ABUSE: PKI for Real-World Email Trust[⋆]

Chris Masone, Sean Smith
Dartmouth College, Hanover NH 03755 USA

Abstract. Current PKI-based email systems (such as X.509 S/MIME and PGP/MIME) potentially enable a recipient to determine a name and organizational affiliation of the sender. This information can suffice for a trust decision when the recipient already knows the sender—but how can a recipient decide whether or not trust email from a *new* correspondent? Current systems 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 the design and prototype of *Attribute-Based, Usefully Secure Email* (ABUSE)—and present experimental evaluation showing that it solves the problem.

1 Introduction

Problem Why should Alice trust an email message allegedly sent by Bob? A natural answer is to use digital signatures. PKIs work to establish a binding between identity and a key pair, and in a small organization, most users probably know each other and this will be enough to establish trust. However, in large organizations or in federations of organizations, 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. *Digital signatures no longer automatically imply trustworthiness.* Systems that focus only on identity are not expressive enough to allow users to specify the right properties for conclusions in human trust settings.

Example Application Domain The electrical power grid (particularly the North American blackout of August 2003) provides wonderful examples of users needing to be able to make quick trust decisions about communication from other humans – often from other enterprises – they haven't met. *In such scenarios, knowing the name and organization of the sender is not sufficient for trustworthiness.* The sender's job, standing within the relying party's professional or social network, and even characteristics that the sender used to possess can all play a role in this trust calculation [2, 3].

Even within the same power company, operational decision makers sit in centralized control facilities that are geographically separated from the power generation and

[⋆] This paper is based on the first author's Ph.D. dissertation [1]; a preliminary design report appeared as [2]. This research was supported in part by the NSF under grants CNS-0448499 and CNS-0524695; views and conclusions do not represent those of the sponsors

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 indicating 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 [4], decreasing the probability that communicants share pre-existing trust relationships even further. 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 these issues, but current technologies do not provide support for the kinds of trust building we saw during the blackout.

Our Solution 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 [5] 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 built *Attribute-Based, Usefully Secure Email* (ABUSE), a PKI-based system to solve this problem. (Our use of "attribute" here refers to special chains of assertions, and should not be confused with X.509 Attribute Certificates.) Our design explicitly allows scalability; ABUSE users distributed across a set of organizations can use these enhanced features for trust judgment, but can these messages are still compatible with mail clients that are not ABUSE-aware.

This Paper This paper discusses the building blocks (Sect. 2), design and implementation of our prototype (Sect. 3). Sect. 4 and Sect. 5 then present the experiments we did to determine whether our system in fact solved this problem. Sect. 6 concludes.

2 Related Work and Building Blocks

S/MIME To address email security and privacy concerns, many organizations in the commercial, federal and educational sectors have deployed S/MIME [6, 7], a secure email standard that leverages an X.509 PKI [8] to provide message integrity and non-repudiation via digital signatures [9, 10]. 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.)

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. 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[1] 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 for our scenarios.

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, is a good starting point for our trustworthy email system, and the public key in particular could provide a way to hook further contextual information about the sender into the message.

Other Approaches X.509 Attribute-Based Messaging (ABM) [12] does not work on the behalf of message recipients. Instead, it focuses on allowing sender to address messages using attributes instead of identities. The problems considered in ABM are orthogonal to our work. Both Lotus Notes [13] and Groove Virtual Office [14] provide some measure of context for their users. However, none focus on the problem of providing for users adequate context for deciding whether to trust unfamiliar correspondents. *Trust Management (TM)* deals with automatically deciding a form of trust based on attributes and policies. TM systems use many different methods of representing credentials, varying across systems [15–23].

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. This approach cannot solve our problem: it would require the automated comprehension of arbitrary text from arbitrary senders; users are incapable of effectively enumerating their personal trust policies (a priori) in a machine comprehensible format; administrator-defined domain policies are difficult (and expensive) to get right; domain policies are even harder and more expensive to maintain over time; and it is unclear that domain policies useful for the average case are still applicable in exceptional circumstances.

Non-identity X.509 PKI Both *X.509 Attribute Certificates (ACs)* [24] and *X.509 Proxy Certificates (PCs)* [25, 26] are expressed in ASN.1, a binary format, just like regular X.509 ID certificates. 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 assertions to identities. Trust would be institutional in such a deployment; users trust that these assertions are granted to individuals based on some policy implemented by the issuing organization, and so they are willing to believe the bindings provided.

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.

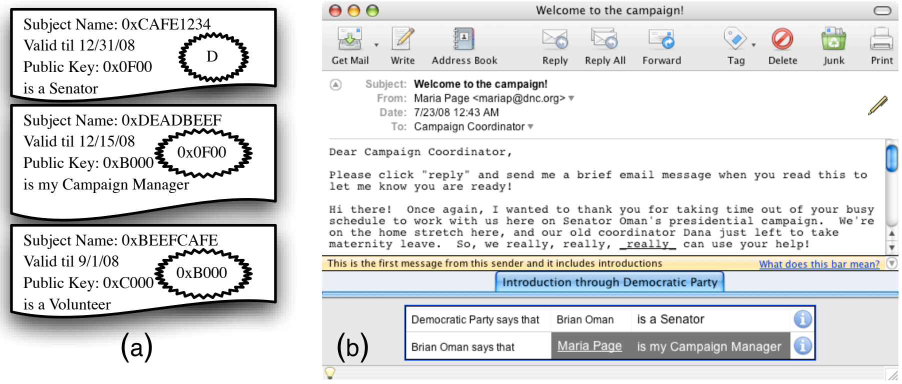
[1] *Process-based trust* leverages prior experience; *institutional* uses formal social constructs [11]

Choosing the right technology ABUSE requires signed assertions. As there are a plethora of formats available for this, it seems unnecessary to define our own. SDSI/SPKI has not seen much use outside of academic prototypes, though there is a C library for manipulating SDSI/SPKI certificates. Prior experience [27] has shown that trying to shoehorn SDSI/SPKI into an X.509-centric world can be frustrating, however. This leaves us with ACs and PCs. OpenSSL [28], a widely used cryptographic library, and NSS [29], 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.

3 ABUSE Design and Prototype

ABUSE was introduced in [2], and the first author's dissertation [1] discusses the prototype at length. The system 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, in the initial prototype, an organization-level centralized store for ABUSE attributes. The ABUSE client participates in a number of different facets of the system: attribute *presentation*, *issuance*, *distribution* and *validation*. A decentralized design for ABUSE [1] would be used in a real deployment, but for expediency and ease of user testing, the centralized design was implemented for this research.

When humans decide trust, the providence of a statement is often as important as the content; thus each ABUSE attribute is a chain of digitally signed assertions Fig. 1 (a). As is the case with a vast number of enterprise X.509 identity CAs, we expect the root of these chains to be a local entity at an organization, perhaps with a sub-CA certificate from a higher root or a cross-certificate from a bridge. Each assertion is an X.509 Proxy



Subject Name: 0xCAFE1234
Valid til 12/31/08
Public Key: 0x0F00
is a Senator

Subject Name: 0xDEADBEEF
Valid til 12/15/08
Public Key: 0xB000
is my Campaign Manager

Subject Name: 0xBEEFCAFE
Valid til 9/1/08
Public Key: 0xC000
is a Volunteer

Welcome to the campaign!

Subject: **Welcome to the campaign!**
From: Maria Page <mariap@dnc.org>
Date: 7/23/08 12:43 AM
To: Campaign Coordinator

```
Dear Campaign Coordinator,

Please click "reply" and send me a brief email message when you read this to
let me know you are ready!

Hi there!  Once again, I wanted to thank you for taking time out of your busy
schedule to work with us here on Senator Oman's presidential campaign.  We're
on the home stretch here, and our old coordinator Dana just left to take
maternity leave.  So, we really, really, _really_ can use your help!
```

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	Maria Page	is my Campaign Manager

Fig. 1: (a) A chain of signed assertions, part of which is shown in (b). We use X.509 Proxy Certificates as our assertion format. 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. (b) An example ABUSE message.

Certificate (PC). The two aspects of the PC specification that we bend relate to *naming* and certificate *validity period*. We use both *distinguished names* and public keys as identifiers in ABUSE; 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. This is the first way in which we depart from the PC specification [25], which calls for distinguished names to be used as identifiers. As with standard PCs, ABUSE does not mandate a maximum validity period. 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. 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. Thus, given a chain of assertions representing an attribute, Alice can use the private key associated with the final PC to append a new assertion to the chain, creating a new attribute for her associate Bob. In this manner Alice can make a signed statement about Bob, and provenance information is built right into the data format.

In a true PKI, individual entities generate and control their own private keys. However, to simplify the implementation of our prototype, we did not implement ABUSE attribute issuance in a distributed fashion; instead, the centralized store plays a key role in the process.

In ABUSE, attribute issuance is really just creating a new signed assertion bound to an existing attribute (which is represented by a chain of PCs). In the current prototype, when Alice wishes to grant a new ABUSE attribute to Bob, she first decides what she wants to say about him. This is a string of arbitrary text, of Alice's choice. Then, she authenticates to the ABUSE attribute store with her identity certificate, downloads her current ABUSE attributes from the centralized store and selects one whose authority she feels allows her to make the desired statement. After inputting the assertion content, Alice indicates for how long she would like this assertion 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 mints a proxy certificate containing the hash of Bob's public key in the subject field, the assertion 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, signing it with the private key associated with the final assertion in the chain indicated by Alice.

All of Alice's ABUSE 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, which Alice's client can search by public key on her behalf. Alice cannot get the private keys associated with her attributes; those never leave the store. Again, a design that obviates the need for this central store is discussed in the final chapter of [1].

PCs are natively formatted as binary data, which must be encoded as printable text in order to be sent along with email. Fortunately, there are standardized ways to do this encoding, which we use to prepare our assertion chains for transmission with an email message. Delineators are inserted between individual assertions and also between chains, so that the client on the receiving end can appropriately parse the ABUSE content for verification and display. By sending information in headers, we prevent graphical email clients that do not recognize the headers from displaying the 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 [33, p. 322].

Standard digital signatures on email cover only the body of the message. As our assertions are contained in headers, it is possible that they might be vulnerable. However, each assertion is packaged as an X.509 Proxy Certificate, digitally signed by the private key associated with the previous certificate in the chain. Thus, none of the assertions 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 ABUSE 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 chains without detection, as long as he deletes them in their entirety. In order to detect such an attack, we generate a hash of all the attributes that the sender has chosen to bind to the message, append the hash to the message text, and then allow the client to generate a signature over the entirety of the message body as usual; this preserves compatibility with current S/MIME clients (again, making it possible for ABUSE to gradually role out among a subset of distributed users).

When a message is received, the email client does its standard S/MIME signature validation. Assuming the signing certificate is not revoked, the assertions are parsed out of the header and individually validated using OpenSSL.

In ABUSE, our goal is to enable the relying party to make a decision at the time the message is received; whether the accompanying chains are still valid a year after the decision is not relevant. Consequently, the problem of rolling over public keys is not an issue for assertions in ABUSE attributes. Similarly, we felt revocation of individual assertions or whole ABUSE attributes would introduce a raft of usability problems without adding significant utility, so we chose not to explore that. (Section 5.5.1 in [1] discusses this tradeoff further.)

Presenting an assertion chain to the user is primarily a GUI issue (Fig. 1 (b)). 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*. Intuitively, the visual impact of our GUI will scale with the novelty of the information ABUSE has to display. Familiar assertions 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. Consequently, our system includes techniques to track familiarity heuristically and to use it to guide presentation; a full discussion is beyond the scope of this paper (but see [1]).

4 Evaluation in Power Grid Scenarios

Our first user study, which used task setups drawn directly from the August 2003 blackout [5], 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: ABUSE enables users to identify trustworthy messages from unfamiliar third parties, and ABUSE users do not exhibit a significantly higher rate of false positives during identification of "trustworthy messages" (those whose assertion chains indicate that it is reasonable to believe that the sender has the authority to request the stated course of action).

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.

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. 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 used scenarios inspired by 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 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 [5, 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 [34]. 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 [5, 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 [5, 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.

Subjects A total of 34 subjects took part in the study, 12 in the ABUSE group and 11 in each of the others. 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 that adequately mirrored the tradeoffs faced by real grid operators while remaining comprehensible to the subjects.

	None	Phoned someone	Checked chart	Both
Reject	$(-20, +10)$	$(-20, +8)$	$(-20, +8)$	$(-20, +6)$
Accept	$(+10, -20)$	$(+5, -20)$	$(+7, -20)$	$(+2, -20)$

Table 1: Scores for trustworthy message and attack message (resp.), depending on out-of-band channel used. 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.

Incentives 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. A subject who makes a wrong decision should still have the opportunity to be above the break-even point. Furthermore, a subject should not be able to simply make the same choice every time and come out ahead.

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 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 1.

The Study 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 cryptographically valid signed assertions). 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 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.

Scenarios Each task scenario was based on an actual event that occurred during the North American August 2003 blackout. 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 assertions bound to the message express the same flows of trust that we distilled from the phone transcripts. 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. There were three different kinds of attackers: a completely external attacker, an *internal-insider*, employed at the same company, and an *external-insider*. The subjects took the study through a web browser (Mozilla Firefox 2). Once the subject inputs his randomly-assigned ID number, the system generates a random ordering of the five scenarios and presents 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.

The Tests 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. 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 explanatory text pulled from the real email client software. Subjects in the ABUSE group also, obviously, see ABUSE assertion chains in the simulated message window.

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 assertions 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.

Results Overall success rates are shown in Table 2. 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 2a, 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 2b 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.

Next, we examine whether subjects in each email type were correct more or less often depending on scenario type—trustworthy vs. untrustworthy. Table 3 shows this analysis. 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.

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.

One might find it surprising, looking back at Table 3, 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 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.

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. 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.

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

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

Table 2: a: the mean overall level of success rates for each email type; b: overall success, ABUSE vs. all non-ABUSE user-rounds

	Trustworthy scenarios			Untrustworthy scenarios		
Email type	n	% right overall	% right, no help	n	% right overall	% right, no help
ABUSE	24	92%	**67%**	36	64%	14%
Plaintext	22	91%	**14%**	33	48%	18%
S/MIME	22	64%	**27%**	33	58%	18%
		$F = 4.24, p = .9186$	$F = 9.31, p = 0003$		$F = 0.83, p = .4405$	$F = 0.15, p = .8605$

Table 3: 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.

Looking at the observations in trustworthy scenarios within Table 3 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! 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.

5 Evaluation of User Understanding

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 assertion presentation GUI.

This time, we provided 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 ABUSE to express trust flows that involve process-based relationships in some way.

Following in the footsteps of previous research into secure email usability [35, 33], we based our study on the trappings used by Whitten and Tygar in their seminal study of PGP usability "Why Johnny Can't Encrypt" [36]. In our work, 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. 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. Previous work by other researchers did not concern itself with attackers adding events to or removing events from the schedule—but we

Expired ABUSE attribute: The attacker leverages 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 tries to convince the subject to trust him using the message body alone.

Vague attribute: The attacker binds a valid assertion chain to his message, but not one that confers authority for the accompanying request.

"John Wilson": 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.

Table 4: The kinds of attacks we explored.

do. This gives us the opportunity to create a wider range of interesting trustworthy and untrustworthy messages. In addition to modifying the subject's task from the original study and Garfinkel's follow-on "Johnny 2" [35, 33], we have expanded the campaign scenario used in previous experiments by adding a wider range of characters with a more diverse set of characteristics.

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 assertions 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 assertions bound to the message than subjects in ABUSE-two. For one group, the presented assertions would justify taking action on the message; the other group would see a different set of assertions, 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 assertions and a ignored when accompanied by another.

Our 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 was introduced to the campaign, informed of the details of his task, provided with the campaign schedule, and asked to update the schedule.

After the three setup messages, subjects began the meat of the task. They received messages requesting urgent action; the order these messages was randomized across subjects to control for order effects as in our previous study. Some messages have the same assertions in both groups; some differ.

Among the seven trustworthy messages, each kind of trust flow we identified from our analysis of the blackout transcripts is expressed at least once. Thus, showing that users can understand the assertions 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–see Table 4.

The Study After arriving in the study location, all subjects received a study procedure information sheet. The subjects were asked to "think aloud", as we needed to keep track of not only what they were doing, but also get insight into their thought process during the study. All subjects also received a short pre-study briefing. The briefing received by the ABUSE groups had added content providing a short (less than one page) introduction to "digital introductions." This is in line with Garfinkel's approach in his revision [35] of Whitten's original study [36]. 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. However, the experimenter would not answer any questions during the study beyond those about basic Thunderbird functionality (sending mail, opening new mail, etc.).

Subjects were allowed to ask for a "phone" at any time, though upon doing so they would discover that the land lines were jammed (as per [37]) 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.

Results The data we collected in this study provides qualitative evidence that users are able to understand the communication coming from the ABUSE assertion presentation GUI. Subjects exhibited an understanding of the six different kinds of trust flows we enumerated; 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 4.

In the ABUSE-one group, we used one message to test role-sourced arbitrary delegation and role-based delegation. In the ABUSE-two group, the same message was paired with different assertions to test resistance to the no-assertions attack. In the latter case, no subjects were fooled; conditioning the subjects to expect assertions on legitimate messages made them reject this attack out of hand.

We used another message 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. Compared to the control, in which 33% of the subjects acted on this message, 46% of the subjects in this group chose to respond to the sender, despite his meaningless assertion chain. These rates are comparable, especially when placed against the 93% in ABUSE-two who acted on the message when it was accompanied by an assertion chain from the campaign manager that expressed friend-sourced arbitrary delegation.

We crafter another "coopetition" message, which was 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. $6/13(46\%)$ ABUSE-one subjects acted on the coopetition message; for them, it was accompanied by a coopetition trust flow. There were also three more ABUSE-onw users who clearly indicated that they understood what was being expressed by the assertions on the message; they simply remained leery of responding with sensitive information.

In the real world, we often see situations where relying parties make trust judgments based on a *former* affiliation a sender had; e.g., in the blackout transcripts, Alice would indicate trust of Bob because of whom he used to work with at a previous job. Thus, we wanted to test how ABUSE users handled valid, unexpired messages supported by assertions that had expired. We crafted a set of messages to cover the necessary cases: the first was trustworthy in the control group, and remained so when bound to an expired assertion chain; the second was *untrustworthy* in the control group, and became trustworthy when a chain of useful, unexpired assertions were sent with the message; the last, essentially the same as the second, was deemed untrustworthy by both the control subjects as well as those who saw a version of the message bound to an expired assertion chain. The numbers we see confirm that users pay attention to expiration status. The first was acted upon in 78% of cases across both ABUSE groups, the last in only 15%. The second message was trusted by 25% of control subjects, mistrusted by 100% of users who saw it with no assertions, and trusted by 83% of those who saw it bound to a chain of useful, valid assertions.

We also tried a "John Wilson" attack, difficult to defend against, especially when the user is not personally familiar with the "John Wilson" being impersonated. 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. 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)

6 Conclusions

In this work, we 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.

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 assertion chain 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. (Future work should also examine the usability of assertion creation and selection.)

Our system thus provides a usable, scalable way for users in such distributed organizations to make meaningful but speedy trust judgments about messages from senders they do not know a priori; previous PKI systems did not.

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.

References

1. Masone, C.: Attribute-Based, Usefully Secure Email. PhD thesis, Dartmouth College (August 2008)
2. Masone, C., Smith, S.: Towards usefully secure email. IEEE Technology and Society Magazine, Special Issue on Security and Usability (March 2007)
3. Smith, S.W., Masone, C., Sinclair, S.: Expressing trust in distributed systems: the mismatch between tools and reality. In: Forty-Second Annual Allerton Conference on Privacy, Security and Trust. (September 2004) 29–39
4. Ilic, M., Galiana, F., Fink, L., eds.: 11. Power Electronics and Power Systems Series. In: Power Systems Restructuring: Engineering and Economics. Kluwer Academic Publishers, Massachusettes, USA (1998)
5. 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.
6. Ramsdell, B.: Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.1 message specification. RFC 3851 (July 2004)
7. Ramsdell, B.: Secure/Multipurpose Internet Mail Extensions (S/MIME) version 3.1 certificate handling. RFC 3850 (July 2004)
8. Cooper, D., Santesson, S., Farrell, S., Boeyan, S., Housley, R., Polk, W.: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 5280 (2008)
9. Kuhn, D.R., Hu, V.C., Polk, W.T., Chang, S.J.: Introduction to public key technology and the federal PKI infrastructure. `http://www.csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf` (February 2001)
10. Nielsen, R.: Observations from the deployment of a large scale PKI. In Neuman, C., Hastings, N.E., Polk, W.T., eds.: 4th Annual PKI R&D Workshop, NIST (August 2005) 159–165
11. Zucker, L.G.: Production of trust: Institutional sources of economic structure, 1840–1920. In: Research in Organizational Behavior. Volume 8. JAI Press Inc. (1986) 53–111
12. Bobba, R., Fatemieh, O., Khan, F., Gunter, C.A., Khurana, H.: 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, Washington, DC, USA, IEEE Computer Society (2006) 403–413
13. Zurko, M.E.: Lotus notes/domino: Embedding security in collaborative applications. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security* O'Reilly, 2005.
14. Moromisato, G., Boyd, P., Asthagiri, N.: Achieving usable security in Groove Virtual Office. In Lorrie Cranor and Simson Garfinkel, editors, *Usability & Security* O'Reilly, 2005.
15. Li, N., Grosof, B.N., Figenbaum, J.: Delegation logic: A logic-based approach to distributed authorization. ACM Transactions on Information and System Security (TISSEC) **6**(1) (February 2003) 128–171

16. Li, N., Mitchell, J.C., Winsborough, W.H.: 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)
17. Li, N., Mitchell, J.C.: RT: A role-based trust-management framework. In: Proceedings of The Third DARPA Information Survivability Conference and Exposition (DISCEX III), IEEE Computer Society Press, Los Alamitos, California (April 2003) 201–212
18. Li, N., Mitchell, J.C., Winsborough, W.H.: Beyond proof-of-compliance: Security analysis in trust management. Journal of the ACM **52**(3) (May 2005)
19. Jim, T.: Sd3: A trust management system with certified evaluation. In: SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2001) 106
20. Herzberg, A., Mass, Y., Michaeli, J., Naor, D., Ravid, Y.: Access control meets public key infrastructure, or: Assigning roles to strangers. In: Proceedings of IEEE Symposium on Security and Privacy. (May 2000) 2–14
21. Blaze, M., Figenbaum, J., Ioannidis, J., Keromytis, A.D.: The KeyNote trust-management system version 2. RFC 2704 (September 1999)
22. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: Proceedings of IEEE Symposium on Security and Privacy. (May 1996) 164–173
23. Chu, Y.H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: REFEREE: Trust management for Web applications. Computer Networks and ISDN Systems **29**(8–13) (1997) 953–964
24. Farrell, S., Housley, R.: An Internet Attribute Certificate Profile for Authorization. RFC 3281 (2002)
25. Tuecke, S., Welch, V., Engert, D., Pearlman, L., Thompson, M.: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (2004)
26. Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Tuecke, S., Gawor, J., Meder, S., Siebenlist, F.: X.509 Proxy Certificates for Dynamic Delegation. In: Proceedings of 3rd Annual PKI R&D Workshop, NIST/Internet2/NIH (2004) 31–47
27. Goffee, N., Kim, S., Smith, S., Taylor, W., Zhao, M., Marchesini, J.: Greenpass: Decentralized, PKI-based Authorization for Wireless LANs. In: Proceedings of 3rd Annual PKI R&D Workshop, NIST/NIH/Internet2 (April 2004)
28. OpenSSL: The Open Source toolkit for SSL/TLS. `http://www.openssl.org`
29. NSS: Network Security Services. `http://www.mozilla.org/projects/security/pki/nss/`
30. Ellison, C.: The nature of a usable PKI. Computer Networks **31**(8) (1999) 823–830
31. Rivest, R., Lampson, B.: SDSI - A Simple Distributed Security Infrastructure. `http://theory.lcs.mit.edu/~rivest/sdsi10.html` (April 1996)
32. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylnen, T.: SPKI Certificate Theory. RFC 2693 (September 1999)
33. Garfinkel, S.: Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable. PhD thesis, Massachusetts Institute of Technology (2005)
34. Dodd, B.: Ameren. personal communication. Oct. 15, 2007
35. Garfinkel, S.L., Miller, R.C.: 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, New York, NY, USA, ACM (2005) 13–24
36. Whitten, A., Tygar, J.: Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In: 8th USENIX Security Symposium. (1999) 169–184
37. Schweitzer, S.: Parties call foul over N. H. phone-jaming suit. The Boston Globe (October 23 2004)