Not Sealed But Delivered: The (Un)Usability of S/MIME Today

Ann Fry, Sonia Chiasson, and Anil Somayaji

***Abstract*—Despite widespread concerns about email privacy and security, today virtually all email is sent without encryption or authentication. Past work has identified usability issues with encrypted and digitally signed email. The most recent work in this area, however, was in 2005, with the more significant work having been done over a decade ago. In this paper we re-examine the issue of the usability of cryptographic email protections through cognitive walkthroughs of the current interfaces to S/MIME functionality in three commonly used email clients. We found that the usability situation has not improved, and in some ways it has gotten worse. In order to address the multiple hurdles that exist in current S/MIME implementations, there need to be significant, coordinated investments in the usability of end-to-end email cryptography.**

***Index Terms*— Email Encryption, Human Computer Interaction, Security, S/MIME**

I. Introduction

EMAIL is a fundamentally insecure means of communication. With email being used to transmit sensitive information such as personal correspondence, financial, health or confidential business data, a significant amount of information is at risk in email correspondence. Standard email, however, is not end-to-end secure, meaning that there are numerous ways in which messages can be intercepted on their way from a sender to a recipient.

Currently there are two widely recognized standards for sending end-to-end secure email: PGP [7] and S/MIME [4]. While PGP was developed first (in 1993), today the most common form of email encryption built into email clients is the S/MIME standard. Using S/MIME it is possible to send email between individuals such that confidentiality and integrity are both assured. Unfortunately, today very few email messages are sent using either S/MIME or PGP.

A key factor in why email is not secured using these standards is the lack of usability of their implementations. This issue was first rigorously examined in 1999 in the "Why Johnny can't encrypt" paper [6]. This paper reported that regular users could not successfully sign or encrypt email. In 2005 the "Johnny 2" paper [2] found significant problems with S/MIME usability. Since 2005, regular users have become more and more concerned about confidentiality on the Internet. Has the situation changed?

To address this question, we present cognitive walkthroughs of three modern email clients used in the context of four exemplar scenarios where individuals would want to exchange secure messages. The cognitive walkthrough is a standard approach in the usability literature for identifying "pain points" that cause users to fail at accomplishing given tasks. These walkthroughs show that users of all skill levels are likely to encounter problems. Regular users are likely to falter early in the setup process of S/MIME, long before they get the chance to compose a confidential email. Further, even experts face many barriers in the process of sending and receiving secured emails.

Our key contribution here is our examination of S/MIME usability in three mail clients that have not been studied previously for email security. This examination shows that the problems previously identified with certificate management have not been addressed, and that problems are widespread, occurring on multiple platforms, in multiple email client implementations, and involving multiple pain points. One surprising result is that most of these pain points do not go away even if we assume users have a proper understanding of digital certificates, the technology underlying S/MIME. These pain points will need to be addressed before it will be reasonable for even expert users to routinely secure their email using S/MIME. Unfortunately, the problems are not isolated to any one program or platform; thus, a coordinated effort will be required in order to improve S/MIME usability.

The rest of this paper proceeds as follows. We first give background on using cryptography to secure email and cover the basics of how standard email works in Section II. In Section III, we explain the cognitive walkthrough methodology. Then we give cognitive walkthroughs for four scenarios involving the three email clients OS X Mail, Evolution and Thunderbird in Sections III, IV, V and VI. Section VII discusses inconsistencies we observed in security interface elements. A discussion of pain points and perspectives on improving S/MIME usability is in Section VIII. Sections IX and X conclude.

II. Background

Electronic mail—email—was developed with the goal of transferring messages reliably between parties running diverse software stacks that may not have continuous network connectivity. To enable interoperability, email messages are encoded and sent as text, with text transformations permitted and even expected (for example, to account for differences in text encodings). To address connectivity issues, email delivery has traditionally been implemented using a store-and-forward

This work was supported by Canada's Natural Sciences and Engineering Research Council (NSERC) through the ISSNet Strategic Network (www.issnet.ca).

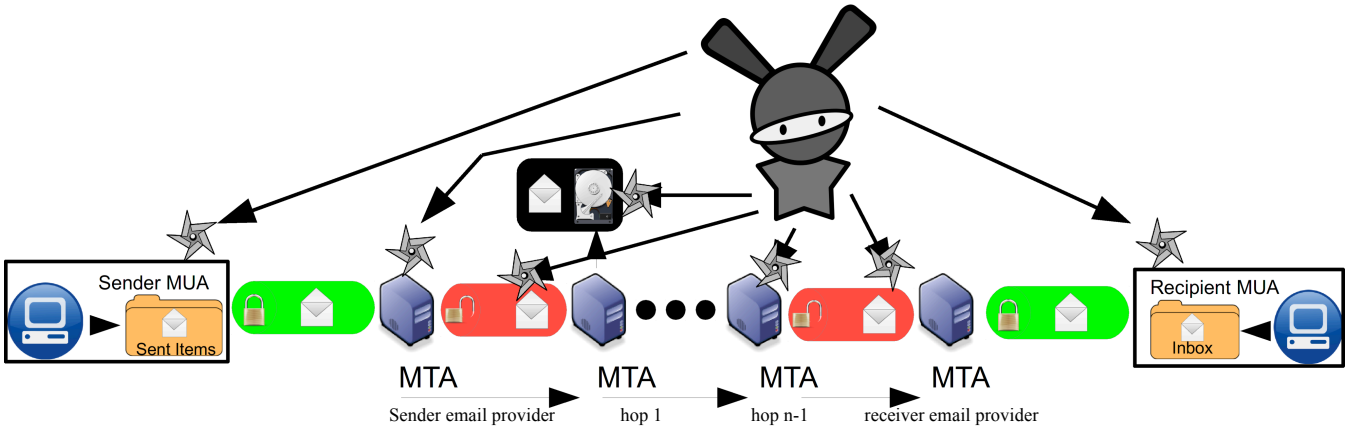

Fig. 1 Message secured with SSL/TLS (e.g., standard web email)

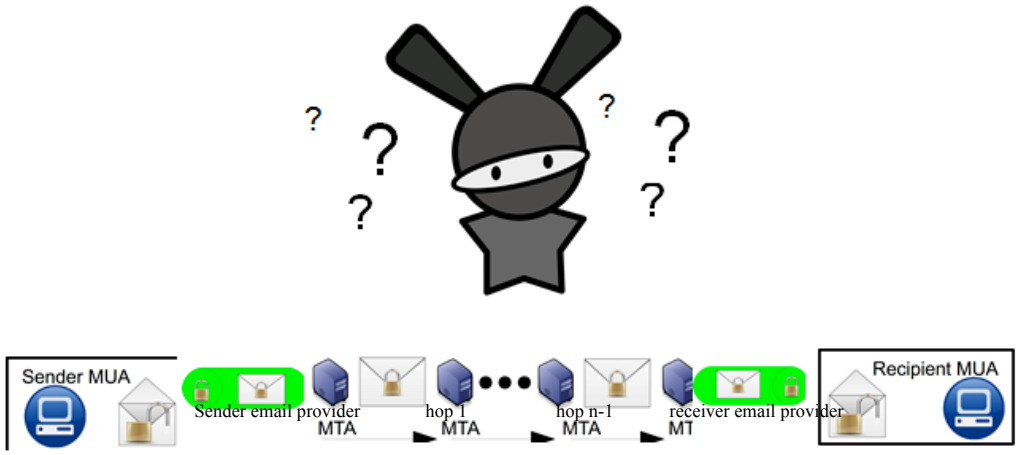

Fig. 2 Message secured with end-to-end encryption (s/MIME or PGP)

architecture: messages would pass through multiple intermediate servers, each of which would store the message for an indeterminate period of time before passing it on.

Because of the constraints of text translation and store-and-forward message processing (among others), email has from the beginning been a means of communication that has no inherent provisions for message integrity or confidentiality. Messages can and are examined and often changed in transit, with copies potentially being archived by many entities. While these characteristics are present in older email systems such as UUCP and BitNet, we can also see them in modern Internet email, particularly as transmitted by the Simple Mail Transport Protocol (SMTP) [3].

In SMTP, a message is relayed from a sender's email client (a Mail User Agent, MUA) to a local email server (a Message Transport Agent, MTA). The local MTA then relays the message to other MTAs, as necessary, until it reaches the MTA of the message recipient. The recipient's MUA then retrieves the message from its local MTA, traditionally using the POP or IMAP protocols, or more recently using a webmail interface. As we can see in Fig. 1, it is often the case that links between MUAs and MTAs or even between MTAs may be authenticated and encrypted using SSL/TLS. Such protection is now very important to minimize email spam; as can be seen in the figure, however, such protections do not prevent attackers from intercepting and modifying messages at multiple points. The problem is that any authentication and encryption that is layered over the message is repeatedly removed while the message is in transit.

In order to provide end-to-end email security, we need a way for MUAs to encode and decode messages such that MTAs cannot view the plaintext and/or cannot modify messages maliciously without being detected. This is what PGP and S/MIME are designed to do, see Fig. 2. Note that while some clients have built-in PGP support (particularly on UNIX-like systems that make use of the free GNU Privacy Guard implementation of the OpenPGP standard) the most widely supported email security standard is S/MIME. We

should mention that most implementations of web-based email, such as Microsoft's Hotmail and Google's Gmail, support neither PGP nor S/MIME. However, S/MIME is natively supported by many popular desktop email clients, including Microsoft's Outlook & Outlook Express, Apple's OS X Mail, Evolution (on Linux), and Mozilla Thunderbird.

In order for individuals to send fully secured email, they must each have email certificates issued to them personally, and they must exchange the publically disclosable portion of these certificates. The sender's private key is used to sign the outgoing message and the receiver's public key is used to encrypt it. The recipient reverses this process, decrypting using their private key and verifying the integrity of the message using the sender's public key.

The concepts underlying digital signatures and encryption using public key cryptography can be surprisingly subtle. Indeed, a key point of both earlier studies of email encryption usability [6] [2] was that user interfaces have to be carefully designed in order to properly convey the underlying cryptographic concepts. In this work, however, we generally assume that the users in question already understand certificates and public key cryptography, at least in an abstract sense. As we shall see, such background knowledge is not necessarily enough to enable users to accomplish the goal of sending and receiving email securely.

III. Cognitive Walkthrough Setup

In a cognitive walkthrough, an expert analyst attempts to simulate the likely behavior of a hypothetical user attempting to accomplish a given task. Our cognitive walkthrough methodology closely follows the one described by Wharton [5]; however, we also incorporate our own modifications (indicated below) in order to help us provide readable yet detailed process observations.

To prepare for a cognitive walkthrough, we first must define/extract four things:

- Users or personas
- Sample task(s) for evaluation
- Action sequences for completing the task(s)
- A description of the interface (in our case, interfaces)

Each user or persona is given a background story and is additionally defined by their assigned knowledge level and their scenario linked goal. Individual scenarios are constructed to coincide with the nature of the individual personas. Normally for a complete cognitive walkthrough, actions are the individual steps a user has to follow within the action sequence that lead toward success, for example, clicking on a button, selecting a menu bar, or a single menu option. To simplify our exposition, here action sequences are comprised of groups of actions rather than individual ones. For each action, the expert notes specific assumptions, prior knowledge, and knowledge acquired during action sequences. These assumptions help the expert choose the appropriate responses to four questions that should be considered for each action sequence:

1) Will the user attempt to achieve the correct effect?
2) Will the user notice that the correct action is available?
3) Will the user associate the correct action with the effect they are trying to achieve?
4) If the correct action is taken, will the user see that progress toward the solution of their task is made?

In addition to these answers, we also include a *persona critique* to describe the rationale behind choices made by each user. In our scenarios, personas wish to either send or receive an email message that has been encrypted and signed using S/MIME. Fig. 3 shows the sub-goals a user must accomplish in order to succeed at these goals.

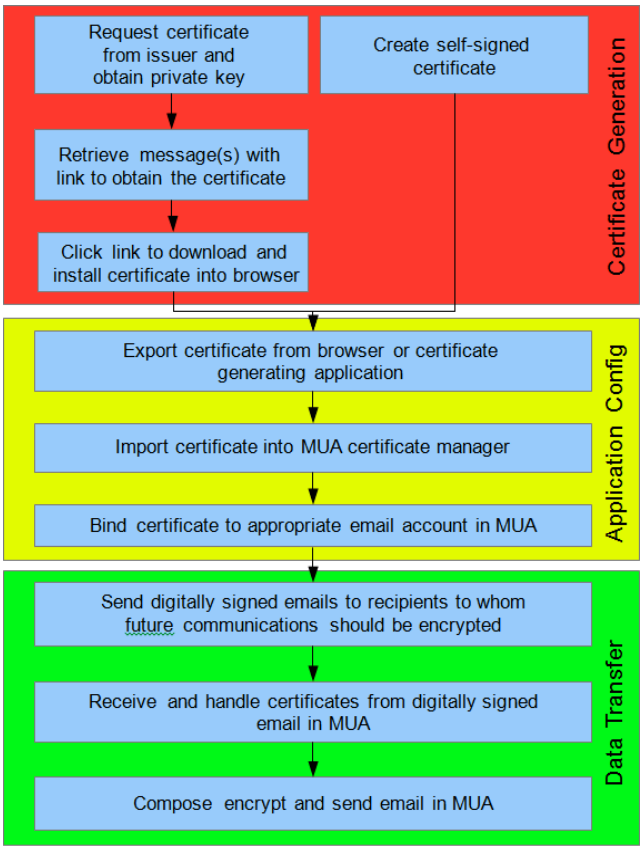

Fig. 3 Steps to encrypt and sign email with S/MIME certificates

Here we evaluate three email clients: the Mail application (v. 5.2) running on Mac OS X Lion (v. 10.7.1 (build #: 11B26)), Evolution (v. 3.2.2) and Thunderbird (v. 11.0.1) running on Xubuntu (v. 11.10), a variant of the commonly used Ubuntu Linux distribution. Because users need their own certificates, we also do walkthroughs of three methods for obtaining free S/MIME mail certificates. One is to create a self-signed certificate locally. For this, we evaluated the certificate generation functionality of the Mac OS X Keychain Access application (v. 5.0 (build #: 55108)). The other two involve certificate authorities that provide web interfaces to their free-for-personal-use email certificates:

1) InstantSSL/Comodo: http://www.instantssl.com/ssl-certificate- products/free-email-certificate.html
2) Tc TrustCenter: http://www.trustcenter.de/en/products/tc_certificates.htm

The web browser used by our personas to obtain certificates

and browse associated issuer websites is Mozilla Firefox (v.11) running on either Mac OS X or Xubuntu.

In our study the personas and scenarios are fictional. To ensure comprehensive coverage of the defined problem space according to the number of email clients and the types of certificates, we use four personas: one non-technical user, two more technically inclined users and one expert user. The first scenario introduces the reader to Karen, a non-technical user and secretary who must transmit financial data to the parent company of her firm. Next, we assume the role of Karen's husband Joe. Joe operates as a stay-at-home dad with a keen interest in the Linux operating system. He is attempting to retrieve a ticket securely for an underground party from his best friend. Third, to examine the use of self-signed certificates with S/MIME, we visit the story of Jasmine and Jacob. They are a brother and sister pair organizing a political protest against an oppressive regime. Lastly, Cameron, an overtly paranoid and computer savvy Ph.D. student must securely send a groundbreaking idea to her supervisor.

IV. Scenario 1: The Secretary

Karen is the secretary of the Chief Executive Officer (CEO) of a German company which does multiple business transactions with a large computer security firm. She was instructed to request and obtain an S/MIME certificate issued by TrustCenter by the company's internal service desk. After this point she received an email that included both a link to the website and rudimentary instructions. The instructions include two tasks. The first is that she must verify the destination of the link. The second indicates she must use the certificate to sign an email. This email should request a digitally signed response sent from the parent company's financial services department. When complete, this would enable future mail to be secure.

A. Scenario setup

Goal: To transmit financial data securely via email to the parent company.

Knowledge Level: Karen is a very non-technical user. She is accustomed to typing emails and other word processing tasks, web browsing, answering phones and keeping track of appointments. Due to the nature of her job, she must both multi-task and retain enough details in her memory sufficiently to complete each of her current tasks. This does not leave much memory space to attempt to understand or manipulate ways around non-trivial tasks. She retains the ability to follow basic instructions, as long as they do not consume too much of her time. Since she has many tasks at hand, and the financial quarter is drawing to the end, her situation is rushed.

Assumptions: Karen assumes that a lock icon means that her communication is secure. She employs Mozilla Firefox as a browser and she just clicked the link within the email from the service desk. The browser window opens at the page for requesting certificates at TrustCenter.de.

B. Certificate Generation

Action 1: On the TrustCenter.de website, click the "Request Certificate" button.

Persona Critique: In order to comply with the instructions to verify the link destination, Karen quickly skims the information presented on the preview segment of the website. She notices three aspects about the preview page. The first indication that she has arrived in the correct place is the page title labeled TrustCenter. The second aspect she notes is that the page does not mention S/MIME, but it states that her details will be transferred in encrypted form utilizing something called "SSL". Finally, she reads that commercial users are supposed to purchase their certificate, but ignores this due to the known frugal nature of her boss. She chuckles to herself when she reaches the part of the page that contains German words within the English text.

Response: Karen clicks the button to request the certificate. The button is clickable, bright orange and centrally located on the website. It is appropriately labeled "Request Certificate", indicating to the user that clicking on the button is required to proceed with the certificate issuing process. This action indirectly indicates progress toward the end goal, since the user had received instructions which stated that certificate issuance was necessary for progression. From this knowledge, Karen believes she is doing the right thing. In response to clicking the button, a web form then loads to indicate the next step in the issuance process.

Action 2: Fill in details (see Appendix A.1 for the complete listing) for the first web form, and click the "Next" button.

Persona Critique: For details numbered 3, 4 and 5 on the web form, two textboxes are provided for the user to fill in their responses. The areas are labeled with what is being requested; however, neither of the individual boxes are directly associated with individual labels. Karen found this confusing. She indicates, for example, that she thought she may put her zip code in the textbox designated location, and her location in the textbox designated zip code.

Response: Since Karen is familiar with filling out ordering forms and address labels on other Internet websites as a part of her everyday activities, she fills out the initial certificate issuer request web form and clicks the "Next" button. Other than the issue mentioned within her personal critique, all steps within this action are made available to Karen. All textboxes can be typed in and the "Next" button is big, blue and clickable. From the information presented in the banner, Karen can see that this is the first web form of three. She is rushed, and therefore feels a growing feeling of annoyance considering that they have to fill out so much information. She still assumes that this is a necessary action to the completion of the certificate issuance process. Upon completion of these action steps, the display of the next web form including new information located in the banner of the page indicates to Karen that she is progressing towards her goal.

Action 3: Fill in details (see Appendix A.2 for the complete listing) for the second web form, accept and confirm acceptance of the terms and conditions in the pop-up window and click the "Next" button.

Persona Critique: In the second section box, under the radio buttons for choosing email or SMS there is a textbox that Karen completely overlooks because she is in a hurry. It also does not have a label. This indicates to the analyst that if

another more cautious user was performing the walkthrough, that persona also may not realize why the textbox exists.

Response: Karen fills out the two textboxes to associate the revocation password. She then selects email as the notification method by clicking on the circle. The next circle she clicks indicates she accepts the general terms and conditions. Upon clicking this, she is presented with a popup window that includes a description of the general terms and conditions. She pulls the window slider on the side of the window skipping the text altogether, and proceeding to the bottom of the page content. She then clicks the "Yes" button. The popup window disappears, and Karen then clicks on the circle to accept the data handling stipulation. Finally, the last step Karen fulfills on this web form is to click the large "Next" button at the bottom of the page. Unfortunately, due to the empty textbox, Karen has failed to provide mandatory information for the web form to be completed. The cause of this is a product of Karen being rushed as well as the textbox being inadequately labeled. Should Karen have completed all of the steps within this action, she would have been taken to a website that indicates she would receive two email messages with further instructions as to how to complete her goal. Instead she is presented with the same web form page with an indication that it has not been completed.

Conclusion: By this point of time, Karen displays frustration and annoyance. She does not want to try to figure out what information was missed. Instead she calls the company's financial institution, and requests that they send the data securely to the parent company. The bank informs her that this action will require payment. Since she has no authorization to spend money for this endeavor she gives up on this methodology to secure the data. She then remembers that her personal webmail account seems secure. She previously saw both the lock icon, and from prior instruction that the connection to the webmail account was secured with "SSL". From the knowledge presented on the certificate issuer's website, and the association to her prior knowledge of the connection to her webmail account, Karen then sends the financial data using her webmail account. A week later her boss is replaced by an upper level executive from Karen's webmail provider, as the software security company was bought out, leveraging the data contained within that email. Note that according to Fig. 3 this user only achieved partial certificate generation. We do not have the other sections included in this walkthrough since the user did not proceed any further.

V. Scenario 2: The Underground Party Invitation

Karen's husband Joe is a stay at home dad. He claims using a Linux-based operating system is his hobby, and enjoys using it every day. One day, he receives an encrypted message from his best friend Jeremy that cannot be decrypted. Jeremy is more technically savvy and runs Xubuntu with Evolution as his MUA. In an instant message conversation Joe asks Jeremy how to encrypt messages. Jeremy informs Joe that he should first obtain a valid S/MIME certificate. Jeremy then requests that Joe send him a digitally signed message using the newly issued certificate. From the conversation Joe understands that this is a required step to facilitate future private communications. Jeremy's original intent is to send Joe an invitation to an underground party. For this specific scenario, neither Joe nor Jeremy is in a rush to accomplish his desired task.

A. Scenario Setup

Goal: Receive and decrypt an encrypted S/MIME email containing the ticket to an invitation-only underground party.

Knowledge Level: Joe is an above average Internet user who has not fully discovered all of the options within the preferences pane of his browser. He is thoughtful, self-taught and likes to read documentation.

Assumptions: Joe runs and is familiar with his operating system (Xubuntu), his MUA (Thunderbird) and his browser (Firefox). His MUA is already set up with his email account information, and he can send and receive emails. He clicks the link shared by Jeremy in the instant message conversation and the page to request a certificate from Comodo, also known as InstantSSL, loads in the Firefox window.

B. Certificate Generation

Persona Critique: Prior to the first action Joe reads the information present on the initial website, and discovers that the page disseminates zero knowledge regarding the installation or usage of the issuer's certificate.

Action 1: Find more information on the installation of the S/MIME certificate by reading and navigating the certificate issuer's website.

Response: On the website Joe clicks the "Secure E-mail Certificates" link in the hopes that he will obtain more instructional information about how to install the certificate. The link provided is highlighted in blue, and can be clicked upon for navigation. Joe realizes from personal web browsing experience that most websites are laid out so that link traversal lands the user on the appropriately relevant page. When Joe clicks on the link, he is taken to the company's commercial website advertising the sale of SSL products. Joe realizes that this will not complete of this action or enable progress toward the overall goal of the scenario, so he clicks the "Back" button on his web browser. Similar to cognitive walkthroughs Joe's self-taught learning process includes a method to learn by doing.

Action 2: Click the "Get It Free Now!" button

Response: Joe clicks the "Get It Free Now!" button. The button itself is beside the description for the free email certificate. It is also centrally located and placed near the top of the page. It is a fairly large button with a long label which makes it noticeable. On the mouse over, it reveals that the button is something that can be clicked on. Joe would recognize this and know from prior browsing experience that the option is available. The label beside the button indicates that this is the next step in the certificate issuing process. The next webpage that appears after clicking the button is a web form application for a secure email certificate. Also from previous web browsing experience, it appears to Joe to be a genuine step in the right direction towards the end goal.

Action 3: Fill out the first web form with contents, read and

accept subscriber agreement, then click the "Next" button.

Persona Critique: Taking into consideration Joe's inability to tolerate spam, the opt-in checkbox for the newsletter on this web form is cleared. While reading the information presented by the agreement, Joe learns that the certificate, once issued, will follow both X.509 and S/MIME specifications. Both of these aspects Joe wanted to verify, and the information was made available to him.

Response: Joe fills out the details (see Appendix B.1 for the complete listing) requested by the form and clicks the "next" button. Instructions are prompted by the process bar on the right side of the page. The "next" button at the bottom of the page is fairly small but clear enough to an Internet savvy user. The process bar on the right indicates that filling out these details is a required step in the certificate issuing process. Clicking the "next" button brings up a small window that flashes and says "generating key"[1]. When that window disappears a page loads which says that the application was successful and that details on how to collect the certificate would be sent to his email address. Joe deems this as significant progress and as instructed attempts to check his email.

Action 4: Click the application menu, and then click Mail Reader to open Thunderbird. In order to receive the email message, click the "get mail" button in the top left corner. Open the certificate email from Comodo / InstantSSL by highlighting the message in the mail message preview pane. Read the information included within the email and click the "Click & Install Comodo Email Certificate" link.

Persona Critique: The email provides a Frequently Asked Questions (FAQ) link, as well as a link to configure your email client to use the certificate for securing email. The text in the email claims that the certificate will be automatically placed into the certificate store on your computer. Joe thinks to himself: Where is the certificate store? His thoughts are preoccupied with wondering about this step that he does not click the link regarding the configuration of email clients. The information in the FAQ link leads to content that only applies to an old version of Internet Explorer or Outlook. Should he have clicked the "configure your email" client link, the options provided to help install the certificate are all for outdated MUAs. (Outlook Express 5 & 6, Outlook 98, 2000, & 2003, as well as an unspecified, but very old version of Mozilla Thunderbird)

Response: Due to familiarity with his operating system and MUA, Joe easily completes the steps necessary to retrieve the email from InstantSSL using Thunderbird. Upon receipt of the message, he clicks on it to highlight the message, and load it in the preview pane. Knowledge of how to do this is based upon past experience with his MUA. The link contained within the email is large, red and available to click. It is also clearly labeled "Click & Install Comodo Email Certificate". When Joe clicks on the link, a Firefox window opens. A script is run that causes an "Alert" window to popup with an exclamation mark circled in red as shown in Fig. 4. Joe believes this indicates definite progress towards his goal because he has now been told the certificate has been installed.

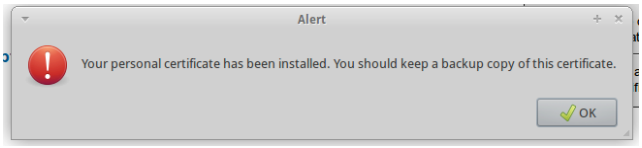

Fig. 4 Firefox alert popup window from Comodo / InstantSSL

C. Application Configuration

Action 5: Export the newly installed certificate

Persona Critique: From the alert window, Joe was last instructed to "keep a backup copy" of the certificate, and clicked the "Ok" button to close it. Unfortunately, at no point in this action sequence were there any indications to inform Joe where the certificate store was located on his computer. He feels he cannot attempt this last action.

Response: Working from the information he was given, Joe would have no idea how to back up the certificate or where it was stored on his machine. He does not know where the correct action is or how to perform it. Joe associates the certificate backup with the correct procedure as indicated from instructions in the email and the alert, but cannot perform the action.

Conclusion: At this point, Joe calls Jeremy and asks him to physically print the ticket. He would meet up with Jeremy later that week for coffee where the invitation would be exchanged in person. Note that according to Fig. 3 this user only achieved partial application configuration. We do not have the other sections included in this walkthrough since the user did not proceed any further.

VI. Scenario 3: The Great Firewall

A college student named Jasmine lives in a country that denies access to many websites. All of the websites that perform certificate issuing are included in this block list. Jasmine wants to contact her brother Jacob, who is working on a cruise ship in the Caribbean. She wishes to inform him, and attempt to enlist his aid in the organization of a political protest of the imposed Internet firewall. Finally, she suggests that he should cut his working vacation short to come home, in case she needs to be bailed out of prison.

A. Scenario Setup

Goal: To send an encrypted and digitally signed email using an OS X Keychain Access generated self-signed S/MIME certificate.

Knowledge Level: Jasmine is a Macintosh Apple user familiar with the Keychain Access application and elementary public key cryptography principles.

Assumptions: Jasmine and Jacob do not have access to a telephone. Written letters or "snail mail" are opened and resealed at the post office. The governmental firewall blocks all other methods of communication including Skype and instant messaging applications. In technical terms, all methods of out-of-band communication are blocked in this scenario.

[1] This is when the private key is generated and installed into the browser. It is later verified when attempting to collect the actual certificate. A user who tries to use a different browser to collect the certificate will be unable to. Joe does not know any of this information.

B. Certificate Generation

Action 1: Open Finder, click on Applications, scroll to the Utilities folder and double-click to open, scroll to and double click on the Keychain Access application.

Response: Her familiarity with both the OS X Lion operating system and Finder application provides her with sufficient knowledge of how to complete all of the steps in this action. All of the steps were visible and available as well as sufficiently labeled. She finds and opens the Keychain Access application successfully. From prior knowledge and from the presentation of certificate information in the application, Jasmine confirms that certificates are stored here. She also remembers seeing the certificate assistant option clearly labeled in the main menu. Upon completing this action, the menu items listed at the top of the screen are modified according to normal OS X Lion operating system behavior.

Persona Critique: Jasmine has never created a self-signed S/MIME certificate on her computer before, so she decides to consult the help documentation for the Keychain Access application next.

Action 2: Click the help menu on the top status bar and type in "create self-signed certificate". Then highlight and click the option "Create self-signed certificates" in the given dropdown menu. The documentation found therein indicates the exact steps required to generate a self-signed S/MIME certificate using the Keychain Access application.

Response: Since Jasmine is familiar with her operating system, she is able to obtain the search help textbox. Due to her familiarity with public key cryptography, and the knowledge that she wishes to obtain a self-signed certificate, she is able to formulate an effective query. Each step within this action was made available to Jasmine. The interactions and results occur as she expects them to. The search term showed up completely unaltered within the search results. Upon selection, a new window opened with the associated documentation that provided instructions both specific and complete. From this information, Jasmine obtained her answer, and can now proceed with the task of generating her own certificate.

Action 3: Click "Keychain Access" menu on the status bar at the top of the screen, highlight "Certificate Assistant...", and click "Create a certificate".

Fig. 5 OS X mail security icons

Response: From the previous action, the help documentation provided adequate instructions to complete all of the necessary steps for the next two actions. Jasmine proceeds and completes all of the steps in this action successfully. All of the menu options were present and available for use in this action. The final label clicked was labeled "Create a certificate" which Jasmine directly associated with the effect of certificate generation she is attempting to achieve. Upon completion of all the steps in this action, a window opens and is labeled "Create your certificate". Given this expected behavior, this result indicates progress to Jasmine.

Action 4: Fill in the name textbox, choose self-signed root and select the certificate type "S/MIME (email)", and then click the button labeled "Create".

Response: As stated previously, the instructions provided by Keychain Access's Help documentation constitute the steps that Jasmine takes to perform this action. All textboxes, dropdowns and checkboxes are present and interactive. All labels for each of the action steps adequately described the individual components that Jasmine interacts with during this action. This includes the window title, and the "Create" button label. She believes that both indicate certificate generation would occur upon final step. Progress was made at the end of this action since the self-signed certificate was shown to be successfully created. An entry in the Keychain Access application under my certificates was generated.

C. Application Configuration

Action 5: In order to associate the certificate to the email account Jasmine employs in the OS X Mail application, she must assign an identity preference. Therefore this action consists of opening Keychain Access to click on the File menu, then click "New identity preference". This should bring up a new window, selecting self-signed certificate and click the option to add encryption.

Response: From previous knowledge, Jasmine will know this action is necessary to perform. Through familiarity with the operating system she precisely and successfully navigates the menu buttons and options to complete this action. The File option on the main menu at the top of the screen, the menu item, textbox and "Add" button are all enabled, descriptively labeled and clearly visible. From prior knowledge that the OS X Mail application will not automatically associate a self-signed certificate (or any imported certificate for that matter) to an email address. Jasmine acknowledges the necessity of this action, and notes that its success contributes to the overall goal she is attempting to complete. After verifying these details and closing the Keychain Access application, Jasmine does not see the results of this action immediately. She must open the OS X Mail application and begin to compose a message before the consequences of this action is made known.

D. Data Transfer

Action 6: The results of the last action completed are the addition of the encryption and digital signing buttons to the chrome of the OS X Mail message composition window on the top right hand side. In order for the identity preferences to be loaded within the operating system, the current user profile must be reloaded. Steps for this action include a logout and login sequence, open the OS X Mail application by clicking on the dock icon, click compose message icon button. The composition of a digitally signed message using the self-signed S/MIME certificate is the next step within this action.

Persona Critique: Even though her end goal is to encrypt a message requesting help from her brother, Jasmine knows it cannot be completed at this action. Had she attempted to click the lock icon button beside the digital signature button, she would receive a message that stated she could not encrypt the message without a valid certificate for the recipient.

Response: After completing the login and logout sequence, Jasmine opens the OS X Mail application and prepares to compose a digitally signed message to her brother. Jasmine types in Jacob's email address into the textbox labeled "To:". She adds an innocuous subject line, and some text requesting a digitally signed email from her brother within the body. In order to digitally sign this message she clicks the digital signature button icon and ensures the icon is an envelope seal with a checkmark superimposed on it. The two buttons for encryption and digitally signing messages appear on the right side of the chrome with icons and rollover text descriptions. They are visible and clickable, but unobtrusive. The icon that shows a seal on an envelope much like sealing wax (See Fig. 5) is an excellent way to represent digital signatures. However, this may create conflicting terminology when describing how to digitally sign a message by clicking on the "Seal" button. The reason for this is that, as described previously, to seal a message means to encrypt its contents. When the message is sent and digitally signed, a copy goes to the sent mail folder. Upon selecting the message, the preview pane shows the message along with a description that states it was digitally signed with the email account associated certificate. To Jasmine the successful signing and sending of the request message is an appropriate indication that she is progressing.

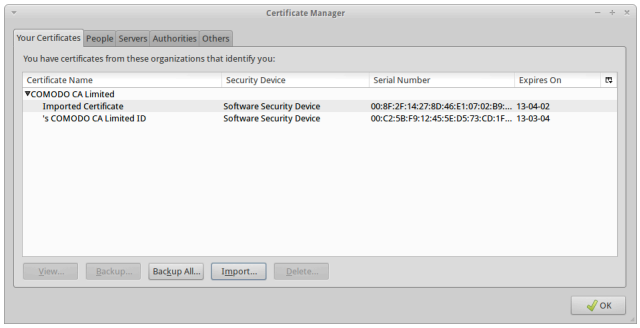

Fig. 6 Import certificate dialog

Action 7: At this point, Jasmine must wait to receive her brother Jacob's digitally signed reply. Once she receives the message, she must somehow verify the public key of his certificate. Next, Jacob must somehow verify the public key of Jasmine's certificate. If using an S/MIME certificate issued by a certification authority, the key can be verified by checking with the Certificate Authority (CA). The problem arises in our situation due to the fact that Jasmine must use her own self-signed certificate. Assuming Jacob has generated his own self-signed certificate, the usual method Jasmine would employ to verify the public key requires an out-of-band means. The interface steps for this action are outlined in Appendix C.1.

Conclusion: According to our scenario's assumption, neither Jasmine nor Jacob could communicate to one another in a secure manner to verify the public keys of their self- signed certificates. Neither user could securely accomplish all of the necessary steps to complete this action successfully. Both Jasmine and Jacob know that that this step needs to be done. In our story they choose not to make the assumption that they are communicating with one another. They could assume they were really communicating, and "verify" without actually performing the secure verification steps. If they did this, they would be susceptible to deception, by a fake certificate posing as either party to the communication. At this point, neither Jasmine nor Jacob could successfully digitally sign or encrypt messages to one another without encountering a warning, and being denied the ability to send the message by the mail program.

VII. Scenario 4: The Graduate Student

Early one morning, a computer science graduate student has just arrived home after attending an underground party. She has brilliantly come up with a novel method of encrypting email that she believes is ultimately user friendly, and is very likely to be adopted for everyday use by the general population. She believes her priority at this time is to securely email this idea to her academic supervisor. She sighs in frustration, at the prospect of knowing she has to use the current "broken" method of encryption to accomplish this. She knows that if the idea is not secured properly it would be "scooped" like a news story by one of her academic peers. It would probably be published prior to a final written paper authored by her, as she knows that she takes a long time to write papers. Relatively speaking, Cameron knows what her peers are focused on for the moment, and therefore does not consider the communication of this message to be urgent. Our story begins with Cameron attempting to send this message right away from her home computer.

A. Scenario Setup

Goal: To encrypt and send an email with the contents of a novel idea to an academic supervisor using Comodo / InstantSSL issued S/MIME certificates.

Knowledge Level: Cameron is an expert technical user. She is familiar with all aspects and configuration of her operating system (Xubuntu), the two MUAs (Thunderbird and Evolution) she uses on a regular basis, and her browser (Firefox).

Assumptions: Cameron is responsible for the management of two systems. Her home machine runs Xubuntu with Evolution as her mail client. Her school machine also runs Xubuntu, but uses Thunderbird as the MUA. She retains two email accounts, one of which is for personal use, and the other is for official academic communication. For both of these she has valid S/MIME certificates issued from Comodo / InstantSSL. Both of which have just been installed to her home Firefox browser. All normal operating system steps such as opening or closing applications are trivial to this user, so not included within the action sequence.

B. Certificate Generation

Action 1: Export the certificate for the home email address from the Firefox certificate management screen to a PKCS#12 [1] file. A collection of interface steps for this action are outlined in Appendix D.1.

Persona Critique: On the bottom of the certificate manager window, there are two highly visible and clickable buttons labeled "Backup" and "Backup all". Since Cameron is an expert user, she reasonably assumes clicking these buttons will result in a process that would be equivalent to the buttons being labeled "Export" and "Export all". She thinks it would be a better idea to keep all the terminology the same, (i.e. renaming these to export). That would make this action easily

associated with the "Import" actions on the certificate managers of other applications.

Response: Since Cameron is an expert in email encryption, she knows the certificates from Comodo were automatically installed to her browser. The menus, menu options, tabs, and selections associated with the steps of this action are familiar to her. All of the steps are completed successfully up to the point where she has to select the certificate itself. At this point both the certificate for her school email address and the certificate for her home email address appear in the Firefox certificate manager. In the certificate manager, groupings of certificates are maintained. In this case, both Comodo / InstantSSL certificates are grouped under the label "Comodo CA Limited". The first certificate installed is named "'s Comodo CA Limited ID". Since the second certificate's name is indistinguishable from the first, when it is installed, it is renamed to "Imported Certificate" (See Figure 6). Due to this renaming, and the data within the columns available (See Appendix D.2 for column headings.) within the Firefox Certificate Manager, Cameron is unable to discern the correct certificate to choose from this dialog to export. This leads to the next additional action within the action sequence as follows.

Action 2: Discern the correct certificate to export by looking at the certificate details. To accomplish this in the Firefox Certificate Manager, select the certificate to highlight it, click view, then click the details tab. For Comodo / InstantSSL issued certificates, the email address will be included in the fields Subject or Certificate Subject altName. Note this step is completely unnecessary for the certificates from TrustCenter, since the common name shows on the main detail table.

Response: Cameron does try to achieve the correct effect for this action. As she is an expert user familiar with how to highlight items in the list, she then clicks the "View" button and peruses the certificate information. During the scan she notes the email address associated with the correct certificate and can now proceed from where the first action finished. The table, the "View" button and the popup window are all appropriately visible to Cameron according to standard operating system procedures. As presented, all steps in this action were sufficiently labeled and indicated the correct effect would be associated with their interaction. At the conclusion of this action, Cameron would gain the knowledge of which certificate is the correct one to export. Even if the certificate details that she looks at were associated with her school email account, she only has two certificates stored in the Certificate manager. By a deductive process, she would then know that the other certificate was the one to be exported.

Action 3: Successfully export the certificate associated with the home email, and save to an appropriately named file. Repeat this process for the school email certificate. The operating system procedure includes saving both files to a memory location to be accessed later (a share, or usb key). Appendix D.3 lists the detailed steps for this action.

Response: With the knowledge gained from the previous action, Cameron can now select and successfully export both of the certificates from the Firefox Certificate Manager. She notices that all steps become available to her throughout the steps in this action. Familiarity with buttons, file save dialogs and alert message popups is due to Cameron's extensive use of a variety of operating systems. The appearance of the file save dialog window indicates to Cameron that she was correct in her assumption regarding the process behind clicking the "Backup" button. The appearance of the alert popup and the existence of the two PKCS12 certificate files within the file browser, indicate to Cameron that they have been exported. Since Cameron is an expert user she knows that they now have to be imported into her MUA.

Action 4: Import the certificate associated with the home email account into Evolution's certificate manager. Detailed action steps are presented in Appendix D.4.

Response: Since Cameron is an expert user, she is able to navigate menus, highlight, and select icons. Due to familiarity with certificate management practices and personal experience, she knows that certificates associated with her email addresses will be found upon clicking the "Your certificates" tab. Again, familiarity with the concept of importing files and the operating system, she clicks the "Import…" button to open the file dialog. Cameron then selects the file and clicks open and completes the two password dialog popups she is presented with. All of the steps in this action are visible to the Cameron upon appropriate selection and inspection. According to Cameron, they also employ adequate labels that enable her to understand their use. Upon successful completion of these steps, Cameron can now see the certificates in the "Your certificates" pane of the Evolution preferences Certificates tab. This indicates to her that each certificate can now be associated to the appropriate email account for use by Evolution in order to digitally sign and encrypt messages.

C. Application Configuration

Action 5: Associate the appropriate S/MIME certificate with the home email address account in Evolution's preferences for digitally signing messages. Individual steps are noted in Appendix D.5.

Response: Due to Cameron's expert knowledge, she is able to navigate the edit menu, and click on preferences. She then highlights the mail accounts icon, and selects her home email address account. The presentation of these two items in a list format indicates that she should highlight the account to proceed. She only does this from extensive familiarity with the operating system. Cameron then clicks the "Edit" button, and the security tab in the new window that is presented. She then reads the copious amount of information retained within the security pane window. Cameron navigates to the Secure S/MIME section and clicks on the "Select..." button. After which a window becomes present, entitled "Select Certificate". This window includes a selection dropdown box and a details textbox that displays the data for the currently selected certificate. Cameron ensures the certificate details match the correct email address from this given information. She then clicks the "Ok" button to proceed. Other than the size of the select certificate popup window, which Cameron had to resize for suitable viewing, all of the action steps were both visible and made available to her. The labels and placement of all the interface components in this action adequately describe and indicate the next step in the action. Upon completing all of the steps in this action, Cameron notices that the textbox shows the certificate that will be used by the Evolution email

account. Since she is an expert, this is sufficient feedback to indicate her progress toward the overall goal of sending an encrypted email.

D. Data Transfer

Action 6: Associate the appropriate S/MIME certificate with the home email address account in Evolution's preferences for encrypting messages. The detailed steps for this action begin at the current position Cameron maintains after the last action is complete. This action commences at the Security tab of the mail accounts Evolution preferences pane. These individual steps are noted in Appendix D.6.

Response: Due to this action's similarity to the last action (Action 5), Cameron accomplishes these steps easily. In order to proceed to the next action in the sequence, the window of the preferences pane must be closed, and this step is trivially accomplished. All of the steps in this action if not present within the current window are made available in a similar fashion to the progression of the last action. For this and all of the previous actions, Cameron recognizes that these actions and steps are fundamental to the completion of the overall goal. Another normal end user unfamiliar with the concepts that Cameron is, would not be able to recognize the progression. The reason for this is the inherent complexity or number of steps in an action, and a lack of knowledge about the cryptographic protocol that must be adhered to. Cameron populates the textbox labeled "Encryption certificate:" by completing all of the steps in this action. To her, this is enough of an indication for progression at this point.

Action 7: At this point, a digitally signed message must be sent to the recipient in order to distribute the issued certificate information. The contents of this message will request the equivalent information from her professor in order to enable future communications to be encrypted. Details of the steps for this action are in Appendix D.7.

Response: Steps within this action are commonly associated with an everyday task that Cameron is used to performing. That everyday task is composition and sending of emails. Opening the "Compose Message" window via the menu, button or the shortcut keys is trivially done by Cameron. After entering the professor's email address in the "To:" field, she enters the text "Certificate Request and Digitally Signed Message" in the subject line field. She composes a small note to her professor requesting a copy of his S/MIME certificate and types this into the email body section of the window. At this point, due to familiarity with Evolution and the mechanism of how the menu works, Cameron clicks the Option menu and selects the S/MIME sign option. Afterwards she reviews the Option menu to ensure that there exists a checkmark beside the options she clicked. She then clicks the "Send" button. Since she is familiar with Evolution and the regular use of email functionality, all of the menus, menu options, buttons and text entry locations are apparent and readily available to Cameron. For the overall goal, Cameron realizes this is the request for the final data necessary to encrypt and send the novel idea to her supervisor. Upon clicking the "Send" button, the composed email shows up in the sent mail folder associated with the email account on Evolution's sidebar. If Cameron would like to verify her message was digitally signed after it is sent, she can preview the message, and a green bar should appear in the body at the bottom of the window. When viewed this indicates progress towards the end goal for Cameron.

Action 8: Receive a digitally signed response and integrate the S/MIME certificate from the digital signature.

Response: In Evolution, the integration of email certificates Cameron will receive has been poorly implemented in the background. This program offers no method to validate self-signed certificates, nor does it check to see if there exists a certificate to the recipient by validating the address entered in a composition's "To:" textbox. This indicates an error in the implementation, since an encrypted email may be sent encrypted using an imported certificate and received certificate to an incorrect recipient. Even though the message contents are encrypted and cannot be decrypted by the unintended recipient; this is still a security risk, since it shows the intended recipient and the sender's information in the email headers. The options to accomplish this action are not made available to Cameron within the Evolution application. Even if she receives the email from her professor, there exists no method to configure the incoming certificate options. The problem here is compounded by the glaring programming error. Since the configuration options do not exist, they cannot be associated with the successful completion of this action. Should Cameron have received the email, no indication would have been offered to her that the certificate had been installed, until she tries to send an encrypted email to the recipient. This would be made apparent two actions later, and not readily apparent to any end user.

Story Continuation: Cameron's professor has a significant large email inbox problem. He has incorporated a whitelist to filter out unknown sourced email and as such, Cameron's home email address was not added to this list. Her professor therefore did not receive her request email. Following this, she could not receive the intended digitally signed reply. Since the time is late, she decides to try again the next day from her school email account at school.

Action 9: Import S/MIME certificates into certificate manager on Thunderbird and associate to appropriate email accounts. Appendix D.8 is the location of step details for this action.

Response: Since Cameron is an expert and already familiar with operating system concepts, her extensive use of MUAs and encryption, she readily performs all action steps with ease. All options are available and presented to the user in the normal and expected fashion. For each step the menus, menu items, ability to expand list items, buttons, password dialog and alert windows are centrally located and accessible. At each step, all of the buttons, popup windows and interaction mechanisms are appropriately labeled. In addition, sufficiently detailed text was also provided to indicate the action would enable the certificate to be properly associated with the email account. There exist two indicators to Cameron that this action is successful. The first is that the certificates show up as list entries in the certificate manager. The second indicator is that both text fields under the descriptions for digital signing and encryption are populated. Cameron realizes this is a necessary preliminary step to the overall task and considers it progress.

Action 10: Compose and send a digitally signed email in Thunderbird using S/MIME certificates. For action details, see Appendix D.9.

Fig. 7 Thunderbird message security icons

Response: Writing an email is a task Cameron is very familiar with. She fills in the appropriate fields in the new message window. She notes that the ability to sign the message with two clicks offers a huge usability advantage in comparison to other MUAs. Since Cameron completed all of Action 9, the option to digitally sign the message was made available for her use. Cameron knows that both parties to an encrypted communication session must share their certificates with one another apriori. In performing this action, she is initiating the first part of the protocol required to eventually encrypt a message to her professor which is her overall goal. After pressing the "Get mail" button repeatedly for half an hour, Cameron receives a digitally signed email from her supervisor. After receipt of the email, Thunderbird automatically imports the certificate information into its certificate store. Cameron now has the capability to generate and send an encrypted message to her supervisor.

Action 11: Compose and send an encrypted email in Thunderbird using S/MIME certificates. Appendix D.10 has all the details.

Response: Simply put, Cameron performs all of the steps she performed to digitally sign the last message she sent, with one key difference. She selects to encrypt rather than digitally sign this message. From past experience, when she digitally signed the message, she assumes that the encrypt option would be made available within the same security dropdown. Her assumption was valid, since upon receipt of her supervisors' email, the option to encrypt this message became available to her. The dropdown is clearly labeled security, and the option to encrypt uses language that is self-explanatory. As before, and during normal mailing operation, the sent message will appear in the sent folder under Cameron's email account. Cameron double checks this to ensure that the email was encrypted as there was no other indication provided to her to verify its encryption status. She opens the message and can clearly see the digital signature and encryption lock icons on the chrome of the message window. The completion of this action signifies the successful completion of the overall goal Cameron was attempting to achieve. No one scoops Cameron's story, and she goes on to write a very well received journal paper on the topic of email encryption.

VIII. Inconsistencies

One consistent pattern we noticed while performing our walkthroughs was the inconsistencies between the applications used, particularly with regards to terminology. Some of these were simple, such as the difference between a digital signature and an email signature. An email signature is a block of text appended to the end of an email message automatically by the user's mail agent [6]. Whereas, the digital signature of an email utilizes cryptographic mechanisms to verify the authenticity of the sender of that signed email [8].

There were several inconsistencies, though, even within security-relevant terminology and interface elements. For example, the icon images associated with digital signatures include an envelope sealed with wax while a lock is used to indicate encryption (See Fig. 7). In cryptographic terms, what it means to "seal" or the "sealing" of a message translates to the encryption and signing of the message contents, not just signing them. It is thus possible for a user to think a message is protected against eavesdropping with a sealed envelope indicator when, in fact, the message has just been protected against tampering.

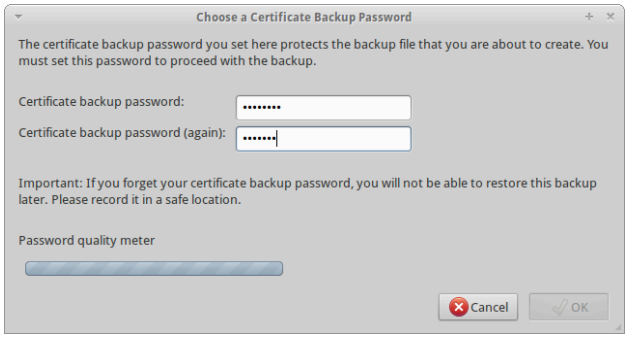

Fig. 8 Firefox certificate backup dialog

Some inconsistencies are clearly due to tools being used for unintended purposes. For example, instructions for end users on the certificate issuer website, and within application certificate managers use the term "Backup" to save the certificate data to a file (See Fig. 8). Yet when integrating the same information from that file into any certificate manager including the originating application, the term used is "Import" (See Fig. 6). Note here the implication of the term "backup"—downloaded certificates should only be extracted from the certificate store for backup purposes rather than being used in a different program.

Altogether, these inconsistencies betray the fact that many of the usability problems with S/MIME are largely due to neglect, with email security not being a high-priority design goal, rather than any inherent complexity in the underlying task.

IX. Discussion

As can be seen in our walkthroughs, usability problems with S/MIME are not limited to a single application; instead, problems arise in interactions with web browsers, email clients, and (sometimes) OS-specific certificate management applications. Each of these programs has different approaches to handling certificates that make sense in their own context; when combined, however, the result is an unusable mess. Specifically, users can be stymied at multiple points, including obtaining private certificates, moving private certificates into the email client, obtaining public certificates from correspondents, verifying potentially untrusted certificates, choosing the appropriate certificate for encryption and signature operations, and in understanding inconsistent terminology for signed and encrypted messages.

We believe the central issue here is that the one type of object—the digital certificate—is being used for many different purposes. Web browsers import certificates from remote websites primarily because SSL/TLS allows users to authenticate users to remote websites using locally stored certificates. Operating systems manage certificates because they are used to verify application integrity and the identity of remote servers, particularly those providing software updates. With S/MIME, these same certificates are used to secure email messages. Interfaces designed to manage certificates for one purpose are, as we have shown, inappropriate for other purposes.

To improve S/MIME usability, we surely need certificate management tools that are better specialized to the use cases of sending and receiving secured email. Key managers such as CoPilot for S/MIME can certainly help [2]. However, the fact is that multiple developers keep creating unusable S/MIME implementations. Perhaps better solutions will have to start with terminology more than technology: so long as we are using the same terms to discuss securely sending a message to a lover as to verify the integrity of a software update, developers will continue to provide interfaces designed for one goal for other goals, inevitably confusing users in the process. We think this is a promising area for future research in the usability of S/MIME and related secure communications technologies.

X. Conclusion

We examined the usability of three modern S/MIME implementations using cognitive walkthroughs. We found numerous usability issues around the use of multiple non-integrated applications and inconsistent terminology arising from the different uses of certificates. From issuance to the delivery of an encrypted email, only the scenario in which the two ends of the communication are expert users could all of the actions within the sequence be completed. Two scenarios did not proceed past the initial certificate generation stage. Even with the expert users, the number of steps required and the potential for multiple errors suggest that even experts could easily decide the effort was not worth it. Although seemingly more user friendly, self-signed certificates come with significant security risks and their own usability issues. A great deal of basic usability work is required in order to make S/MIME usable in email clients, even for users who are well versed in the concepts of public key cryptography. Such work will need to be done before we can even begin to educate users in the subtle semantics of digital certificates and public key cryptography as they are used for code, data, and communications security.

References

[1] *PKCS#12: Personal Information Exchange Syntax Standard*. ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf, June 1999.

[2] W. Diffie and M. Hellman. “New directions in cryptography.” *Information Theory*, IEEE Transactions on, 22(6):644–654, 1976.

[3] S. Farrell. “Why don’t we encrypt our email?” *Internet Computing*, IEEE, 13(1):82–85, 2009.

[4] Simson L. Garfinkel and Robert C. Miller. “Johnny 2: a user test of key continuity management with S/MIME and Outlook Express”. In Proceedings of the *2005 Symposium On Usable Privacy and Security*, SOUPS ’05, pages 13–24, New York, NY, USA, 2005. ACM.

[5] S.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.

[6] S. Hambridge. Rfc 1855: *Netiquette guidelines*. 1995. N.E. Jacobsen and B.E. John. “Two case studies in using cognitive walkthrough for interface evaluation.” Technical report, DTIC Document, 2000.

[7] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC, August 1997.

[8] J.B. Postel and S.M.L.T. Protoco. Rfc 821. Simple Mail Transfer Protocol,1982.

[9] B. Ramsdell. Rfc 3851: *Secure. Multipurpose Internet Mail Extensions* (S/MIME) Version, 3, 2004.

Appendix A: Scenario 1 Interface Description

1. TrustCenter Web form 1 Details

The following are a list of the details requested by the first web form in the second action:

1. Salutation*
2. Title
3. Name* and Surname*
4. Street* and Number*
5. Zip Code* and Location*
6. State or Province
7. Country*
8. Email Address*
9. Phone Number*

Mandatory fields are marked with an *

2. TrustCenter Web form 2 Details

The following are a list of the details requested by the second web form in the third action:

1. Revocation Password*
2. Confirm Revocation Password*
3. Delivery choice to Email / SMS*
4. Acceptance of General Terms and Conditions*
5. Data Handling*

Mandatory fields are marked with an *. The data handling description states the user must accept that both the public key of the certificate and all of its data fields are published to the TrustCenter’s public key server.

Appendix B: Scenario 2 Interface Description

The following are a list of the details requested by the first web form in the second action:

1. Enter name*
2. Enter email*
3. Country
4. Select Key Size dropdown box*
5. Revocation Password*
6. Confirm Revocation Password*
7. Checkbox to agree to subscriber agreement*

Mandatory fields are marked with an *

Appendix C: Scenario 3 Interface Description

The following outlines the steps required to validate a self-signed certificate within the OS X Mail application:

1. Highlight digitally signed email for it to be shown in the mail preview pane
2. Click the "Show Details" button along the top of the body pane in the yellow bar
3. Click the "Show Certificate" button
4. Under details, cross check the public key values with values provided by the sender obtained through an out- of-band channel
5. Once verified, click the checkbox beside "messages from jasminemail@evilcountry.com" are valid if signed by Jasmine

APPENDIX D: SCENARIO 4 INTERFACE DESCRIPTION

1. Initial steps to export a certificate from the Firefox browser

In Firefox, click or select:

1. Edit Menu Title
2. Preferences Menu Option
3. Advanced Icon
4. Encryption Tab
5. "View Certificates" Button
6. "Your Certificates" Tab
7. Highlight the Certificate

2. Certificate data columns shown in Firefox Certificate Manager

1. Certificate Name*
2. Security Device*
3. Serial Number*
4. Expires On*
5. Issued On

Items marked with an * are shown by default.

3. Successfully export a certificate from Firefox

1. Highlight the Certificate
2. Click Backup
3. Use File Dialog to choose a location to save
4. Create and enter a filename to dialog textbox
5. Press enter, or click "Save" button
6. Enter a password in textbox labeled "Certificate backup password:"
7. Confirm the same password in textbox labeled Certificate backup password (again):"
8. Click the "Ok" button

4. Import certificate into Evolution's certificate manager

In Evolution, click or select:

1. Edit Menu Title
2. Preferences Menu Option
3. Highlight Certificates icon on the left slider menu
4. "Your Certificates" Tab
5. Import (which will open a file dialog)
6. Navigate file dialog to select the certificate file
7. "Open" Button
8. A popup appears if no certificates are stored. It requests a new password for the certificate database. Enter password in textbox.
9. "Ok" Button
10. A popup appears requesting the password for the certificate file.
11. "Ok" Button

5. Associate certificate to Email account in Evolution for Digitally Signing emails

In Evolution, click or select:

1. Edit Menu Title
2. Preferences Menu Option
3. Mail Accounts icon on left slider bar
4. Highlight Email addressaccount to associate the certificate with
5. "Edit" button
6. Security tab
7. Under Secure S/MIME, beside textbox labeled "Signing certificate:", click "Select..." button
8. Ensure certificate details specify the correct email address (Otherwise select the other imported certificate using the dropdown button on the top of the popup window.)
9. "Ok" button

6. Associate certificate to Email account in Evolution for Encrypting email

In Evolution, click or select:

1. Under Secure S/MIME, beside textbox labeled "Encryption certificate:", click Select button
2. Ensure certificate details specify the correct email address (Otherwise select the other imported certificate using the dropdown button on the top of the popup window.)
3. "Ok" button
4. "Ok" button
5. Close Evolution preferences dialog window by clicking the X or "Close" button in the top right corner

7. Send digitally signed email in Evolution

In Evolution, click or select:

1. New message icon / press Shift-Ctrl-M / Message menu, new message
2. Enter recipient's email address in "To:" textbox
3. Enter subject in "Subject:" textbox
4. Enter textual information in the body pane
5. Options menu
6. "S/MIME sign" menu item to ensure checkmark appears
7. Send icon (top left and looks like a stamp)

8. Import and associate certificates to email accounts in Thunderbird

In Thunderbird, click or select:

1. Edit menu
2. Account Settings...

3. Arrow beside email address to expand selections associated with the certificate
4. Highlight Security
5. View Certificates button
6. "Your Certificates" tab
7. Import...
8. Navigate the file open dialog popup, and select certificate filename
9. Open button
10. Fill in textbox labeled "Enter the backup password:"
11. Press enter or click ok button
12. ok button on Alert window
13. ok button on certificate manager window
14. Under Digital signing on Account Settings screen, click Select... button
15. Verify the certificate is associated with the right email address by looking at and choosing options from the dropdown, once selected, certificate details are shown in a preview pane below
16. ok button
17. An alert appears asking if you wish to use the same certificate for encryption, click yes button
18. Change the default encryption settings to required using the radio button
19. ok button

9. Send digitally signed email from Thunderbird

In Thunderbird, click or select:

1. Write button / press Ctrl-N / Message menu, new message
2. Enter recipient's email address in "To:" textbox
3. Enter subject in "Subject:" textbox
4. Enter textual information in the body pane
5. Security dropdown
6. Digitally sign this message dropdown item
7. Send icon (top left and looks like an empty picture frame)

10. Send encrypted email from Thunderbird

In Thunderbird, click or select:

1. Write button / press Ctrl-N / Message menu, new message
2. Enter recipient's email address in "To:" textbox
3. Enter subject in "Subject:" textbox
4. Enter textual information in the body pane
5. Security dropdown
6. Encrypt this message dropdown item
7. Send icon (top left and looks like an empty picture frame)