Client Certificates

Security Professionals 2012 Preconference Seminar

8:30-Noon, Tuesday, May 15th, 2012
White River Ballroom B, JW Marriott, Indianapolis IN

Joe St Sauver, Ph.D. (joe@internet2 or joe@uoregon.edu)
InCommon Certificate Program Manager and
Internet2 Nationwide Security Programs Manager

http://pages.uoregon.edu/joe/secprof2012/

Disclaimer: The opinions expressed in this talk represent those of its author, and do not necessarily represent the opinion of any other entity.

Preface

Our Time Together Today

- Since three hours is a relatively long time for a single session, we're going to go through material for about an hour and a half (until about 10:00), and then we'll take a coffee break outside of room 103 for a half hour or so. Around 10:30, we'll crank back up and finish the rest of the material we want to go over.
- If you have any questions at any time, feel free to speak up. While I've prepared a fairly structured session given the number of attendees that are expected, I've still tried to build in time for discussion, and I know that some of you may already be experienced wih client certs and have much to share yourselves.
- Finally, I also want to make sure we've got time to help you actually get a client cert installed and up and running on your system, if you'd like to try doing this.
- Are there any questions at this point?

Introductions

- Let's take a minute or two to go around the room and introduce ourselves.

- Please say:

  -- who you are
  -- what school you're with
  -- anything your site may currently be doing with client certs
  -- why you're interested in client certs/anything you particularly hope we cover today

Strong Cryptography and Federal/International Law

- **Strong cryptography is critical to computer and network security**, including enabling secure authentication and online commerce, protecting personally identifiable information (PII) stored online, and legitimately ensuring personal privacy for law-abiding citizens.
- At the same time, **strong cryptography is subject to complex regulation** in many countries, including the United States. Why? Use of encryption makes it harder for national security agencies and law enforcement organizations to lawfully intercept criminal communications and national-security-related communications.
- Therefore, **our goal when talking about strong cryptography is to always abide by federal laws and international treaties relating to controls over strong cryptography**, and to do what what we can to ensure that strong cryptography doesn't get misused in ways that might either harm our national security or interfere with the lawful investigation and prosecution of criminals.

Since We'll Be Giving You Strong Hardware Crypto Products

- **You warrant that you aren't barred from obtaining and using strong crypto products or software, NOR are you barred from receiving training on it.**
- Specifically, this means that you assert that you are NOT a citizen, national, or resident of Burma, Cuba, Iran, Iraq, North Korea, Sudan, Syria, or any other country blocked from obtaining strong cryptography products.
- You are NOT a "denied person," a "specially designated national," or any similar individual forbidden to access strong cryptography by the US government (www.bis.doc.gov/complianceandenforcement/liststocheck.htm)
- You are neither a terrorist nor a trafficker/user of illegal controlled substances, NOR are you directly or indirectly involved in the design, development, fabrication or use of weapons of mass destruction (including improvised explosive devices, nuclear, chemical, biological, or radiological weapons, nor missile technology, see 18 USC Chapter 113B)
- You agree NOT to redistribute or retransfer cryptographic products or software to anyone who is in one of the previously mentioned prohibited categories.
- You understand and agree that the forgoing is by way of example and is not an exhaustive description of all prohibited entities, and that this is not legal advice. For legal advice relating to strong crypto, please consult your own attorney.

"First, Do No Harm"

- Some of you may want to “follow along” as we go through today’s training materials. If so, that’s terrific. However please ONLY do so if you’ve got a recent backup of your system, and your system (if supplied by your university) is NOT "locked down" by your university IT department.
- If you have NOT backed up your system recently, or your university IT department does NOT want you to tinker with your laptop, please feel free to watch we we go over today but please do not try to install any new software or otherwise modify your system.
- Also, if you already have a client certificate installed on your system, you may want to refrain from installing another one, and in particular **PLEASE do NOT intentionally delete any client certificates you may already have installed on your system!**

Oh, And For Those of You Who May Have Been Worried, No, We're *Not* Going to Dive Into Any Advanced Crypto-Related Mathematics Today

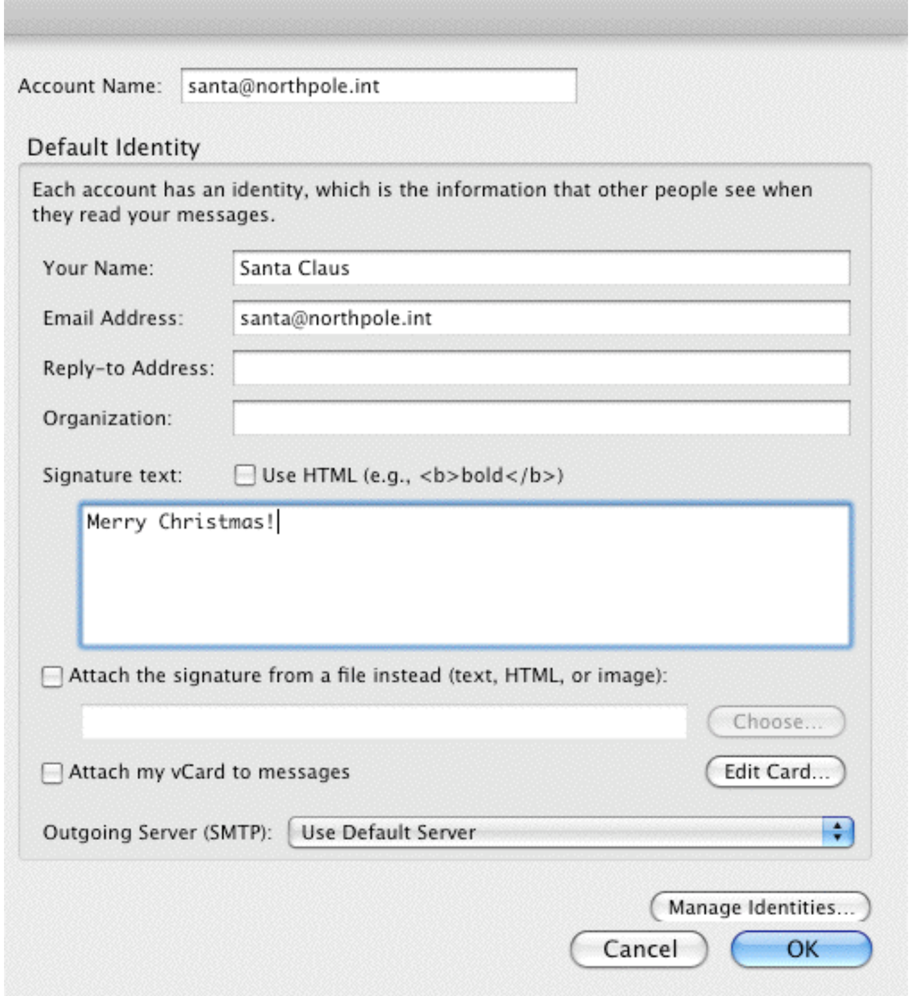
- Our focus today is on helping you get to the point where you can actually use client certificates, particularly for secure email, and getting you to the point where you understand the practical limitations associated with those technologies. You don't need advanced mathematics to do that.
- So if you hated mathematics while going through school, relax. :-) Virtually everything we're going to talk about today should be non-mathematical.
- Let's dive right in. We'll begin by talking about why you might want to use client certificates, particularly for signing and encrypting email.

I. Motivating An Interest in Client Certificates ("PKI"): Securing Email

Why Might We Need To Sign and/or Encrypt Email?

- Put simply, regular email is horribly insecure.
- Email is trivial to **spoof**: even technically unskilled users can simply put bogus identity information into the preferences panel of their email client and voila, they're "Santa" (or pretty much anyone else they want to be). You just can't trust the non-cryptographically-signed contents of email that you may receive – it may all be complete rubbish.
- Most email is also trivial to **sniff** on the wire (or read in the mail spool): messages normally aren't encrypted when transmitted or stored, so unauthorized parties can read your communications. "Trusted insiders" may also access confidential communications.
- Let's take a look at a couple of practical examples of these sort of exposures.

The Simple Road to Spoofing Email: Just Change Your Preferences in Mozilla Thunderbird



Account Name: santa@northpole.int

Default Identity

Each account has an identity, which is the information that other people see when they read your messages.

Your Name: Santa Claus

Email Address: santa@northpole.int

Reply-to Address:

Organization:

Signature text: □ Use HTML (e.g., <b>bold</b>)

Merry Christmas!

□ Attach the signature from a file instead (text, HTML, or image):

□ Attach my vCard to messages

Outgoing Server (SMTP): Use Default Server

[Yes, this will work. But no, please don't actually do this.]

"But Won't SPF and/or DKIM Eliminate the Spoofing Problem?"

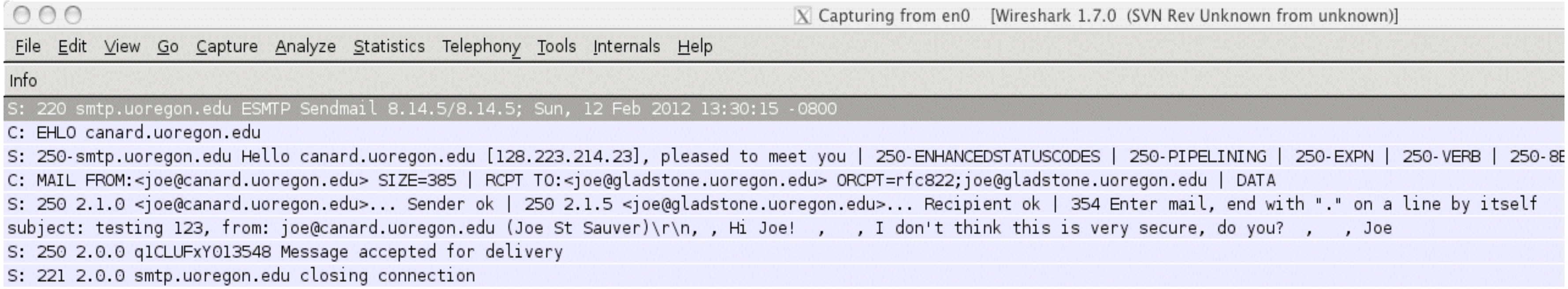
- SPF (www.openspf.org) and DKIM (www.dkim.org) were meant to help fix spoofing, and they do, but they're not a total solution.
- For instance, SPF/DKIM cannot protect you against spoofed email that is injected from an authorized source. Classic example:
  -- College faculty member and her students all have accounts in the same example.edu domain, and all send from "on campus"
  -- A malicious class member forges message from a campus computer lab, pretending to be the faculty member, "cancelling class" or "assigning extra homework" (or whatever). SPF and DKIM aren't designed to defend against this sort of attack.
- Security folks tend to like belt-and-suspender ("defense in depth") solutions anyhow, and just because you're doing SPF or DKIM, that doesn't preclude **also** doing message level crypto, right?

A Simple Example of How Easy It Is To <u>Sniff</u> Typical Plain Text Email Using Wireshark

- Send a simple mail message...

```
% mailx -s "testing 123" joe@gladstone.uoregon.edu
Hi Joe!

I don't think this is very secure, do you?

Joe
.
```

- If someone is using Wireshark to watch your traffic, they'd see:



```
S: 220 smtp.uoregon.edu ESMTP Sendmail 8.14.5/8.14.5; Sun, 12 Feb 2012 13:30:15 -0800
C: EHLO canard.uoregon.edu
S: 250-smtp.uoregon.edu Hello canard.uoregon.edu [128.223.214.23], pleased to meet you | 250-ENHANCEDSTATUSCODES | 250-PIPELINING | 250-EXPN | 250-VERB | 250-8E
C: MAIL FROM:<joe@canard.uoregon.edu> SIZE=385 | RCPT TO:<joe@gladstone.uoregon.edu> ORCPT=rfc822;joe@gladstone.uoregon.edu | DATA
S: 250 2.1.0 <joe@canard.uoregon.edu>... Sender ok | 250 2.1.5 <joe@gladstone.uoregon.edu>... Recipient ok | 354 Enter mail, end with "." on a line by itself
subject: testing 123, from: joe@canard.uoregon.edu (Joe St Sauver)\r\n, , Hi Joe!  ,   , I don't think this is very secure, do you?  ,   , Joe
S: 250 2.0.0 q1CLUFxY013548 Message accepted for delivery
S: 221 2.0.0 smtp.uoregon.edu closing connection
```

"But Joe! All Our Networks Are *Switched Ethernet!* There'd Be No Traffic to Sniff!"

- Sites sometimes have a false sense of security when it comes to their vulnerability to sniffing. Specifically, some may believe that because they use switched ethernet, traffic intended for a given system will ONLY flow to the appropriate system's switch port.
- You may already be aware that many switches can be forced to act like hubs through a variety of well known techniques (see for example http://ettercap.sourceforge.net/). Thus, even if your infrastructure is intended to isolate traffic on a per-port basis, in practice, that process may fail to maintain traffic separation.
- You also can't ensure that traffic won't be sniffed once it leaves your local network.
- Therefore, you should assume that any unencrypted network traffic, including most email, ***can*** be sniffed and read.

Of Course, If Someone's Got Root, They Can Look At Anything On The System, Including Email Messages...

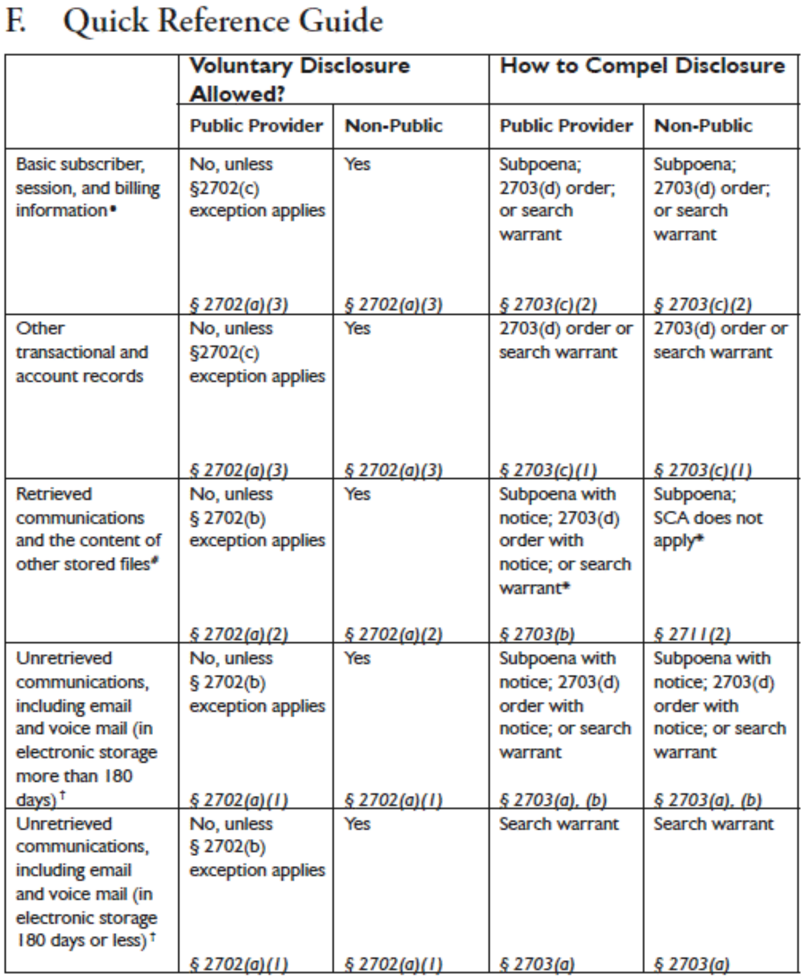
```
% su
Password:
# cat /var/mail/joe
From joe@canard.uoregon.edu  Sun Feb 12 14:30:54 2012
Return-Path: <joe@canard.uoregon.edu>
Received: by canard.uoregon.edu (Postfix, from userid 501)
  id 5C221D537D4; Sun, 12 Feb 2012 14:30:54 -0800 (PST)
To: joe@canard.uoregon.edu
Subject: Some thoughts on the insider threat
Message-Id: <20120212223054.5C221D537D4@canard.uoregon.edu>
Date: Sun, 12 Feb 2012 14:30:54 -0800 (PST)
From: joe@canard.uoregon.edu (Joe St Sauver)
Status: O

Hi Joe,

I wonder if a system admin with root priv could read the mail
that's sitting in my mail spool? You know, I bet s/he could...

Joe
```

BUT If Your Email Is Encrypted, It May Not Matter If Someone Does A Little "Browsing:" The Following Isn't Very Informative, Is It?

MIAGCSqGSIb3DQEHA6CAMIACAQAxggNbMIIBkQIBADB5MGQxCzAJBgNVBAYTAlVTMRIwEAYD
VQQKEwlJbnRlcm5ldDIxETAPBgNVBAsTCEluQ29tbW9uMS4wLAYDVQQDEyVJbkNvbW1vbiBT
dGFuZGFyZCBBc3N1cmFuY2UgQ2xpZW50IENBAhEAowXASR0JSE0KE5HSe8RXCTANBgkqhkiG
9w0BAQEFAASCAQAphc3r5MLFw43hOcMzlb/UG9DEaFPyFtcaiN8koelnok2DVdcAtSb9wulU
iKjw4jps8GwqPeonzC8o+RMyktiFwMvM/QfN4zMUbfxsJr0i7FpnveROp+V8Cyo2hDuJpa/d
GjRI560cDnH2z4tnYOO9/SJBCvLIIRjfnnnuJlS12VF00kcA9sfJI23QWhauisoef0ZhvAOw
11wHi8o+4icSe6iT18rR+Sr9MDhulDdfVCfmYwDfBi4SAqzbLK1FZfSj7aIjphlcFV4JKXr3
HyEz2afYRCGYUUaGk1zjcfhh4Eqkah6TwZ8QCtWUTsYdhuZdHGHw6zbBuSUYxzRG2NiRMIIB
wgIBADCBqTCBkzELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQ
MA4GA1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxOTA3BgNVBAMT
MENPTU9ETyBDbGllbnQgQXV0aGVudGljYXRpb24gYW5kIFNlY3VyZSBFbWFpbCBDQQIRAKgC
OyLlmfFLiBBlWracUfMwDQYJKoZIhvcNAQEBBQAEggEAOc1JpNLx+62m1To69oxFd3/fMEvo
UDkL1nSQe5LDhKnH3DXmH2vvTN0Q0h8vjGbkcGklCD11164VRi380QrtVYTsYCl9tB1kuHam
SH+xJIIsLkNasYWnCXwzji+Uw80GiAP9/CgB/aYJhhYJt1HRQ+43S9m3xgpdK//aCOIjmKLl
prFiQ1Jk5Wx3Sqm/Kkg89m9ulln1ckpIBrvTxNsikZmFwh4QGcCtz42+mTGZXcbrrn9yfT0F
4ds9xDbBm5e/Se/aq4vpfX0yi0/UP8/ywJ5+zG2ufyJw4i2h2O3vyD6WzX7PiYuzsn232RkR

[This base64 encoded file is actually a base64 encoded *encrypted* file]

Email Is Also Potentially Subject to Lawful Intercept and/or Compulsory (or Even *Voluntary*) Disclosure



F. Quick Reference Guide

	Voluntary Disclosure Allowed?		How to Compel Disclosure	
	Public Provider	**Non-Public**	**Public Provider**	**Non-Public**
Basic subscriber, session, and billing information•	No, unless §2702(c) exception applies *§ 2702(a)(3)*	Yes *§ 2702(a)(3)*	Subpoena; 2703(d) order; or search warrant *§ 2703(c)(2)*	Subpoena; 2703(d) order; or search warrant *§ 2703(c)(2)*
Other transactional and account records	No, unless §2702(c) exception applies *§ 2702(a)(3)*	Yes *§ 2702(a)(3)*	2703(d) order or search warrant *§ 2703(c)(1)*	2703(d) order or search warrant *§ 2703(c)(1)*
Retrieved communications and the content of other stored files#	No, unless § 2702(b) exception applies *§ 2702(a)(2)*	Yes *§ 2702(a)(2)*	Subpoena with notice; 2703(d) order with notice; or search warrant* *§ 2703(b)*	Subpoena; SCA does not apply* *§ 2711(2)*
Unretrieved communications, including email and voice mail (in electronic storage more than 180 days)†	No, unless § 2702(b) exception applies *§ 2702(a)(1)*	Yes *§ 2702(a)(1)*	Subpoena with notice; 2703(d) order with notice; or search warrant *§ 2703(a), (b)*	Subpoena with notice; 2703(d) order with notice; or search warrant *§ 2703(a), (b)*
Unretrieved communications, including email and voice mail (in electronic storage 180 days or less)†	No, unless § 2702(b) exception applies *§ 2702(a)(1)*	Yes *§ 2702(a)(1)*	Search warrant *§ 2703(a)*	Search warrant *§ 2703(a)*

http://www.cybercrime.gov/ssmanual/ssmanual2009.pdf at page 138

Reducing The Transport Email Sniffing Vulnerability: Opportunistic SSL/TLS Encryption

- You can reduce the extent to which email traffic is subject to sniffing on the wire by enabling opportunistic SSL/TLS encryption. This means that if the MTAs on both sides of the conversation are ready and willing to do SSL/TLS encryption, it will be negotiated and used whenever it can be. See for example:

  http://www.**exim**.org/exim-html-3.20/doc/html/spec_38.html
  http://www.**postfix**.org/TLS_README.html
  http://www.**sendmail**.org/~ca/email/starttls.html

- However, SSL/TLS will **not** protect email over links that don't have TLS/SSL enabled, nor does it protect **stored mail** once it has been received and saved to disk at its destination. That is, it is not "end-to-end."

Obtaining *End-to-End* Protection Requires *Message-Level* Signing and Encryption
E.G., Use of PGP/GPG, or Use of S/MIME

- There are two basic approaches to getting end-to-end protection for email messages:
  - Pretty Good Privacy (PGP) (or GNU Privacy Guard (GPG)), see RFC4880, *OR*
  - S/MIME (RFC5751) with personal certificates.

- PGP/GPG is probably the more common of those two options, and one that many of you may already use, but today we're going to talk about using S/MIME with client certificates, instead.

- Before we can dig in, however, we need a little "crypto backfill"

II. A Miniscule Little Bit of Cryptographic Backfill

Public Key Cryptography

- There are basically two types of cryptography: symmetric key crypto, and public key (asymmetric) crypto.
- In symmetric key cryptography, a message gets encrypted AND decrypted using the *same* secret key. That means that before you can share a secret message with someone, you need a secret key you've both previously agreed upon (chicken, meet egg).
- Both PGP/GPG and S/MIME with personal certificates, on the other hand, rely on public key cryptography to sign or encrypt messages. In public key cryptography, the user creates a *pair* of mathematically-related cryptographic keys: one private key that only the user knows, plus a related public key that can be freely shared with anyone who's interested. Having a user's public key doesn't allow you to derive that user's corresponding private key, but it does allow you to create an encrypted message for that user via a "one way" or "trap door" mathematical process.

But Wait, There's More! Public Key Cryptography Can Slice, Dice and Make Julienne Fries, Too...

- Well, that may be a *slight* exaggeration.

- But public key cryptography *does* allow you to do at least one more cool trick: the holder of the private key can also digitally sign a file with their private key. Once that file is digitally signed:

  -- it can't be changed without invalidating the message signature
  (e.g., it acts as an anti-tampering checksum value)

  -- anyone who has a copy of the corresponding public key
  can verify that it was signed by someone who had access to
  the corresponding private key

How Do Certificates Fit Into All This?

- So far we've only been talking about public keys and private keys. You may wonder how certificates fit into all this.

- The answer is that certificates attach an identity to a cryptographic keypair.

- If you're like most folks, when you hear "certificates" in an online context, you think of SSL web server certificates. That's not what we're going to be talking about today. Those certificates are issued to servers. The certs we're going to talk about today get issued to *people*, instead.

- But first, let's begin with something we're all familiar with: meeting a new person in real life.

Mapping Users to Identities In "Real Life"

- If I meet you face-to-face, perhaps at the hotel bar, you might tell me, "Hi, I'm Robert Jones. Nice to meet you!" In a casual context at a social event of that sort, we might smile, shake hands, exchange cards, engage in some chit chat, and leave it at that – it doesn't really matter if you are (or aren't) who you claim to be. I'll just temporarily accept (and then unfortunately probably quickly forget) your "self-asserted identity." That's OK.
- If it turns out that I eventually need confirmation of who you are, I might ask trusted colleagues, "Hey, see that guy over there? Who is he?" If they all say, "Oh, that's Robert Jones. I've known *him* for *years*," that might give me confidence that you really are him.
- Other times, for example if you're in a strange city, or someone's trusting you with a valuable asset (such as a rental car), you might need to show a drivers license or other government issued ID since no one "knows your name." (ObCheers: "Norm!")

Mapping Users To Identities *Online: PGP/GPG*

- A similar problem exists online. How do you know which publicly offered PGP/GPG keys is the real one that a person's actually using, and not a pretender's credentials? In PGP/GPG, this is done via a "web of trust."
- In PGP/GPG, a PGP/GPG public key gets digitally signed by other PGP/GPG users who have personally confirmed that person's ID. (This often gets done at PGP/GPG "key signing parties," like the one that will happen at 6:30PM on Wednesday night). Normally a keyholder will get signatures from multiple friends or colleagues.
- Recursively, how do you know that you should trust *those* signatures? Well, *those* signatures were made with keys that have ALSO been signed by other colleagues, and so on and so forth.
- While this sounds incredibly *ad hoc* and kludgy, in practice, it actually works pretty well (at least for technical users) – it really is a small world out there, "six degrees of Kevin Bacon"-wise.

The Web of Trust Is For *Keys* (Not Necessarily Their *Owners*)

- An important note about the cryptographic "web of trust:"

  > Someone signing a PGP/GPG key is *not* saying that that *person* who's key they've signed is a "trustworthy" person.

  Completely evil people may have well-signed PGP/GPG keys!

- When some signs another person's PGP/PGP key, they're only saying that:

  -- they've looked at that person's government issued ID,
  -- that person indicated that that that public key is theirs.

  That is, they're binding an *identity* to a *cryptographic credential*.

Personal Certificates

- In the case of S/MIME with personal certificates, a web of trust isn't used. In the S/MIME case, trust gets established <u>hierarchically</u> ("top down").

- That is, a personal certificate is trusted because it has been issued by a broadly accepted certificate authority ("CA"), an entity that you (and most other Internet users) accept as reliable for the purpose of binding identities to credentials.

- CAs tend to be very careful when it comes to doing what they say they're going to do (specifically, very careful to do what they say they're going to do in their "Certificate Practices Statement"), because if they don't, people (including browser vendors and the CAB Forum) will stop trusting them and then they'll quickly be totally out of business (literally).

'So What's this "CAB Forum?"'

- No, it's *not* a taxicab association.
- The Certificate and Browser Forum is an influential body made up of Certificate Authorities (that's the "CA" in their name) and Browser Vendors (that's the "B" in their name).
- Their website is http://www.cabforum.org
- As a practical matter, increasingly they're effectively establishing the practices/norms that apply to the entire certificate industry, and FWIW, they're making the ship far more shipshape. :-)
- Previously, various industry groups, such as the Mozilla Foundation, had a lot to do with what was or wasn't acceptable: put simply, if you wanted your certificates to be trusted in Firefox, you complied with what the Mozilla Foundation required. Ditto for Internet Explorer and Microsoft, etc.

"What Does a CPS Actually Look Like?"

- CPS documents as a class are probably one of the most widely ignored categories of documents in the world.
- Howver, sometimes folks who have a hard time sleeping actually want to read Certificate Practices Statements. If you'd like to check some out, you can see, for example, InCommon's Certificate Service CPS: https://www.incommon.org/cert/repository/
- You'll see separate CPS for the InCommon standard SSL certificate offering, the extended validation certificate offering, the client certificate offering, and the code signing certificate offering. The various "profile" documents are also potentially quite informative.
- Similar documents should be available for any public certificate issuer.
- One of the things they cover is how identity gets validated, and what expectations should be for a particular type of cert.

III. Identities and Levels of Assurance

A Real Name, or Just An Email Address?

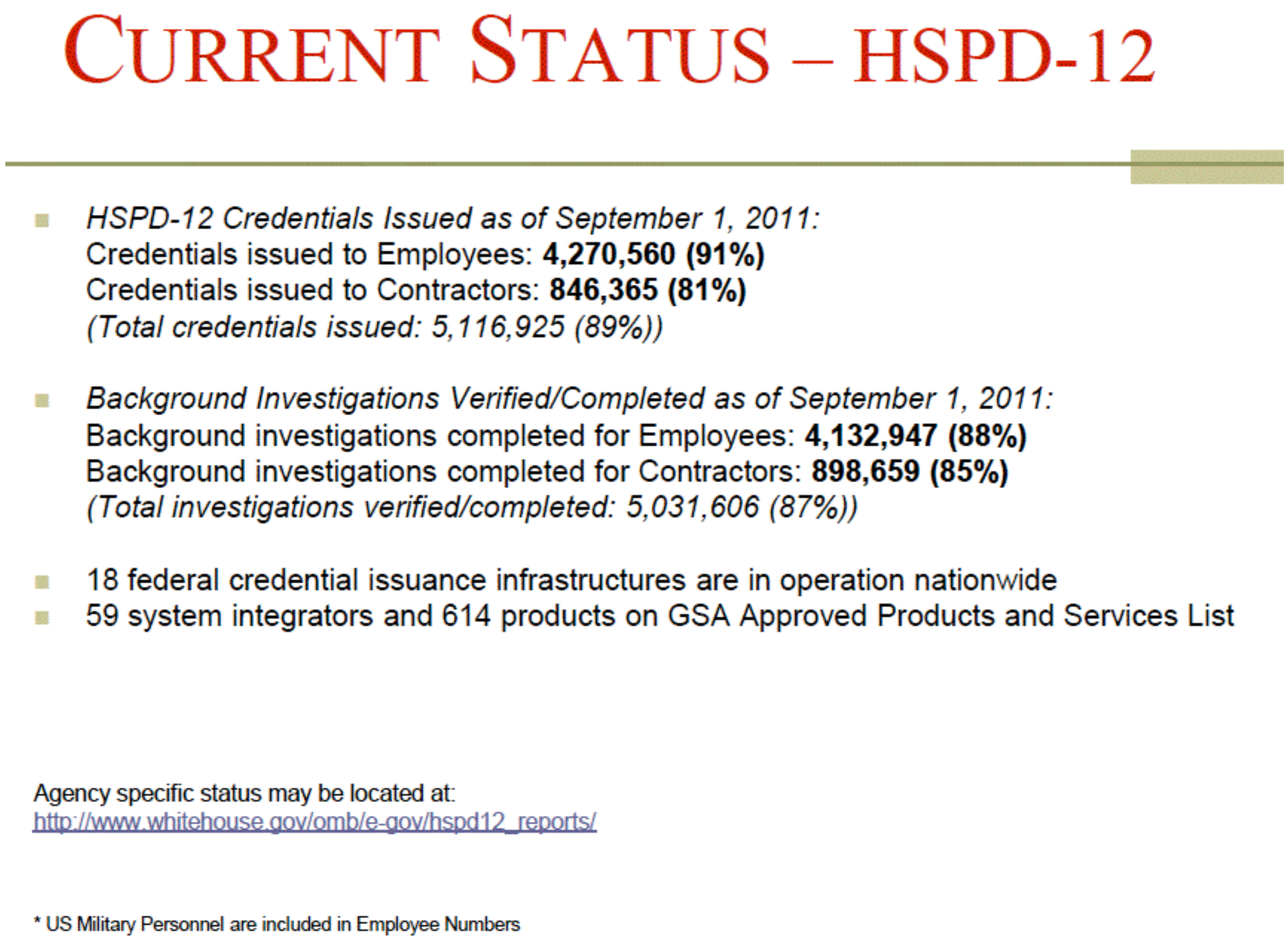
- There may be some confusion when it comes to the "identity" that a cryptographic credential asserts – is it a person's “real name” (e.g., as shown on their driver's license or their passport), or is it something more ephemeral, such as just their email address?

- The answer is, “it may depend.” Some standard assurance personal certificates only validate a user's control over an email address, typically by sending a cryptographic challenge to that address. That's the sort of client certs we'll be working with today.

- Other client certificates may require much more rigorous "identity proofing," perhaps requiring the user to supply government issued identification (or even to undergo a complete background check) before they get issued a higher assurance client cert.

HSPD-12 and Federal CAC/PIV-I Cards

- On August 27th, 2004, then-President George W. Bush issued "Homeland Security Presidential Directive 12," (see http://www.idmanagement.gov/documents/HSPD-12.htm) mandating the establishment of a common identity standard for federal employees and contractors.
- As a result, the federal government (and approved commercial contractors acting on the government's behalf) have already collectively issued millions of "Common Access Cards" ("CACs") and "Personal Identity Verification-Interoperable" ("PIV-I") smart cards.
- "First responders" alone (as defined in HSPD-8) may ultimately require issuance of over 25.3 million such cards. (see http://www.dhs.gov/xlibrary/assets/Partnership_Program_Benefits_Tax_Payers_Public_and_Private_Sector.pdf)
- Part of that process is identity proofing those users – including, in ths case, even doing background investigations.



CURRENT STATUS – HSPD-12

- *HSPD-12 Credentials Issued as of September 1, 2011:*
  Credentials issued to Employees: **4,270,560 (91%)**
  Credentials issued to Contractors: **846,365 (81%)**
  (Total credentials issued: 5,116,925 (89%))

- *Background Investigations Verified/Completed as of September 1, 2011:*
  Background investigations completed for Employees: **4,132,947 (88%)**
  Background investigations completed for Contractors: **898,659 (85%)**
  (Total investigations verified/completed: 5,031,606 (87%))

- 18 federal credential issuance infrastructures are in operation nationwide
- 59 system integrators and 614 products on GSA Approved Products and Services List

Agency specific status may be located at:
http://www.whitehouse.gov/omb/e-gov/hspd12_reports/

* US Military Personnel are included in Employee Numbers

Source: http://www.idmanagement.gov/presentations/HSPD12_Current_Status.pdf

An Aside: CAC/PIV Is A "Proof By Example" That Certs Are Usable By "Mere Mortal" End-Users

- If it was too hard to issue or use a CAC/PIV card, millions of federal employees and contractors would be having trouble doing so. But they're not. For the most part, PKI on hard tokens or smart cards now "just works." This is a real testimony to the hard work of the federal employees and contractors who have been involved with that project.
- This is not to say that there aren't *some* intricacies that may need to be explained. One site that's done a terrific job of user education is the Naval Postgraduate School. Check out their outstanding tri-fold brochure explaining how to use a military CAC card: www.nps.edu/Technology/Security/CAC-guide.pdf

  With the help of that guide, I think most folks would be able to figure out how to do basic CAC/PIV tasks.

<u>Why</u> Are The Feds Using Client Certs? If You Need NIST "LOA-4", They're Basically Your Only Practical Option

- NIST 800-63 Version 1.0.2 (see csrc.nist.gov/publications/nistpubs/ 800-63/SP800-63V1_0_2.pdf) says:

  "Level 4 – Level 4 is intended to provide the highest practical remote network authentication assurance. Level 4 authentication is based on proof of possession of a key through a cryptographic protocol. Level 4 is similar to Level 3 except that only “hard” cryptographic tokens are allowed, FIPS 140-2 cryptographic module validation requirements are strengthened, and subsequent critical data transfers must be authenticated via a key bound to the authentication process. The token shall be a hardware cryptographic module validated at FIPS 140-2 Level 2 or higher overall with at least FIPS 140-2 Level 3 physical security. By requiring a physical token, which cannot readily be copied and since FIPS 140-2 requires operator authentication at Level 2 and higher, this level ensures good, two factor remote authentication."

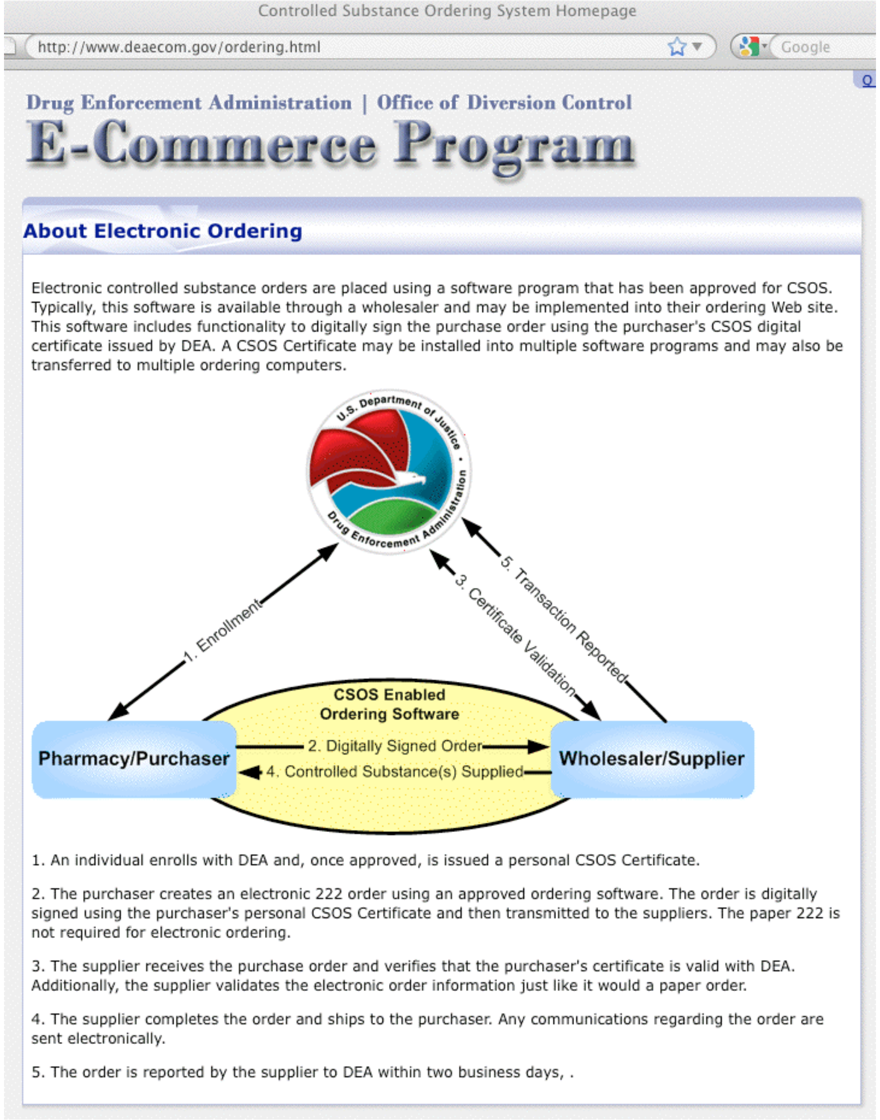
An Aside.... Does Higher Ed **HAVE** Any Use Cases That Actually Require LOA-4?

- Wearing my InCommon Certificate Program Manager hat for a minute, currently InCommon has only one client certificate offering, standard assurance client certs. Should we also have a client certificate offerings tied to the InCommon Assurance Program (e.g., Bronze, Silver, etc.)?
- Do we have any usage case that would require LOA-4, or would LOA-3 be "good enough" for all potential higher ed usage scenarios? (LOA-3 requires two factor, but not necessarily client certs). I'm strongly interested in understanding what might drive LOA-4 adoption...
- If we did offer an LOA-3 or LOA-4 compliant cert profile, it would imply stronger identity proofing. Would higher education users be willing to put up with rigorous identity proofing hassles? (by way of comparison, we haven't seen a tremendous number of extended validation server certificates requested, even though they're available at no additional cost as part of the InCommon Certificate Program)

An Aside: "Identity Proofing" for Regular Citizens

- If you travel extensively, you've probably run into long lines at customs, either while coming into the U.S., or perhaps while travelling into Canada or Mexico. If so, you may have noticed that some folks ("Trusted Travellers") can use the "Global Online Entry System" ("GOES") and/or NEXUS/SENTRI to avoid those lines. A growing number of airports also offer "TSA PreCheck" lines for participants in that program. (see http://www.globalentry.gov/). "Trusted Travellers" are issued a machine readable high-assurance credential ($50 for 5 years) for that purpose.
- Obviously, however, it would be bad to issue a credential of this sort to a person you hadn't thoroughly identity proofed. Therefore, if you apply to be a Trusted Traveller, your identity is validated in multiple ways including a review of government records (you don't want to issue a card to a criminal, for example!); review of existing documents (such as your passport); collection of biometrics, e.g., a photograph, fingerprints, and in some cases a picture of iris/retina. You also need to physically appear in person for an interview. Travellers weary of being stalled at the border will put up with those hassles, but would regular higher ed users do so?

Some Federal High Security Applications That Now Use Client Certs May Be Surprising



Controlled Substance Ordering System Homepage
http://www.deaecom.gov/ordering.html
Google

Drug Enforcement Administration | Office of Diversion Control

E-Commerce Program

About Electronic Ordering

Electronic controlled substance orders are placed using a software program that has been approved for CSOS. Typically, this software is available through a wholesaler and may be implemented into their ordering Web site. This software includes functionality to digitally sign the purchase order using the purchaser's CSOS digital certificate issued by DEA. A CSOS Certificate may be installed into multiple software programs and may also be transferred to multiple ordering computers.

U.S. Department of Justice • Drug Enforcement Administration
1. Enrollment
3. Certificate Validation
5. Transaction Reported
CSOS Enabled Ordering Software
Pharmacy/Purchaser
2. Digitally Signed Order
4. Controlled Substance(s) Supplied
Wholesaler/Supplier

1. An individual enrolls with DEA and, once approved, is issued a personal CSOS Certificate.

2. The purchaser creates an electronic 222 order using an approved ordering software. The order is digitally signed using the purchaser's personal CSOS Certificate and then transmitted to the suppliers. The paper 222 is not required for electronic ordering.

3. The supplier receives the purchase order and verifies that the purchaser's certificate is valid with DEA. Additionally, the supplier validates the electronic order information just like it would a paper order.

4. The supplier completes the order and ships to the purchaser. Any communications regarding the order are sent electronically.

5. The order is reported by the supplier to DEA within two business days, .

Client Certs Can Even Be Secure Enough for Use in Conjunction with National Security Systems

- See the "National Policy for Public Key Infrastructure in National Security Systems," March 2009 (http://www.cnss.gov/Assets/pdf/CNSSP-25.pdf) makes it clear that client certs even form the foundation for NSS uses:

  "(U) NSS operating at the unclassified level shall obtain PKI support from the established Federal PKI Architecture.
  "(U) NSS operating at the Secret level shall obtain PKI support from the NSS-PKI.
  "(U) The NSS-PKI hierarchy shall rest on a Root Certificate Authority (CA) operated on behalf of the national security community in accordance with policies established by the CNSS PKI Member Governing Body. The NSS-PKI Root
  CA shall serve as the anchor of trust for the NSS-PKI."
- TS/SCI ("JWICS") counterpart of the NSS-PKI? IC-PKI.

Certificates Are Now Also Being Used to Secure National Critical Infrastructure

- For example, consider the national electrical grid. The North American Energy Standards Board's ("NAESB") 2012 Annual Plan for the Wholesale Electric Quadrant specifically discusses their plans for deploying PKI on pages 4 and following. (See http://www.naesb.org/pdf4/weq_2012_annual_plan.docx and http://www.naesb.org/weq/weq_pki.asp)
- This is begining to be deployed/made real, too, right now:

-- "Shift Systems Identified as the First NAESB Authorized Certification Authority," Feb 16, 2012, http://www.prnewswire.com/news-releases/shift-systems-identified-as-the-first-naesb-authorized-certification-authority-139493283.html

-- "OATI webCARES Authorized by NAESB for webRegistry," Apr 11, 2012, http://www.prweb.com/releases/2012/4/prweb9390545.htm

-- "GlobalSign Announces Accreditatin as Authorized Certificate Authority for the North American Energy Standards Board," Apr 23, 2012,
http://www.prweb.com/releases/2012/4/prweb9431614.htm

And, Of Course, Some Large Corporations and Agencies Have Used Client Certificates for Years

- A nice indication of interest in/use of client certificates can be seen in things like participation in the "Smart Card Alliance," see http://www.smartcardalliance.org/pages/alliance-members including: American Express, Bank of America, Booz Allen Hamilton, Capital One, Chase, CSC, Deloitte & Touche, Hewlett-Packard, Ingersoll Rand, Lockheed Martin, MasterCard, SAIC, Visa, Wells Fargo, and many others.
- To understand how smart cards relate to client certificates, note that smart cards are a way to securely store client certificates on what looks like a credit card (if you look closely, you'll see that a smart card differs from a traditional credit card in that it has a small set of flush gold-colored contacts on the front).
- Many large companies use smart cards as the foundation for their corporate employee ID cards.

IV. "Non Adoption" of Client Certs

So Why Haven't Client Certs "Taken Off" More Broadly?

- And what can we do to fix this, assuming we want to?

- It isn't simply that client certs are new... http://en.wikipedia.org/wiki/Public_key_infrastructure#History ties the origin of PKI to 1969, with public disclosure of some of the key algorithms dating to 1976 – that's thirty five years ago. The RSA PKCS ("Public Key Cryptography Standards") documents date to 1993 – that's eighteen years ago. By Internet standards, all of this work is "ancient" (or "well established," if you prefer).

- So it *isn't* simply that PKI's the "new kid on the block."

- There are (or may be) many other possible reasons why client certificates have struggled so far....

Economics? Are Client Certs Too Expensive?

- "There are several reasons PKI has failed, says Peter Tippett, head of the industry solutions and security practice at Verizon Business.

  > "The main reason organisation do not use PKI, he told attendees of RSA Conference 2011, is that **it costs too much.** "Speaking on a debate on the importance of identity to internet security, he said very few organisations are able to make a business case for spending $200 to $300 per user, per year."

  "Why Public Key Infrastructure Has Failed", http://www.computerweekly.com/blogs/read-all-about-it/2011/02/why-public-key-infrastructure.html [emphasis added]

  How much would YOUR school pay per user, per year?

My Target Cost for Client Certs: $1/user/month

- Lacking hard data, I'm going to suggest a nominal amount that might be acceptable: $1/user/month (inclusive of all costs), over a normal four year undergraduate enrollment, or $48.00 per user over a quadrennial period.

- For context: (a) www.nacs.org states that the average price for a new textbook in 2009-2010 was $62.00
  (b) one major online vendor quotes quotes 3 year RSA SecurID 700 one time password Tokens (in a 5 pack) @ $55.60/token

- InCommon sells hard tokens for $19.80/unit to Internet2 members (see http://www.incommon.org/safenet/pricing.html) which would leave ~$6/user/year to cover other costs, assuming client certs are getting deployed on USB format hard tokens.

In Some Cases, The Client Certs Themselves Are "Free"

- If you've signed up to participate in the InCommon Certificate program, you get the bundled ability to issue client certs at no additional cost, and even if your school doesn't participate in the InCommon Certificate program, individuals can still get free client certificates for personal/home use, see:

  www.comodo.com/home/email-security/free-email-certificate.php

- That said, obviously the cost of the certs themselves are not the only costs associated with rolling out client certs (for example, on the preceding page, we talked about hard token costs).

- So what other non-technical explanations, other than cost, do people offer for client certificate non-deployment?

Is *Usability* Actually The Problem?

- "Despite many years of effort, PKI technology has failed to take off except in a few niche areas. Reasons for this abound […] Probably the primary factor at the user level […] is the high level of difficulty involved in deploying and using a PKI. There is considerable evidence from mailing lists, Usenet newsgroups and web forums, and directly from the users themselves, that acquiring a certificate is the single biggest hurdle faced by users. For example various user comments indicate that it takes a skilled technical user between 30 minutes and 4 hours work to obtain a certificate from a public CA that performs little to no verification [...] [A] set of highly technical users, most with PhDs in computer science, took over two hours to set up a certificate for their own use and rated it as the most difficult computer task that they'd ever been asked to perform."

  Peter Gutmann, University of Auckland, Usenix '03,
  http://dl.acm.org/citation.cfm?id=1251353.1251357

Things Have Come A Long Way, Usability-Wise

- For example, these days, the process for obtaining a client certificate can be as simple as:
  -- Complete a short online secure web form
  -- Click on a link sent to you by email to download your client certificate into your browser.
  Don't believe it? We'll have everyone try getting their own client cert later in this session. (We might also talk about whether this has swung too far in the "too easy" direction, I suppose)
- There may still be some ugly bits to do after getting your cert (depending on how you want to use it), but at least some edu sites have developed local scripts that make the installation process pretty painless for their users.
- Internet2/InCommon is/soon will be working on offering a generally available certificate installation tool, based on/modeled after those site-specific installation tools.

Or Is The Problem That Other Solutions Have Usurped PKI's Market Niche(s)?

- If you've got PGP (or GNU Privacy Guard) to sign or encrypt email, do you also need PKI client certs and S/MIME for signed/encrypted email?
- If your site is using one time password (OTP) crypto fobs (or you use ssh with preshared keys), do you still need client certs for auth to sensitive systems? (And what about a 2nd ***channel*** solution leveraging smart phones, such as InCommon's new offering with Duo Security, see http://www.incommon.org/duo/index.html)
- Has the success of InCommon (and other federated authentication efforts) eliminated the need for PKI-based cross-entity credentials? Federation seems to be the direction that the National Strategy for Trusted Identities in Cyberspace (NSTIC) is going, and it may be worth noting that some have always worried about the privacy implications of PKI-style "national ID cards" online...

"Is NSTIC a plan to introduce a national ID card or an internet driver's license? Do I have to get one?"

"No. The government will not require that you get a trusted ID. If you want to get one, you will be able to choose among multiple identity providers — both private and public — and among multiple digital credentials. Such a marketplace will ensure that no single credential or centralized database can emerge. Even if you do choose to get a credential from an ID provider, you would still be able to surf the Web, write a blog, visit chat rooms, or do other things online anonymously or under a pseudonym". [FAQ item response continues here]

* http://www.nist.gov/nstic/faqs.html

.

A Humorous Comment (With An Underlying Grain of Truth?): The PKI DeLorean* Hypothesis

- "[M]aybe the possible future in which everything is PKI-enabled and digital certificates are ubiquitous is so horrendous that it actually sent ripples of bad luck back through time that sabotaged the development and deployment of PKI technology. Some things actually seem to make a lot of sense from this point of view."

  "Why PKI Failed," Luther Martin, 29 October 2009,
  http://superconductor.voltage.com/2009/10/why-pki-failed.html
  [a blog about security, cryptography and usability]

  * C.F. http://en.wikipedia.org/wiki/Back_to_the_Future

"Fixing PKI" – A Cottage Industry of Its Own

- PKI has been successful in one (quite perverse way): it has succeeded in inspiring hundreds of papers and talks attempting to explain precisely why PKI has failed so far.

- One author even went so far as to say,

  > '[I]t seems a rite of passage for the serious security researcher to write a paper with a title such as "Improving PKI..." Never in the field of security research has so much been written by so many, to be read by so few.'
  > http://iang.org/ssl/pki_considered_harmful.html

Or Are Some Fundamental Technical Bits So Broken That They Make Sane People Run Away From PKI?

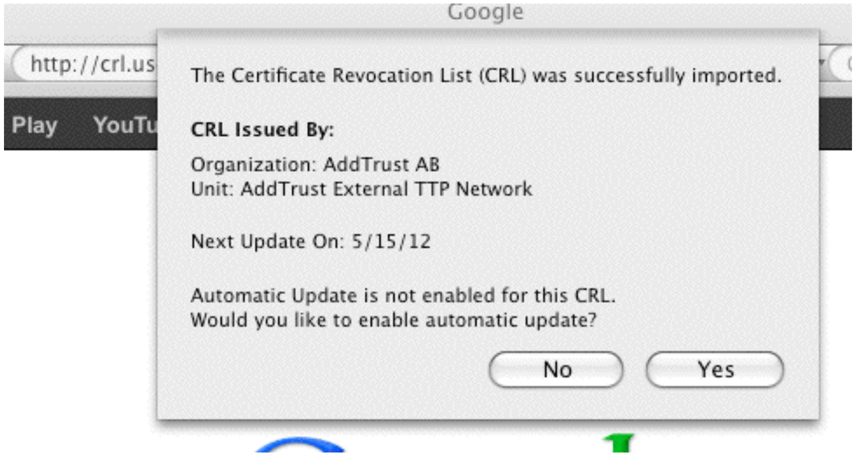
- For example, what about revoking or cancelling client certificates?
- Hypothetically imagine that you're a manager and you're firing an employee. As part of doing that, you collect their door key and company credit card (or you have the locks changed and the credit card cancelled if they've been "lost").
- But what about revoking a client certificate they might have been issued? (For now, let's assume that it wasn't issued in non-exportable form on a smart card or PKI hard token)
- How would you cancel or revoke it?

Revoking A Client Cert

- Unfortunately, unlike "taking back" a physical door key or cutting up a credit card, it's harder to "take back" an electronic credential.
- CRLs ("certificate revocation lists", see RFC3280 and RFC5280) were meant to handle this problem, much like those printed books of stolen or revoked credit card numbers that stores used to get from the bank card companies bank in the old days. Most CAs currently publish a CRL once a day. Some users may check or download those daily CRLs, but most don't. And if you're a CA, or you're a user with a compromised cert, you really don't want to have to wait up to 24 hours to sort-of-revoke a compromised credential, nor do you really want millions of user to have to potentially download a huge file listing piles of revoked certs!
- OCSP ("online certificate status protocol", RFC2560) was meant to handle this issue much more directly, and interactively, but many browsers and email clients don't check a cert's OCSP status. Ugh.

Locally Importing a CRL

- An example of a CRL is: http://crl.usertrust.com/AddTrustExternalCARoot.crl
- If you visit that URL, it will be imported into your browser.
- You can also schedule the CRL to be automatically updated, if you'd like to do so...



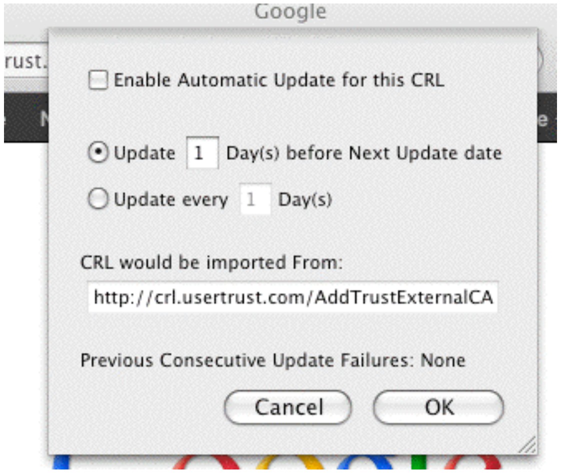
The Certificate Revocation List (CRL) was successfully imported.

CRL Issued By:

Organization: AddTrust AB
Unit: AddTrust External TTP Network

Next Update On: 5/15/12

Automatic Update is not enabled for this CRL.
Would you like to enable automatic update?



☐ Enable Automatic Update for this CRL

● Update 1 Day(s) before Next Update date

○ Update every 1 Day(s)

CRL would be imported From:

http://crl.usertrust.com/AddTrustExternalCA

Previous Consecutive Update Failures: None

- But, and this is critical if you believe scalability is important: you shouldn't need to download an ever growing list of killed certs.

CRLs: The "hosts" File of PKI

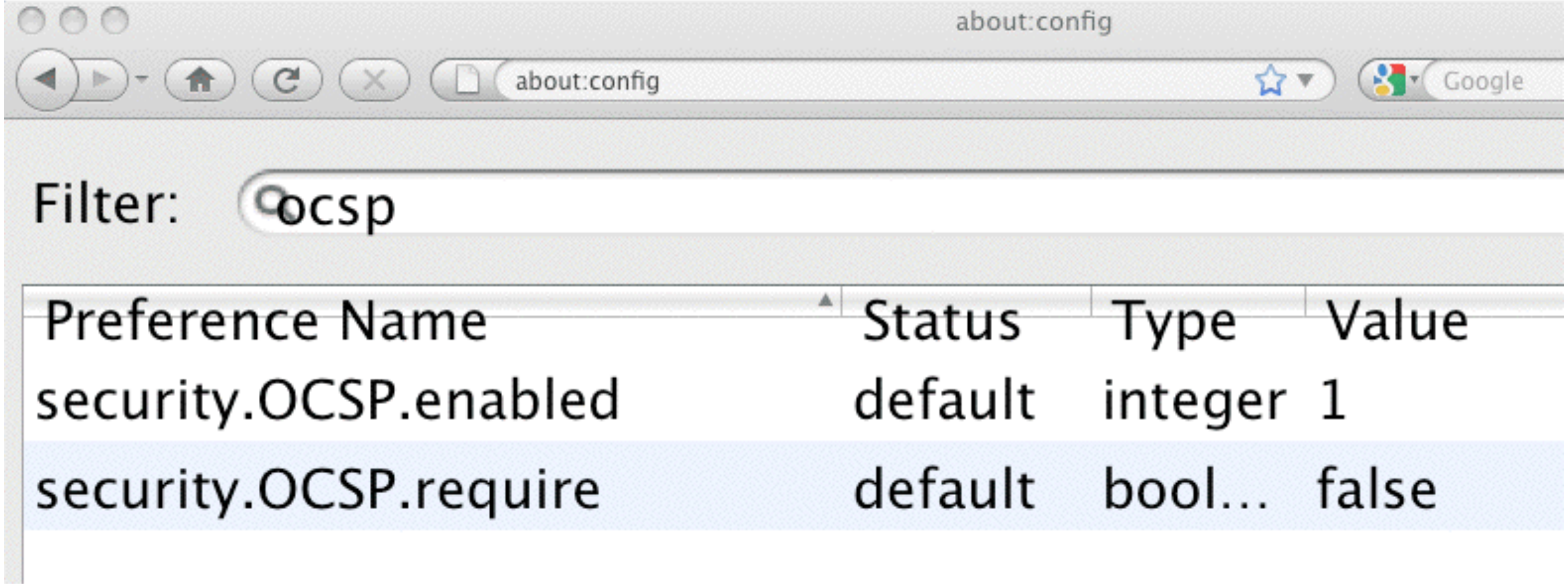
- Note that *each* CA will offer one or more CRLs, and there are hundreds of CAs out there! Normally you would NOT want to routinely import all those CRLs all the time on each system! This simply doesn't scale to Internet-size audiences.

- In many ways, this reminds me strongly of "hosts" files in the old pre-DNS days – you know, people would copy around static files with mappings of hostnames to IP addresses.

- Do you really think we'd have the size Internet we have today, if that sort of thing still had to happen? Clearly, no.

So What About OCSP?

- You can check to see how OCSP is configured in Firefox by going to about:config and then filtering for ocsp. For example (enlarged for ease of viewing):



Filter: ocsp

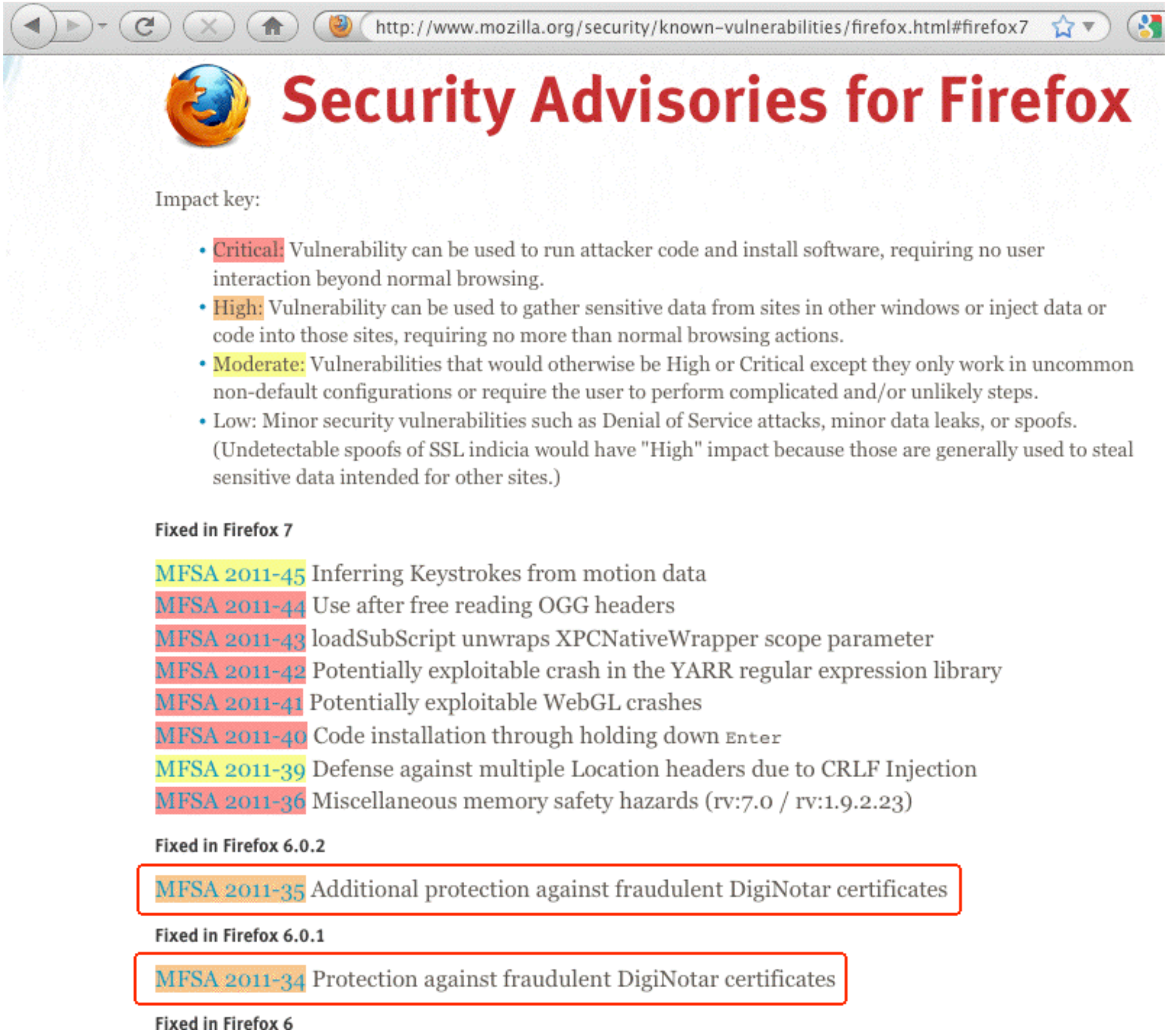
Preference Name	Status	Type	Value
security.OCSP.enabled	default	integer	1
security.OCSP.require	default	bool…	false

- Note that OCSP is checked but is NOT REQUIRED by default in Firefox. You can change it to be required if you want to, but in doing so, you'll break access to some SSL/TLS-secured sites.

Chicken/Egg Interactions and Insisting on OCSP

- Assume you're connecting via a captive portal, and the captive portal blocks all external access by default until you've logged in to an SSL/TLS-secured pages.
- Now assume that you are using a browser that strictly requires OCSP validation... but OCSP validation requires the ability to connect to the OCSP responder, and that requires the ability to resolve the DNS name, and to connect to that host... but that requires network access... Nice circular deadlock, eh?
- My point in dwelling on CRLs and OCSPs early in today's session is to give you a heads up that there are some architectural and security complexities that do exist, and that may be necessary to "resolve" if you want certs to work in some environments... but those don't need to be "show stoppers" in my opinion.
- Clearly cert revocation is (or can potentially be) tricky. This is why, when it really matters, browser vendors issue patches to kill certs

A List of Some Firefox Security Advisories



http://www.mozilla.org/security/known-vulnerabilities/firefox.html#firefox7

Security Advisories for Firefox

Impact key:

- Critical: Vulnerability can be used to run attacker code and install software, requiring no user interaction beyond normal browsing.
- High: Vulnerability can be used to gather sensitive data from sites in other windows or inject data or code into those sites, requiring no more than normal browsing actions.
- Moderate: Vulnerabilities that would otherwise be High or Critical except they only work in uncommon non-default configurations or require the user to perform complicated and/or unlikely steps.
- Low: Minor security vulnerabilities such as Denial of Service attacks, minor data leaks, or spoofs. (Undetectable spoofs of SSL indicia would have "High" impact because those are generally used to steal sensitive data intended for other sites.)

Fixed in Firefox 7

MFSA 2011-45 Inferring Keystrokes from motion data
MFSA 2011-44 Use after free reading OGG headers
MFSA 2011-43 loadSubScript unwraps XPCNativeWrapper scope parameter
MFSA 2011-42 Potentially exploitable crash in the YARR regular expression library
MFSA 2011-41 Potentially exploitable WebGL crashes
MFSA 2011-40 Code installation through holding down `Enter`
MFSA 2011-39 Defense against multiple Location headers due to CRLF Injection
MFSA 2011-36 Miscellaneous memory safety hazards (rv:7.0 / rv:1.9.2.23)

Fixed in Firefox 6.0.2

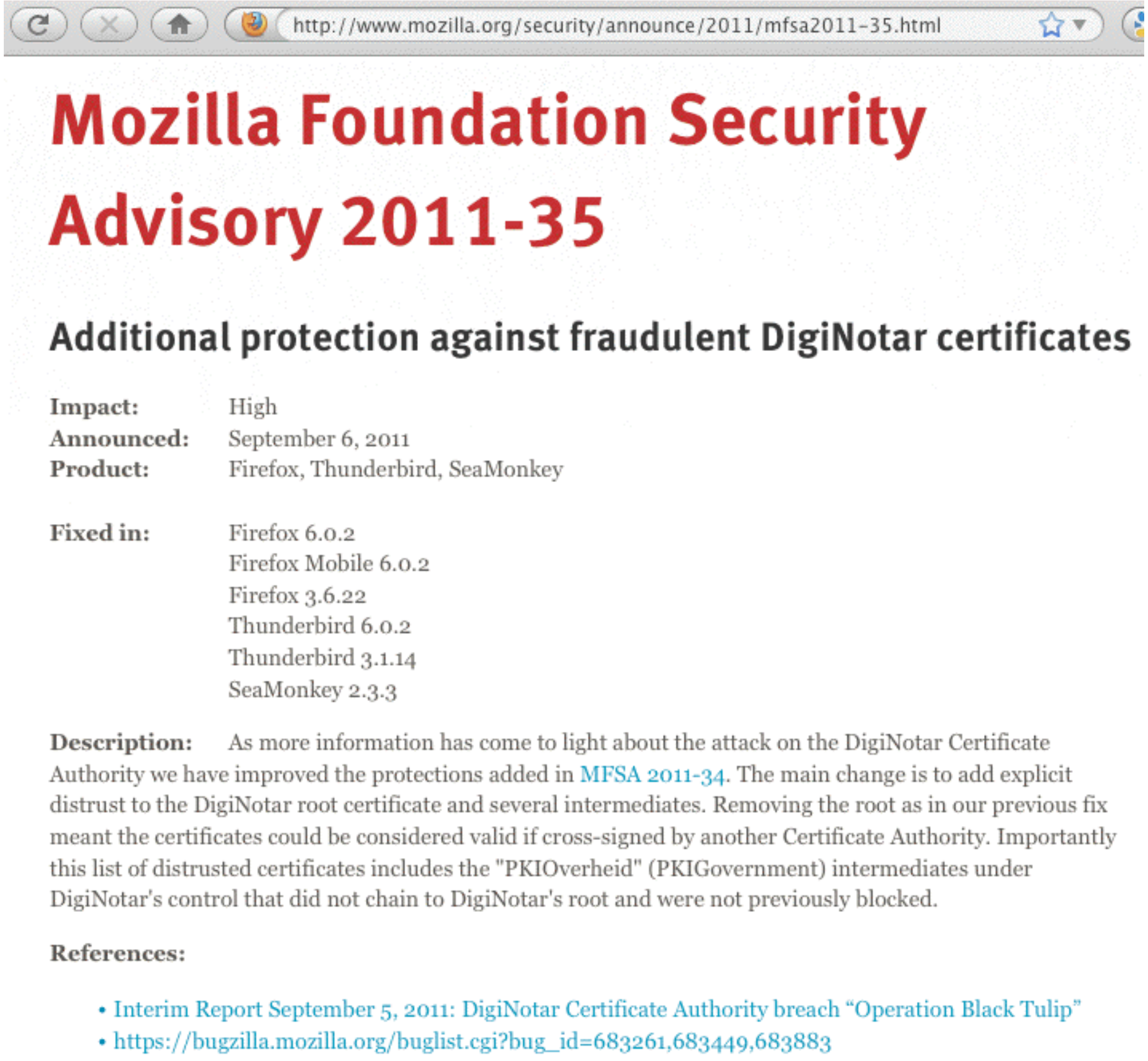
MFSA 2011-35 Additional protection against fraudulent DigiNotar certificates

Fixed in Firefox 6.0.1

MFSA 2011-34 Protection against fraudulent DigiNotar certificates

Fixed in Firefox 6

Example of One of Those Specific Advisories



http://www.mozilla.org/security/announce/2011/mfsa2011-35.html

Mozilla Foundation Security Advisory 2011-35

Additional protection against fraudulent DigiNotar certificates

Impact: High
Announced: September 6, 2011
Product: Firefox, Thunderbird, SeaMonkey

Fixed in: Firefox 6.0.2
Firefox Mobile 6.0.2
Firefox 3.6.22
Thunderbird 6.0.2
Thunderbird 3.1.14
SeaMonkey 2.3.3

Description: As more information has come to light about the attack on the DigiNotar Certificate Authority we have improved the protections added in MFSA 2011-34. The main change is to add explicit distrust to the DigiNotar root certificate and several intermediates. Removing the root as in our previous fix meant the certificates could be considered valid if cross-signed by another Certificate Authority. Importantly this list of distrusted certificates includes the "PKIOverheid" (PKIGovernment) intermediates under DigiNotar's control that did not chain to DigiNotar's root and were not previously blocked.

References:

- Interim Report September 5, 2011: DigiNotar Certificate Authority breach “Operation Black Tulip”
- https://bugzilla.mozilla.org/buglist.cgi?bug_id=683261,683449,683883

I've Rambled Enough...

- We could talk for hours when it comes to providing crypto background, but let's see how this all actually works... let's get a client cert and get set up to send and receive secure email.

- The next part of today's session thus looks like:

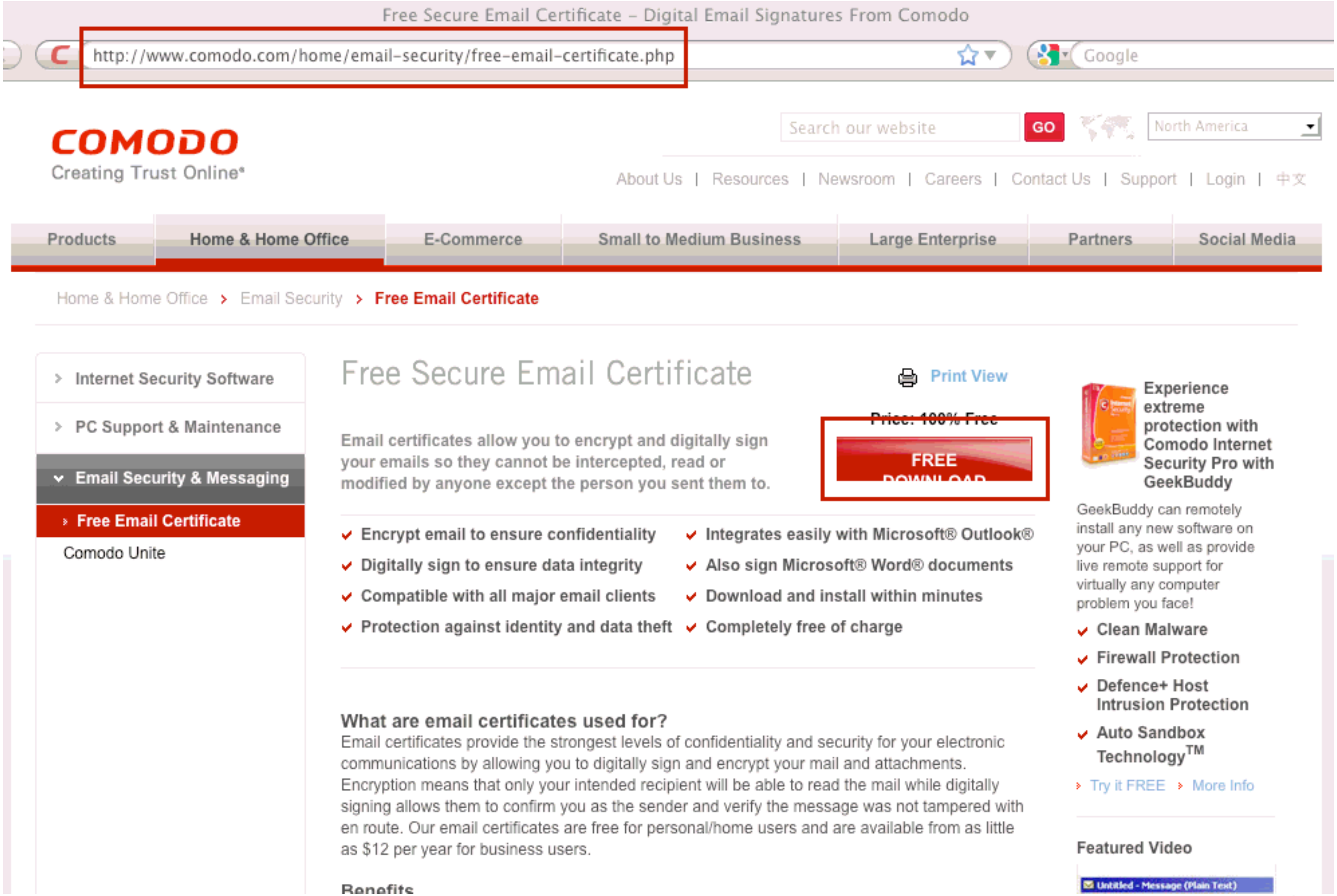
  -- applying for a client cert
  -- successfully downloading/installing it in Firefox
  -- backing it up
  -- installing the cert in Thunderbird
  -- configuring Thunderbird to do S/MIME

V. Getting A Free S/MIME Client Certificate

Getting a Free Client Cert for S/MIME With Firefox

- To do S/MIME, you'll need an email account and a client cert. We'll assume you already have an email account you can use, and we'll get our free-for-personal-use client certificate from Comodo. Thank you, Comodo! To get it, go to: http://tinyurl.com/free-cert (http://www.comodo.com/home/email-security/free-email-certificate.php)
- We're going to use Firefox to apply for and download our cert from Comodo. While you can use pretty much any popular browser with client certs, for the purpose of this training, if you're following along, as we go through this, please ONLY use Firefox. If you don't already have Firefox, you can get it for free from: http://www.mozilla.org/en-US/firefox/fx/
- Mac vs. PC or Linux: Although we'll be using Firefox on a Mac in these slides, Firefox on Microsoft Windows or Linux will be virtually identical.

Comodo’s Free Secure Email Certificate Web Site



Free Secure Email Certificate – Digital Email Signatures From Comodo
http://www.comodo.com/home/email-security/free-email-certificate.php
Google
COMODO
Creating Trust Online®
Search our website
GO
North America
About Us | Resources | Newsroom | Careers | Contact Us | Support | Login | 中文
Products
Home & Home Office
E-Commerce
Small to Medium Business
Large Enterprise
Partners
Social Media
Home & Home Office › Email Security › Free Email Certificate
Internet Security Software
PC Support & Maintenance
Email Security & Messaging
Free Email Certificate
Comodo Unite
Print View
Price: 100% Free
FREE DOWNLOAD

Free Secure Email Certificate

Email certificates allow you to encrypt and digitally sign your emails so they cannot be intercepted, read or modified by anyone except the person you sent them to.

- ✓ Encrypt email to ensure confidentiality
- ✓ Digitally sign to ensure data integrity
- ✓ Compatible with all major email clients
- ✓ Protection against identity and data theft
- ✓ Integrates easily with Microsoft® Outlook®
- ✓ Also sign Microsoft® Word® documents
- ✓ Download and install within minutes
- ✓ Completely free of charge

What are email certificates used for?

Email certificates provide the strongest levels of confidentiality and security for your electronic communications by allowing you to digitally sign and encrypt your mail and attachments. Encryption means that only your intended recipient will be able to read the mail while digitally signing allows them to confirm you as the sender and verify the message was not tampered with en route. Our email certificates are free for personal/home users and are available from as little as $12 per year for business users.

Benefits

Experience extreme protection with Comodo Internet Security Pro with GeekBuddy

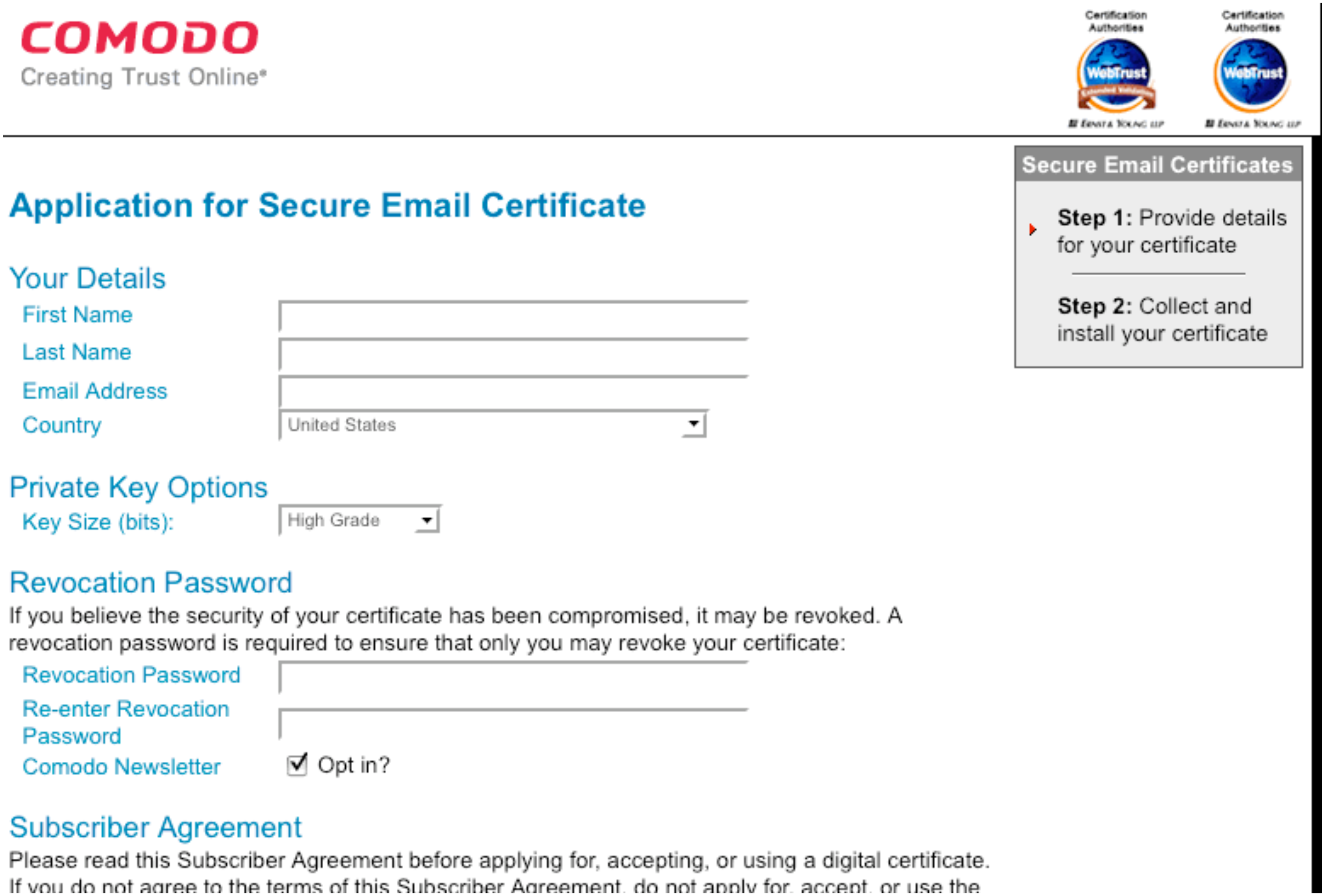
GeekBuddy can remotely install any new software on your PC, as well as provide live remote support for virtually any computer problem you face!

- ✓ Clean Malware
- ✓ Firewall Protection
- ✓ Defence+ Host Intrusion Protection
- ✓ Auto Sandbox Technology™

› Try it FREE › More Info

Featured Video

The Application Form You'll Complete



Application for Secure Email Certificate

Your Details

First Name

Last Name

Email Address

Country: United States

Private Key Options

Key Size (bits): High Grade

Revocation Password

If you believe the security of your certificate has been compromised, it may be revoked. A revocation password is required to ensure that only you may revoke your certificate:

Revocation Password

Re-enter Revocation Password

Comodo Newsletter ☑ Opt in?

Subscriber Agreement

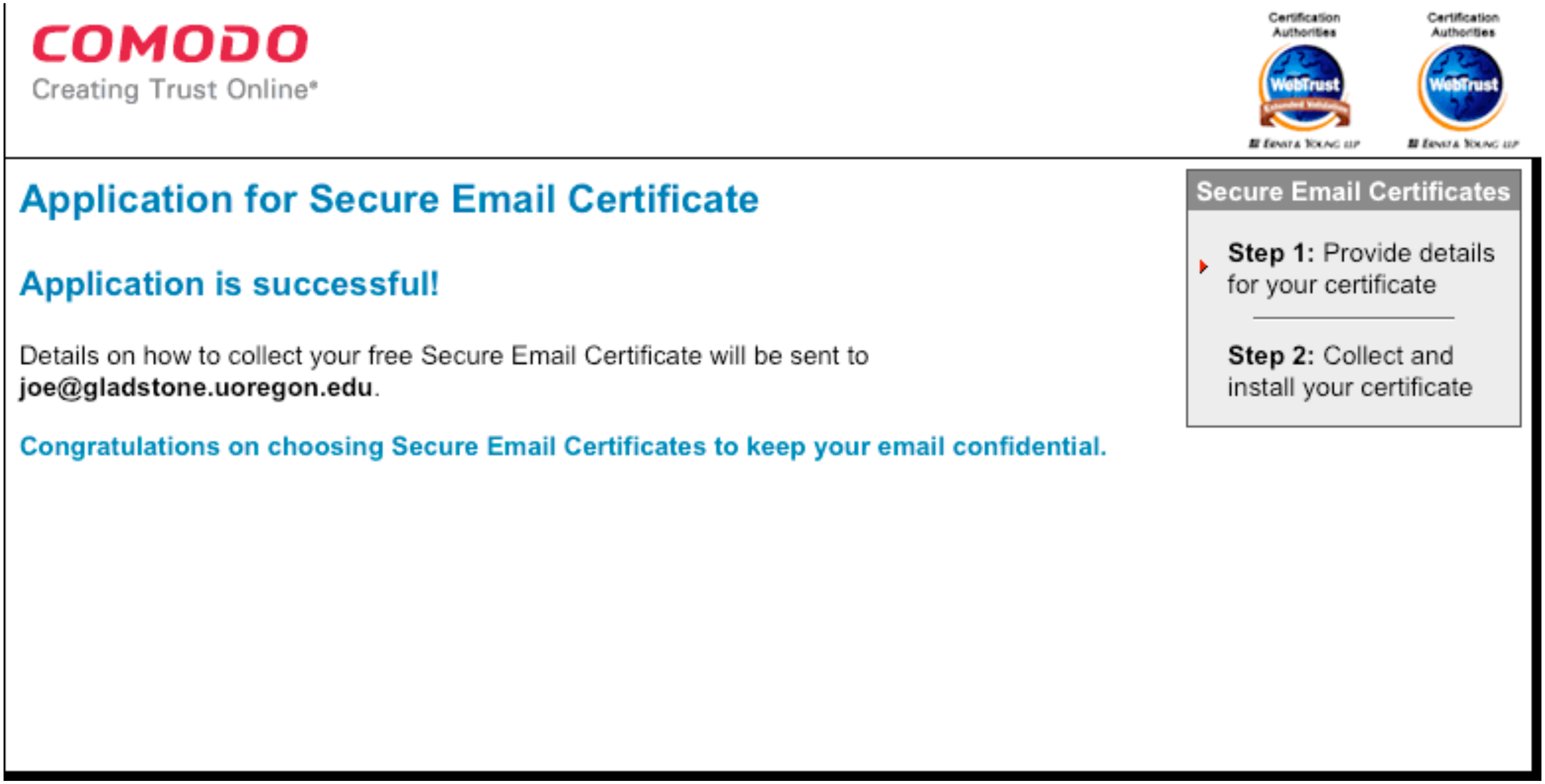
Please read this Subscriber Agreement before applying for, accepting, or using a digital certificate. If you do not agree to the terms of this Subscriber Agreement, do not apply for, accept, or use the

Secure Email Certificates

Step 1: Provide details for your certificate

Step 2: Collect and install your certificate

Successful Application…



Application for Secure Email Certificate

Application is successful!

Details on how to collect your free Secure Email Certificate will be sent to **joe@gladstone.uoregon.edu**.

Congratulations on choosing Secure Email Certificates to keep your email confidential.

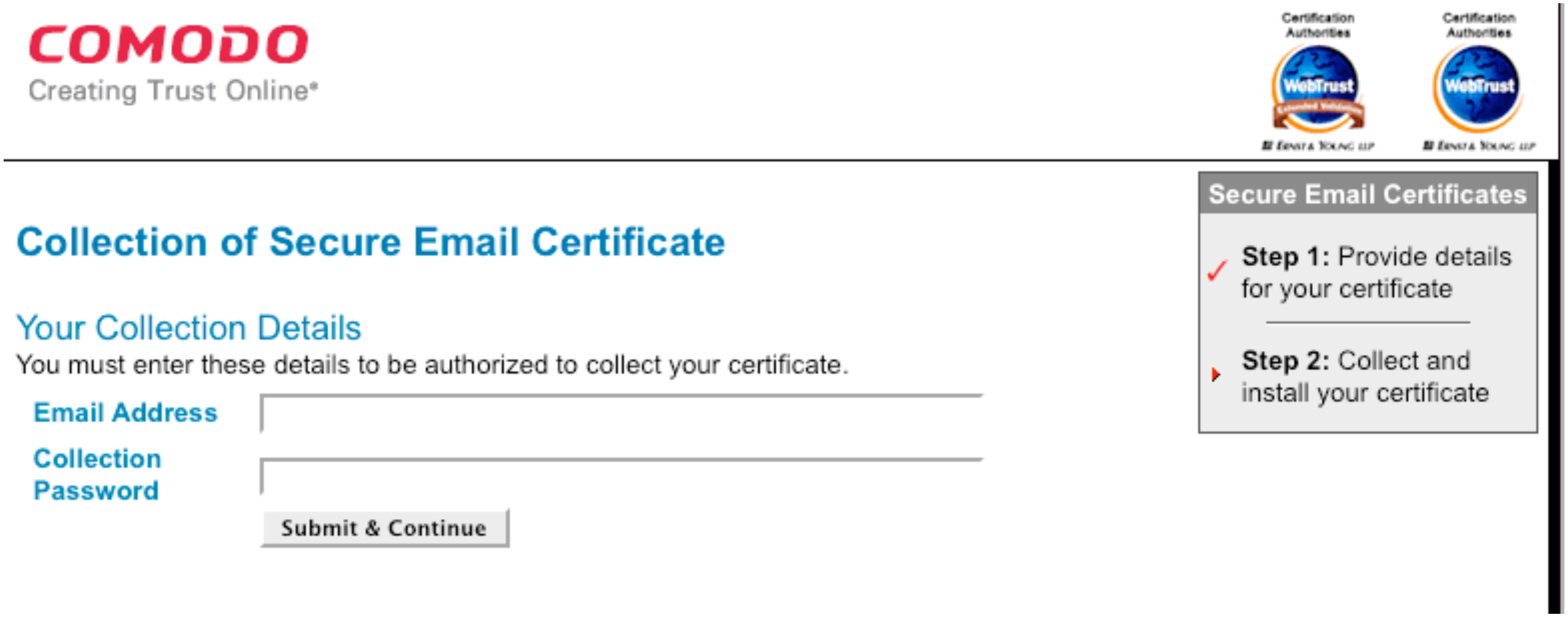
Secure Email Certificates

Step 1: Provide details for your certificate

Step 2: Collect and install your certificate

At this point, folks, please check your email from Comodo. You'll need to go to the web link that they've sent you…

Collecting Your Certificate



Collection of Secure Email Certificate

Your Collection Details

You must enter these details to be authorized to collect your certificate.

Email Address

Collection Password

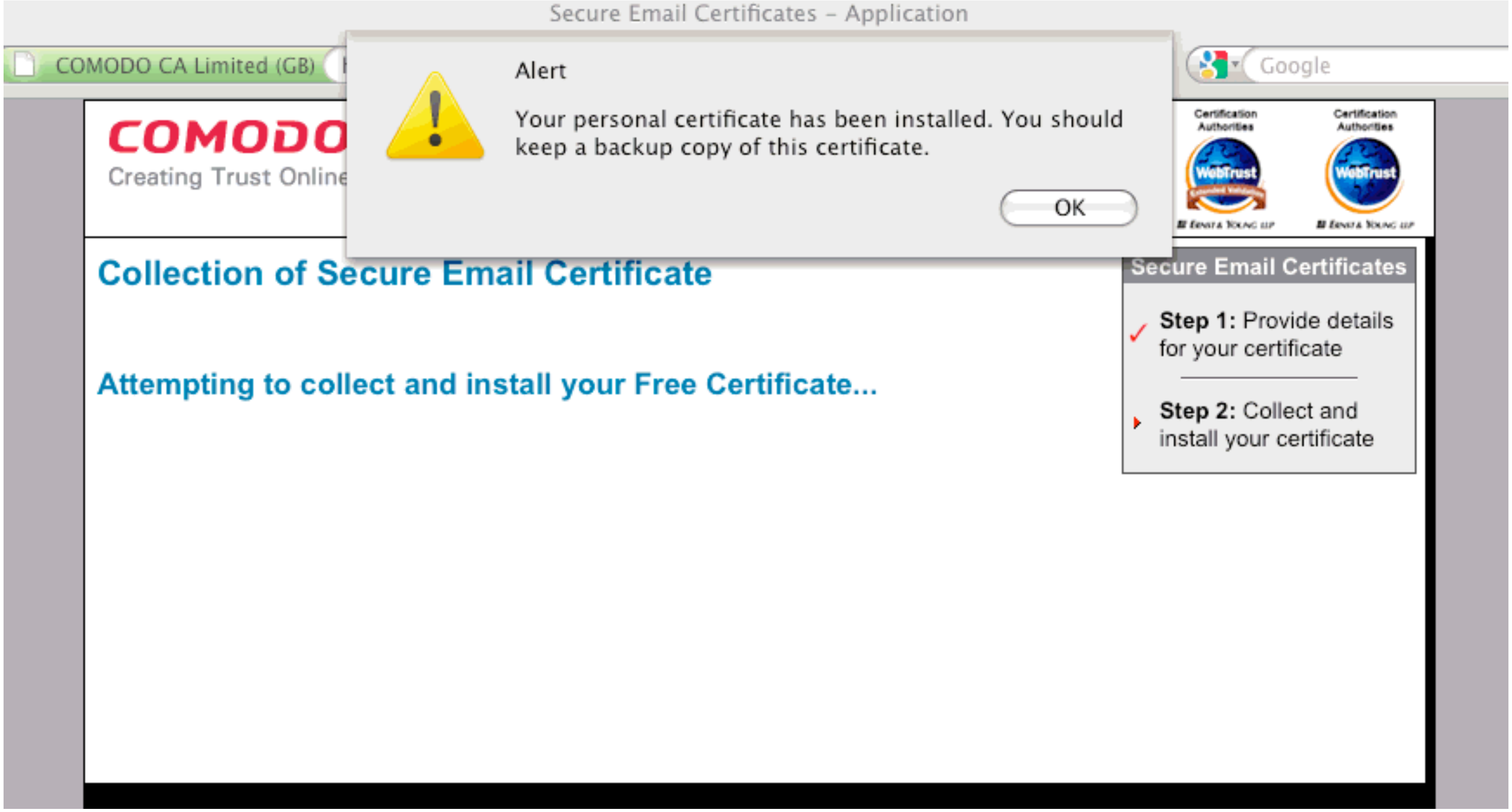
Secure Email Certificates

✓ **Step 1:** Provide details for your certificate

Step 2: Collect and install your certificate

To collect your certificate, using the SAME BROWSER on the SAME SYSTEM you used to apply for your certificate, go to the URL you were sent in email and plug in your email address and the unique password that they provided

Successful Certificate Download…



Alert

Your personal certificate has been installed. You should keep a backup copy of this certificate.

Collection of Secure Email Certificate

Attempting to collect and install your Free Certificate...

Secure Email Certificates

✓ **Step 1:** Provide details for your certificate

Step 2: Collect and install your certificate

"Where *Else* Can I Get Client Certs?"

- While we're only going to show use of the free ***one year*** Comodo client cert for personal use in this training, you can also get a paid client cert from Comodo's "EnterpriseSSL" division, and free or paid client certs from other vendors. See, for example:

  -- http://www.enterprisessl.com/ssl-certificate-products/addsupport/secure-email-certificates.html

  -- http://www.globalsign.com/authentication-secure-email/digital-id/compare-digital-id.html

  -- http://www.symantec.com/verisign/digital-id/buy

  -- http://www.trustcenter.de/en/products/tc_personal_id.htm

InCommon's Client Certificate Program

- Because this is a higher education audience, I'll also note that if you sign up for InCommon's Client Certificate Service (see http://www.incommon.org/cert/), InCommon includes the ability for you to issue client certificates as well as traditional SSL/TLS server certificates at no extra charge.
- Also note that if you participate in InCommon's Certificate Program, you can issue certs both via a web interface (the "Comodo Certificate Manager") and via a programmable API with synchronous client cert issuance within five seconds.
- See https://www.incommon.org/cert/repository/ for the InCommon Certificate Manager (CM) Guide, the End User Guide for Client Certificates, and the Certificate Manager (CM) SMIME Enroll API Guide for more information.

VI. Examining and Backing Up Your New Client Certificate

"Okay, I've Got My Client Cert. What Do I Do Now?"

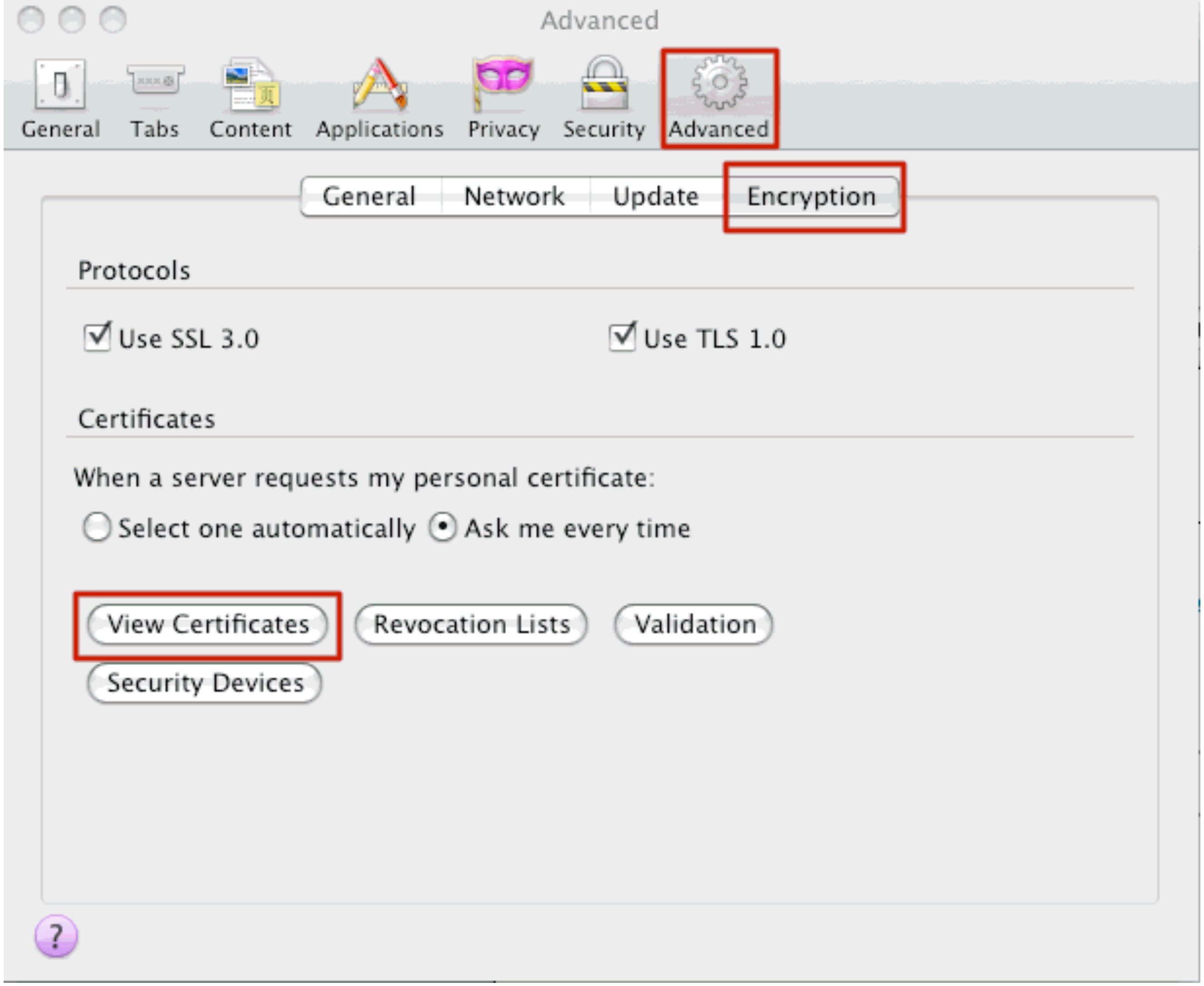
- When Comodo gave you your client cert, remember that they recommended that you back it up.

- We agree that's a good idea.

- You also need to "backup your certificate" in order to be able to get it into Thunderbird for use in email.

- Therefore, launch Firefox if you aren't already running it.

In Firefox, Go to Firefox --> Preferences…



Protocols

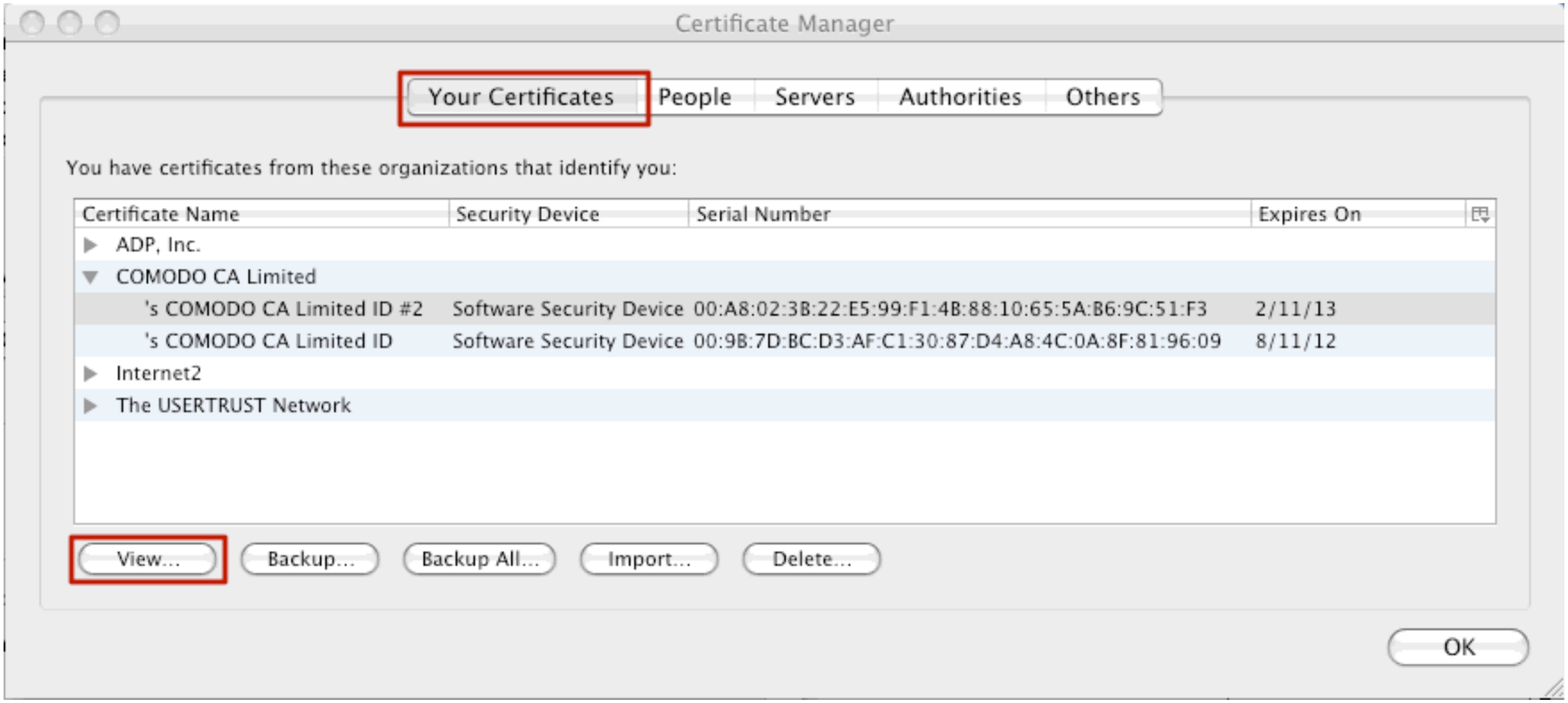
☑ Use SSL 3.0 ☑ Use TLS 1.0

Certificates

When a server requests my personal certificate:

○ Select one automatically ● Ask me every time

The Firefox Certificate Manager

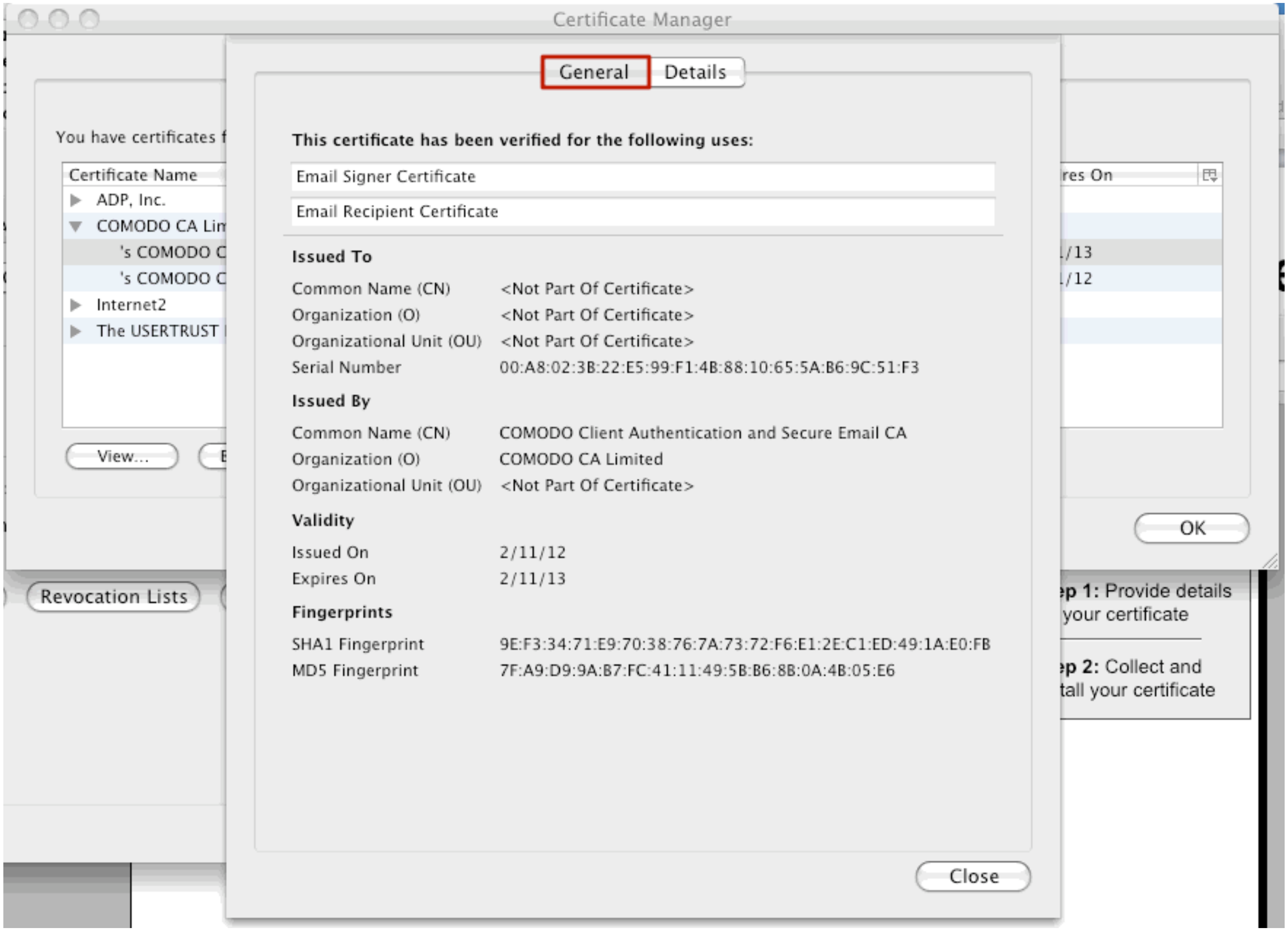


You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
▶ ADP, Inc.			
▼ COMODO CA Limited			
's COMODO CA Limited ID #2	Software Security Device	00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3	2/11/13
's COMODO CA Limited ID	Software Security Device	00:9B:7D:BC:D3:AF:C1:30:87:D4:A8:4C:0A:8F:81:96:09	8/11/12
▶ Internet2			
▶ The USERTRUST Network			

Notes: Select the "Your Certificates" tab on the Certificate Manager panel.
If necessary, hit the triangular arrow to expand the list of Comodo certificates.
You'll probably only see one certificate, the one you just got from Comodo.
But just as a matter of form, let's confirm that it really is yours…

The General Tab Tells Us When The Cert Expires



This certificate has been verified for the following uses:

Email Signer Certificate

Email Recipient Certificate

Issued To

Common Name (CN)	<Not Part Of Certificate>
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3

Issued By

Common Name (CN)	COMODO Client Authentication and Secure Email CA
Organization (O)	COMODO CA Limited
Organizational Unit (OU)	<Not Part Of Certificate>

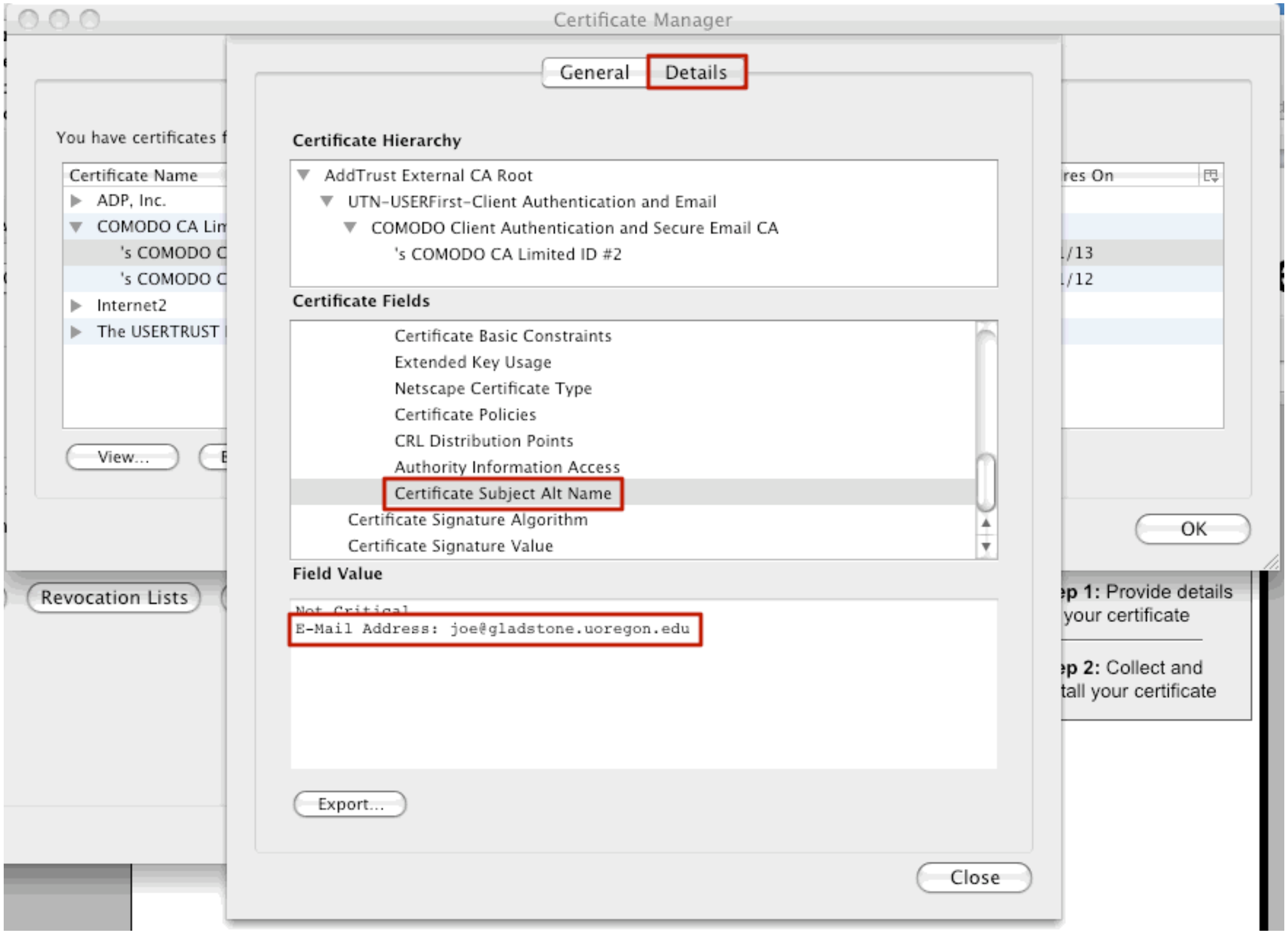
Validity

Issued On	2/11/12
Expires On	2/11/13

Fingerprints

SHA1 Fingerprint	9E:F3:34:71:E9:70:38:76:7A:73:72:F6:E1:2E:C1:ED:49:1A:E0:FB
MD5 Fingerprint	7F:A9:D9:9A:B7:FC:41:11:49:5B:B6:8B:0A:4B:05:E6

The Details "View Cert" Tab Will Let Us See The Email Address Associated With Our New Cert



Certificate Hierarchy

```
▼ AddTrust External CA Root
  ▼ UTN-USERFirst-Client Authentication and Email
    ▼ COMODO Client Authentication and Secure Email CA
        's COMODO CA Limited ID #2
```

Certificate Fields

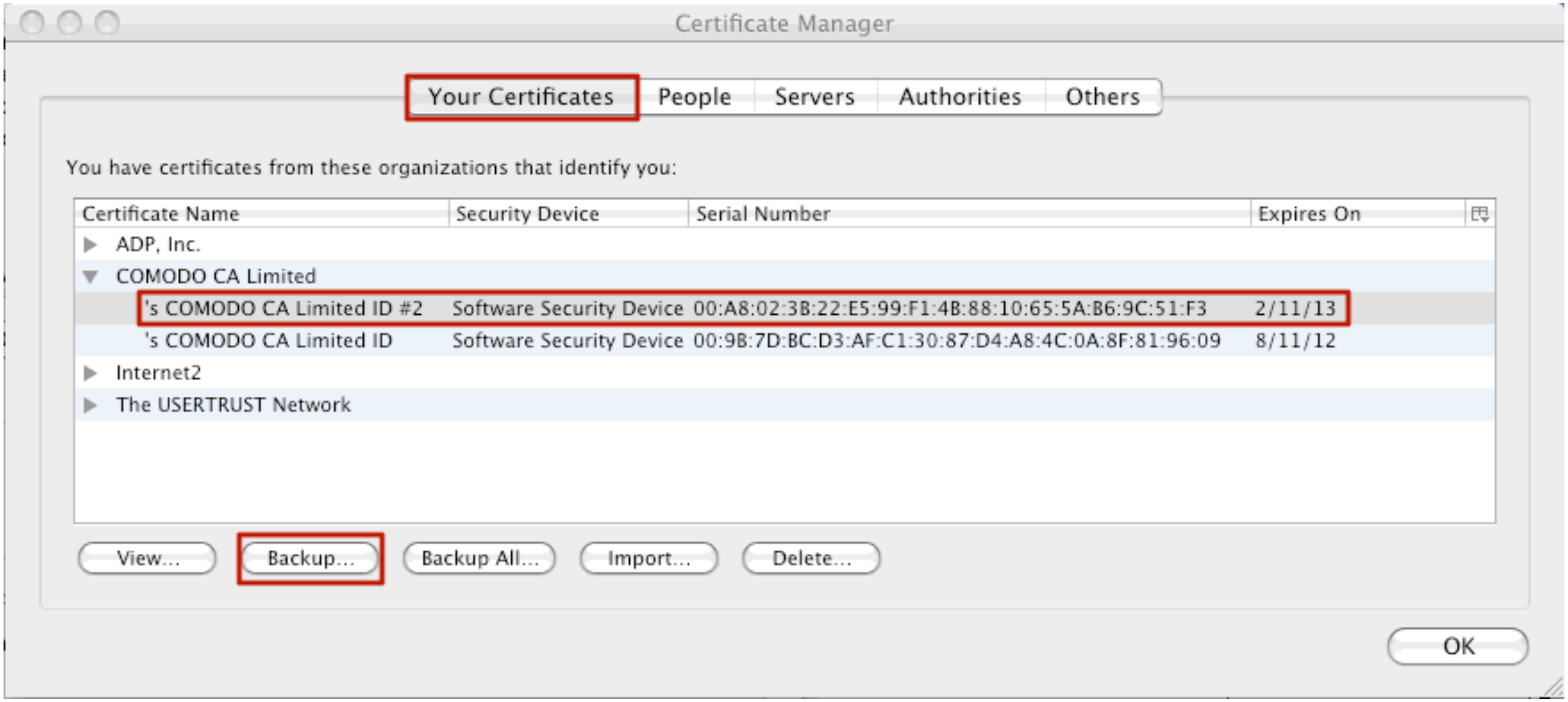
```
        Certificate Basic Constraints
        Extended Key Usage
        Netscape Certificate Type
        Certificate Policies
        CRL Distribution Points
        Authority Information Access
        Certificate Subject Alt Name
    Certificate Signature Algorithm
    Certificate Signature Value
```

Field Value

```
Not Critical
E-Mail Address: joe@gladstone.uoregon.edu
```

[Close the "View Certificate" box when you're done looking at it]

Okay, We've Picked The "Right One," So Let's Back It Up…

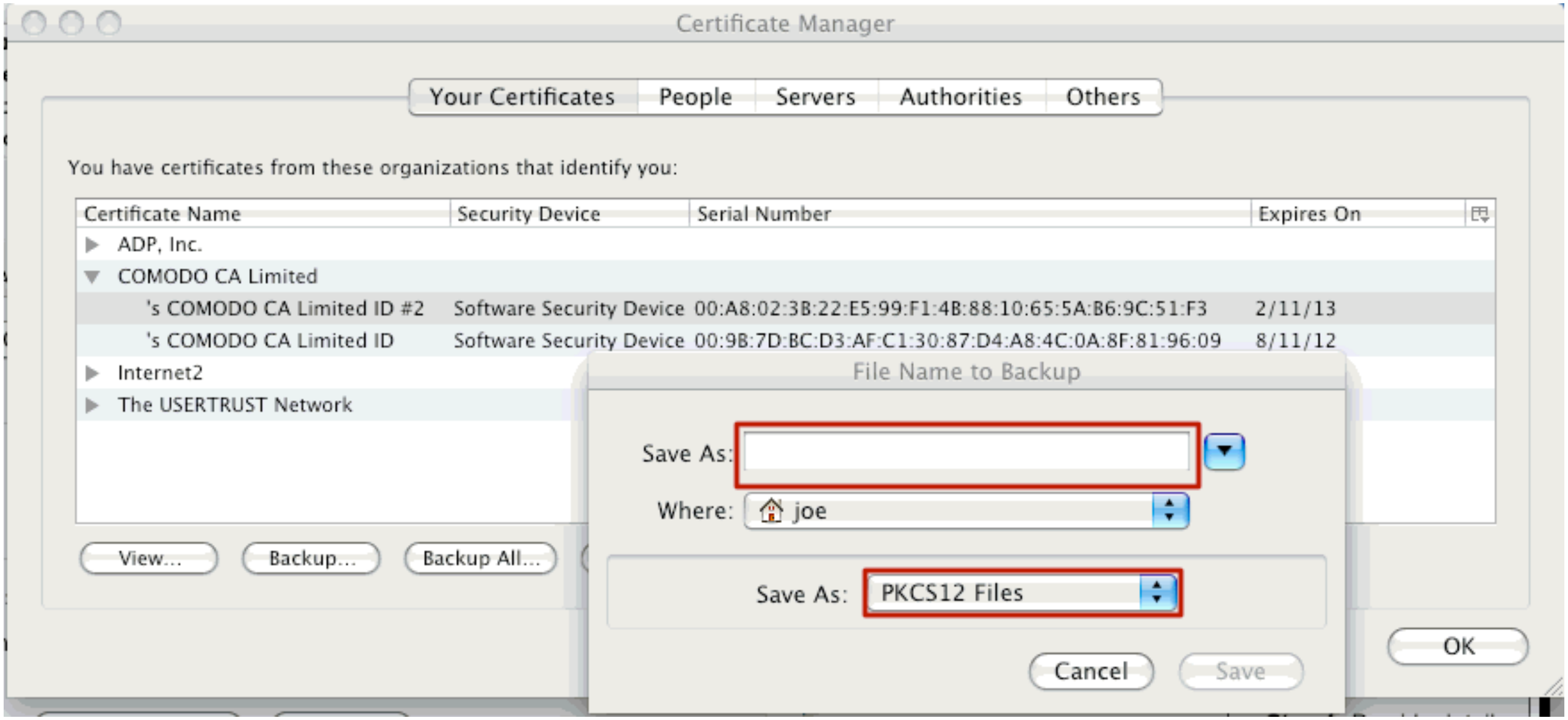


Certificate Manager

You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
▶ ADP, Inc.			
▼ COMODO CA Limited			
's COMODO CA Limited ID #2	Software Security Device	00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3	2/11/13
's COMODO CA Limited ID	Software Security Device	00:9B:7D:BC:D3:AF:C1:30:87:D4:A8:4C:0A:8F:81:96:09	8/11/12
▶ Internet2			
▶ The USERTRUST Network			

The "Name Your Backup" Dialog Box



You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
▶ ADP, Inc.			
▼ COMODO CA Limited			
's COMODO CA Limited ID #2	Software Security Device	00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3	2/11/13
's COMODO CA Limited ID	Software Security Device	00:9B:7D:BC:D3:AF:C1:30:87:D4:A8:4C:0A:8F:81:96:09	8/11/12
▶ Internet2			
▶ The USERTRUST Network			

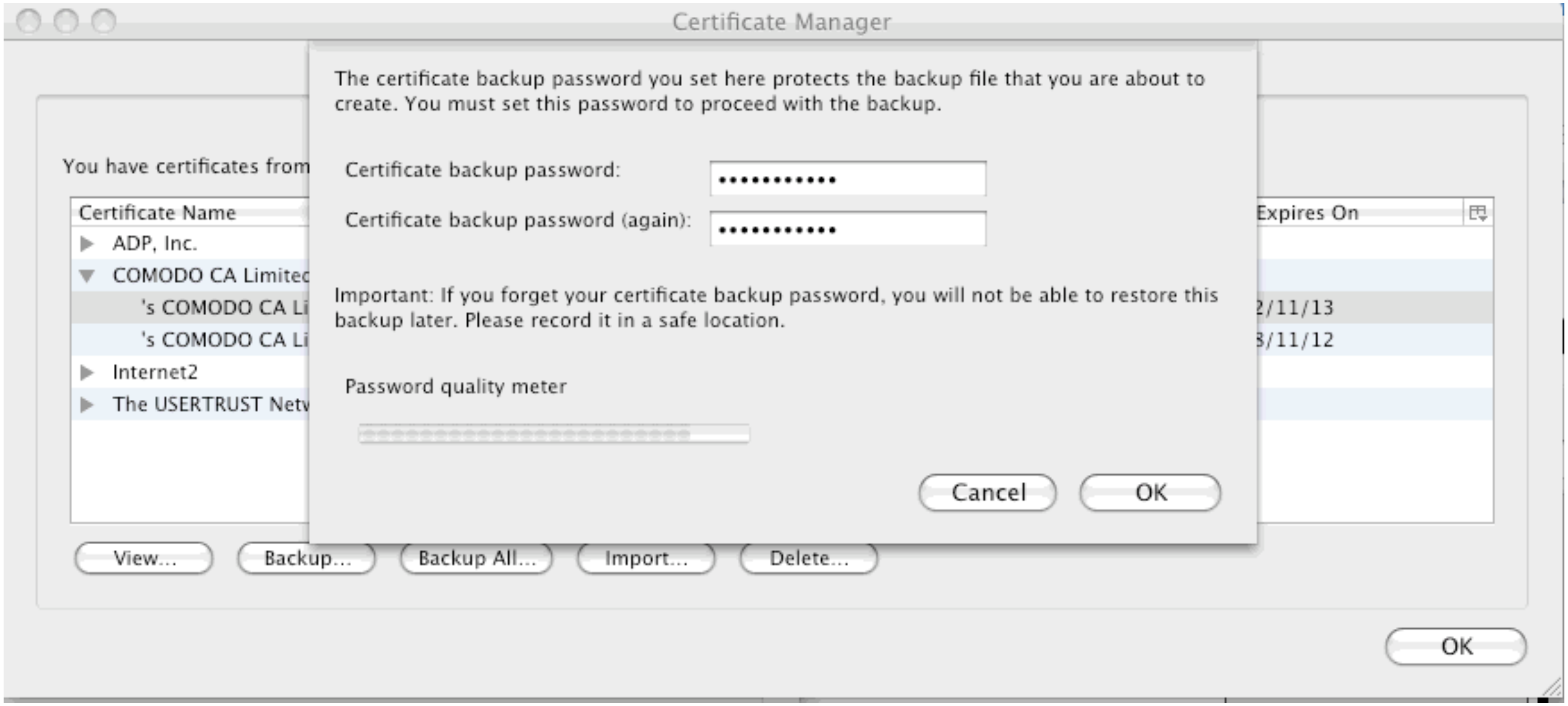
Save As:

Where: joe

Save As: PKCS12 Files

Pick a name for your certificate backup file.
It should end with a .p12 file extension.
For example, you might call this file *mycertbackup.p12*
Be sure you save it as a PKCS12 type file.

The Firefox Cert Manager Backup-Password Dialog Box



The certificate backup password you set here protects the backup file that you are about to create. You must set this password to proceed with the backup.

Certificate backup password: ••••••••••••

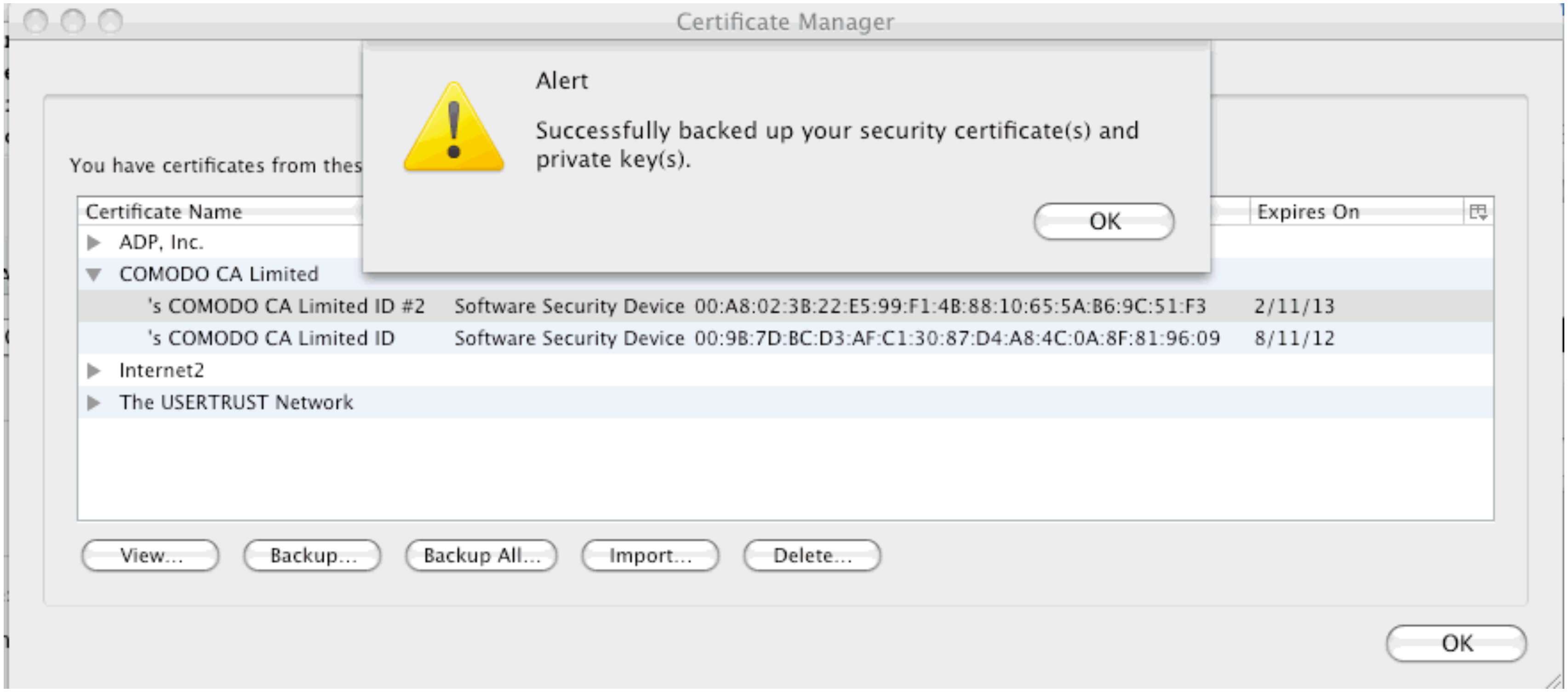
Certificate backup password (again): ••••••••••••

Important: If you forget your certificate backup password, you will not be able to restore this backup later. Please record it in a safe location.

Pick a strong password to secure your cert backup file.

PLEASE DO **NOT** FORGET THAT PASSWORD! YOU WILL NEED IT!

Backup Successful…



Alert

Successfully backed up your security certificate(s) and private key(s).

You have certificates from thes

Certificate Name			Expires On
ADP, Inc.			
COMODO CA Limited			
's COMODO CA Limited ID #2	Software Security Device	00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3	2/11/13
's COMODO CA Limited ID	Software Security Device	00:9B:7D:BC:D3:AF:C1:30:87:D4:A8:4C:0A:8F:81:96:09	8/11/12
Internet2			
The USERTRUST Network			

Note that you should save a copy of your backup to a CD, a thumb drive, or some external device just in case you lose your system, your drive crashes, etc.

VII. Importing Your Certificate Into Thunderbird

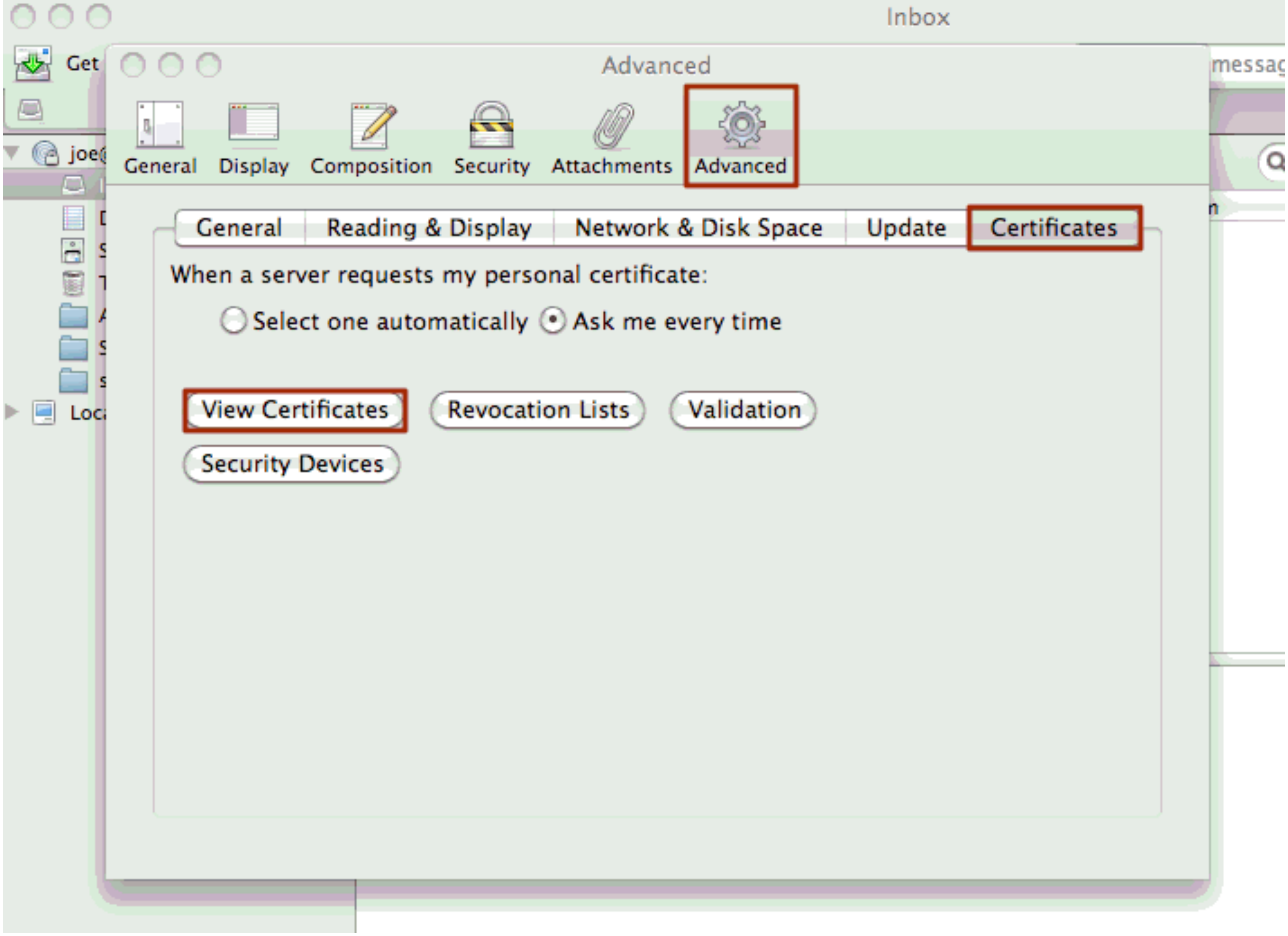
We're Now Going To Import Our New Certificate Into Thunderbird

- While there are many different popular email clients, we're going to show you how to import your client cert into Thunderbird. (Later we'll also explain how to use Outlook, and how to use client certs in Gmail web email with Penango, but for now, we're going to focus on Thunderbird)
- If you don't already have Thunderbird, and you'd like to get and install it now, you can get it for free from: http://www.mozilla.org/en-US/thunderbird/
- Note that Thunderbird has an automated installation wizard that should be able to correctly configure itself in most cases. **A caution to any non-technical person looking at these slides later: in setting up your account, choose IMAP (and *NOT* POP) for your account type! If you select POP, you may download (and then delete) all the mail that you've had stored on your account!**

"Why Can't Thunderbird Just Use The Cert That I've Already Got Installed in Firefox? They're Both Mozilla Applications, Aren't They?"

- Yes, both Firefox and Thunderbird ARE from Mozilla.
- While some applications rely on certificates stored centrally in a single operating-system-provided certificate store (e.g., in the "keychain" on the Mac), Firefox and Thunderbird do NOT do this.
- Firefox and Thunderbird use separate per-application certificate stores, instead. This gives users the flexibility to tailor what certs get potentially shown to each such application, but the downside is a slightly more complicated initial setup (you need to install your new certificate in multiple locations)
- For what it may be worth, at least Thunderbird's preferences should look very familiar to you after looking at Firefox's

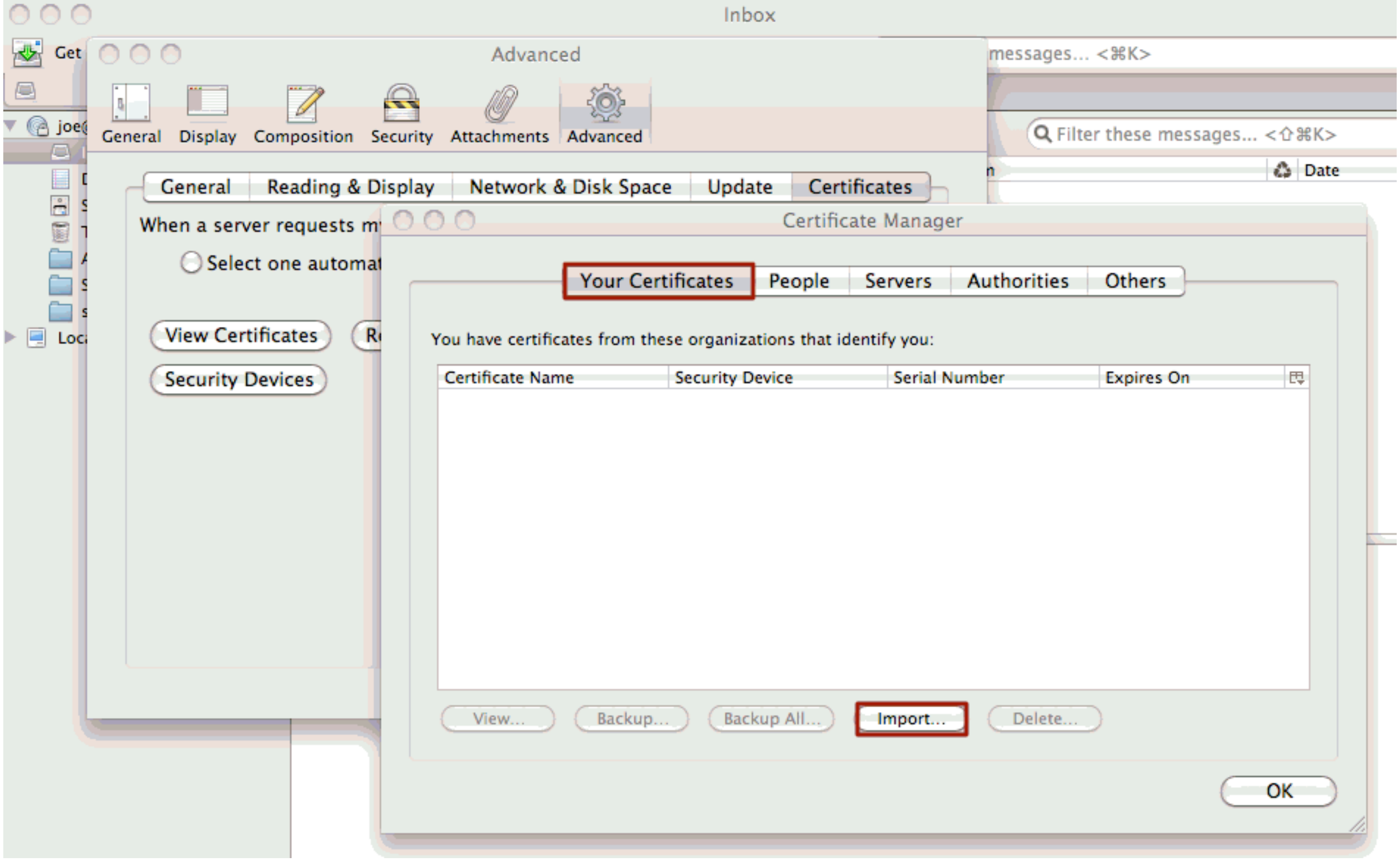
In Thunderbird, Go to Thunderbird --> Preferences…



When a server requests my personal certificate:

○ Select one automatically ● Ask me every time

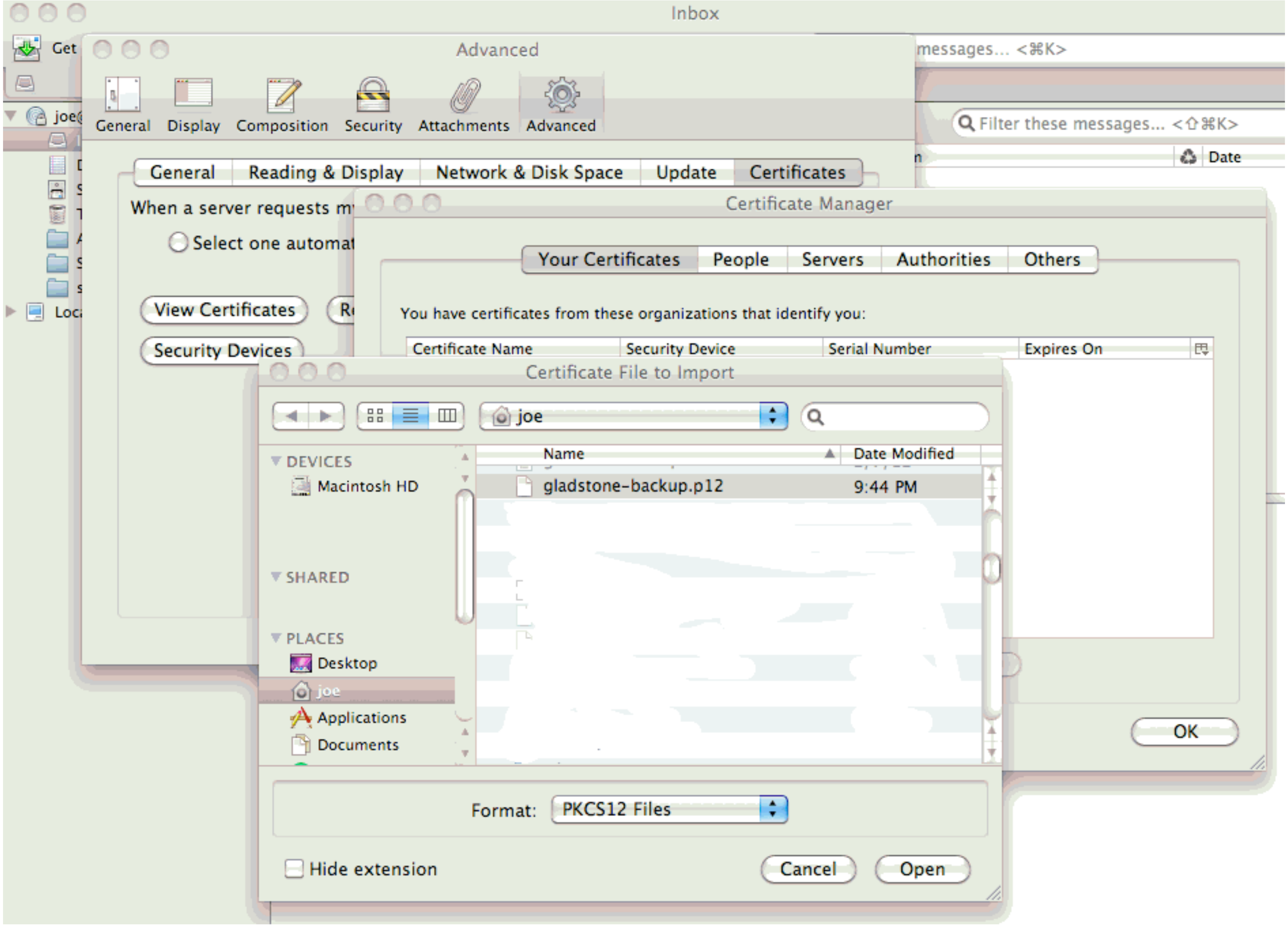
In The Certificate Manager, "Your Certificates" Tab, Click on Import



You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On

Select The .p12 Backup File You Want To Import



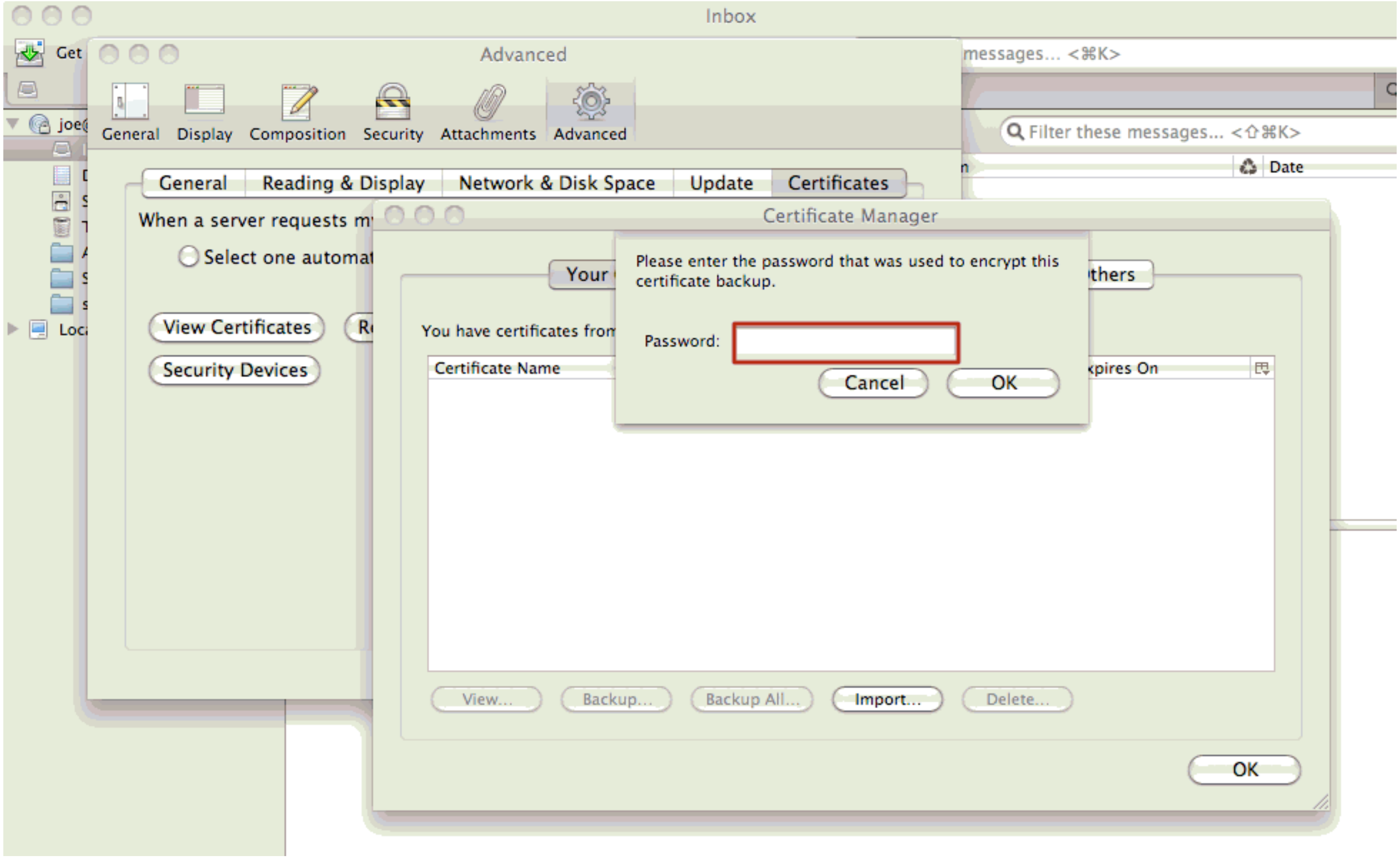
You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On

Name	Date Modified
gladstone-backup.p12	9:44 PM

Format: PKCS12 Files

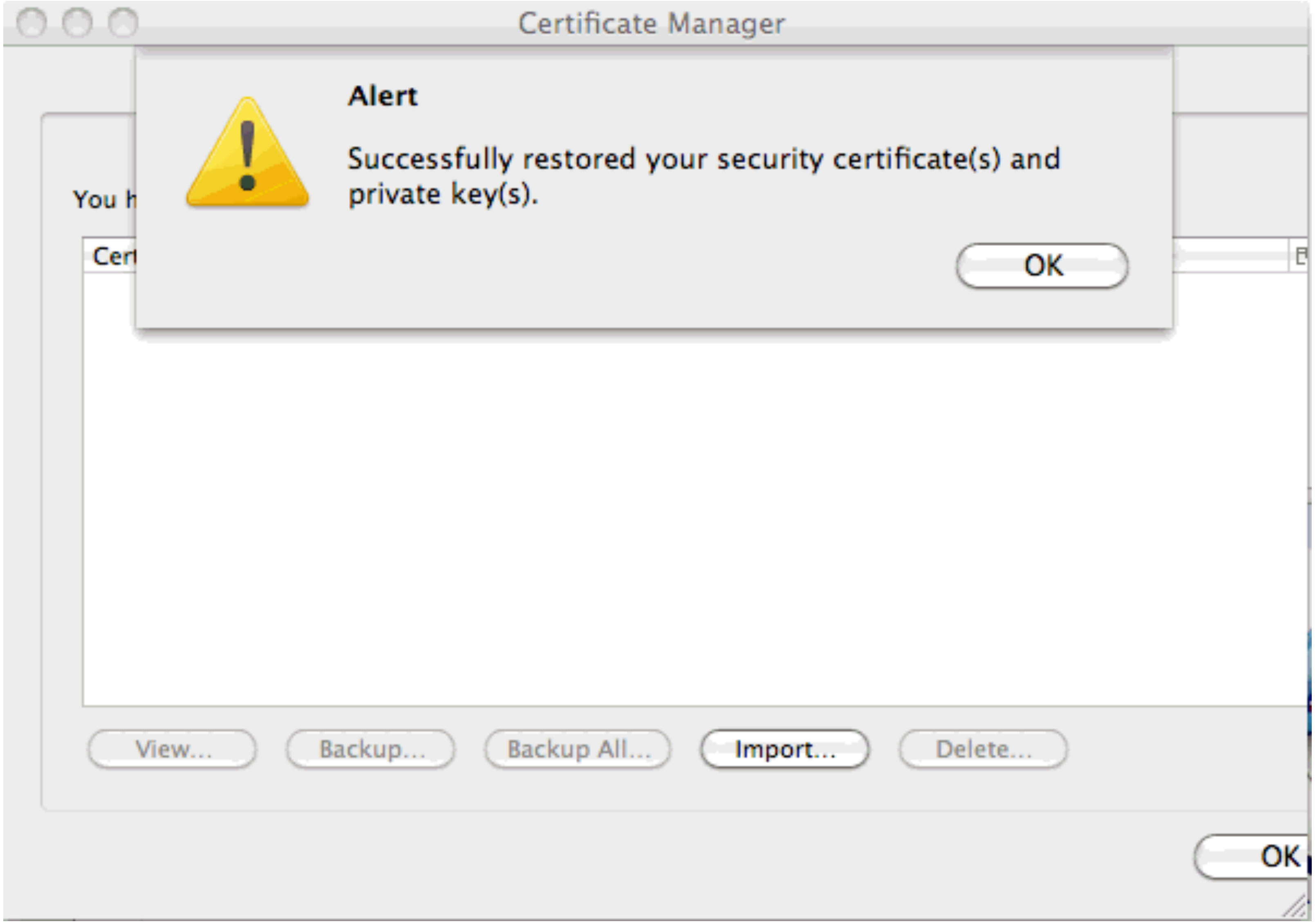
Supply the Password You Used for The Cert Backup



Please enter the password that was used to encrypt this certificate backup.

Password:

Successful Importation of The Cert Into Thunderbird

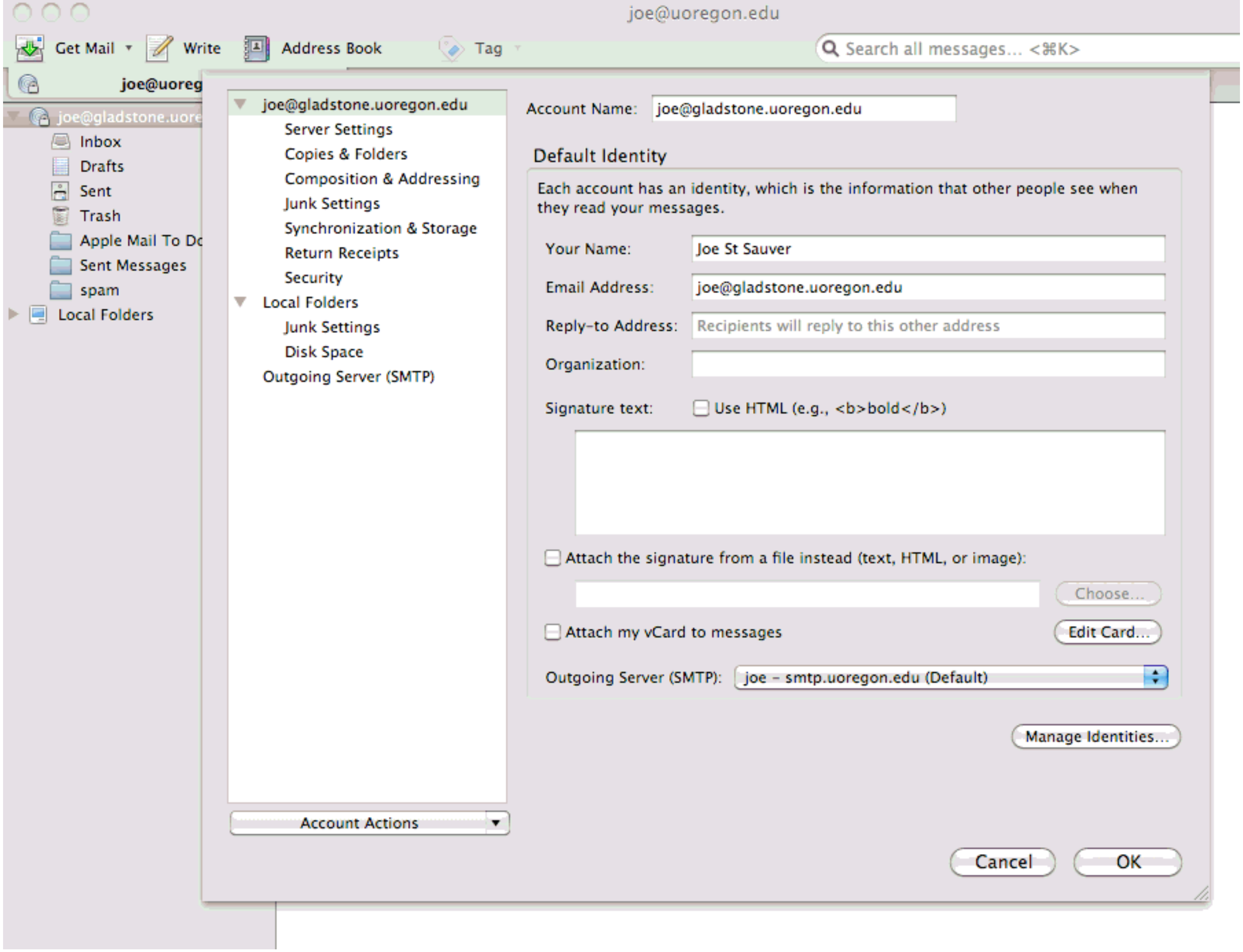


Alert

Successfully restored your security certificate(s) and private key(s).

VIII. In Thunderbird, Associate Your Certificate With Your Email Account And Configure Thunderbird To Do Digital Signing

Thunderbird: Tools --> Account Settings



joe@gladstone.uoregon.edu
- Server Settings
- Copies & Folders
- Composition & Addressing
- Junk Settings
- Synchronization & Storage
- Return Receipts
- Security

Local Folders
- Junk Settings
- Disk Space

Outgoing Server (SMTP)

Account Actions

Account Name: joe@gladstone.uoregon.edu

Default Identity

Each account has an identity, which is the information that other people see when they read your messages.

Your Name: Joe St Sauver

Email Address: joe@gladstone.uoregon.edu

Reply-to Address: Recipients will reply to this other address

Organization:

Signature text: □ Use HTML (e.g., <b>bold</b>)

□ Attach the signature from a file instead (text, HTML, or image):

Choose...

□ Attach my vCard to messages

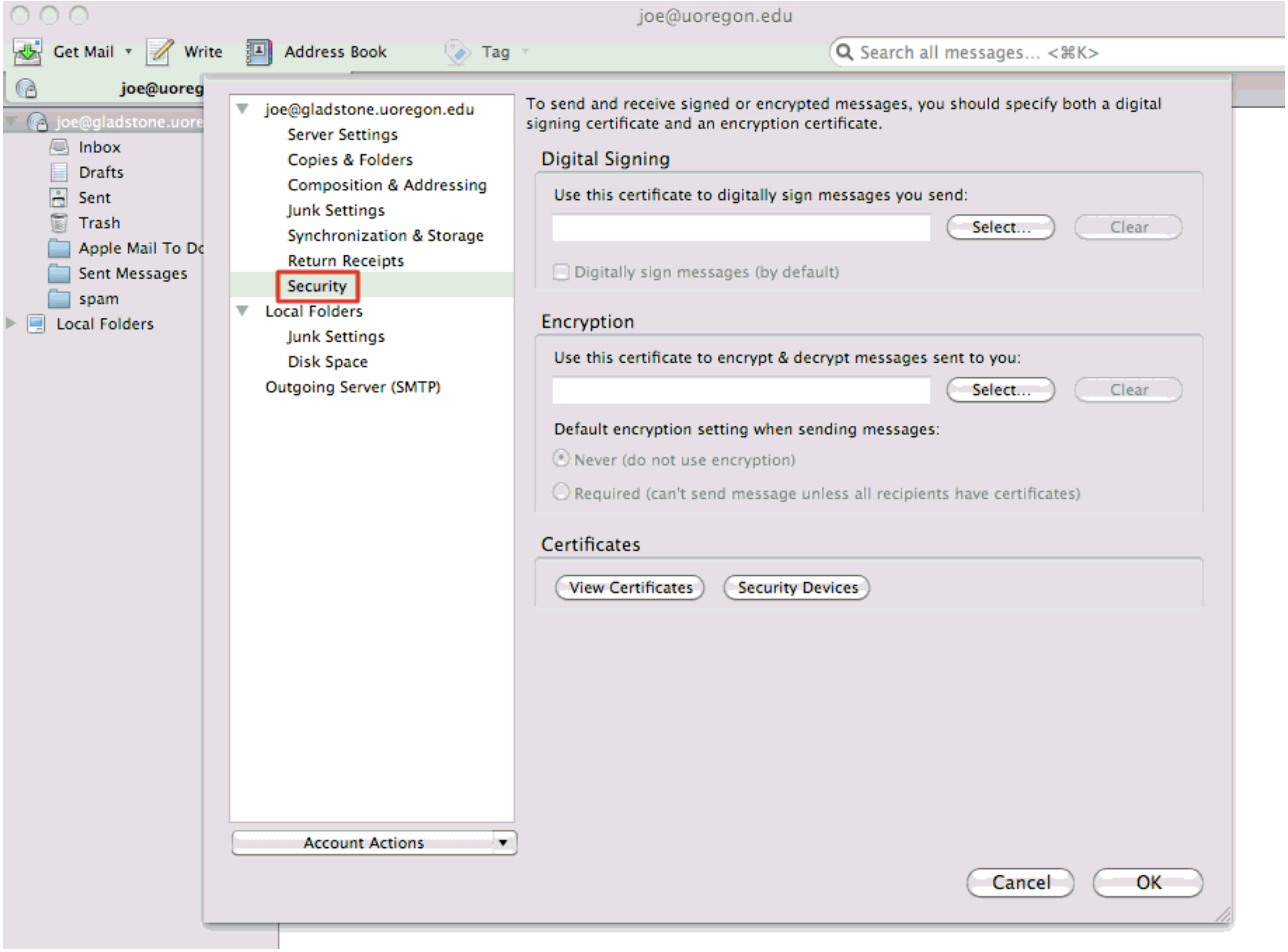
Edit Card...

Outgoing Server (SMTP): joe - smtp.uoregon.edu (Default)

Manage Identities...

Cancel OK

Security



To send and receive signed or encrypted messages, you should specify both a digital signing certificate and an encryption certificate.

Digital Signing

Use this certificate to digitally sign messages you send:

□ Digitally sign messages (by default)

Encryption

Use this certificate to encrypt & decrypt messages sent to you:

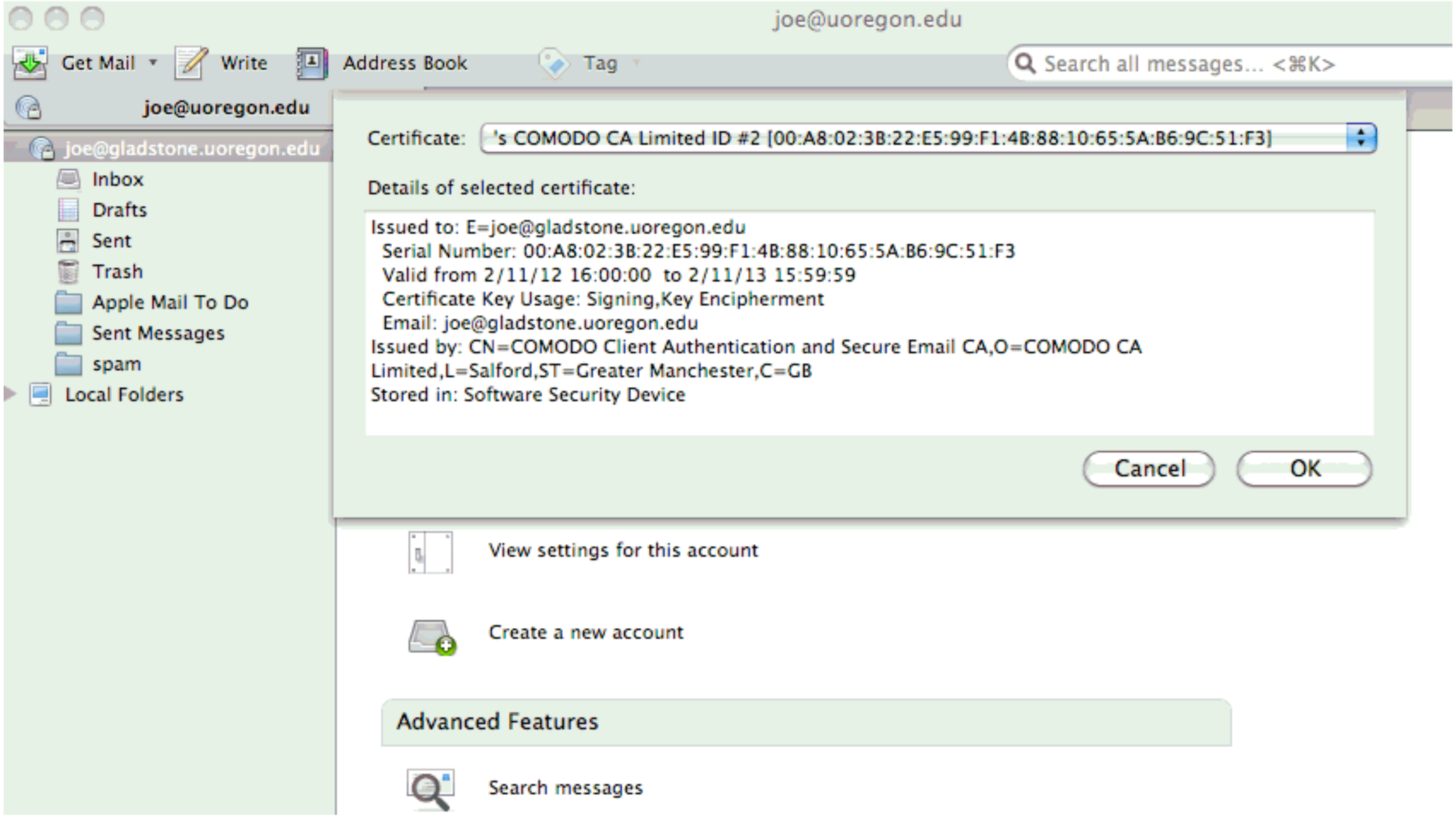
Default encryption setting when sending messages:

● Never (do not use encryption)

○ Required (can't send message unless all recipients have certificates)

Certificates

Select The Cert You Want To Use For Digital Signing

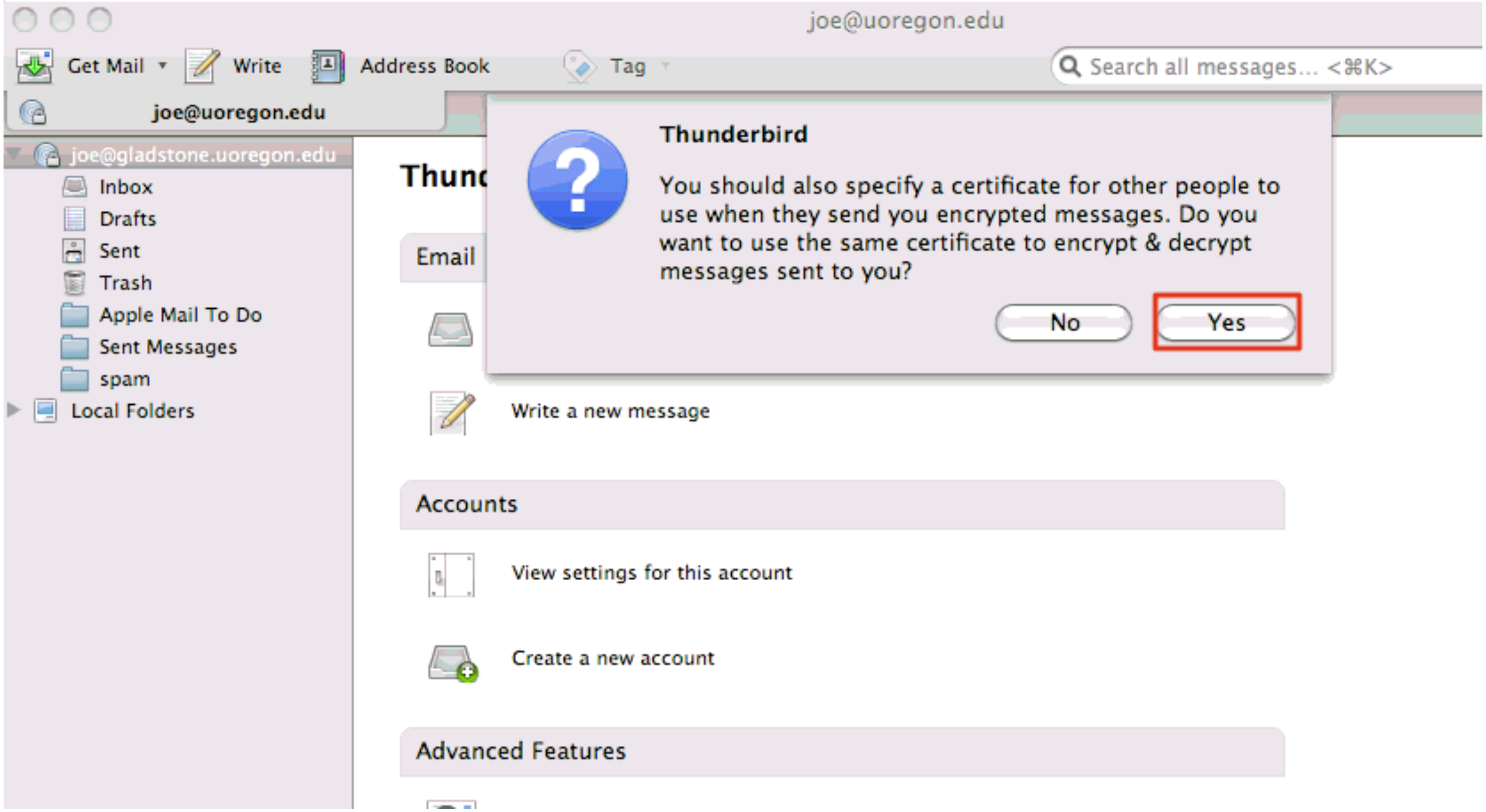


Certificate: 's COMODO CA Limited ID #2 [00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3]

Details of selected certificate:

```
Issued to: E=joe@gladstone.uoregon.edu
  Serial Number: 00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3
  Valid from 2/11/12 16:00:00  to 2/11/13 15:59:59
  Certificate Key Usage: Signing,Key Encipherment
  Email: joe@gladstone.uoregon.edu
Issued by: CN=COMODO Client Authentication and Secure Email CA,O=COMODO CA
Limited,L=Salford,ST=Greater Manchester,C=GB
Stored in: Software Security Device
```

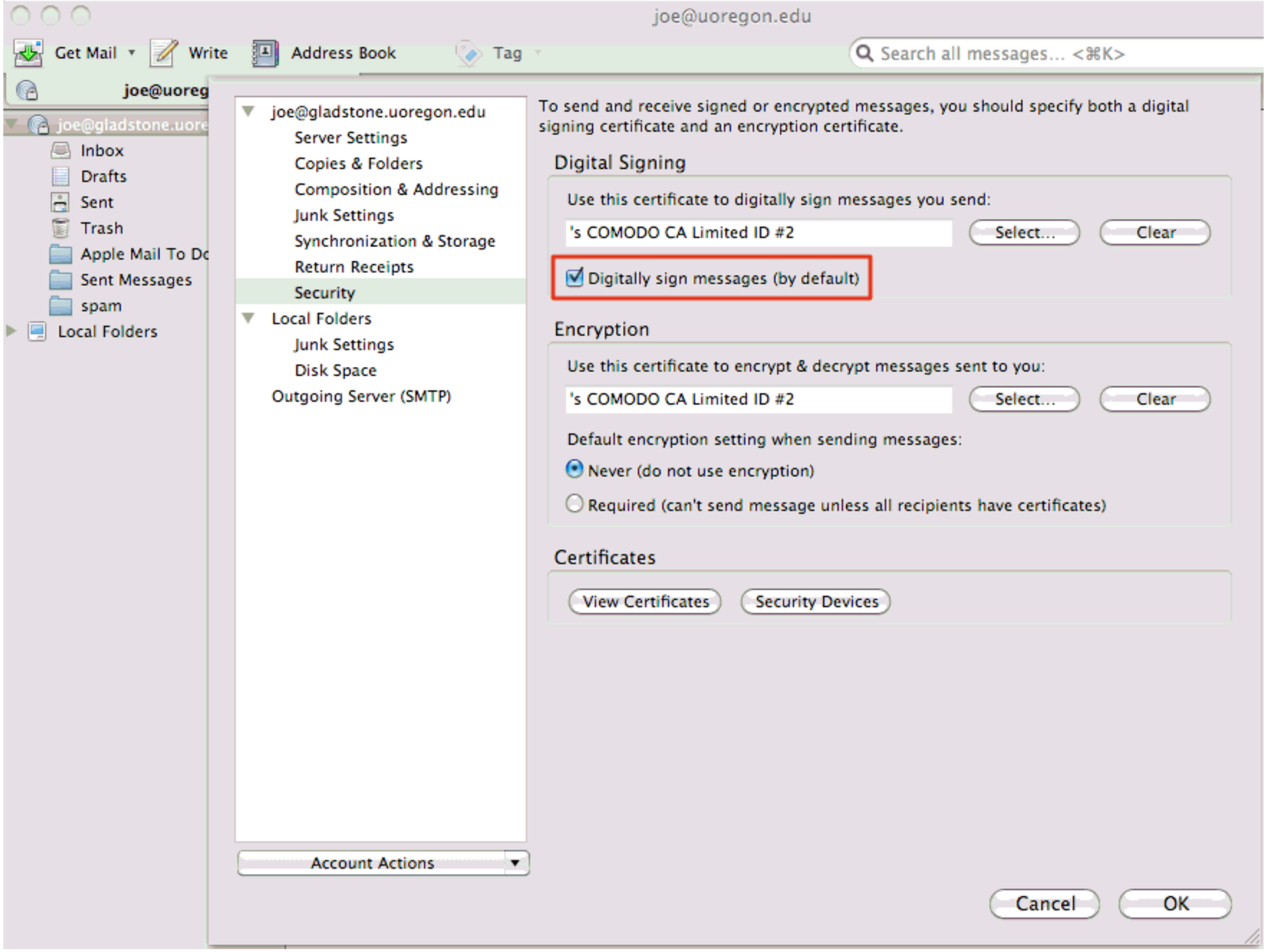
Confirm That You Want To Also Use That Same Cert for Encrypting/Decrypting Messages



Thunderbird

You should also specify a certificate for other people to use when they send you encrypted messages. Do you want to use the same certificate to encrypt & decrypt messages sent to you?

Make Sure You're Set To Digitally Sign Your Messages By Default



To send and receive signed or encrypted messages, you should specify both a digital signing certificate and an encryption certificate.

Digital Signing

Use this certificate to digitally sign messages you send: 's COMODO CA Limited ID #2

☑ Digitally sign messages (by default)

Encryption

Use this certificate to encrypt & decrypt messages sent to you: 's COMODO CA Limited ID #2

Default encryption setting when sending messages:

● Never (do not use encryption)

○ Required (can't send message unless all recipients have certificates)

Certificates

Thunderbird Configuration Is Now Complete…

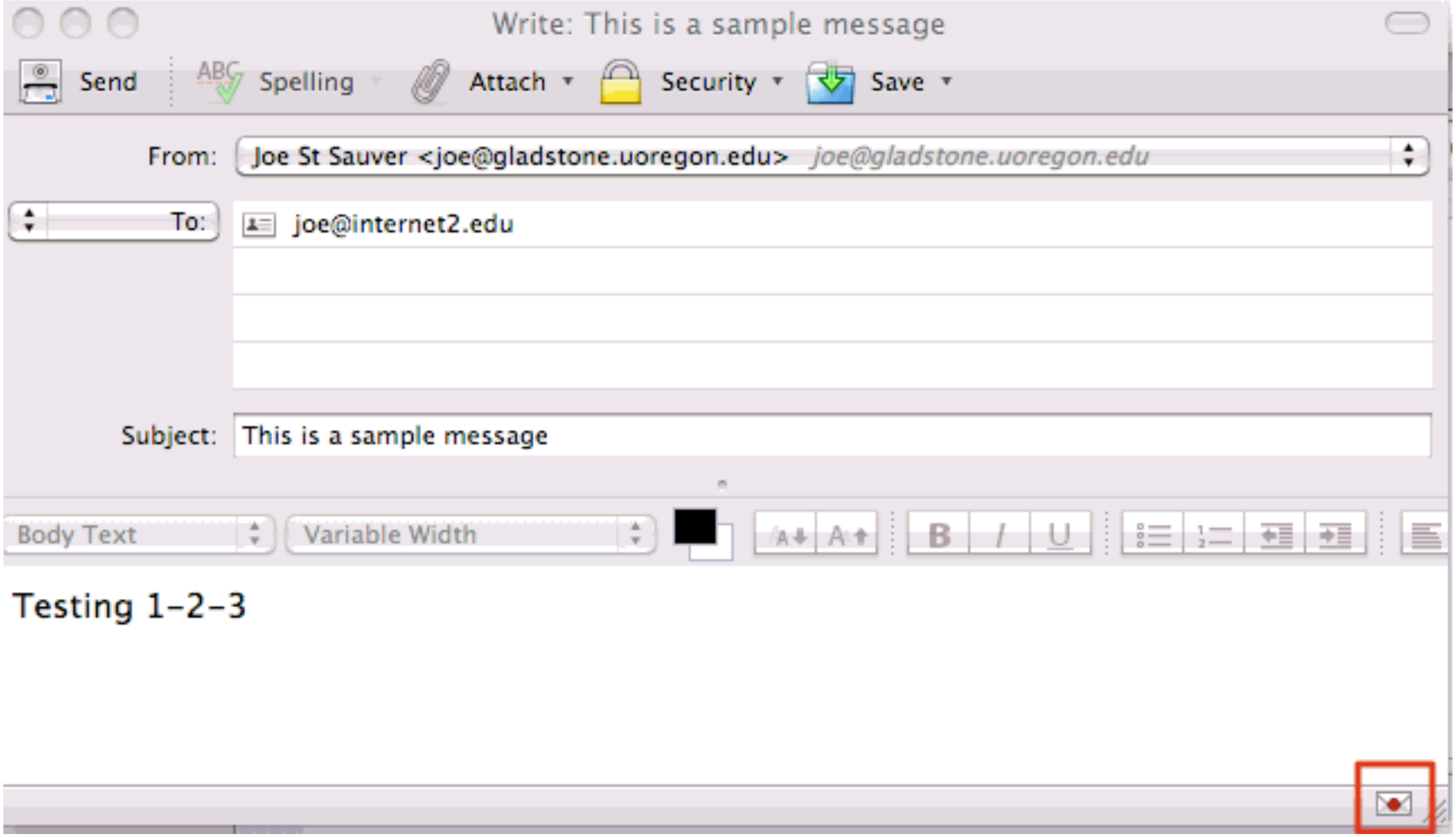
- The hard part is over! You are now set to automatically digitally sign your Thunderbird email messages by default.

- And the good part is that now that you've got yourself successfully configured, you won't have to screw around with any of this for roughly a year (e.g., until just before your free Comodo personal certificate is close to expiring)

- Huzzah!

IX. Digitally Signing A Message In Thunderbird

Start Writing A Message The Way You Normally Would



From: Joe St Sauver <joe@gladstone.uoregon.edu> joe@gladstone.uoregon.edu

To: joe@internet2.edu

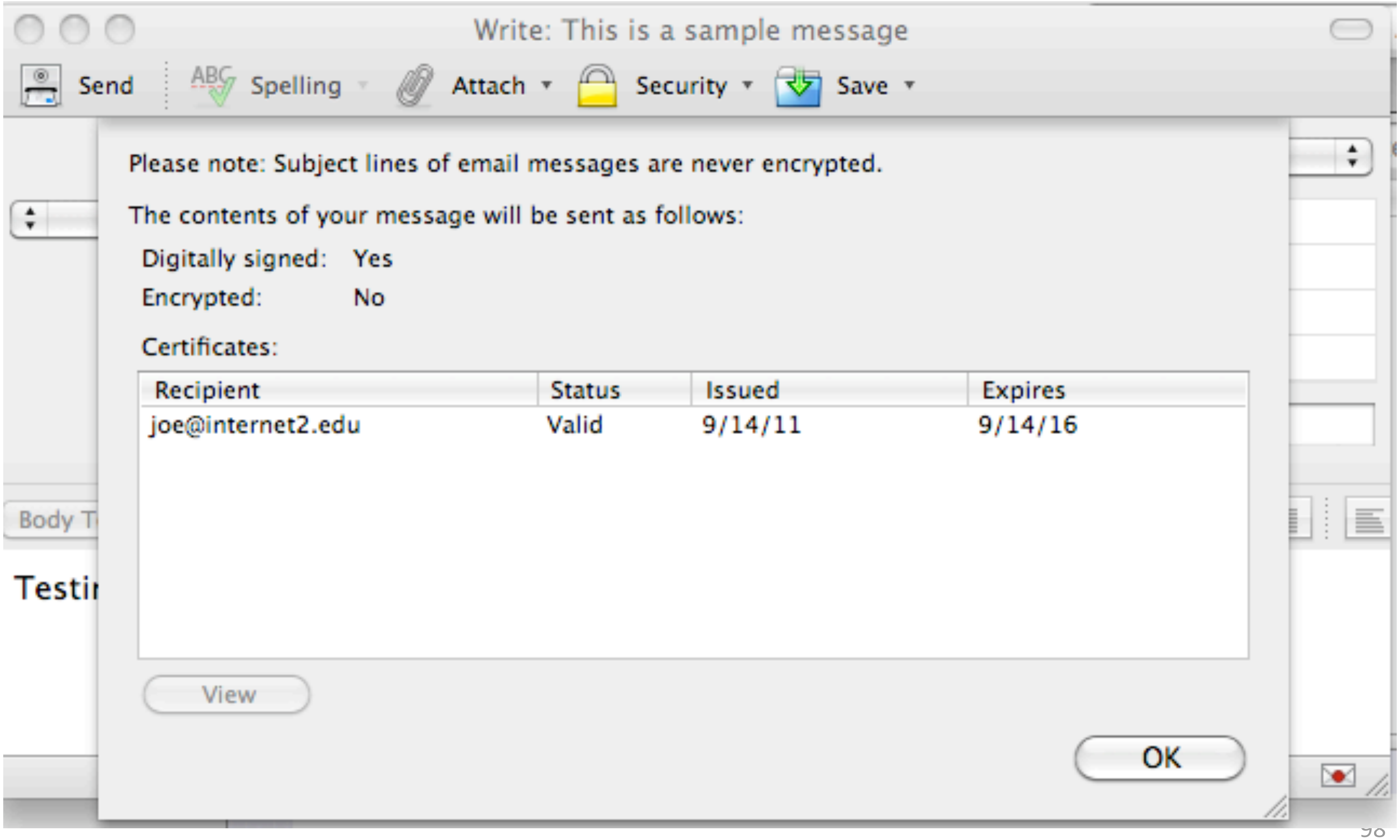
Subject: This is a sample message

Testing 1-2-3

NOTE THE "DIGITALLY SIGNED" SEAL AT THE BOTTOM RIGHT CORNER!

Optional: Confirm That The Message *Will* Be Signed

Click On The Padlock Icon On The Bar Or The Little Red Seal In The Bottom Right Corner If You Ever Want To Double Check!



Please note: Subject lines of email messages are never encrypted.

The contents of your message will be sent as follows:

Digitally signed: Yes
Encrypted: No

Certificates:

Recipient	Status	Issued	Expires
joe@internet2.edu	Valid	9/14/11	9/14/16

Proceed to Send Your Message

- … just like you normally would. It will automatically be digitally signed with your certificate.

- Your recipients will see your normal message, plus an additional “p7s” attachment that will have your public key/certificate. (no, that's not malware :-))

- If your correspondent’s email client supports S/MIME, it will automatically check and validate your digital signature.

- If your correspondent’s email client doesn’t support S/MIME, they can just safely ignore the extra p7s attachment.

X. Encrypting A Message In Thunderbird

Signing vs. Encrypting

- Digitally signed messages establish who prepared the body of the message, but anyone can still *read* that message: it’s cryptographically *signed*, it’s not *encrypted*.

- If the body of your message is sensitive, you may also want to consider encrypting it so that only the intended recipient (or someone with access to his private key) can read it.

- Oh, and it goes without saying that a message can be both signed AND encrypted, if that's appropriate.

Getting The Public Key of Your Correspondent

- To encrypt a message you'll need your **correspondent's** public key.

- But how will you get his public key? Answer: you'll have the recipient send you a digitally signed message, first.

- Your email client will automatically extract the public key and cert it needs from that digitally signed message you received from him.

- If digital certs are deployed throughout your enterprise, you may also be able to get public keys and client certs for your correspondents from your enterprise directory, but that model falls apart when you attempt to extend it Internet-wide.

A Meta Question: ***Should*** I Encrypt The Mail I Send?

- Maybe yes, maybe no.
- First of all, note that you usually won't be ***able*** to encrypt unless your colleague is ALSO set up to do S/MIME, and your correspondent has already sent you at least one signed message (so that you'll have his public key and cert)
- If the content of your email isn't sensitive, you probably don't *need* to encrypt it. It may be "cool" to encrypt all the messages you can, but if you don't need to, you *might* want to skip it. Why?
  - Well, if you receive encrypted content, you won't be able to subsequently easily search those messages.
  - And, if you happen to lose your private key, you will be S-O-L unless you have your key backed up (and you can remember its password!), or your key has been escrowed. If your key isn't backed up or escrowed, can you *really* afford to potentially lose all the content encrypted with that key?
  - You'll drive command line email client users nuts.

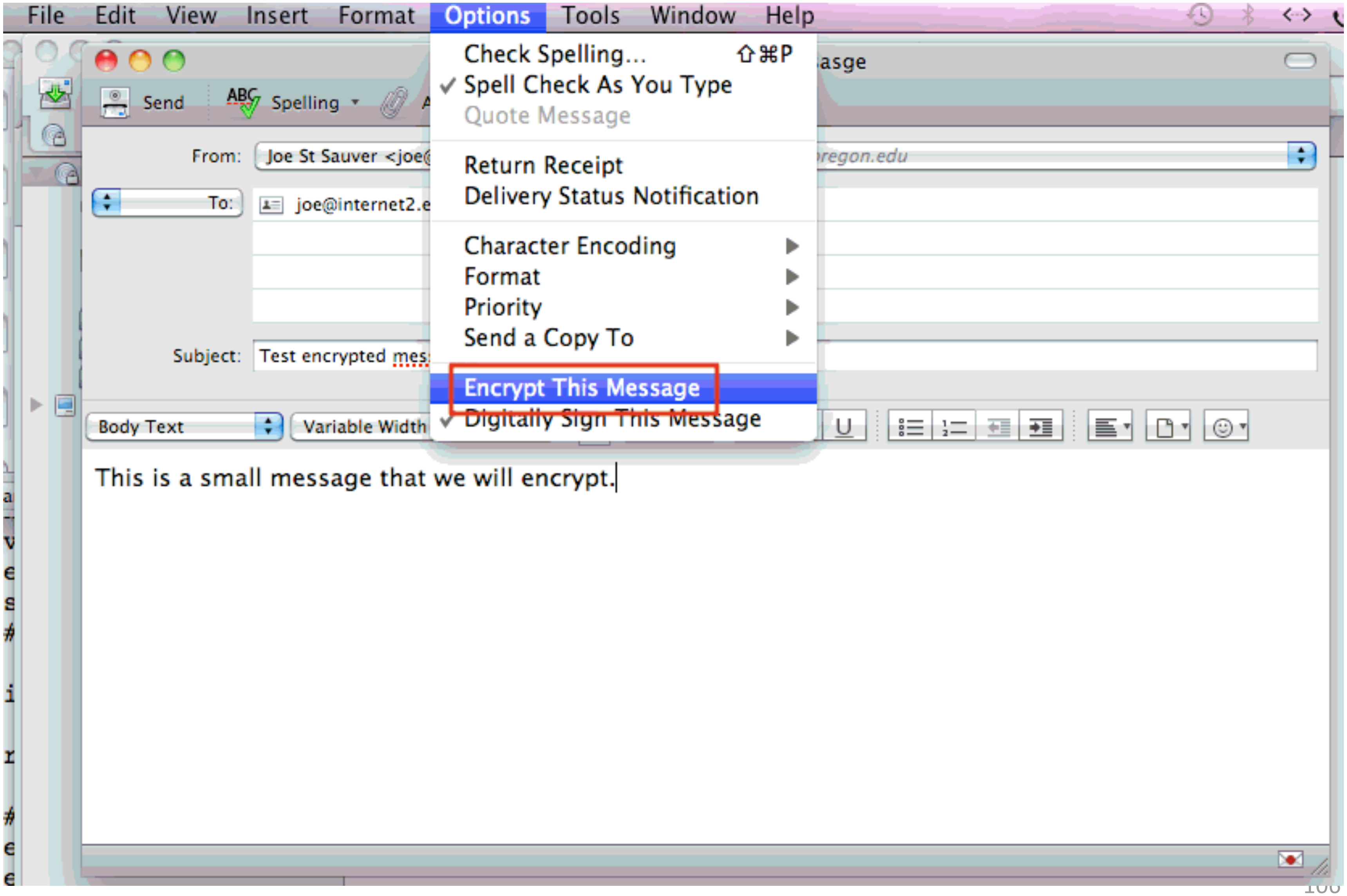
And Some Arguments In Favor of Routine Encryption

- What's not sensitive to me, might be sensitive to someone else. Likewise, it might not be sensitive NOW, but it might be sensitive LATER.
- If you *only* encrypt sensitive messages, that sure makes them stands out, doesn't it? Wouldn't it be nice if those messages were just part of a larger volume of routinely encrypted messages?
- It's relatively easy to forget to enable encryption, and to accidentally send out a sensitive message in clear text. If you routinely encrypt, that won't happen.
- If you want people to secure their email, you need to set the example and nudge them along. If they get set up to do encrypted email, but then never get any, they may feel like they're wasting their time.
- Finally, it *is* sort of cool/fun to do so. :-)

Hedging The Risk of Data Loss: Key Escrow

- Let's pretend that you have a faculty member who's doing absolutely critical (and highly sensitive) work for your school, and you want them to routinely encrypt as a result. At the same time, assume that person is overweight, has high blood pressure, drinks and smokes, crosses the street while distracted, drives without a seatbelt and lives in a gang infested neighborhood. Frankly, you worry that critical faculty person will die or be killed, or maybe just quit and start a business making home-made premium soap some day. If that happens, how will you get at all their encrypted work messages and files? Will all that work product be lost?
- Escrowing encryption keys allows you to get a copy of otherwise unavailable encryption keys in a variety of carefully predefined emergency situations. Companies normally pay extra for this "insurance." Keys recovered via escrow may have the associated cert revoked at the same time.

"It IS Worth It. I DO Want To Encrypt My Message -- How Do I Do That In Thunderbird?"



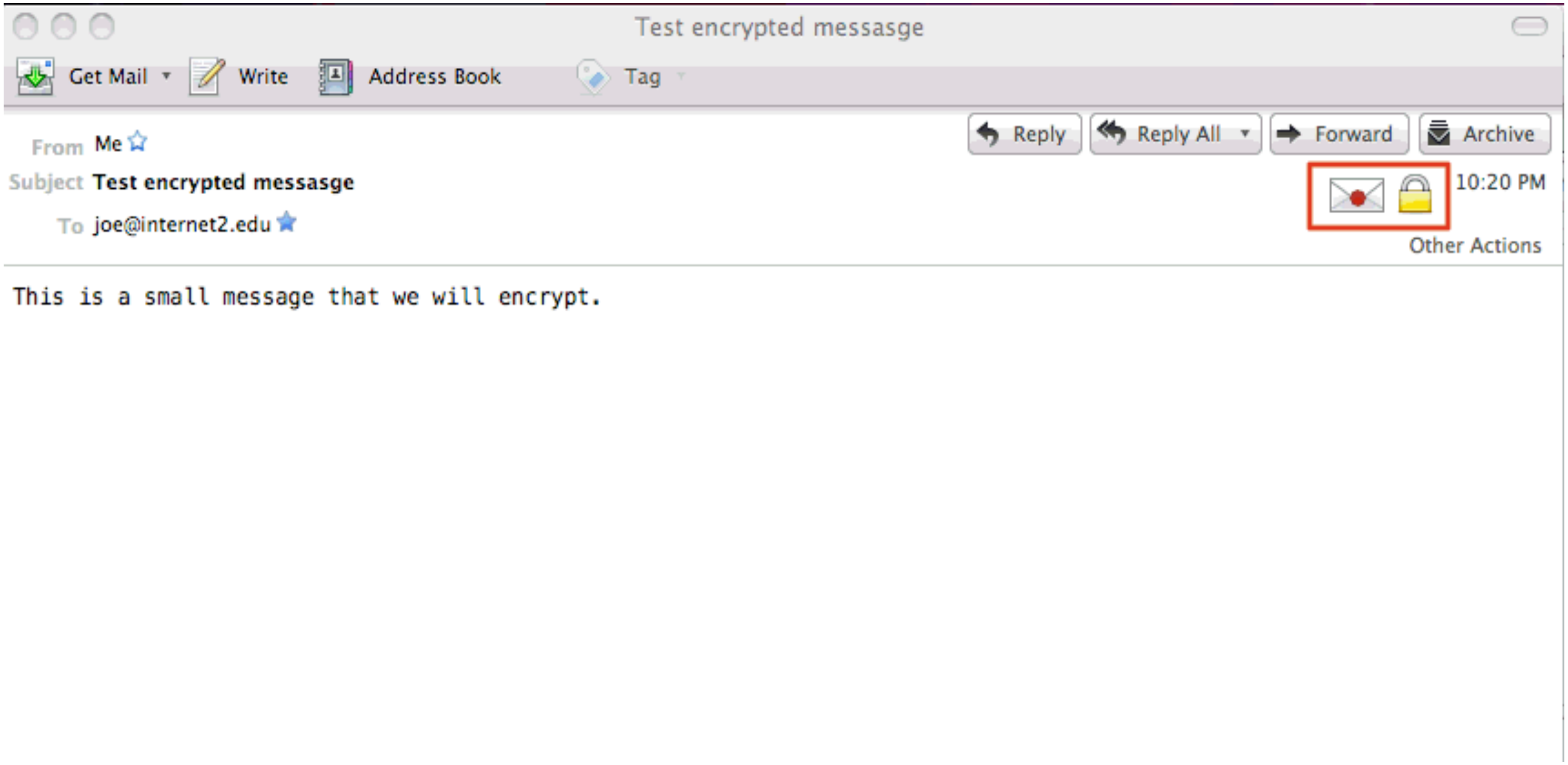
From: Joe St Sauver <joe@ ...regon.edu

To: joe@internet2.e

Subject: Test encrypted mes

This is a small message that we will encrypt.

"When I Get A Signed and Encrypted Message, What Will It Look Like?"



From Me

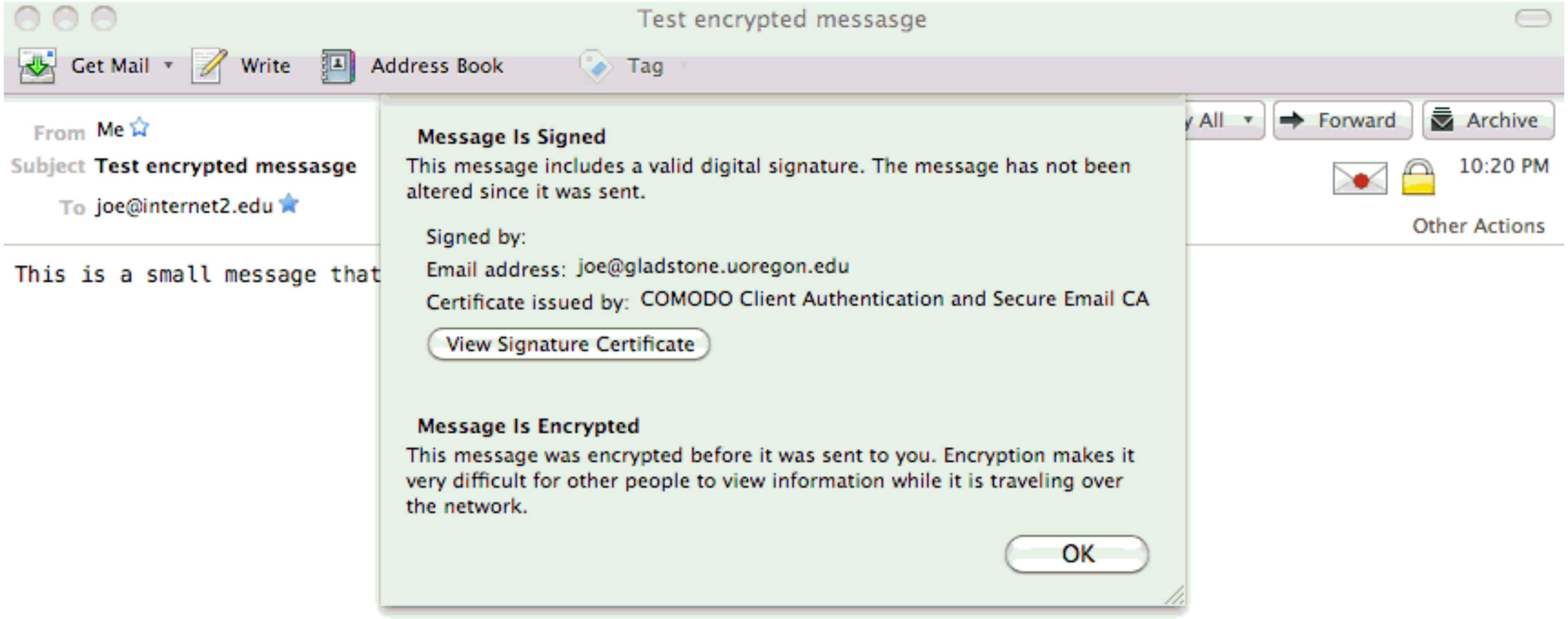
Subject Test encrypted messasge

To joe@internet2.edu

10:20 PM

```
This is a small message that we will encrypt.
```

Who <u>Signed</u> That Message? (Note: It May *Not* Be The Person Who <u>Sent</u> The Message)



From Me
Subject Test encrypted messasge
To joe@internet2.edu

This is a small message that

Message Is Signed

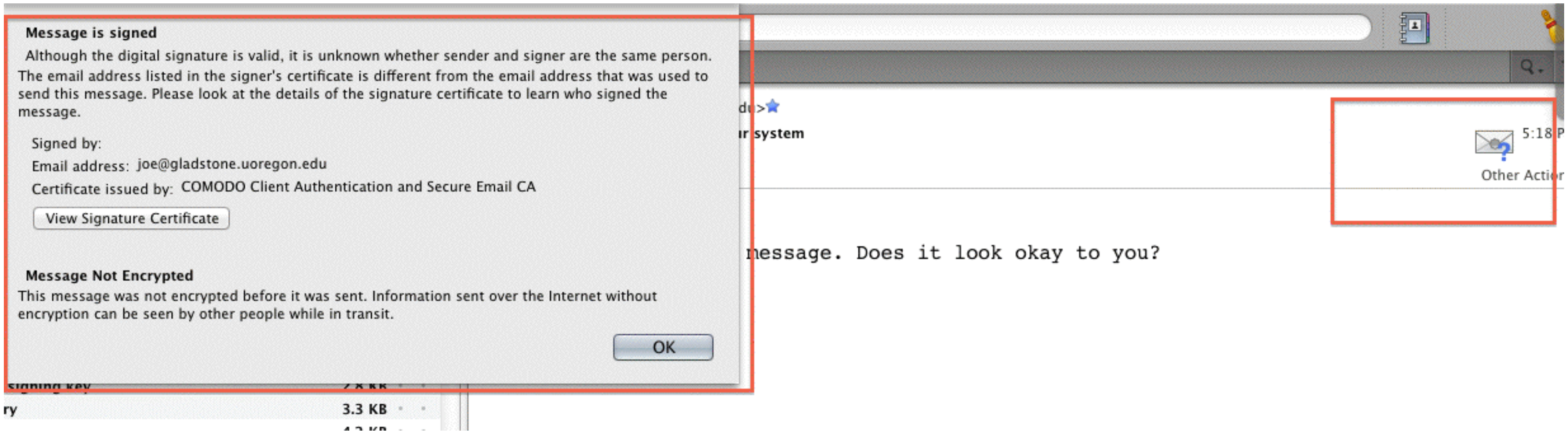
This message includes a valid digital signature. The message has not been altered since it was sent.

Signed by:
Email address: joe@gladstone.uoregon.edu
Certificate issued by: COMODO Client Authentication and Secure Email CA

Message Is Encrypted

This message was encrypted before it was sent to you. Encryption makes it very difficult for other people to view information while it is traveling over the network.

An Example of Using a Non-Matching Cert



Message is signed

Although the digital signature is valid, it is unknown whether sender and signer are the same person. The email address listed in the signer's certificate is different from the email address that was used to send this message. Please look at the details of the signature certificate to learn who signed the message.

Signed by:

Email address: joe@gladstone.uoregon.edu

Certificate issued by: COMODO Client Authentication and Secure Email CA

Message Not Encrypted

This message was not encrypted before it was sent. Information sent over the Internet without encryption can be seen by other people while in transit.

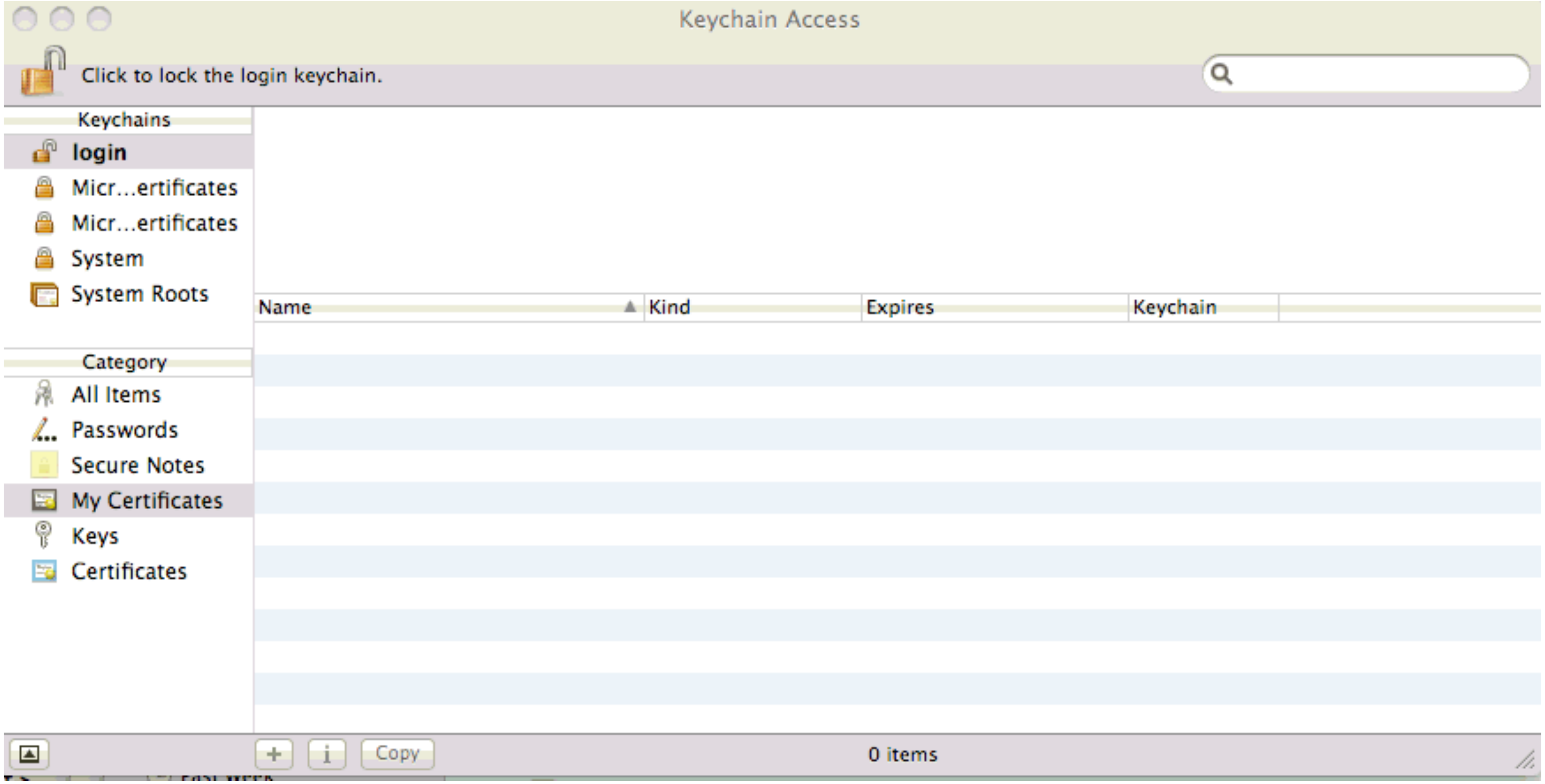
nessage. Does it look okay to you?

Additional Important S/MIME Caveats

- S/MIME encrypts the BODY of the message, **ONLY.** S/MIME DOES NOT ENCRYPT THE SUBJECT HEADER (or any other message header). Therefore, **DO NOT** put anything that needs to be kept confidential in the Subject of an encrypted message. In fact, you may want to get in the habit of never putting ANYTHING into the subject line of encrypted messages.
- Encrypted message bodies cannot be automatically scanned on the network for viruses or other malware.
- Some mailing list programs may tamper with messages by doing things like adding footers or rewriting links or stripping attachments (including p7s digital signatures). If that happens, your signature won't validate. If you send messages to mailing lists that do these sort of things, you may want to manually disable digital signing for messages to those lists.

XI. What If I Want To Use Outlook Instead of Thunderbird?

Outlook On Apple OS X Uses the Apple Keychain; To Do S/MIME with Outlook, We Need To Get Our Cert Into It

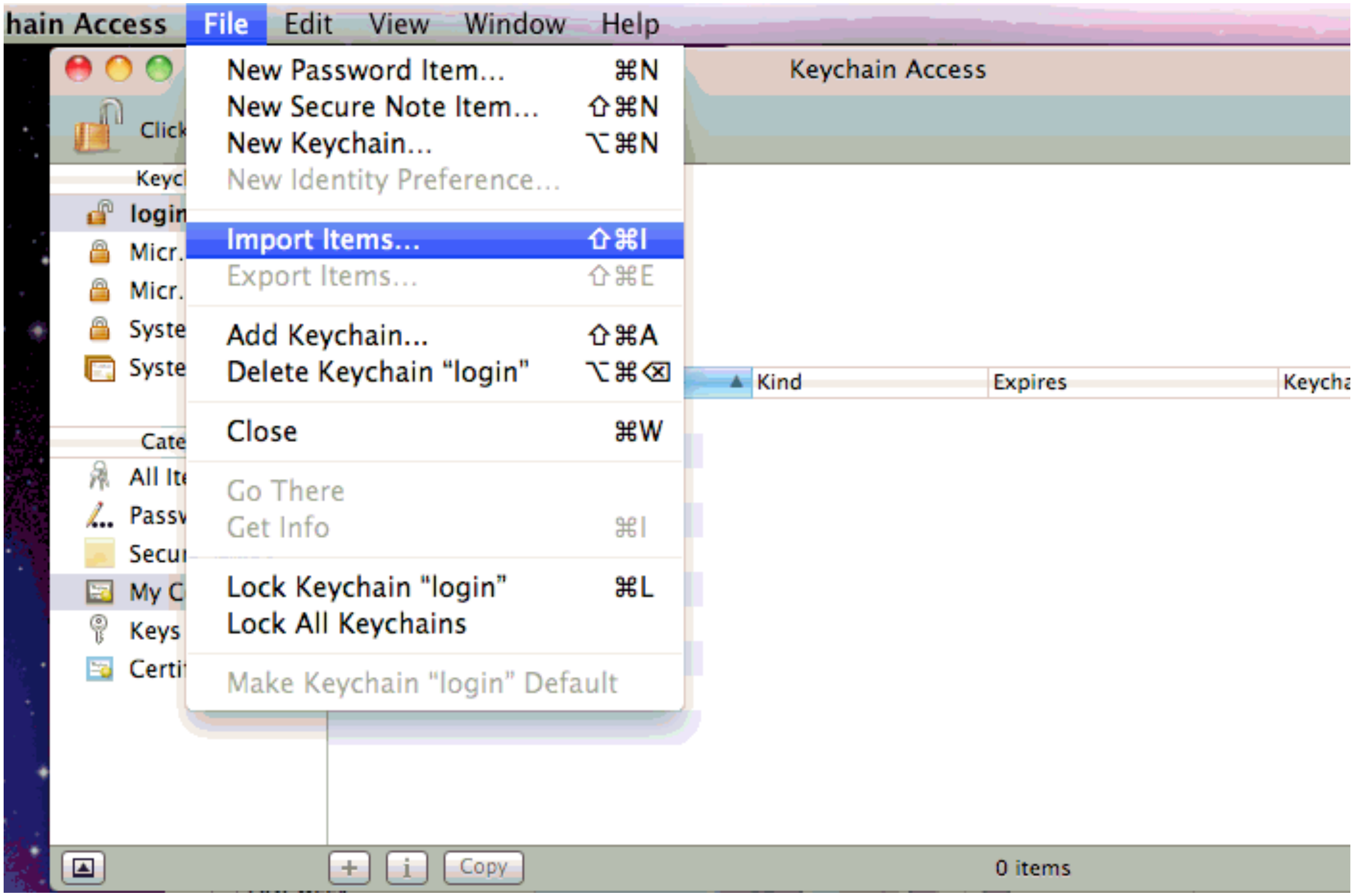


Click to lock the login keychain.

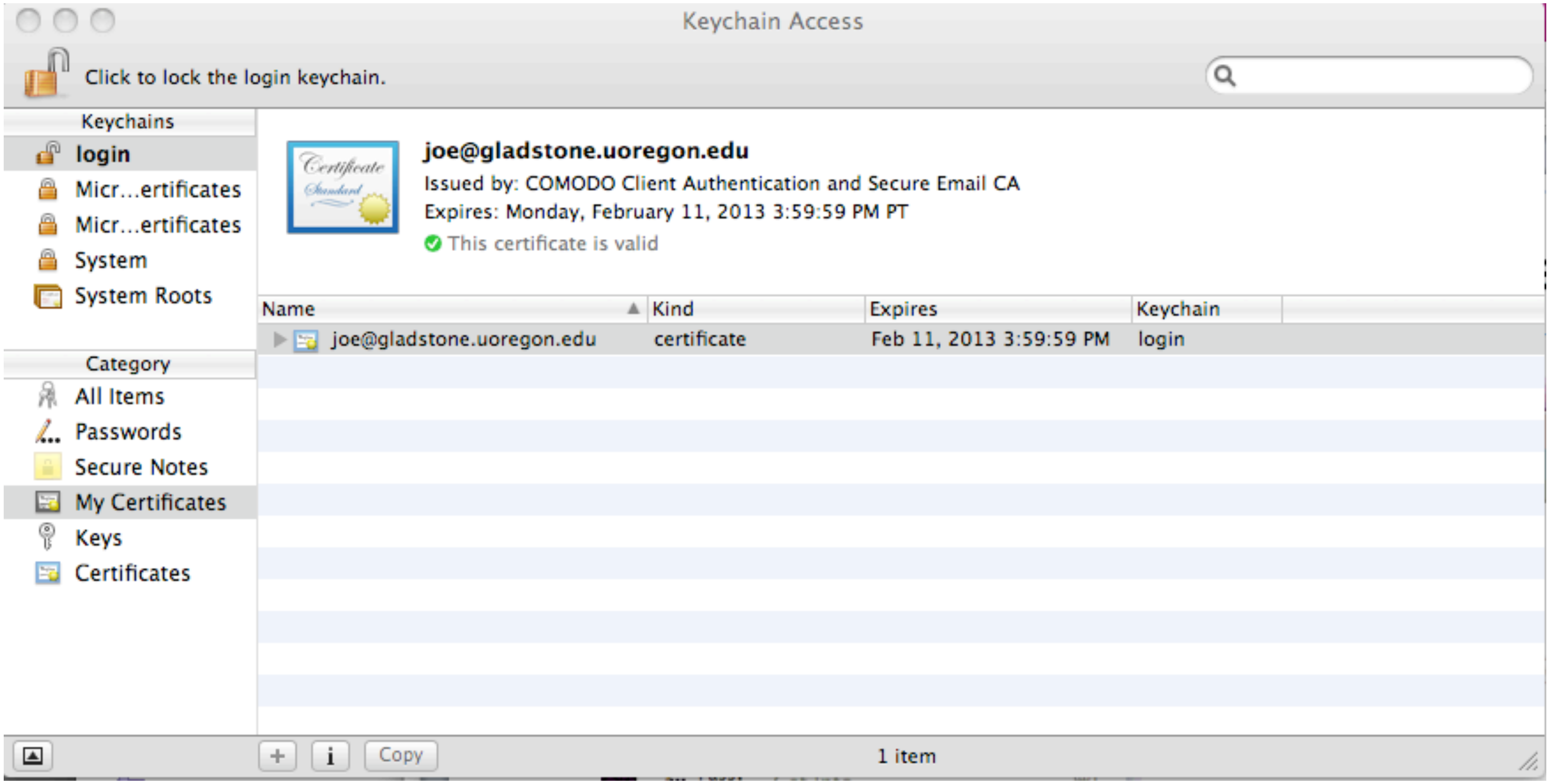
0 items

Can’t find Keychain Access? Check Applications --> Utilities

Importing Our Key/Cert



Success Importing Our Key and Cert

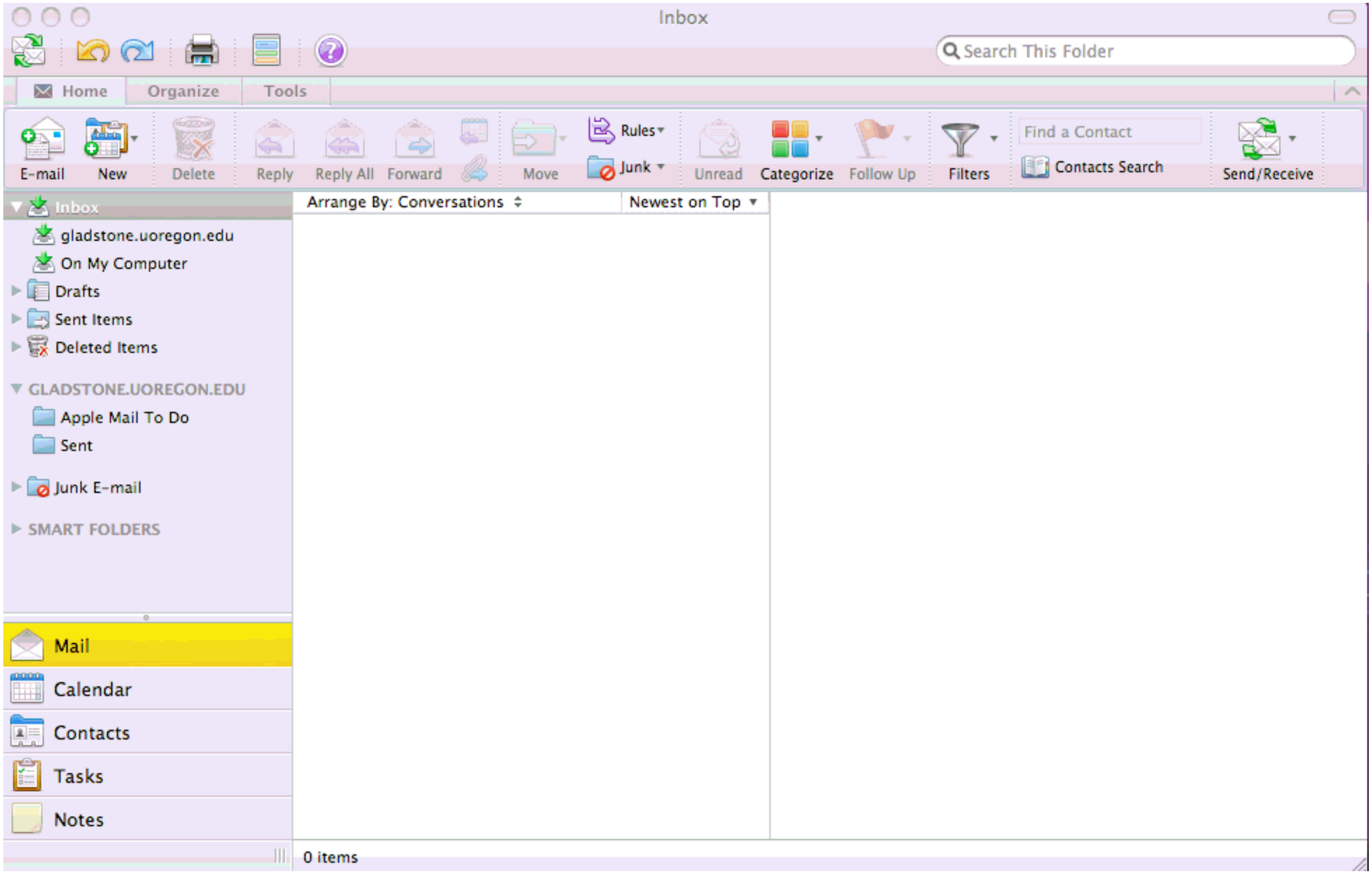


joe@gladstone.uoregon.edu
Issued by: COMODO Client Authentication and Secure Email CA
Expires: Monday, February 11, 2013 3:59:59 PM PT
✓ This certificate is valid

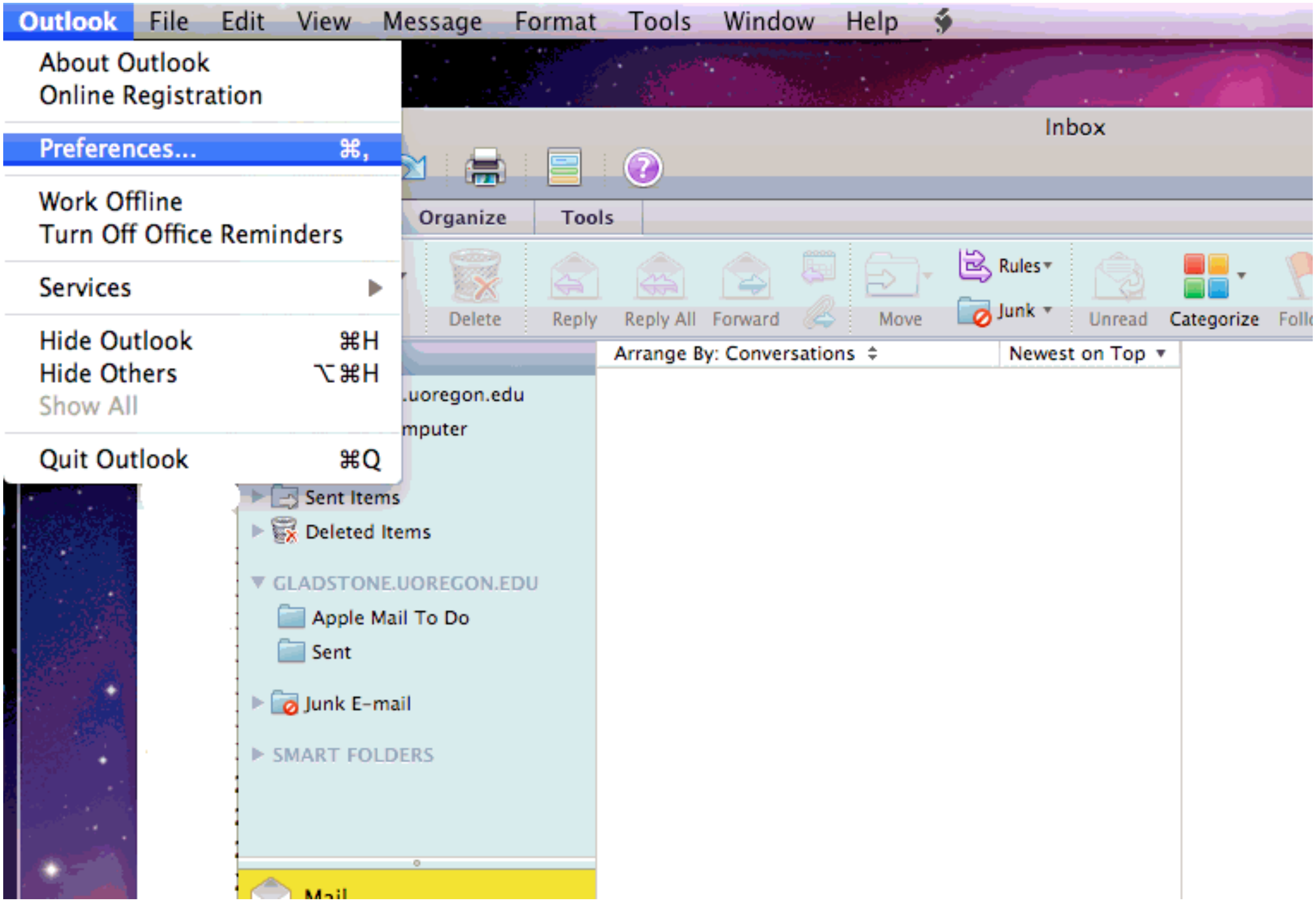
Name	Kind	Expires	Keychain
joe@gladstone.uoregon.edu	certificate	Feb 11, 2013 3:59:59 PM	login

Now we're ready to launch Outlook…

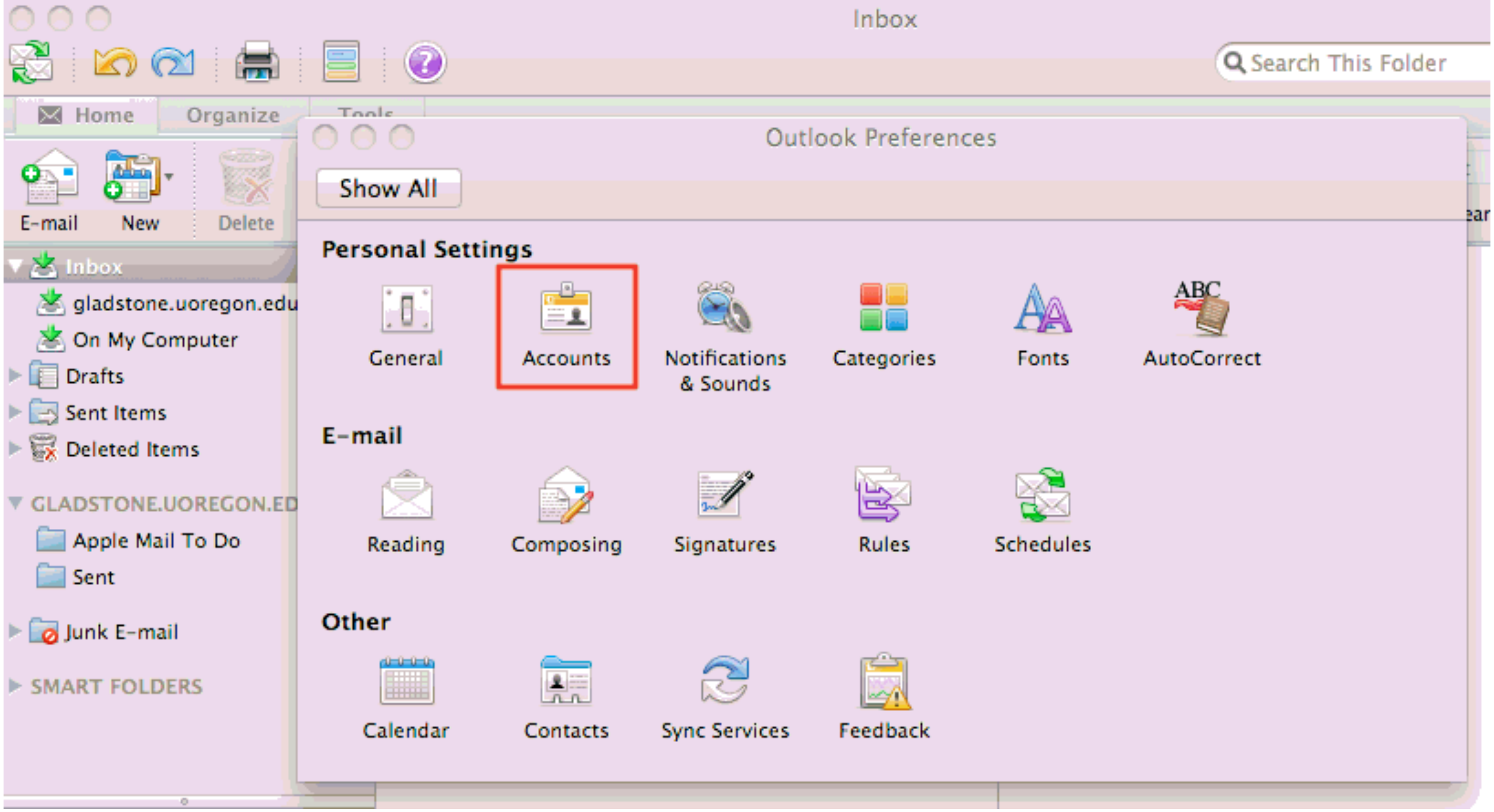
Outlook's Opening Screen…



Outlook --> Preferences…

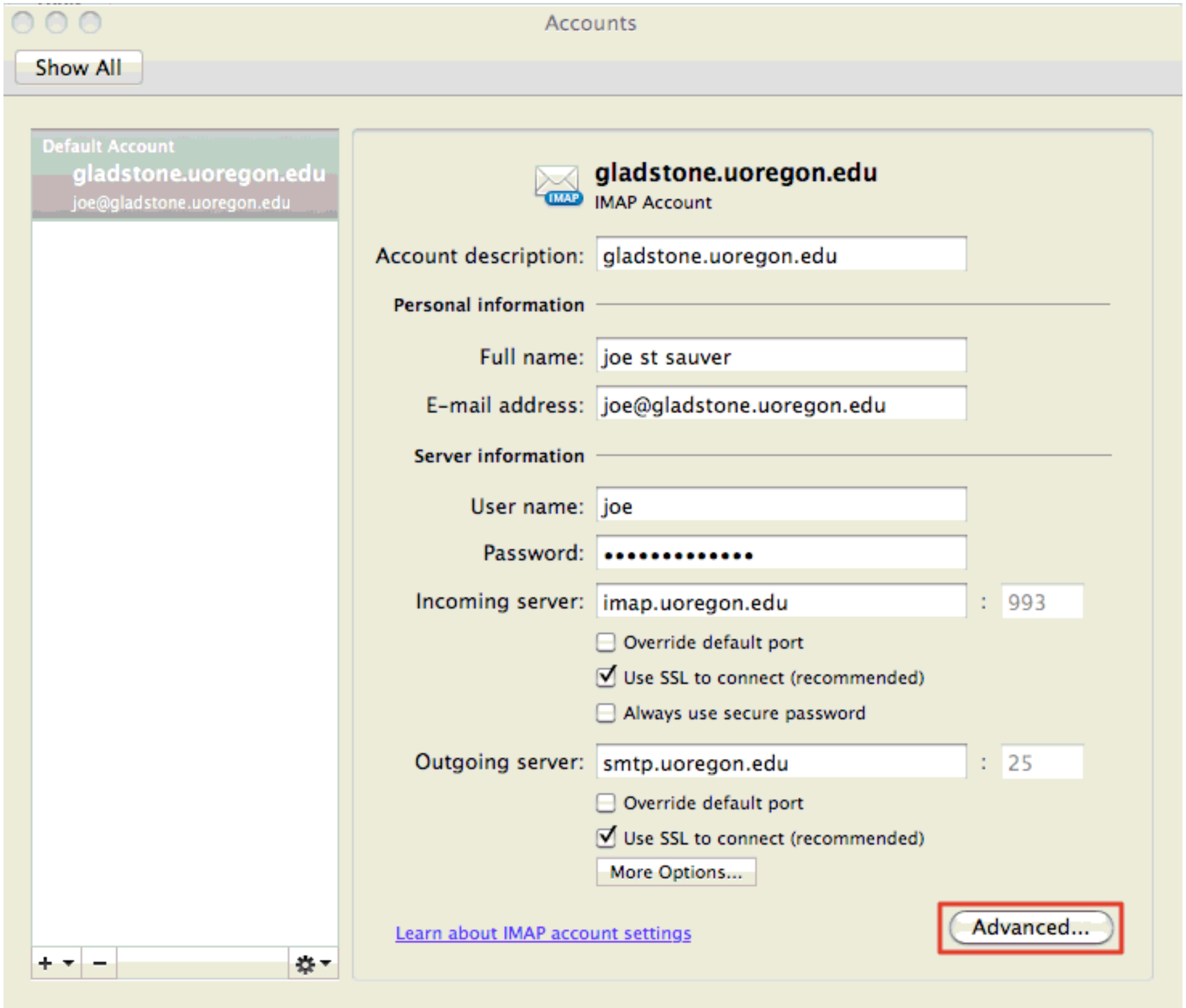


Accounts



Inbox
Search This Folder
Home
Organize
Tools
E-mail
New
Delete
Inbox
gladstone.uoregon.edu
On My Computer
Drafts
Sent Items
Deleted Items
GLADSTONE.UOREGON.ED
Apple Mail To Do
Sent
Junk E-mail
SMART FOLDERS
Outlook Preferences
Show All
Personal Settings
General
Accounts
Notifications & Sounds
Categories
Fonts
AutoCorrect
E-mail
Reading
Composing
Signatures
Rules
Schedules
Other
Calendar
Contacts
Sync Services
Feedback

Advanced Button...



Default Account
gladstone.uoregon.edu
joe@gladstone.uoregon.edu

gladstone.uoregon.edu
IMAP Account

Account description: gladstone.uoregon.edu

Personal information

Full name: joe st sauver

E-mail address: joe@gladstone.uoregon.edu

Server information

User name: joe

Password: •••••••••••••

Incoming server: imap.uoregon.edu : 993

□ Override default port

☑ Use SSL to connect (recommended)

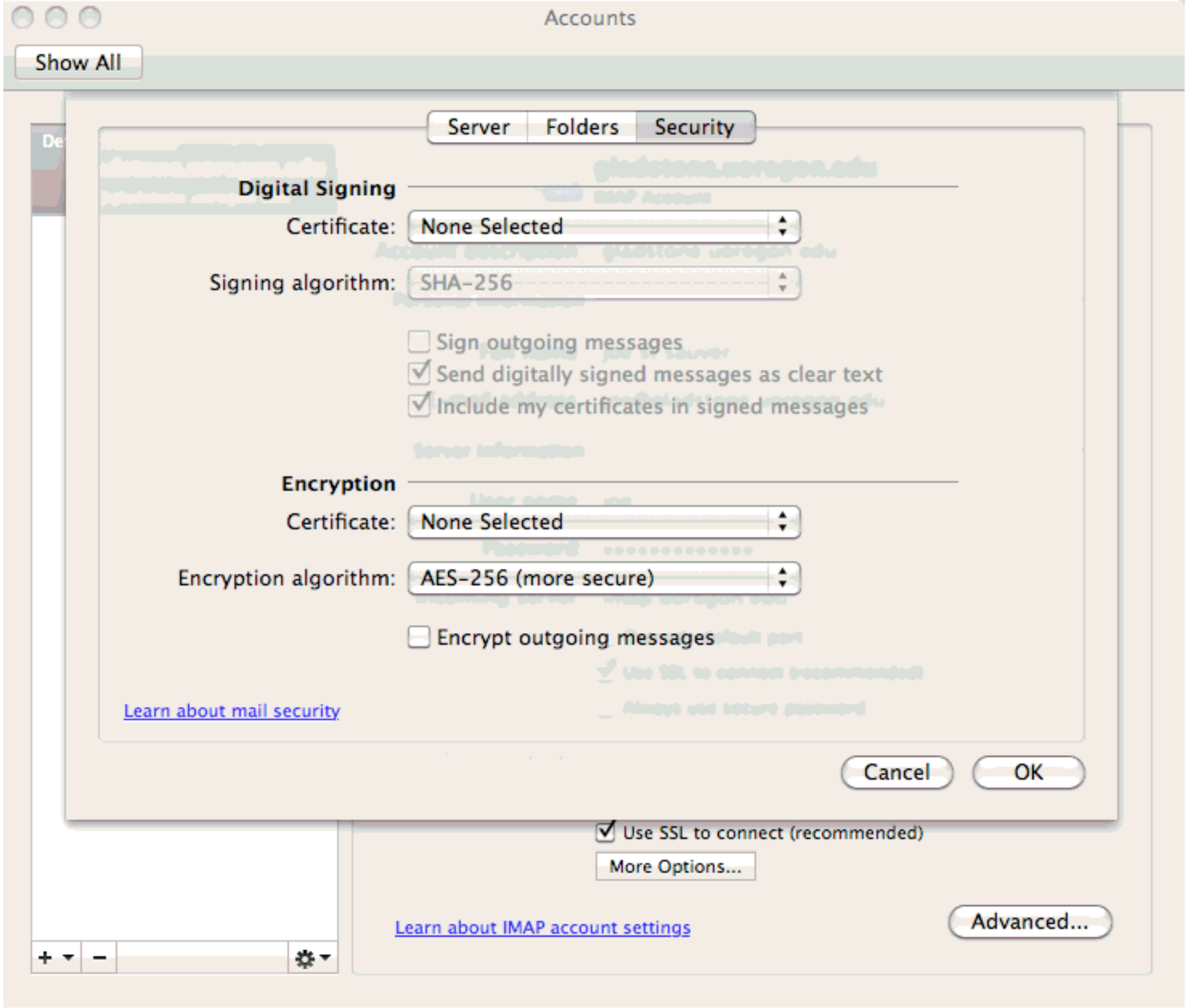
□ Always use secure password

Outgoing server: smtp.uoregon.edu : 25

□ Override default port

☑ Use SSL to connect (recommended)

Picking A Cert on the Account Security Tab



Digital Signing

Certificate: None Selected

Signing algorithm: SHA-256

□ Sign outgoing messages

☑ Send digitally signed messages as clear text

☑ Include my certificates in signed messages

Encryption

Certificate: None Selected

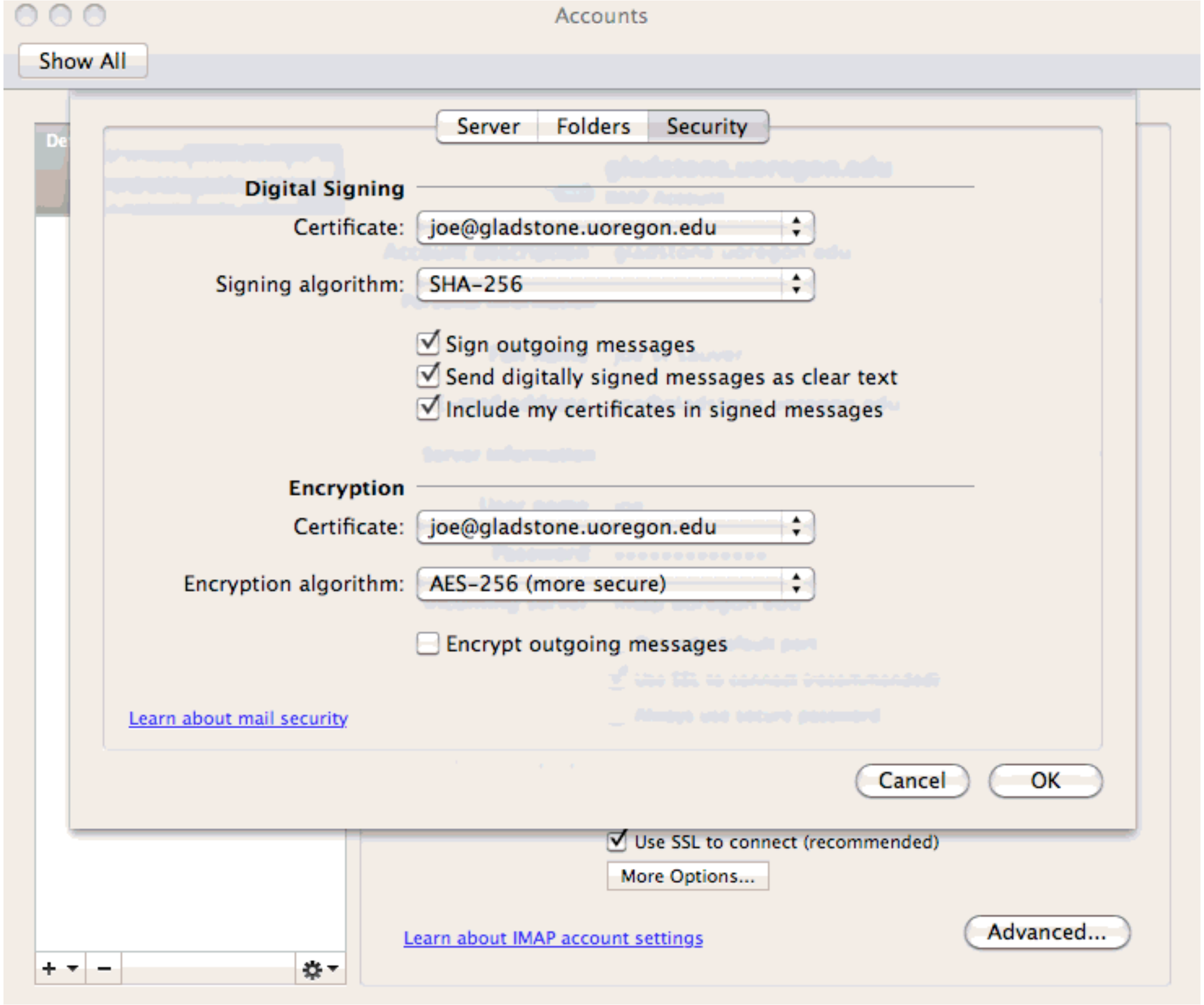
Encryption algorithm: AES-256 (more secure)

□ Encrypt outgoing messages

Learn about mail security

☑ Use SSL to connect (recommended)

Learn about IMAP account settings



Digital Signing

Certificate: joe@gladstone.uoregon.edu

Signing algorithm: SHA-256

☑ Sign outgoing messages
☑ Send digitally signed messages as clear text
☑ Include my certificates in signed messages

Encryption

Certificate: joe@gladstone.uoregon.edu

Encryption algorithm: AES-256 (more secure)

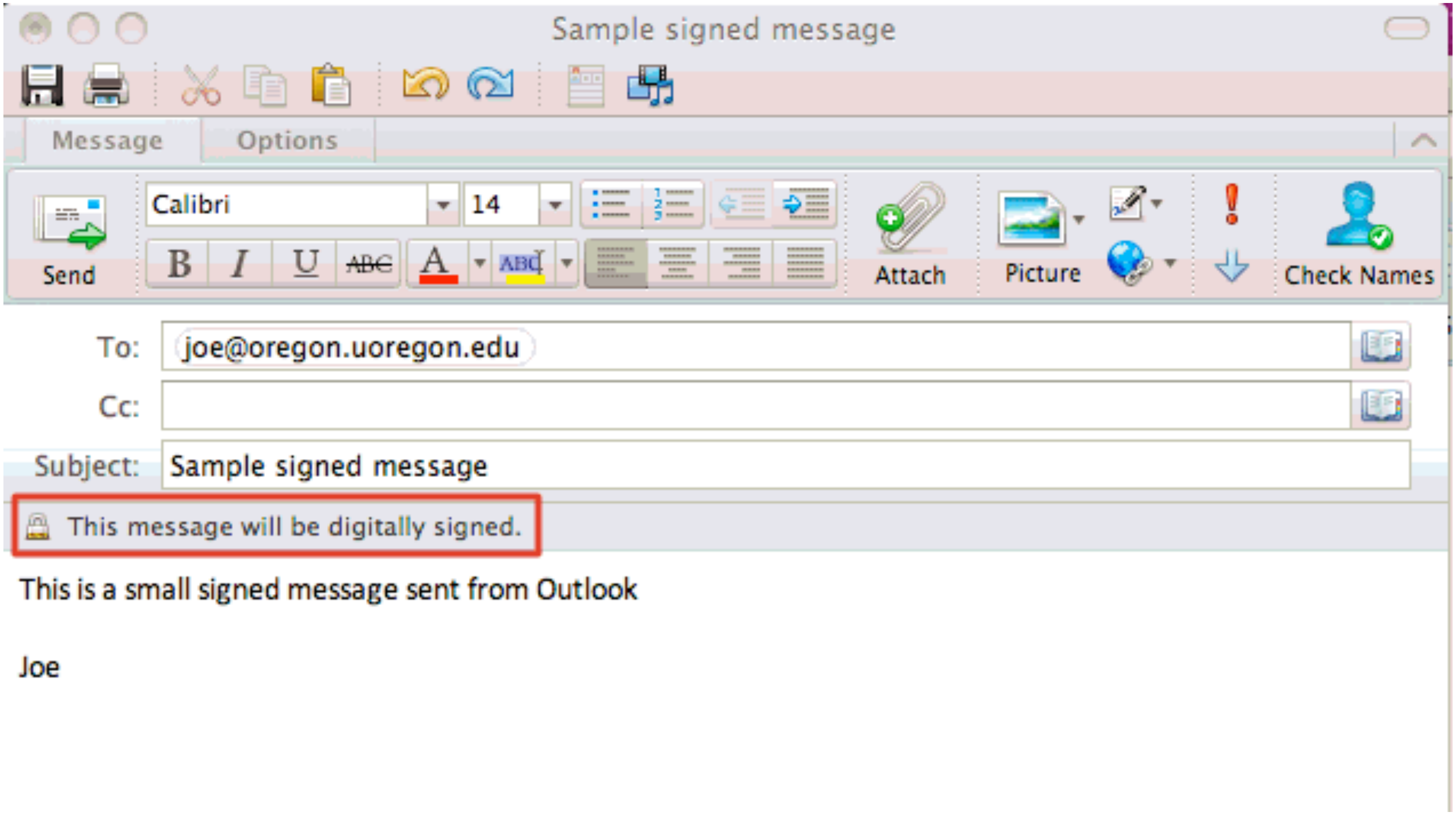
☐ Encrypt outgoing messages

Learn about mail security

☑ Use SSL to connect (recommended)

Learn about IMAP account settings

What The Sender Sees When Sending A Signed Message in Outlook



To: joe@oregon.uoregon.edu

Cc:

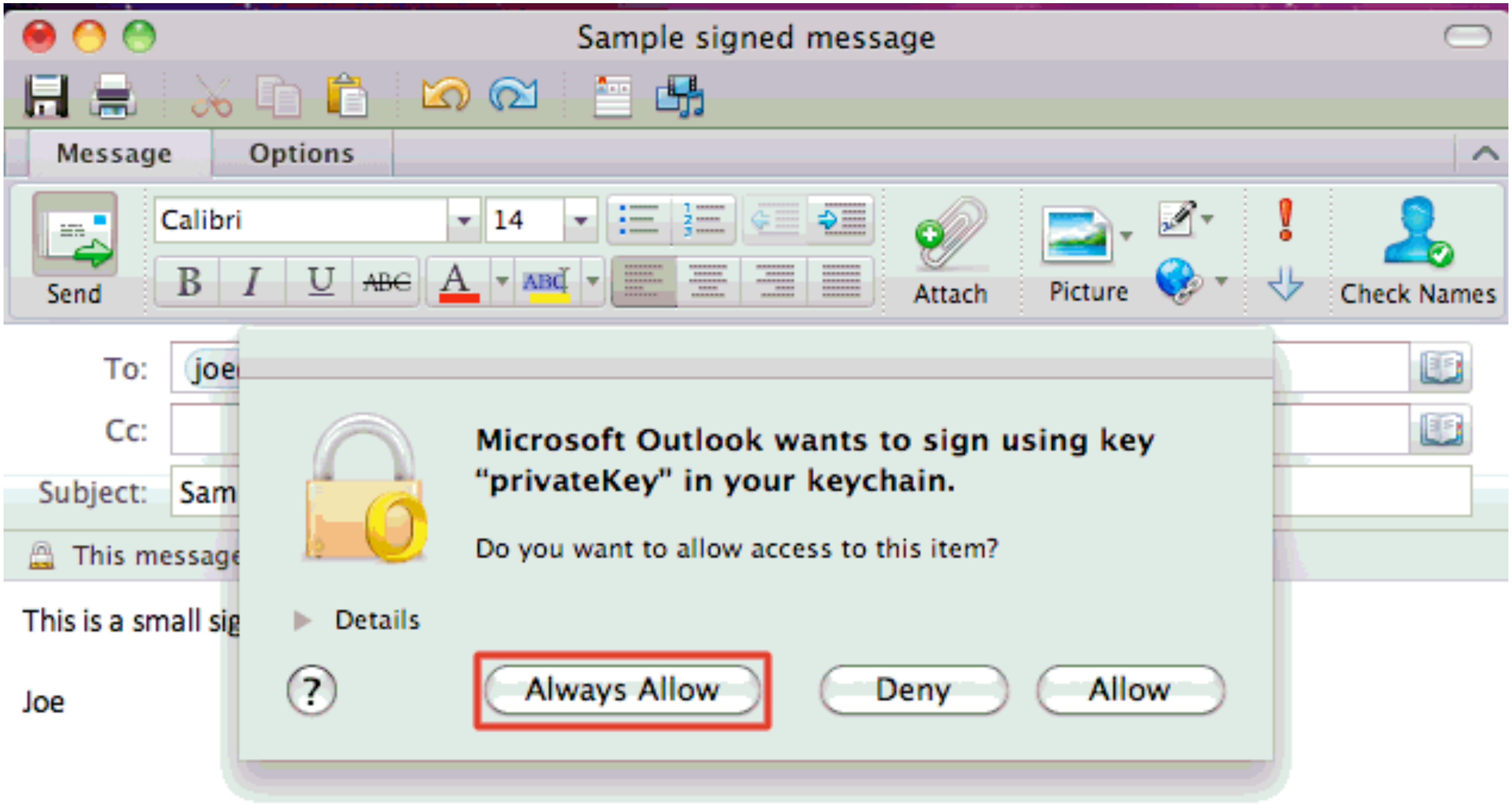
Subject: Sample signed message

This message will be digitally signed.

This is a small signed message sent from Outlook

Joe

Outlook Asks For Confirmation The First Time It Uses Your Private Key/Certificate



To: joe

Cc:

Subject: Sam

This message

This is a small sig

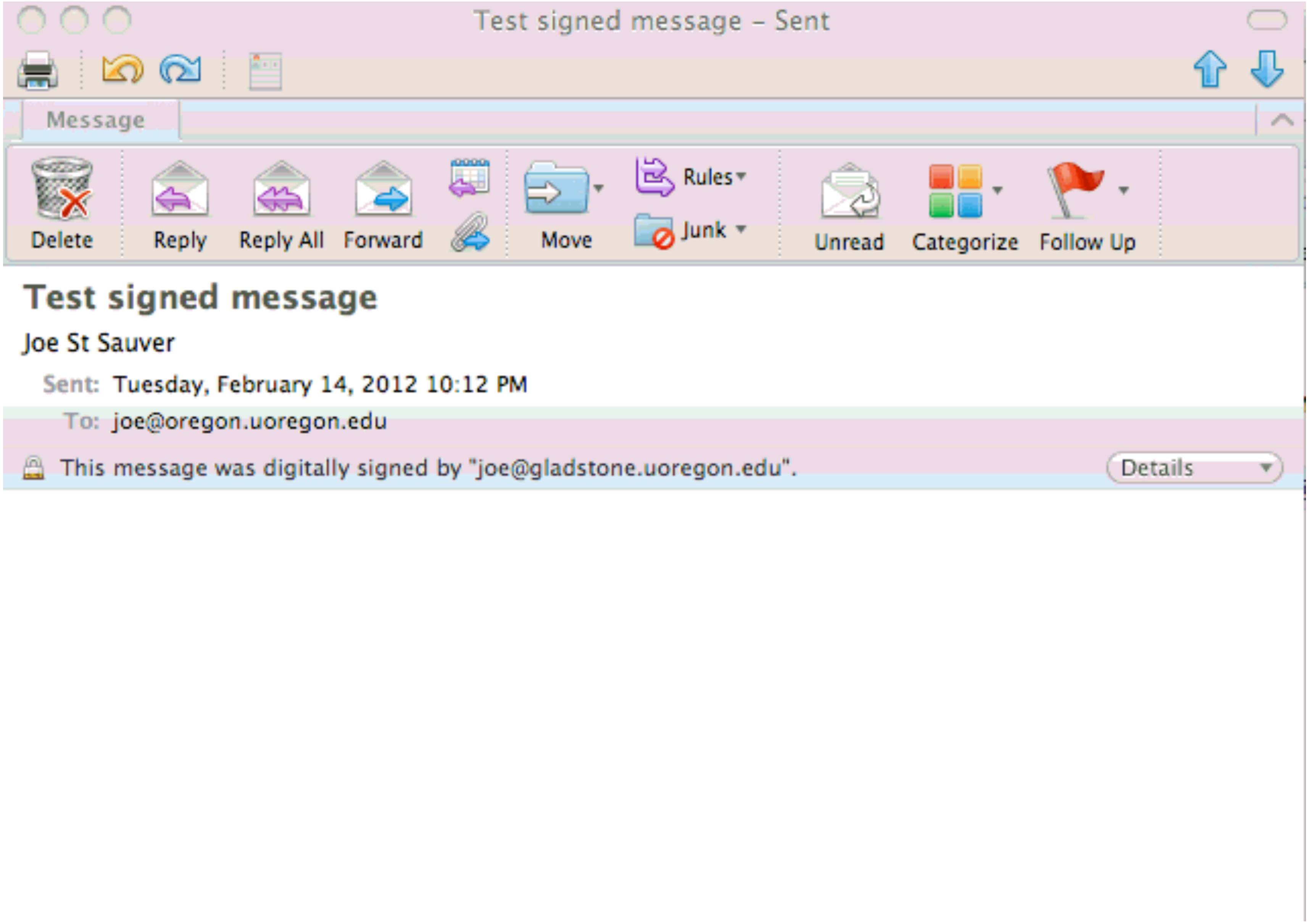
Joe

Microsoft Outlook wants to sign using key "privateKey" in your keychain.

Do you want to allow access to this item?

[**Note:** if you're particularly security conscious, you may just want to click "Allow" rather than "Always Allow"]

What The Recipient Sees In Outlook When Getting A Message That's Signed



Test signed message

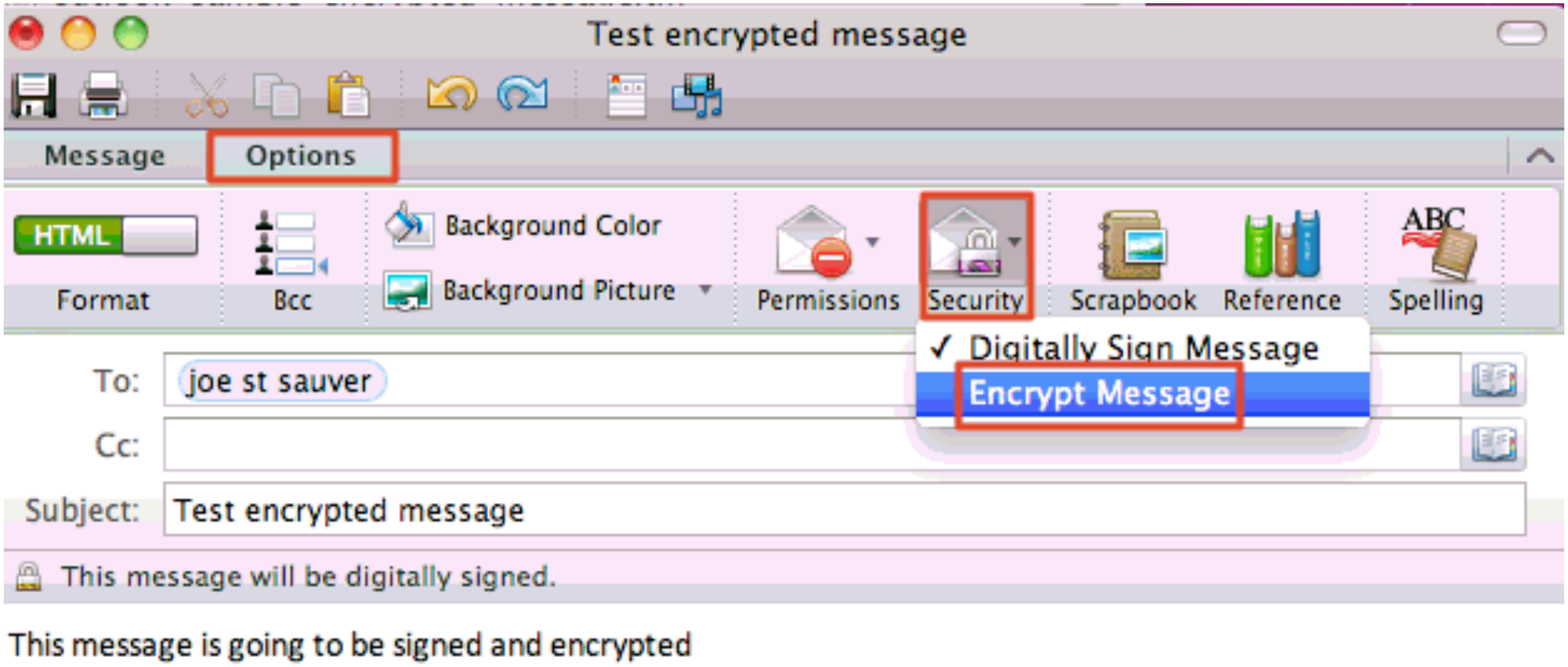
Joe St Sauver

Sent: Tuesday, February 14, 2012 10:12 PM

To: joe@oregon.uoregon.edu

This message was digitally signed by "joe@gladstone.uoregon.edu".

What If We Want To Encrypt A Message?



Test encrypted message

To: joe st sauver

Cc:

Subject: Test encrypted message

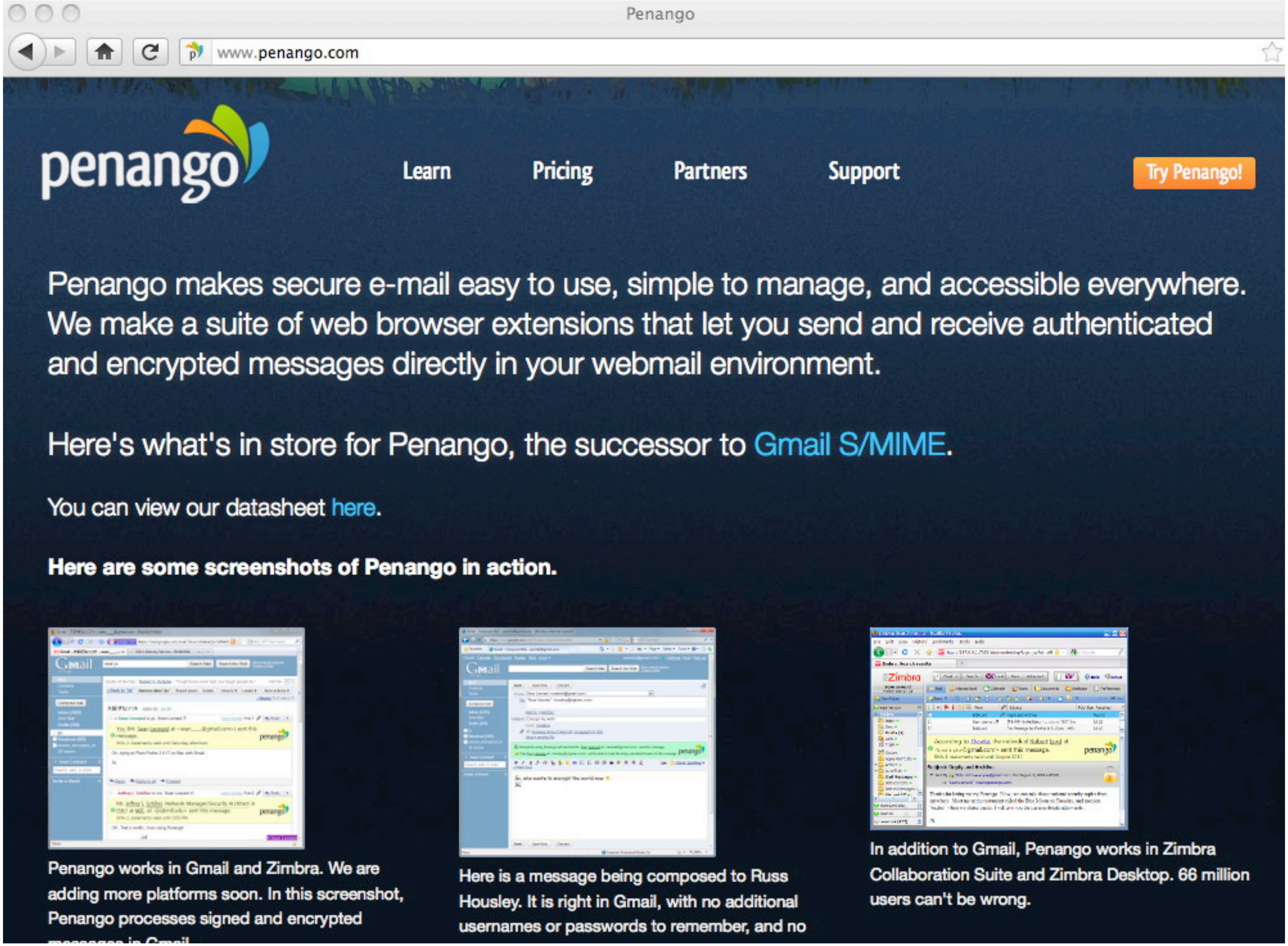
This message will be digitally signed.

This message is going to be signed and encrypted

XII. "What If I Use Gmail Web Email And I Want to Do S/MIME?"

Gmail Does NOT Natively Support S/MIME

- You CAN do S/MIME with a Gmail account if you read your Gmail via a dedicated mail client (such as Thunderbird or Outlook)
- However, if you read your Gmail via Gmail's web email interface, you won't be able to natively S/MIME sign or encrypt your mail traffic. Why? Well, remember that Gmail's business model is based around selling contextual ads (e.g., if you send an email message talking about going on vacation to Honolulu, don't be surprised if you suddenly start to see Gmail ads for airfare to Oahu or discount hotel rooms overlooking Ala Moana).
- Fortunately, you can get a third party browser plugin, Penango, that will help. Penango is free for free Gmail accounts. Thank you Penango! (click on the "Pricing" link to request a download link)
- **Warning: Penango is closely integrated with Firefox, and only supports some versions. Check the version you're using!**



Penango makes secure e-mail easy to use, simple to manage, and accessible everywhere. We make a suite of web browser extensions that let you send and receive authenticated and encrypted messages directly in your webmail environment.

Here's what's in store for Penango, the successor to Gmail S/MIME.

You can view our datasheet here.

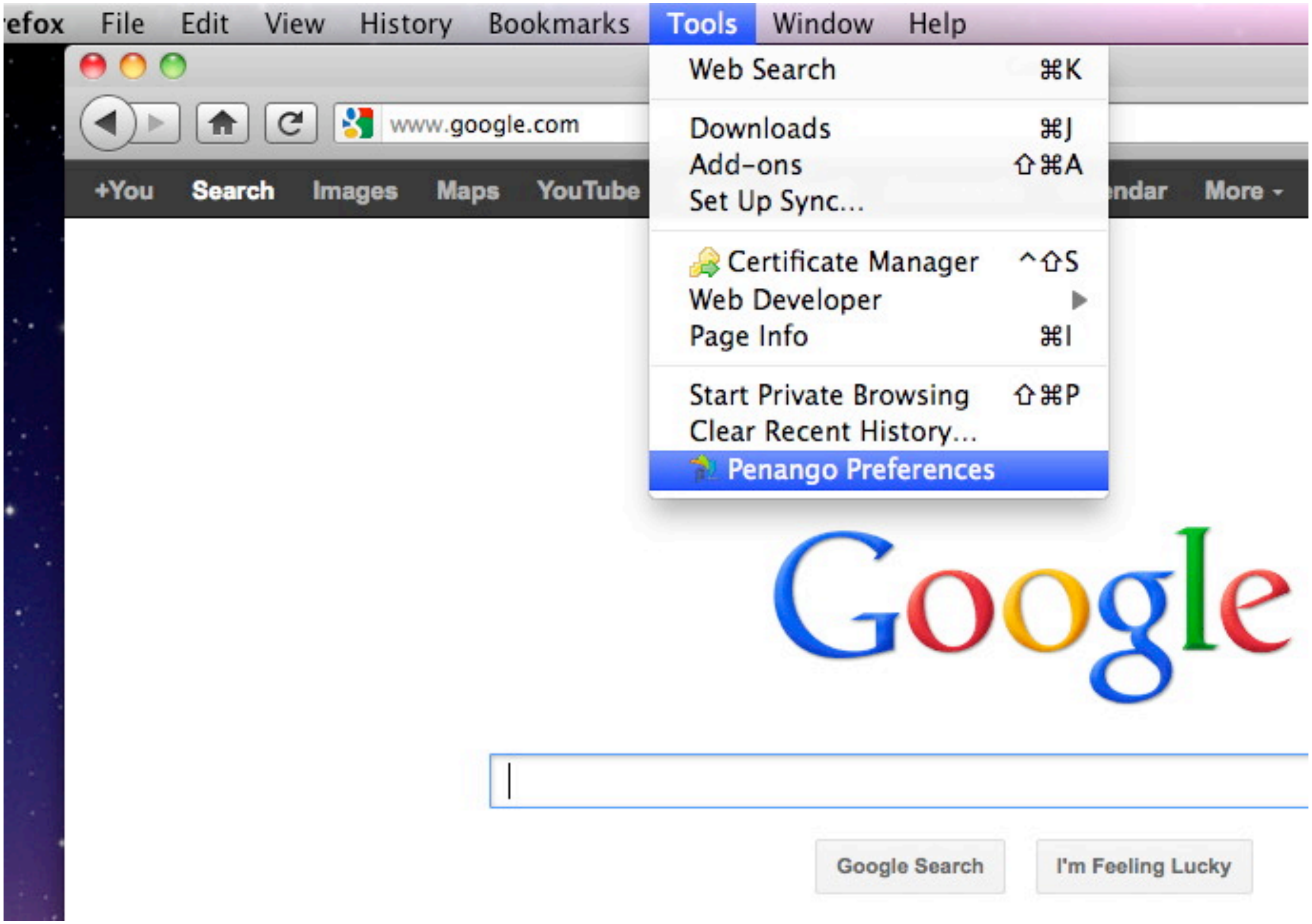
Here are some screenshots of Penango in action.

Penango works in Gmail and Zimbra. We are adding more platforms soon. In this screenshot, Penango processes signed and encrypted messages in Gmail

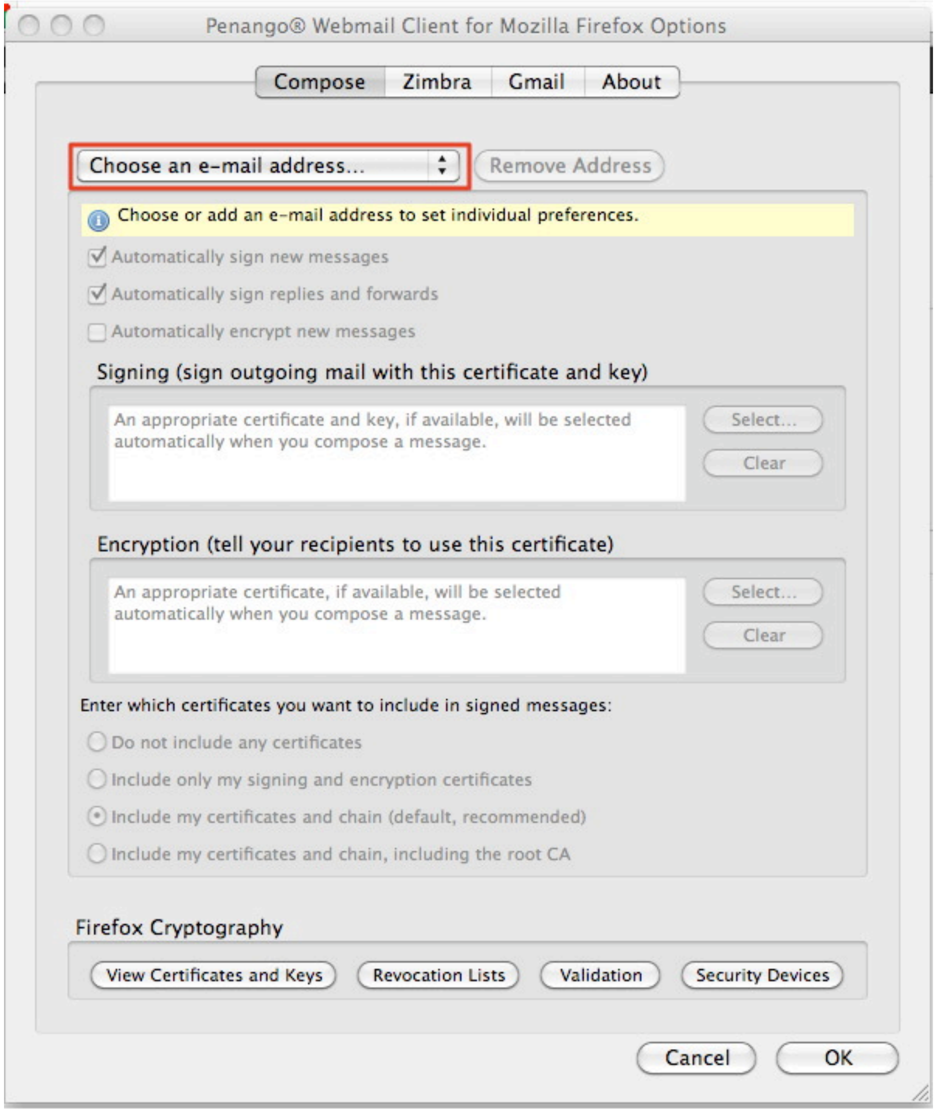
Here is a message being composed to Russ Housley. It is right in Gmail, with no additional usernames or passwords to remember, and no

In addition to Gmail, Penango works in Zimbra Collaboration Suite and Zimbra Desktop. 66 million users can't be wrong.

Once You Have Penango Installed, Open Penango's Preferences in Firefox



Plug In Your Gmail Address



Penango® Webmail Client for Mozilla Firefox Options

Choose an e-mail address...

Choose or add an e-mail address to set individual preferences.

☑ Automatically sign new messages

☑ Automatically sign replies and forwards

□ Automatically encrypt new messages

Signing (sign outgoing mail with this certificate and key)

An appropriate certificate and key, if available, will be selected automatically when you compose a message.

Encryption (tell your recipients to use this certificate)

An appropriate certificate, if available, will be selected automatically when you compose a message.

Enter which certificates you want to include in signed messages:

○ Do not include any certificates

○ Include only my signing and encryption certificates

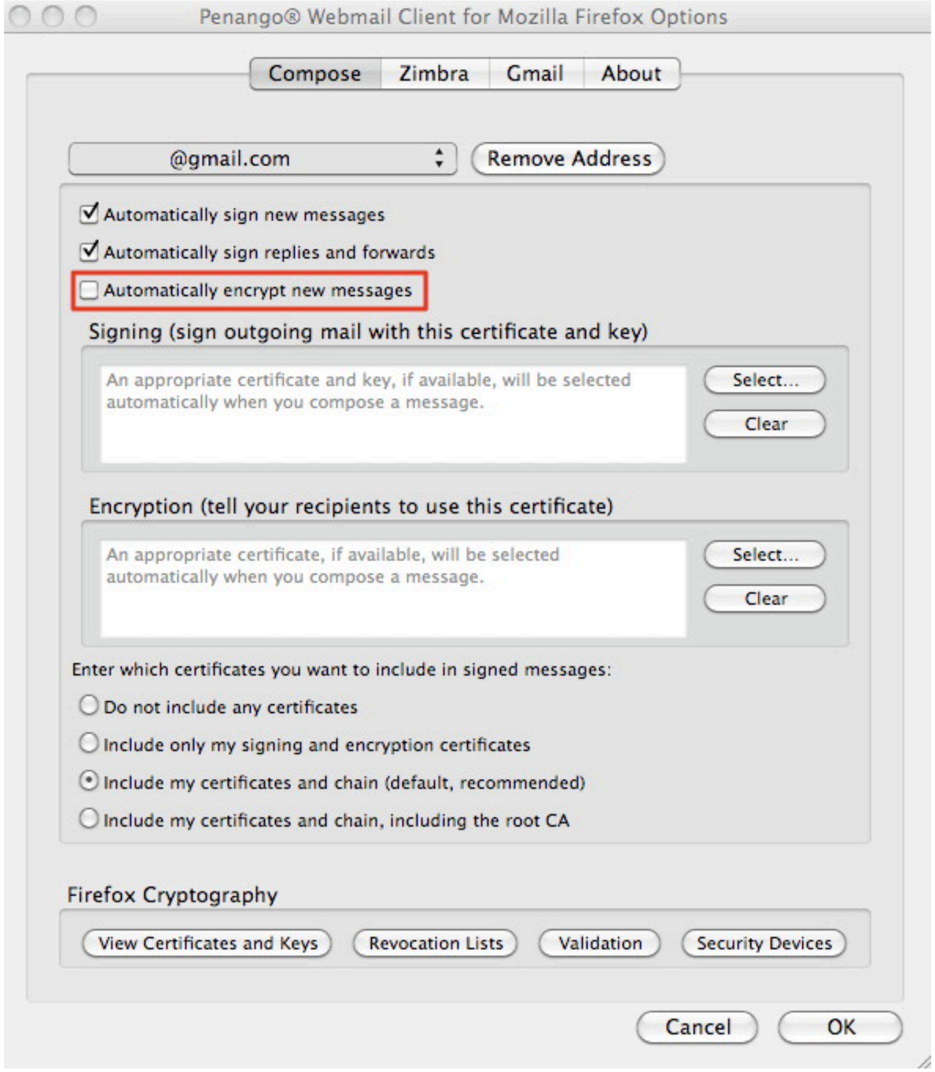
● Include my certificates and chain (default, recommended)

○ Include my certificates and chain, including the root CA

Firefox Cryptography

[some account details elided above]

Uncheck "Automatically encrypt new messages"



@gmail.com

☑ Automatically sign new messages

☑ Automatically sign replies and forwards

□ Automatically encrypt new messages

Signing (sign outgoing mail with this certificate and key)

An appropriate certificate and key, if available, will be selected automatically when you compose a message.

Encryption (tell your recipients to use this certificate)

An appropriate certificate, if available, will be selected automatically when you compose a message.

Enter which certificates you want to include in signed messages:

○ Do not include any certificates

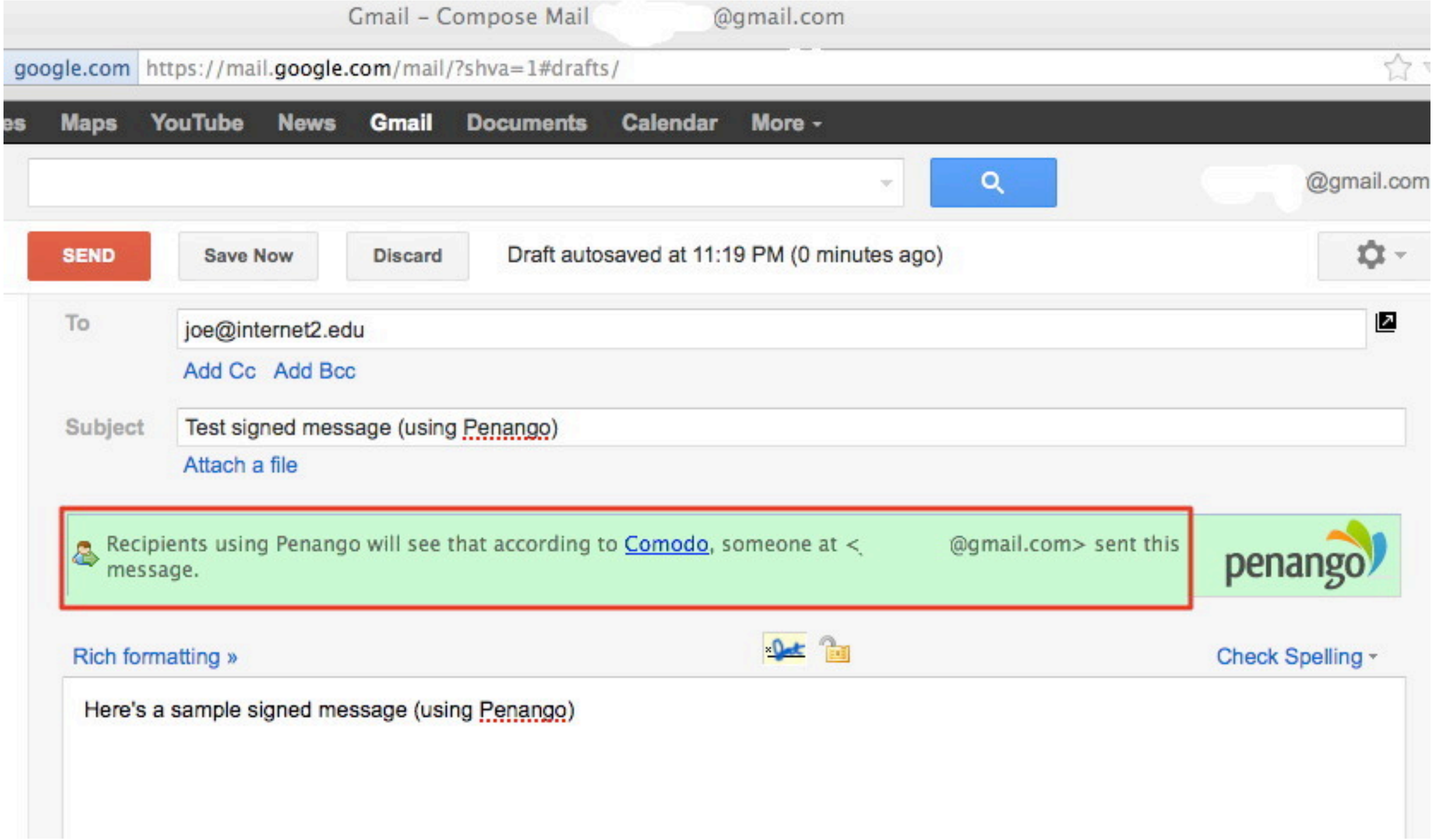
○ Include only my signing and encryption certificates

● Include my certificates and chain (default, recommended)

○ Include my certificates and chain, including the root CA

[some account details elided above]

Composing a Signed Gmail Msg With Penango


Gmail – Compose Mail @gmail.com
google.com https://mail.google.com/mail/?shva=1#drafts/
es Maps YouTube News Gmail Documents Calendar More
@gmail.com
SEND
Save Now
Discard
Add Cc Add Bcc
Attach a file
penango
Rich formatting »
Check Spelling

Draft autosaved at 11:19 PM (0 minutes ago)

To: joe@internet2.edu

Subject: Test signed message (using Penango)

Recipients using Penango will see that according to Comodo, someone at < @gmail.com> sent this message.

Here's a sample signed message (using Penango)

[some account details elided above]

Some Penango-Related Sending Idiosyncrasies

- When you send a signed or encrypted message using Penango, the message gets submitted “outside” of Gmail's web interface (e.g., via SMTPS to smtp.gmail.com). It does NOT get sent within the Gmail web interface. This is necessary because Penango needs to set the top-level message Content-Type appropriately for S/MIME.
- They submit via port 465 (grr!) and not STARTTLS on port 587; if proxies are in use, Penango will endeavor to use them, too.
- The IP of the handoff host does appear in the Gmail headers.
- The body of the message may be base64 encoded even if the message you're signing is plain-text-only. Penango also uses a long/ugly name for the .p7s attachment
- Speaking of, some message text/message formatting may make it appear as if you must use Penango to process a Penango-generated S/MIME message. That's an incorrect impression.

XIII. Hard Tokens/Smart Cards

Alternatives To Storing Your Keys and Certs On Your Desktop or Laptop

- In higher education, many users don't have a clean one-to-one mapping of users to systems.
- For example, a security conscious user might have both a desktop and a laptop, and might want to use their certificates on both those systems, but might not want to leave their credentials stored on multiple systems if they don't have to.
- A less well-off user might not have a system of their own, working from shared systems in a campus computer lab, instead. Obviously it would be bad for that user to download and install their credentials on a shared system in that lab if that system will soon be used by someone else, or if they may be assigned to use some other system the next time they visit the lab.
- What we really need is a way for users to save and carry their S/MIME certs with them wherever they go.

Hard Tokens/Smart Cards Advantages

- Users can use one set of PKI credentials everywhere.
- Users can carry their credentials with them wherever they go (it's just another blob on your keychain, or another "credit card" in your wallet or purse)
- The user's private/public keypair can potentially* be generated on-token (or on-smart card), with the private key never leaving the device
- The user can insert and unlock their token or smart card only when they need it, keeping that credential offline (and sheltered from online attack) the rest of the time
- **Client cert issuance can mimic other well established credential <u>issuance</u> processes (such as those for ID cards or door keys); ditto for client cert <u>use</u> processes.**

* Not currently possible for InCommon client certificates.

Getting **An Institutional ID (or Door Key)**

Getting a university ID card or a door key usually involves:

-- Obtaining proof of authorization, such as a letter of admission or a signed contract (or a completed key auth form)

-- Taking your paperwork and a drivers license or passport, and visiting the campuscard office (or a distributed credential distribution site, perhaps located in the student housing office or personnel department)

-- Paperwork and current proof of identity get reviewed and OK'd

-- One's photo gets taken (for the ID card) or a deposit gets collected for a key, and it gets issued while-you-wait.

This works. Not painless, but not horrible, and it's relatively secure.Now visualize the ID card as actually a smart card (with a client cert on it), or the "key" actually being a USB format PKI hard token... would that process need to be materially different than the current process of issuing ID cards or door keys? No...

Using An Institutional ID (or Door Key)

Everyone knows how to use their ID card (or keys):

-- Carry it with you, so you have it with you when you need it

-- When needed, allow your card to be scanned or inspected (or stick your key in the lock and turn it to open the door); this is simple, so training is not required.

-- If you lose your ID or your key(s), you report it so you can get a replacement, and so your old one can be marked as invalid (or so any locks associated with the lost key can be potentially changed)

-- If your key doesn't get you into a space you need to access, you'll be given another one (repeat the "getting a key" process).

-- Your ID card or keys get collected if you leave or are kicked out.

Using client certs needs to be as easy as using an ID card or door key, and can be if hard tokens/smart cards are used.

USB-Format PKI Hard Tokens

- USB-format PKI hard tokens look a lot like a regular USB thumb drive, but a USB-format PKI hard token is actually a completely different animal that just coincidentally *looks* like a thumb drive.
- Specifically, a USB-format PKI hard token is actually a highly specialized secure cryptographic processor with integrated secure storage. Correctly configured, it allows you to save and USE your S/MIME keys and certificate, but without putting those credentials at risk of being "harvested"/stolen. These days, with all the credential harvesting malware that's out there, that's a pretty cool thing.
- In fact, USB-format PKI hard tokens have the ability to potentially generate private/public keypairs *on the token itself*, so that the private key NEVER leaves the token, although we will not be taking advantage of that capability during today's session (and in fact that's also not supported for InCommon Client Certificates)

Safenet eToken PRO 72K

- Through the generosity of Chen Arbel at Safenet, we're able to provide each Security Professionals client cert training participant with a free USB format PKI hard token today, the Safenet eToken PRO 72K, as well as the driver software and documentation. Thank you, Chen and Safenet!

- This token, formerly marketed by Aladdin, is the most popular USB format PKI hard token used in higher education, and is particularly nice if you work in a cross platform environment since it is supported under Microsoft Windows, Mac OS X, and Linux.

Image credit: http://commons.wikimedia.org/wiki/File:EToken_PRO_USB.jpg

"Thanks for *One*, But I Need A *Bunch of* Them!"

- USB-format PKI hard tokens are available from many major IT channels. For example, CDW-G currently offers the Safenet e-Token Pro for $38.89/each (qty 1-100), and the SAC (required software drivers) costs $18.94. If you throw on one of the little protective shells (like the one we provided for you today), that's another couple bucks from CDW-G, bringing the price right up to around $60.00/unit. Naturally, while ~$60/unit isn't a big deal for a small number of users, it adds up pretty quickly if you want to issue hard tokens to a whole campus, particularly if there are competing two factor auth solutions that may be ~$5/user.
- Fortunately, InCommon has arranged to be able to sell deeply discounted SafeNet PKI hard tokens to InCommon higher education subscribers. For more information, see
http://www.incommon.org/safenet/index.html
(note: a minimum order of two hundred units applies)

"But I Only Want To Order A Dozen Tokens!"

- If you're only buying a small number of tokens for a test deployment, you can already get those on the open market. Internet2/InCommon doesn't need to get involved in order for that to be practical. Our goal is explicitly ***not*** to make small-scale test PKI deployments cheap(er).
- On the other hand, if the community is trying to deploy thousands, tens of thousands, hundreds of thousands, or even millions of client certificates, THAT's the sort of process we want to facilitate, and where central coordination may be critical.
- Put another way, Internet2/InCommon is, and *should be*, all about facilitating "deployment at scale."
- This is an important principle that Randy Frank deserves special acknowledgement for correctly emphasizing.

Safenet Drivers, Local Token Management Software, And Documentation

- Most systems will require the installation of token drivers and/or local token management software (so you can load your existing certificate onto the token). With Safenet's permission we are making that software and documentation for this product, available to you for installation via CD-ROM. **We ask that you respect this copyrighted software: please do NOT redistribute it!**
- You should see three files:
  -- SAC 8_1 SP1.zip (Windows) 206.9 MB
  MD5sum=55876842e6e13e6c8ee6cdf9dd16986a
  -- 610-011815-002_SAC_Linux_v8.1.zip 42.2 MB
  MD5sum=d66c9ff919f3b35180dba137857eb88c
  -- 610-001816-002_SAC8.1Mac.zip 18.2 MB
  MD5sum=c2e9e9b0e2706ffab310538574cf009b

Installing the SAC On the Mac

- Insert the CD-ROM and drag the 610-011816-002_SAC8.1Mac.zip file to your desktop. Unzip it with the Archive Utility, Stuffit, or whatever application you normally use to unzip files. You should end up with a folder called "SAC 8.1.0.5" with two subfolders: "Documentation" and "Mac Installer."
- **READ THE DOCUMENTATION IN THE DOCUMENTATION FOLDER! In particular, <u>read the Administrator's Guide</u> and <u>read the ReadMe</u> file, particularly "Known Issues/Limitations"**
- **Really, I kid you not, read the dang documentation, please!**
- Then go to the Mac Installer folder, and run the installer that's in there: SafeNetAuthenticationClient.8.1.0.5.dmg
- When you mount that dmg file, you will see
  Install SafeNet Authentication Client 8.1.mpkg
- Install it. **You'll need to reboot when it finishes**

Firefox Security Module

- As mentioned in the document (which you ARE going to read, right?) when you install the Safenet Authentication Client, it doesn't automatically install the securitymodule in Firefox. You need to do that manually.

- Firefox --> Preferences... --> Advanced
  In the Encryption tab, click on Security Devices
  In the Device Manager window, click Load
  In the Load PKCS#11 Device window, Module filename, enter:
  /usr/local/lib/libeTPkcs11.dylib
  In the Confirm window, click OK

- Repeat this process for Thunderbird, too.

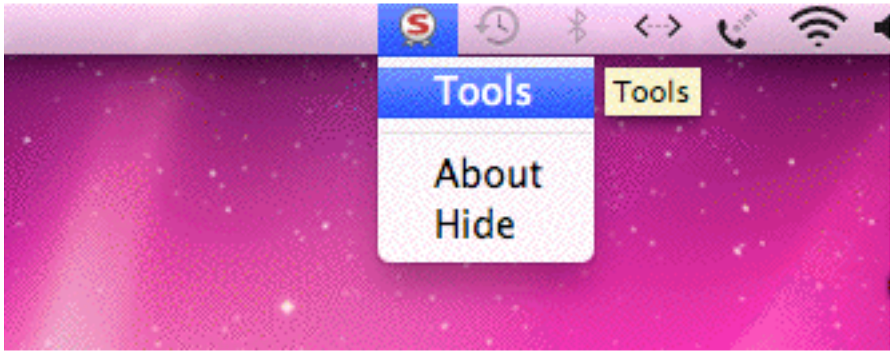
"But I'm Using Windows, Not A Mac!"

- Windows users should see Appendix I at the end of these slides.

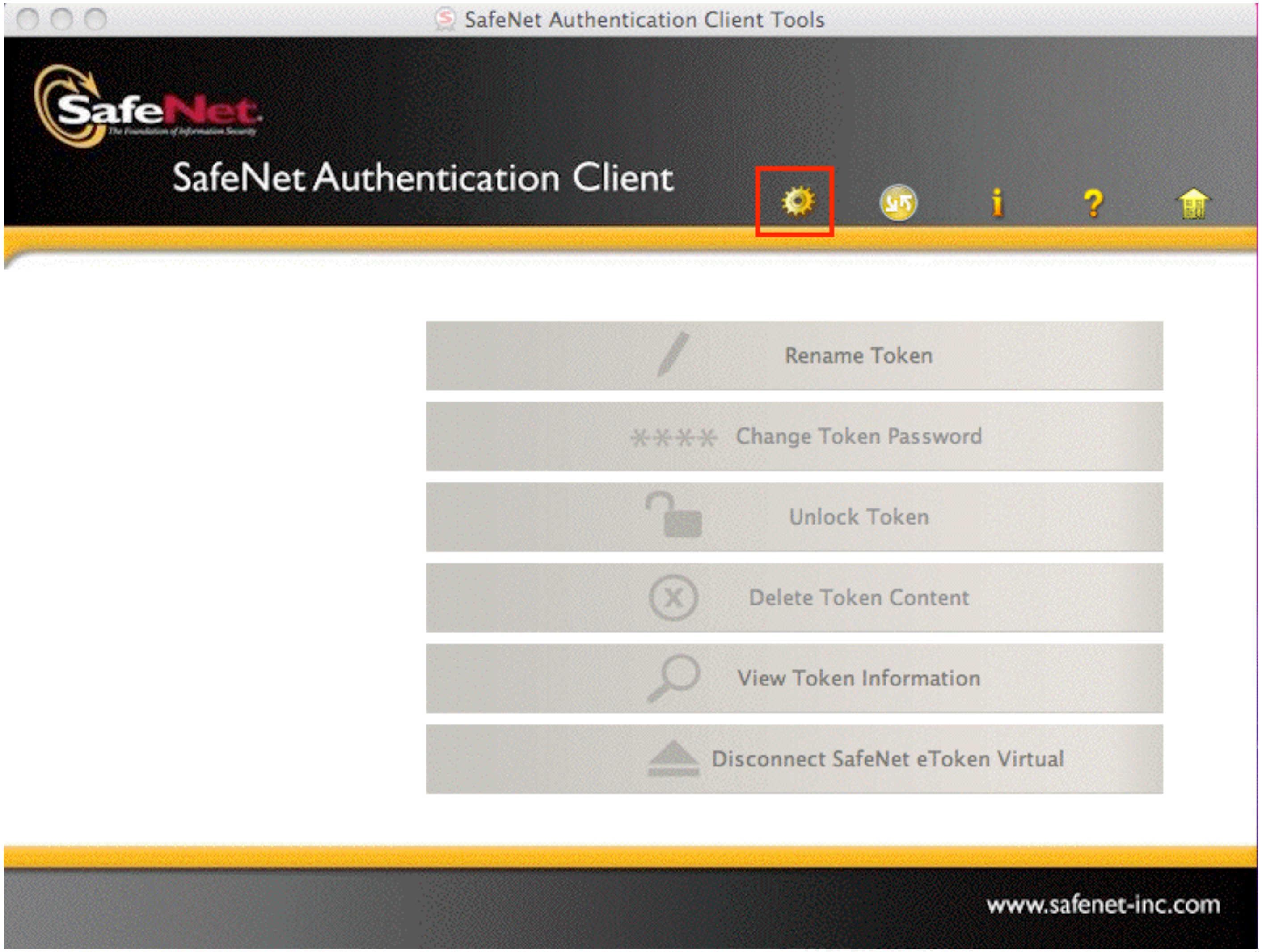
  It has instructions for setting up your SafeNet hard token with a Windows 7 box.

- We'd have bundled them in here, in line, but we didn't want to interrupt things/confuse the Mac users.

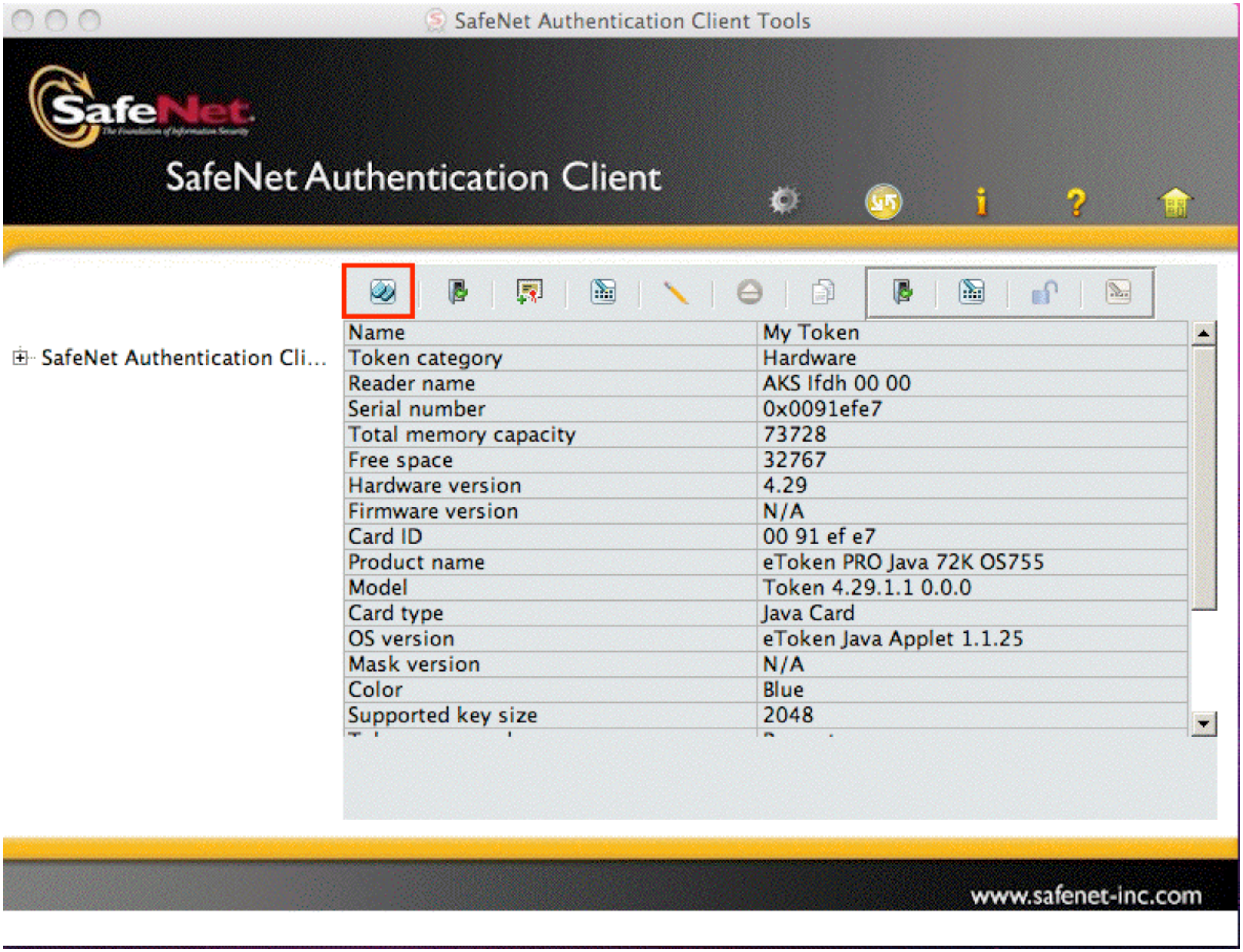
Now Launch the SafeNet Authentication Tools



Go To The Gear Menu ("Advanced")

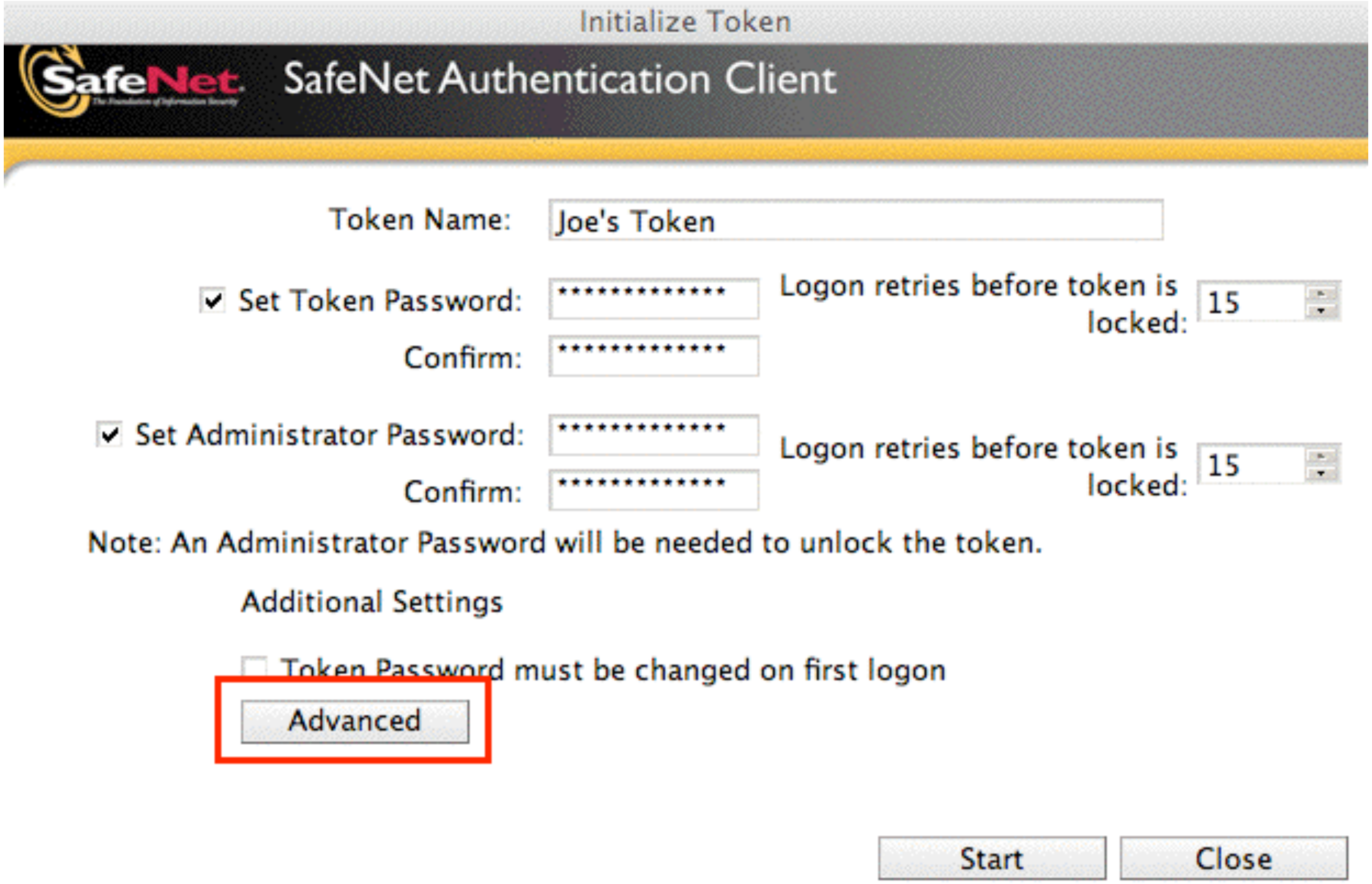


Select "View Token Information," Then Initialize It


SafeNet Authentication Client Tools
SafeNet
The Foundation of Information Security
SafeNet Authentication Client
SafeNet Authentication Cli...
www.safenet-inc.com

Name	My Token
Token category	Hardware
Reader name	AKS Ifdh 00 00
Serial number	0x0091efe7
Total memory capacity	73728
Free space	32767
Hardware version	4.29
Firmware version	N/A
Card ID	00 91 ef e7
Product name	eToken PRO Java 72K OS755
Model	Token 4.29.1.1 0.0.0
Card type	Java Card
OS version	eToken Java Applet 1.1.25
Mask version	N/A
Color	Blue
Supported key size	2048

Enter Your New Passwords and Then Go To The Advanced Screen



Initialize Token

SafeNet Authentication Client

Token Name: Joe's Token

☑ Set Token Password: *************

Confirm: *************

Logon retries before token is locked: 15

☑ Set Administrator Password: *************

Confirm: *************

Logon retries before token is locked: 15

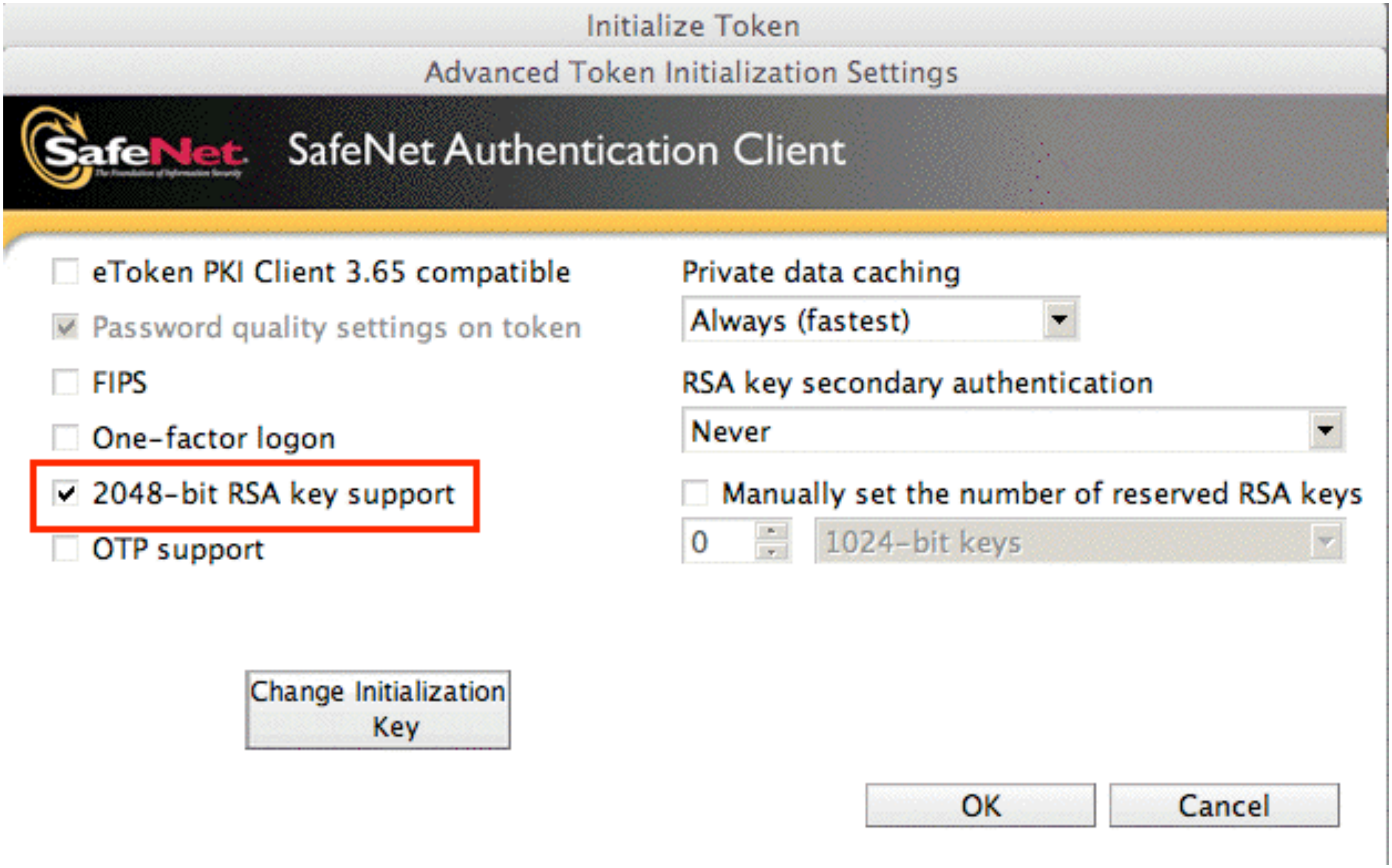
Note: An Administrator Password will be needed to unlock the token.

Additional Settings

□ Token Password must be changed on first logon

DO **NOT FORGET THESE CRITICAL PASSWORDS!**

Be Sure To Ask for 2048 bit key support



Initialize Token
Advanced Token Initialization Settings
SafeNet
SafeNet Authentication Client
Change Initialization Key
OK
Cancel

□ eToken PKI Client 3.65 compatible

☑ Password quality settings on token

□ FIPS

□ One-factor logon

☑ 2048-bit RSA key support

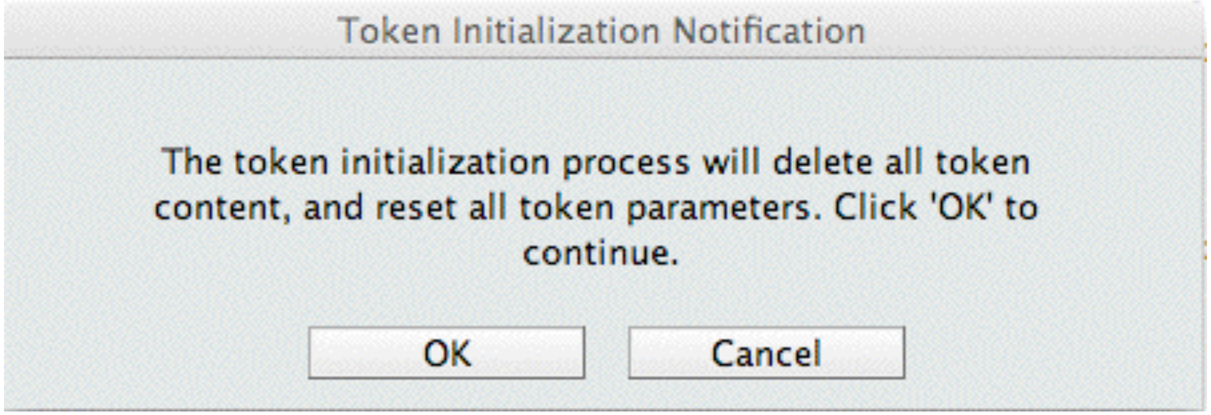
□ OTP support

Private data caching: Always (fastest)

RSA key secondary authentication: Never

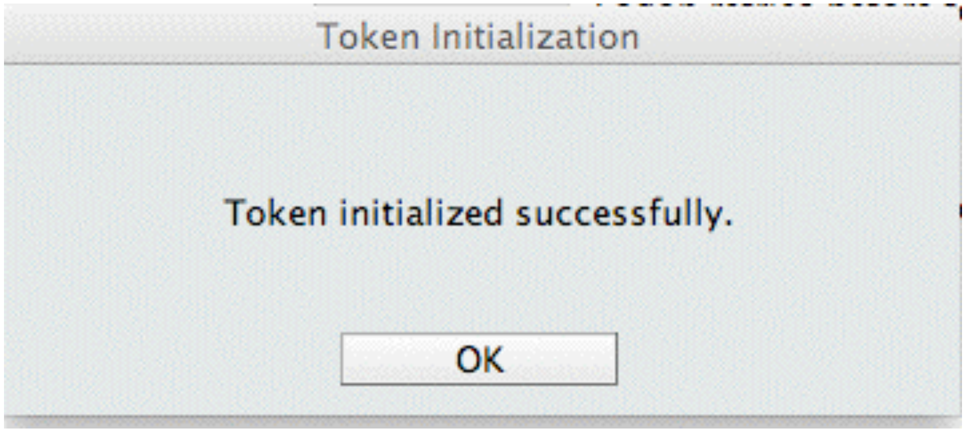
□ Manually set the number of reserved RSA keys: 0 1024-bit keys

Now Actually Initialize The Hard Token...



Token Initialization Notification

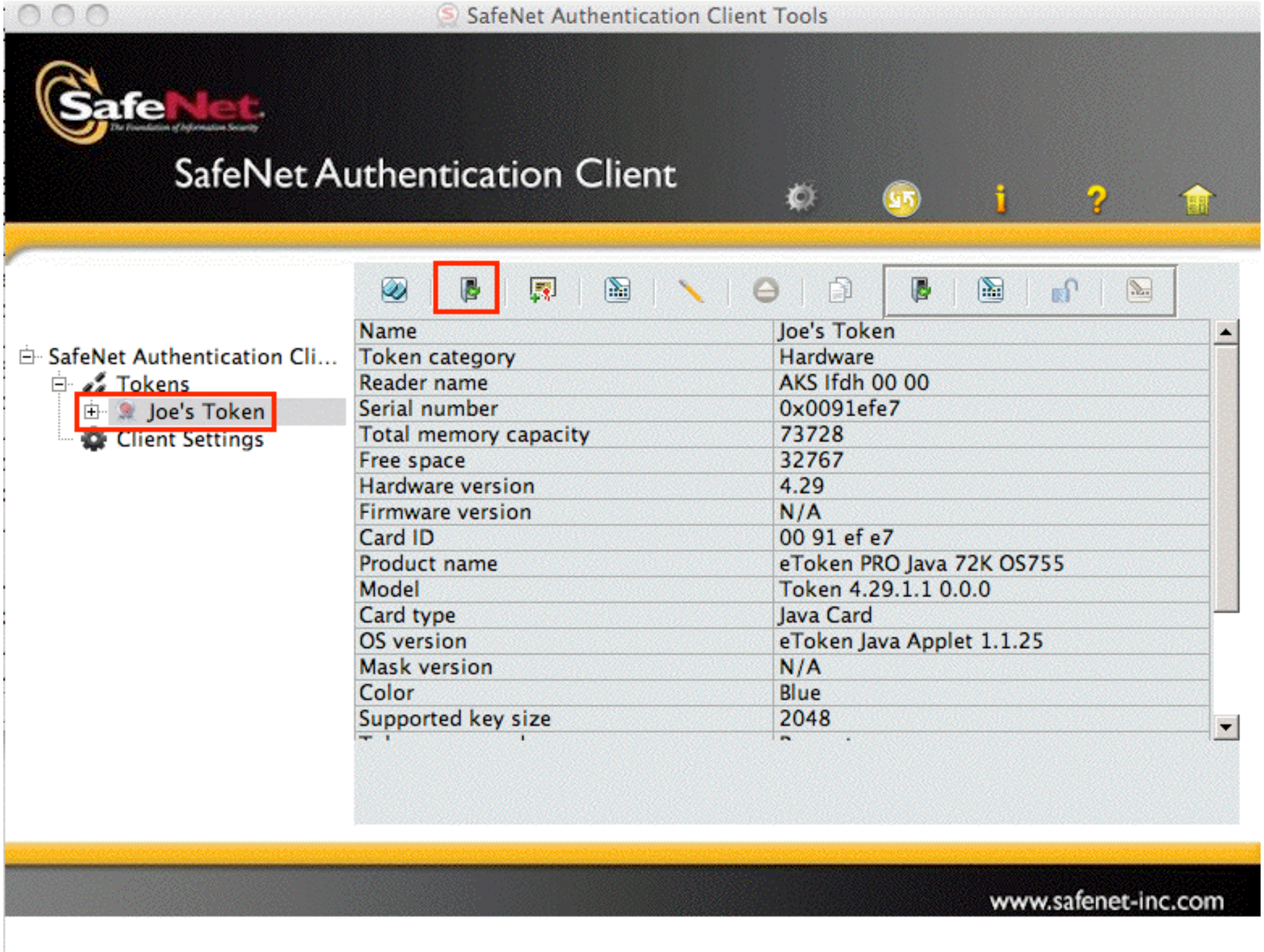
The token initialization process will delete all token content, and reset all token parameters. Click 'OK' to continue.



Token Initialization

Token initialized successfully.

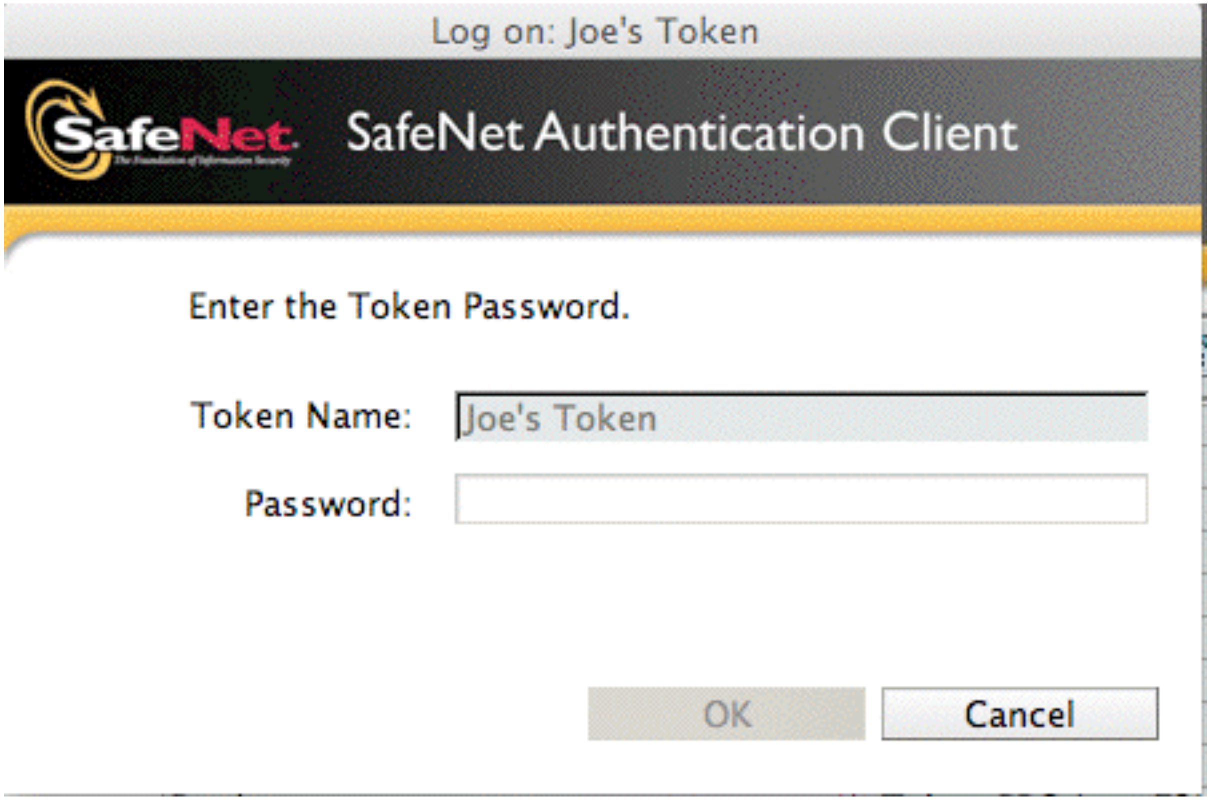
Login To The Hard Token



SafeNet Authentication Client Tools
SafeNet
The Foundation of Information Security
SafeNet Authentication Client
SafeNet Authentication Cli...
Tokens
Joe's Token
Client Settings
www.safenet-inc.com

Name	Joe's Token
Token category	Hardware
Reader name	AKS Ifdh 00 00
Serial number	0x0091efe7
Total memory capacity	73728
Free space	32767
Hardware version	4.29
Firmware version	N/A
Card ID	00 91 ef e7
Product name	eToken PRO Java 72K OS755
Model	Token 4.29.1.1 0.0.0
Card type	Java Card
OS version	eToken Java Applet 1.1.25
Mask version	N/A
Color	Blue
Supported key size	2048

You'll Need To Enter Your Password For It



Log on: Joe's Token

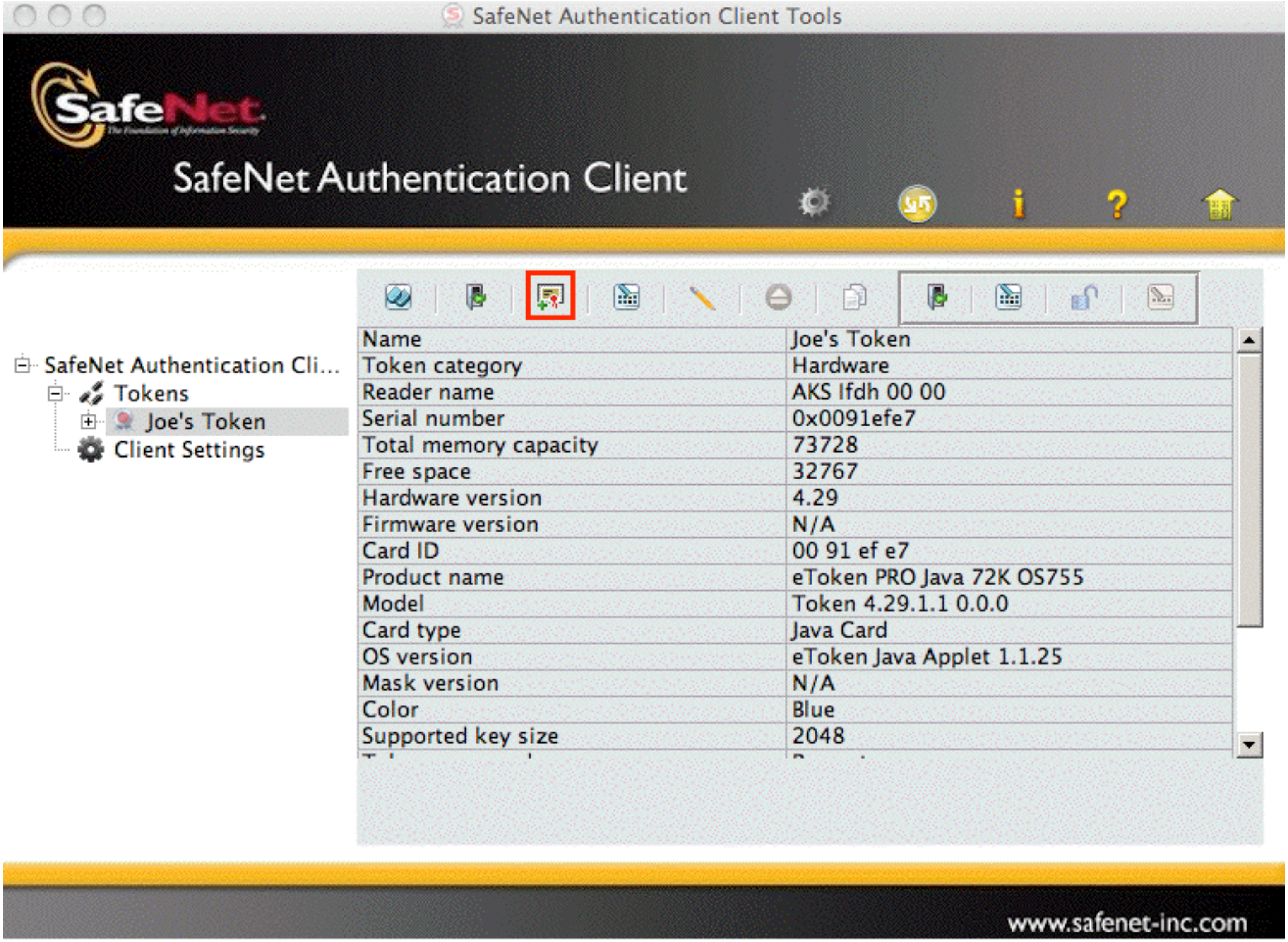
SafeNet Authentication Client

Enter the Token Password.

Token Name: Joe's Token

Password:

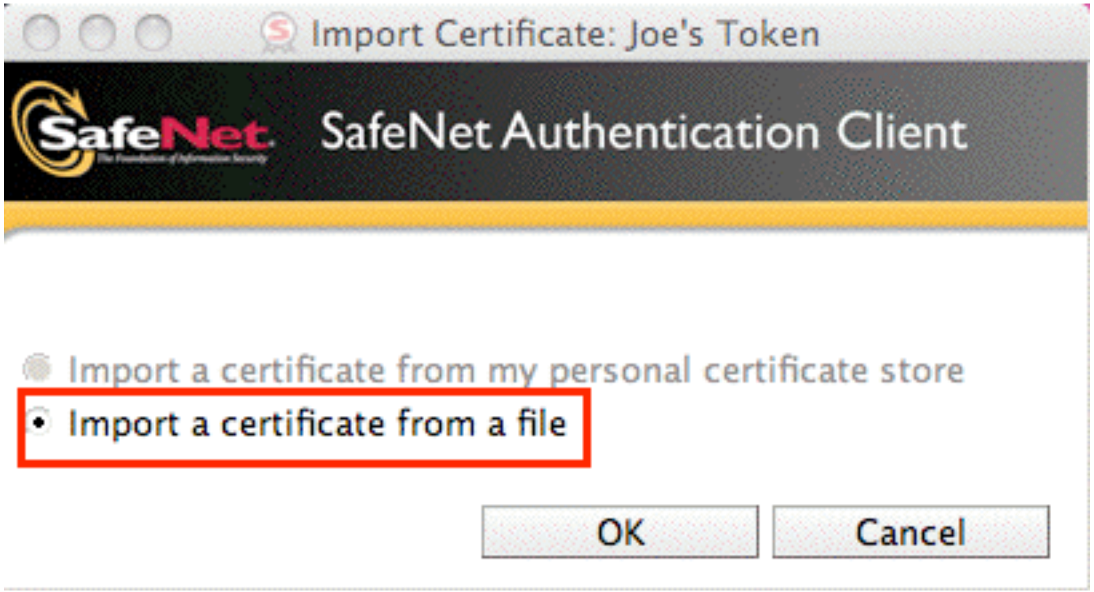
Go To The Import Cert Screen



SafeNet Authentication Client Tools
SafeNet
The Foundation of Information Security
SafeNet Authentication Client
SafeNet Authentication Cli...
Tokens
Joe's Token
Client Settings
www.safenet-inc.com

Name	Joe's Token
Token category	Hardware
Reader name	AKS Ifdh 00 00
Serial number	0x0091efe7
Total memory capacity	73728
Free space	32767
Hardware version	4.29
Firmware version	N/A
Card ID	00 91 ef e7
Product name	eToken PRO Java 72K OS755
Model	Token 4.29.1.1 0.0.0
Card type	Java Card
OS version	eToken Java Applet 1.1.25
Mask version	N/A
Color	Blue
Supported key size	2048

Import Our Certificate



Import Certificate: Joe's Token

SafeNet Authentication Client

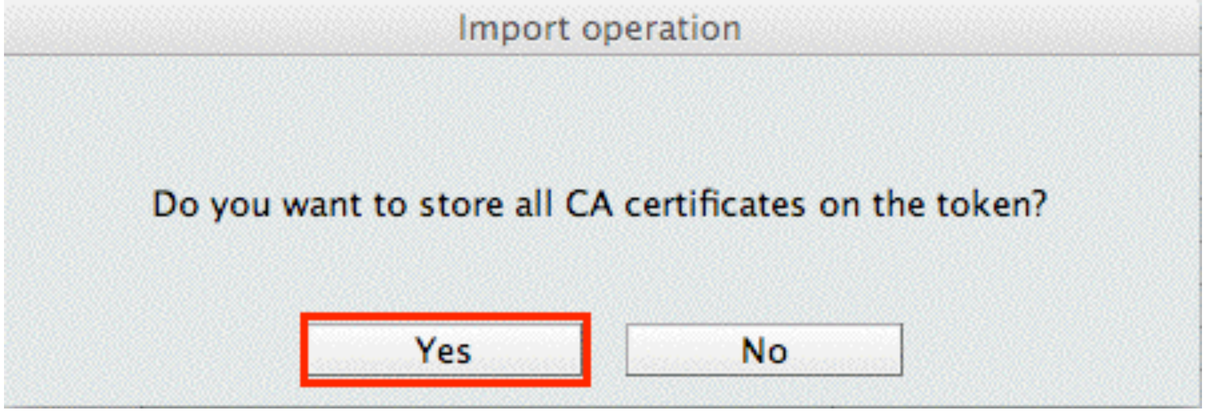
○ Import a certificate from my personal certificate store

● Import a certificate from a file

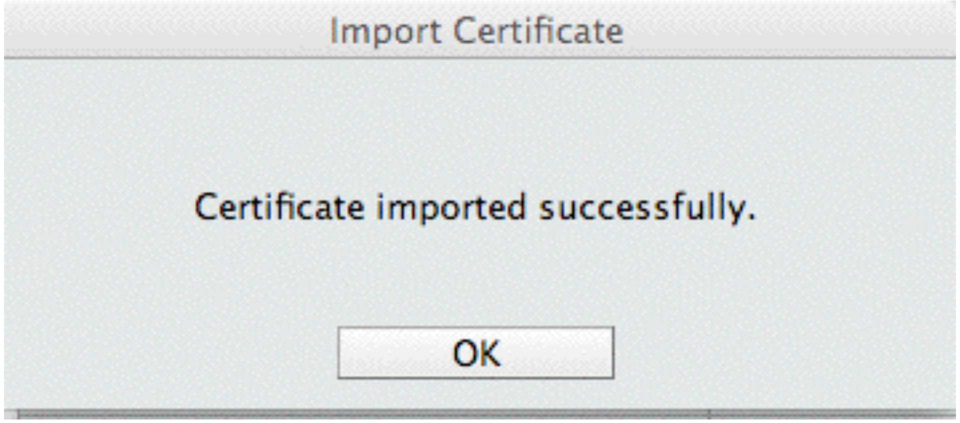
Pick the p12 backup file we saved earlier.

Note that you'll need to provide the password for that backup file in order to load it onto the token.

Be Sure To Include the CA Certs On The Token, Too

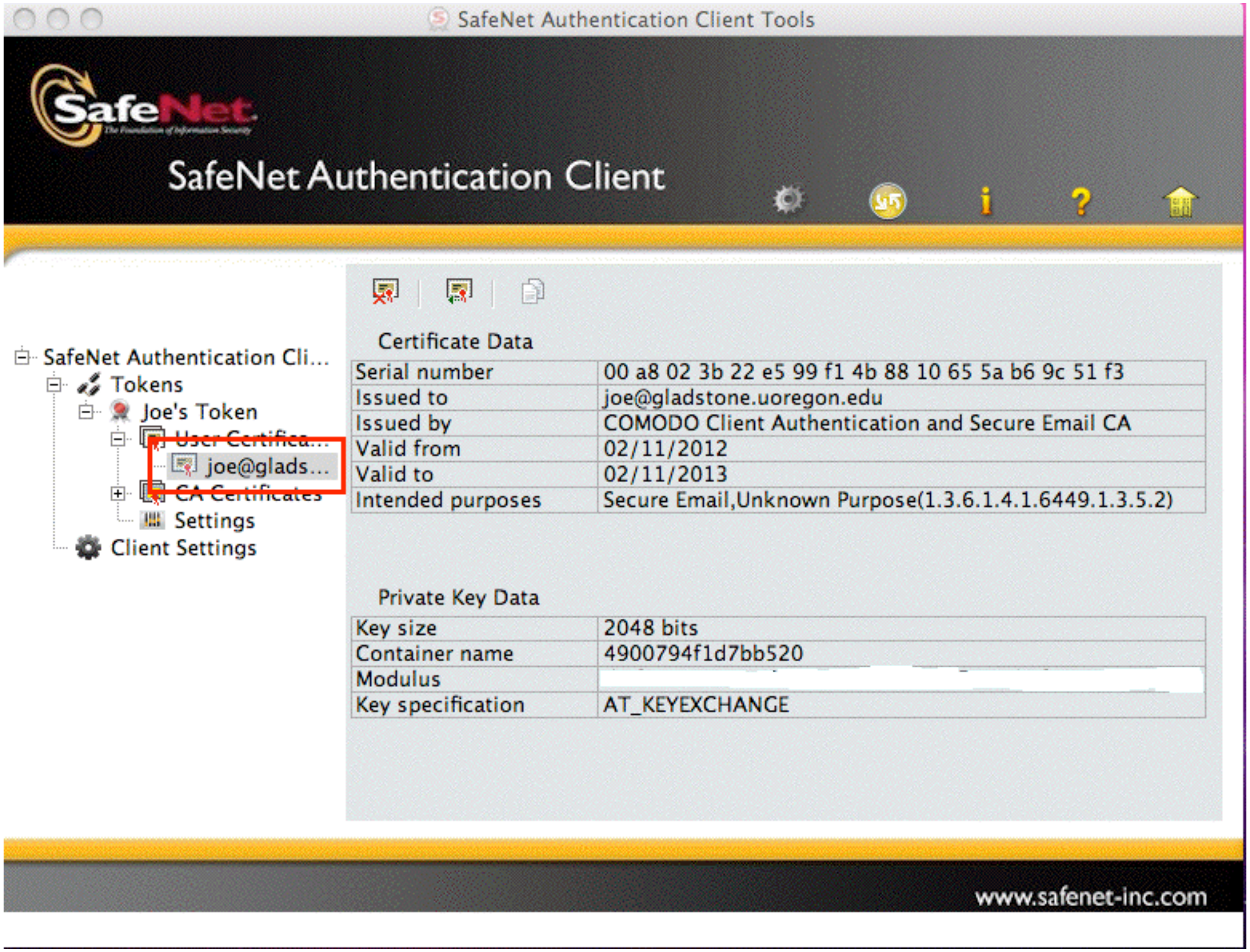


Do you want to store all CA certificates on the token?



Certificate imported successfully.

View Our Cert On The Hard Token


SafeNet Authentication Client Tools
SafeNet
The Foundation of Information Security
SafeNet Authentication Client
SafeNet Authentication Cli...
Tokens
Joe's Token
User Certifica...
joe@glads...
CA Certificates
Settings
Client Settings
www.safenet-inc.com

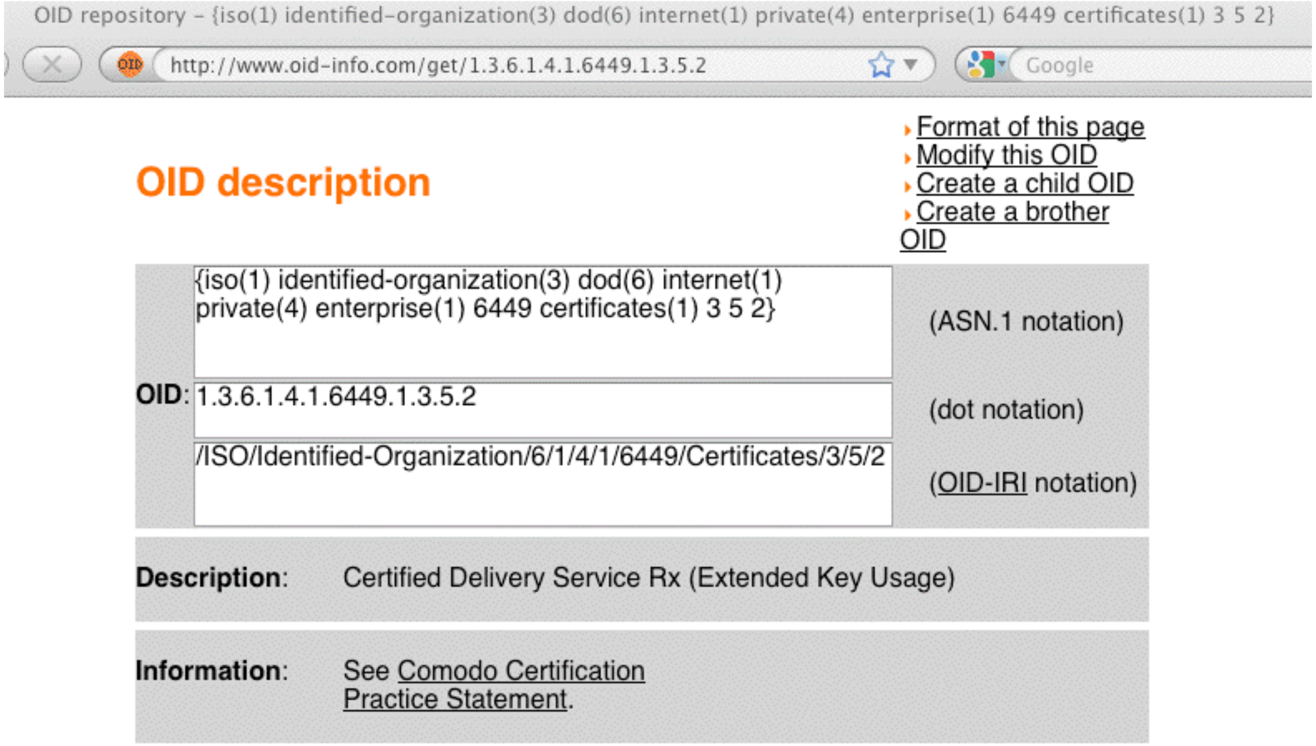
Certificate Data

Serial number	00 a8 02 3b 22 e5 99 f1 4b 88 10 65 5a b6 9c 51 f3
Issued to	joe@gladstone.uoregon.edu
Issued by	COMODO Client Authentication and Secure Email CA
Valid from	02/11/2012
Valid to	02/11/2013
Intended purposes	Secure Email,Unknown Purpose(1.3.6.1.4.1.6449.1.3.5.2)

Private Key Data

Key size	2048 bits
Container name	4900794f1d7bb520
Modulus	[redacted]
Key specification	AT_KEYEXCHANGE

An Aside: What's That "Unknown Purpose" Note?


OID repository - {iso(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1) 6449 certificates(1) 3 5 2}
http://www.oid-info.com/get/1.3.6.1.4.1.6449.1.3.5.2
Google
Format of this page
Modify this OID
Create a child OID
Create a brother OID

OID description

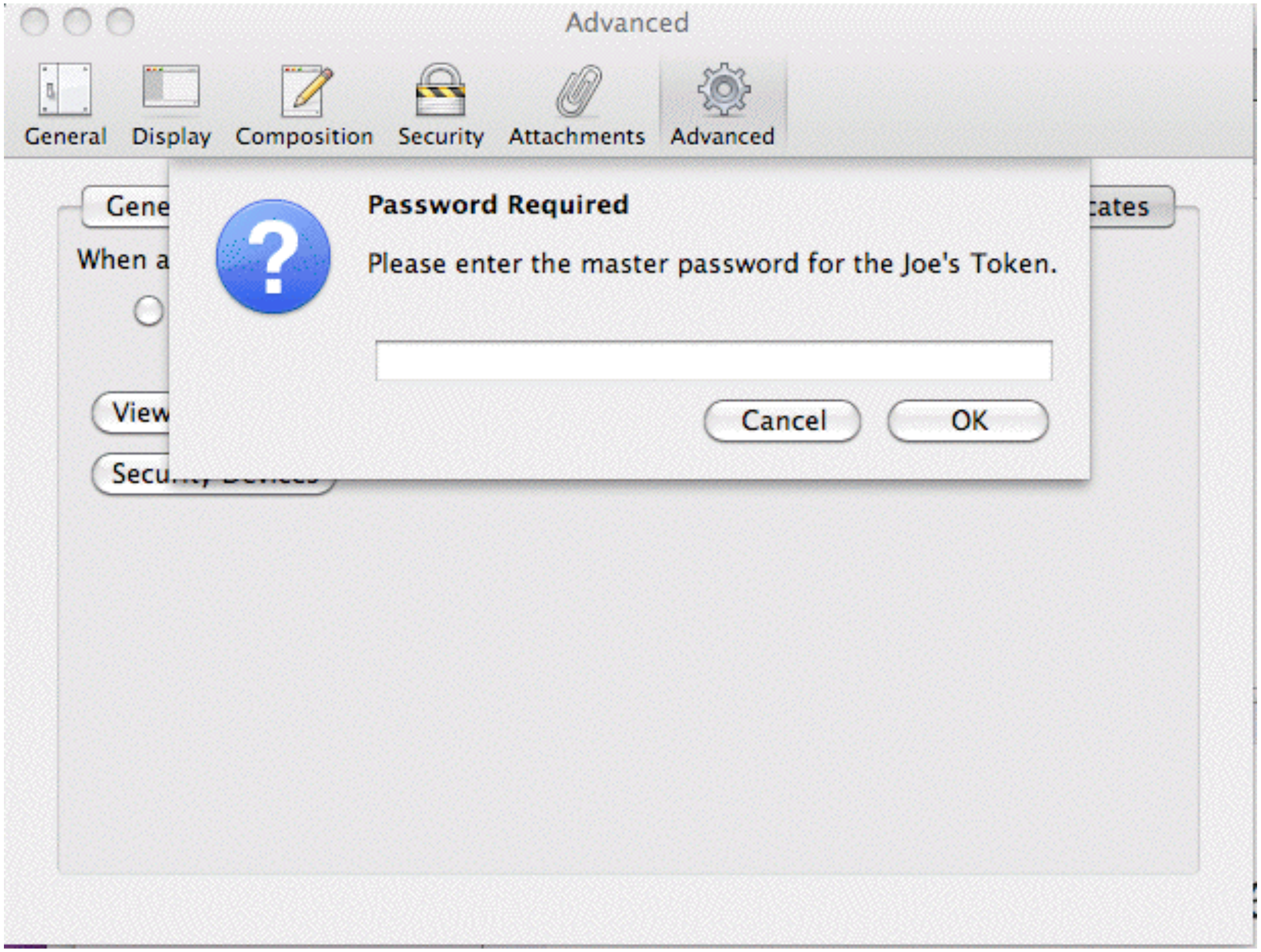
OID:	{iso(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1) 6449 certificates(1) 3 5 2}	(ASN.1 notation)
	1.3.6.1.4.1.6449.1.3.5.2	(dot notation)
	/ISO/Identified-Organization/6/1/4/1/6449/Certificates/3/5/2	(OID-IRI notation)

Description: Certified Delivery Service Rx (Extended Key Usage)

Information: See Comodo Certification Practice Statement.

But coming back to actually using our hard token...

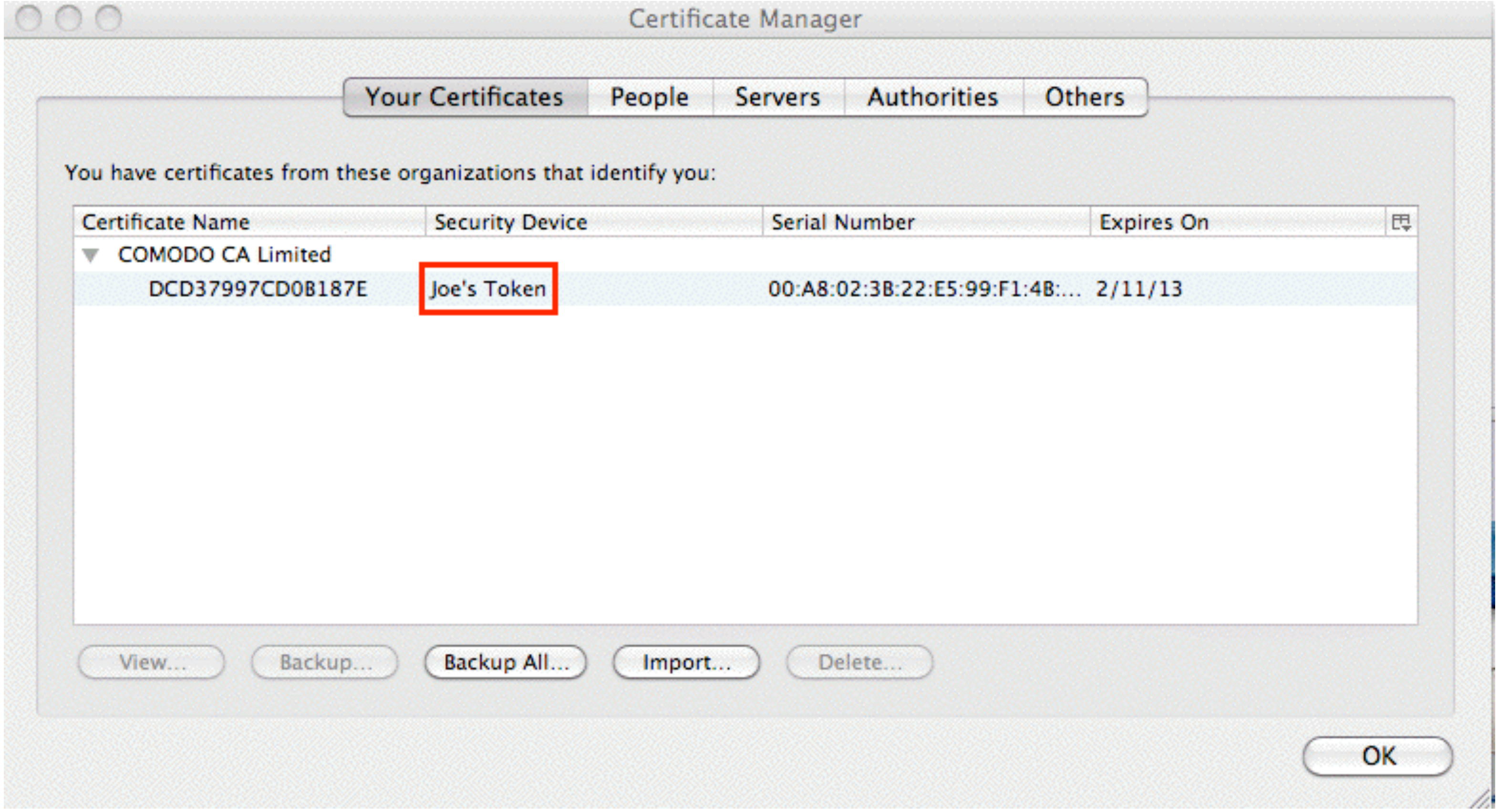
Telling Thunderbird To Use The Hard Token (We Need To Unlock The Token, First)



Password Required

Please enter the master password for the Joe's Token.

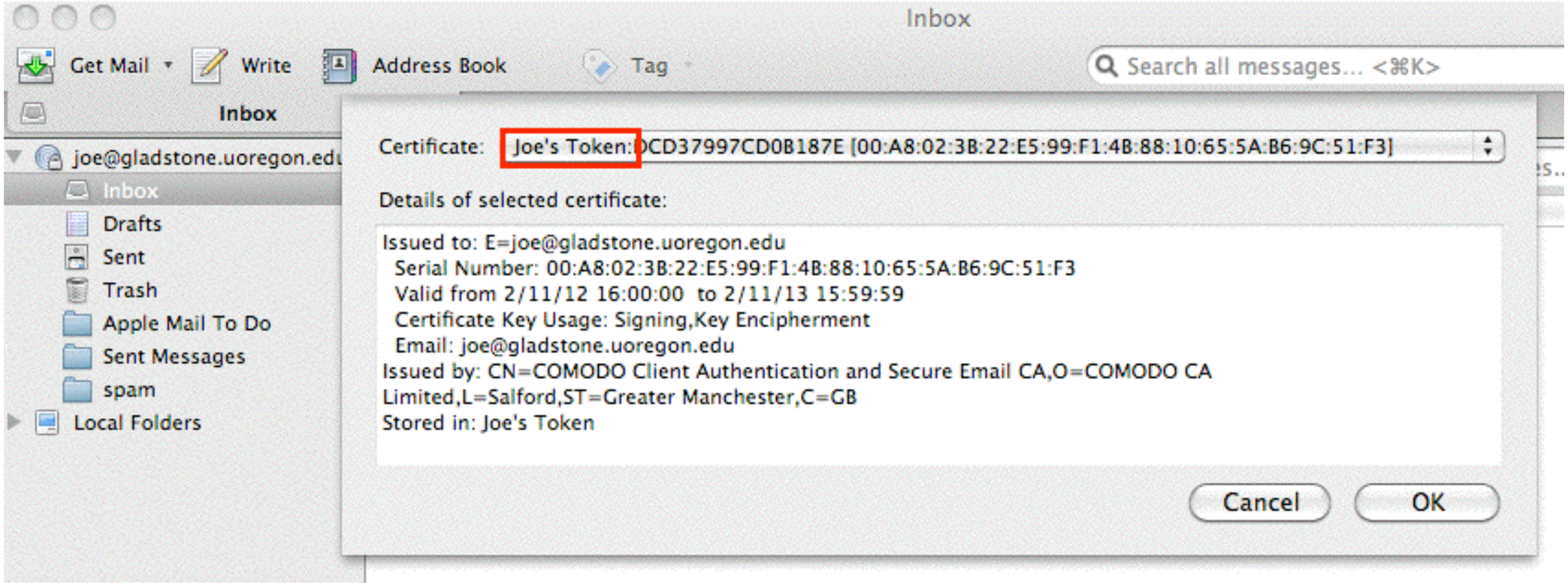
We're Then Shown The Token and Its Cert



You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
▼ COMODO CA Limited			
DCD37997CD0B187E	Joe's Token	00:A8:02:3B:22:E5:99:F1:4B:...	2/11/13

Now We Go To Thunderbird Accounts --> Security, And Select The Hard Token To Use

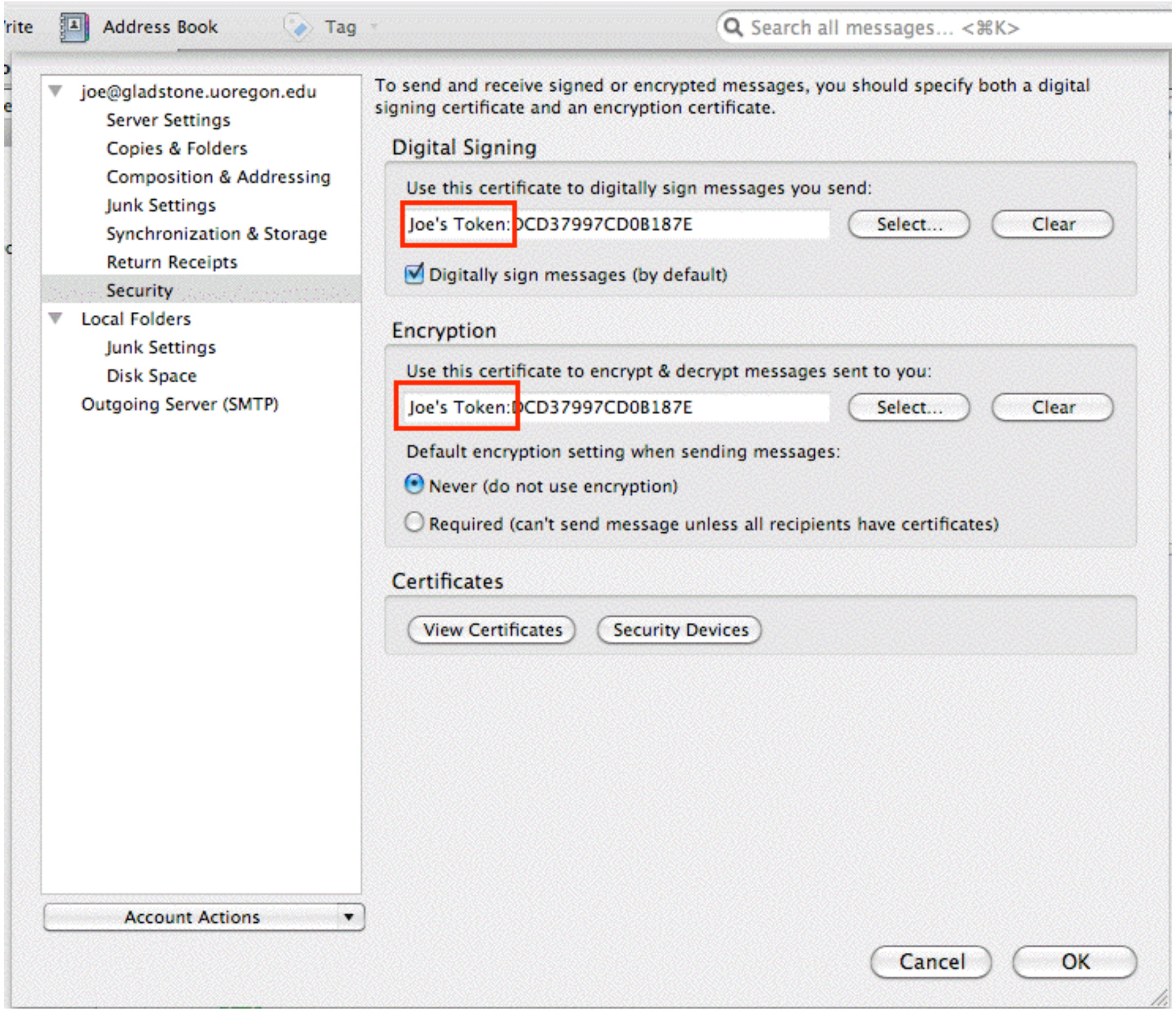


Certificate: Joe's Token:0CD37997CD0B187E [00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3]

Details of selected certificate:

```
Issued to: E=joe@gladstone.uoregon.edu
  Serial Number: 00:A8:02:3B:22:E5:99:F1:4B:88:10:65:5A:B6:9C:51:F3
  Valid from 2/11/12 16:00:00  to 2/11/13 15:59:59
  Certificate Key Usage: Signing,Key Encipherment
  Email: joe@gladstone.uoregon.edu
Issued by: CN=COMODO Client Authentication and Secure Email CA,O=COMODO CA
Limited,L=Salford,ST=Greater Manchester,C=GB
Stored in: Joe's Token
```

And At That Point We're Good To Go Using The Hard Token For Our Cert... Huzzah!



To send and receive signed or encrypted messages, you should specify both a digital signing certificate and an encryption certificate.

Digital Signing

Use this certificate to digitally sign messages you send: Joe's Token:0CD37997CD0B187E [Select...] [Clear]

☑ Digitally sign messages (by default)

Encryption

Use this certificate to encrypt & decrypt messages sent to you: Joe's Token:0CD37997CD0B187E [Select...] [Clear]

Default encryption setting when sending messages:

● Never (do not use encryption)

○ Required (can't send message unless all recipients have certificates)

Certificates

[View Certificates] [Security Devices]

[Cancel] [OK]

XI. Doing All This "At Scale"

Get A Little Experience, First

- It's sometimes tempting to "swing for the bleachers," trying to hit a grand slam the first time you're up to bat, when in fact the prudent thing might be to make sure you just get on base. This is true for client certs, as for baseball.
- I'd like to urge you, before you embark on a big project involving client certs, or even a pilot scale project that might involve some of your most sensitive systems, to first spend a little time just experimenting with client certs.
- Get a free client cert for yourself, and for your team members.
- Use them for relatively low impact activities, such as signing your email, while you gain familiarity with them.
- Try purchasing and using hardware tokens or smart cards. What works? What doesn't work on your devices or in your environment? In an experimental environment, you've got the freedom to push the envelope without worrying *too* much.

Client Cert Deployment Scale: Test, Departmental, Site-Wide, edu-Wide?

- We can imagine four different "scales" of client cert deployment:
  -- Test deployment (maybe half a dozen or a dozen client certs, perhaps issued only to highly technical systems or security staff)
  -- Departmental-scale deployment (hundreds or even thousands of certs, perhaps issued to all authorized administrative computing users or to all authorized high performance computing users at a site)
  -- Site-wide deployment to "everyone" (all faculty/staff, all students, and potentially even to all "other" users)
  -- Or maybe even broad edu-wide (cross-realm) deployment?
- ***These are radically different animals.*** If we DON'T need to do the cross-realm case, we might even be able to get along with locally issued client certs. Do you think that's one reason why email, a classic inter-realm app, has lead to client certs often being called 'S/MIME certs?'
  (If you're only issuing client certs for intra-realm use, at the same time you issue a cert, you could just push a local root cert).

Small Deployments? ==> Targeted Benefits
Larger Deployments? ==> Broad Acceptance

- While I don't mean to imply that there's no benefit to folks doing PKI testing, or even small scale deployments for a carefully defined local community, those sort of projects deliver a *different sort* of benefit than more broadly adopted efforts. **Has the time come for us to consider a broadly accepted cross-institutional client cert effort?**
- Contrast a locally-issued library card with a passport:
  -- A locally-issued library card is terrifically useful if I want to check out some books, but unfortunately no one except my library, e.g., the one that issued it, will recognize or accept it
  -- A passport, on the other hand, while not a document that will be accepted for the purpose of checking out library materials, is universally accepted as a proof of personal identity (including being potentially used or things like *getting* a local library card)

Time For A Standardized Higher-Ed-Wide ID Card?

- One of the reasons passports are useful is that they're ***standardized.*** Currently each university issues its own unique type of ID card, with little in the way of formal higher ed-wide standardization. Most have a name, a number (hopefully not a SSN!) and a picture. Most also have a mag swipe strip, a bar code, and maybe an RFID tag.
- **Has the time come for college and university ID cards to *<u>also</u>* have smart card functionality and a client cert? In fact, should higher ed be striving to establish a community-wide general standard for college and university ID cards? (arguably, there's already considerable *de facto* standardization)**
- *Note: I explicitly have no desire to step on card office "turf" at schools all across the country by innocently asking those questions! I do also recognize that there are a *lot* of subtle issues that are raised just by asking those two questions.*

What Works For Onesie-Twosie Won't Work For Tens of Thousands

- The processes you saw earlier in this session, which can be made to work for a small number of technically savvy users, won't work if you're trying to "cook for thousands" (or tens of thousands) of users. A more scalable approach is needed.
- For example, if you're going to install certificates directly on user systems, you need a better way to drop certificates on those systems, and a better way to configure the user's applications to know about and use them (InCommon is working on this).
- Similarly, if you're going to use hardware tokens, instead, you likely need enterprise grade tools to provision and manage those devices. Those tools can be purchased, or may be written locally.
- Heck, if we're thinking about a big deployment, we even need to carefully consider what SORT of hardware tokens we might want to use... USB format PKI hard tokens are NOT the only option.

Smartcards?

- The USB format PKI hard tokens you received are basically a smart card with an integrated smart card reader (with a built-in USB interface). That can be very convenient – it's "all in one."
- However, smart cards tend to be somewhat cheaper than USB format tokens (e.g., \$15.13 vs. \$19.80), which can be important if you're buying thousands of them. On the other hand, they do need smart card readers wherever the cards are going to be used (fortunately smart card readers need not be very expensive)
- A distinct advantage of smart cards is that they can be used as an employee badge or ID card, formatted to include things like the employee's name and picture, a mag stripe and one or more barcodes, while ALSO containing a smart card in a secure certificate store. This may be the best of all possible worlds.
- But what will you do for mobile devices, such as smart phones or tablets?

Slick-Sided Mobile Devices and Hard Tokens

- Mobile devices are increasingly important on campus, so we should be sure to think about how we'll integrate hard tokens or smart cards with mobile devices that your users may have, such as the iPad, the iPhone, Android devices, Blackberries, etc.
- The problem is that most hard tokens, and most smart card readers for that matter, connect via USB. Some portable devices may not have a readily accessible USB port into which you can plug a hard token or smart card reader.
- The solution? You can try Bluetooth-connected smartcard readers (sometimes also known as "CAC sleds"), but they aren't cheap and they don't support all devices or all smart cards.
- In the future, it may be possible to store client certs securely by storing part of the client cert directly on the device, while storing the rest of the client cert in the cloud, using threshold cryptography to reconstitute the client cert securely.

What About Directories

- One of the subtle things that can really make life easier if you're deploying client certificates at scale is a directory of all the public keys and certificates for the users you might need to communicate with (that means that people don't first need to exchange signed email messages before they can exchange encrypted email messages).
- Traditional key distribution also breaks down if you need non-repudiable keys for digital signing, but escrowed keys for encryption. You need an alternative source for keys in that case.
- When it comes to deploying a directory, deploying one for your company is one thing. Even deploying a directory for an entity as big as the federal government is something that's doable (heck, they've done it!). But it's not clear to me that there's a scalable Internet-wide directory solution that would work to hold client certificates for all Internet users (assuming everyone had them).

Some Directory Complications

- **Organizational directories are for local correspondents:** If all my email is local, and my site is doing client certs, I can probably just check my local directory, but these days, many users exchange more email **off**-site than on. And what if I'm an "isolated adopter," and there's not even an organizational directory for me to even use?
- **Organizational directories (distributed, Internet-wide):** How do I find the *right* directory to use to look up someone else's S/MIME creds? There's currently no "directory of directories" (nor do I think there's momentum/community support to create such an animal, given spam problems and security worries – many sites may be reluctant to allow unfettered public directory access due to potential harvesting issues).
- **What about a centralized/consolidate Internet-wide directory that lists "everyone?"** Um, no. People just won't want to contribute their data, it would be impossible to keep current, and there are O(20 million) users in US higher ed! We need to take a lesson from DNS. The architects of DNS did a distributed model for good reasons!

PGP/GPG-ish S/MIME Keyservers?

- There is one alternative cryptographic directory model that seems to have worked pretty well to-date, and that's the PGP/GPG model. Users can submit their keys if they want to. Other users can look for keys in those directories if they want to. If you can't find the one you need, you can always fall back on old standby approaches, like asking users to send their key directly.

- I've developed a very rough prototype server that demonstrates that it is at least conceptually possible to construct a PGP/GPG-like key server for S/MIME. If you're interested, see
  http://pages.uoregon.edu/joe/simple-keyserver/
  for a detailed description of what I have in mind.

S/MIME Isn't The Only Use for Client Certs

- Client certificates can be used for a bunch of things other than just signing or encrypting email.
- For example, client certificates can also be used to sign documents, or for authentication, or as a building entry credential. (Note that if you're headed in the "authentication" or "building access control" direction, you *will* probably need a traditional enterprise PKI directory to support that application)
- Once you have client certs deployed, you might be surprised at how many different ways they can actually be used.
- **NOTE: Client certs should only be used for purposes consistent with their approved uses. For example, the client cert we downloaded earlier specified that it was for use in conjunction with secure email**. However, many applications do NOT strictly check/enforce the Object IDs ("OIDs") associated with a cert, so you may be able to use a given cert for other purposes, too.

Signing Stuff (Other Than Just S/MIME Signing)

- Signing **Microsoft Word documents** (Windows only), see http://pages.uoregon.edu/joe/signing-a-word-document/

- Need to sign documents on a Mac? Try **OpenOffice**: http://tinyurl.com/openoffice-signing

- Adobe has an extensive guide to securing PDFs, including use of digital certificates for **signing PDFs**, see: http://tinyurl.com/adobe-signing (PDF, 114 pages)

  Note that this is different than Adobe's "Certified Document Services" program which also involves digital signatures, but is more expensive (and not supported by Comodo/InCommon client certs at this time)

Encryption Using Client Certs (Other Than S/MIME)

- **PGP Whole Disk Encryption** (see the datasheet linked from http://www.symantec.com/business/whole-disk-encryption)

- **Microsoft Windows Encrypted File System** http://technet.microsoft.com/en-us/library/bb457116.aspx

- **IPsec VPNs** (Most IPsec VPNs are deployed without use of client certificates, however at least some VPNs can be configured to use client certificates if desired — see, for example, http://www.strongswan.org/ and http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/DCertPKI.html)

Authentication Using Smart Cards/Client Certs

- **RedHat Enterprise Linux** Smart Card Login
  See http://tinyurl.com/redhat-smartcards

- **Windows Active Directory** Login with Smart Cards
  See http://support.microsoft.com/kb/281245

- **OpenSSH authentication** (via third party X.509 patches)
  http://roumenpetrov.info/openssh/

- **Mac OS X** has been going through some changes when it comes to native support for smart cards, but see http://smartcardservices.macosforge.org/ and http://www.thursby.com/mac-enterprise-management-high-security-smart-cards.html

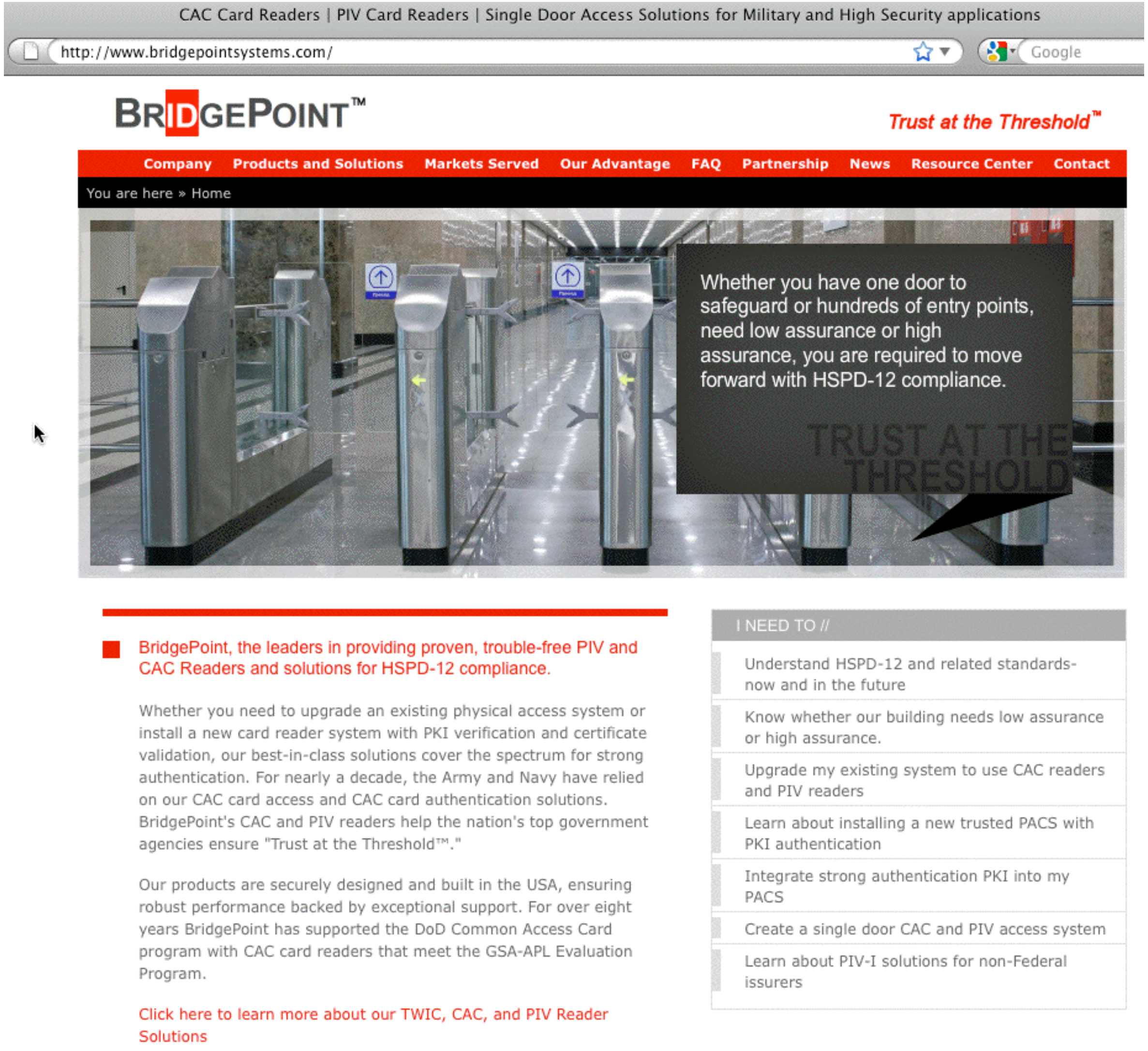
Authentication Using Client Certs (cont.)

- Controlling access to web content served by **Apache**: www.dwheeler.com/essays/apache-cac-configuration.html (it's much more helpful than the more general page at httpd.apache.org/docs/2.5/mod/mod_ssl.html#sslrequire)
- Controlling access to web content served by **Microsoft IIS7** http://technet.microsoft.com/en-us/library/cc732996%28v=ws.10%29.aspx
- Controlling access to **wireless networks** via EAP-TLS, including configuring **Eduroam**. See

  http://www.cisco.com/en/US/tech/tk722/tk809/technologies_white_paper09186a008009256b.shtml and

  http://www.internet2.edu/presentations/jt2011summer/20110710-hagley-eduroamtutorial.pdf

Client Certificates Can Even Potentially Be Used For Building Access Control Purposes



Whether you have one door to safeguard or hundreds of entry points, need low assurance or high assurance, you are required to move forward with HSPD-12 compliance.

BridgePoint, the leaders in providing proven, trouble-free PIV and CAC Readers and solutions for HSPD-12 compliance.

Whether you need to upgrade an existing physical access system or install a new card reader system with PKI verification and certificate validation, our best-in-class solutions cover the spectrum for strong authentication. For nearly a decade, the Army and Navy have relied on our CAC card access and CAC card authentication solutions. BridgePoint's CAC and PIV readers help the nation's top government agencies ensure "Trust at the Threshold™."

Our products are securely designed and built in the USA, ensuring robust performance backed by exceptional support. For over eight years BridgePoint has supported the DoD Common Access Card program with CAC card readers that meet the GSA-APL Evaluation Program.

Click here to learn more about our TWIC, CAC, and PIV Reader Solutions

I NEED TO //

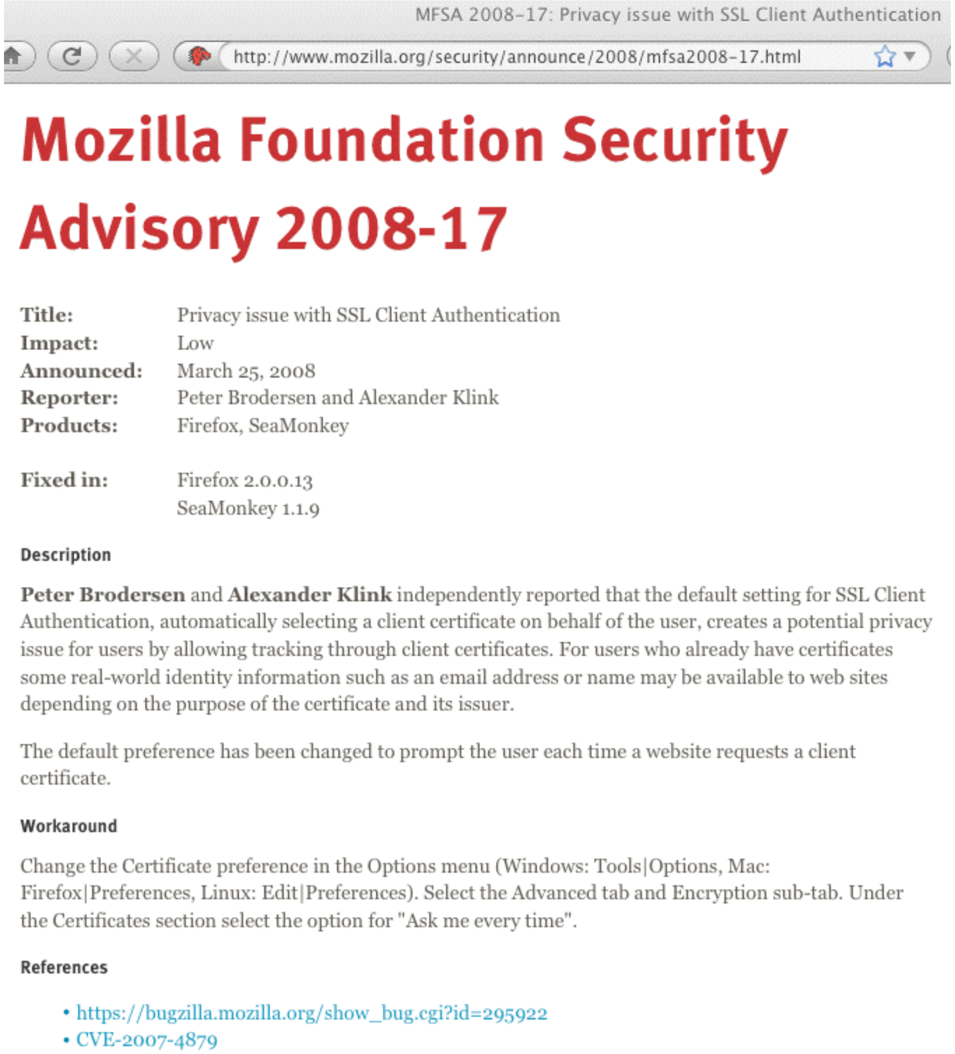
- Understand HSPD-12 and related standards- now and in the future
- Know whether our building needs low assurance or high assurance.
- Upgrade my existing system to use CAC readers and PIV readers
- Learn about installing a new trusted PACS with PKI authentication
- Integrate strong authentication PKI into my PACS
- Create a single door CAC and PIV access system
- Learn about PIV-I solutions for non-Federal issurers

XII. Don't Forget About Policies, Governance And Potential Legal Issues

Client Certs (The Technology) Need to Be Supported By Appropriate Policies and Governance Structures

- In looking at successful deployments of client certs, such as the federal government's HSPD-12 CAC/PIV card project, one of the things that's hard to miss is that its success is not just a technological thing, it's a sign that appropriate policies were developed by the issuing and relying communities.
- If you're planning on doing a major client cert project, please be sure you are also considering the policy implications of moving to client certs, not just the technology issues.
- For example, what about privacy? Does use of client certs have any impact on user privacy? Maybe...
- What if your email client checked a directory for a public key/cert for every email correspondent you exchanged email with?
- Or how about this little exposure... see the next slide...

Any Web Site Can Ask For Your Browser's Client Cert And Thus Potentially Get Your Name/Email Address


MFSA 2008-17: Privacy issue with SSL Client Authentication
http://www.mozilla.org/security/announce/2008/mfsa2008-17.html

Mozilla Foundation Security Advisory 2008-17

Title:	Privacy issue with SSL Client Authentication
Impact:	Low
Announced:	March 25, 2008
Reporter:	Peter Brodersen and Alexander Klink
Products:	Firefox, SeaMonkey
Fixed in:	Firefox 2.0.0.13 SeaMonkey 1.1.9

Description

Peter Brodersen and **Alexander Klink** independently reported that the default setting for SSL Client Authentication, automatically selecting a client certificate on behalf of the user, creates a potential privacy issue for users by allowing tracking through client certificates. For users who already have certificates some real-world identity information such as an email address or name may be available to web sites depending on the purpose of the certificate and its issuer.

The default preference has been changed to prompt the user each time a website requests a client certificate.

Workaround

Change the Certificate preference in the Options menu (Windows: Tools|Options, Mac: Firefox|Preferences, Linux: Edit|Preferences). Select the Advanced tab and Encryption sub-tab. Under the Certificates section select the option for "Ask me every time".

References

- https://bugzilla.mozilla.org/show_bug.cgi?id=295922
- CVE-2007-4879

Another Privacy Threat: Client Certs Are Now Being Targeted By Malware

- Users who employed client certs for two factor authentication have long enjoyed feeling relatively "above the fray" when it came to hacker/cracker attacks. However, in 2012, it became clear that at least one malware family, Sykipot, has begun to specifically target federal CAC/PIV client certificate credentials. See, for example: http://labs.alienvault.com/labs/index.php/2012/when-the-apt-owns-your-smart-cards-and-certs
- Because client cert credentials are typically "nonexportable" from smart cards, malware targeting client certs will normally attempt to execute a "man in the browser" or "man in the machine" attack:
  -- intercept the user's smart card PIN,
  -- use the client cert "in-situ," proxying requests for resources
     controlled by certs through the compromised machine itself, then
  -- exfiltrate the surreptitiously accessed materials offsite.
- Conscientious patching and aggressive measures to control malware, remain extremely important, even if (especially if?) you're using client certificates to control access to sensitive content.

Keep Your Lawyers In The Loop, Too

- Why? Well, let me give you one closing example... strong cryptography is export controlled by the U.S. Bureau of Industry and Security, including being subject to the "deemed export" rule.

  If you plan to issue client certificates to all your employees remember that some users, as mentioned at the beginning of this talk, may not be eligible for access to strong cryptographic technologies, including potentially client certificates. For more on this point, please consult with your attorney regarding the provisions of the "Deemed Export" rule. As a starting point, see http://www.bis.doc.gov/deemedexports/deemedexportsfaqs.html

- Increased use of encryption for official records, may also raise long term record management and access issues.

Thanks for the Chance To Talk Today!

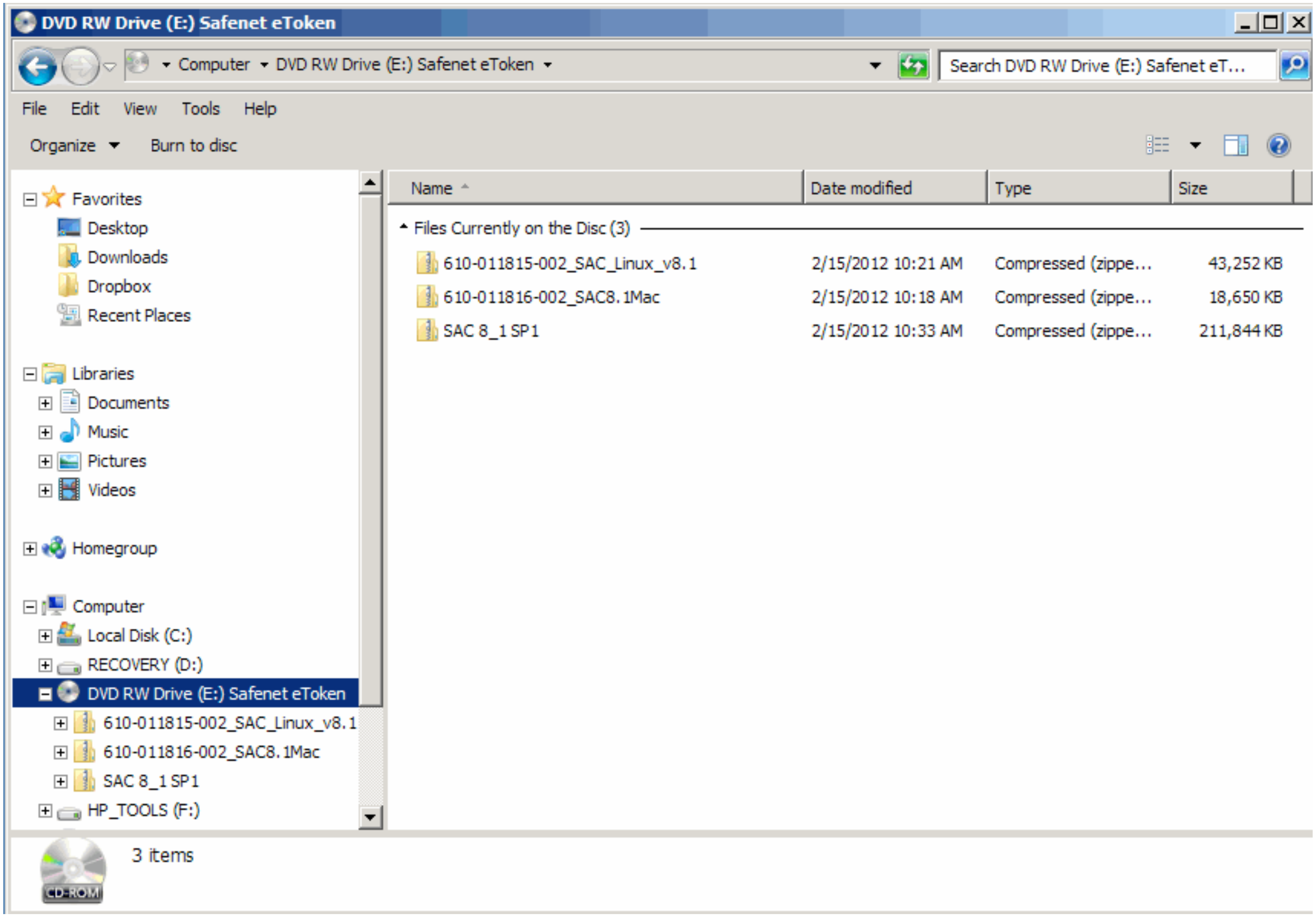
- Are there any questions?

Appendix I: Using The SafeNet Hard Token on Windows 7

"I'm Using Windows, Not A Mac!"

- There's a version of the SAC for Windows 7 on the CD we gave you, too.
- Drag the SAC 8_1 SP 1 zipped archive from the CD to your desktop. Double click on it, then select the SAC 8_1 SP 1 folder.
- Go to the 32X64Installer folder. Drag the application you'll see there onto your desktop.
- Assuming you're running Windows 7, right click on the installer and select Run as Administrator.
- You should see then go through a series of screens where the default answers will usually fine... see the next slides.

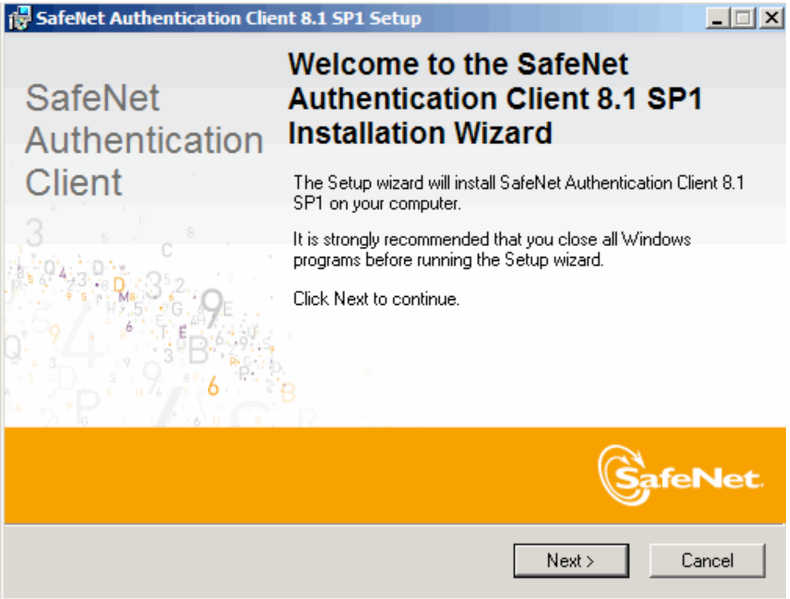
The CD's Contents



Files Currently on the Disc (3)

Name	Date modified	Type	Size
610-011815-002_SAC_Linux_v8.1	2/15/2012 10:21 AM	Compressed (zippe...	43,252 KB
610-011816-002_SAC8.1Mac	2/15/2012 10:18 AM	Compressed (zippe...	18,650 KB
SAC 8_1 SP1	2/15/2012 10:33 AM	Compressed (zippe...	211,844 KB

3 items

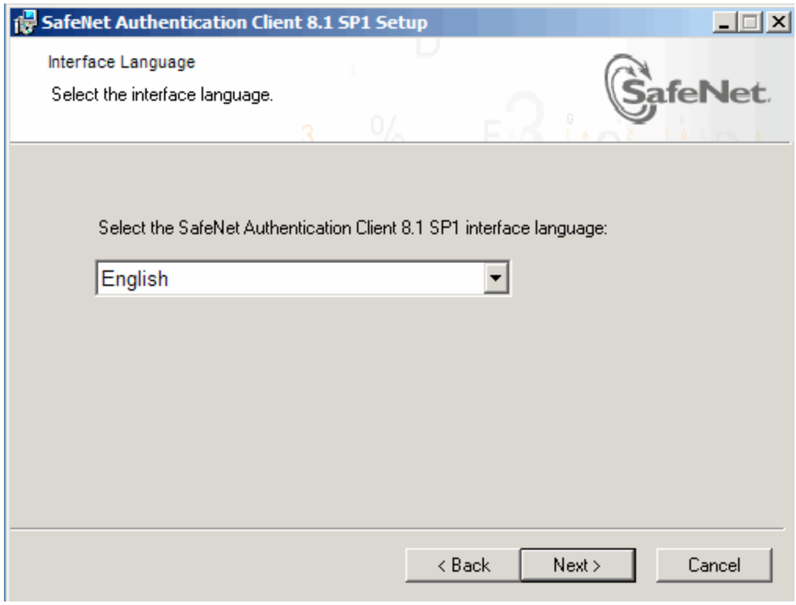

SafeNet Authentication Client 8.1 SP1 Setup
SafeNet Authentication Client
SafeNet
Next >
Cancel

Welcome to the SafeNet Authentication Client 8.1 SP1 Installation Wizard

The Setup wizard will install SafeNet Authentication Client 8.1 SP1 on your computer.

It is strongly recommended that you close all Windows programs before running the Setup wizard.

Click Next to continue.

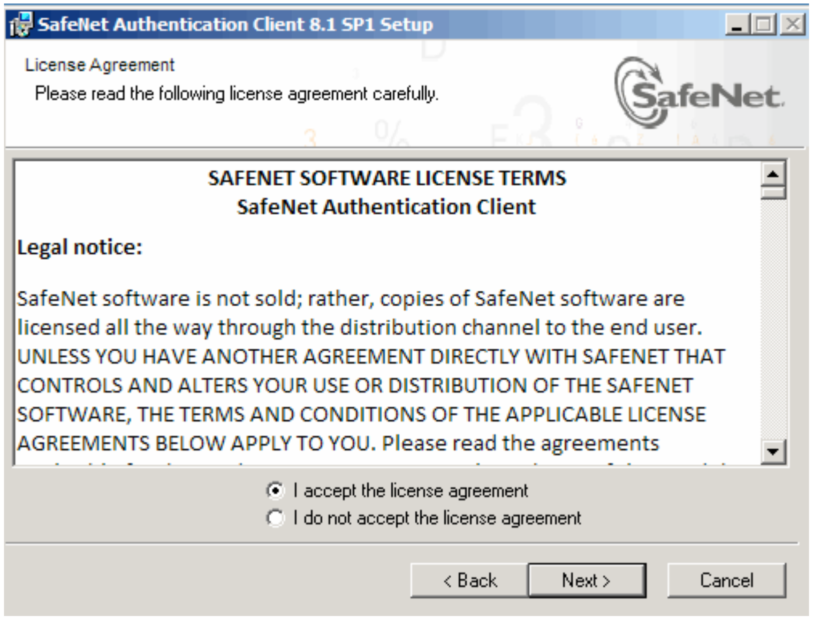


Interface Language

Select the interface language.

Select the SafeNet Authentication Client 8.1 SP1 interface language:

English



License Agreement

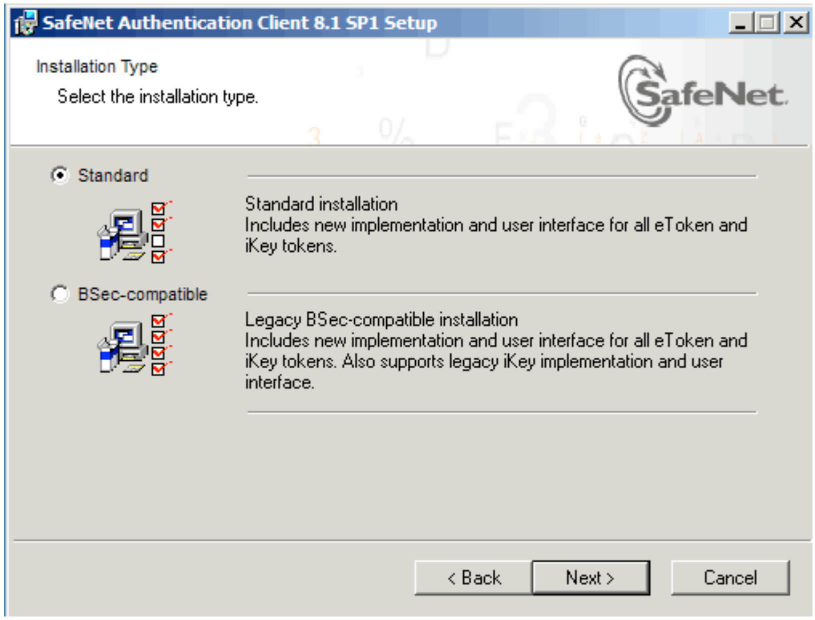
Please read the following license agreement carefully.

SAFENET SOFTWARE LICENSE TERMS
SafeNet Authentication Client

Legal notice:

SafeNet software is not sold; rather, copies of SafeNet software are licensed all the way through the distribution channel to the end user. UNLESS YOU HAVE ANOTHER AGREEMENT DIRECTLY WITH SAFENET THAT CONTROLS AND ALTERS YOUR USE OR DISTRIBUTION OF THE SAFENET SOFTWARE, THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENTS BELOW APPLY TO YOU. Please read the agreements

● I accept the license agreement
○ I do not accept the license agreement



Installation Type

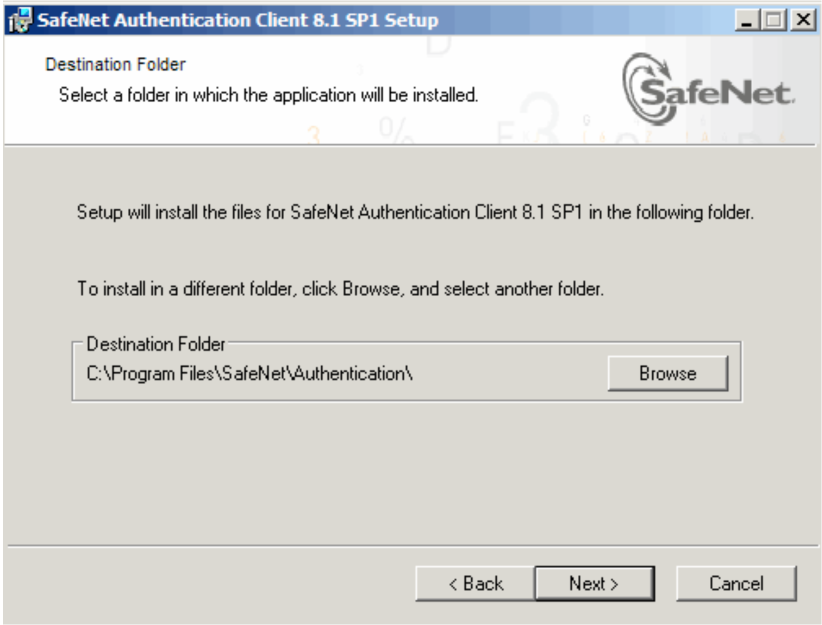
Select the installation type.

● Standard

Standard installation
Includes new implementation and user interface for all eToken and iKey tokens.

○ BSec-compatible

Legacy BSec-compatible installation
Includes new implementation and user interface for all eToken and iKey tokens. Also supports legacy iKey implementation and user interface.



SafeNet Authentication Client 8.1 SP1 Setup

Destination Folder

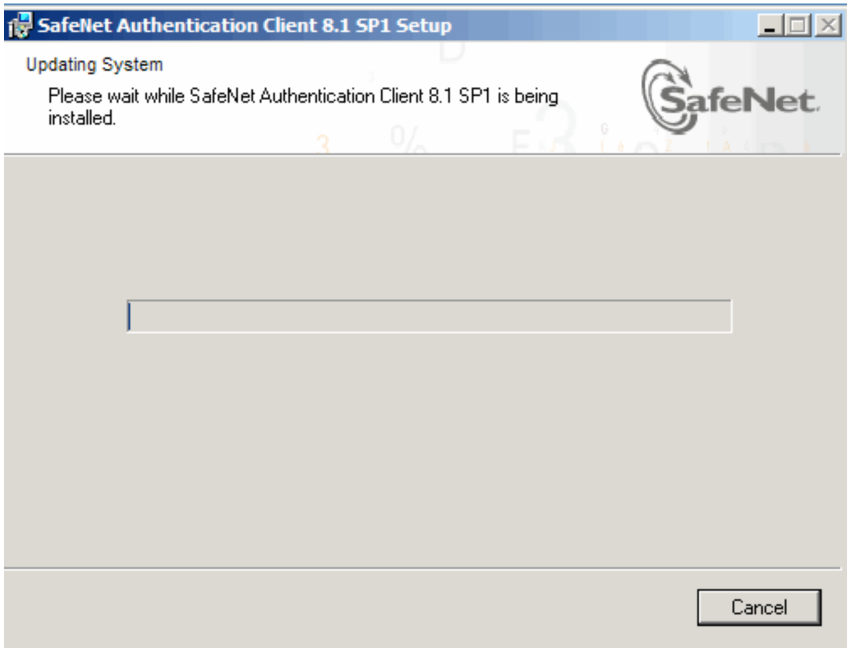
Select a folder in which the application will be installed.

Setup will install the files for SafeNet Authentication Client 8.1 SP1 in the following folder.

To install in a different folder, click Browse, and select another folder.

Destination Folder

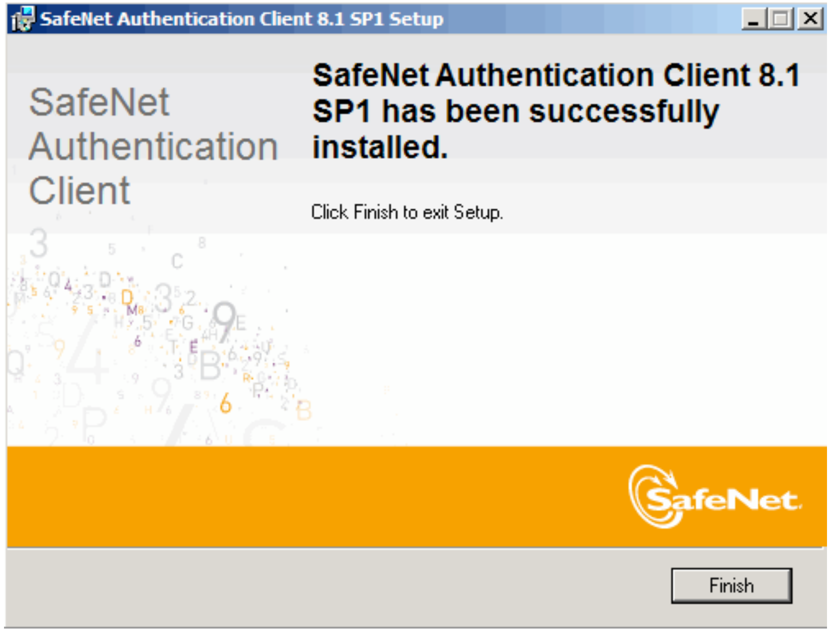
C:\Program Files\SafeNet\Authentication\



SafeNet Authentication Client 8.1 SP1 Setup

Updating System

Please wait while SafeNet Authentication Client 8.1 SP1 is being installed.

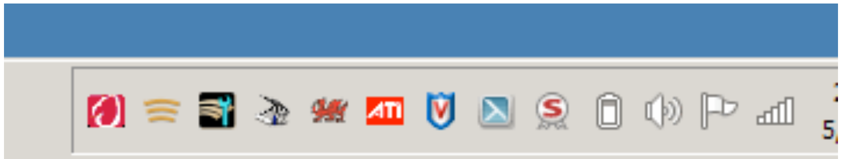


SafeNet Authentication Client 8.1 SP1 Setup

SafeNet Authentication Client

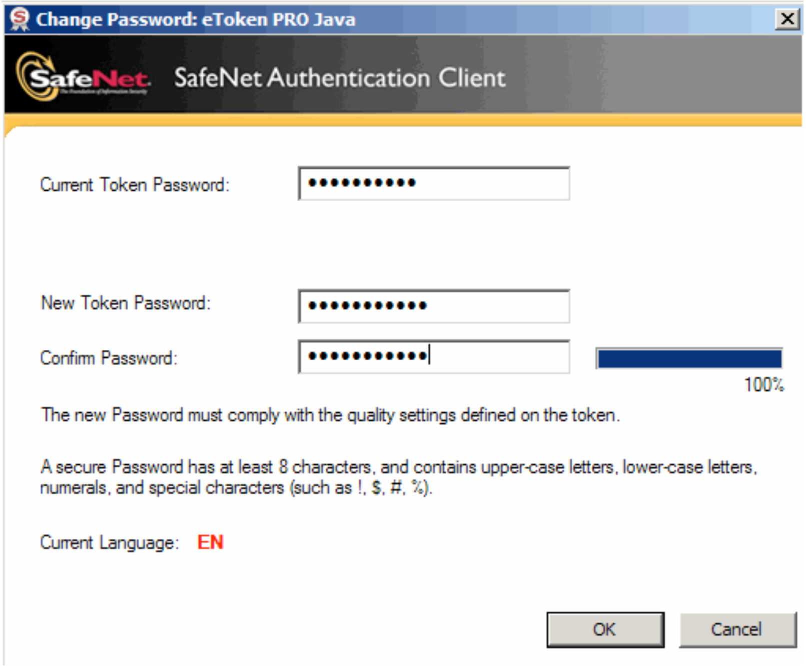
SafeNet Authentication Client 8.1 SP1 has been successfully installed.

Click Finish to exit Setup.



Plug In Your Token

- When you do, it may automatically download additional drivers from Windows Update. The first time, when it finishes, it will prompt you to change your token's password. The default password is 1234567890 as mentioned in the documentation.



Change Password: eToken PRO Java

SafeNet Authentication Client

Current Token Password: ••••••••••

New Token Password: •••••••••••

Confirm Password: •••••••••••

100%

The new Password must comply with the quality settings defined on the token.

A secure Password has at least 8 characters, and contains upper-case letters, lower-case letters, numerals, and special characters (such as !, $, #, %).

Current Language: EN

Thunderbird Can't See The SafeNet Hard Tokens?

- Initially, Thunderbird (and potentially Firefox) may not "see" the SafeNet hard token. If you experience that, you'll need to manually load the eTPKCS11.dll file from either

  ```
  c:\Windows\System32\eTPKCS11.dll     (32 bit)   or
  c:\Windows\SysWOW64\eTPKCS11.dll     (64 bit)
  ```

  Firefox --> Preferences... --> Advanced
  In the Encryption tab, click on Security Devices
  In the Device Manager window, click Load
  In the Load PKCS#11 Device window, under Module filename, enter the appropriate filename (as shown above)
  In the Confirm window, click OK