since we are left with no choice but to rely on humans for some security-critical decisions, Cranor advances her framework as a way to "systematically analyze the human role in a secure system to identify potential failure modes and find ways to reduce the likelihood of failure." [21] Built on top of a simple communication-processing framework borrowed from warnings science literature, Cranor's framework is centered on the idea that the secure system is attempting to send a communication to a non-malicious human receiver in order to trigger some behavior. These communications could be in the form of a status icon, a training manual, a configuration wizard, security policies, notification dialogs, or anything else that can be considered as some component of a system attempting to convey security information to a user. A number of factors can influence the communication-to-behavior pathway, at a variety of points in the process. Cranor's framework describes a number of stages that make up the reception and processing of a communication by a receiver and discusses them individually, noting pitfalls and considerations for each. It is possible that not every stage will occur in every communication, though some factors could impact sev-

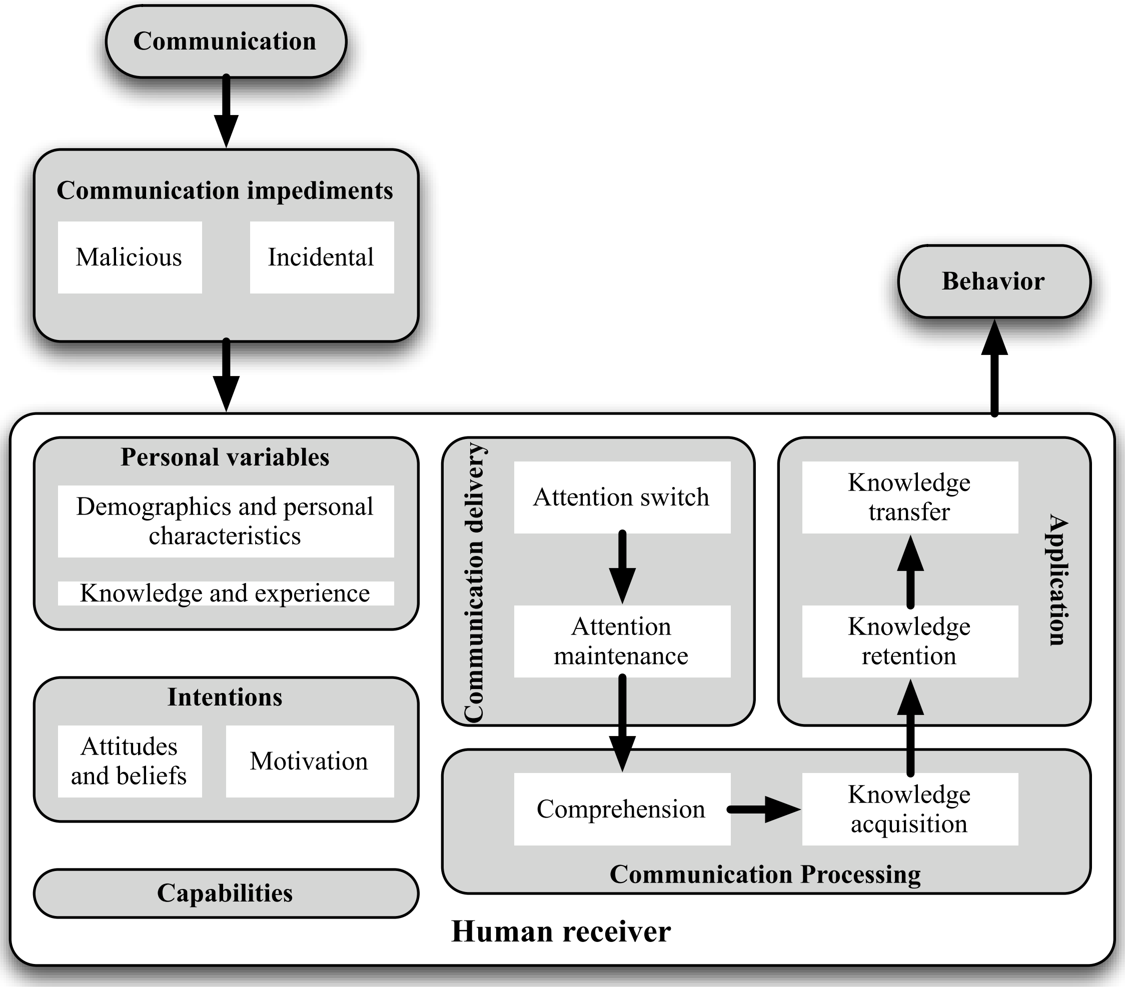


Figure 7.1: Slightly modified diagram of Cranor's human-in-the-loop framework. We have redefined the components of the "communication impediments" stage. Adapted from [21].

eral different stages. Using this framework to discuss ABUSE provides ample opportunity for confusion, since ABUSE is a tool for facilitating communication between people. In Cranor's terminology, ABUSE would be said to be ”communicating" to a human receiver about a communication (email) that he's received from some other human being. To alleviate this problem, we refer throughout this chapter to information sent from one human to another as *messages*, while any signals provided by ABUSE for the consumption of a human will be referred to as *communication*.

An overview of each stage in Cranor's framework and its attendant issues follows. The material presented here is paraphrased from Cranor's original paper on the framework [21], which contains a much more thorough description.

7.1.1 Before Reception

The Communication

The first thing to consider in our analysis is the communication itself. Cranor breaks the space of possible communication types into several categories:

- **Status Indicators** - express a small set of possible states (e.g. an SSL lock icon)
- **Notice** - for expressing a richer set of characteristics (e.g. SSL certificate or privacy policy)
- **Warning** - an alert requiring immediate action (these should be a last resort)
- **Training** - communication used to convey and promote retention of information (e.g. teaching wizards)
- **Policy** - a policy statement, perhaps as a part of a training regimen

Communications can also range from fully passive (in no way interrupts the receiver's workflow) to fully active (the receiver cannot do *anything* until he performs the action demanded by the communication).

The system designer must weigh a number of factors when deciding on what kind of communication to use, and where to place it on the passive–active spectrum. Severity of the threat, expected frequency of threat conditions, and the complexity of the user action required to avoid danger all come into play.

Impediments

Cranor classifies any factors that could negatively impact the receiver's ability to perceive a communication as *impediments*. She breaks these impediments into two categories: *environmental stimuli* that may divert the receiver's attention from the communication wholesale and *interference* which may obscure the communication in some fashion. Interference may be intentional or coincidental (environmental stimuli can cause interference as well), but the idea is that the receiver has missed some part of the communication. We consider this distinction to be somewhat awkward, as environmental stimuli can be discussed in both categories. Instead, we posit *malicious* versus *incidental* impediments as a more useful distinction. Adversarial behavior that in any way impedes reception of the communication as the sender intends would fall into the former category. Any other stimulus (other tasks, other communications, ambient noise or light, etc.) or technological failure not caused by an adversary that obscures or prevents reception of the communication falls into the latter. Malicious impediments are the kind that are commonly considered in security literature: whether the system can be made to lie, can be spoofed so that it appears to lie or mislead, can be manipulated to provide enough false positives or negatives that the user ceases to trust it, or can be in some way obscured or removed from the receiver's experience. Incidental impediments raise a similar set of questions, but rather than looking for malicious input that may trigger these behaviors, the designer must consider a wider set of factors. For instance, when designing an active warning for a system in a steel plant, the designer must keep in mind that audio notification is likely to be obscured. If the receiver is required to field multiple communications per minute, a passive status indicator is likely to be overlooked. That said, if the designer chooses to use an active communication, the system had better be correct—otherwise the user will ignore, work around, or disable it in order to get her job done. [118, 125]

7.1.2 The Human Receiver

Once the communication gets to the receiver, there are two classes of considerations with which a system designer must be concerned:

- factors involved in the actual reception and processing of the incoming communication, and
- characteristics of the receiver that can color his ability to process and act on the communication.

Communication delivery, *communication processing*, and *application* make up the first group, while the latter consists of *personal variables*, *intentions* and *capabilities*.

Communication Delivery

The delivery of a communication consists of two steps, *attention switch* and *attention maintenance*. First, the communication must attract the receiver's attention and then keep his focus long enough to be understood. Both of these steps can obviously be impacted by the impediments of Section 7.1.1, but they are also vulnerable to *habituation*, the tendency for users to pay less attention to stimuli they experience frequently. The color, font, size, motion, sound and other characteristics of the communication come into play here, though the desire to increase the odds of delivery must be balanced against the potential for annoying the receiver—the impact of which is addressed in Section 7.1.2. It has traditionally been easy to overlook a failed delivery as a source of design error; after all, the communication was sent as the system architect intended. The successful sending of a communication does not mean that it was successfully received, however. [27, 31, 124]

Communication Processing

Communication processing can only happen provided that attention maintenance has been achieved. The receiver must focus on the communication long enough to consume the content, and then actually be able to understand it. This is a tremendous challenge for security software, as evidenced by users' difficulties with browser security warnings and indicators [31, 44], as well as our own experiences here at Dartmouth transitioning users to the PKI-authenticated secure wireless network. After comprehension, the receiver must also be able to figure out what to do in response to the communication, which may need to happen over time or with the help of training. Cranor refers to this second stage as *knowledge acquisition*.

Application

If a communication is not present in all cases in which it may be applicable (like training or a policy), the receiver will need to both recognize when the communication would be relevant, and then remember it and figure out how to apply it. Cranor calls these *knowledge retention* and *knowledge transfer*, respectively. Consider a training dialog that discusses the meaning of an SSL lock icon in a web browser. This training communication will not be present every time the user navigates to an SSL-secured web page, and thus she must realize that the training is relevant in this case, remember the meanings of the various icon states, and decide how to behave based on what she sees.

Personal Variables, Intentions and Capabilities

These three groups of factors are all internal to the receiver. There is some overlap among them, so we address them as a group. *Personal variables* include the receiver's age, gender, education, culture, occupation, and so on. All of these factors can play into the receiver's comprehension of an incoming communication. As ABUSE is part of an email system, no real assumptions can be made about personal variables. Since it is designed for use inside organizations, we can assume that receivers will have some knowledge and experience relevant to interpreting the content of attributes, but not to interpreting ABUSE communications as a whole. For example, deploying ABUSE in a power company would allow us to make some useful assumptions about the occupation of users *in that deployment*, but we cannot make these assumptions when analyzing ABUSE in the general case. Thus, we do not consider personal variables elsewhere in this discussion.

Even assuming that a receiver has taken in, processed and understood a communication, designers must still consider his *intentions*. He might choose to ignore a communication, guided by his *attitudes and beliefs* as well as his *motivation*. There are a variety of attitudes and beliefs that might impact this decision. Among other things, the receiver might:

- be annoyed by a communication,
- not trust the communication,
- feel as though he cannot complete a recommended action,
- feel as though recommended actions do not help, or
- believe that the action will take too long. [13]

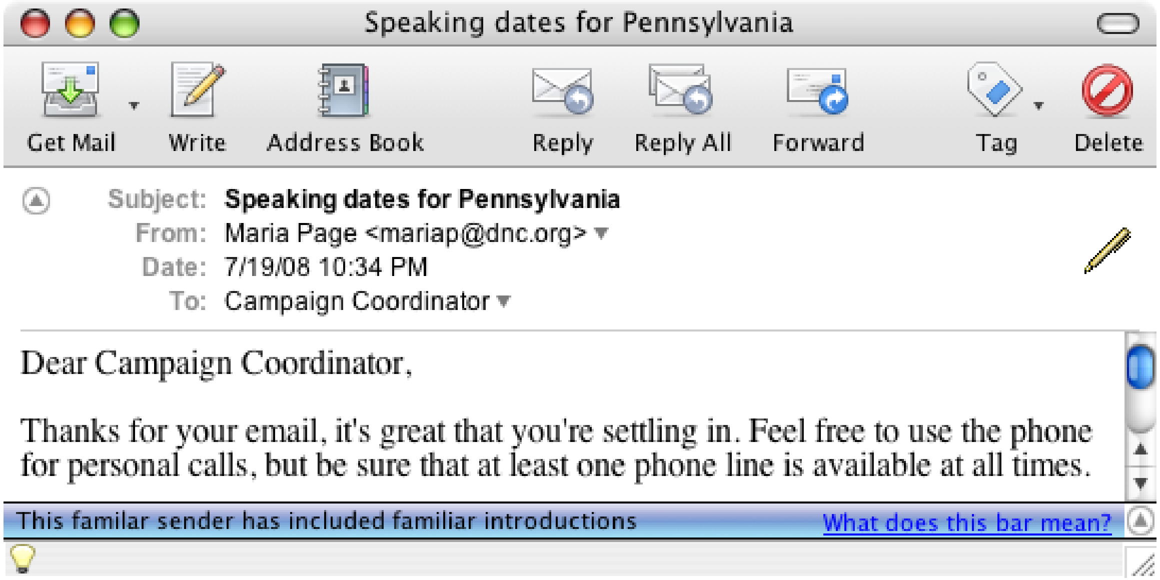
Many factors can influence these attitudes and beliefs, not the least of which is prior experience with the system. *Motivation* deals with the incentives the receiver has to comply with the communication. Since the right “security behavior” often distracts from the receiver’s task at hand [125, 126], motivation can be difficult to provide. Research is actively being done in this area (Section 4.3.1), and goes beyond the scope of ABUSE.

Lastly, the designer must consider whether the receiver has the proper *capabilities* to perform an action recommended by a security communication. Even assuming he decides to exhibit the proper behavior, it remains possible that the system may not enable him to do so.

7.1.3 Behavior

The ultimate goal of a communication is to elicit a desired behavior. There is already a body of work examining what can go wrong between a receiver deciding to behave in the desired way and correctly achieving his goal. A *Gulf of Execution* can exist, between the user’s intentions to do something and the mechanisms provided by the system to help him. [85] A user may also achieve the desired outcome, but be unable to determine that he did so—a *Gulf of Evaluation*. [85] Users may make *mistakes* (plans that will not achieve the desired goal), *lapses* (skipping a required step), and *slips* (performing an action incorrectly). [97]

Having explained Cranor’s framework in full, we now move on to analyze the three ABUSE GUIs introduced in Section 5.4.



Subject: **Speaking dates for Pennsylvania**
From: Maria Page <mariap@dnc.org>
Date: 7/19/08 10:34 PM
To: Campaign Coordinator

Dear Campaign Coordinator,

Thanks for your email, it's great that you're settling in. Feel free to use the phone for personal calls, but be sure that at least one phone line is available at all times.

This familar sender has included familiar introductions What does this bar mean?

Figure 7.2: The attribute presentation GUI at its most passive. The digital introductions bar is a neutral color and the attribute pane is minimized by default.

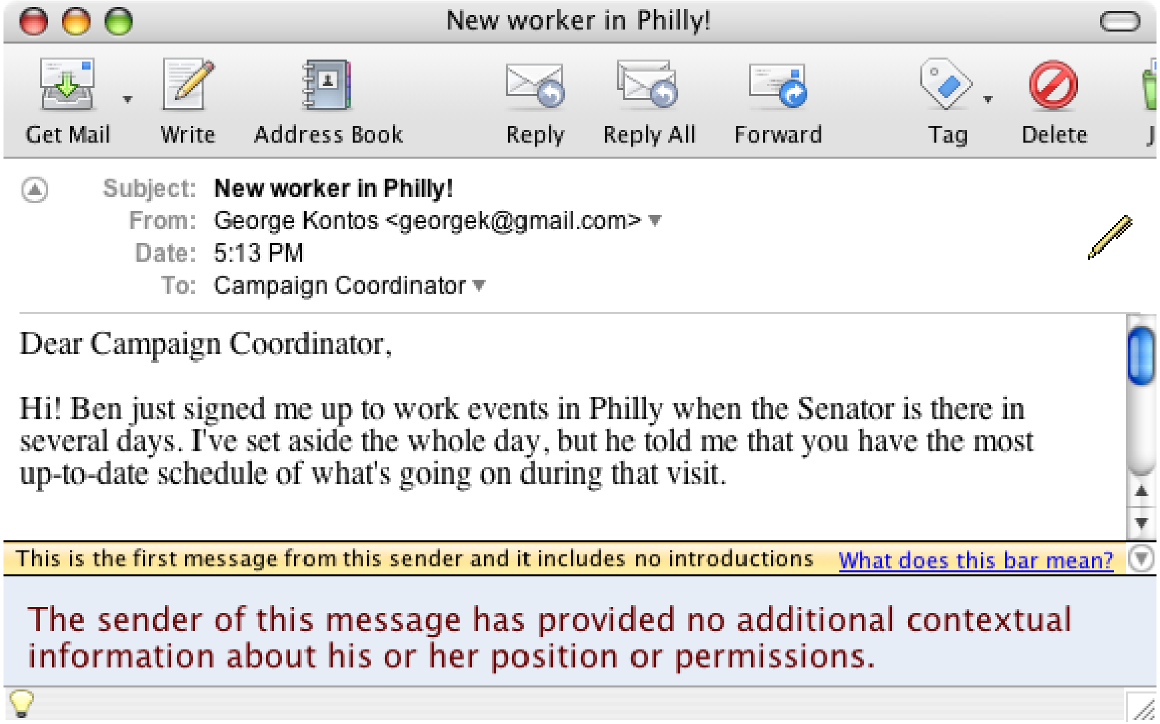
7.2 Human in the loop analysis: ABUSE presentation GUI

ABUSE relies on the human in the loop to look at attributes bound to an incoming message—if any—and decide whether or not to take whatever action is requested in the message body. If he decides that the attributes do not lead him to trust the message, he ignores it.

7.2.1 Communication Type

The ABUSE presentation GUI is, in Cranor's nomenclature, a passive security notice. The system scales the passivity of attribute presentation according to the familiarity of the message sender and the presence/familiarity of attributes. We discuss familiarity metrics back in Section 5.2.6 and present the entire range of GUI behaviors in Table 5.3. The presumed common-case behavior (familiar sender with familiar attributes) is shown in Figure 7.2. This is the most passive version of the presentation GUI; the digital introductions bar is a neutral color, and the attribute pane is minimized when the message is opened. The least passive version, displayed when an unfamiliar sender binds no attributes to her message, is shown in Figure 7.3.

Since the common case is that the user doesn't really need ABUSE's help, making the presentation GUI indicator passive makes the most sense. Furthermore, given that ABUSE is never able to determine with certitude that a user should ignore a given message, it would be inappropriate to interrupt the user's primary task of reading email. Moving toward the



Subject: **New worker in Philly!**
From: George Kontos <georgek@gmail.com>
Date: 5:13 PM
To: Campaign Coordinator

Dear Campaign Coordinator,

Hi! Ben just signed me up to work events in Philly when the Senator is there in several days. I've set aside the whole day, but he told me that you have the most up-to-date schedule of what's going on during that visit.

This is the first message from this sender and it includes no introductions What does this bar mean?

The sender of this message has provided no additional contextual information about his or her position or permissions.

Figure 7.3: The attribute presentation GUI at its least passive. The digital introductions bar is amber and the attribute pane is open by default, displaying an alert message in red text. The message depicted in this screenshot is from the user study detailed in Chapter 9, the basic elements of which are drawn from prior secure email usability research.

active end of the spectrum in cases where ABUSE is more likely to be useful is a good way to increase the likelihood a user will notice the extra information when it matters, but avoid habituating them to the presence of ABUSE communications.

7.2.2 Communication impediments

Malicious impediments

Often, when a security indicator does not apply, it is simply not present. For example, when most modern web browsers receive a web page over an SSL-encrypted channel, they display a security indicator of some kind. When they download a page over plain HTTP, they do not display an alternate "insecure" indicator. Instead, those security indicators are simply absent. This allows an attacker to put expected "security" indicators (e.g., a lock icon) into the content presented to the receiver somehow, and can often trick the unwary. [32, 129] A similar situation exists with S/MIME signature indictors in email clients. Normally, unsigned messages simply lack the "digitally signed" indicator, as opposed to having some kind of "unsigned" indicator. ABUSE avoids falling prey to this issue by having an ABUSE indicator always present in the chrome of the messaging client, though it is made unobtrusive when it is unlikely to be useful to the receiver. If concerns about the spoofability of the ABUSE chrome persist upon further evaluation, one could certainly apply one of the known methods of providing users with a trustworthy way to distinguish content provided by their system from content provided by a remote party. [26, 129]

Incidental impediments

ABUSE communicates attribute presence with a passive status indicator at the moment that a message is viewed by the human receiver. Thus, other passive or active indicators appearing at that moment might distract the user from the presence of a communication from ABUSE. Much of that is beyond the system's control. One avenue that we have chosen to mitigate this problem is to de-emphasize or remove ABUSE communication in situations where it is unlikely to be useful or necessary, as discussed earlier. Keep in mind that the user's primary task is to read their incoming messages and, if a request is being made of them, to service that request. ABUSE does not interrupt this task and, by remaining unobtrusive when it is unlikely to be needed, can optimize its impact in situations when it *should* be heeded, despite being passive. Another avenue that could be explored is making an attempt to integrate ABUSE more tightly with the Mozilla Thunderbird, the email client upon which we have built our prototype. For instance, Thunderbird has a built-in method of communicating to the user that it believes a message to be "Junk". Presumably, the designers have taken steps to help the reception of this communication of junk status; ABUSE could potentially take advantage of this by communicating in a similar fashion.

7.2.3 The human receiver

Attitudes and beliefs

As per Section 7.1.2, the receiver's prior experience with a communication influences her perception of it. If ABUSE communications frequently make the receiver suspicious of messages that he eventually deems trustworthy, he will be more likely to discount ABUSE notices in the future. In addition, frequent distractions from the primary task (reading email) could lead to frustration and create a negative attitudes towards the system. [125] ABUSE mitigates the latter problem by scaling its passivity as discussed in Section 7.2.1. As for the former, ABUSE never makes a behavioral recommendation to the receiver; the system never tells the user what to do. If senders frequently bind irrelevant or misleading attributes to messages the receiver winds up deciding are trustworthy, this could negatively impact the receiver's attitude towards ABUSE. This could be ameliorated by educating senders about how to choose attributes appropriate to their message.

Motivation

Motivation is intimately tied to the receiver's primary task. Naively, in the case of the presentation GUI, this is always "reading messages", but the real issue is what the receiver is then bidden to *do* by that message. The potential risks of the action requested by the message vary widely across domains. As related in Section 1.2.3, users have a bad track record when it comes to being motivated to be skeptical of incoming email. To recap, interviews conducted here at Dartmouth with a selection of staffers indicate that average users will do one of two things when confronted with a message from someone they do not know: if the request is sufficiently sensitive, they will consult some out-of-band resources to verify that the sender and/or message are legitimate; otherwise, the user will just assume that everything is fine and satisfy the request. Similar behavior can be seen in the phone transcripts from the 2003 blackout. "Sufficiently sensitive" is, of course, subjective and varies across domains. ABUSE allows senders to move some of these out-of-band checks in-band. The level of effort required to check on legitimate senders is lowered, and thus the bar for ”sufficiently sensitive” should be similarly lowered.

Capabilities

The receiver in an ABUSE-enhanced system is always making a choice between acting on a message and ignoring that message. Any human always has the capability to make this choice. Whether they can actually usefully act upon the message in question is addressed in Section 7.2.4, and is immaterial for this portion of the analysis. The decision is whether to act or not; the receiver is always capable of making this choice.

Communication Delivery

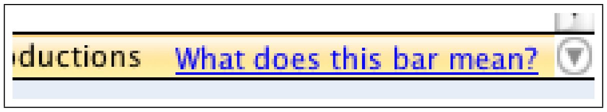
The first goal of a security communication is to get the receiver to notice it. In addition to potential impediments discussed in Section 7.2.2, habituation is also a big hurdle here. ABUSE's strategy for avoiding these issues has been mentioned already (Section 7.2.1). The next step is getting the receiver to focus on the notice long enough to understand the message being conveyed. Active notices can attempt to force the receiver to do so, but this can cause problems in other areas (negative attitude towards the system, habituation, etc.). The ideal way to explore whether ABUSE is effective in this dimension would be to deploy it and see what happens. As this is infeasible, we performed a pair of user studies to explore the efficacy of the ABUSE approach. The results of our first study, detailed in Chapter 8, suggest that users do pay attention to ABUSE attributes long enough to take in their content.

Communication Processing

The process by which we designed the presentation GUI was focused on coming up with a comprehensible representation of an attribute. The discussion in Chapter 6 supports our decision to go with a tabular format, familiar iconography to denote assertions that are outside of their validity period, and link-styling to indicate the names of familiar principals. We also provide on-line help to allow users to investigate attributes (detailed in Section 5.4.1) and to remind themselves of the purpose of the presentation GUI (Figure 7.4).

The ideal way to explore whether receivers comprehend ABUSE attribute presentation communications would be to deploy the system. Since we cannot do that, the user study detailed in Chapter 9 seeks to confirm that users of ABUSE-enhanced email clients effectively use communications from the system to understand the meaning of attributes bound to incoming messages.

The issue of knowledge acquisition is difficult to discuss with respect to ABUSE. There is no simple mapping between a set of attributes with a particular status and the correct action. The message plays a large role in the determination of the right thing to do, as



When a message is selected, this bar informs you of the status of any "Digital Introductions" bound to the message.

Digital Introductions allow email senders to inform you of the ways in which they are connected to people you already know, or to your home organization.

If you are unsure whether to trust this message, use the information in the pane below (click the bar to maximize it, if it is closed) to help you decide.

Figure 7.4: A piece of on-line help available from within the presentation GUI. Clicking on the link shown opens a dialog containing the message shown above.

does the specific content of the attributes. It is possible that one message with a single expired attribute would be trustworthy, while a second should be ignored. A pre-training step (perhaps Whitten's safe-staging [126]) could help address this if it turns out to be a problem. The ABUSE system does provide mouse-overs that explain new notices and status indicators when they appear.

Application

ABUSE helps the receiver recognize times when communications from the system will be applicable by making itself more prominent in appropriate situations. We have attempted to ameliorate knowledge retention issues by choosing symbols that are familiar and providing informational mouse-overs at all times. We have no training phase, though we would expect ABUSE communications to become more immediately comprehensible with greater familiarity.

7.2.4 Behavior

The receiver's behavior as a result of communications from the presentation GUI is to either choose to act on a message, or to ignore it. This decision is the behavior with which we are concerned.

The results in Chapter 8 and Chapter 9 show that users understand communications from ABUSE and that they *do* make more accurate trust decisions when armed with our system.

7.3 Human in the loop analysis: the other GUIs

As we discuss in Section 5.4.2 and Section 5.4.3, the other GUIs in ABUSE rely heavily on the presentation GUI analyzed above. The vast majority of security information communicated during either the attribute selection or issuance process is in the form of attributes. In the selection UI, as a matter of fact, there are no security communications at all outside of attribute information—which is handled by an instance of the presentation GUI. In the issuance GUI, suggested assertion values and expiration dates can be seen as security policies being communicated to the user. We treat them as such and discuss them here.

7.3.1 Communication type

The suggested values are passive policies. The receiver is meant to understand that the organization would prefer for them to choose one of pre-defined options. Especially in the case of recommended assertion values, there is a slippery slope here towards the creation of a "policy kit" approach that overwhelms users with choice. Domain-specific studies would need to be completed in order to determine how many suggestions make sense, and new GUIs would likely be needed to allow users to effectively make use of large number of suggestions.

7.3.2 Communication impediments

Malicious impediments

We do not consider the case in which adversary-controllable content is active during the attribute issuance process.

Incidental impediments

The ABUSE issuance GUI appears as a modal dialog; Thunderbird cannot take control from the GUI, and so the only concerns in this space arise from factors outside the application, over which ABUSE has no control

7.3.3 The human receiver

Attitudes, beliefs and motivation

It is difficult to judge these aspects of the receiver's mindset without real-world information. User studies that try to get at these questions would be of limited utility; these factors develop over time. We do know that users are overwhelmed by overly complicated policies and too much choice [1, 47], and so we believe that more than a small number of options would be ill-advised.

Capabilities

A user could certainly use a small number of well-chosen suggestions to issue an attribute that expires within a timeframe condoned by the policy. Again, too much choice here would likely be paralyzing.

Communication Delivery

If the set of options does not change frequently, the receiver needs only to notice when modifications to the policy are made. An icon indicating new content could easily attract the attention of the user in this case. A constantly changing set of dates or assertion options would likely quickly cause habituation, and so this is recommended against.

Communication Processing

Future researchers could conduct user studies (proposed in Section 10.3) in order to explore how well receivers responded to pre-defined assertions and expiration dates.

Application

As the communication is present at all times during attribute issuance, we are not concerned with the receiver's ability to apply these policies. If they are not overwhelmed by choice, they should be able to use the suggestions without difficulty. This expectation can be be supported by studies such as those discussed in Section 10.3.

7.3.4 Behavior

The desired behavior here is the issuance of attributes that use suggested values when deemed appropriate by the human receiver. With only a small set of suggestions, this should be easily achieved. Verification of this belief can be acquired by real-world deployment or by user studies.

7.4 Conclusion

According to Cranor's framework, the ABUSE attribute presentation GUI should effectively allow users to understand attributes and use them to make decisions. The attribute issuance GUI can, in a sense, communicate a security policy to users as well; as long at this "policy" is kept simple, we believe that Cranor's framework indicates that users should be able to comply.

In this chapter, we have demonstrated consideration of all the usability issues that gave rise to Cranor's work and made reasonable decisions about the ways in which the GUI should address them. We performed a pair of user studies in order to validate the choices we made, and we now move on to discuss them.

Chapter 8

Evaluation: ABUSE in Power Grid Scenarios

We performed a pair of user studies in order to gather in support of ABUSE. The first study, which used task setups drawn directly from the August 2003 blackout [88], was designed to compare users' ability to make trust judgments when equipped with ABUSE-enhanced email versus their ability to do so when equipped only with current email technologies. We hoped to verify two hypotheses during this comparison:

1. ABUSE enables users to identify trustworthy messages from unfamiliar third parties, and

2. ABUSE users do not exhibit a significantly higher rate of false positives during trustworthy message identification.

We define "trustworthy messages" in this case to be those whose attributes indicate that it is reasonable to believe that the sender has the authority to request the stated course of action. If subjects in the group using ABUSE-enhanced mail were no more able to distinguish the trustworthy messages presented in the study than were the rest of the subjects, this would be evidence that ABUSE is not functioning as intended. If, however, the subjects *were* more able to pick out the trustworthy message, this could indicate that ABUSE enables users to migrate their trust-building strategies from real life into the world of digital communication. The second point would then become the crux of the matter: if subjects treated the mere presence of ABUSE attributes as an indication that a message was trustworthy, then any perceived improvement in their ability to pick out the good messages from the bad should be dismissed as noise. Subjects would be treating attributes as a *binary security indicator*; the mere presence of attributes would be leading them to trust messages. However, if they were no more likely to trust attack messages than their peers, *but* were more readily able to identify the actual trustworthy messages, then it would be clear that ABUSE is actually helpful.

8.1 Methodology

Risk and trust go hand in hand. ABUSE seeks to provide users with better context for making risky decisions than S/MIME and plaintext email. Setting plaintext against S/MIME against ABUSE, however, is not an entirely fair comparison. ABUSE includes extra contextual information that the others do not. Currently, users may resort to out-of-band channels to get this extra context. To more fairly compare these pre-existing technologies to ABUSE, it is necessary to simulate for subjects the ability to consult those extra sources of information. Framed in terms of risk, subjects had to want to make a choice based on only what was presented in the email in front of them, but still be able to fall back on other sources of information if they decided the risk was too great. Thus, the metric for comparison becomes not simply whether subjects can make the right decision more often with ABUSE over other technologies, but rather whether subjects armed with ABUSE make the right trust decisions *without resorting to out of band channels* more often than subjects using plaintext email or S/MIME. More precisely, the questions are these:

> Can subjects armed with ABUSE identify trustworthy messages from unfamiliar third parties without resorting to out-of-band channels with greater frequency than other subjects? Does the same group erroneously identify untrustworthy messages as trustworthy more often than subjects using S/MIME or plaintext email?

Consulting out-of-band channels causes delay, which users in time-sensitive situations (like crises in the grid) may not be able to afford. Indeed there may be cases in which these channels are not even available. Moreover, as we discussed in Section 1.2.1, we know that users have a tendency to assume that messages are trustworthy and reliable when getting additional confirmation for the request is too difficult. ABUSE seeks to remove the extra barriers, making it easier for users to make an appropriately informed decision.

To answer the questions posed above, we needed to put subjects in situations in which they needed to trust messages from unfamiliar third parties in order to complete a task, and also had reason to worry about getting fooled by untrustworthy messages—but were not so afraid of getting tricked that they would invariably seek the reassurance of traditional trust-building methods (i.e. contacting some trusted individual with knowledge of the situation).

We accomplished this by creating the incentive structure detailed in Section 8.1.1.

Armed with this reasoning, we chose to use scenarios lifted from the August 2003 blackout for this study. An emergency in the power grid (a *contingency* in the parlance of the industry) is clearly a high-stakes situation and, as we discussed in Section 1.2.2, the people working to keep the system under control frequently have to trust people that they have not encountered before. Furthermore, they use informal methods to build that trust. They currently either leverage human connections by making phone calls to people they *do* know [88, pp. 56–58], or they assume that anyone who knows the right phone numbers to call and can "talk the talk" is worthy of at least a measure of trust. [30] These operators know that, when a contingency arises, the more quickly it can be mitigated the better—and that doing nothing can sometimes be just as bad as doing the wrong thing [88, pp. 480–484]. However, the conversations in the phone transcripts indicated that the operators were simultaneously hesitant to act unless they felt confident in the decision they were making—or, at least, confident that someone with the appropriate authority was ordering the action they were about to take [88, pp. 236–238]. So, in these power grid scenarios, operators need to trust third parties they do not know in order to do their jobs, but concern over a variety of factors gives them pause.

Ideally, we would have been able to perform the study on actual grid operators, provide as much realism as possible, and report the results. However, this was infeasible; like most academic researchers performing these kinds of experiments, our pool of subjects is mostly limited to college students. Choosing this scenario, therefore, required us to devise an incentive structure, detailed in Section 8.1.1, that adequately mirrored the tradeoffs faced by real grid operators while remaining comprehensible to the subjects.

8.1.1 The incentive structure

Devising an incentive structure for this study was challenging for a number of reasons. First, power companies are disinclined to disclose cases of operator failure, unless there is a case of "gross (criminal) negligence." [91] It is therefore difficult to determine what consequences exist for employees that make bad decisions, since the companies don't want to admit that errors occur at all. Second, quantifying the risk of information leakage is an open question. There could be economic consequences, as the power market can be gamed using inside information just like many others. There could be collateral damage, if the information is used to better target a physical attack. In either case, if generation or transmission capacity is lost or disabled accidentally, how much money is lost by being unable to sell that power? Wholesale power pricing is constantly changing in accordance with market conditions. [75] There may have been an outage, or perhaps competing companies were able to provide coverage. Either way there is some cost. Determining an accurate model for this is undeniably interesting, but beyond the scope of this research.

For the purposes of this study, the subjects required an incentive structure that would provide them with a mindset similar to that of a grid operator. So, we set out to collect anecdotal evidence of potential consequences of operator error and discovered the following:

- the obvious outages and/or equipment damage;
- internal reprimand or other disciplinary action;
- an investigation, perhaps by the Federal Energy Regulatory Commission (FERC) or Congress if the scale becomes large enough;
- guilt, leading to nervous breakdowns [107];
- legal trouble, akin to an insider-trading-type scandal; or
- if the grid is under physical attack, leaking status information could expose weaknesses in the grid to the adversary. [91, 107]

Based on this and some personal conversations with industry insiders, we determined that the disincentive for making the wrong decision about a message had to be more highly negative than the reward for making the right choice was positive. Breaking-even after one correct choice and one incorrect choice would be unrealistic. So, if r is the incentive to be right, and w is the penalty for being wrong, we wanted $r < w$ to hold. That said, a subject who makes a wrong decision should still have the opportunity to be above the break-even point. We had five tasks for the subjects to perform, so $4r - w$ needed to be greater than zero. Furthermore, a subject should not be able to simply make the same choice every time and come out ahead. As there are three attack scenarios and two trustworthy scenarios, rejecting every time nets the subject three correct decisions and two incorrect decisions. Thus, $2w - 3r < 0$ needed to hold as well. Setting $r = +10$ and $w = -20$ satisfied all these conditions, and had the added benefit of being easily memorable by the subjects.

As a secondary concern, though, we also wanted to provide a disincentive against delaying a trust decision by consulting out-of-band sources. In real situations, doing so delays operator action, exposing the grid to more risk. Leaving such a disincentive out of the study would likely cause subjects to go for the potential extra certainty every time. We provided two simulated out-of-band channels for the subject to get extra context: calling

Chosen	Out-of-band channels used			
action	None	Phoned someone	Checked chart	Both
Reject	Trustworthy: −20 Attack: +10	Trustworthy: −20 Attack: +8	Trustworthy: −20 Attack: +8	Trustworthy: −20 Attack: +6
Accept	Trustworthy: +10 Attack: −20	Trustworthy: +5 Attack: −20	Trustworthy: +7 Attack: −20	Trustworthy: +2 Attack: −20

Table 8.1: The incentive structure used in the study, as discussed in Section 8.1.1. Subjects are rewarded for making correct choices, penalized for delaying in proportion with how problematic the delay might be, and strongly penalized for making the wrong choices.

an acquaintance for more information and using the company organizational chart to check for someone's presence or position. As the former would take more time than the latter in the real world, it was assigned a stronger disincentive. Pre-tests indicated that the stronger of the two needed to be half of the potential gain; subjects were never disinclined to "phone a friend" otherwise.

Finally, we wanted to moderate the disincentive for delaying a decision in the event that a message turned out to be untrustworthy. Logically, the right thing to do with an attack message in real life is to ignore it; while the subject is still wasting time, he should not be penalized as much for delaying a choice to do nothing as for delaying a choice to act. Taking all this into account, the final structure is presented in Table 8.1.

8.1.2 The protocol

The study employed a between-subjects design to examine whether ABUSE users are better able to identify trustworthy messages compared to users of other email clients. Subjects were randomly assigned to one of three groups defined by the type of simulated email client used during the study: (1) Plaintext, (2) S/MIME (with validly signed messages), and (3) ABUSE (with signed and cryptographically valid attributes). The pre-study instructions, the simulated email headers displayed, the bodies of the messages, the task scenarios and the post-study debriefing were identical across the three groups. We randomized the order of tasks within each email group to control for order effects; the results could be muddied if our chosen ordering predisposed users in a given group to make certain types of decisions.

The study was composed of three separate phases: the instruction phase, the task phase and the debriefing phase. During the instruction phase, each subject was read the same set of instructions (Figure A.3) while following along in a handout they had been given (Figure A.4), which contained a bullet-pointed version of the same information they were hearing orally as well as some diagrams that were helpful in explicating the structure of the organizations that manage the power grid and their relationships to one another. Read-

> We are investigating a new method of enabling power grid operators to collaborate in order to maintain North America's power generation and distribution infrastructure. You will take the role of a grid operator, and face a series of scenarios that simulate instability in the grid. Consider each as totally independent; information presented in one scenario should not impact your decisions in any other. In each scenario, you will be presented with a message sent to you by a third party that asks you to perform some action in response to a problem. Though the message will come from someone you do not know, your task is to decide whether to trust it or not. If you choose to trust an untrustworthy message (or to ignore a trustworthy message), there are several negative consequences that could ensue:
>
> - First, part of the grid could collapse, leading to a wide-area blackout. This could lead to anything from a reprimand for you to a Congressional investigation of your employer. Additionally, guilt over causing an outage of that scale has led some operators in your situation to experience a nervous breakdown.
> - Second, you could wind up disclosing grid status information, which is uniformly regarded as sensitive. Not only would this expose the grid to a targeted physical attack, but it could also land you in legal trouble.

Figure 8.1: The text read to subjects in order to introduce the study and sensitize them to the dangers of making mistakes as a grid operator.

ing the instructions to the subjects too approximately four minutes. The purpose of these instructions was manifold:

- to sensitize the subjects to the risks of making the wrong trust decisions,
- to educate them about the study interface and overall structure,
- to make sure they were capable of interpreting the job titles and roles they would encounter, and
- to explain the incentive structure used to score the study.

Subjects were not sensitized to the particular kinds of attacks they would be facing, or to particular "gotchas" inherent in any of the three communication technologies that were simulated. They were simply told that they would be receiving some messages that were trustworthy and some that were not. Figure 8.1 provides some detail. After being read the instructions, the subjects were allowed to start performing the study tasks.

The study was constructed as a game that consisted of a set of five tasks. Having five tasks allowed us to test each of three interesting attack scenarios while also having a pair of trustworthy scenarios to help verify that the subjects who correctly identified the trustworthy messages didn't just get lucky. In each scenario, the subject was given a new persona with a name, email address, social network, and position at some power grid organization unique to that task scenario. The subject was also presented with a summary of the status of the portion of the grid over which she exercised control, and told of a problem that had arisen. Her goal was to help return the grid to stability as soon as possible, but she was incapable of doing this on her own. The study infrastructure then presented her with a message that provided her with a strategy that the sender claimed would help mitigate the contingency. The subject had to decide whether to heed the message right away, reject it out of hand, or consult out-of-band channels to attempt to get more context for her decision. These out-of-band channels (the ability to query an acquaintance or consult a company organizational chart) are provided for two reasons: first, to provide greater verisimilitude; and second, to enable subjects without the benefit of ABUSE to have a chance to make the correct trust decisions in all cases. Basing the study incentives upon making correct trust decisions and then making it impossible for some subjects to "get it right" would be unethical. After completing the five tasks, the subjects passed into the final phase of the study.

At the beginning of the debriefing phase, subjects were informed that they had completed the tasks and told that they would now be asked to review their answers. They were also reminded that they would not be able to change them; they would only be allowed to indicate whether or not they, in retrospect, would change the decision they made. In addition, the subjects were allowed to provide free-form comments about why they were comfortable with their initial trust decision or not. These comments allowed us to not only see some very encouraging signs that people were actually reading and using ABUSE attributes in their decision-making process, but also allowed for us to find cases in which a subject had become confused by the study interface or by some of the power-grid trappings of the setup. Upon completing this review, subjects were informed of their score, thanked, paid, and allowed to leave.

8.1.3 Power grid background

In order to discuss the setup of the study, it is necessary to flesh out the power grid introduction we provided in Section 2.2. The subjects received a more extensive primer, seen in Figure A.3. It is important to note that the organizational relationships we enumerate here reflect our understanding of the generation and transmission management side of things at the time of the 2003 blackout. We make no claims about the business relationships among these entities, either back in 2003 or at the current time.

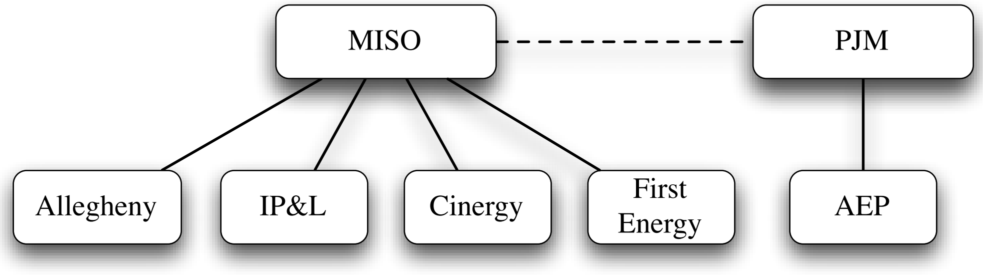


Figure 8.2: The relationships between MISO, PJM, and their associated operation companies. Note that MISO and PJM are peer organizations, while the operation companies each have a subordinate relationship to one of the two.

Power organizations

As discussed in Section 2.2.2, there are two classes of management organizations in the grid: operations companies that own and operate infrastructure, and Regional Transmission Organizations (RTOs) that mediate among them. The subjects encountered two RTOs—PJM Interconnect, and Midwest ISO (MISO)—as well as a number of operation companies. These organizations and their relationships to each other are shown in Figure 8.2.

Power jobs

We already introduced a smattering of power grid jobs in Section 2.2.3. We reiterate those here, and include a few more that we used in this study.

In every scenario in the study, subjects were in the employ of an operation company, so they were assigned one of the following roles in each of the five scenarios:

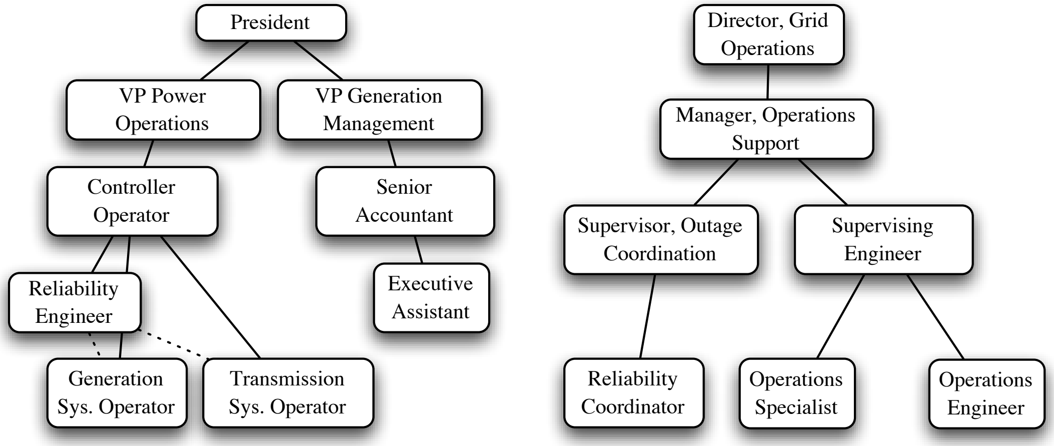
- **Generator System Operator** controls his company's power generators.

- **Transmission System Operator** controls her company's power lines and other transmission infrastructure.

- **Reliability Engineer** works with Generator and Transmission System Operators at the company to maintain stability in the portion of the grid over which the company has control. Interfaces with Reliability Coordinators at RTOs during wider-scale problems.

Message senders also sometimes claimed one of the jobs above, or they might reference someone in a superior position like one of the following:

- **Controller Operator** manages Generator and Transmission System Operators at an operation company.



(a) The structure of an operations company (b) The structure of an RTO

Figure 8.3: Sample organizational charts for power grid entities

- **Reliability Coordinator** works with Reliability Engineers at operation companies and Reliability Coordinators at other RTOs to effect the changes necessary to mitigate outages and grid instability.

Subjects who saw attributes would be exposed to more job titles, and thus needed to have a sense of what those people were allowed to do. The text that they saw to explain these roles can be seen in Figure A.4.

Power problems

The subjects face three different contingencies:

- **Loaded Lines** - the subject is a Transmission System Operator at IP&L. Due to outages, some lines between substations belonging to the company are heavily loaded. This information is sensitive, as this knowledge could be used to better target physical attacks on the power infrastructure. The specter of coordinated physical attacks was a concern that arose during the 2003 blackout [88, p. 256].

- **Tie Lines** - the subject is a Reliability Engineer at First Energy. First Energy is connected to AEP by several *tie lines*, high-voltage lines that are used to ship power from one area to another in bulk. Losing all the tie lines between two areas of the grid cuts off the ability to directly ship power between the two. In this contingency, some of the tie lines between First Energy and AEP are down and so the remaining ones are overloaded, though still functional. This case is distinct from the above because there are actual deactivated transmission lines here, and because the lines cross organizational boundaries.

Scenario	Contingency	Subject role	Response
Attack coopetition	Tie lines	Reliability Engineer at First Energy	Ignore
Attack role-based	Overloaded lines	Transmission system operator at IP&L	Ignore
Attack delegation	Over-generation	Generation system operator at Cinergy	Ignore
Legitimate coopetition	Tie lines	Reliability Engineer at First Energy	Accept
Legitimate delegation	Over-generation	Generation system operator at Cinergy	Accept

Table 8.2: Summary of the five scenarios faced by the subjects.

- **Over-generation** - the subject is a Generation System Operator at Cinergy. Allegheny is sending too much power into Cinergy's area from the west. Cinergy's east-west transmission lines are thus overloaded. Either Allegheny needs to bring down their generation, or Cinergy needs to generate less in the west and more in the east.

8.1.4 The five scenarios

Each task scenario was based on an actual event that occurred during the North American August 2003 blackout. Some scenarios unfolded over multiple phone conversations, while some were contained in a single call. In each scenario, we identified the relying party, the trust source, the trust sink, the authorizer and any intermediaries. The experimental subject was put in the position of the relying party. The scenarios in which the subjects received trustworthy messages are actual contingencies that actually occurred, and the action requested by the sender is the strategy that was actually used to mitigate the real problem. The attributes bound to the message express the same flows of trust that we distilled from the phone transcripts discussed in Chapter 2. The scenarios in which the subjects received attack messages were designed to closely ape the trustworthy cases, with contingencies that were analogous to real problems that arose in the grid. The "mitigation strategy" recommended by each attack message was designed to seem plausible when evaluated in the context of its accompanying scenario. [23–25] There were three different kinds of attackers:

1. a completely external attacker,

2. an *internal-insider*, employed at the same company at which the subject's persona worked, and

3. an *external-insider*, employed at another power grid organization.

The attackers are individuals without any fellow conspirators on the inside. While two of the attackers are insiders, neither is a person who has the authority to directly order potentially damaging actions. The insiders carry out their attacks in their own name, behavior that has been seen in the real world. [112] For the purposes of the study, it was assumed that there was some kind of power-grid-wide PKI bridge, so that S/MIME signatures from and attributes granted by people at other power grid organizations would all be verifiable. This is reasonable, as the aerospace industry [14], the pharmacological industry [102] and several other groups that must all interact with the federal government regularly already employ such technology. By the time ABUSE could be deployed, it is certainly plausible that a bridge could be as well. The external attacker is not able to claim any attributes at all that chain back to this bridge; he would require an inside conspirator or access to the organization's attribute boot-strapping infrastructure. The insiders were assumed to have a set of attributes that were granted to them by their home organizations and the people working there. ABUSE does not require senders to bind all their attributes to their messages, so the attackers were allowed to bind any attributes that they felt would help them mislead the subjects into trusting them.

The makeup of a single scenario

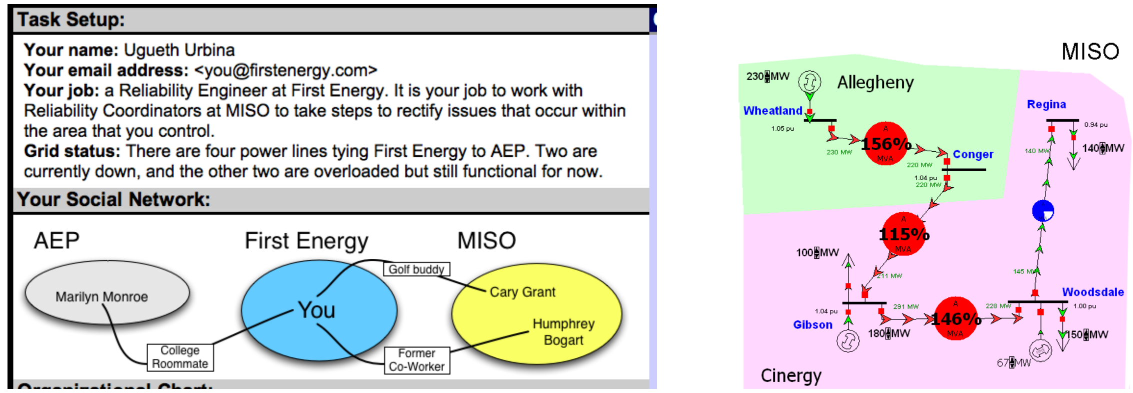
A single scenario consists of several pieces of information:

- the subject's identity: name, email address, job, affiliation, social network
- the sender: affiliation (or claimed affiliation), certificate issuer, attributes
- the contingency: what has gone wrong and where
- the message: kind of attack (if any), message text
- out-of-band information: subject's company organizational chart, information available from a peer

The five scenarios are detailed in Section A.4 and summarized in Table 8.2. The messages received in each scenario are paraphrased in Table 8.3 The attackers either eschewed attributes altogether, or included some valid attribute that would not specifically lead to trust, but might lend more credence to their message (e.g. an "is an employee" attribute from their company). The trustworthy senders bound attributes to their messages indicating that

Scenario	Content
Attack coopetition	The sender claims to be having an issue with her work email, thus motivating her use of a `yahoo.com` email address. She attaches no attributes, though her message is signed—albeit with a non-grid-company-issued certificate. She claims to be a transmission systems operator at AEP and tells the subject that the downed lines are about to come back up, so it is ok for First Energy to begin drawing more power.
Attack role-based	The sender signs his message with a cert from the MISO CA, and binds an attribute to his message stating that he's a MISO employee—which the subject should see as insufficient to create trust. He claims that something's gone awry with his system and he needs the subject to provide him with information about which lines are currently at risk of failure.
Attack delegation	The sender signs her message with a cert from the Cinergy CA, and binds an attribute to her message indicating that she's a Cinergy employee. The subject also works at Cinergy. She claims she's been working with a Cinergy controller operator and a MISO reliability coordinator to get Allegheny to bring down their generation, and asks the subject to boost generation in the west.
Legitimate coopetition	The sender signs his message with a cert from the AEP CA, and binds attributes from the company saying that he is a reliability engineer, as well as one through a reliability coordinator at MISO that indicates he is allowed to ask First Energy to buy power or shed load. He requests that the subject do so.
Legitimate delegation	The sender signs his message with a cert from the MISO CA, and binds attributes from the organization saying that he is a reliability coordinator, as well as another rooted at Cinergy that indicates he is allowed to ask Cinergy to shift some generation from the west to the east. He requests that the subject do so.

Table 8.3: Messages received, by scenario



Task Setup:

Your name: Ugueth Urbina
Your email address: <you@firstenergy.com>
Your job: a Reliability Engineer at First Energy. It is your job to work with Reliability Coordinators at MISO to take steps to rectify issues that occur within the area that you control.
Grid status: There are four power lines tying First Energy to AEP. Two are currently down, and the other two are overloaded but still functional for now.

Your Social Network:

(a) Screenshot showing how the setup for each scenario was presented to the subjects.

(b) A graphical representation of grid status.

Figure 8.4: Two different methods of presenting scenario setup information to subjects.

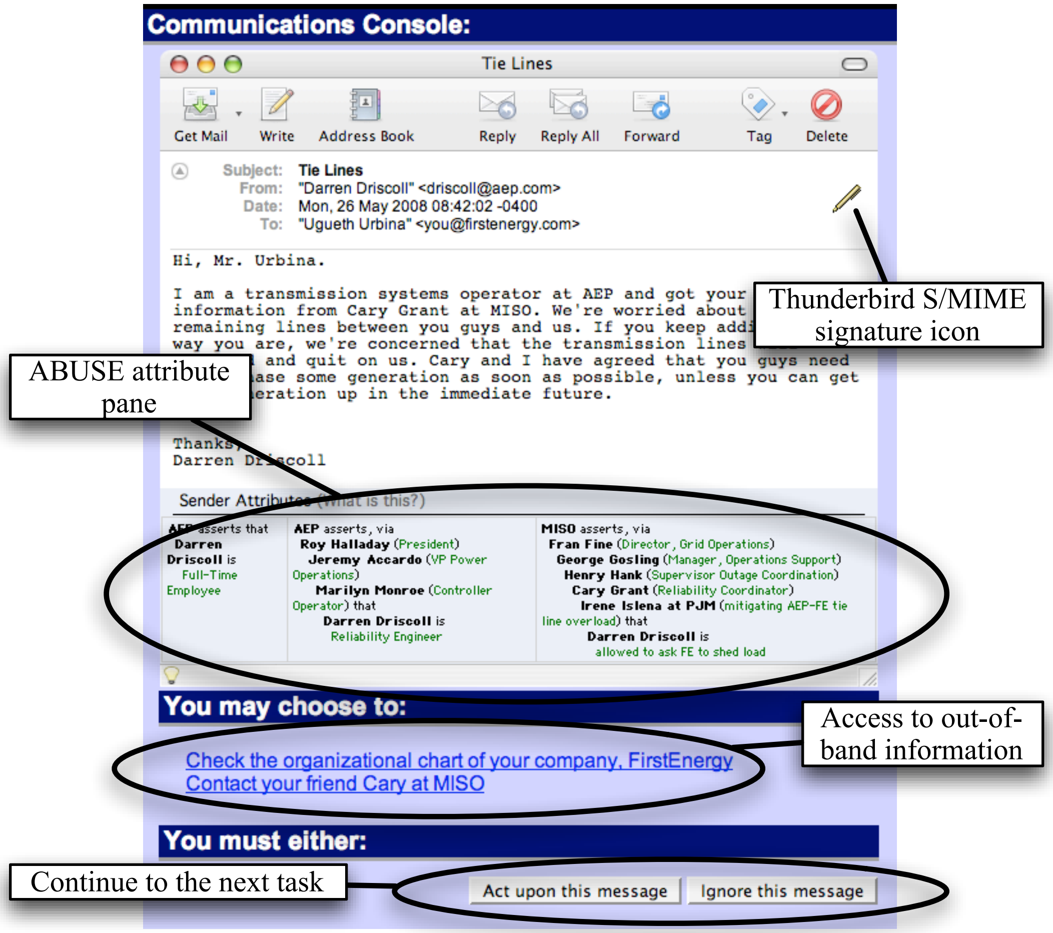
had been granted permission to request the proposed mitigation by someone with the authority to do so. Subjects who saw ABUSE attributes bound to messages were still allowed to fall back on the out-of-band channels provided by the study infrastructure, but ideally they would realize that they did not need to.

8.2 Study infrastructure

The subjects took the study through a web browser (Mozilla Firefox 2). Each scenario is defined by an XML file. Once the subject inputs his randomly-assigned ID number, the system generates a random ordering of the five scenarios and parses them one-by-one, populating a standard GUI each time. After the subject makes his decision, the system moves to the next scenario in its random ordering. Stepping through the presentation of the "Legitimate coopetition" scenario will be illustrative. Screenshots of all five scenarios are shown in Appendix A.

8.2.1 Initial conditions

The subject is first presented with the initial scenario conditions: the details of the subject persona and the current contingency. In early versions of the study, the grid status and contingency information was presented graphically using a screenshot of an actual power grid state estimator [94], a piece of software that is used in real power facilities to visualize power flows. Though we had trimmed the model used by the estimator down significantly (only a few busses, generators and loads), pre-test subjects still found this to be overwhelming. As a result, we swapped this graphical representation for a concise textual explanation confused the subjects less. It reduced verisimilitude but, after consulting with a colleague



Subject: Tie Lines
From: "Darren Driscoll" <driscoll@aep.com>
Date: Mon, 26 May 2008 08:42:02 -0400
To: "Ugueth Urbina" <you@firstenergy.com>

Hi, Mr. Urbina.

I am a transmission systems operator at AEP and got your
information from Cary Grant at MISO. We're worried about
remaining lines between you guys and us. If you keep addi
way you are, we're concerned that the transmission lines
d and quit on us. Cary and I have agreed that you guys need
ase some generation as soon as possible, unless you can get
eration up in the immediate future.

Thanks,
Darren Driscoll

Sender Attributes (What is this?)

AEP asserts that
Darren Driscoll is
Full-Time Employee

AEP asserts, via
Roy Halladay (President)
Jeremy Accardo (VP Power Operations)
Marilyn Monroe (Controller Operator) that
Darren Driscoll is
Reliability Engineer

MISO asserts, via
Fran Fine (Director, Grid Operations)
George Gosling (Manager, Operations Support)
Henry Hank (Supervisor Outage Coordination)
Cary Grant (Reliability Coordinator)
Irene Islena at PJM (mitigating AEP-FE tie line overload) that
Darren Driscoll is
allowed to ask FE to shed load

You may choose to:

Check the organizational chart of your company, FirstEnergy
Contact your friend Cary at MISO

You must either:

Thunderbird S/MIME signature icon

ABUSE attribute pane

Access to out-of-band information

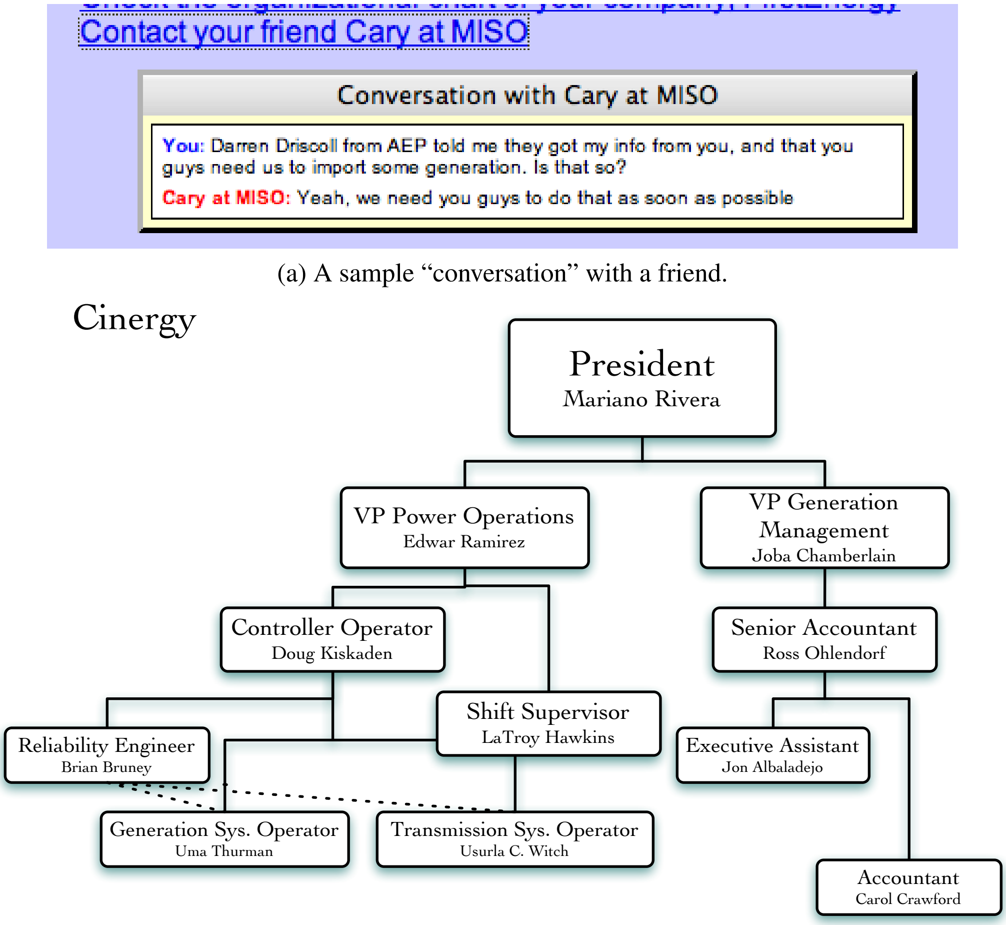
Continue to the next task

Figure 8.5: A screenshot of the interactive portion of the study interface.

who has more experience crafting user studies [106], we believed it would remove a source of error from our data. At this stage, the subject learns his name, email address, job, responsibilities and social connections (Figure 8.4).

8.2.2 The message

After having some time to digest the initial setup information, the subject "receives" the message for this scenario. The links to access out-of-band information are also presented at this time, as well as the buttons that allow the user to indicate whether they wish to act upon or disregard the message (Figure 8.5). If the subject is in the S/MIME or ABUSE groups, the simulated message will have the standard Thunderbird signature indicator present, and mousing over it will provide the text given to the user in the Thunderbird dialog box that appears when a user clicks on a signature icon in the real client. Subjects in the ABUSE group also, obviously, see ABUSE attributes in the simulated message window. We display



Contact your friend Cary at MISO

Conversation with Cary at MISO

You: Darren Driscoll from AEP told me they got my info from you, and that you guys need us to import some generation. Is that so?

Cary at MISO: Yeah, we need you guys to do that as soon as possible

(a) A sample "conversation" with a friend.

(b) A sample organizational chart from the study.

Figure 8.6: The two out-of-band channels available to subjects in the study.

the attributes using an early mockup of the final attribute presentation GUI. We discussed the mockup in Section 6.1 and the final design in Section 5.4.1.

8.2.3 Out-of-band information

As mentioned earlier, the subjects were given the opportunity to "contact a friend" or consult their company organizational chart in an attempt to figure out whether to trust a message or not. Information was always available by these channels, but subjects were informed that it may not be useful. In pre-tests, contacting a friend always told the subjects the right course of action. This trained subjects to always use this information, even though they lost some points. If they felt it would always work, they were willing to absorb the loss in exchange for certainty. The test was, therefore, modified. A summary of which pieces of information were useful in which scenarios is in Table 8.4.

It was challenging to decide how to appropriately simulate the ability to contact a friend.

Scenario	Sender type	Attributes	Acquaintance helpful?	Org chart helpful?
Attack coopetition	External attacker	None	No	No
Attack role-based	External-insider	Vague but valid	Yes	No
Attack delegation	Internal-insider	Vague but valid	No	Yes
Legitimate coopetition	Trustworthy	Specific permissions	Yes	No
Legitimate delegation	Trustworthy	Specific permissions	Yes	No

Table 8.4: A summary of which kinds of senders were active in each scenario, and what information the subject had available to help his trust decision. In the "attack coopetition" scenario, the usage of a non-company email address should have been enough to indicate that the subject should not act upon the message.

It may have been nice to allow the subject to contact any of the people in the social network provided. However, for the purposes of the study, this was pointless; all that mattered was whether they consulted an out-of-band channel or not. It would also have complicated the scoring of the study and the study infrastructure. Thus, we decided to allow them to indicate that they wished to contact an acquaintance, and then the system would provide for them an instant-messaging-style transcript of a short conversation with a person of our choosing (Figure 8.6a). A few subjects in a few scenarios indicated that they would have chosen a different person than the one we provided; however, those scenarios were designed such that contacting an acquaintance would not be helpful, so it would not have made a difference in the results.

8.2.4 The review

During the portion of the study in which subjects were allow to review their answers, they were presented much the same GUI as shown above. Instead of being allowed to access all the out-of-band channels, they were shown only the information they had chosen to look at during the study. The purpose of this was not to see what they would have done had they accessed more information, but to control for whether or not things they saw later in the study made them reconsider what they did early on. For example, one subject in the ABUSE group saw an "employee" attribute during an early scenario and thought it was enough to go on. After seeing the more extensive attributes in later scenarios, the subject indicated that, in retrospect, "the sender attributes were weak compared to what we saw later."

8.3 Results and discussion

A total of 34 subjects took part in the study, 12 in the ABUSE group and 11 in each of the others. Details of subject recruitment appear in Section A.1.

Completing the tasks in the study took a mean of 14 minutes, and no subject took more than 20 minutes to complete the tasks. All subjects completed the debriefing stage and finished the entire study in thirty minutes or less..

8.3.1 Task comprehension

Based on the comments provided in the followup review, most subjects understood the scenarios provided and the tasks which they were to perform. One subject (7641) admitted being confused about the study setup and infrastructure on the first task he was presented, but indicated that he understood what was happening on later tasks. That observation was removed from the sample, as was one observation of the external-insider in the S/MIME group. In these two cases, the subject's comments indicated that their choices reflected only confusion with the mechanics of the study and so the exclusion of the two observations is warranted. Other than that, the comments indicated that the subjects bought into the scenario and understood their responsibilities and those of the people around them. Encouragingly, several subjects in the ABUSE group specifically noted that they were paying attention to the "sender attributes" bound to the messages they were seeing, in trustworthy scenarios as well as attack scenarios:

> "The sender attributes listed many high-ranking people that support his authority to request this action." (Subject 1093)

> "we used the Sender Attributes to determine that Darren was an appropriate source of information." (Subject 1063)

> "The sender attributes all seemed valid," (Subject 1073)

> "[The external-insider] had no other people listed under sender attributes to back up his authority to request information." (Subject 7603)

Subject 1073 also indicated regret about falling for the internal-insider attack, noting that "the sender attributes were weak compared to ones we saw later," so he clearly learned more about the meaning of ABUSE attributes as he was exposed to more of them. Results of a second study, performed to dig deeper into questions surrounding user comprehension of attribute content, are discussed in Chapter 9

Email type	n	% correct overall
ABUSE	60	75%
Plaintext	55	65%
S/MIME	55	60%
		$F = 1.5$ $p = .224$

(a) The mean overall level of success rates for each email type

Email type	n	% correct overall
ABUSE	60	75%
Non-ABUSE	110	63%
		$F = 2.66$ $p = .105$

(b) Overall success, ABUSE vs. all non-ABUSE user-rounds

Table 8.5: Percent of correct responses to tasks by type of email.

8.3.2 Evaluating ABUSE versus existing technologies

We discovered that subjects using ABUSE are indeed able to identify trustworthy messages from unfamiliar third parties without needing out-of-band channels significantly more frequently than subjects provided only with S/MIME or plaintext mail. Furthermore, ABUSE subjects were no more likely than subjects in other groups to fall for attack messages, indicating that ABUSE confers an advantage upon users of the system without making them any more vulnerable to attacks than users with current technology.

Overall success

Overall success rates are shown in Table 8.5. Success is measured as the percent of all tasks ($n = 5$ per subject) that were correctly completed, i.e. the subject acted on trustworthy messages and chose to ignore untrustworthy messages. In Table 8.5a, we look at the performance of subjects in all scenarios across the three email groups. Subjects using ABUSE were correct 75% of the time, compared to rates of 65% and 60% among plaintext users and S/MIME users respectively. Statistically comparing the mean percent correct in each group using an analysis of variance (ANOVA) test indicates that there is no statistically significant difference between the three email groups ($F = 1.5$, $P = .224$). Table 8.5b compares overall success for ABUSE (again, 75%) to success in all non-ABUSE user-rounds, showing a nearly statistically significant difference considering a p-value of .10 ($F = .266$, $P = .105$). This is nice to see, but it really isn't actually what we want to know. We wish to see whether users armed with ABUSE can identify trustworthy messages without requiring out-of-band help more frequently than users with existing technologies.

Success level by scenario

Next, we examine whether subjects in each email type were correct more or less often depending on scenario type—trustworthy vs. untrustworthy. Table 8.6 shows this analysis.

	Trustworthy scenarios			Untrustworthy scenarios		
Email type	n	% correct overall	% correct without help	n	% correct overall	% correct without help
ABUSE	24	92%	**67%**	36	64%	14%
Plaintext	22	91%	**14%**	33	48%	18%
S/MIME	22	64%	**27%**	33	58%	18%
		$F = 4.24$ $p = .0186$	$F = 9.31$ $p = .0003$		$F = 0.83$ $p = .4405$	$F = 0.15$ $p = .8605$

Table 8.6: Success rates by type of scenario and whether subjects resorted to out-of-band channels before making a decision. In trustworthy scenarios, a significantly higher percentage of ABUSE users were correct overall, and correct without help (column in bold). In untrustworthy scenarios, we see no significant difference in percentage correct across email types, with or without help.

In trustworthy scenarios, a significantly higher percentage of ABUSE users were correct overall ($F = 4.24$, $P = .019$), though additional analysis (a Bonferroni test) shows that there is no statistical difference between the plaintext and ABUSE email groups. Both, however, are significantly higher than S/MIME. More importantly, a significantly higher percentage of ABUSE users were correct without help in trustworthy scenarios (column in bold) compared to plaintext and S/MIME users. These results support the hypothesis that ABUSE subjects can correctly identify trustworthy messages without getting help significantly more often than either the S/MIME or plaintext subjects. We also see that the subjects in all three groups were similarly able to resist attacks; there is no significant difference in percentage correct across email types, with or without help.

To examine the relationship between email type and success level in more depth, we performed a logistic regression analysis of the likelihood of success in trustworthy scenarios, controlling for any positive correlation between subjects seeking help and having success. We found a significant correlation ($p = .033$) between using ABUSE and the correct identification of trustworthy messages without seeking help.

ABUSE in untrustworthy scenarios

One might find it surprising, looking back at Table 8.6, that ABUSE users were no more successful than others in untrustworthy scenarios, with or without help. In these scenarios, use of ABUSE does not significantly correlate with willingness to forego help ($p = .193$). This implies that subjects in all three groups, when faced with an untrustworthy message, were similarly likely to resort to out-of-band channels before making their decision. We believe this can be explained by looking back at the incentive structure shown in Table 8.1. Recall that the penalty for going out-of-band in situations where the subject suspected that the message was an attack was quite low. We designed the study to attempt to mimic real world costs and benefits; when the subject believes he is being attacked, and the correct response to an attack is to do nothing, there is little harm in taking extra time to be certain. Thus, it is likely that the study disincentive was low enough that subjects almost always sought help when they believed they were being attacked, though we cannot be sure this is the case. Denying subjects access to out-of-band channels in some cases could have allowed us to tease out this information, but we were ethically prevented from doing so. We can, however, say that subjects using ABUSE are no more likely to fall for attacks than subjects in the other groups, showing that ABUSE exhibits *attack resistance* as defined back in Section 1.4.3.

There may have been ways to work around the ethics issue mentioned above. For example, we could have forced subjects to record a choice before allowing them access to out-of-band context information. In this way, we could have determined what the subjects would have done if the "phone" was unavailable or the company directory was down; it is possible that subjects using with ABUSE might have been more successful when forced to decide in the absence of out-of-band channels. It is also possible, however, that forcing subjects to come to a decision before getting the extra context information would have impacted their behavior on future tasks. Asking subjects to review their answers allowed us to gather similar information without impacting the subjects' decision making process. We review this information in the next section.

Analyzing the "regret" data

In addition to commenting on their decisions at the conclusion of the study, the subjects were asked which choices they would change, given the chance. They were *not* provided with the correct answers at any point. Looking at the regret data for all subjects does not allow for any conclusions about messaging technologies; subjects who consulted out-of-band sources have multiple channels of access to information, all of which may have impacted their post-facto feelings of certainty about their decision. By taking only the observations in which subjects did *not* request out-of-band help ($n = 81$) and looking at their answers to the question of regret (Figure 8.7), it is possible to see some trends emerge. Subjects in all three groups were similarly likely to feel no regret over their choices. However, it is clear that subjects in the ABUSE group were much more justified in this sentiment; they were wrong much less often. In both of the other groups, subjects had similar levels of confidence (no regrets) but were more often wrong.

It is also interesting to note that in rounds in which subjects did not seek help, subjects in the ABUSE group who felt confident were more likely to have made the right choice. There seems to be a link, then, between the presence of ABUSE attributes and subjects

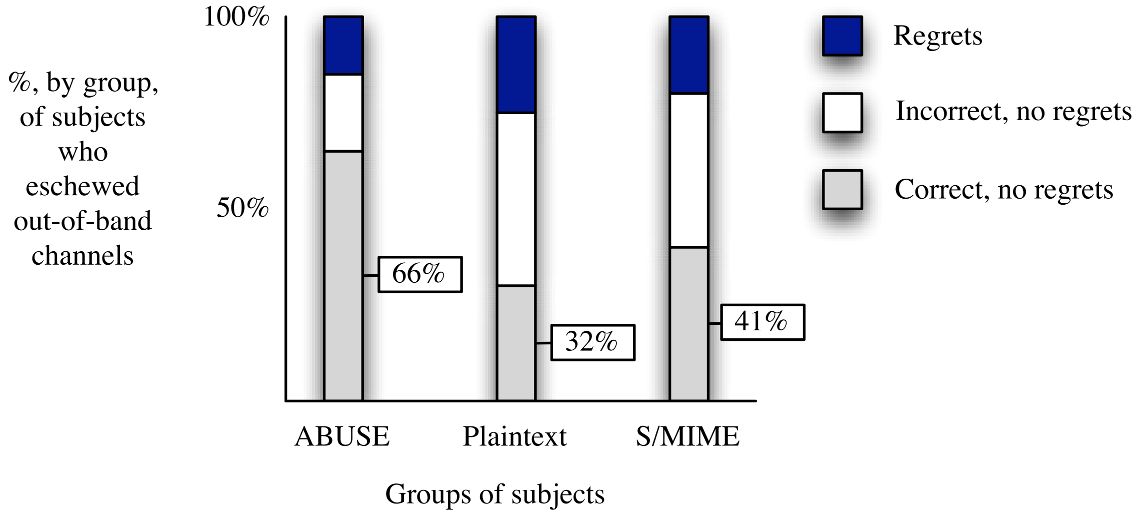


Figure 8.7: Regret data among subjects who did not consult out-of-band channels. Comparing the frequency with which subjects were correct with no regrets, ABUSE comes out significantly better ($F = 3.61$, $p = .0307$).

being confidently correct. This lends credence to the idea, supported by comments from the review phase, that subjects are reading and understanding the ABUSE attributes. If the subjects were treating the mere presence of attributes as an indicator that messages were trustworthy, one would expect to see high levels of confidence regardless of whether they were right or not. However, while ABUSE subjects regret their decision at a rate similar to that of subjects in the other groups, the percent of "confidently wrong" subjects is much lower. At the same time, the percent of subjects in the ABUSE group who are "confidently right" is much higher. The attributes bound to ABUSE messages allow them to make the right decisions more often, and feel confident in their choices.

Other interesting results

Looking at the observations in trustworthy scenarios within Table 8.6 again, it is interesting to note that the S/MIME group performed significantly worse when compared to the plaintext group. This is counterintuitive; S/MIME, when compared to plaintext email, is supposed to help users better identify messages that are trustworthy! The next column in the table shows that, when the subjects seek help, the S/MIME group and the plaintext group perform comparably. The discrepancy in the numbers for the plaintext group indicates that most of those subjects relied on out-of-band channels for information and subsequently made the correct decision. The lesser discrepancy in the S/MIME groups numbers shows that either they are eschewing assistance and getting the decision wrong, or they are getting help and still getting the decision wrong. As shown in Figure 8.8, the

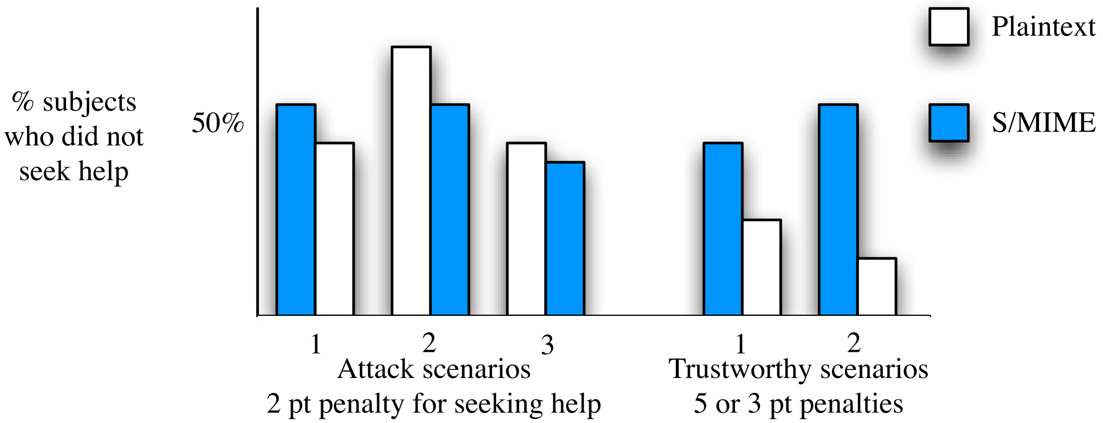


Figure 8.8: Percentage of S/MIME and plaintext subjects willing to do without out-of-band channels, per scenario.

subjects in the S/MIME group fell back on out-of-band channels with a similar frequency in every scenario. The subject in the plaintext group, however, did not. When the penalty for consulting out-band-channels was higher, the subjects in the latter group became less willing to go out on a limb. The willingness of the S/MIME subjects to accept risk remained the same despite the differing incentives, leading to more incorrect decisions. This indicates not only that the presence of S/MIME signals was noticed by the subjects in that group, but also that some of them seem to have been given a false sense of confidence in their own decision-making process by the presence of the simulated digital signatures.

8.4 Summary and conclusions

Using scenarios drawn from the August 2003 blackout, we developed a decision-making game to compare how well users were able to correctly extend trust to third parties they do not know when equipped with plaintext email, S/MIME and ABUSE-enhanced mail. We found that subjects equipped with ABUSE were able to identify trustworthy messages without needing to resort to out-of-band channels significantly more often than subjects using S/MIME or plaintext email.

We also found that the presence of ABUSE attributes correlated favorably with higher rates of subjects reporting post-facto confidence in correct trust decisions. In both of the other groups, subjects were confidently right and confidently wrong with similar frequency. This provides initial support for the assertion that subjects actually understand what the attributes bound to ABUSE messages mean.

The study also uncovered a result totally unrelated to ABUSE. Subjects in these power grid scenarios seemed to miss trustworthy messages more often when equipped with S/MIME than with even just plaintext email. This correlated with the increase in penalty for resorting to out-of-band channels in trustworthy scenarios; when subjects had something to lose, S/MIME made them more willing to go out on a limb. They wound up losing more frequently than the plaintext users who, because they really did not have much to go on, played it safe, took the help, and took the smaller number of more certain points.

Taking all this together, it seems that the ABUSE approach has promise. Questions about how much users actually understand about ABUSE attributes abound, however. We address these questions in Chapter 9.

Chapter 9

Evaluation: Abusing Johnny

The second user study that we performed to evaluate ABUSE focused on finding qualitative evidence that users could understand the information conveyed by the ABUSE attribute presentation GUI. Our hypothesis, then, is as follows:

> When presented with an ABUSE-enhanced email, users notice the attributes bound to the message, process them accurately, and use the information to make a rational trust decision about the request contained in the message text.

A *rational trust decision* in this case means a decision that follows logically from the user's internal trust policy, given the information she gleaned from the message and its attendant attributes. Recall from the discussion in Section 3.1.1 that users do indeed *have* these internal trust policies; they decide to whom to release private information every day, in a "nuanced and seamless manner," constantly refining their thought process, re-categorizing individuals, and encountering exceptional cases. [1] The challenge for us as we designed this experiment was to determine how to get users to feel compelled to process attribute information, run it through their mental trust model, and relate to us their thought process while coming to a trust decision.

9.1 Experiment overview

As in our power-grid-inspired user study, we sought to provide users with a scenario that motivated them to make decisions about acting on *individual* incoming email requests from people they did not know. This time, however, we wanted to provide a *series* of related tasks; in this fashion, we could allow subjects to build at least some semblance of a trust relationship with the characters in the study and investigate the ability of ABSUE to express trust flows that involve process-based relationships in some way. We also wanted to make sure that we could create messages that would express all of the different trust flows detailed in Chapter 2. Thus, showing that users could understand the messages in the study would support the conclusion that ABUSE is sufficiently expressive, according to the definition of expressiveness that we advanced in Chapter 3.

The subjects again needed a task to focus on outside of deciding email trust; in our earlier study, they were focused on keeping the power grid stable—a task that has important real-world implications, but with which our subjects were completely unfamiliar. In particular, the risks of failure were somewhat abstract to kinds of subjects to which we have access. We addressed this by providing an incentive structure that mimicked the trade-offs experienced by power grid operators in the field. In this study, we wanted to find a scenario that was more accessible to our subject base. Following in the footsteps of previous research into secure email usability [47, 50], we based our study on the trappings used by Whitten and Tygar in their seminal study of PGP usability "Why Johnny Can't Encrypt." [125] Hence the name of our study: "Abusing Johnny."

In our study, the subject is placed in the role of a political campaign volunteer who is charged with maintaining his candidate's schedule for the next week. This seemed particularly appropriate, as we performed this study in the midst of presidential primary campaign season. The subject is to update the schedule in response to authorized requests and to distribute it to other individuals working on the campaign upon request—but no one else. The main difference here is that previous work by other researchers did not concern itself with attackers adding events to or removing events from the schedule. This gives us the opportunity to create a wider range of interesting trustworthy and untrustworthy messages. Similar to previous work, we introduced the subjects to this scenario, asked them to think out loud while taking the study, and then took copious notes about their behavior while they received a series of email messages. We also recorded the emails they sent and received for later analysis.

In addition to modifying the subject's task from the original study and Garfinkel's follow-on "Johnny 2" [47, 50], we have expanded the campaign scenario used in previous experiments by adding a wider range of characters with a more diverse set of characteristics. Our cast is outlined in Table 9.1. The subject and campaign members work for Senator Oman, one of two fictional Democratic presidential candidates represented in the study. Every character is in some way affiliated with the Democratic party, though not all are aligned with a particular campaign. The attackers in our experiment are drawn from the set of "DNC insiders", though not every persona on this list attacks every subject. The particular attackers vary across the subject groups, and we detail this in Section 9.2.4. For now, we discuss the attackers in general. All of the attackers are attempting to use their

Experimental subject:		
Campaign Coordinator	ccord@dnc.org	The subjects are told: “You are the Campaign Coordinator.”
Campaign insiders:		
Brian Oman	no email	The Senator for whose campaign the subject works.
Maria Page	mariap@dnc.org	Oman’s Campaign Manager; Subject’s boss.
Paul Schmidt	pauls@dnc.org	Oman’s Chief Strategist. Can authorize schedule changes.
Ben Donnelly	bend@dnc.org	Paul’s assistant. Also recruits short-term volunteers at campaign stops.
Dana McIntyre	dmi@dnc.org	The former Campaign Coordinator. Went on maternity leave.
DNC insiders:		
Hannah Copeland	no email	Senator Oman’s opponent.
John Oren	no email	Copeland’s Campaign Manager.
Jane Smith	janes@dnc.org	John’s assistant.
Bill Brasky	billb@gmail.com	A former volunteer on the Oman campaign.
George Kontos	georgek@gmail.com	Varies across groups.
Phoebe Mann	no email	Varies across groups.
Frank Shemp	franks@dnc.org	varies across groups.
Paul Schmidt	pschmidt@dnc.org	A second Paul Schmidt, also employed by the DNC.

Table 9.1: Abusing Johnny cast of characters. All characters with email addresses actually communicate directly with the subject. The others are referenced either in text or in attributes.

affiliation with the Democratic party to trick the subject into mistakenly modifying or releasing the schedule. Some even have some kind of connection to characters on the list of "campaign insiders" that they try to leverage. They are all equipped with knowledge of the campaign personnel, perhaps obtained from the candidate's website or by some other sort of social engineering attack. The attackers do not collude with one another, and none of the legitimate campaign staff are colluding with any of the attackers either. Due to a variety of circumstances, the other campaign staff are all out of the office; despite it being the subject's first day on the job, he is alone at work. Furthermore, one of the attackers has decided to increase his odds of success by jamming the phone lines at campaign headquarters, denying the subjects use of out-of-band channels for trust-building. Such attacks are not that difficult to carry out, and have in fact been executed before, in the context of a political campaign. New Hampshire's Democratic party was victimized by a phone-jamming scheme in 2002, designed to interrupt their efforts to get voters to the polling place on Election Day. [113]

Thus, we have designed an experiment around a scenario that has already been accepted in the literature as being suitably motivating to cause subjects to buy in and legitimately try to complete the tasks set before them.

9.2 Methodology

9.2.1 The three subject groups

Like our first experiment, this study employed a between-subjects design. The subjects were randomly assigned to one of three groups. All three groups saw the same validly-signed message content sent by the same senders. The first group, who we will call the *control* group, saw no ABUSE content. The other two groups, *ABUSE-one* and *ABUSE-two*, saw different sets of ABUSE attributes over the course of the study. For a given message, subjects would see the same text, signed by the same sender, but in some cases subjects in ABUSE-one would see different attributes bound to the message than subjects in ABUSE-two. For one group, the presented attributes would justify taking action on the message; the other group would see a different set of attributes, which should not lead them to trust the accompanying message. The exact same message, received under the exact same circumstances, should be heeded when accompanied by one set of attributes and a ignored when accompanied by another.

Message	Sender	Content
Welcome	Maria Page	Welcomes the subject to the campaign, informs him that things are very busy, and that Dana just left on maternity leave. Adds that some important information is forthcoming.
Setup	Maria Page	Provides subject with schedule and task details. Indicates updates should have go-ahead from either her or Chief Strategist Paul Schmidt. Says that legitimate campaign staff should be given schedule quickly upon request. Introduces the opponent (Senator Copeland), and suggests that her people might be actively trying to create a scheduling snafu for Oman. Signs off saying she tried to set her phone up to access email on the road, but it might not be working; she might not be consistently available.
Update	Ben Donnelly	Ben says that Paul has set up new events; asks for them to be added. Also requests a copy of new, up-to-date schedule.

Table 9.2: The setup messages for the study. These are always received in order and have the same attributes for subjects in both ABUSE groups. The control group sees these (and all other) messages with no attributes.

9.2.2 Abusing Johnny message overview

The Abusing Johnny study consisted of ten email messages sent to the subject, who was playing the role of a Campaign Coordinator on fictional Democratic Senator Oman's presidential primary campaign. In the scenario presented, the campaign was ramping up for the Pennsylvania primary election when their former Coordinator had a baby and went on maternity leave; the subject stepped in for her in the middle of a very busy time. Over the course of the first three messages, which the subject always received in order, he is introduced to the campaign, informed of the details of his task, provided with the campaign schedule, and asked to update the schedule. These three messages are summarized in Table 9.2. The attributes bound to all three are the same in both ABUSE groups. The purpose of these messages is threefold:

- introduce the subject to the task,
- introduce him to Maria (his boss, with whom he shares process-based trust), and
- to present him with some simple trust relationships expressed in ABUSE attributes.

We provide a detailed discussion of the message content and attributes in Section 9.2.3.

After the three setup messages, subjects began the meat of the task. They received six messages, the order of which was randomized across subjects to control for order effects as in our previous study. Recall, however, that the attributes bound to some of these messages differ between the two ABUSE groups. There are three messages that have the

Message	Sender	Content
New Volunteer	George Kontos	George asks to receive the schedule, saying that Ben recruited him as a volunteer in Philadelphia.
Consultant	Frank Shemp	Frank, claiming to be a consultant who works for the party, says that he's going to be working with the campaign in Pittsburgh. He asks urgently for the latest version of the schedule.
Coopetition	Jane Smith	Jane is the assistant to Senator Copeland's Campaign Manager. She indicates that the Senators are working together on an event, and that she needs to know Oman's schedule to set something up.
Former worker	Dana McIntyre	Dana, who the subject replaced, informs the subject of an event that she was working on setting up before she left. She that this event be added to the schedule.
Former volunteer	Bill Brasky	Bill indicates that he was hired as a volunteer by Ben, and complains that the campaign has failed to communicate well. He asks for the schedule for tomorrow's events in Pennsylvania.
"John Wilson"	Paul Schmidt	A Paul Schmidt asks for the schedule. *But this is not the Paul Schmidt who works for Senator Oman!*

Table 9.3: The six messages received during the meat of the study.

same attributes in both groups and three that differ. We summarize the content of all six in Table 9.3 and go through the nine different content/attribute pairings in Section 9.2.3.

The last message the subjects received was a simple wrap-up note from Maria, thanking them for completing the study and telling them to ask the experimenter for the debriefing interview form.

9.2.3 Abusing Johnny message details

We provide a detailed discussion of the messages presented to subjects for several reasons. First, the attributes bound to three of them differ across the two ABUSE groups; it is important to note these differences. Second, between the three setup messages and the nine different possible content/attribute combinations in the meat of the study, we have seven trustworthy messages and five attack messages. Among the seven trustworthy messages, each kind of trust flow detailed in Chapter 2 is expressed at least once. Thus, showing that users can understand the attributes on these messages shows that ABUSE is sufficiently expressive. The other five messages exhibit a selection of possible social engineering attacks that remain possible in ABUSE. We explain these attacks in Table 9.4. Two of these attacks, the *no attribute* and *vague attribute* attacks, were used by the adversaries in our other study. The adversaries in this study are a bit smarter. Showing that users can recognize these attacks supports our assertion that ABUSE is attack resistant (as defined in Chapter 3). In Table 9.5 we go through the messages received by the subjects in the two

Attack	Explanation
Expired attribute	The attacker attempts to leverage an expired attribute to earn trust he does not deserve.
Nonsense chain	The attacker binds an attribute whose assertions do not logically follow from one another. Requires collusion on the part of some issuer in the attribute chain.
No attribute	The attacker binds no attributes, trying to convince the subject to trust him using the message body alone.
Vague attribute	The attacker binds a valid, sensical attribute to his message, but not one that confers authority for the accompanying request.
"John Wilson"[1]	The attacker's name is similar to someone in a position of authority. He tries to leverage this to get the subject to trust him. Best combined with the no-attribute or vague-attribute attacks.

Table 9.4: The kinds of attacks explored in Abusing Johnny.

Message	ABUSE-one	ABUSE-two
Welcome	Process-based trust	Process-based trust
Setup	Process-based trust	Process-based trust
Update	Role-based delegation, Role-based trust	Role-based delegation, Role-based trust
New Volunteer	Role-sourced arbitrary delegation, Role-based delegation	No attribute
Consultant	Nonsense chain	Friend-sourced arbitrary delegation
Coopetition	Coopetition	Vague attribute
Former Worker	Non-contemporaneous trust	Non-contemporaneous trust
Former Volunteer	Expired attribute	Expired attribute
"John Wilson"	"John Wilson"	"John Wilson"

Table 9.5: The kinds of attacks and trust flows expressed by the messages as received by each of the two ABUSE groups.

ABUSE groups and indicate which are trustworthy and which are attacks. If trustworthy, we indicate which trust flow(s) they are expressing; if not, we indicate which attack the message sender is attempting to execute.

Trustworthy messages

Welcome and Setup In both groups, these messages leverage process-based trust that the subject shares with Maria. He is told beforehand that he knows her.

Attributes:

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager

Update In this message, Ben asks the subject to change the schedule on Paul's behalf. This is a *role-based delegation* trust flow, introduced in Section 2.3.2. The subject knows from the earlier setup messages that Paul is allowed to update the schedule; Ben's role as his assistant implies that Ben can relay information on Paul's behalf. A shared understanding of the relationship between assistants and their busy bosses should exist between Ben and the subject, causing trust to flow from Paul to Ben. Ben's request for the schedule information later in the same message should be heeded due to a simple *role-based trust flow* (Section 2.3.1); he works on the campaign, and thus is entitled to ask for the schedule. Heeding Ben's requests would indicate that the subjects understood that these trust flows were active.

Attributes:

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Paul Schmidt	is my Chief Strategist
Paul Schmidt says that	Ben Donnelly	is my Exec. Assistant

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	Ben Donnelly	recruits new temporary workers in Philadelphia

New volunteer, ABUSE-one As shown above, Ben Donnelly possesses an attribute that was given to him by Maria, indicating that she has granted him the right to designate new volunteers in Philadelphia. There is a *role-sourced arbitrary delegation* flow (Section 2.3.2) from Maria to Ben based on this attribute. In the version of the New Volunteer message seen by subjects in ABUSE-one, George Kontos leverages this flow, and then also a role-based delegation flow deriving from the recruiter-volunteer relationship between Ben and himself.

Attributes:

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	Ben Donnelly	recruits new temporary workers in Philadelphia
Ben Donnelly says that	George Kontos	is a local campaign coordinator

Consultant, ABUSE-two In the version of the Consultant message seen by subjects in ABUSE-two, Frank binds to his message two attributes: one indicating that he is a campaign consultant active in Pittsburgh, and one from Maria indicating that he should have access to information relating to the Oman campaign for the purpose of assisting in events being run there. Maria, who shares process-based trust with the subject, delegates a subset of her rights (access to information about Pittsburgh events) to Frank. This is a *friend-sourced arbitrary delegation* trust flow (Section 2.3.2).

Attributes:

Democratic Party says that	Phoebe Mann	is the Pittsburgh Party Chair
Phoebe Mann says that	Frank Shemp	is a Local Campaign Consultant

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	Frank Shemp	is working with us in Pittsburgh

Coopetition, ABUSE-one In the version of the Coopetition message seen by subjects in ABUSE-one, Jane Smith indicates that the Senators Oman and Copeland are working together on an event, and that she needs to know Oman's schedule to set something up. Jane binds two attributes to her message: one indicating that she is the assistant to Senator Copeland's Campaign Manager and one from Maria indicating that she condones Jane's request for the schedule in this case. As the subject is pre-disposed to distrust Jane (she works for the opponent!), Maria's explicit blessing constitutes a *coopetition* flow to Jane (Section 2.3.3).

Attributes:

Democratic Party says that	Hannah Copeland	is a Senator
Hannah Copeland says that	John Oren	is my Campaign Manager
John Oren says that	Jane Smith	is my Exec. Assistant

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	John Oren	can ask about Oman Harrisburg Schedule
John Oren says that	Jane Smith	can ask about Oman Harrisburg Schedule

Former worker Dana McIntyre, whom you replaced, informs the subject of an event that she was working on setting up before she left. She binds to her message her old attribute from Maria that indicates she was the Campaign Coordinator and asks the subject to add this event to the schedule. Though Dana's attribute is expired, the message content indicates that she was working on this event back when she was still with the campaign. The subject should understand that Dana had the appropriate authority back when this decision was made; thus it makes sense to accept the role-based trust that used to exist; an example of a *non-contemporaneous trust flow* (Section 2.3.4).

Attributes:

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	Dana McIntyre	is our Campaign Coordinator (expired)

The five attacks

New volunteer, ABUSE-two In the version of George's message seen by subjects in ABUSE-two, he claims to be a new volunteer hired by Ben, but provides no evidentiary support. He then asks for the schedule. This is an example of the *no attribute* attack. Recall that the subjects in this group do not see the other version of George's message, only this one. This allows us to pinpoint the ABUSE attribute as the cause if the two groups perform differently on this message.

Consultant, ABUSE-one The content of Frank's message seen by subjects in ABUSE-one is the same as indicated above. However, in this attack, he binds only one attribute to his message. The attribute says that Frank is a local campaign consultant, but it is chained off of an attribute that confers only "Full-time Employee" status on the holder. This is an example of the *nonsense chain* attack.

Attributes:

Democratic Party says that	Phoebe Mann	is a Full-time Employee
Phoebe Mann says that	Frank Shemp	is a Local Campaign Consultant

Coopetition, ABUSE-two The version of the Coopetition message seen by subjects in ABUSE-two has only one attribute bound to it as well. The attribute indicates that Jane Smith works as John Oren's assistant. The text of the message, as in the trustworthy case, claims that Maria is there in Senator Copeland's office but incommunicado. However, Jane provides no evidence that Maria has, in fact, delegated to her the right to ask for Senator Oman's schedule. This is an example of the *vague attribute* attack that was also used in our previous study.

Attributes:

Democratic Party says that	Hannah Copeland	is a Senator
Hannah Copeland says that	John Oren	is my Campaign Manager
John Oren says that	Jane Smith	is my Exec. Assistant

Expired attribute Bill Brasky indicates that he was hired by Ben, and asks for the schedule for tomorrow's events in Pennsylvania. He binds to his message an attribute saying that he is a volunteer, but it is expired; it is also chained off of an expired attribute indicating that Ben "recruits new temp workers in New York." Bill is attempting to use the *expired attribute* attack.

Attributes:

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager
Maria Page says that	Ben Donnelly	recruits new temp. workers in New York (expired)
Ben Donnelly says that	Bill Brasky	is a local campaign coordinator (expired)

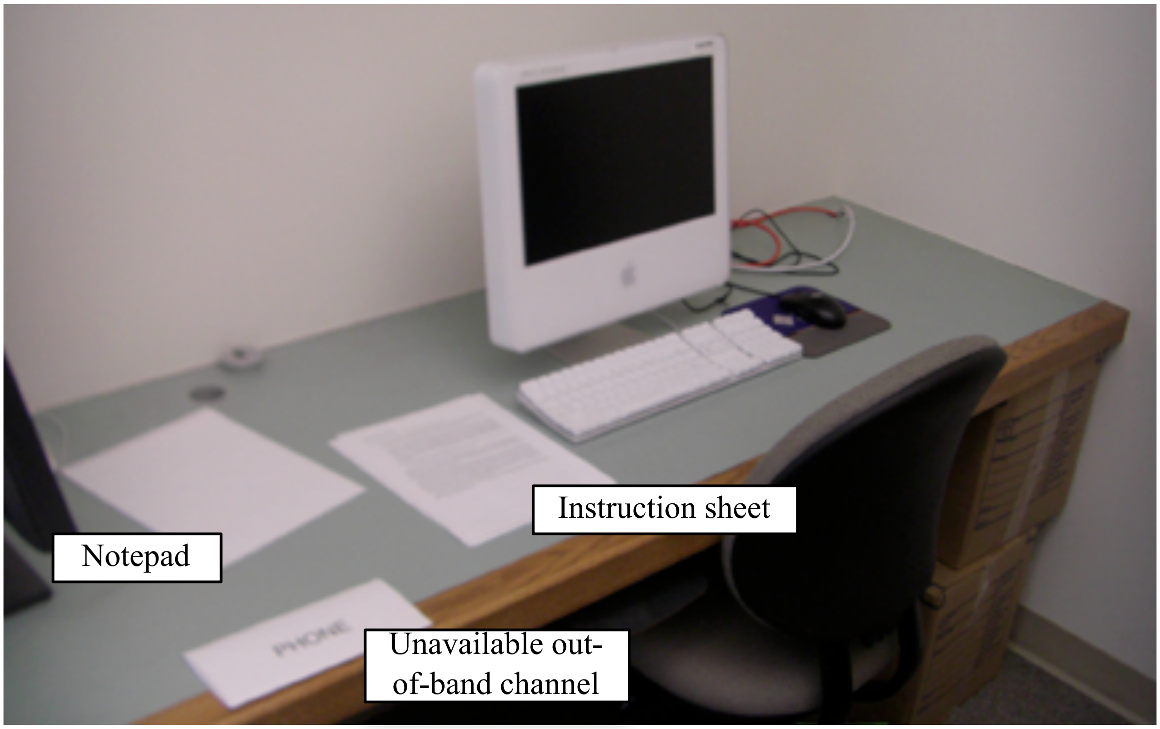


Figure 9.1: The study environment. We provided a "phone" to determine the frequency with which subjects would want to resort to out-of-band channels. The "phone" provided no useful information, however.

"John Wilson" Obviously, this message is executing the "John Wilson" attack. A Paul Schmidt asks for the schedule, binding an attribute that indicates he works for the DNC. *But this is not the Paul Schmidt who works for Senator Oman!*

Attributes:

Democratic Party says that Paul Schmidt is a Full-time Employee

9.2.4 The protocol

This study employed a between-subjects design to examine how well users understand attribute content. The subjects were randomly assigned to one of the three groups discussed in Section 9.2.1. After arriving in the study location (Figure 9.1), all subjects received a study procedure information sheet containing information shown in Figure B.3. This sheet served several purposes:

- Ask the subjects to "think aloud", as we would need to keep track of not only what they were doing, but also get insight into their thought process during the study.

> For the purposes of this study, you may assume that all email you send or receive can only be read by the sender and the intended recipient(s). Furthermore, you may assume that a message that appears to be from "Jane Doe" with the email address "jdoe@example.com" is actually from a person that the Democratic Party recognizes to be named "Jane Doe" with the email address "jdoe@example.com". Any further judgments about who Jane is or what Jane does must be made by you.

Figure 9.2: Text from the study procedure information sheet. Subjects are instructed to assume message secrecy, message integrity and sender assurance.

- Introduce the subject's role, Maria Page, and the two Senators.
- Introduce the task: maintain and distribute the schedule, but only upon authorized requests.
- Assure the subjects of sender authenticity and message integrity.

The text performing this last purpose is shown in Figure 9.2. Essentially, we are assuring the sender that the guarantees provided by correctly-implemented and correctly-used S/MIME are in place. The reason for this is to control for S/MIME usability problems. We are not interested in exploring problems that arise due to confusion about Certificate Authorities or ill-configured revocation checking, or any of the myriad of other issues that arise in a real S/MIME deployment. We already know those problems exist, and realize that there could be an interesting interplay between these issues and the use of ABUSE. However, we have already motivated that issues exist even in the face of perfect S/MIME; we developed ABUSE to attack those problems, and its efficacy in that space is what we explore in this experiment. Exploring into how ABUSE performs atop a real, potentially flawed S/MIME deployment is outside the scope of this work, though we propose such an investigation in Section 10.2.

In addition to the procedure information sheet, all subjects received a short pre-study briefing (Figure 9.3) to set the stage. The briefing received by the ABUSE groups had added content proving a short (less than one page) introduction to "digital introductions." This is in line with Garfinkel's approach in his revision [50] of Whitten's original study. [125] The idea is that, in an environment in which ABUSE would be deployed, users would not be asked to figure everything out on their own. They would have at least *some* help. The "training" we provided is shown in Figure 9.4; the experimenter would not answer any questions during the study, however, beyond those about basic Thunderbird functionality (sending mail, opening new mail, etc.).

After being given a brief primer on the message sending and receiving functionality of Mozilla Thunderbird, subjects were also informed about a text-editor on the computer that

You are the campaign coordinator.

You are working for the campaign manager, Maria Page, `mariap@dnc.org`. You have emailed with her before.

You have arrived early for work. No one else from the campaign is in the office.

If you wish to use the telephone to call a campaign member, please ask the experimenter for a "phone."

Maria will send you the initial campaign schedule. Once she has done this, wait for incoming messages from other people **working for** or **involved with** the campaign and follow any directions they give you.

Don't forget to "think aloud" as much as you can.

Figure 9.3: Content from the pre-study briefing seen by all groups.

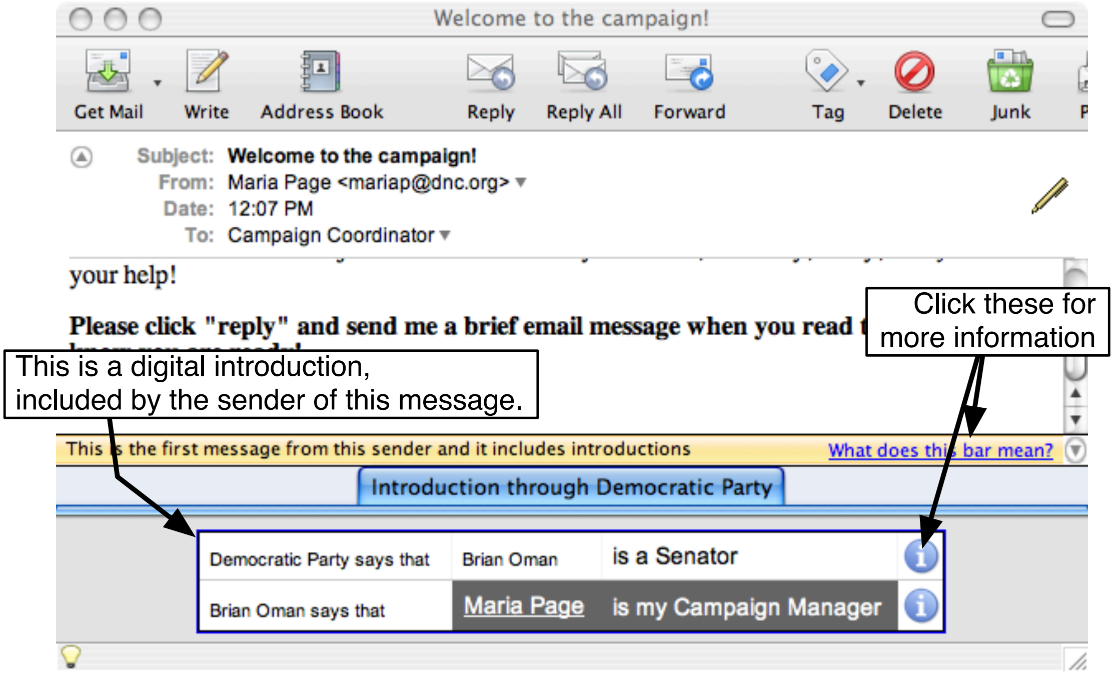
they could use for notes. Then, the subjects were sent the "Welcome" message, and the study began.

9.2.5 Running the study

Messages were pre-generated before the study began, and sent to the subjects using a web-based interface to custom command-line scripts. The ordering of the messages for a given trial was randomly determined by computer when the subject arrived. The next message in the sequence was sent by the experimenter when the subject had either responded to the current message as requested, or (by thinking aloud) indicated that they had decided to ignore the message at hand. Subjects were allowed to ask for a "phone" at any time, though upon doing so they would discover that the land lines were jammed and that they had forgotten to charge their cell phone. If the subjects became quiet for any period of time, they were gently reminded to think out loud. Upon completion of the task, subjects were given a debriefing questionnaire. Subjects in the control group received the form shown in Figure B.4, while the ABUSE subjects got a longer version, shown in Figure B.5. After filling this out to their satisfaction, subjects were thanked, paid, and allowed to leave.

NOTE: This version of Thunderbird has been configured to support Digital Introductions.

Digital Introductions allow senders to inform you of the ways in which they are connected to people you already know, or to the Democratic Party. The header bar of the Digital Introduction pane is always present at the bottom of the email window, colored blue if you have emailed with the sender before and yellow if you have not.



Subject: **Welcome to the campaign!**
From: Maria Page <mariap@dnc.org>
Date: 12:07 PM
To: Campaign Coordinator

your help!

Please click "reply" and send me a brief email message when you read t

Click these for more information

This is a digital introduction, included by the sender of this message.

This is the first message from this sender and it includes introductions — What does this bar mean?

Introduction through Democratic Party

Democratic Party says that	Brian Oman	is a Senator
Brian Oman says that	Maria Page	is my Campaign Manager

When a message is accompanied by new introductions, this pane will be automatically opened for you when the message is read.

An introduction reading "Alice says that Bob is a shepherd" means that Alice did, in fact, assert at some point that Bob is a shepherd. Whether you believe Alice or not is up to you, as is the interpretation of what this says to you about Bob.

Figure 9.4: The extra briefing seen by subjects in the ABUSE groups.

9.3 Results and discussion

The data we collected in this study provides qualitative evidence that users are able to understand the communication coming from the ABUSE attribute presentation GUI. Subjects exhibited an understanding of the six different kinds of trust flows enumerated in Chapter 2

1. role-based trust,
2. role-based delegation,
3. role-sourced arbitrary delegation,
4. friend-sources arbitrary delegation,
5. coopetition, and
6. non-contemporaneous trust.

In addition, subjects using ABUSE showed that they had not become any more vulnerable than the control group when attacked in any of the five ways we detailed in Table 9.4.

We saw a wide range of reaction to ABUSE in this study. There was one subject who never seemed to notice the "digital introductions" at all (S9). On his debriefing questionnaire, he said that he had thought they were advertisements, so he ignored them. One subject, after struggling to make a decision about whether to trust the "update" message from Ben, remembered the attributes, looked down at the pane and said "Oh, I can use these to tell...I totally forgot about that, that's awesome." (S27) Subject S12 initially did not want to trust Ben's message either, but then noticed the attribute from Paul that indicated Ben's status as his assistant and said "Oh, that's a very useful tool...Yeah...if he's Paul's executive assistant, ok." A third subject, S24, noted that it "seems that Ben's trustworthy, as Paul's executive assistant, because the digital introduction says he is. And I know who Paul is because my boss [Maria] said so." These statements, and the fact that every subject accepted Ben's message, support the conclusion that ABUSE can express role-based delegation. That is expected, however, as these are very simple and most familiar kinds of flows. We now go on to discuss our results with respect to more complex trust flows as well as the attacks that we executed on our subjects.

9.3.1 Comparing "New volunteer" across groups

In the ABUSE-one group, the "new volunteer" message was used to test role-sourced arbitrary delegation and role-based delegation. In the ABUSE-two group, it tested resistance to the no-attributes attack. No subject fell for the no-attributes attack. Conditioning the subjects to expect attributes on legitimate messages made them reject this attack out of hand. S36 pointed out "this message doesn't have any extra information about position or permissions," as he deleted the email. Another subject, S3, replied to George telling him to seek out an introduction from Ben before continuing to work with the campaign. A third, S13, immediately responded "No, I'm just not gonna send you *anything*."

Subjects in the ABUSE-one group, on the other hand, responded very positively to the message when accompanied by an attribute from Ben identifying George as a campaign volunteer. All but two chose to act on it. One of these subjects, S14, had expressed uncertainty about the whole concept of re-delegation. She did not accept any messages that did not have attributes in which Maria or Paul had issued the final assertion. Subjects in the control group, working without ABUSE at all, were highly unlikely to act on the message; only three out of twelve chose to respond with schedule information.

9.3.2 Comparing "Consultant" across groups

The "consultant" message is used to test both the nonsense-chain attack and friend-sourced arbitrary delegation. In the ABUSE-one group, subjects saw the message with an assertion chained off of a generic "employee" attribute. Subject S35 pointed out that Frank *said* he was in touch with Maria, but that has no attribute issued through her. Furthermore, he said, "the only person saying this guy's a consultant is just an Employee. So, that kind of concerns me." Subject S32 concurred: "[Frank] says he's been in touch with Maria, but there's no intro through Maria, so I don't know if I should trust him. I am not going to give [the schedule] to him." Compared to the control, in which 33% of the subjects acted on the consultant message, 46% of the subjects in this group chose to send Frank the information he wanted, despite his meaningless attribute.

These rates are comparable, especially when placed against the 93% in ABUSE-two who acted on Frank's message when it was accompanied by an attribute from Maria that expressed friend-sourced arbitrary delegation. Subject S15 inspected both attributes on Frank's message, clicked on Maria's name to view *her* attributes and, when satisfied that Frank's permission came from the "right Maria," decided to send Frank the information he requested. A second subject, S17, justified his response to Frank's email by saying that he "trust[s] old Maria!"

Comparing “Coopetition” across groups

The coopetition message is inherently suspicious. We expected subjects to be pre-disposed against trusting it, and we were correct. 17% of subjects in the control group acted on this message, with that number dropping to 15% in the ABUSE-two group, who saw the message paired with the vague-attribute attack. We already have confirmation from our previous study (Chapter 8) that ABUSE is resistant to vague-attribute attacks, so we are not concerned about the general untrustworthiness of this message impacting our attack-resistance conclusions.

The percentage of ABUSE-one subjects who acted on the coopetition message when it was accompanied by a coopetition trust flow was 46%(6/13). However, there were three more users (23%) who clearly indicated that they understood what was being expressed by the attributes on the message; they simply remained leery of responding with sensitive information. This was usually because of a concern about delegation; S16 said “I’m reading Maria says that John Oren can ask about the schedule, and John Oren says that Jane Smith can ask,” but kept wanting to see Jane be “more directly connected” to Maria. Subject S32 expressed a concern over delegation even in the case of Ben’s original schedule update message (though she accepted it), and the coopetition message was suspicious enough to begin with that she could not overcome her concern about delegation. Other subjects, however, were fine with it, saying things like “If Maria actually said these things, if the introductions are accurate, this seems reasonable. Since 1PM doesn’t work, according to the most recent update, I’m going to send [Jane] the Harrisburg schedule.” (S19) Subject S35 offered this: “Ok, so someone is emailing me from the other side, from Copeland. However, Maria has said that John can ask about the Harrisburg schedule, and John said that this person can ask.”

9.3.3 On expiration

Since the study was structured as a single continuous experience, all the messages had to make sense within the plot. We could not devise a single message text that could be either trustworthy or untrustworthy given two different sets of expired attributes. Thus, we chose to have two separate messages, one which remained trustworthy despite its accompanying expired attribute and one which remained untrustworthy. We can still use this setup to figure out what we need to know:

- whether users understand the GUI elements that express expiration, and
- how binding expired attributes to a message impacts its perceived trustworthiness.

We can use their behavior and statements to determine the former, and there are two options for the latter: warning signals from expired attributes could make subjects more suspicious in general, or they might ignore expiration and become more trusting overall.

According to the results from the control group, the message “expired”, sent by Dana is easy to trust; eight of twelve subjects (66%) trusted it without the benefit of ABUSE. “Former volunteer,” from Bill, is less so; only three of twelve control subjects trusted this message. If users do not consider attributes, their expiration status, and message context in concert, we should expect to see trust rates for both messages impacted in the same fashion. If expiration makes messages more suspicious regardless of surrounding context, we should see Dana’s message being trusted less often; Bill’s should exhibit little change. Conversely, if appropriate attributes confer trust upon the messages to which they are bound regardless of expiration status, both messages should receive some kind of boost.

The numbers we see confirm that users pay attention to expiration status. Dana’s message was acted upon in 78% of cases across both ABUSE groups, Bill’s in only 15%. And, we know that subjects accepted valid attributes like the one bound to Bill’s message; George’s message with a similar attribute was accepted nearly universally. Furthermore, many subjects showed an inclination to explore the GUI elements that indicate that an assertion is outside of its validity period. Subject S17 noticed that “something’s in...hazard?” on Bill’s message before clicking for more information. S13 pointed out that he was rejecting Dana’s request because “she could be working for someone else now.” So, while he chose not to trust her, he clearly understood the information which the GUI was telling him. On Bill’s message, S12 clicked on the content of both expired assertions, pointing out that the dates had already passed (“5/1...that’s already happened”). When S16 opened Dana’s message, he exclaimed “Whoa! What’s this yellow thing?...this statement expired before?” Clearly, he understood what he was being told by the GUI. Subject S22, upon inspecting the attributes on Bill’s message pointed out that “Ben’s recruiting in Philly now, I guess he used to recruit in New York.” Subject S29 looked down and exclaimed “Why is there...? Oh, it’s expired.”

Taking this together, we certainly have evidence that subjects understand the way in which ABUSE expresses the validity status of assertions. They also showed that they became no more vulnerable to the expired attribute attack carried out here, and were not dissuaded from trusting Dana’s message—which was expressing a non-contemporaneous trust flow.

9.3.4 The “John Wilson” attack

This attack is very difficult to defend against, especially when the user is not personally familiar with the “John Wilson” being impersonated. Also, in the context of a study that requires subjects to make tricky trust decision after tricky trust decision, some jumped at what seemed like an easy choice. They saw the name Paul Schmidt and responded without thinking much. The numbers were consistent across the groups; six of twelve fell for the attack in the control group as opposed to $13/27$ in the ABUSE groups. However, the subjects who avoided this attack in the control group were mostly those who generally refused to trust messages in the study at all. Three of the six only trusted mail from Maria and no one else; two of the others trusted Maria and Dana alone. The ABUSE subjects who rejected the message did not show any such pattern, and many verbally indicated that it was odd that this message “doesn’t say he’s part of the campaign.” (S 32) Subject S33 specifically noted that he’s got an attribute indicating his employee status, but no specifics at all. S31 began to compose a reply, and then stopped himself when he looked back at the original message; he had noticed that *this* Paul Schmidt had not demonstrated that he was the Chief Strategist at all; this, he judged, was odd.

9.4 Conclusions

In this chapter, we detailed a study designed to provide qualitative evidence that users do, in fact, understand ABUSE attributes when presented by our GUI. We crafted a set of ABUSE-enhanced messages that expressed among them each kind of trust flow that we detailed in Chapter 2 and showed first-hand evidence that subjects viewed them, parsed them, understood them, and then made rational decisions based on the information they gleaned. In addition, we enhanced the attacker model from our previous user study, allowing for some more nefarious attribute-based social engineering attacks. Users showed the ability to avoid these attacks at least as well as a control group, though future researchers could expand the volume of subjects used in a study of this type to generate some stronger quantitative results in that area. Our goal was primarily to get qualitative information from users indicating that they understood the content presented to them by the ABUSE GUI and were capable of using it to inform their decisions. We believe we have achieved that.

Chapter 10

Conclusions and future work

In this chapter, we first discuss the places that we envision ABUSE going in the future. We answer some implementation questions, identify areas for future study, and provide initial thoughts on some potential follow-on experiments suggested by our research. Finally, we sum up the contributions made by this thesis and provide some concluding remarks.

10.1 Implementation issues

10.1.1 Decentralized attribute issuance

We acknowledged back in Section 5.2.2 that our prototype, centralized implementation of attribute issuance is less than ideal. Here, we discuss a decentralized protocol by which Alice can directly issue a new attribute to Bob, shown in Figure 10.1. Before engaging in this protocol, Alice provides to her client all the information to issue a new attribute: Bob's public key (gotten from his identity certificate), the desired content, the desired validity period, and the attribute off of which she wishes to chain. Alice and Bob begin by establishing a mutually authenticated SSL connection. Alice checks that Bob has presented a certificate with the expected public key, while Bob's only concern is that he can verify Alice's certificate. After this, Bob generates a key pair for the new attribute and returns the new public key. This step is necessary, as Bob would otherwise not be able to chain new assertions off of this attribute. Alice creates a new CSR from the received key and the information she specified earlier, which she then signs using the private key associated with the desired attribute. The resulting assertion, combined with the attribute off of which Alice chose to chain, is Bob's new attribute. Alice does not have the corresponding private key, however; Bob is the only one who has this information. Alice forwards the attribute to Bob, and the two part ways. Bob can choose to escrow this attribute in the centralized store,

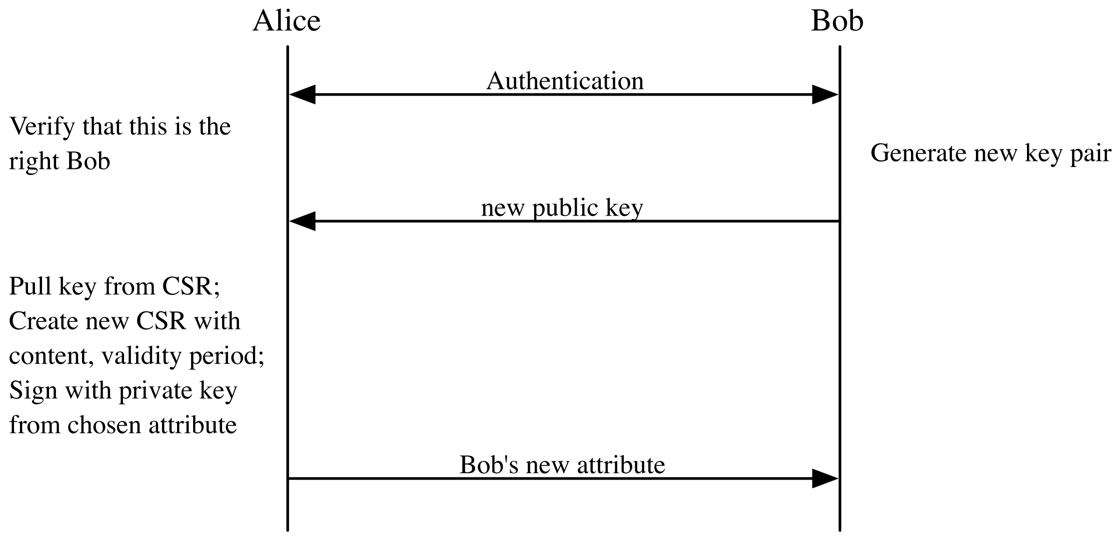

Alice
Bob
Authentication
Verify that this is the right Bob
Generate new key pair
new public key
Pull key from CSR; Create new CSR with content, validity period; Sign with private key from chosen attribute
Bob's new attribute

Figure 10.1: A distributed attribute issuance protocol. Using this method, the attribute store becomes nothing more than an attribute escrow service.

if he wants. To do so, he enciphers the whole thing (private key included) with his identity private key, so that only he can retrieve the stored information. The primary advantage of decentralizing the issuance process is that no user ever gives up control of the key material associated with his attributes; furthermore, users can issue attributes to one another even if the centralized store becomes unavailable

The downside of this approach is that Alice and Bob must both be available at the same time; at least, Bob's ABUSE client must be awake and able to perform cryptographic operations with his identity private key. It is possible to create an asynchronous version of this protocol using the centralized store as a go-between, but not as a key generation and signing service. Alice provides to her client all the information necessary to create a new attribute, which stores this information and associates it with a nonce. This nonce is signed with Alice's identity private key and uploaded it to the centralized store to be downloaded by Bob the next time he checks in. When Bob's client next connects to the store, it pulls down the nonce, verifies Alice's signature, and generates a new key pair. After signing the nonce and public key, Bob's client uploads them (along with his signature) to the store to wait for Alice. The key pair is cached along with the nonce. Alice's client, next time it connects to the store, pulls down the signed key/nonce pair and verifies Bob's signature. Using the nonce to find the appropriate assertion specification, Alice's client then creates a CSR and signs it with the appropriate attribute private key. This assertion and the attribute off which Alice chose to chain are uploaded to the store for Bob's client to download at its leisure. In this way, Bob's attribute keys still do not leave his control, but he never has to be available at the same time as Alice. We consider the possibilities of a scheme that uses

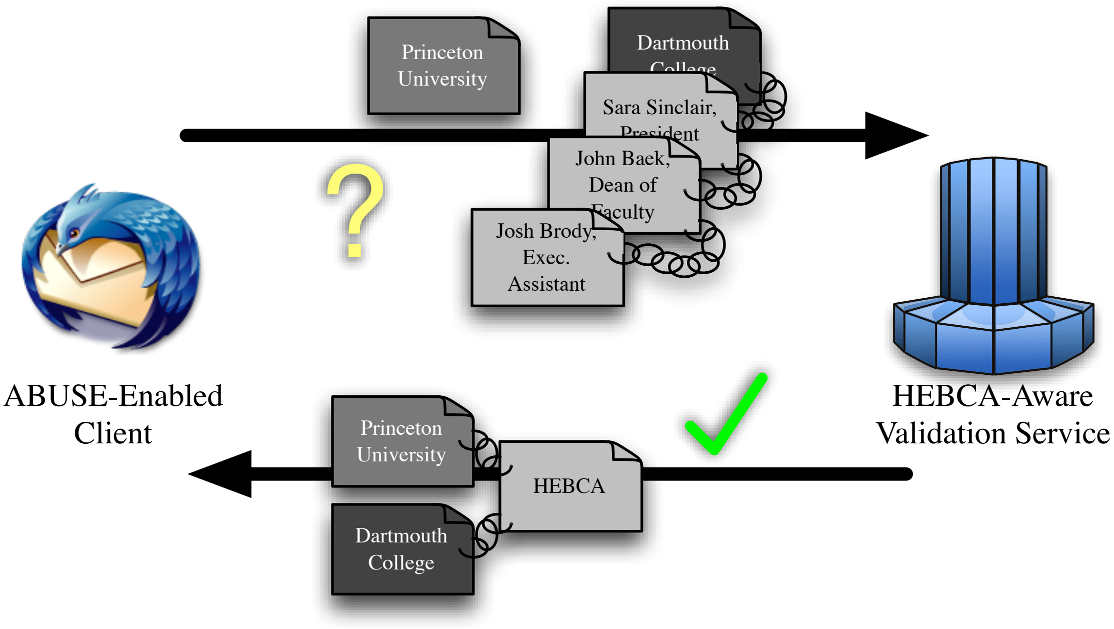


Figure 10.2: An ABUSE-enabled client using a HEBCA-aware validation service to verify an attribute from another institution. The client, at Princeton, trusts Princeton's root certificate. It presents the validation service with the attribute chain (rooted in Dartmouth's trust root) and its local trust root. The validation service, aware of the PKIs with which HEBCA has cross-certified, responds with a chain of certificates that connect Princeton's root to Dartmouth's root. The client can validate this chain on its own, allowing it to build a full path from its local trust root (Princeton) all the way down through the entire attribute chain.

only Bob and Alice's identity keys, potentially easing deployment, in Section 10.2.2.

10.1.2 ABUSE across domain boundaries

ABUSE relies upon users being able to interpret the meaning of assertion content. This becomes difficult when users try to communicate across domain boundaries. Dartmouth's "Dean of Pluralism and Leadership" might be roughly equivalent to the "Diversity Director" at another institution, for example. There is, essentially, a namespace issue. Domains such as the power grid are overseen collectively enough that these issues are alleviated. In that particular case, NERC provides guidelines that specify the names and duties of roles at the different organizations that interact during crises. [83] Terminology surrounding equipment and actions that can be taken is uniform enough that users can make themselves understood. In such a domain, where namespace issues are not particularly relevant, the problem is one of verifying attribute chains from foreign sources. Bridge Certificate Authorities [59] (discussed in Section 1.1.2) provide a method of joining disparate hierarchical X.509-based PKIs in a non-hierarchical way. By making both ABUSE attributes and

S/MIME digital signatures on ABUSE-enabled emails “bridge aware” (Figure 10.2), one could address the problem of verifying foreign attributes. The Higher Education Bridge Certification Authority (HEBCA) [56] has been set up and deployed at Dartmouth, so future researchers would be able to evaluate such bridged applications as they work with real, deployed infrastructure.

To take ABUSE beyond a single namespace (a single organization, or a domain like the power grid), both the above issue of attribute chain verification and also that of mapping unfamiliar attributes into a locally sensical context must be addressed.

To help users make sense of attributes created at outside institutions, one could consider applying some ontology mapping results from the W3C’s Semantic Web project. [122] Ontology mapping uses machine learning techniques to attempt to map one hierarchical classification structure onto another. [10, 29] An organization’s attribute space can be viewed as an ontology, since it is structured like a tree, rooted at the local CA. It is possible that an ontology mapper could be trained on this “home ontology” , and then treat incoming sets of attributes as portions of a foreign ontology and attempt to perform a mapping (Figure 10.3). While it is unlikely that this approach will provide a perfect solution, it should to be able to provide some kind of confidence measure along with the resultant mappings.[1]

10.2 Studying ABUSE in the real world

There are a number of questions about ABUSE that we cannot answer in a controlled laboratory setting:

- How granular are users when delegating permissions or vouching for other users? Will we see general assertions like “works for me” and “is an employee” or specific ones, like “can send email on my behalf about X” and “works on the foobar project as a product tester”?
- What kinds of validity periods are used in practice?
- What patterns arise in the selection of attributes to bind to messages? Do users frequently bind *all* their attributes to messages? None, unless they’re making a specific request? Would a default set be useful?

The only way to explore some of the issues surrounding ABUSE is to begin testing the system in a real-world deployment. Other researchers have realized the need for work such as this; Schneiderman names it “Science 2.0,” [108] and calls for the realization of

[1] Portions of the preceding section were adapted from my thesis proposal.

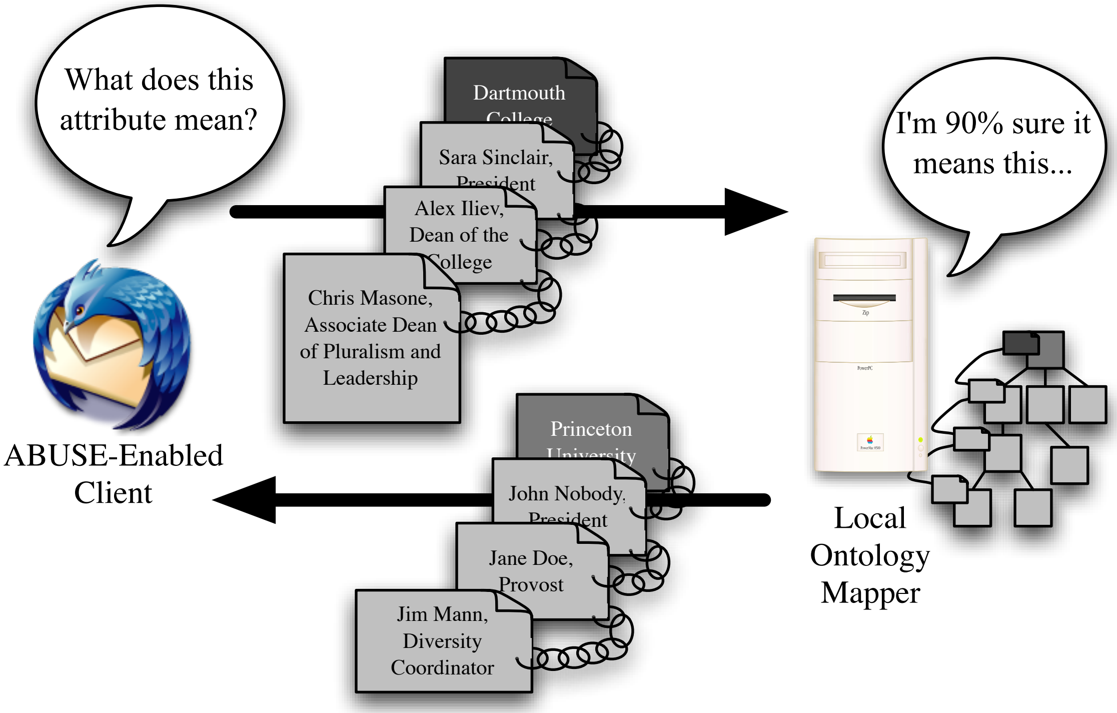

What does this attribute mean?
ABUSE-Enabled Client
Dartmouth College
Sara Sinclair, President
Alex Iliev, Dean of the College
Chris Masone, Associate Dean of Pluralism and Leadership
I'm 90% sure it means this...
Local Ontology Mapper
Princeton University
John Nobody, President
Jane Doe, Provost
Jim Mann, Diversity Coordinator

Figure 10.3: An ABUSE-enabled client using an ontology mapper at its home institution to try to place a foreign attribute in a local context. The attribute is forwarded to the ontology mapper, which attempts to find the local attribute that most closely matches it and return some kind of confidence measure in the matching. Performing the mapping may involve using a natural language database to attempt matches on the content of individual certificates in the attribute chain, in addition to trying to find similarities between the structure of the chain and portions of the home ontology.

replicable and generalizable conclusions through a greater commitment to case studies. We believe that this is the right approach to truly evaluate ABUSE, though it is often an uphill battle to convince an organization to deploy experimental technology that it did not develop in-house. As we mentioned in Section 6.1, Dartmouth is ill-suited for an ABUSE testbed due to the small and well-connected nature of the community here. A larger academic institution with a more impenetrable bureaucracy could be fruitful grounds for study. Most such institutions are already running centralized databases that comply with the eduperson profile [33]; this would provide excellent fodder for bootstrapping an ABUSE deployment. The academic calendar also provides a convenient set of recommended validity periods: term boundaries, well-defined vacation periods, exam periods, etc. A researcher might try the following:

- Deploy ABUSE, bootstrapped using eduperson profile data.
- Check email headers for the presence of attributes.
- Collect attributes as they are issued, store a hash, strip out subject/issuer information and keep just chains of assertion content.
- Track validity period information.
- Hash individual attributes seen in email headers, check against the collected attributes to determine usage patterns.
- Survey users periodically to determine what they are using the system for, if they have ever wanted to revoke an attribute, and for feedback about the usability of the system.

Researchers in NSF's TCIP project have connections to power companies and power grid management organizations. However, these users are managing critical infrastructure and it is unlikely they would use such an experimental technology in their day-to-day work. It might be possible to perform smaller-scale classical user studies on them, however, if the tasks can be kept very short.

10.2.1 Towards deploying ABUSE

ABUSE has significant deployability advantages over other technologies that attempt to address similar problems. The organization would need to roll out an updated email client, but since we have designed ABUSE to avoid push-back, there would be no need to do so for every user at the same time. Furthermore, many solutions to the problem we have considered would require such a step.[2] ABUSE requires neither the enumeration of all roles nor the specification of all permissions within an organization; deploying an algorithmic approach (Section 3.1.1), Role-Based Messaging or other related technologies would require at least some subset of these undertakings. Such approaches also require the specification of policies by some users within the system; as per Section 3.1.1, it is unclear that this is even possible. ABUSE imposes no such requirement. We do require an identity PKI, but this stipulation is not unique to ABUSE. Moreover, a PKI can be useful in a variety of applications, especially for industries that interact heavily with the federal government—like aerospace, pharmaceuticals, or the power grid. Above, we mention that ABUSE would also need to have the attribute space "bootstrapped" in some way; we mention the eduPerson LDAP profile as a starting point for higher education attributes, and expect that other organizations would have some similar set of employee information that could be used to create an initial set of user attributes. Setting the clients to bind one or two of these attributes to outgoing messages by default could begin to create interest in using the system; combined with user education, this approach should enable the organization to get a deployment off the ground with minimal hassle.

10.2.2 Reducing the number of key pairs

ABUSE does introduce one piece of complexity that may hamper deployment: every attribute comes with its own private key that must be protected. Designing the system thus allowed us to use commodity tools for the validation of ABUSE attributes, as well as parts of the issuance process. It should be possible to, by writing new code to handle attribute issuance and validation, modify ABUSE so that all signature operations are performed with users' identity private keys. Care would need to be taken to ensure that the modified attributes maintain the properties possessed by attributes in the current system; that assertions in a chain follow cryptographically in the appropriate order, that assertions cannot be transposed from one chain to another, and so on. If done appropriately, this modification would significantly reduce the number of keys that must be kept private, reducing the potential burden of deployment.

[2] A proxy-based solution in the spirit of the work discussed in Section 4.4.1 would require only client-side configuration changes

10.3 ABUSE attribute issuance studies

One arena in which we could potentially use grid operators is in the study of the attribute issuance process. The NERC Reliability Functional Model [83] provides a common set of roles for the industry, which could be a useful set of suggestions for assertion content. These do not cover the entire organizational structure, but a set that does so might be overlarge. Performing user studies on this population could help us to determine the "right" number of suggested assertions. The subjects would need to be familiarized with ABUSE and its approach, and then set to the task of issuing some set of attributes to some set of other users. We could also perform similar studies using an academic population, perhaps using the information we gather as feedback for adding the appropriate number of assertion content suggestions to a real-world case study deployment of the kind discussed in Section 10.2.

10.4 Investigating surprising our S/MIME result

In Section 8.3.2, we noted that data we took during our power-grid-inspired user study indicated that S/MIME users actually performed *worse* than plaintext email users on some tasks. Under the incentive structure we used, which rewarded users for deciding what to do about incoming email without seeking outside assistance, subjects in the S/MIME group were able to identify trustworthy messages significantly less often. They were not falling for attack messages, but they were deciding to forego asking for assistance on trustworthy messages, trying to earn more points at the cost of reassurance that their decisions were correct. Since the only difference between the two groups was the presence of simulated S/MIME signatures, we hypothesize that S/MIME signals, when noticed, make users more confident in their decisions about the trustworthiness of email—regardless of whether their decision is correct or not. It would be interesting to devise a new experiment, one that simulates just S/MIME and plaintext email, to validate this hypothesis.

10.5 Conclusion

In this dissertation, we have presented ABUSE, a system that enables users to build calculus-based trust with third parties with whom they communicate over the Internet. We first applied tools from the social sciences (economics, sociology, psychology, etc.) to real-world scenarios in order to understand the ways in which humans decide to trust people that they have never encountered before. Phone transcripts from the August 2003 North American blackout provided a rich set of example cases, as power grid operators at a variety of geographically distributed locations had to use the telephone to coordinate potentially risky actions and share sensitive information—without, in many cases, knowing each other beforehand. Using these test cases, and several that we drew from the academic setting, we set out to characterize the ways in which trust flowed in these decentralized, human-to-human scenarios.

We contribute a set of *trust flows* that allow for the classification of the kinds of trust scenarios with which we are concerned. These flows provided us with a guide for designing and a tool for evaluating our work as well; the system we created had to be expressive enough to enable humans to leverage any kind of flow that we had enumerated. Only such a system can successfully allow for the migration of human calculus-based trust into the digital world. We also contribute a set of design criteria for such a system, based on usably secure software design patterns and socio-technical research.

We contribute the design and implementation of ABUSE, a usably and usefully secure email system. By starting with the appropriate tools to understand the issues underlying the extension of human calculus-based trust and then designing with usability goals in mind from the start, we were able to create a system capable of expressing and reliably conveying to users the kinds of information they need to decide trust.

We evaluate ABUSE through user studies. The first is based directly on scenarios drawn from the power grid. ABUSE is compared to plaintext email and S/MIME, and determined to enable users to better identify trustworthy messages from senders that they do not know without needing to resort to out-of-band channels for assistance. This information, while useful, does not necessarily confirm that users are really understanding attribute content. To investigate this issue, we performed a second user study, based on a venerable scenario in secure email usability research. Subjects indicated by thinking aloud during the study, and through the answers on their debriefing questionnaires, that the information communicated by ABUSE was comprehensible and contributed to their ability to succeed at the task set before them.

The problem of human trust requires large amounts of human context to decide, and computers are ill-suited for these kinds of tasks. Our approach has been to build a system that gets the right information from one human to another, and then lets the relying party decide what she wants to do. Applying tools from the social science was a key part of exploring what that “right information” is, and we hope that more computer science researchers will take these tools into account when studying problems that involve users. One cannot create the right solution without first correctly characterizing the problem to be solved.

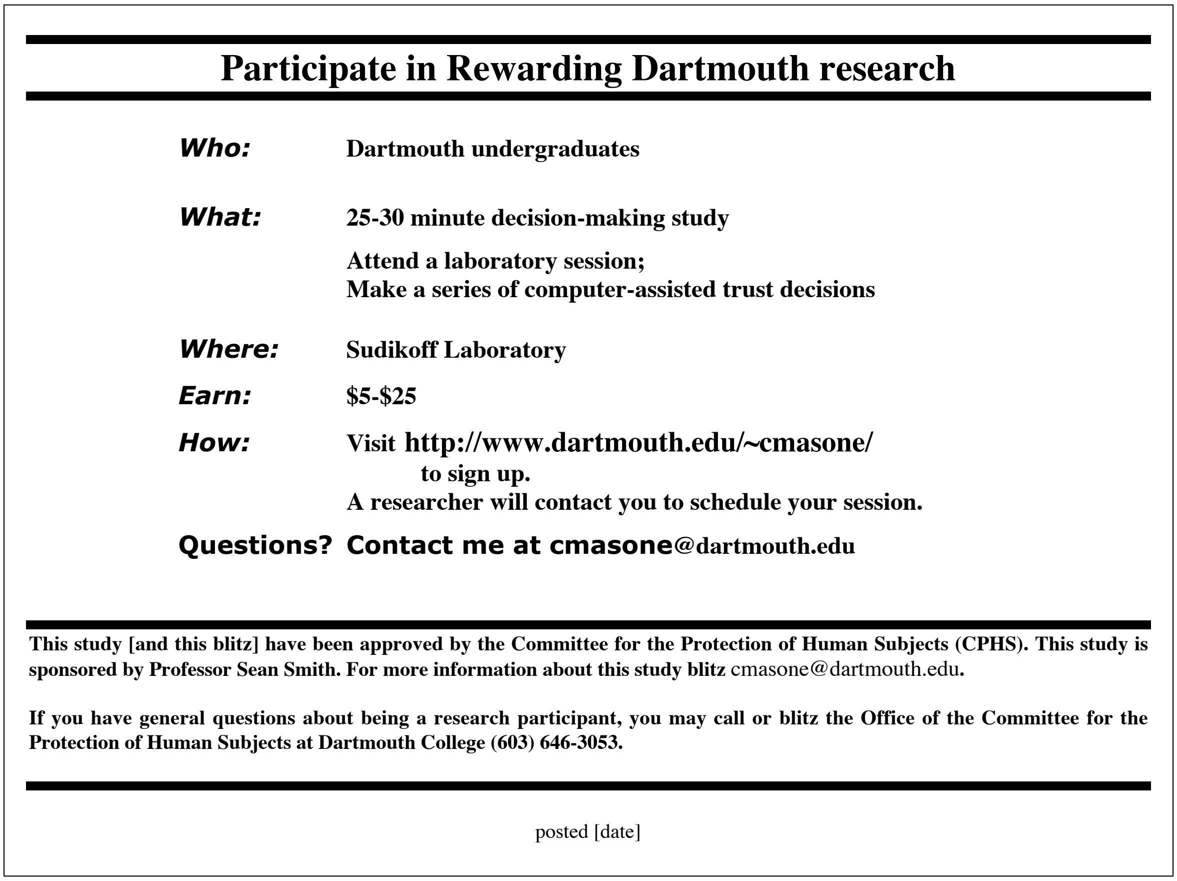
Appendix A

ABUSE in power grid scenarios appendix

In this appendix, we provide scenario details for our power-grid-based ABUSE user study. We also provide the materials viewed by the subjects.

A.1 Grid study recruitment

In addition to posting the flyer shown in Figure A.1, we sent an email with the same information out to a number of campus email lists, with an encouragement to forward it to anyone in the Dartmouth community who might be interested in participating.



Participate in Rewarding Dartmouth research

Who:	**Dartmouth undergraduates**
What:	**25-30 minute decision-making study** **Attend a laboratory session;** **Make a series of computer-assisted trust decisions**
Where:	**Sudikoff Laboratory**
Earn:	**$5-$25**
How:	**Visit http://www.dartmouth.edu/~cmasone/** **to sign up.** **A researcher will contact you to schedule your session.**
Questions?	**Contact me at cmasone@dartmouth.edu**

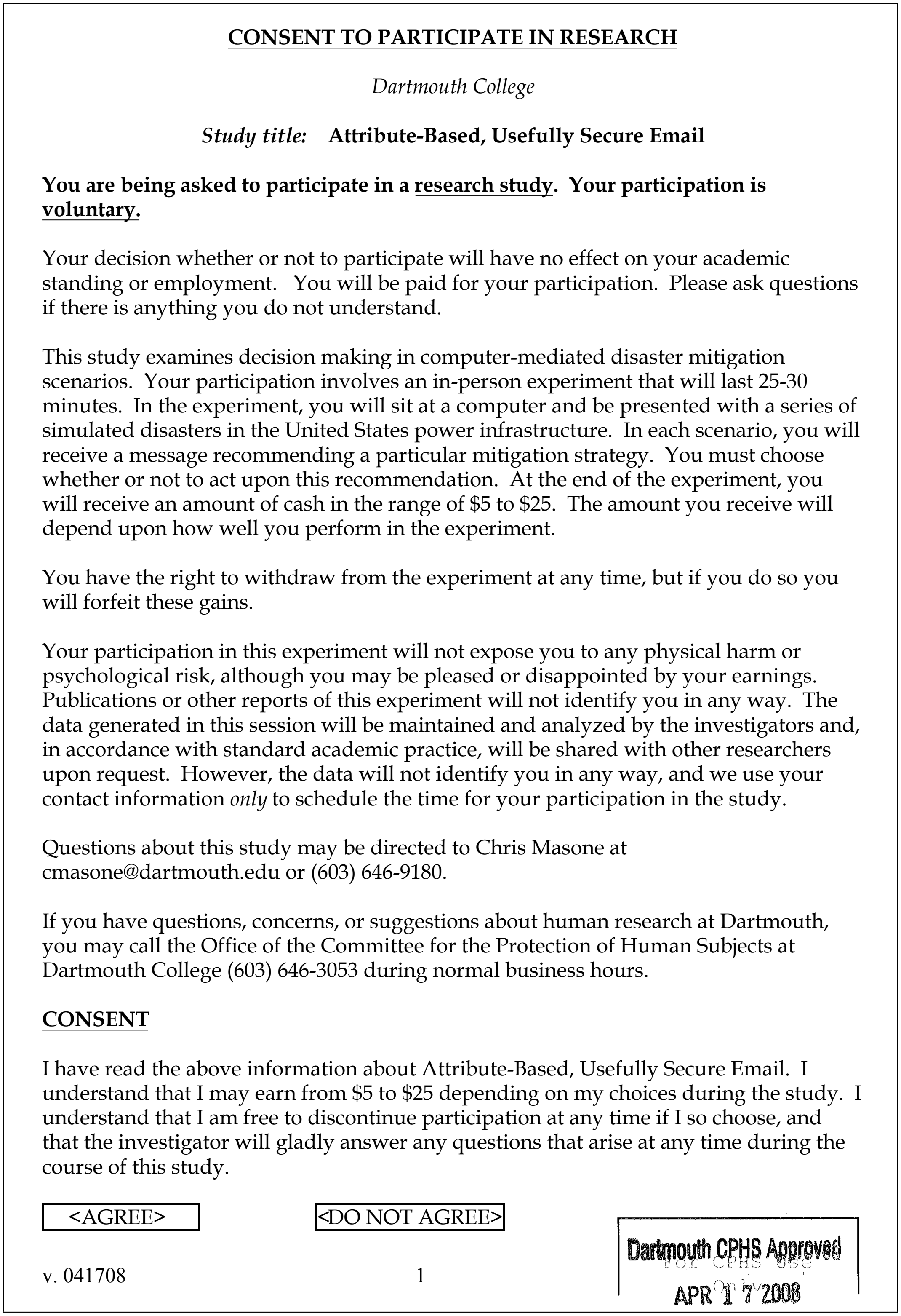
This study [and this blitz] have been approved by the Committee for the Protection of Human Subjects (CPHS). This study is sponsored by Professor Sean Smith. For more information about this study blitz cmasone@dartmouth.edu**.**

If you have general questions about being a research participant, you may call or blitz the Office of the Committee for the Protection of Human Subjects at Dartmouth College (603) 646-3053.

posted [date]

Figure A.1: Recruitment flyer. An email with the same content was also used for recruitment.

A.2 Grid study consent form



CONSENT TO PARTICIPATE IN RESEARCH

Dartmouth College

Study title: **Attribute-Based, Usefully Secure Email**

You are being asked to participate in a research study. Your participation is voluntary.

Your decision whether or not to participate will have no effect on your academic standing or employment. You will be paid for your participation. Please ask questions if there is anything you do not understand.

This study examines decision making in computer-mediated disaster mitigation scenarios. Your participation involves an in-person experiment that will last 25-30 minutes. In the experiment, you will sit at a computer and be presented with a series of simulated disasters in the United States power infrastructure. In each scenario, you will receive a message recommending a particular mitigation strategy. You must choose whether or not to act upon this recommendation. At the end of the experiment, you will receive an amount of cash in the range of $5 to $25. The amount you receive will depend upon how well you perform in the experiment.

You have the right to withdraw from the experiment at any time, but if you do so you will forfeit these gains.

Your participation in this experiment will not expose you to any physical harm or psychological risk, although you may be pleased or disappointed by your earnings. Publications or other reports of this experiment will not identify you in any way. The data generated in this session will be maintained and analyzed by the investigators and, in accordance with standard academic practice, will be shared with other researchers upon request. However, the data will not identify you in any way, and we use your contact information *only* to schedule the time for your participation in the study.

Questions about this study may be directed to Chris Masone at cmasone@dartmouth.edu or (603) 646-9180.

If you have questions, concerns, or suggestions about human research at Dartmouth, you may call the Office of the Committee for the Protection of Human Subjects at Dartmouth College (603) 646-3053 during normal business hours.

CONSENT

I have read the above information about Attribute-Based, Usefully Secure Email. I understand that I may earn from $5 to $25 depending on my choices during the study. I understand that I am free to discontinue participation at any time if I so choose, and that the investigator will gladly answer any questions that arise at any time during the course of this study.

<AGREE> <DO NOT AGREE>

Dartmouth CPHS Approved
For CPHS Use Only
APR 1 7 2008

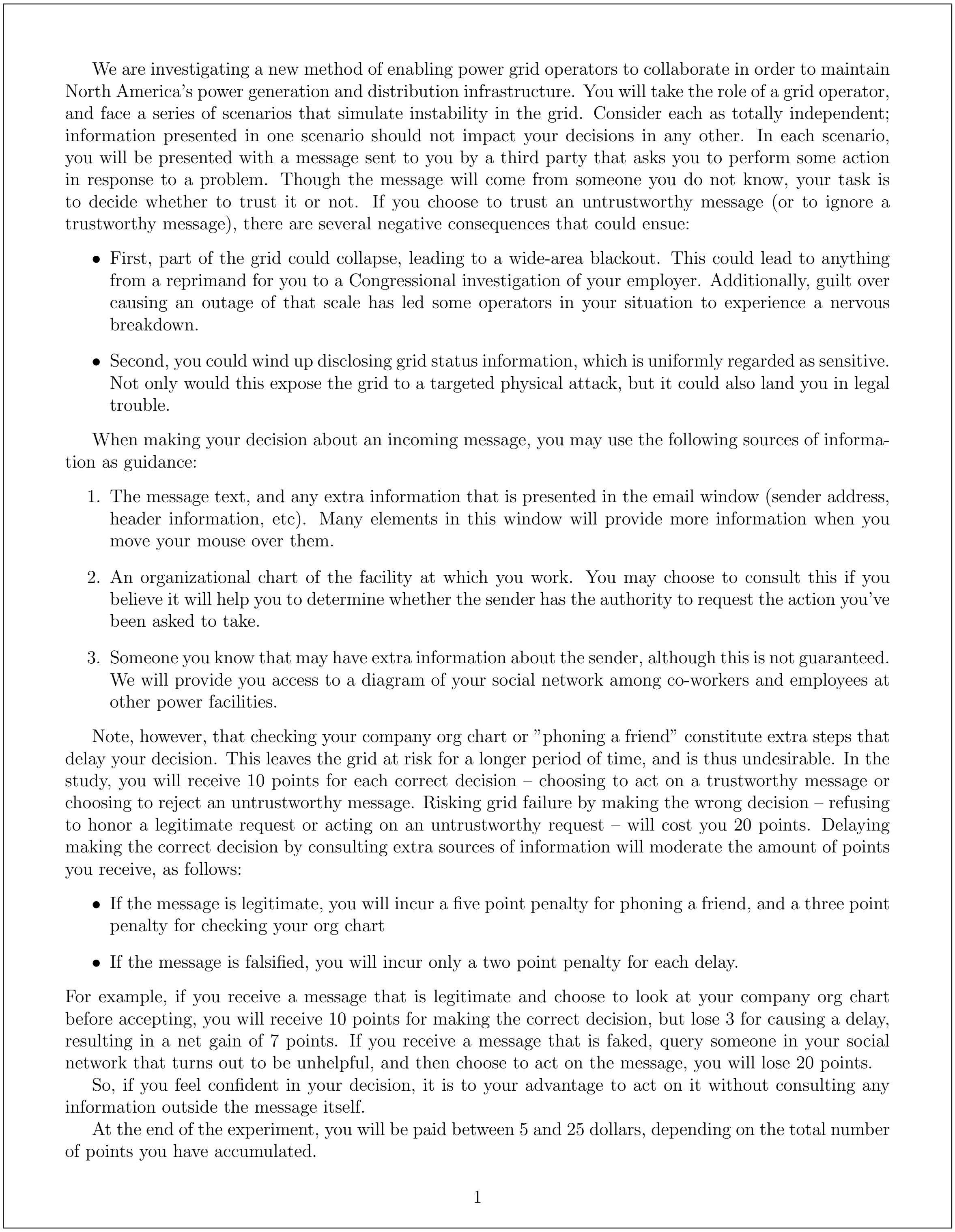
v. 041708 1

Figure A.2: Consent form

A.3 Grid study instructions

The subjects in the study were given a four page handout (Figure A.4) summarizing information on the task they were to complete, background information about the power grid, and an explanation of the study's incentive structure. They were also each read a set of verbal instructions, fleshing out the summarized information they received in the handout. The verbal instructions are provided in Figure A.3.

A.3.1 Verbal instructions



We are investigating a new method of enabling power grid operators to collaborate in order to maintain North America's power generation and distribution infrastructure. You will take the role of a grid operator, and face a series of scenarios that simulate instability in the grid. Consider each as totally independent; information presented in one scenario should not impact your decisions in any other. In each scenario, you will be presented with a message sent to you by a third party that asks you to perform some action in response to a problem. Though the message will come from someone you do not know, your task is to decide whether to trust it or not. If you choose to trust an untrustworthy message (or to ignore a trustworthy message), there are several negative consequences that could ensue:

- First, part of the grid could collapse, leading to a wide-area blackout. This could lead to anything from a reprimand for you to a Congressional investigation of your employer. Additionally, guilt over causing an outage of that scale has led some operators in your situation to experience a nervous breakdown.
- Second, you could wind up disclosing grid status information, which is uniformly regarded as sensitive. Not only would this expose the grid to a targeted physical attack, but it could also land you in legal trouble.

When making your decision about an incoming message, you may use the following sources of information as guidance:

1. The message text, and any extra information that is presented in the email window (sender address, header information, etc). Many elements in this window will provide more information when you move your mouse over them.
2. An organizational chart of the facility at which you work. You may choose to consult this if you believe it will help you to determine whether the sender has the authority to request the action you've been asked to take.
3. Someone you know that may have extra information about the sender, although this is not guaranteed. We will provide you access to a diagram of your social network among co-workers and employees at other power facilities.

Note, however, that checking your company org chart or ”phoning a friend” constitute extra steps that delay your decision. This leaves the grid at risk for a longer period of time, and is thus undesirable. In the study, you will receive 10 points for each correct decision – choosing to act on a trustworthy message or choosing to reject an untrustworthy message. Risking grid failure by making the wrong decision – refusing to honor a legitimate request or acting on an untrustworthy request – will cost you 20 points. Delaying making the correct decision by consulting extra sources of information will moderate the amount of points you receive, as follows:

- If the message is legitimate, you will incur a five point penalty for phoning a friend, and a three point penalty for checking your org chart
- If the message is falsified, you will incur only a two point penalty for each delay.

For example, if you receive a message that is legitimate and choose to look at your company org chart before accepting, you will receive 10 points for making the correct decision, but lose 3 for causing a delay, resulting in a net gain of 7 points. If you receive a message that is faked, query someone in your social network that turns out to be unhelpful, and then choose to act on the message, you will lose 20 points.

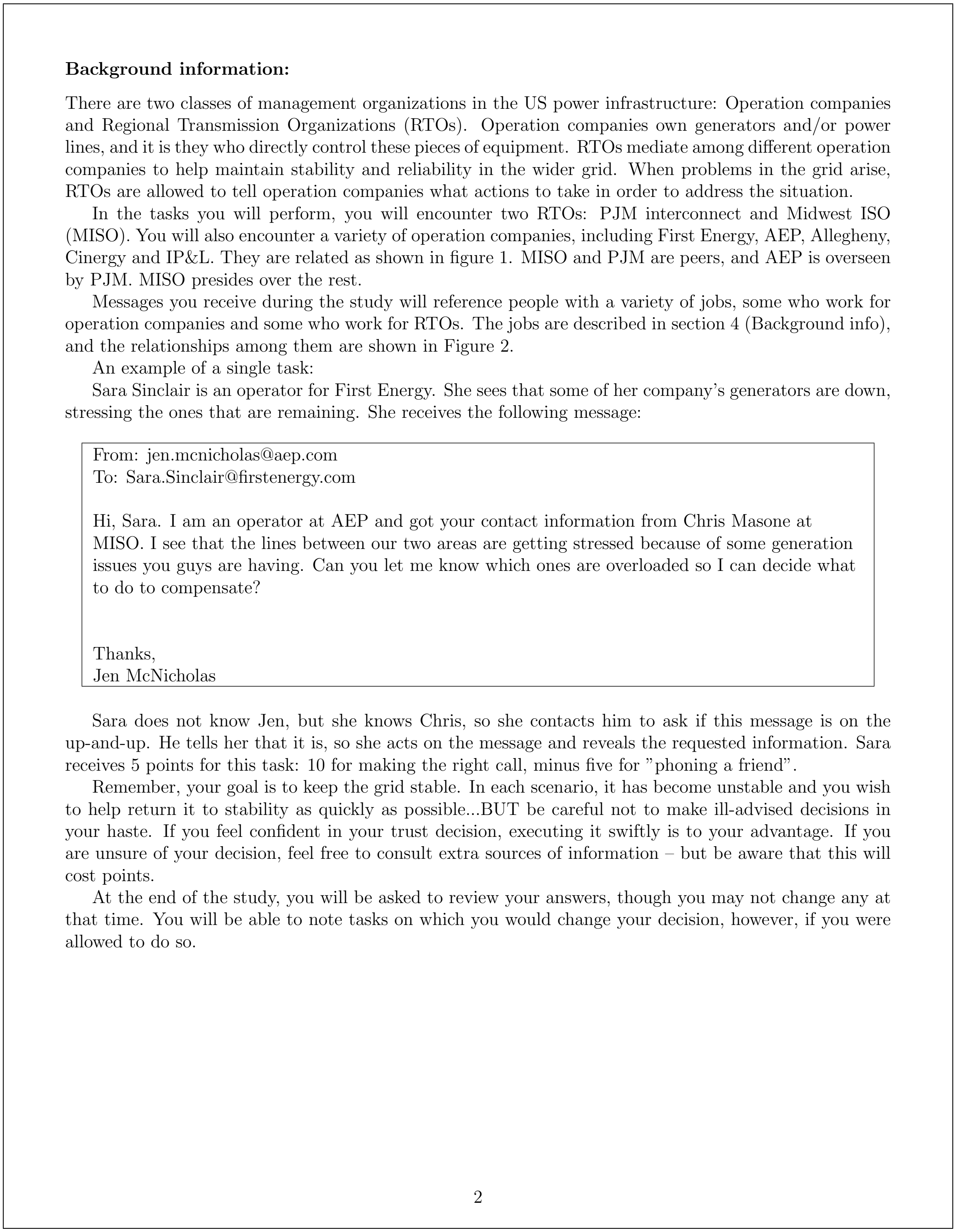
So, if you feel confident in your decision, it is to your advantage to act on it without consulting any information outside the message itself.

At the end of the experiment, you will be paid between 5 and 25 dollars, depending on the total number of points you have accumulated.

1

(a)

Figure A.3: Verbal instructions, page 1



Background information:

There are two classes of management organizations in the US power infrastructure: Operation companies and Regional Transmission Organizations (RTOs). Operation companies own generators and/or power lines, and it is they who directly control these pieces of equipment. RTOs mediate among different operation companies to help maintain stability and reliability in the wider grid. When problems in the grid arise, RTOs are allowed to tell operation companies what actions to take in order to address the situation.

In the tasks you will perform, you will encounter two RTOs: PJM interconnect and Midwest ISO (MISO). You will also encounter a variety of operation companies, including First Energy, AEP, Allegheny, Cinergy and IP&L. They are related as shown in figure 1. MISO and PJM are peers, and AEP is overseen by PJM. MISO presides over the rest.

Messages you receive during the study will reference people with a variety of jobs, some who work for operation companies and some who work for RTOs. The jobs are described in section 4 (Background info), and the relationships among them are shown in Figure 2.

An example of a single task:

Sara Sinclair is an operator for First Energy. She sees that some of her company's generators are down, stressing the ones that are remaining. She receives the following message:

> From: jen.mcnicholas@aep.com
> To: Sara.Sinclair@firstenergy.com
>
> Hi, Sara. I am an operator at AEP and got your contact information from Chris Masone at MISO. I see that the lines between our two areas are getting stressed because of some generation issues you guys are having. Can you let me know which ones are overloaded so I can decide what to do to compensate?
>
> Thanks,
> Jen McNicholas

Sara does not know Jen, but she knows Chris, so she contacts him to ask if this message is on the up-and-up. He tells her that it is, so she acts on the message and reveals the requested information. Sara receives 5 points for this task: 10 for making the right call, minus five for ”phoning a friend”.

Remember, your goal is to keep the grid stable. In each scenario, it has become unstable and you wish to help return it to stability as quickly as possible...BUT be careful not to make ill-advised decisions in your haste. If you feel confident in your trust decision, executing it swiftly is to your advantage. If you are unsure of your decision, feel free to consult extra sources of information – but be aware that this will cost points.

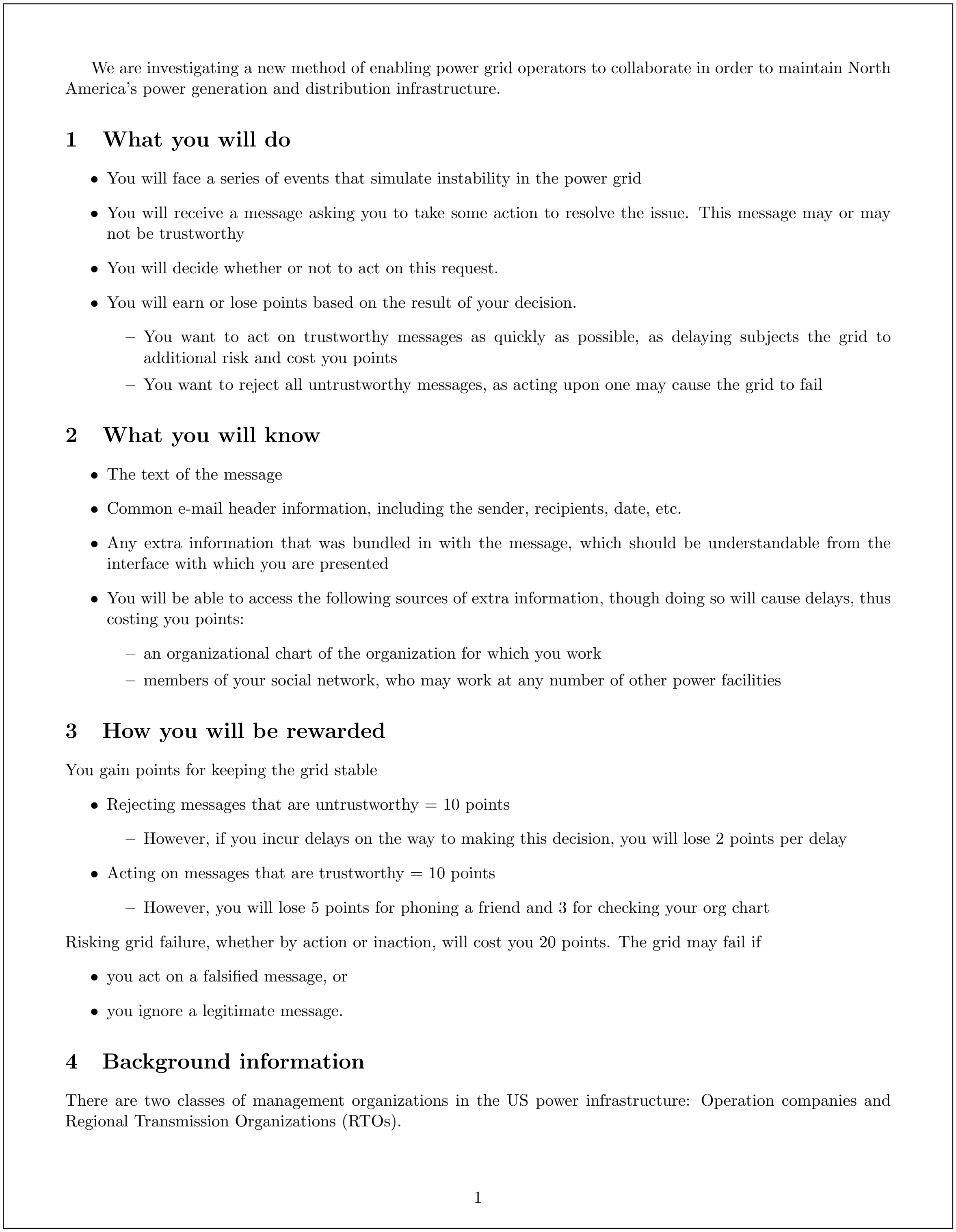
At the end of the study, you will be asked to review your answers, though you may not change any at that time. You will be able to note tasks on which you would change your decision, however, if you were allowed to do so.

2

(b)

Figure A.3: Verbal instructions, page 2

A.3.2 Written instructions



We are investigating a new method of enabling power grid operators to collaborate in order to maintain North America's power generation and distribution infrastructure.

1 What you will do

- You will face a series of events that simulate instability in the power grid
- You will receive a message asking you to take some action to resolve the issue. This message may or may not be trustworthy
- You will decide whether or not to act on this request.
- You will earn or lose points based on the result of your decision.
  - You want to act on trustworthy messages as quickly as possible, as delaying subjects the grid to additional risk and cost you points
  - You want to reject all untrustworthy messages, as acting upon one may cause the grid to fail

2 What you will know

- The text of the message
- Common e-mail header information, including the sender, recipients, date, etc.
- Any extra information that was bundled in with the message, which should be understandable from the interface with which you are presented
- You will be able to access the following sources of extra information, though doing so will cause delays, thus costing you points:
  - an organizational chart of the organization for which you work
  - members of your social network, who may work at any number of other power facilities

3 How you will be rewarded

You gain points for keeping the grid stable

- Rejecting messages that are untrustworthy = 10 points
  - However, if you incur delays on the way to making this decision, you will lose 2 points per delay
- Acting on messages that are trustworthy = 10 points
  - However, you will lose 5 points for phoning a friend and 3 for checking your org chart

Risking grid failure, whether by action or inaction, will cost you 20 points. The grid may fail if

- you act on a falsified message, or
- you ignore a legitimate message.

4 Background information

There are two classes of management organizations in the US power infrastructure: Operation companies and Regional Transmission Organizations (RTOs).

(a)

Figure A.4: Written instructions, page 1

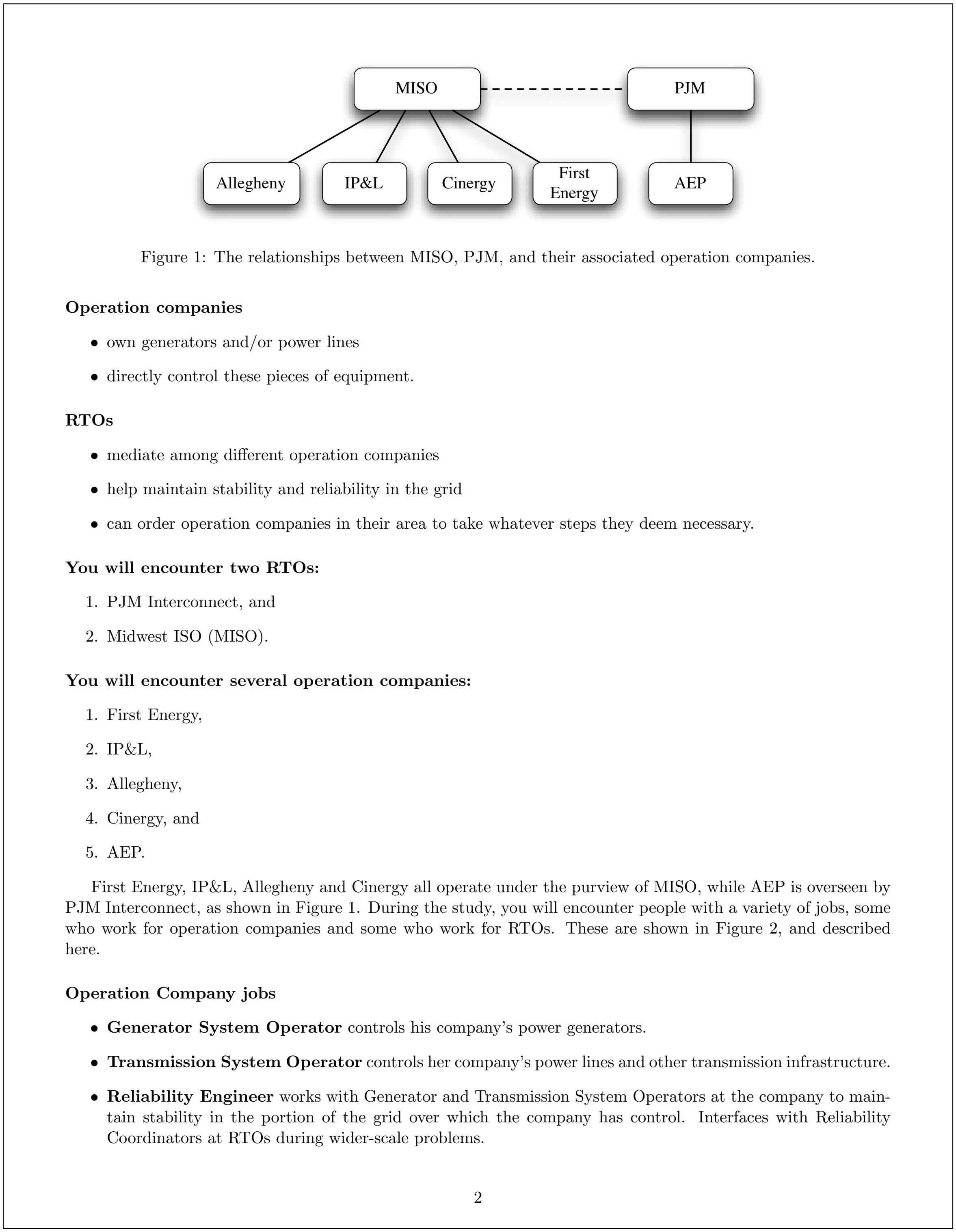

MISO
PJM
Allegheny
IP&L
Cinergy
First Energy
AEP

Figure 1: The relationships between MISO, PJM, and their associated operation companies.

Operation companies

- own generators and/or power lines
- directly control these pieces of equipment.

RTOs

- mediate among different operation companies
- help maintain stability and reliability in the grid
- can order operation companies in their area to take whatever steps they deem necessary.

You will encounter two RTOs:

1. PJM Interconnect, and
2. Midwest ISO (MISO).

You will encounter several operation companies:

1. First Energy,
2. IP&L,
3. Allegheny,
4. Cinergy, and
5. AEP.

First Energy, IP&L, Allegheny and Cinergy all operate under the purview of MISO, while AEP is overseen by PJM Interconnect, as shown in Figure 1. During the study, you will encounter people with a variety of jobs, some who work for operation companies and some who work for RTOs. These are shown in Figure 2, and described here.

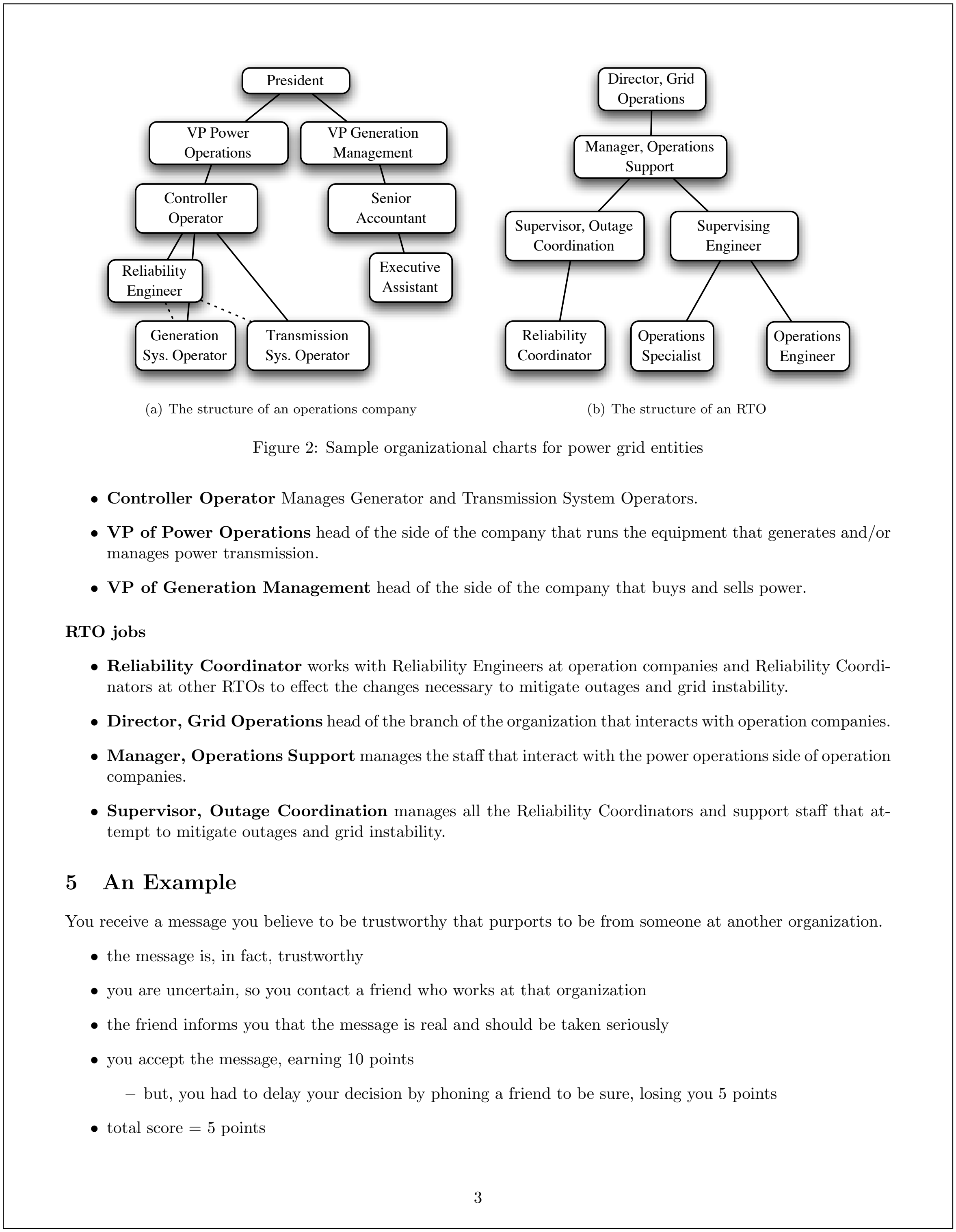
Operation Company jobs

- **Generator System Operator** controls his company's power generators.
- **Transmission System Operator** controls her company's power lines and other transmission infrastructure.
- **Reliability Engineer** works with Generator and Transmission System Operators at the company to maintain stability in the portion of the grid over which the company has control. Interfaces with Reliability Coordinators at RTOs during wider-scale problems.

2

(b)

Figure A.4: Written instructions, page 2



President
VP Power Operations
VP Generation Management
Controller Operator
Senior Accountant
Reliability Engineer
Executive Assistant
Generation Sys. Operator
Transmission Sys. Operator
Director, Grid Operations
Manager, Operations Support
Supervisor, Outage Coordination
Supervising Engineer
Reliability Coordinator
Operations Specialist
Operations Engineer

(a) The structure of an operations company

(b) The structure of an RTO

Figure 2: Sample organizational charts for power grid entities

- **Controller Operator** Manages Generator and Transmission System Operators.
- **VP of Power Operations** head of the side of the company that runs the equipment that generates and/or manages power transmission.
- **VP of Generation Management** head of the side of the company that buys and sells power.

RTO jobs

- **Reliability Coordinator** works with Reliability Engineers at operation companies and Reliability Coordinators at other RTOs to effect the changes necessary to mitigate outages and grid instability.
- **Director, Grid Operations** head of the branch of the organization that interacts with operation companies.
- **Manager, Operations Support** manages the staff that interact with the power operations side of operation companies.
- **Supervisor, Outage Coordination** manages all the Reliability Coordinators and support staff that attempt to mitigate outages and grid instability.

5 An Example

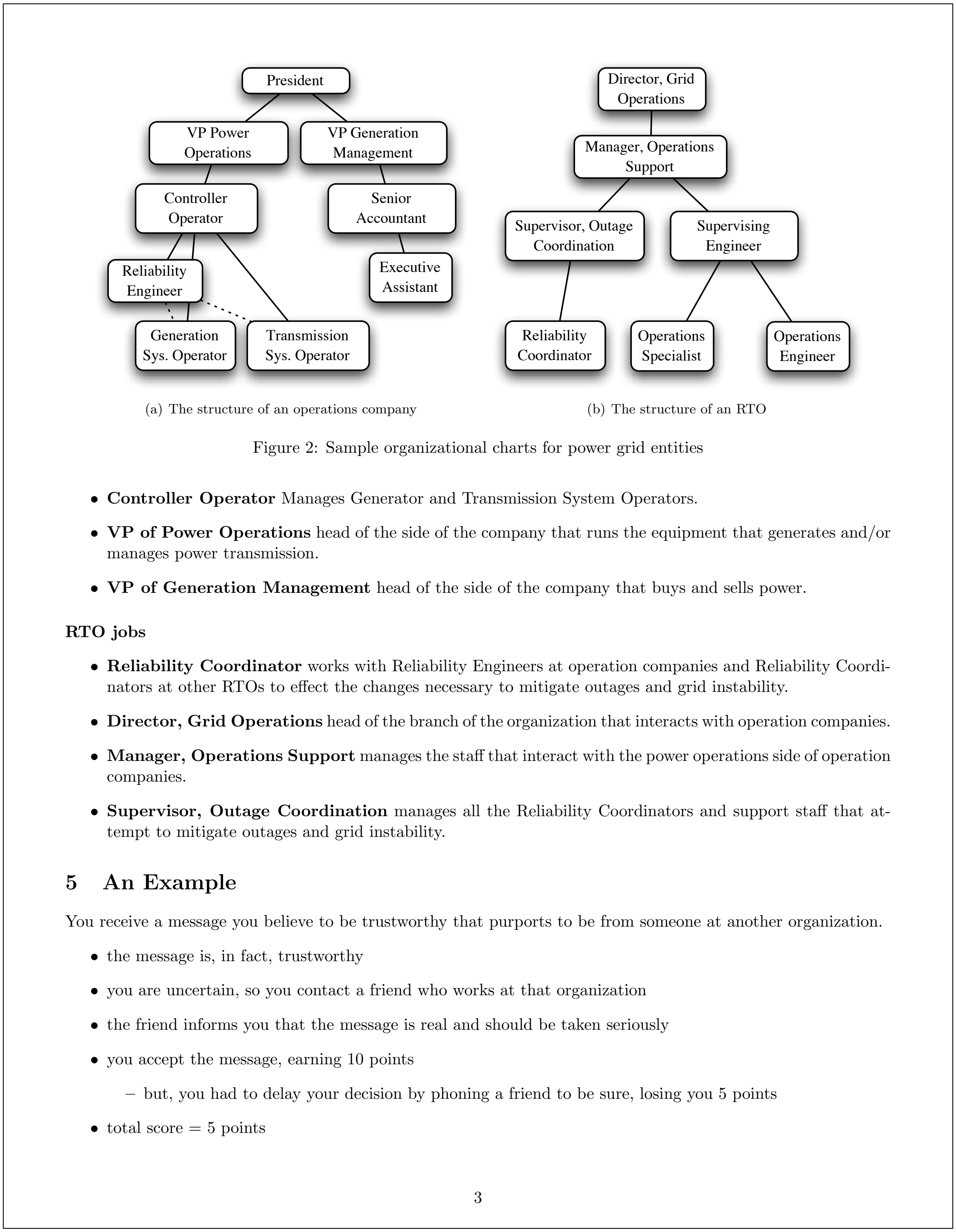
You receive a message you believe to be trustworthy that purports to be from someone at another organization.

- the message is, in fact, trustworthy
- you are uncertain, so you contact a friend who works at that organization
- the friend informs you that the message is real and should be taken seriously
- you accept the message, earning 10 points
  - but, you had to delay your decision by phoning a friend to be sure, losing you 5 points
- total score = 5 points

3

(c)

Figure A.4: Written instructions, page 3


President
VP Power Operations
VP Generation Management
Controller Operator
Senior Accountant
Reliability Engineer
Executive Assistant
Generation Sys. Operator
Transmission Sys. Operator
Director, Grid Operations
Manager, Operations Support
Supervisor, Outage Coordination
Supervising Engineer
Reliability Coordinator
Operations Specialist
Operations Engineer

(a) The structure of an operations company

(b) The structure of an RTO

Figure 2: Sample organizational charts for power grid entities

- **Controller Operator** Manages Generator and Transmission System Operators.
- **VP of Power Operations** head of the side of the company that runs the equipment that generates and/or manages power transmission.
- **VP of Generation Management** head of the side of the company that buys and sells power.

RTO jobs

- **Reliability Coordinator** works with Reliability Engineers at operation companies and Reliability Coordinators at other RTOs to effect the changes necessary to mitigate outages and grid instability.
- **Director, Grid Operations** head of the branch of the organization that interacts with operation companies.
- **Manager, Operations Support** manages the staff that interact with the power operations side of operation companies.
- **Supervisor, Outage Coordination** manages all the Reliability Coordinators and support staff that attempt to mitigate outages and grid instability.

5 An Example

You receive a message you believe to be trustworthy that purports to be from someone at another organization.

- the message is, in fact, trustworthy
- you are uncertain, so you contact a friend who works at that organization
- the friend informs you that the message is real and should be taken seriously
- you accept the message, earning 10 points
  - but, you had to delay your decision by phoning a friend to be sure, losing you 5 points
- total score = 5 points

(d)

Figure A.4: Written instructions, page 4

A.4 Grid study scenario details

A.4.1 The external-attacker scenario

The attacker, who sits completely outside the grid's trust infrastructure, requests an action that will further hamper the grid. The subject should reject the message.

Characteristics of the subject

Subject affiliation:	First Energy
Subject role:	Reliability engineer

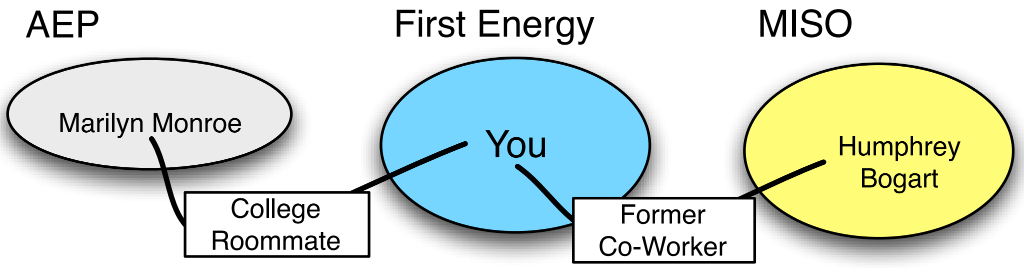
Characteristics of the attacker

Attacker:	Alice Apples
Attacker email address:	`<alicea_personal@yahoo.com>`
Attacker affiliation:	None (claims AEP)
Signature CA:	Thawte Freemail

Parameters of the situation

Trust-flow:	Faked coopetition
Contingency:	There are four power lines tying First Energy to AEP. Two are currently down, and the other two are overloaded but still functional for now.

Pertinent social connections:



Message

Signature:	Valid, external CA
Attributes:	None
Text:	I'm having a problem with my work email, so please forgive my use of my home email account! I am a transmission systems operator at AEP and got your contact information from Humphrey Bogart at MISO. I wanted to let you know that we're on the verge of getting some more of those tie lines back up, to alleviate the overload on the ones that're there right now. So, feel free to start bringing customers back on line and adding load.

Out-of-band information

Phonable friend:	Marilyn at AEP
Information:	Marilyn is at home sick and has no useful info.
Org chart information:	None applicable

How the subject should decide

Although there is no useful information available through out-of-band channels, the use of a personal email address should be a red flag. Subjects in the S/MIME and ABUSE groups would have had extra cues. The CA on the certificate used to sign the message, for example, was not associated with the sender's claimed company. Also, the message had no attributes bound to it. For subjects who had seen other messages with attributes that they found useful, this was a sign that the something was amiss.

A.4.2 The external-insider scenario

The attacker, who is within the grid's trust infrastructure (though at different company than the subject) requests an action that will further hamper the grid. The subject should reject the message.

Characteristics of the subject

Subject affiliation:	IP&L
Subject role:	Transmission system operator

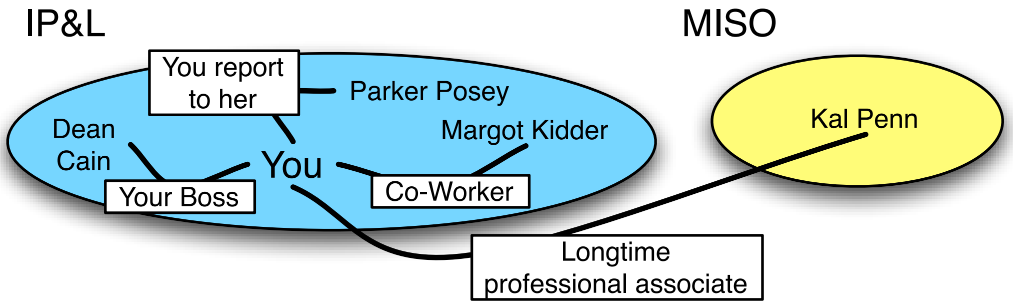
Characteristics of the attacker

Attacker:	Bob Bison
Attacker email address:	`<bbison@miso.ORG>`
Attacker affiliation:	MISO
Signature CA:	Midwest ISO CA

Parameters of the situation

Trust-flow:	Faked role-based
Contingency:	Power lines from Thompson to Fredericksburg and Thompson to Shoreham are overloaded. Lines to Peterson and Copeland are fine.

Pertinent social connections:

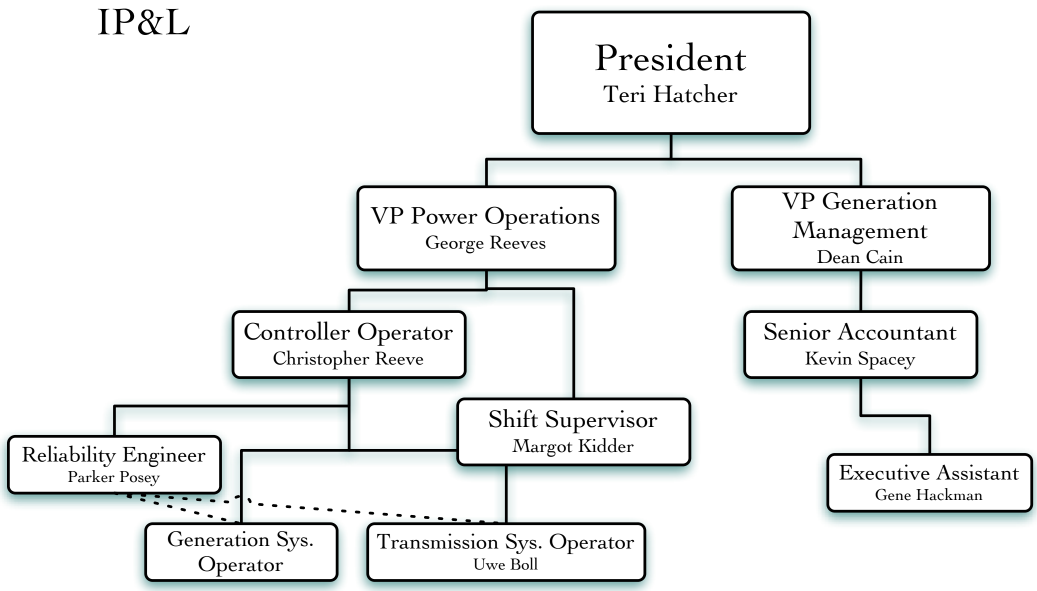


Message

Signature:	Valid, sender's company CA
Attributes:	MISO says that Bob Bison is a full-time employee
Text:	This is Bob Bison at MISO. I got your contact information from Parker Posey over there at IP and L. We're not currently seeing any grid status updates on our console over here for some reason...can you let us know the status of all the lines coming out of Thompson?

Out-of-band information

Phonable friend:	Kal at MISO
Information:	Bob has no reason to ask for that information.
Org chart information:	Parker Posey outranks the subject.
Org chart:	



How the subject should decide

The S/MIME and plaintext groups should have needed to ask their associate Kal at MISO for advice. He would have informed them that Bob should be ignored. The ABUSE group should have seen that mere employee status is not enough to give Bob the authority to ask for the information he requests.

A.4.3 The internal-insider scenario

The attacker, who is within the subject's company's trust infrastructure requests an action that will further hamper the grid. The subject should reject the message.

Characteristics of the subject

Subject affiliation:	Cinergy
Subject role:	Generation system operator

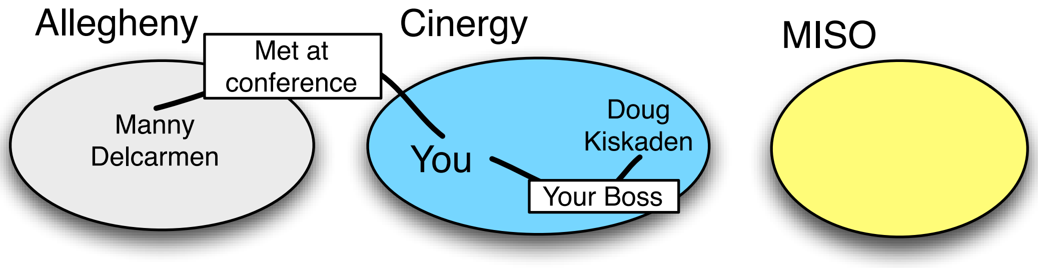
Characteristics of the attacker

Attacker:	Carol Crawford
Attacker email address:	`<carolc@cinergy.com>`
Attacker affiliation:	Cinergy
Signature CA:	Cinergy CA

Parameters of the situation

Trust-flow:	Faked delegation
Contingency:	Allegheny is generating too much power at Conger, which is overloading the power lines coming into Regina in the western part of Cinergy's area. You have generators at Regina, and also at Northumberland to the east of Regina.

Pertinent social connections:

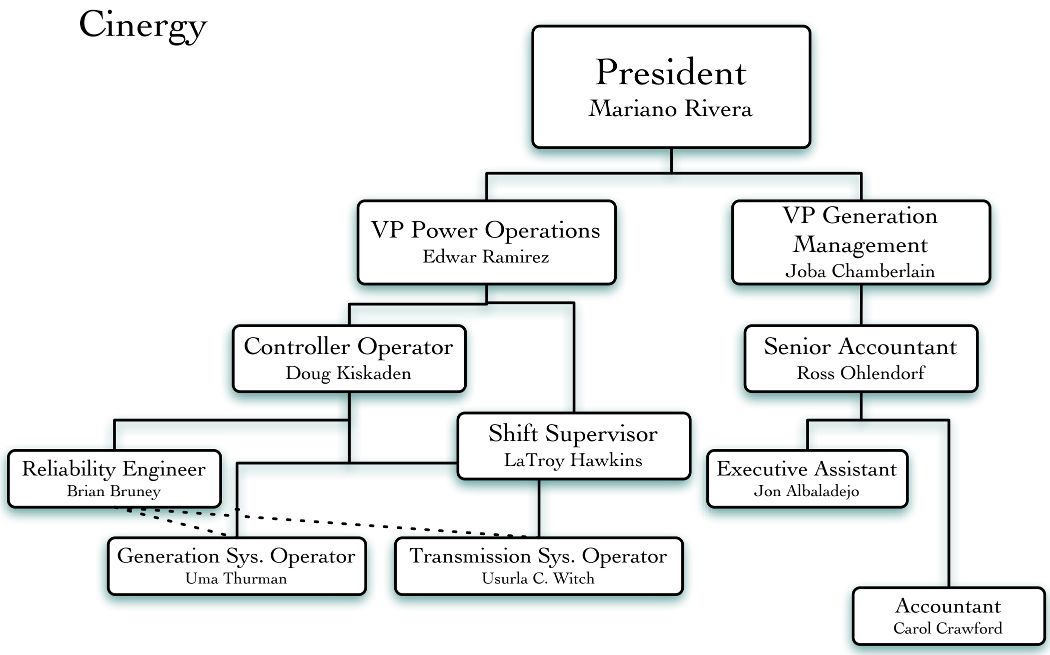


Message

Signature:	Valid, sender and subject's company CA
Attributes:	Cinergy says that Carol Crawford is a full-time employee
Text:	I know you're seeing some grid instability on the border with Allegheny. Doug Kiskaden and I have been talking to a reliability coordinator at MISO, who's been in touch with them. They've gotten Allegheny to bring down their generation at Conger to address the instability you're seeing, but they'll need to take a whole unit offline to do it. That'll drop generation too far, so they told me to ask you to bring up generation at Regina by 80 MW to provide a cushion for that drop off.

Out-of-band information

Phonable friend:	Manny at Allegheny
Information:	Manny is on travel; has no idea what's going on at work.
Org chart information:	Doug Kiskaden is a controller operator. The subject is subordinate to him. Carol (the attacker) is an accountant.
Org chart:	



How the subject should decide

The S/MIME and plaintext groups should have needed to check their org chart to see that Carol is an accountant; she would never be involved in the kind of conversations to which she claims to have been privy. The ABUSE group should have seen that mere employee status is not enough to give Carol the kind of access or authority she claims.

A.4.4 Legitimate coopetition

An employee at AEP leverages a legitimate coopetition trust flow that comes from MISO to ask the subject, at First Energy, to make an operational change that will improve grid stability.

Characteristics of the subject

Subject affiliation:	First Energy
Subject role:	Reliability engineer

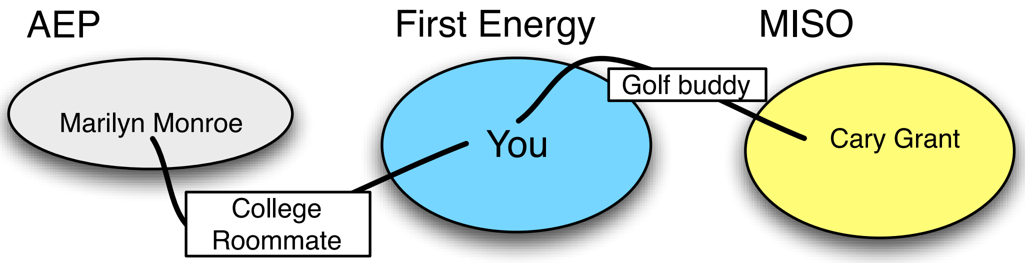
Characteristics of the sender

Attacker:	Darren Driscoll
Attacker email address:	`<driscoll@aep.com>`
Attacker affiliation:	AEP
Signature CA:	American Electric Power CA

Parameters of the situation

Trust-flow:	Legitimate coopetition
Contingency:	There are four power lines tying First Energy to AEP. Two are currently down, and the other two are overloaded but still functional for now.

Pertinent social connections:



Message

Signature:	Valid, sender’s company CA
Attributes:	AEP says that Darren Driscoll is a full-time employee

AEP says that	Roy Halladay	is the President
Roy Hallady says that	Jeremy Accardo	is a VP of Power Operations
Jeremy Accardo says that	Marilyn Monroe	is a Controller Operator
Marilyn Monroe says that	Darren Driscoll	is a Reliability Engineer

MISO says that	Fran Fine	is the Director, Grid Ops.
Fran Fine says that	George Gosling	is a Manager, Ops. Support
George Gosling says that	Henry Hank	is a Supervisor, Outage Coordination
Henry Hank says that	Cary Grant	is a Reliability Coordinator
Cary Grant says that	Irene Islena @ PJM	is mitigating the AEP-FE tie line overload
Irene Islena @ PJM says that	Darren Driscoll	is allowed to ask FE to buy generation

Text:	I am a transmission systems operator at AEP and got your contact information from Cary Grant at MISO. We’re worried about the few remaining lines between you guys and us. If you keep adding load the way you are, we’re concerned that the transmission lines will overload and quit on us. Cary and I have agreed that you guys need to purchase some generation as soon as possible, unless you can get some generation up in the immediate future.

Out-of-band information

Phonable friend:	Cary at MISO
Information:	Darren’s request should be heeded.
Org chart information:	None applicable

How the subject should decide

Subjects in the S/MIME and plaintext groups must contact Cary to verify that the request is acceptable. The ABUSE subjects should be able to learn from the attributes that Darren both works at AEP in a position that deals with maintaining grid stability *and* that he has been given permission to ask for a remediation action from a MISO reliability coordinator.

A.4.5 Legitimate Delegation

An employee at MISO leverages a legitimate delegation trust flow that comes from Cinergy to ask the subject, also at Cinergy, to make an operational change that will improve grid stability.

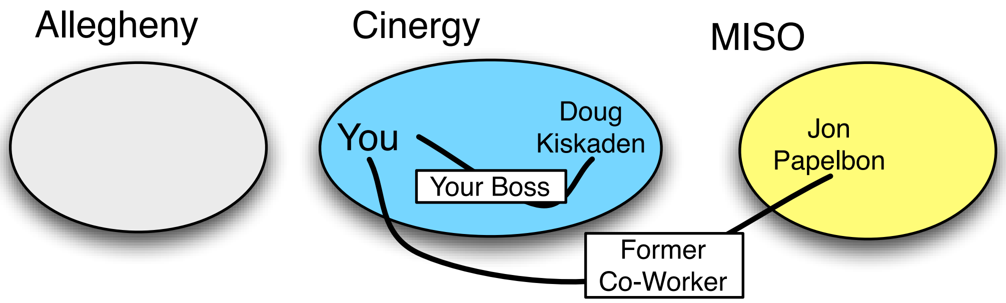
Characteristics of the subject

Subject affiliation:	Cinergy
Subject role:	Generation system operator

Characteristics of the sender

Attacker:	Ernie Ells
Attacker email address:	`<ells@miso.org>`
Attacker affiliation:	MISO
Signature CA:	Midwest ISO CA

Parameters of the situation

Trust-flow:	Legitimate delegation
Contingency:	Allegheny is generating too much power at Wheatland, which is overloading the power lines coming into Gibson in the western part of Cinergy's area. You have generators at Gibson, and also at Woodsdale to the east of Gibson.
Pertinent social connections:  Allegheny Cinergy MISO You Doug Kiskaden Your Boss Former Co-Worker Jon Papelbon	

Message

Signature: Valid, sender's company CA

Attributes:

MISO says that	Ernie Ells	is a full-time employee

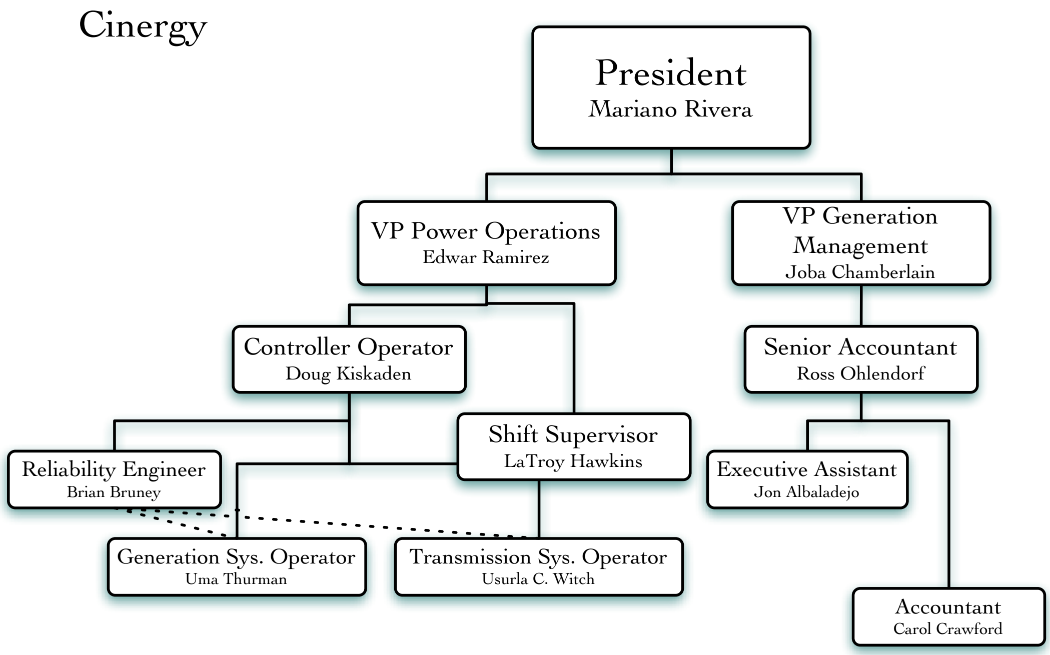
MISO says that	Fran Fine	is the Director, Grid Ops.
Fran Fine says that	George Gosling	is a Manager, Ops. Support
George Gosling says that	Henry Hank	is a Supervisor, Outage Coordination
Henry Hank says that	Ernie Ells	is a Reliability Coordinator

Cinergy says that	Mariano Rivera	is the President
Mariano Rivera says that	Edwar Ramirez	is a VP of Power Operations
Edwar Ramirez says that	Doug Kiskaden	is a Controller Operator
Doug Kiskaden says that	Ernie Ells	is allowed to ask for 80MW drop at Gibson, 80MW boost at Woodsdale

Text: I'm a reliability coordinator at MISO, and I've been talking to Doug Kiskaden, the Controller Operator over there at Cinergy. We're seeing some instability due to generation coming from Allegheny, and have agreed that we need you to bring down generation at Gibson by 80 MW and up at Woodsdale by the same amount.

Out-of-band information

Phonable friend:	Jon at MISO
Information:	Ernie has been talking to Allegheny; he knows what to do.
Org chart information:	Doug Kiskaden is a controller operator.
	The subject is subordinate to him.
Org chart:	



How the subject should decide

Subjects in the S/MIME and plaintext groups must contact Jon to verify that the request is acceptable. The ABUSE subjects should be able to learn from the attributes that Ernie is a reliability coordinator at MISO, and thus it makes sense that he'd be talking to Allegheny. Also, the attributes show that Doug Kiskaden has given Ernie permission to ask for the generation changes he requests.

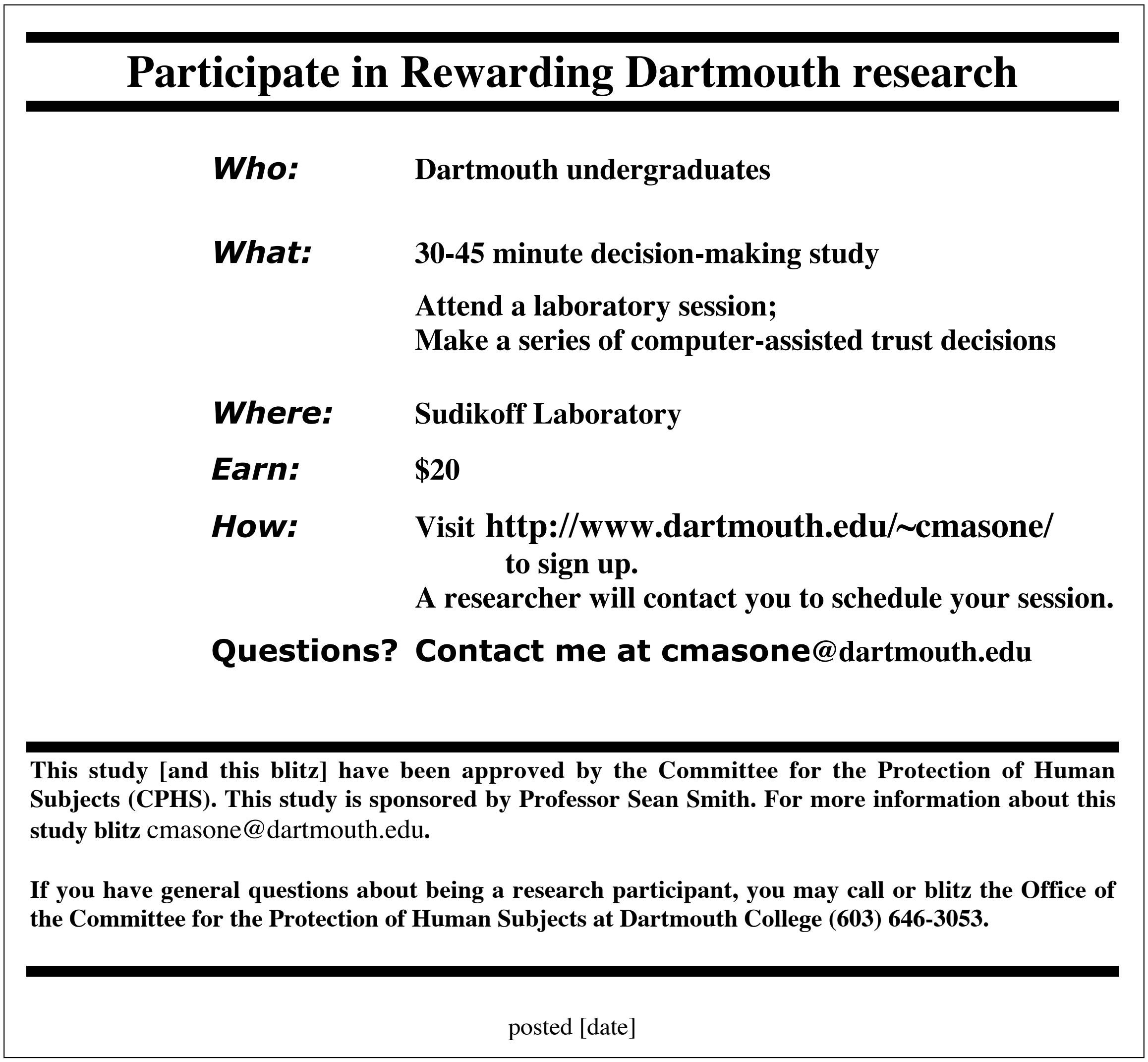
Appendix B

Abusing Johnny appendix

In this appendix, we provide the materials viewed by the subjects during Abusing Johnny.

B.1 Abusing Johnny recruitment

In addition to posting the flyer shown in Figure B.1, we sent an email with the same information out to a number of campus email lists, with an encouragement to forward it to anyone in the Dartmouth community who might be interested in participating.



Participate in Rewarding Dartmouth research

Who: **Dartmouth undergraduates**

What: **30-45 minute decision-making study**

Attend a laboratory session;
Make a series of computer-assisted trust decisions

Where: **Sudikoff Laboratory**

Earn: **$20**

How: **Visit http://www.dartmouth.edu/~cmasone/**
to sign up.
A researcher will contact you to schedule your session.

Questions? Contact me at cmasone@dartmouth.edu

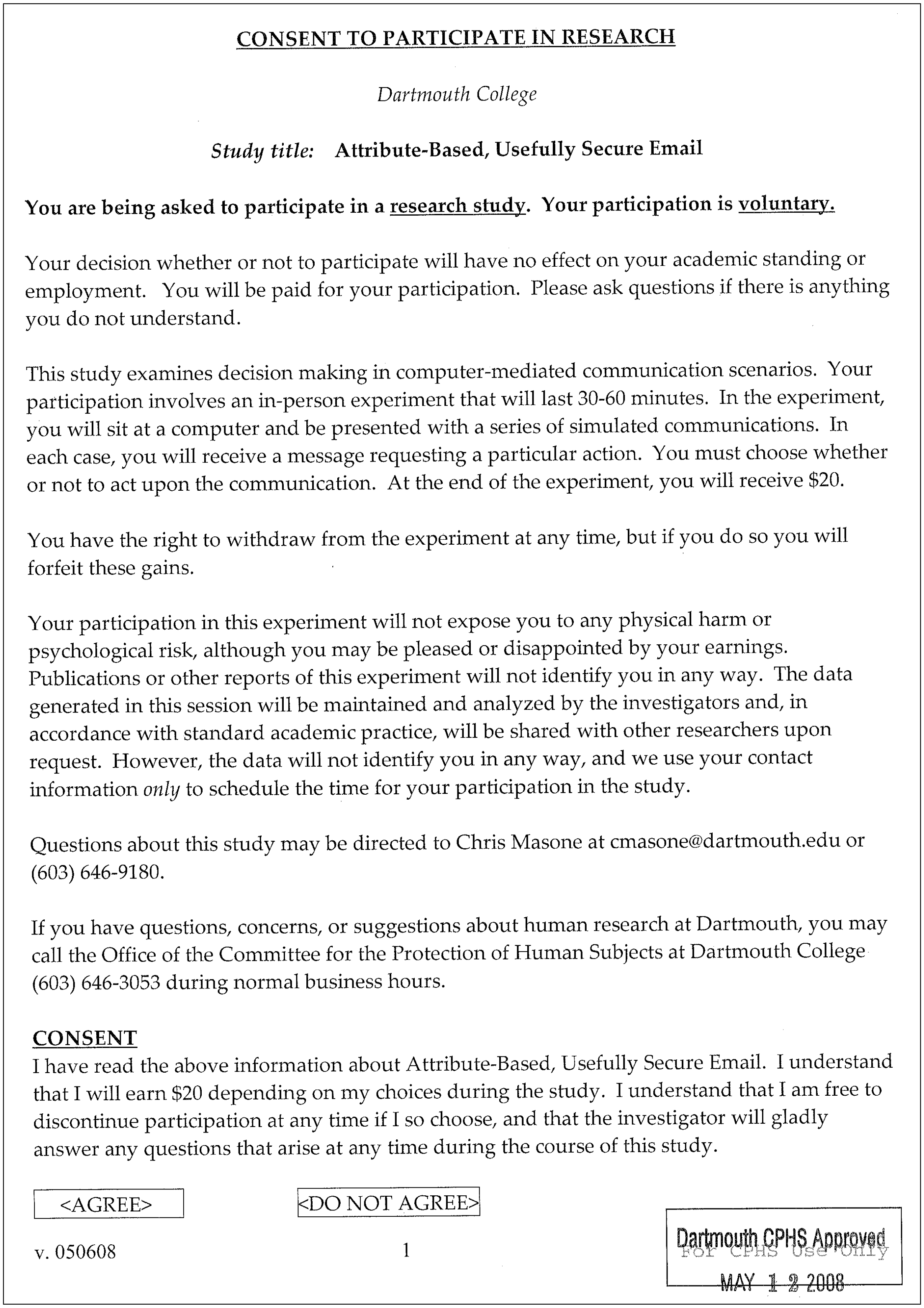
This study [and this blitz] have been approved by the Committee for the Protection of Human Subjects (CPHS). This study is sponsored by Professor Sean Smith. For more information about this study blitz cmasone@dartmouth.edu**.**

If you have general questions about being a research participant, you may call or blitz the Office of the Committee for the Protection of Human Subjects at Dartmouth College (603) 646-3053.

posted [date]

Figure B.1: Recruitment flyer. An email with the same content was also used for recruitment.

B.2 Abusing Johnny consent form



CONSENT TO PARTICIPATE IN RESEARCH

Dartmouth College

Study title: **Attribute-Based, Usefully Secure Email**

You are being asked to participate in a research study. Your participation is voluntary.

Your decision whether or not to participate will have no effect on your academic standing or employment. You will be paid for your participation. Please ask questions if there is anything you do not understand.

This study examines decision making in computer-mediated communication scenarios. Your participation involves an in-person experiment that will last 30-60 minutes. In the experiment, you will sit at a computer and be presented with a series of simulated communications. In each case, you will receive a message requesting a particular action. You must choose whether or not to act upon the communication. At the end of the experiment, you will receive $20.

You have the right to withdraw from the experiment at any time, but if you do so you will forfeit these gains.

Your participation in this experiment will not expose you to any physical harm or psychological risk, although you may be pleased or disappointed by your earnings. Publications or other reports of this experiment will not identify you in any way. The data generated in this session will be maintained and analyzed by the investigators and, in accordance with standard academic practice, will be shared with other researchers upon request. However, the data will not identify you in any way, and we use your contact information *only* to schedule the time for your participation in the study.

Questions about this study may be directed to Chris Masone at cmasone@dartmouth.edu or (603) 646-9180.

If you have questions, concerns, or suggestions about human research at Dartmouth, you may call the Office of the Committee for the Protection of Human Subjects at Dartmouth College (603) 646-3053 during normal business hours.

CONSENT
I have read the above information about Attribute-Based, Usefully Secure Email. I understand that I will earn $20 depending on my choices during the study. I understand that I am free to discontinue participation at any time if I so choose, and that the investigator will gladly answer any questions that arise at any time during the course of this study.

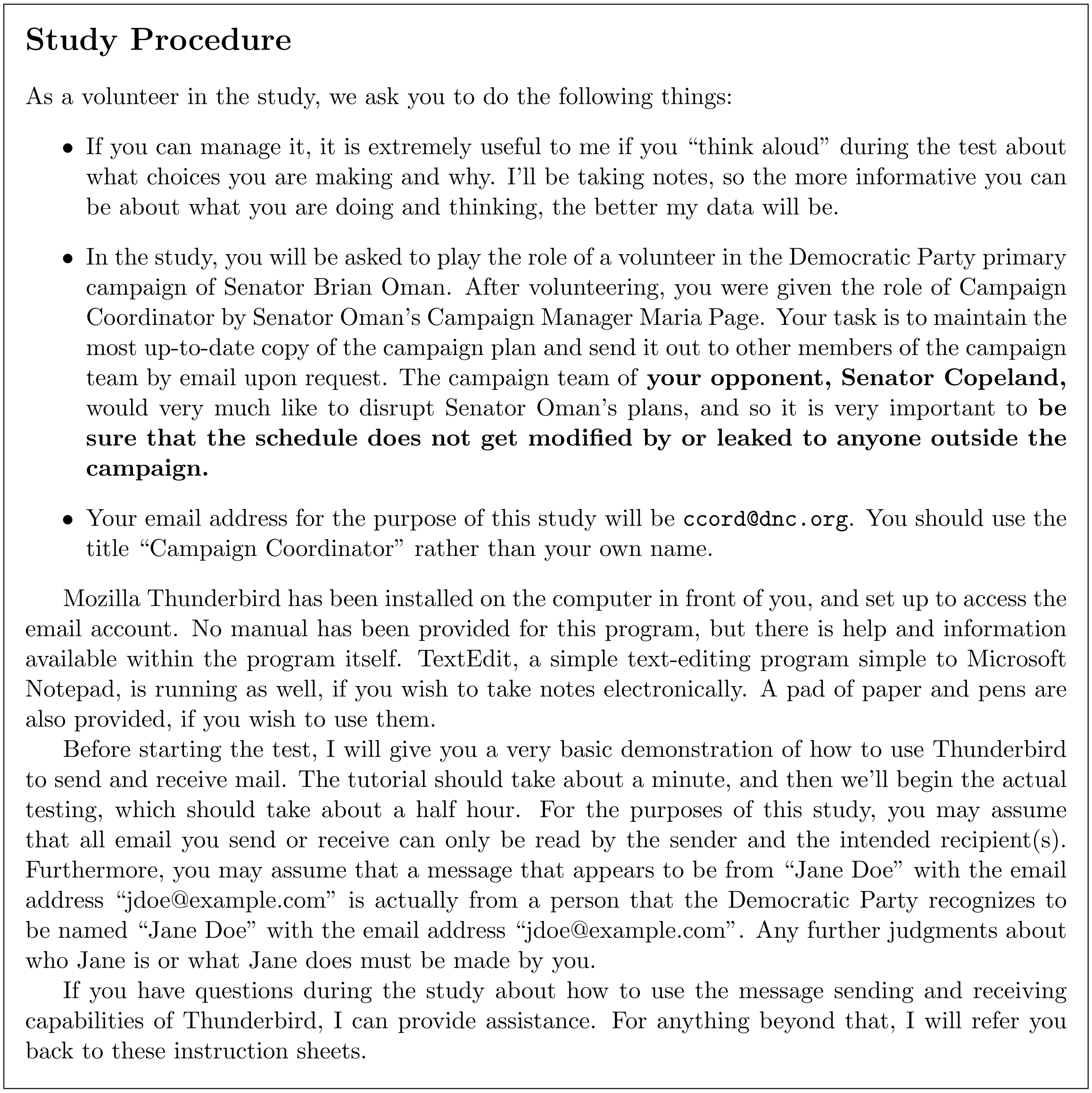
<AGREE> <DO NOT AGREE>

v. 050608 1

Dartmouth CPHS Approved
For CPHS Use Only
MAY 1 2 2008

Figure B.2: Consent form

B.3 Abusing Johnny setup information



Study Procedure

As a volunteer in the study, we ask you to do the following things:

- If you can manage it, it is extremely useful to me if you “think aloud” during the test about what choices you are making and why. I’ll be taking notes, so the more informative you can be about what you are doing and thinking, the better my data will be.
- In the study, you will be asked to play the role of a volunteer in the Democratic Party primary campaign of Senator Brian Oman. After volunteering, you were given the role of Campaign Coordinator by Senator Oman’s Campaign Manager Maria Page. Your task is to maintain the most up-to-date copy of the campaign plan and send it out to other members of the campaign team by email upon request. The campaign team of **your opponent, Senator Copeland,** would very much like to disrupt Senator Oman’s plans, and so it is very important to **be sure that the schedule does not get modified by or leaked to anyone outside the campaign.**
- Your email address for the purpose of this study will be `ccord@dnc.org`. You should use the title “Campaign Coordinator” rather than your own name.

Mozilla Thunderbird has been installed on the computer in front of you, and set up to access the email account. No manual has been provided for this program, but there is help and information available within the program itself. TextEdit, a simple text-editing program simple to Microsoft Notepad, is running as well, if you wish to take notes electronically. A pad of paper and pens are also provided, if you wish to use them.

Before starting the test, I will give you a very basic demonstration of how to use Thunderbird to send and receive mail. The tutorial should take about a minute, and then we’ll begin the actual testing, which should take about a half hour. For the purposes of this study, you may assume that all email you send or receive can only be read by the sender and the intended recipient(s). Furthermore, you may assume that a message that appears to be from “Jane Doe” with the email address “jdoe@example.com” is actually from a person that the Democratic Party recognizes to be named “Jane Doe” with the email address “jdoe@example.com”. Any further judgments about who Jane is or what Jane does must be made by you.

If you have questions during the study about how to use the message sending and receiving capabilities of Thunderbird, I can provide assistance. For anything beyond that, I will refer you back to these instruction sheets.

Figure B.3: Setup information for Abusing Johnny.

B.4 Abusing Johnny debriefing questionnaires

1. On a scale of 1 to 5, how important did you feel security was in this particular test scenario, where 1 is least important and 5 is most important?

2. Do you think you sent the schedule to someone not associated with the campaign?

   Comments?

3. Was there anything you thought about doing but then decided not to bother with?

4. Is there anything you think you would have done differently if this had been a real scenario rather than a test?

5. Were there any aspects of the software you found particularly helpful?

6. Were there any aspects of the software you found particularly confusing?

7. Are there any other comments you'd like to make at this time?

Figure B.4: Debriefing questionnaire for the control group.

1. On a scale of 1 to 5, how important did you feel security was in this particular test scenario, where 1 is least important and 5 is most important?

2. Do you think you sent the schedule to someone not associated with the campaign?

   Comments?

3. Did you notice the colored bar at the base of the email window?

4. What did the yellow bar mean? The blue?

5. Did you notice the "Digital Introductions" box?

6. What did it mean when a person's name in an introduction was printed in large text?

7. What did it mean when part of an introduction was printed in yellow text?

8. Was there anything you thought about doing but then decided not to bother with?

9. Is there anything you think you would have done differently if this had been a real scenario rather than a test?

10. Were there any aspects of the software you found particularly helpful?

11. Were there any aspects of the software you found particularly confusing?

12. Are there any other comments you'd like to make at this time?

Figure B.5: Debriefing questionnaire for the ABUSE groups.

Bibliography

[1] Mark S. Ackerman. The intellectual challenge of CSCW: The gap between social requirements and technical feasibility. *Human-Computer Interaction*, 15(2):179–203, September 2000.

[2] Anne Adams and Martina Angela Sasse. Users are not the enemy. *Commun. ACM*, 42(12):40–46, 1999.

[3] E. Allman, J. Callas, M. Delaney, M. Libbey, J. Fenton, and M. Thomas. DomainKeys Identified Mail Signatures (DKIM). Internet Draft, `http://www.ietf.org/internet-drafts/draft-ietf-dkim-base-01.txt`, April 2006.

[4] Dirk Balfanz, Glenn Durfee, D. K. Smetters, and R. E. Grinter. In search of usable security: five lessons from the field. *Security & Privacy Magazine*, 2(5):19–24, Sept.–Oct. 2004.

[5] John A. Barry. *Technobabble*. The MIT Press, 1993.

[6] Jay Beale. personal communication. Sept. 3, 2006.

[7] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 164–173, May 1996.

[8] Matt Blaze, Joan Figenbaum, John Ioannidis, and Angelos D. Keromytis. The KeyNote trust-management system version 2. RFC 2704, September 1999.

[9] Rakesh Bobba, Omid Fatemieh, Fariba Khan, Carl A. Gunter, and Himanshu Khurana. Using attribute-based access control to enable attribute-based messaging. In *ACSAC '06: Proceedings of the 22nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference*, pages 403–413, Washington, DC, USA, 2006. IEEE Computer Society.

[10] Paolo Bouquet, Luciano Serafini, and Atefano Zanobini. Semantic Coordination: A New Approach and an Application. In *2nd International Semantic Web Conference*, pages 20–23, October 2003.

[11] Sacha Brostoff and M. Angela Sasse. Safe and sound: a safety-critical approach to security. In *NSPW '01: Proceedings of the 2001 workshop on New security paradigms*, pages 41–50, New York, NY, USA, 2001. ACM.

[12] Ian Brown and C. R. Snow. A proxy approach to e-mail security. *Software—Practice & Experience*, 29(12):1049–1060, 1999.

[13] K. Cameron and D. DeJoy. The persuasive functions of warnings: Theory and models. In M. S. Wogalter, ed., Handbook of Warnings. Lawrence Erlbaum Associates, Mahwah, NJ, 2006. pp. 301–312.

[14] CertiPath: enabling digital identities globally. `http://www.certipath.com/`. Visited on Jun. 13, 2008.

[15] D. Chadwick. The PERMIS X.509 role based privilege management infrastructure. In *Proceedings of 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002)*, 2002.

[16] David Chadwick, Graeme Lunt, and Gansen Zhao. Secure Role-based Messaging. In *Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS 2004),Windermere, UK*, unknown 2004.

[17] Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Martin Strauss. REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems*, 29(8–13):953–964, 1997.

[18] D. Clark, J. Elien, C. Ellison, M. Fredette, A. Morcos, and R. Rivest. Certificate Chain Discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.

[19] D. Cooper, S. Santesson, S. Farrell, S. Boeyan, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 5280, 2008.

[20] Lorrie Faith Cranor. *Web Privacy with P3P*. O'Reilly & Associates, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, September 2002.

[21] Lorrie Faith Cranor. A framework for reasoning about the human in the loop. In *Proceedings of Usability, Psychology, and Security 2008 (UPSEC '08)*, San Francisco, CA, April 2008.

[22] H. Cunningham. Information Extraction, Automatic. *Encyclopedia of Language and Linguistics, 2nd Edition*, 2005.

[23] Matt Davis. Personal communication. Dec. 17, 2007.

[24] Matt Davis. Personal communication. Jan. 3, 2008.

[25] Matt Davis. Personal communication. Jan. 18, 2008.

[26] Rachna Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM.

[27] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 581–590, 2006.

[28] Ruslan Y. Dimov. Making RBAC Work in Dynamic, Fast-Changing Corporate Environments. Technical Report TR2008-624, Dartmouth College, Computer Science, Hanover, NH, June 2008.

[29] AnHai Doan, Jayant Modhavan, Pedro Domingos, and Alon Halevy. Learning to Map Between Ontologies on the Semantic Web. In *Proceedings of The Eleventh International WWW Conference*, May 2002.

[30] Bob Dodd. Ameren. personal communication. Oct. 15, 2007.

[31] Julie S. Downs, Mandy B. Holbrook, and Lorrie Faith Cranor. Decision strategies and susceptibility to phishing. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 79–90, New York, NY, USA, 2006. ACM.

[32] Christine E. Drake, Jonathan J. Oliver, and Eugene J. Koontz. Anatomy of a phishing email. In *Proceedings of The First Conference on Email and Anti-Spam (CEAS)*, July 2004.

[33] EDUCAUSE. eduPerson Object Class — EDUCAUSE. `http://www.educause.edu/eduperson/949`. Visited on July 2, 2008.

[34] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. Security automation considered harmful? In *Proceedings of the 2007 New Security Paradigms Workshop (NSPW 2007)*, North Conway, NH, USA, September 18–21 2007.

[35] M. Elkins. MIME security with pretty good privacy (PGP). RFC 2015, October 1996.

[36] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylnen. SPKI Certificate Theory. RFC 2693, September 1999.

[37] Carl Ellison. The nature of a usable PKI. *Computer Networks*, 31(8):823–830, 1999.

[38] S. Farrell and R. Housley. An Internet Attribute Certificate Profile for Authorization. RFC 3281, 2002.

[39] Entrust government resources : PKI & the Federal Bridge Certification Authority. `http://www.entrust.com/e_government/federal_bridge.htm`. Visited on Jun. 13, 2008.

[40] Ivan Flechais, Cecilia Mascolo, and M. Angela Sasse. Integrating security and usability into the requirements and design process. *International Journal of Electronic Security and Digital Forensics*, 1(1):12–26, 2007.

[41] Ivan Flechais, Jens Riegelsberger, and M. Angela Sasse. Divide and conquer: the role of trust and assurance in the design of secure socio-technical systems. In *NSPW '05: Proceedings of the 2005 workshop on New security paradigms*, pages 33–41, New York, NY, USA, 2005. ACM.

[42] Ivan Flechais, M. Angela Sasse, and Stephen M. V. Hailes. Bringing security home: a process for developing secure and usable systems. In *NSPW '03: Proceedings of the 2003 workshop on New security paradigms*, pages 49–57, New York, NY, USA, 2003. ACM.

[43] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk. Server-Based Certificate Validation Protocol (SCVP). RFC 5055, 2007.

[44] B. Friedman, P. Lin, and J. K. Miller. Informed consent by design. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security*. O'Reilly, 2005.

[45] H. Garfinkel. A conception of and experiments with "trust" as a condition of stable concerted actions. In O.J. Harvey, editor, *Motivation and social interaction: Cognitive determinants*, pages 187–239. Ronald Press, New York, 1963.

[46] S. Garfinkel. Usable security: Design principles for creating systems that are simultaneously usable and secure, September 2003. PhD Thesis Proposal, MIT Department of Electrical Engineering and Computer Science.

[47] S. Garfinkel. *Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable*. PhD thesis, Massachusetts Institute of Technology, 2005.

[48] Simson Garfinkel. *PGP : Pretty Good Privacy*. Sebastopol: O'Reilly, 1994.

[49] Simson L. Garfinkel. Enabling email confidentiality through the use of opportunistic encryption. In *Proceedings of the 2003 annual national conference on Digital government research*, pages 1–4. Digital Government Society of North America, 2003.

[50] Simson L. Garfinkel and Robert C. Miller. Johnny 2: a user test of key continuity management with s/mime and outlook express. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24, New York, NY, USA, 2005. ACM.

[51] Alla Genkina and L. Jean Camp. Re-embedding existing social networks into online experiences to aid in trust assessment. `http://ssrn.com/abstract=707139`, April 2005.

[52] N. Goffee, S.H. Kim, S.W. Smith, W. Taylor, M. Zhao, and J. Marchesini. Greenpass: Decentralized, PKI-based Authorization for Wireless LANs. In *Proceedings of 3rd Annual PKI R&D Workshop*. NIST/NIH/Internet2, April 2004.

[53] P. Gutmann. PKI: It's Not Dead, It's Just Resting. *IEEE Computer*, 35(8):41–49, 2002.

[54] Phillip Hallam-Baker. Achieving email security usability. In *5th Annual PKI R&D Workshop: Making PKI Easy to Use*, Gaithersburg, MD, April 4–6 2006. NIST.

[55] Carl H. Hauser, David E. Bakken, Ioanna Dionysiou, K. Harald Gjermundrod, Venkata S. Irava, Joel Helkey, and Anjan Bose. Security, trust, and QoS in next-generation control and communication for large power systems. *International Journal of Critical Infrastructures*, 4:3–16(14), 5 December 2007.

[56] Educause — educause major initiatives — higher education bridge certification authority. `http://www.educause.edu/HEBCA/623`. Visited on Jan. 24, 2007.

[57] Amir Herzberg and Ahmad Gbara. Security and identification indicators for browsers against spoofing and phishing attacks. Cryptology ePrint Archive, Report 2004/155, 2004. `http://eprint.iacr.org/`.

[58] Amir Herzberg, Yosi Mass, Joris Michaeli, Dalit Naor, and Yiftach Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 2–14, May 2000.

[59] Russ Houskey and Tim Polk. *Plannning for PKI.* Wiley, 2001.

[60] Jonathan Howell. *Naming and sharing resources across administrative boundaries*. PhD thesis, Dartmouth College, 2000.

[61] HushMail - free email with privacy - about. `http://www.hushmail.com/about-how`. Accessed on June 16, 2008.

[62] Marija Ilic, Francisco Galiana, and Lester Fink, editors. *Power Systems Restructuring: Engineering and Economics*, chapter 11. Power Electronics and Power Systems Series. Kluwer Academic Publishers, Massachusettes, USA, 1998.

[63] InfoMosaic. SecureSign desktop. `http://www.infomosaic.net/SecureSignInfo.htm`. Visited on July 21, 2008.

[64] Trevor Jim. Sd3: A trust management system with certified evaluation. In *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, page 106, Washington, DC, USA, 2001. IEEE Computer Society.

[65] Clare-Marie Karat. Iterative usability testing of a security application computer systems: Approaches to user interface design. In *Proceedings of the Human Factors Society 33rd Annual Meeting*, volume 1, pages pp. 273–277, 1989.

[66] Reto Kohlas and Ueli M. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *PKC '00: Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography*, pages 93–112, London, UK, 2000. Springer-Verlag.

[67] D. Richard Kuhn, Vincent C. Hu, W. Timothy Polk, and Shu-Jen Chang. Introduction to public key technology and the federal PKI infrastructure. `http://www.csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf`, February 2001.

[68] Prabha Kundar. *Power System Stability and Control.* McGraw-Hill, 1994.

[69] Butler W. Lampson. Hints for computer system design. *SIGOPS Oper. Syst. Rev.*, 17(5):33–48, 1983.

[70] Adam J. Lee, Marianne Winslett, Jim Basney, and Von Welch. Traust: a trust negotiation-based authorization service for open systems. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 39–48, New York, NY, USA, 2006. ACM.

[71] Ninghui Li, Benjamin N. Grosof, and Joan Figenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1):128–171, February 2003.

[72] Ninghui Li and John C. Mitchell. RT: A role-based trust-management framework. In *Proceedings of The Third DARPA Information Survivability Conference and Exposition (DISCEX III)*, pages 201–212. IEEE Computer Society Press, Los Alamitos, California, April 2003.

[73] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, Los Alamitos, California, May 2002.

[74] Ninghui Li, John C. Mitchell, and William H. Winsborough. Beyond proof-of-compliance: Security analysis in trust management. *Journal of the ACM*, 52(3), May 2005.

[75] William D. Liggett. *The Changing Structure of the Electric Power Industry 2000: An Update*, chapter 7. Energy Information Administration, U.S. Department of Energy, Washington, DC 20585, 2000.

[76] J. Lyon and M. Wong. Sender ID: authenticating email. Internet Draft, `http://www.ietf.org/rfc/rfc4406.txt`, April 2006.

[77] J. Marchesini, S.W. Smith, O. Wild, and Rich MacDonald. Experimenting with TCPA/TCG hardware, or: How I learned to stop worrying and love the bear. Technical Report TR2003-476, Dartmouth College Computer Science, December 2003.

[78] Chris Masone, Kwang-Hyun Baek, and Sean W. Smith. WSKE: Web server key enabled cookies. In *Usable Security (USEC)*, February 2007.

[79] Chris Masone and Sean Smith. Towards usefully secure email. *IEEE Technology and Society Magazine, Special Issue on Security and Usability*, March 2007.

[80] George Moromisato, Paul Boyd, and Nimisha Asthagiri. Achieving usable security in Groove Virtual Office. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security*. O'Reilly, 2005.

[81] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure Online Certificate Status Protocol - OCSP. RFC 2560, 1999.

[82] nCipher. Document Signing nCipher. `http://www.ncipher.com/document-signing.html`. Visited on July 21, 2008.

[83] NERC. NERC reliability functional model version 3. `http://www.nerc.com/~filez/Functional_Model_Version3_Board_Approved_13Feb07.pdf`. Accessed on Feb. 18, 2008.

[84] Rebecca Nielsen. Observations from the deployment of a large scale PKI. In Clifford Neuman, Nelson E. Hastings, and William T. Polk, editors, *4th Annual PKI R&D Workshop*, pages 159–165. NIST, August 2005.

[85] Donald A. Norman. *The Design of Everyday Things*. Basic Books, 1988.

[86] Network security services (NSS). `http://www.mozilla.org/projects/security/pki/nss/`.

[87] Department of Defense. Nuclear weapon accident response procedures (NARP). `http://www.fas.org/nuke/guide/usa/doctrine/dod/5100-52m/chap15.pdf`, September 1990. DoD 5100.52-M, Chapter 15.

[88] U.S. House. Committee on Energy and Commerce. Blackout 2003: How did it happen and why. `http://energycommerce.house.gov/108/hearings/09032003Hearing1061/hearing.htm#docs`, September 2003. Telephone transcripts from MISO.

[89] OpenLDAP: Main page. `http://www.openldap.org/`. Visited on July 20, 2008.

[90] OpenSSL: The open source toolkit for SSL/TLS. `http://www.openssl.org`. Visited on Sept. 30, 2006.

[91] Tom Overbye. Personal communication. April 24, 2008.

[92] Mindy Pereira. Trusted S/MIME gateways. Technical Report TR2003-461, Department of Computer Science, Dartmouth College, 2003.

[93] Adrian Perrig and Dawn Song. Hash visualization: A new technique to improve real-world security. In *the proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce*, 1999.

[94] PowerWorld corporation — the visual approach to analyzing power systems. `http://www.powerworld.com`. Visited on Jun. 13, 2008.

[95] B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.1 certificate handling. RFC 3850, July 2004.

[96] B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.1 message specification. RFC 3851, July 2004.

[97] J. Reason. *Human Error*. Cambridge University Press, 1990.

[98] Jens Riegelsberger, M. Angela Sasse, and John D. McCarthy. The mechanics of trust: a framework for research and design. *International Journal of Human-Computer Studies*, 62(3):381–422, 2005.

[99] R. Rivest. S-expressions (draft-rivest-sexp-00.txt). `http://theory.lcs.mit.edu/~rivest/sexp.html`, May 1997.

[100] R. Rivest and B. Lampson. SDSI - A Simple Distributed Security Infrastructure. `http://theory.lcs.mit.edu/~rivest/sdsi10.html`, April 1996.

[101] D. M. Rousseau, Sim B. Sitkin, R. S. Burt, and C. Camerer. Not so different after all: A cross-discipline view of trust. *The Academy of Management Review*, 23(3):393–404, 1998.

[102] SAFE BioPharma association. `http://www.safe-biopharma.org/`. Visited on Jun. 13, 2008.

[103] J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, Sept. 1975.

[104] Revi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[105] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' — a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, 2001.

[106] M. Angela Sasse. Personal communication. May 2008.

[107] Pete Sauer. Personal communication. April 24, 2008.

[108] Ben Schneiderman. Science 2.0. *Science*, 319:1349–1350, March 7 2008.

[109] Bruce Schneier. *Applied Crypto*, chapter 20. Wiley, 1996.

[110] Bruce Schneier. *Applied Cryptography*, chapter 19. Wiley, 1996.

[111] A. Schutz. On multiple realities. In M. Natanson, editor, *Collected papers 1: the problem of social reality*, pages 207–259. Martinus Nijhoff, The Hague, 1962.

[112] Nelson D. Schwartz and Katrin Bennhold. Socit gnrale scandal: 'a suspicion that this was inevitable'. International Herald Tribune (Paris), February 5 2008. Finance, p. 17.

[113] Sarah Schweitzer. Parties call foul over n. h. phone-jaming suit. The Boston Globe, October 23 2004.

[114] Sara Sinclair and Sean W. Smith. Preventative directions for insider threat mitigation via access control. In *Insider Attack and Cyber Security*. Springer, 2008.

[115] Sean W. Smith and John Marchesini. *The Craft of System Security*, chapter 10. Addison-Wesley, 2007.

[116] Sean W. Smith and John Marchesini. *The Craft of System Security*, pages 243–244. Addison-Wesley, 2007.

[117] Sean W. Smith, Chris Masone, and Sara Sinclair. Expressing trust in distributed systems: the mismatch between tools and reality. In *Forty-Second Annual Allerton Conference on Privacy, Security and Trust*, pages 29–39, September 2004.

[118] S. J. Stolfo, S. M. Bellovin, S. Hershkop, A. Keromytis, S. Sinclair, and S.W. Smith, editors. *Insider Attack and Cyber Security: Beyond the Hacker*, volume 39 of *Advances in Information Security*, pages 177–181. Springer, 2008.

[119] Bruce Tognazzini. Design for usability. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security*. O'Reilly, 2005.

[120] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820, 2004.

[121] U.S.-Canada Power System Outage Task Force. Final Report on the August 14th Blackout in the United States and Canada. accessed on June 14, 2008, April 2004.

[122] W3C semantic web. `http://www.w3.org/2001/sw/`. Visited on Sept. 30, 2006.

[123] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist. X.509 Proxy Certificates for Dynamic Delegation. In *Proceedings of 3rd Annual PKI R&D Workshop*, pages 31–47. NIST/Internet2/NIH, 2004.

[124] Tara Whalen and Kori M. Inkpen. Gathering evidence: use of visual security cues in web browsers. In *GI '05: Proceedings of Graphics Interface 2005*, pages 137–144, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[125] A. Whitten and J.D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *8th USENIX Security Symposium*, pages 169–184, 1999.

[126] Alma Whitten. *Making Security Usable*. PhD thesis, Carnegie Mellon University School of Computer Science, 2003.

[127] Alma Whitten and J.D. Tygar. Usability of security: A case study. Technical Report CMU-CS-98-155, Carnegie Mellon University School of Computer Science, December 1998.

[128] M. Wong and W. Schlitt. Sender Policy Framework (SPF) for authorizing use of domains in e-mail, version 1. Internet Draft, `http://www.ietf.org/rfc/rfc4408.txt`, April 2006.

[129] Zishuang (Eileen) Ye, Sean Smith, and Denise Anthony. Trusted paths for browsers. *ACM Trans. Inf. Syst. Secur.*, 8(2):153–186, 2005.

[130] Ka-Ping Yee. Security and usability: Designing secure systems that people can use. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security*. O'Reilly, 2005.

[131] Gansen Zhao. Personal communication. Apr. 12, 2006.

[132] Gansen Zhao. Personal communication. Apr. 11, 2006.

[133] Gansen Zhao and David Chadwick. Evolving messaging systems for secure role based messaging. In *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*, pages 216–223, June 2005.

[134] Gansen Zhao and David Chadwick. Trust infrastructure for policy based messaging in open environments. In *14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2005)*, Linkping University, Sweden, June 2005.

[135] Lynne G. Zucker. Production of trust: Institutional sources of economic structure, 1840–1920. In *Research in Organizational Behavior*, volume 8, pages 53–111. JAI Press Inc., 1986.

[136] Mary Ellen Zurko. Lotus notes/domino: Embedding security in collaborative applications. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security*. O'Reilly, 2005.

[137] Mary Ellen Zurko and Richard T. Simon. User-centered security. In *NSPW '96: Proceedings of the 1996 workshop on New security paradigms*, pages 27–33, New York, NY, USA, 1996. ACM.