

# Client Certs and S/MIME Signing and Encryption: An Introduction

MAAWG 24

12:30-2:30, Monday, Feb 20, 2012  
Olympic Room, Westin Market St, SFO

Joe St Sauver, Ph.D. (joe@uoregon.edu)  
MAAWG Senior Technical Advisor

<http://pages.uoregon.edu/joe/maawg24/>

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

# **Preface**

# 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 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 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](http://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 employer) is NOT "locked down" by your corporate IT department.
- If you have NOT backed up your system recently, or your corporate 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 S/MIME and client certificates, and getting you to the point where you understand the practical limitations associated with those technologies. You do not need advanced mathematics to do that.
- So if you hated mathematics in high school or college, relax. :-)  
Virtually everything we're going to talk about today should be non-mathematical.
- Let's dive right in.

# **I. Introduction**

# 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:

**Default Identity**

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

Your Name:

Email Address:

Reply-to Address:

Organization:

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

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

☐ Attach my vCard to messages

Outgoing Server (SMTP):

[Yes, this will work. But no, good little boys and girls shouldn't try it.]

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

- Since this is MAAWG, I \*knew\* that someone would ask this. :-)
- Let me ask YOU: is phishing still a problem, eh?
- More fundamentally, SPF/DKIM also cannot protect you against 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 scenario.
- 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 Sniff Typical Plain Text Email Using Wireshark

- Send a simple mail message...

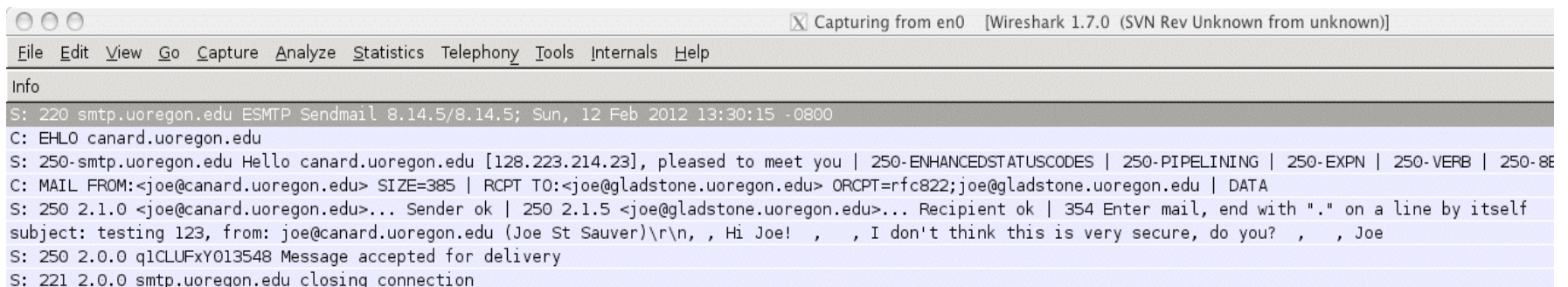
```
% mailx -s "testing 123" joe@gladstone.uoregon.edu
Hi Joe!
```

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

```
Joe
```

```
.
```

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



The image shows a Wireshark 1.7.0 interface with a packet capture from interface en0. The selected packet is an SMTP message. The packet details pane shows the following structure:

- Info: 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 | 250-8BITMIME
- 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 should 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 Msgs...

```
% 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?**

MIAGCSqGSib3DQEHA6CAMIACAQAxggNbMIIBkQIBADB5MGQxCzAJBgNVBAYTA1VTMRIwEAYD  
VQQKEwlJbnRlcm5ldDIxETAPBgNVBAsTCEluQ29tbW9uMS4wLAYDVQDEyVJbkNvbW1vb1BT  
dGFuZGFyZCBBc3N1cmFuY2UgQ2xpZW50IENBAhEAowXASR0JSE0KE5HSe8RXCTANBgkqhkiG  
9w0BAQEFAASCAQAphc3r5MLFw43h0cMzlb/UG9DEaFPyFtcaiN8koelnok2DVdcAtSb9wulU  
iKjw4jps8GwqPeonzC8o+RMyktiFwMvM/QfN4zMUbfxsJr0i7FpnveROp+V8Cyo2hDuJpa/d  
GjRI560cDnH2z4tnY009/SJBCvLIIRjfnnnuJlS12VF00kcA9sfJI23QWhauisoef0ZhvAOw  
1lwHi8o+4icSe6iT18rR+Sr9MDhulDdfVCfmYwDfBi4SAqzbLK1FZfSj7aIjphlcFV4JKXr3  
HyEz2afYRCGYUUAgk1zjcfhh4Eqkah6TwZ8QCtWUTsYdhuZdHGHw6zbBuSUYxzRG2NiRMIIB  
wgIBADCBqTCBkzELMAkGA1UEBhMCR0IxGzAZBgNVBAGTEkdyZWFOZXIgdWTFuY2hlc3RlcjEQ  
MA4GA1UEBxMHU2FsZm9yZDEaMBGGA1UEChMRQ09NTORPIENBIExpbWl0ZWQxOTA3BgNVBAMT  
MENPTU9ETyBDG1lbnQgQXV0aGVudGljYXRpb24gYW5kIFNlY3VyZSBFbWVpbCBDOQIRAKgC  
OyLlImfFLiBB1WracUfMwDQYJKoZIhvcNAQEBBQAEggEA0c1JpNLx+62m1To69oxFd3/fMEvo  
UDkL1nSQe5LDhKnH3DXmH2vvTN0Q0h8vjGbkCgk1CD11164VRi380QrtVYTsYC19tB1kuHam  
SH+xJIIIsLkNasYWnCXwzji+Uw80GiAP9/CgB/aYJhhYJt1HRQ+43S9m3xgpdK//aCOIjmKLl  
prFiQ1Jk5Wx3Sqm/Kkg89m9ulln1ckpIBrvTxNsikZmFwh4QGcCtz42+mTGZXcbrn9yftOF  
4ds9xDbBm5e/Se/aq4vpfX0yi0/UP8/ywJ5+zG2ufyJw4i2h203vyD6WzX7PiYuzsn232RkR

[That 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) <sup>†</sup>	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) <sup>†</sup>	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.exim.org/exim-html-3.20/doc/html/spec_38.html)

[http://www.postfix.org/TLS\\_README.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, but today we're going to talk about S/MIME with client certificates, instead.
- Before we can dig in, however, we need a little "crypto 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 MAAWG social event, 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."

# 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"). 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.

Totally evil people may have properly 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 hierarchically ("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 (e.g., 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!) will stop trusting them and then they'll quickly be totally out of business (literally).



# 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 27<sup>th</sup>, 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](http://www.dhs.gov/xlibrary/assets/Partnership_Program_Benefits_Tax_Payers_Public_and_Private_Sector.pdf) )
- That is **\*NOT\*** a toy-scale cert project by any means!

# CURRENT STATUS – HSPD-12

---

- *HSPD-12 Credentials Issued as of June 1, 2011:*  
Credentials issued to Employees: **4,151,358 (88%)**  
Credentials issued to Contractors: **842,946 (81%)**  
(Total credentials issued: 4,994,304 (87%))
- *Background Investigations Verified/Completed as of June 1, 2011:*  
Background investigations completed for Employees: **4,128,415 (87%)**  
Background investigations completed for Contractors: **886,137 (85%)**  
(Total investigations verified/completed: 5,014,552 (86%))
- 18 federal credential issuance infrastructures are in operation nationwide
- 59 system integrators and 592 products on GSA Approved Products and Services List

Agency specific status may be located at:  
[http://www.whitehouse.gov/omb/e-gov/hspd12\\_reports/](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](http://www.idmanagement.gov/presentations/HSPD12_Current_Status.pdf)

# **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 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, see

<http://www.nps.edu/Technology/Security/CAC-guide.pdf>

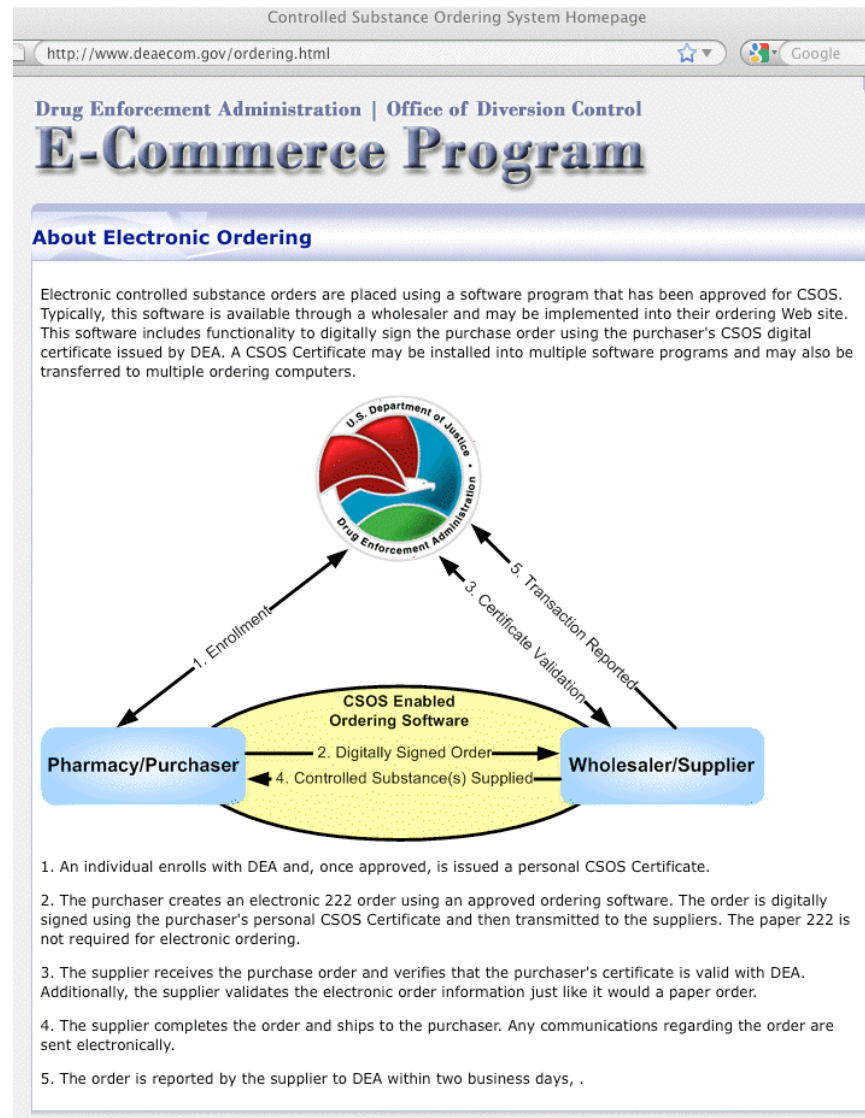
With the help of that guide, I think most folks would be able to figure out how to do basic CAC/PIV tasks.

# **Why Are The Feds Using Client Certs? If You Need "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](https://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."

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



## **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.

# What If A User (or CA) Needs To Revoke A 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") were meant to handle this problem, much like those printed books of stolen or revoked credit card numbers that every merchant used to get from the bank card companies in the old days. Most CAs currently publish a CRL once a day. Some users may 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 revoke a compromised credential, nor do you really want millions of user to each have to potentially download a huge file listing piles of revoked certificates!
- OCSP ("online certificate status protocol") was meant to handle this issue much more directly, and interactively, but many browsers and email clients don't bother checking a cert's OCSP status. Ugh.



# OK, That's Enough Background – Let's Get Started

- We could talk for hours when it comes to providing crypto background, but let's just dive right in and see how this all practically fits together.
- The next part of our agenda 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


## **II. 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    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

## Free Secure Email Certificate

 [Print View](#)

Price: 100% Free

**FREE**  
DOWNLOAD

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

✓ Integrates easily with Microsoft® Outlook®

✓ Digitally sign to ensure data integrity

✓ Also sign Microsoft® Word® documents

✓ Compatible with all major email clients

✓ Download and install within minutes


✓ Protection against identity and data theft

✓ 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**  


36

# The Application Form You'll Complete



## Application for Secure Email Certificate

### Your Details

First Name	<input type="text"/>
Last Name	<input type="text"/>
Email Address	<input type="text"/>
Country	<input type="text" value="United States"/>

### Private Key Options

Key Size (bits):	<input type="text" value="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	<input type="text"/>
Re-enter Revocation Password	<input type="text"/>
Comodo Newsletter	<input checked="" type="checkbox"/> 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...



COMODO  
Creating Trust Online®



Certification Authorities  
Certification Authorities

ENISA YOUNG LLP

## 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](mailto: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

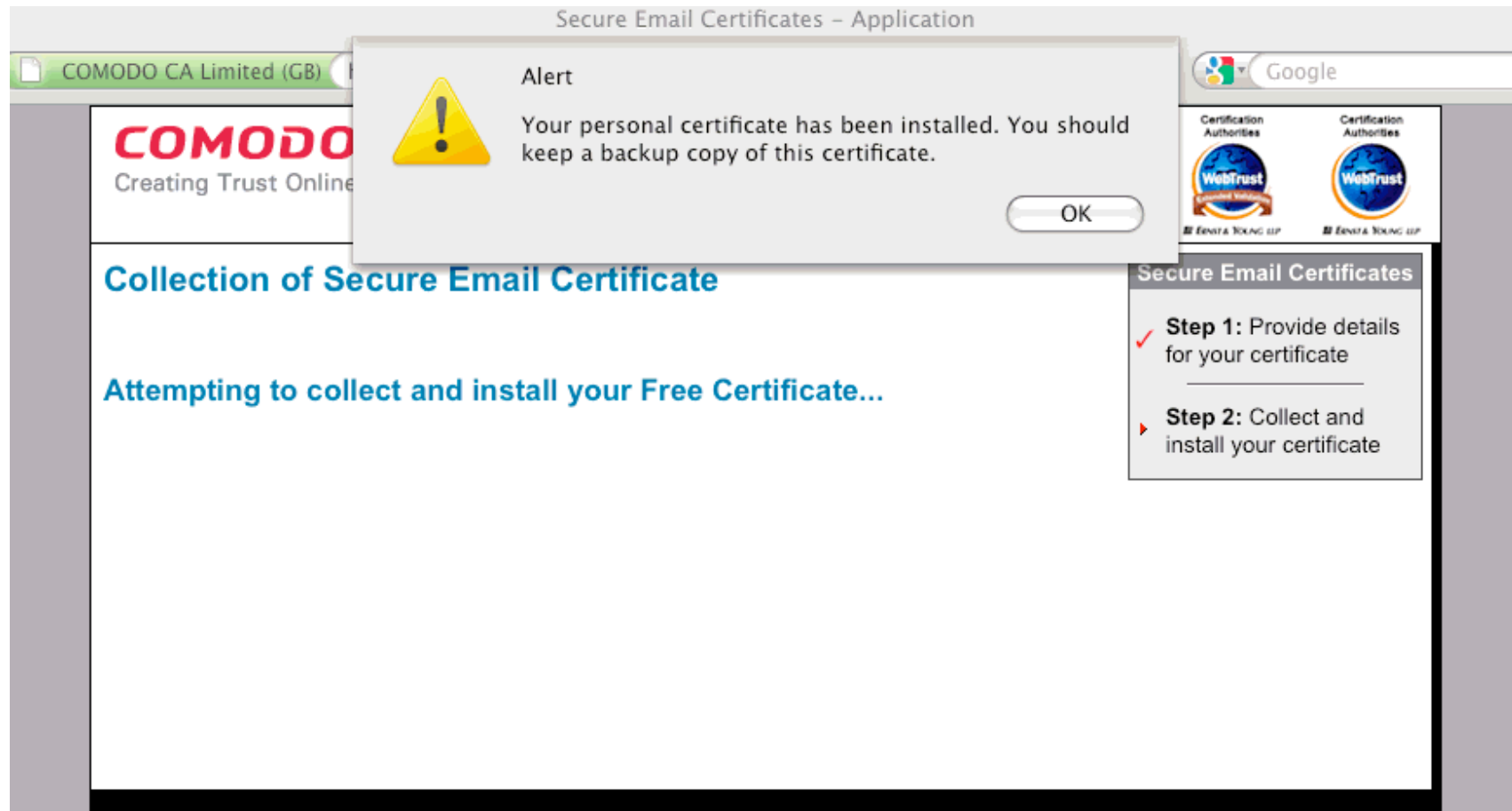
Submit & Continue

### 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...





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

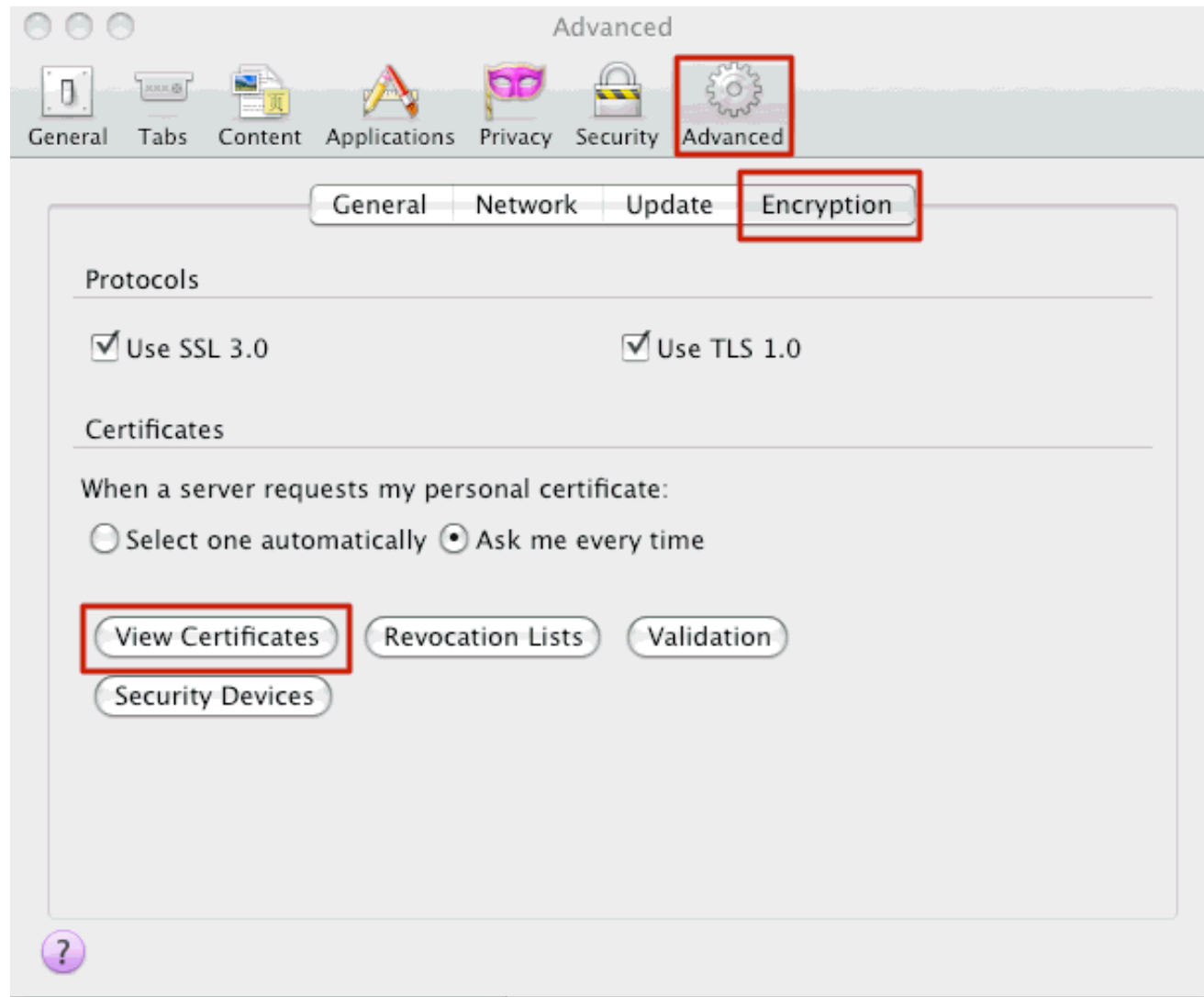
- While we're only going to show use of the free one year Comodo client cert for personal use in this training, you can also get a paid client cert from Comodo's "EnterpriseSSL" division, and free or paid client certs from other vendors. See, for example:
  - <http://www.enterprisessl.com/ssl-certificate-products/addsupport/secure-email-certificates.html>
  - <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](http://www.trustcenter.de/en/products/tc_personal_id.htm)

### **III. Examining and Backing Up Your New Client Certificate**

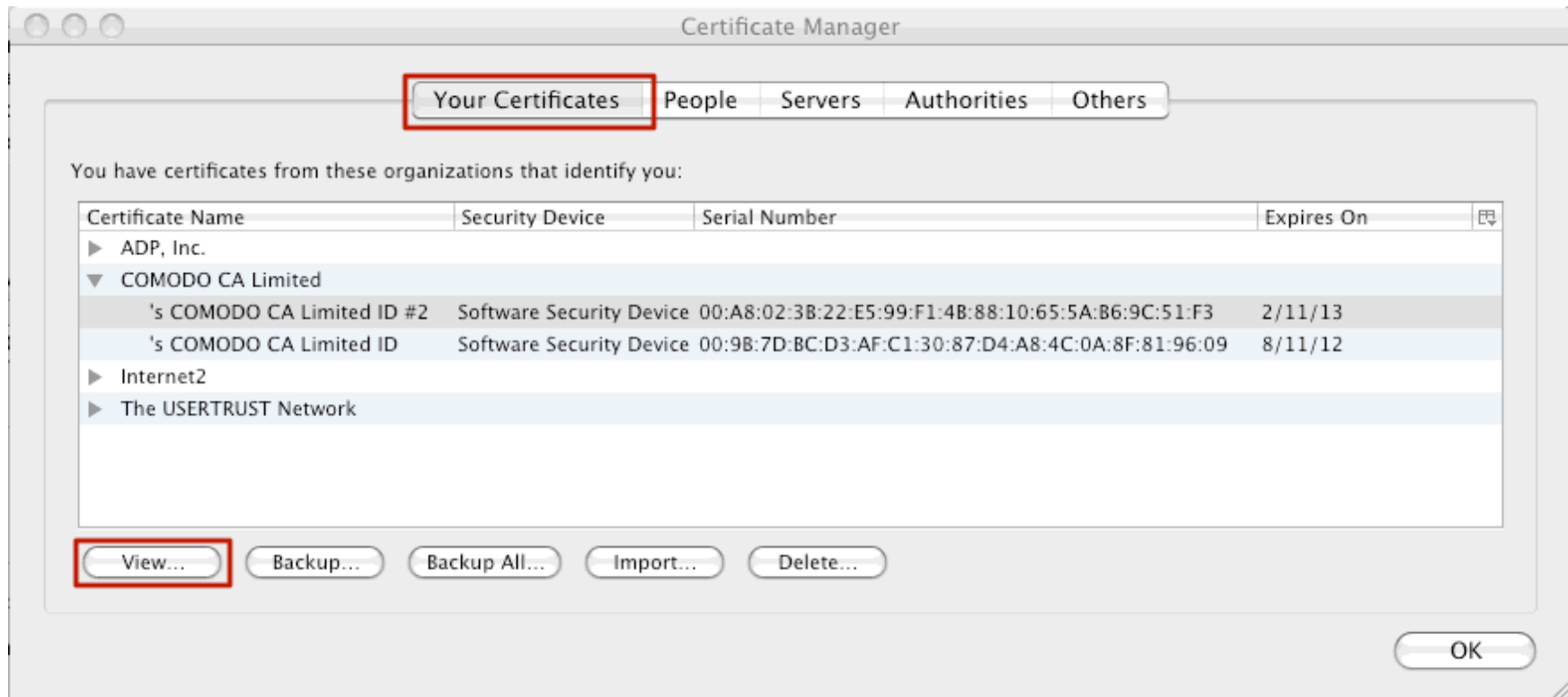
# **"Okay, I've Got My Client Cert. What Do I Do Now?"**

- 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...

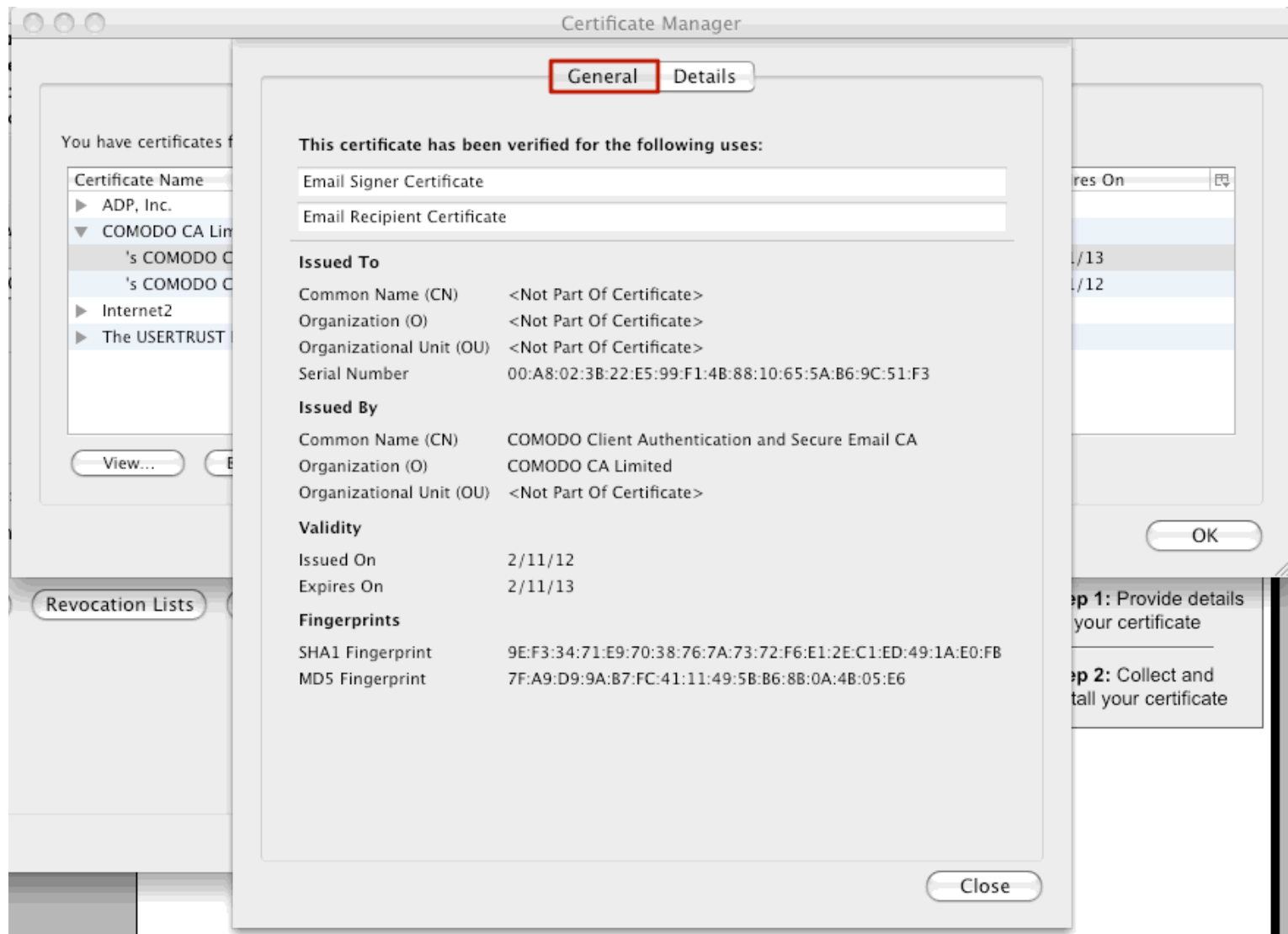


# The Firefox Certificate Manager

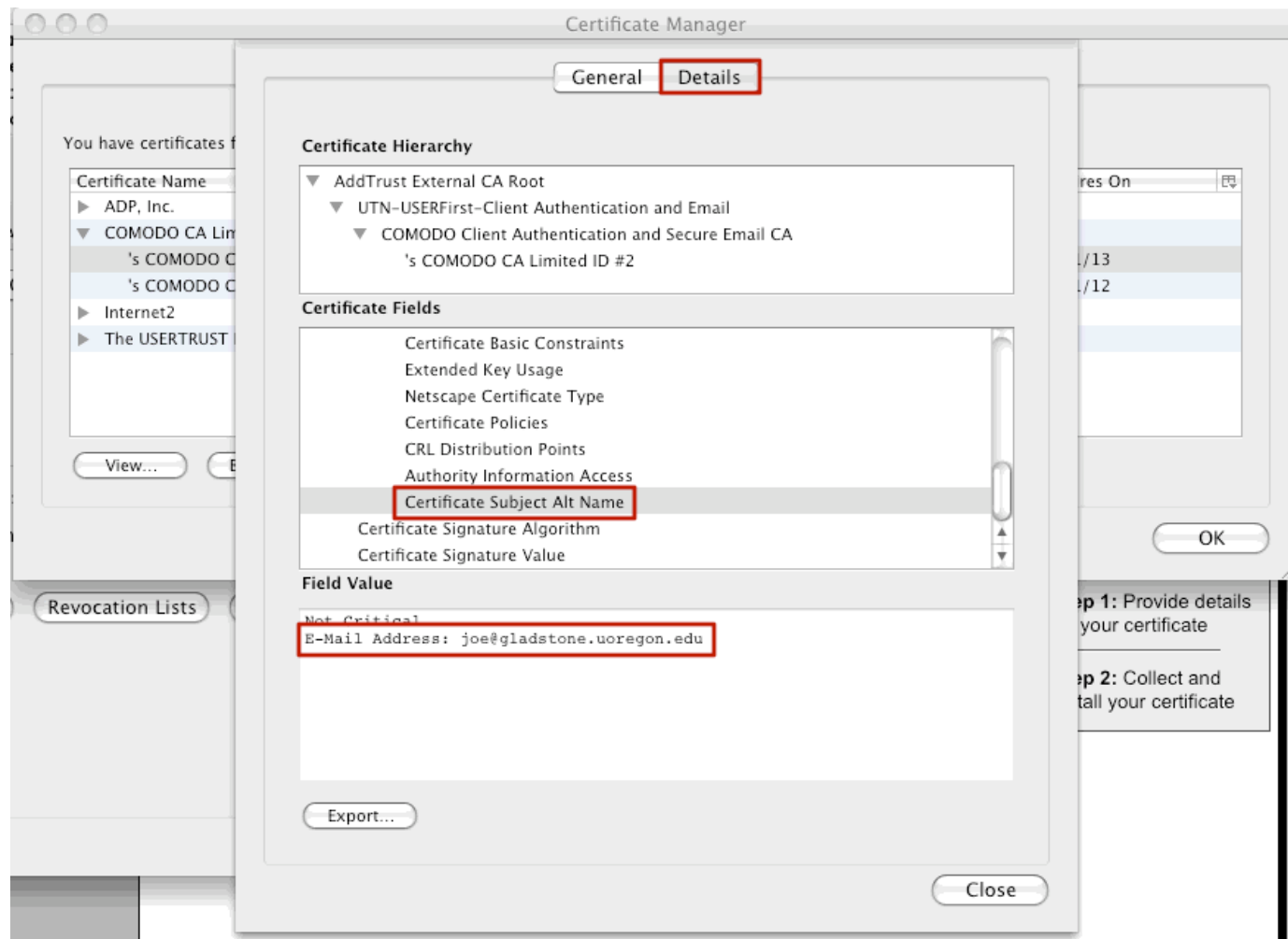


**Notes:** Select the “Your Certificates” tab on the Certificate Manager panel. If necessary, hit the triangular arrow to expand the list of Comodo certificates. You’ll probably only see one certificate, the one you just got from Comodo. But just as a matter of form, let’s confirm that it really is yours...

# The General Tab Tells Us When The Cert Expires

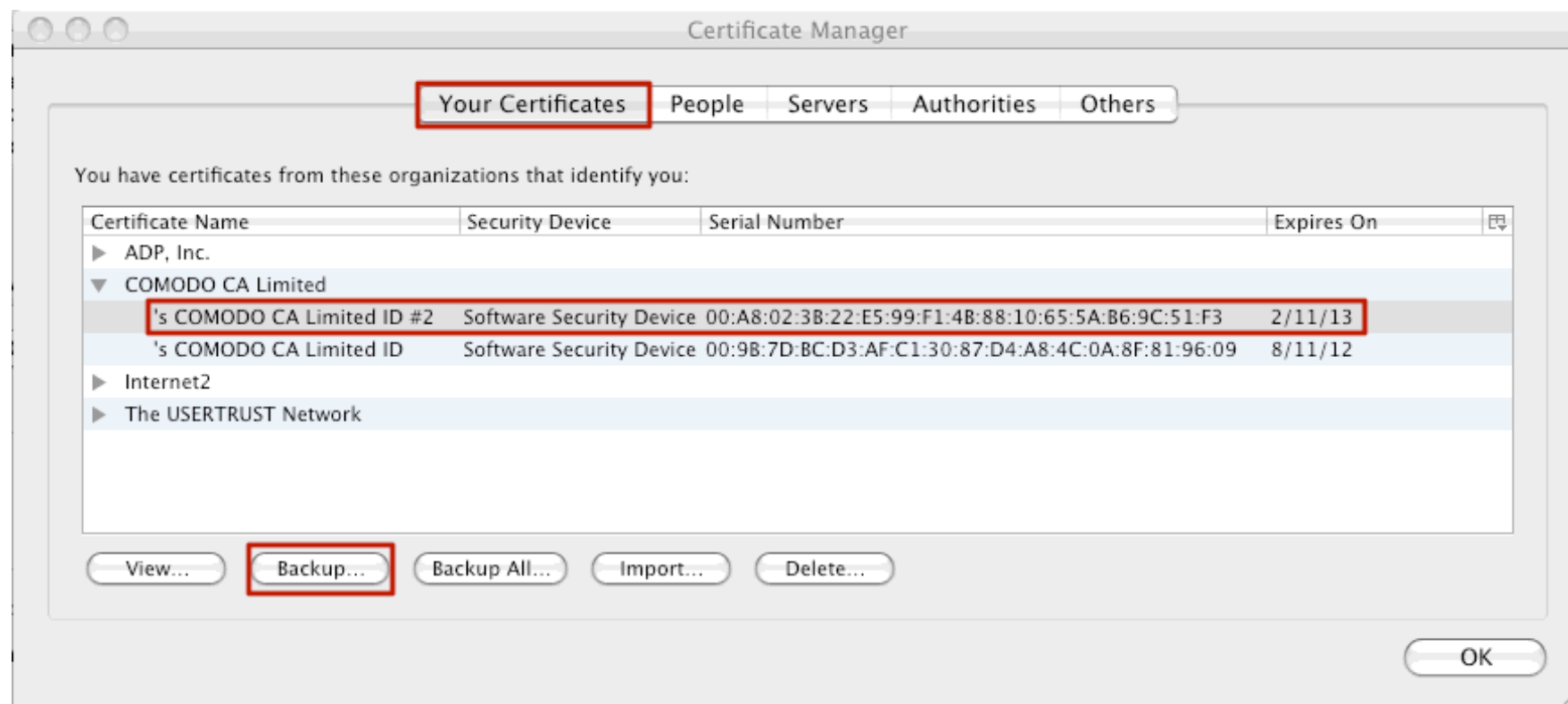


# The Details “View Cert” Tab Will Let Us See The Email Address Associated With Our New Cert



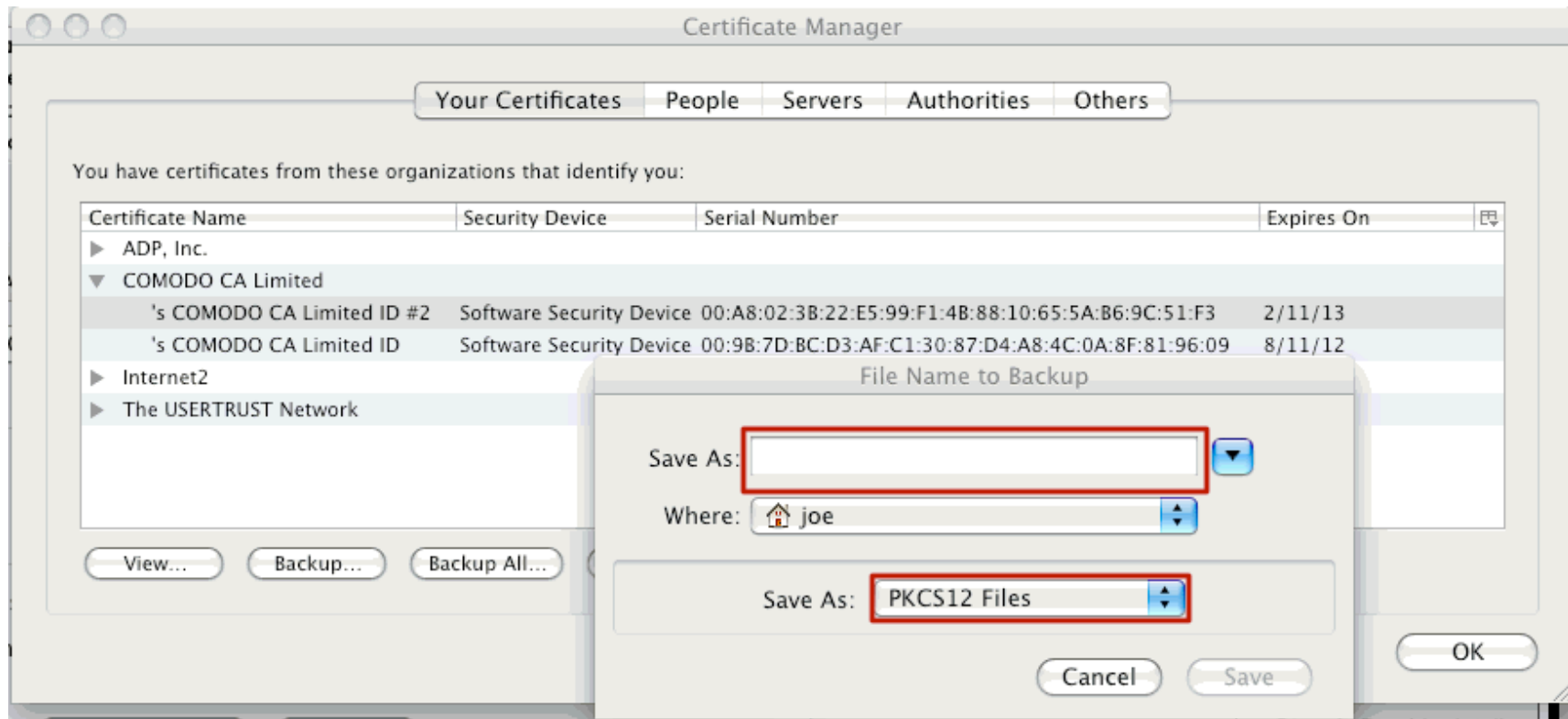
[Close the “View Certificate” box when you’re done looking at it]

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





# The “Name Your Backup” Dialog Box



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.

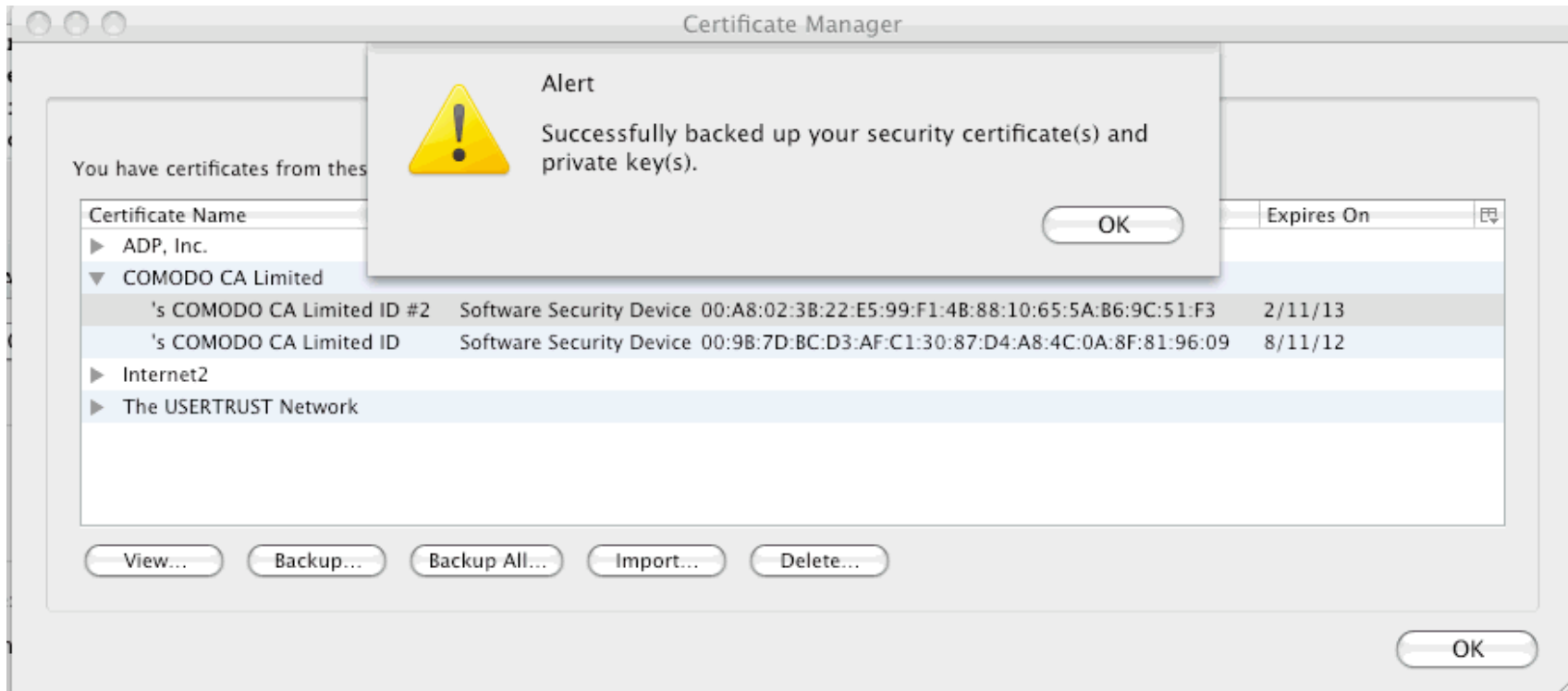
# The Cert Manager Backup-Password Dialog Box



Pick a strong password to secure your cert backup file.

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

# Backup Successful...



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.

## **IV. Importing Your Certificate Into Thunderbird**

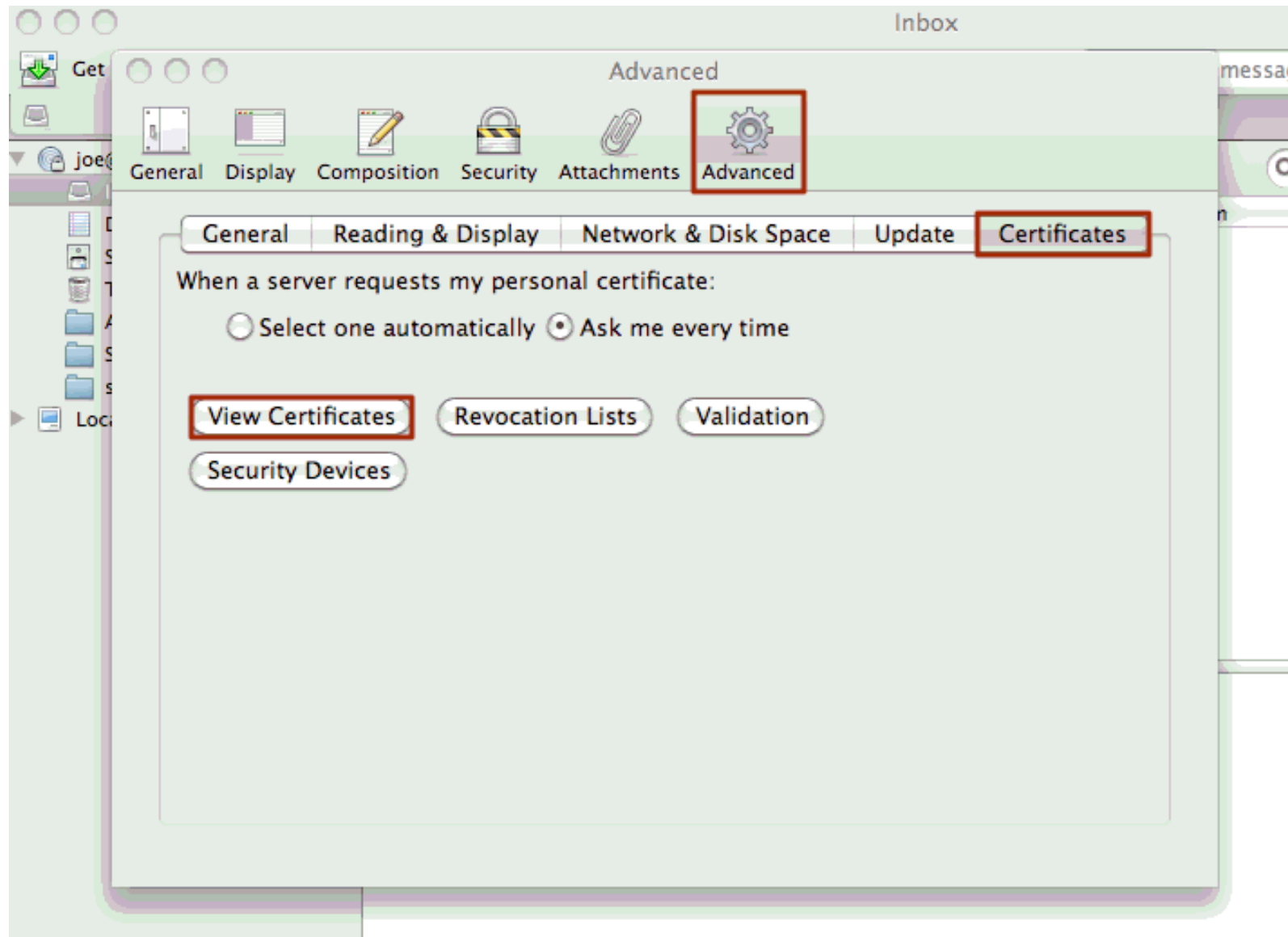
# We're Now Going To Import Our New Certificate Into Thunderbird

- 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.  
**One caution to any non-technical person looking at these slides: 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!**

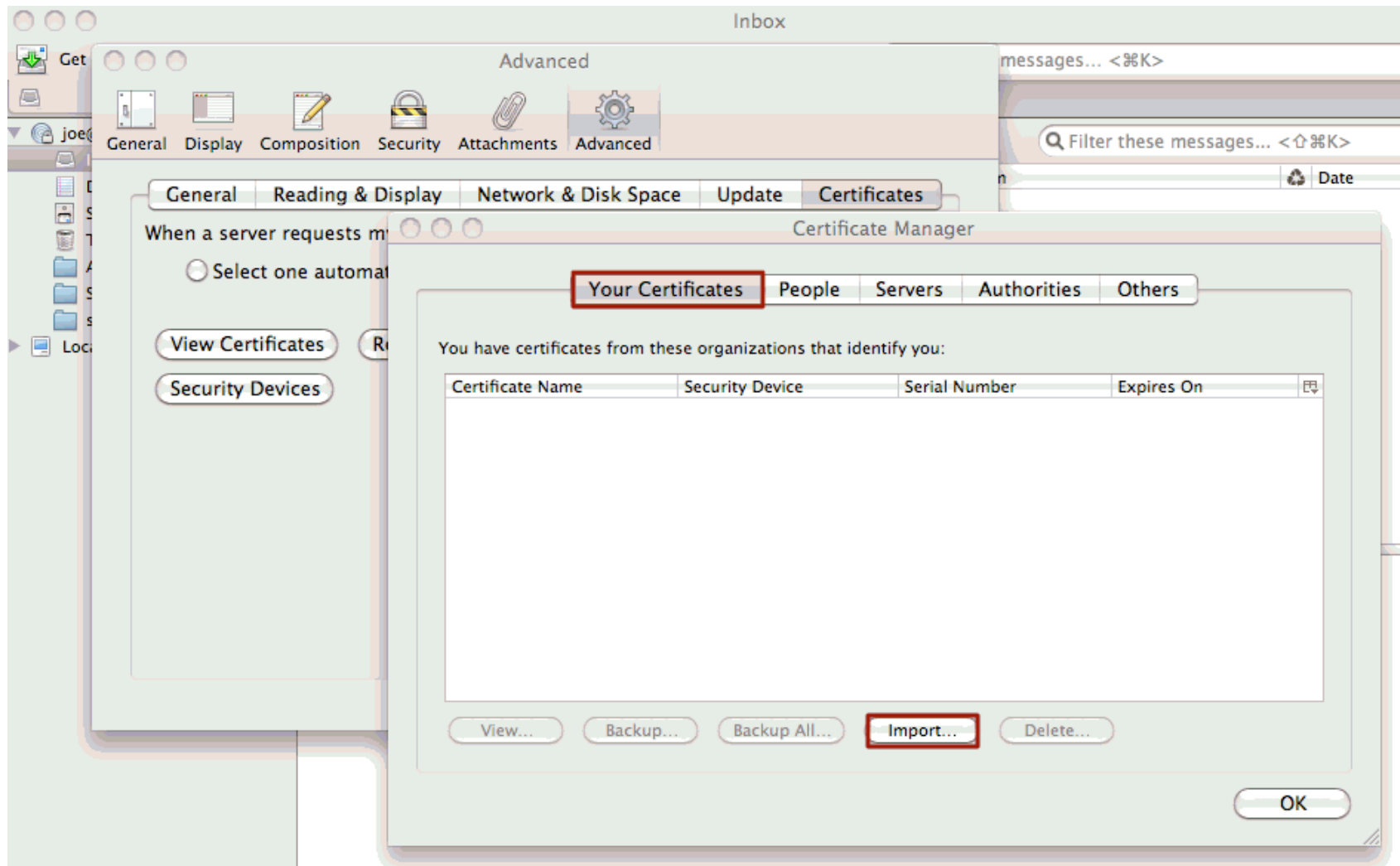
# **“Why Can’t Thunderbird Just Use The Cert That I’ve Already Got Installed in Firefox? They're Both Mozilla Applications, Aren't They?”**

- 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...

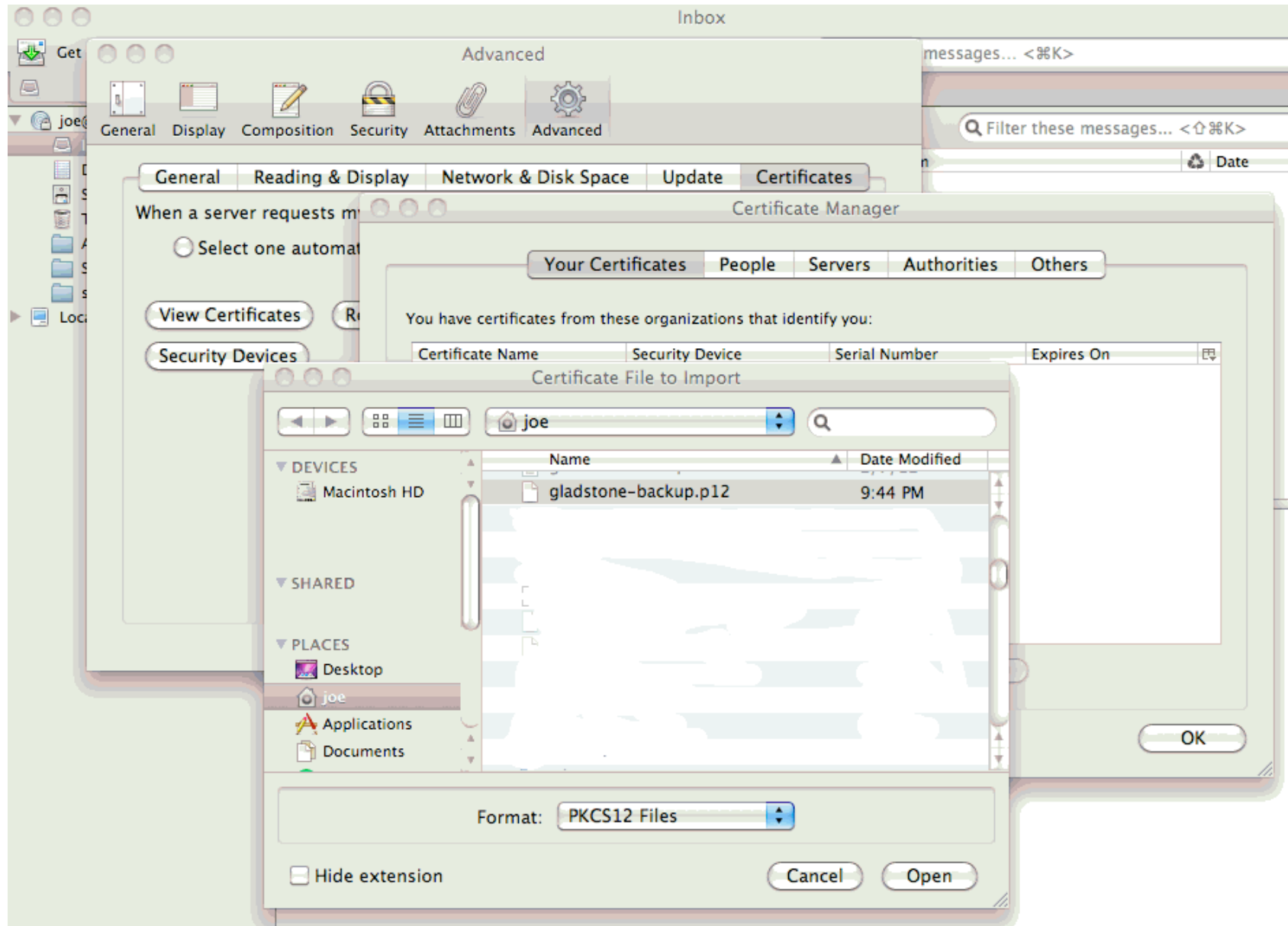


# In The Certificate Manager, “Your Certificates” Tab, Click on Import

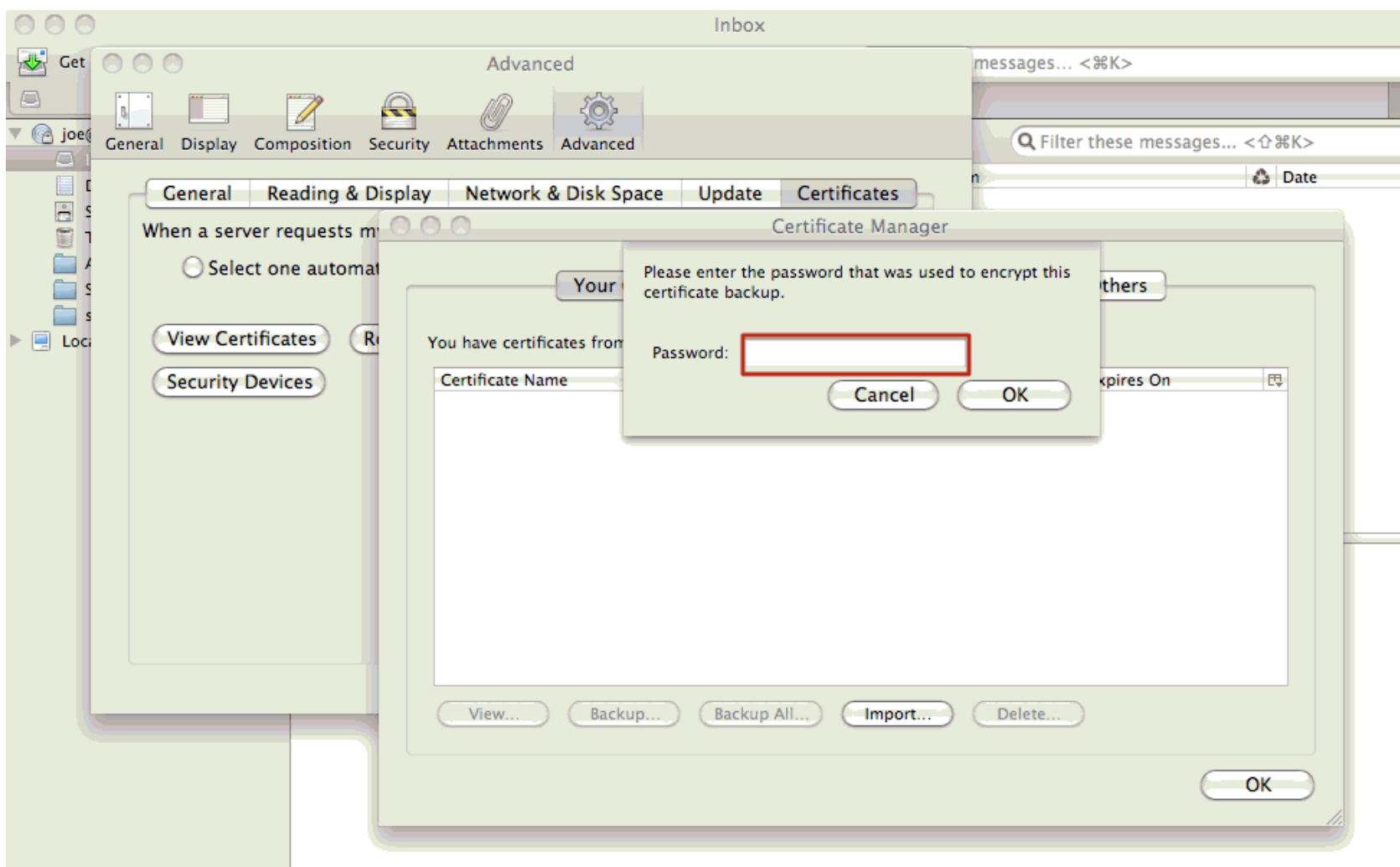




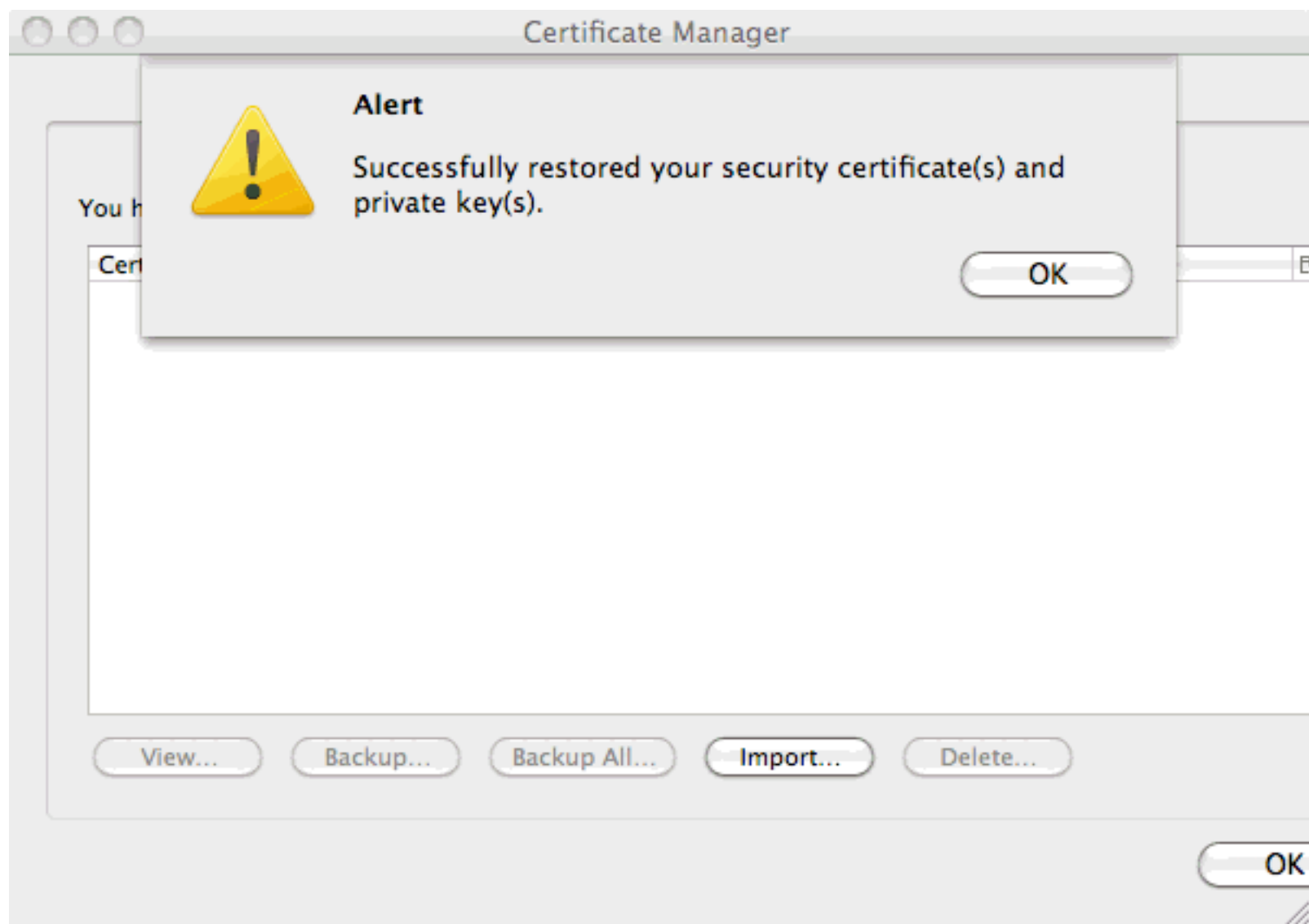
# Select The .p12 Backup File You Want To Import



# Supply the Password You Used for The Cert Backup

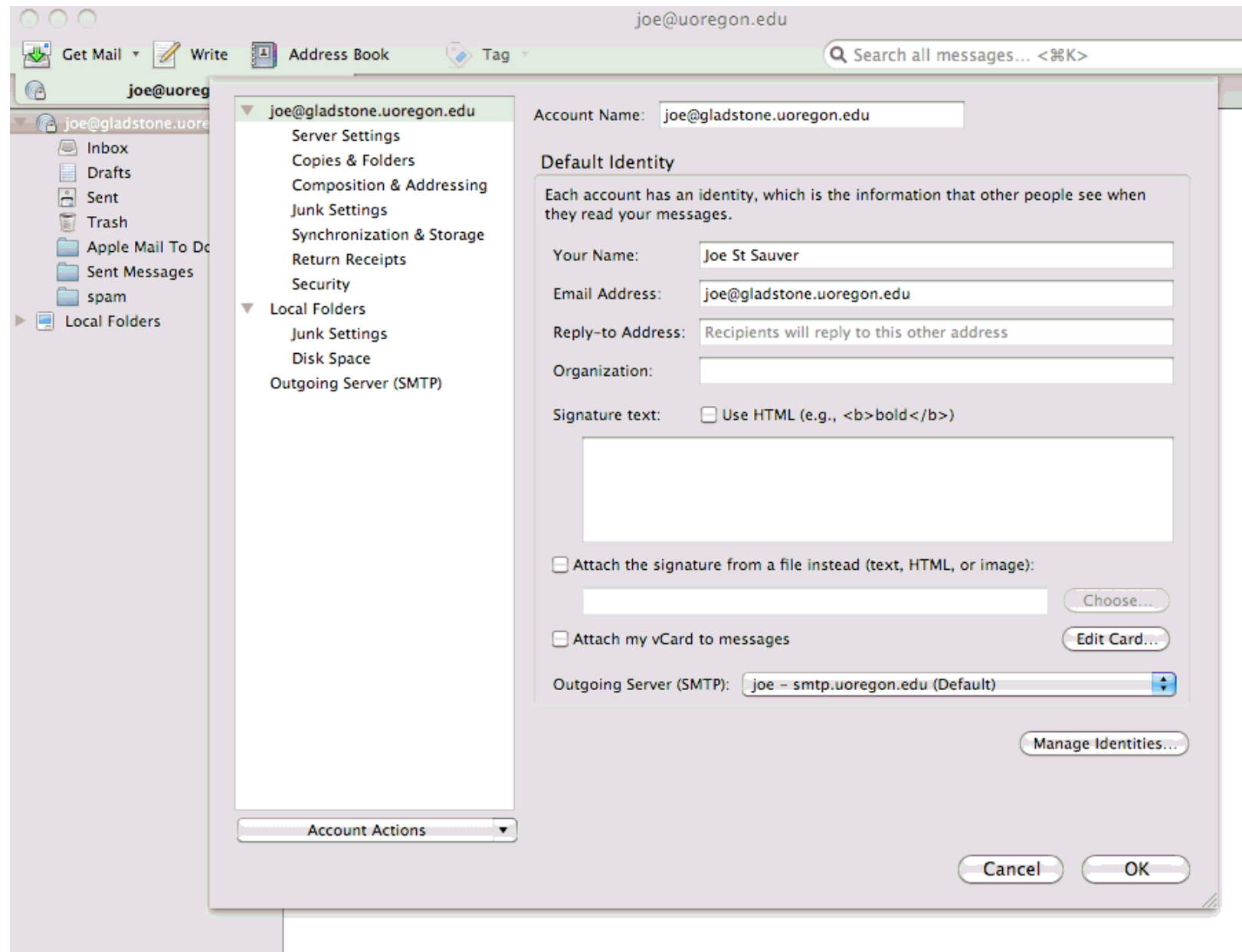


# Successful Importation of The Cert Into Thunderbird

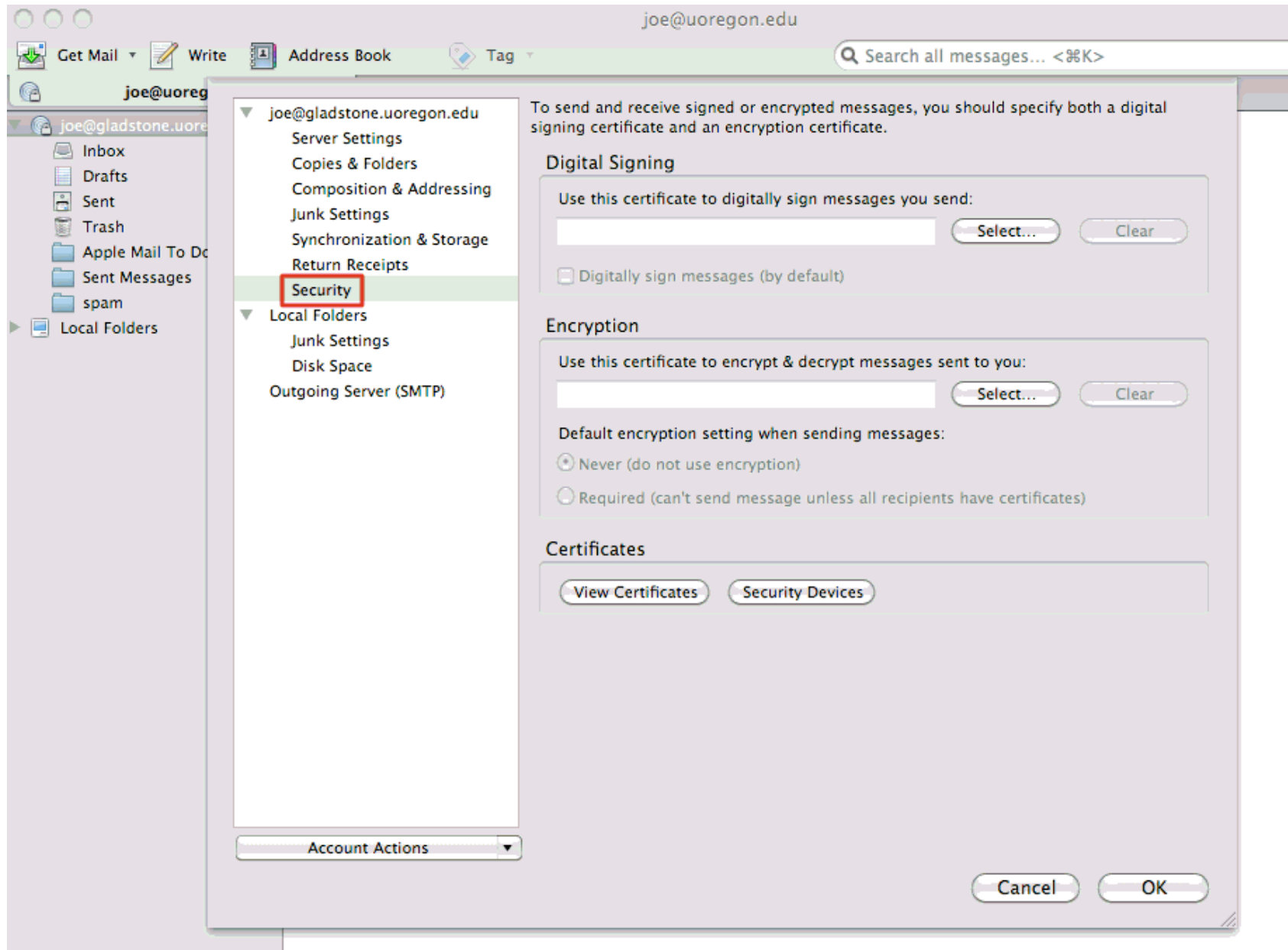


## **V. In Thunderbird, Associate Your Certificate With Your Email Account And Configure Thunderbird To Do Digital Signing**

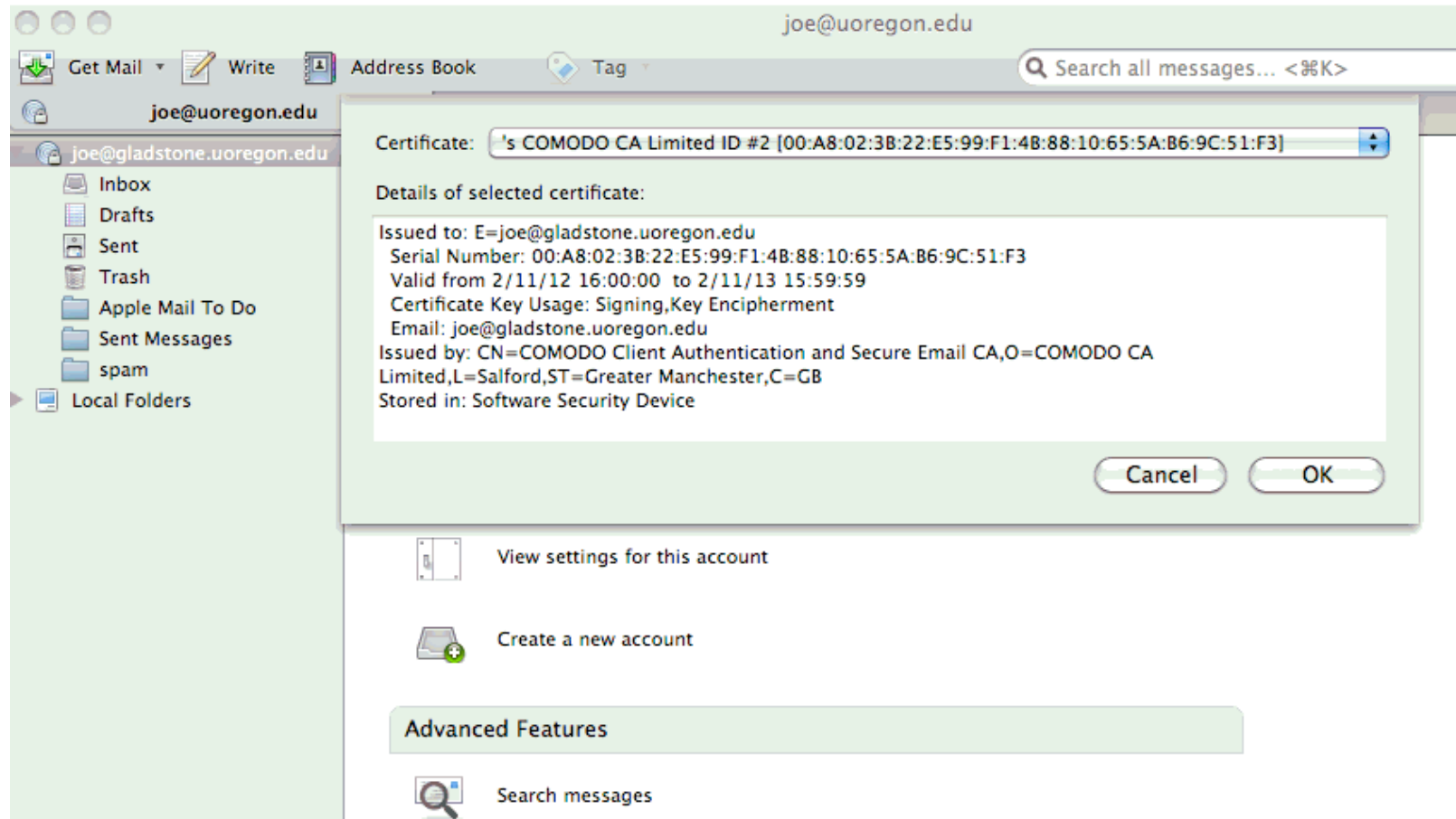
# Thunderbird: Tools --> Account Settings



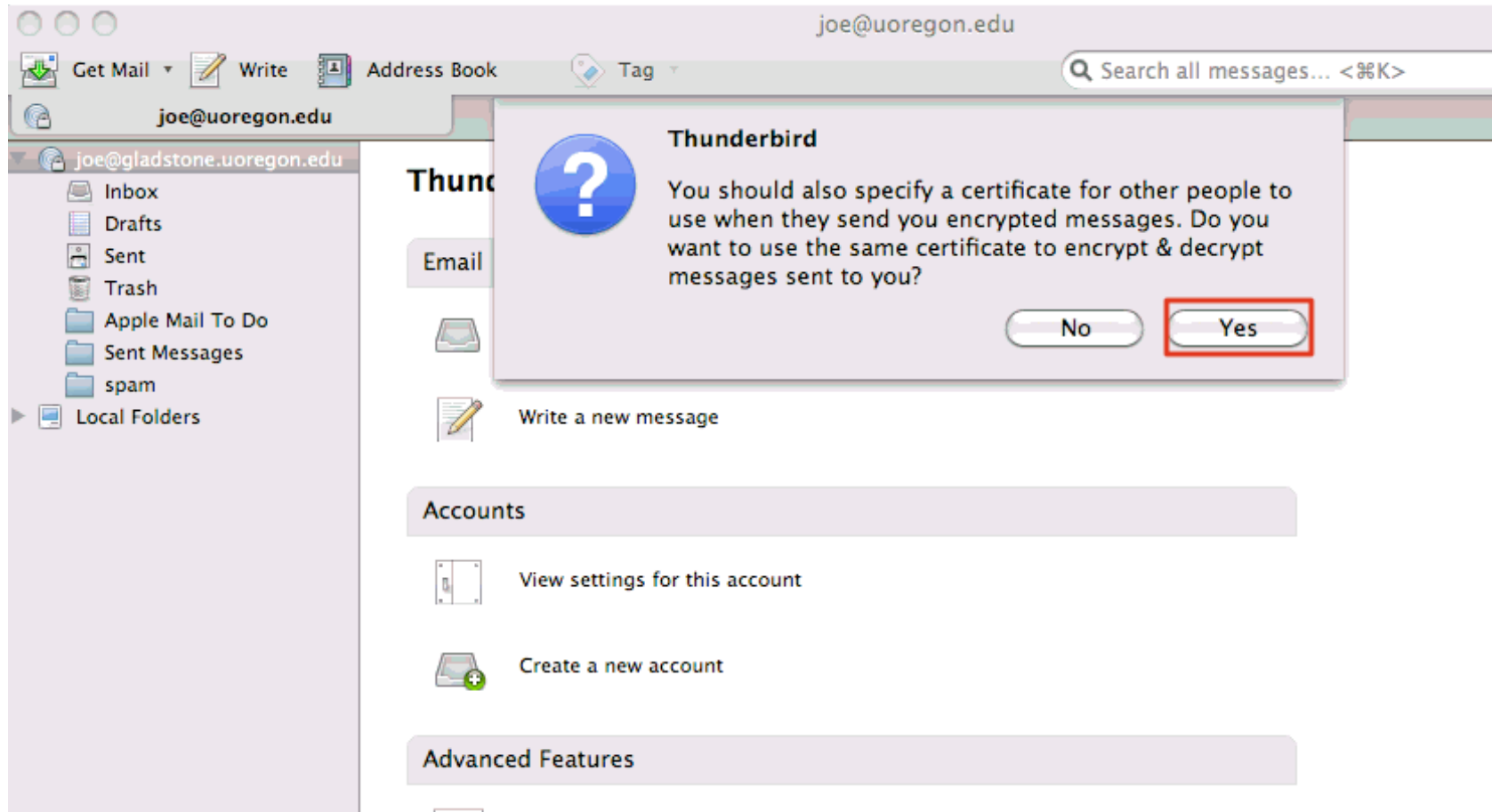
# Security



# Select The Cert You Want To Use For Digital Signing

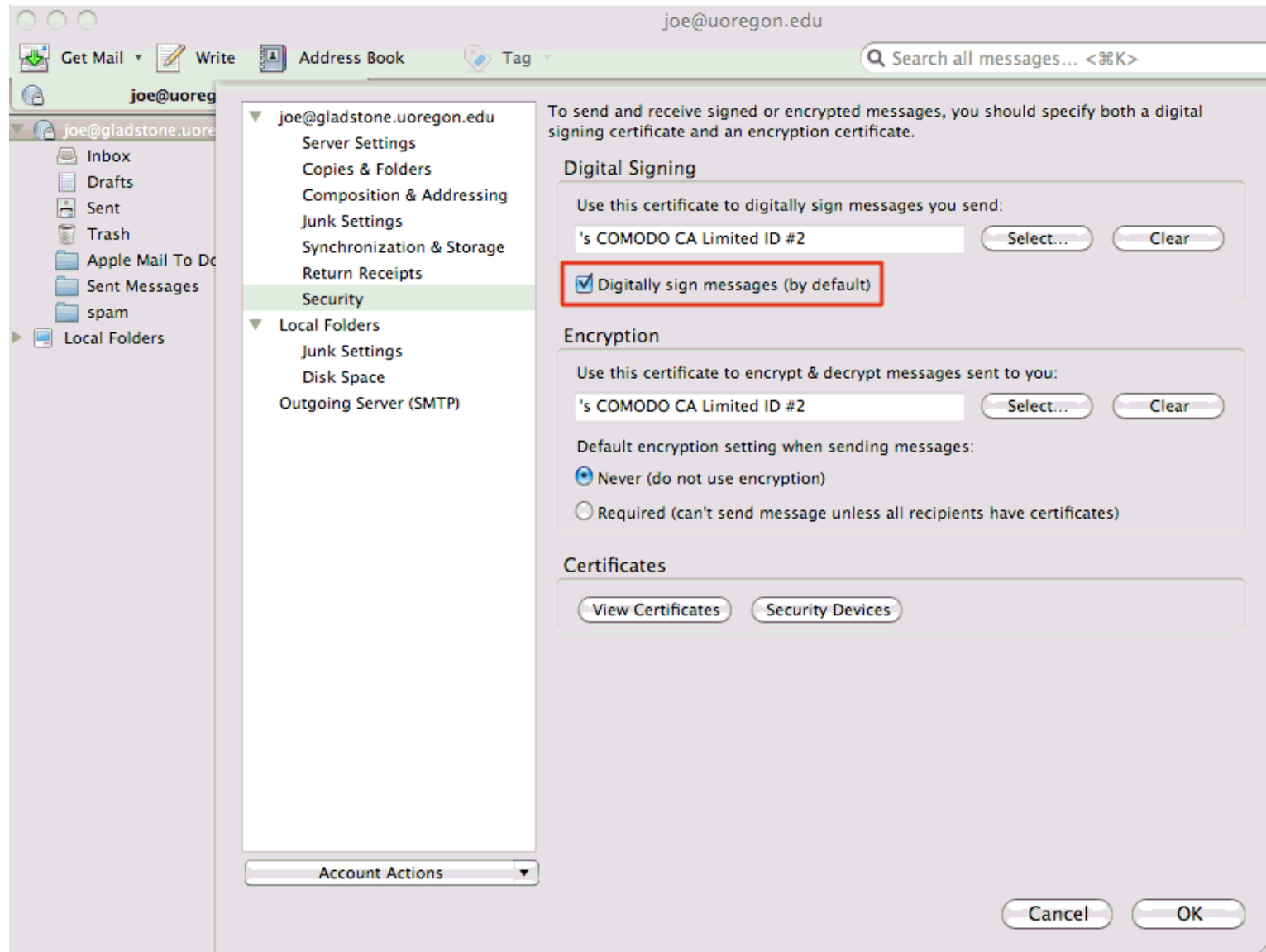


# Confirm That You Want To Also Use That Same Cert for Encrypting/Decrypting Messages





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

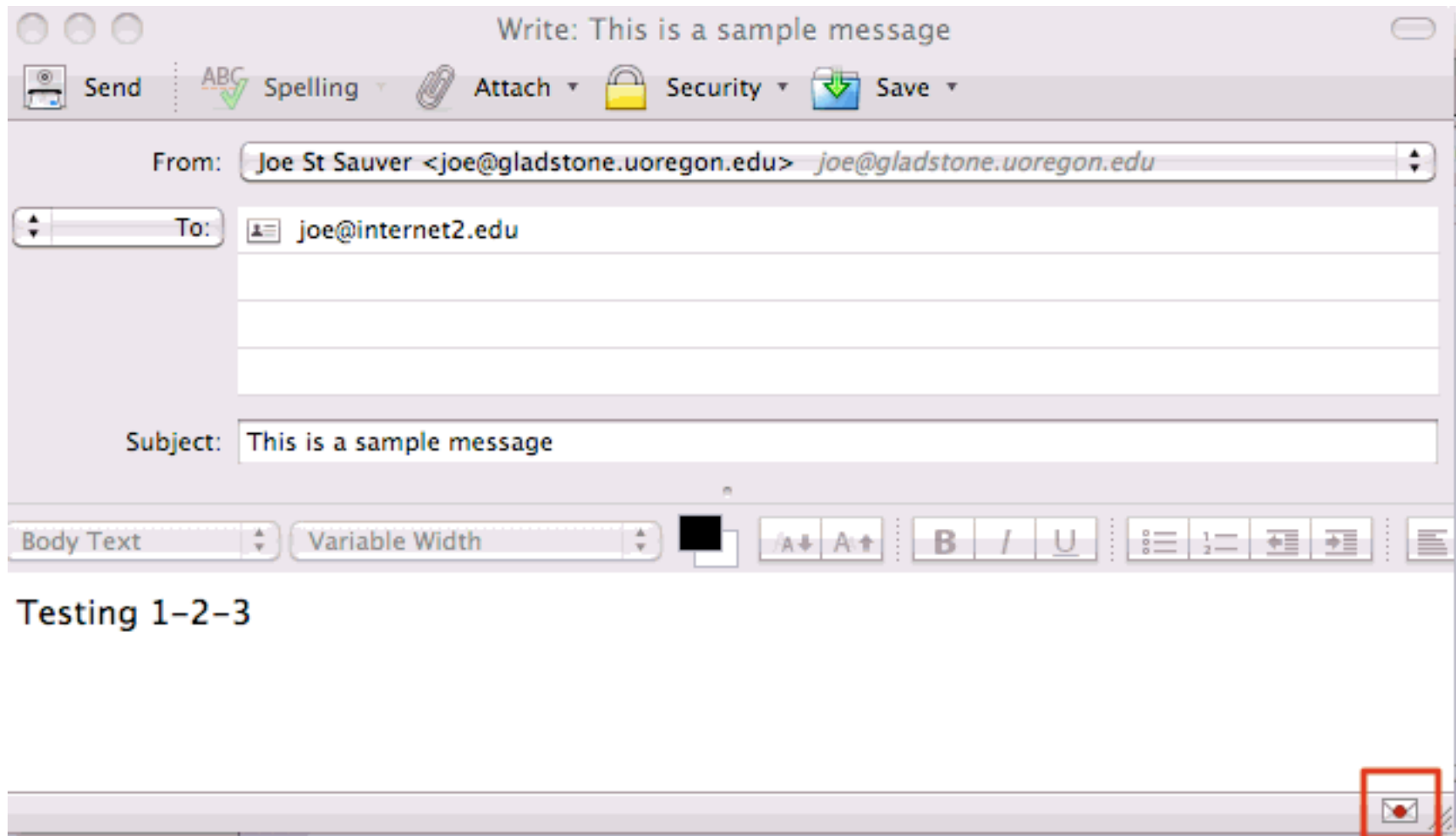


# Thunderbird Configuration Is Now Complete...

- 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!

## **VI. Digitally Signing A Message In Thunderbird**

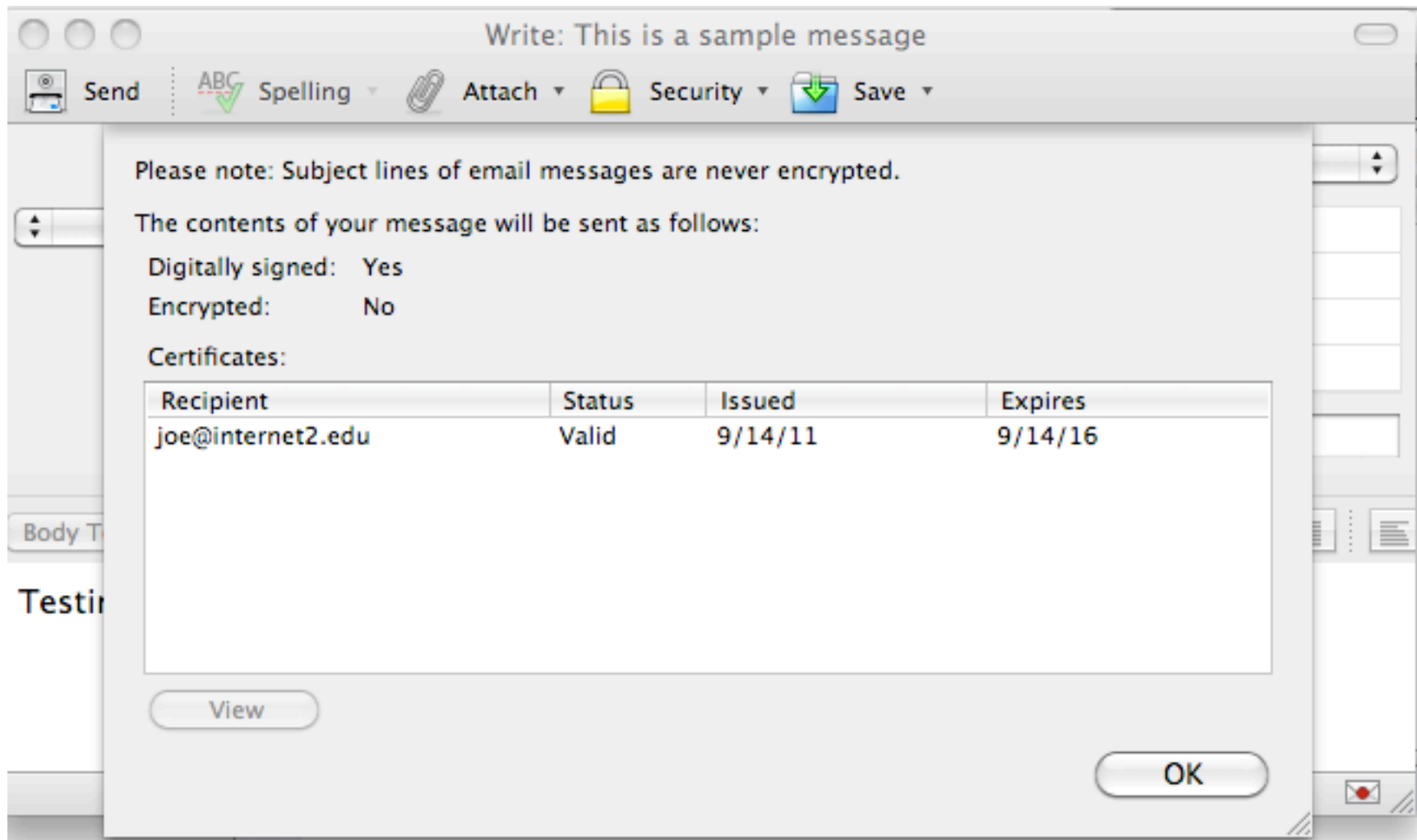
# Start Writing A Message The Way You Normally Would



***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!*



## 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.
- 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.

## **VII. 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 his 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.

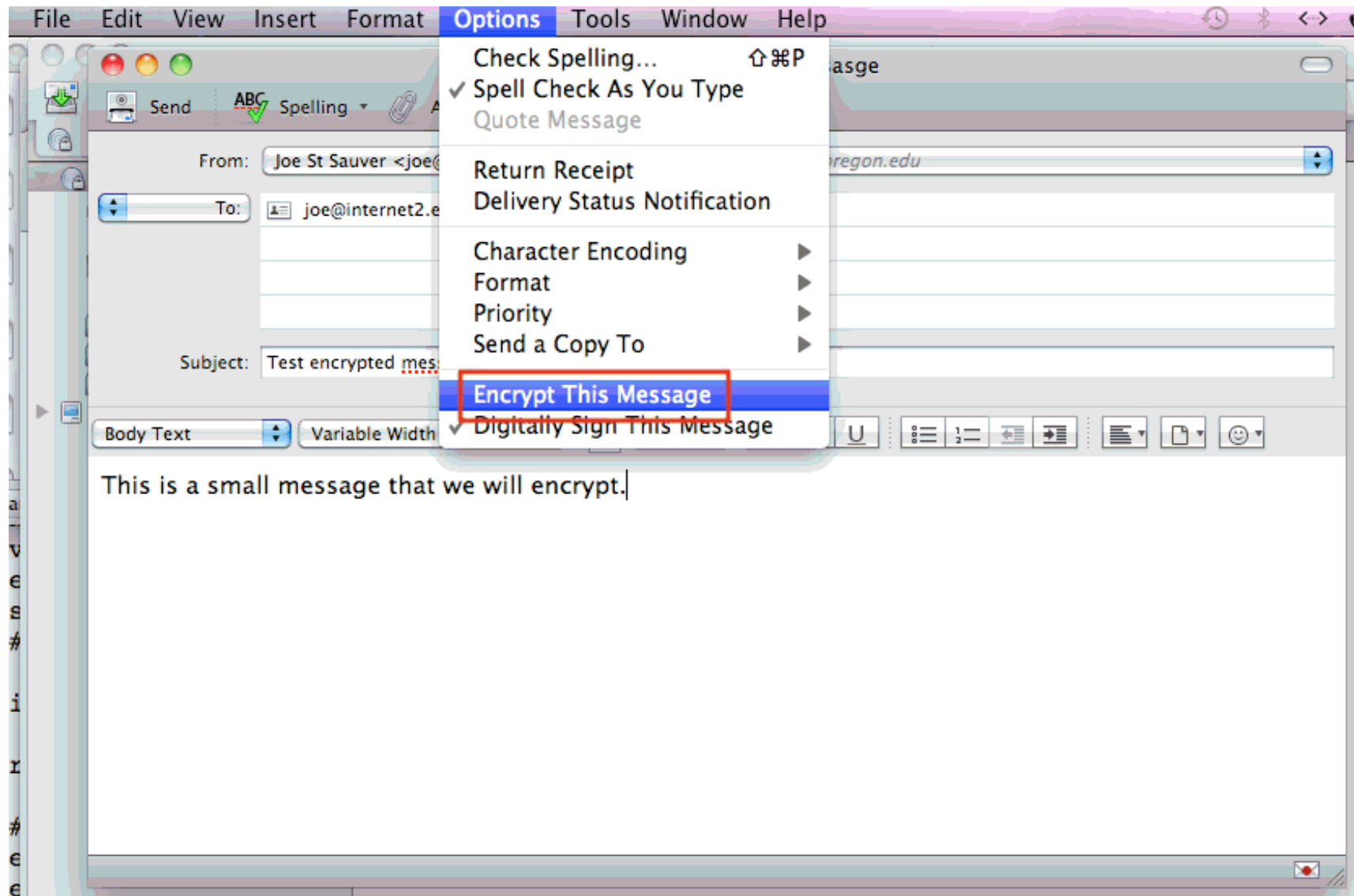
# A Meta Question: *Should* I Encrypt The Mail I Send?

- Maybe yes, maybe no.
- First of all, note that you 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 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?

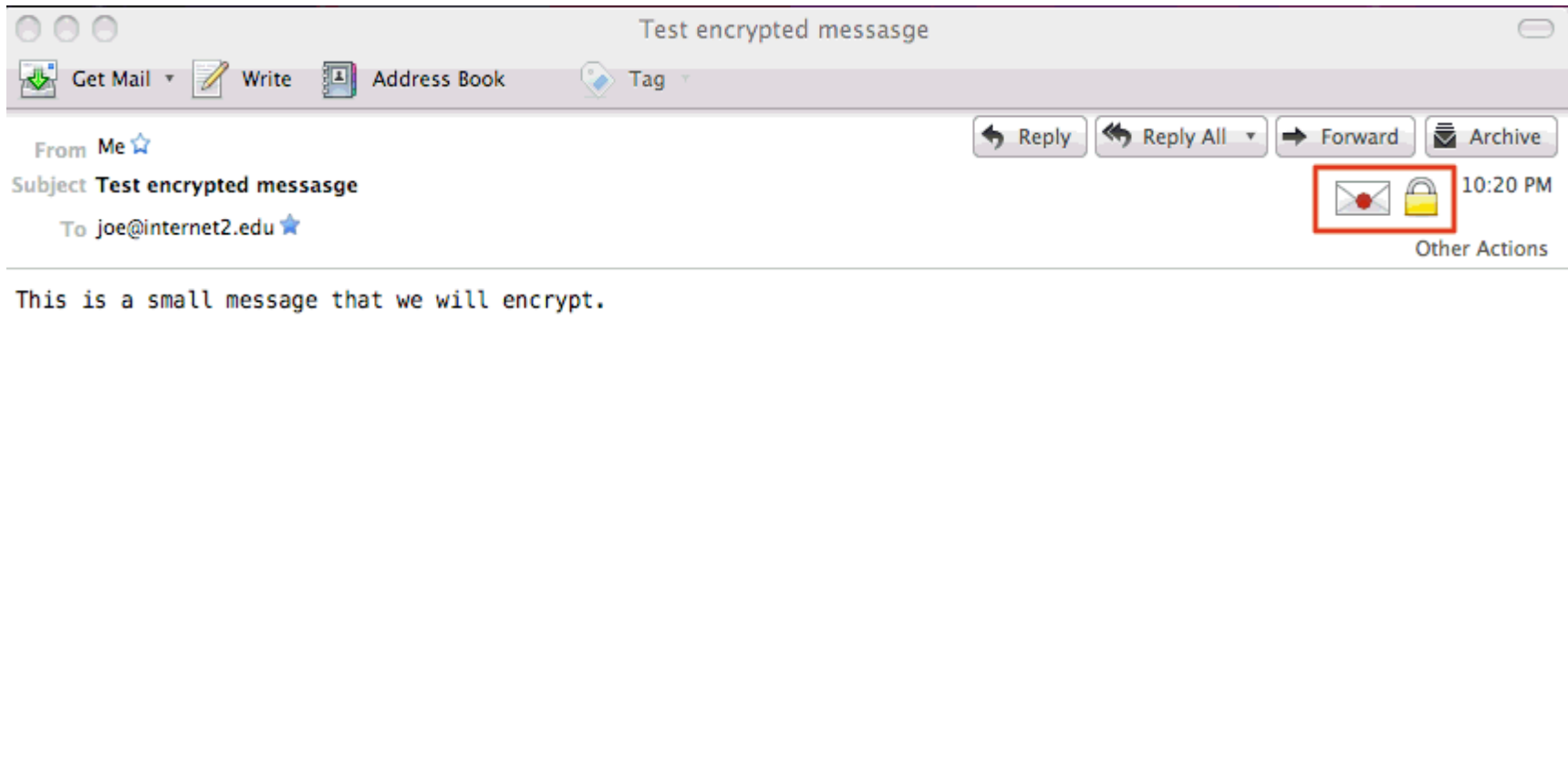
# Hedging The Risk of Data Loss: Key Escrow

- Let's pretend that you have a person who's doing absolutely critical (and highly sensitive) work for you or your company, 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 employee's going to die or be killed, or maybe just go to work for someone else (giving you "the finger" on the way out). 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 will typically have the associated cert revoked at the same time.

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



# “When I Get A Signed and Encrypted Message, What Will It Look Like?”



# Who Signed That Message? (Note: It May *Not* Be The Person Who Sent The Message)



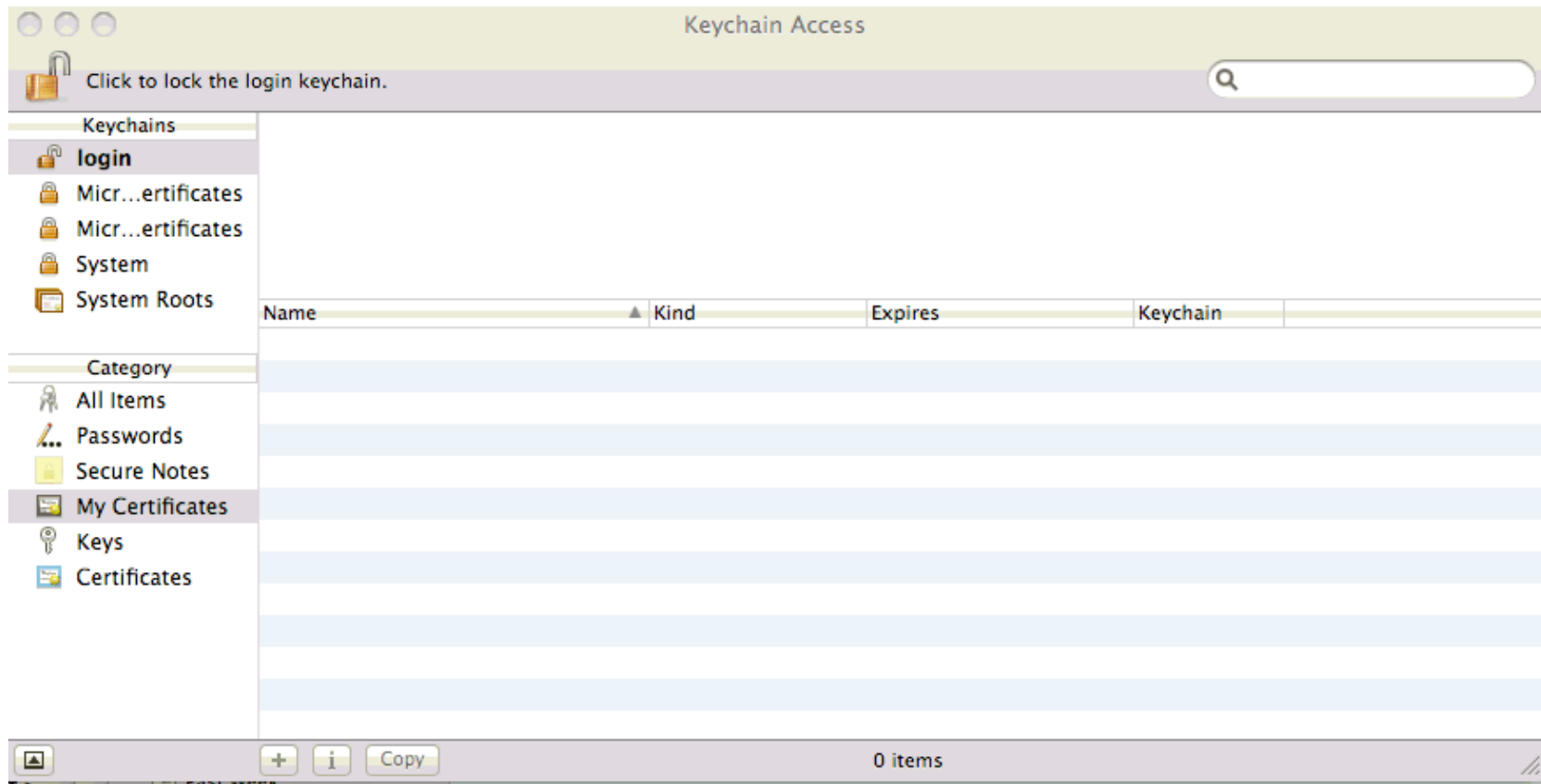
## Additional Important S/MIME Caveats

- 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 strip attachments (including p7s digital signatures). If that happens, your signature won't validate. If you send messages to mailing lists, you may want to manually disable digital signing for messages to those lists.

## **VIII. What If I Want To Use Outlook Instead of Thunderbird?**

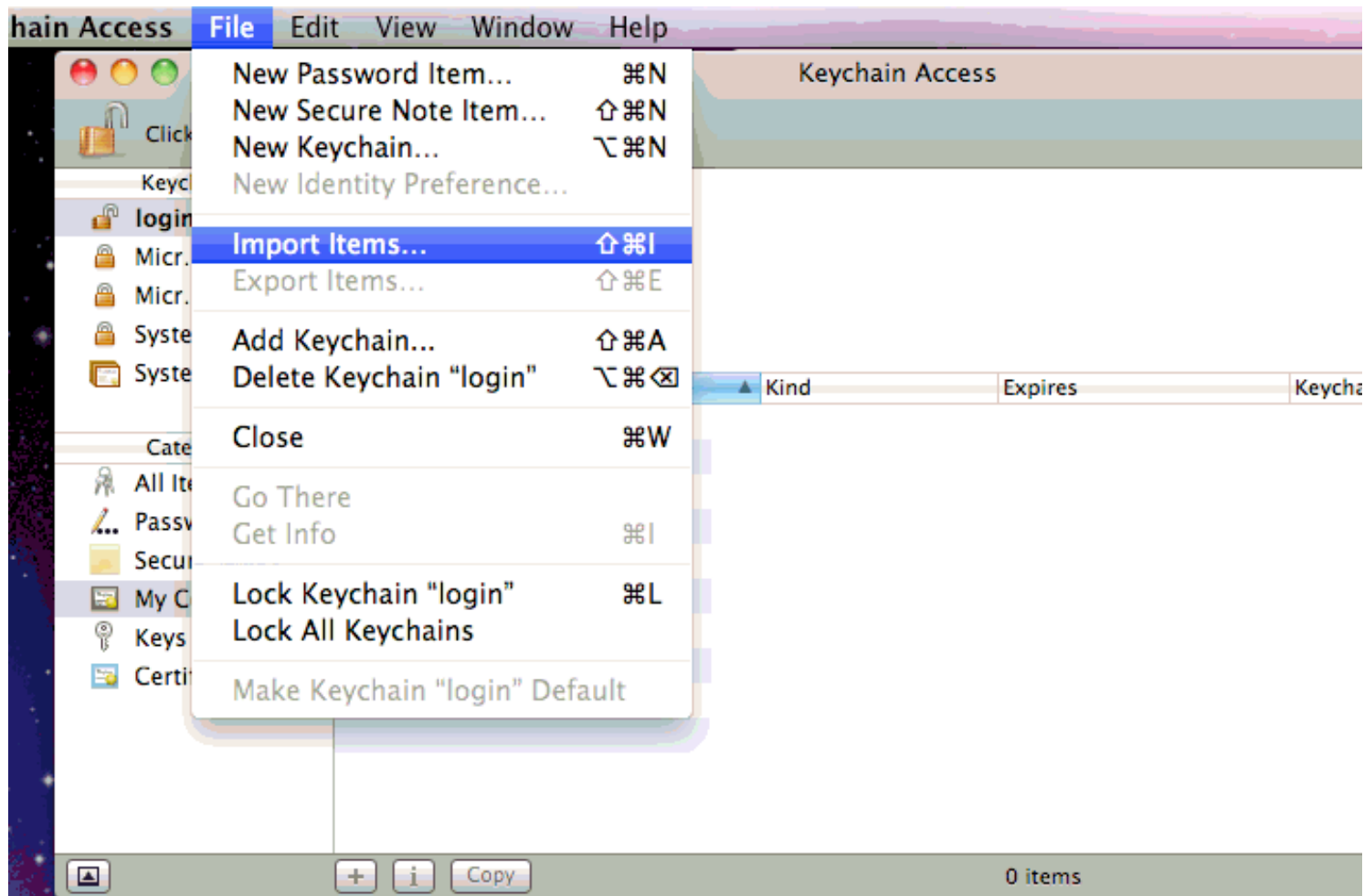


# Outlook On Apple OS X Uses the Apple Keychain; To Do S/MIME with Outlook, We Need To Get Our Cert Into It

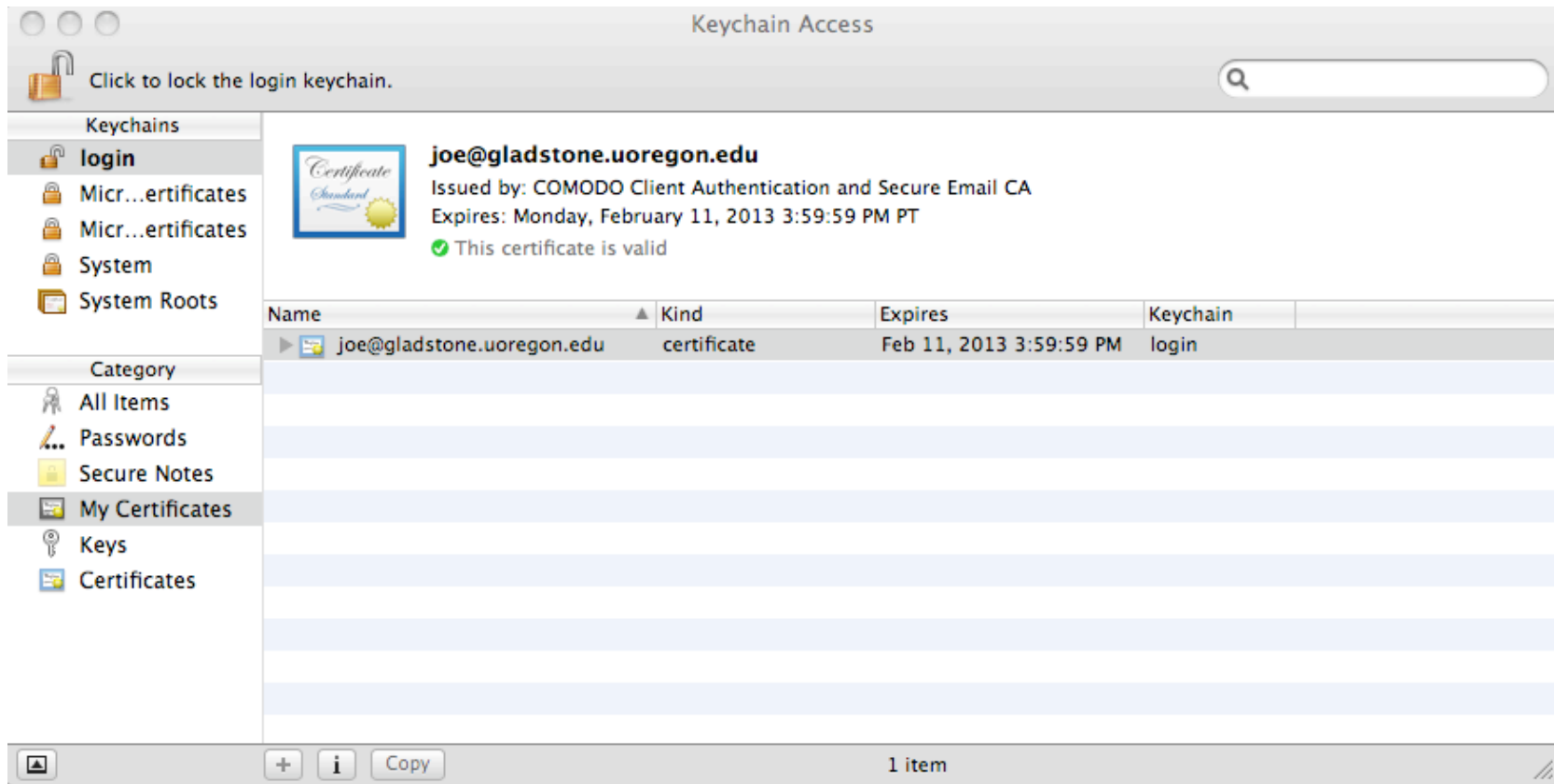


Can't find Keychain Access? Check Applications --> Utilities

# Importing Our Key/Cert

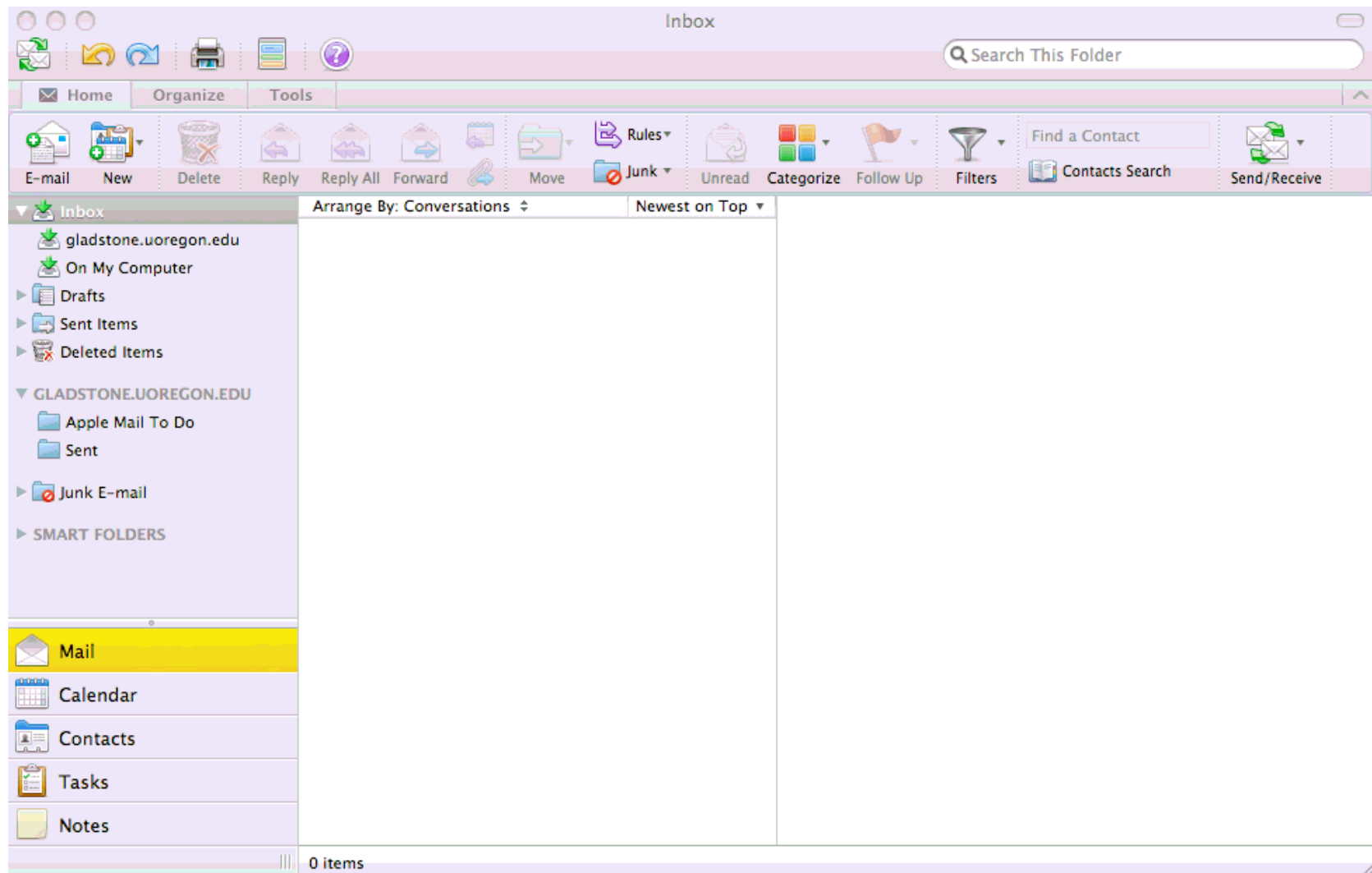


# Success Importing Our Key and Cert

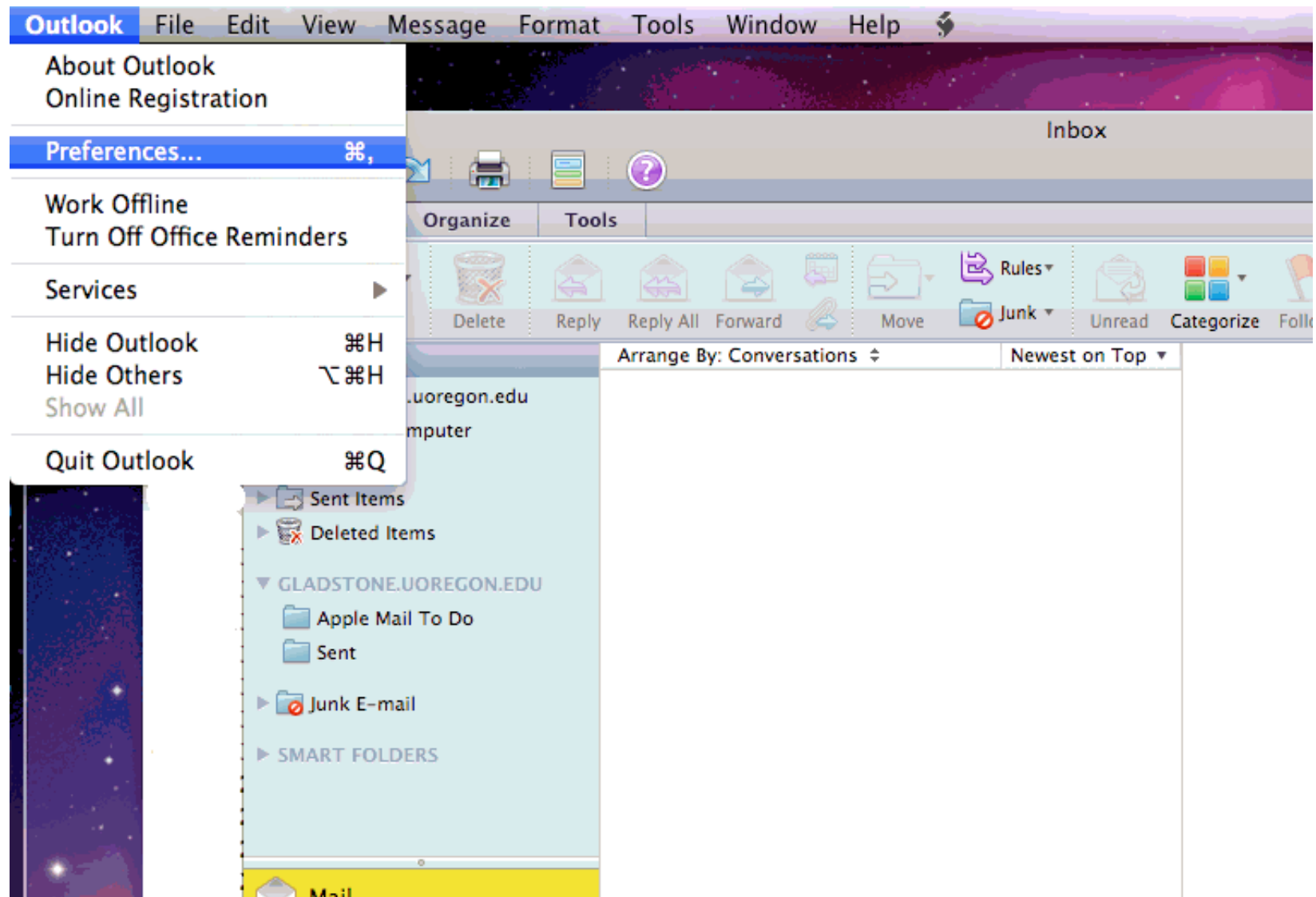


Now we're ready to launch Outlook...

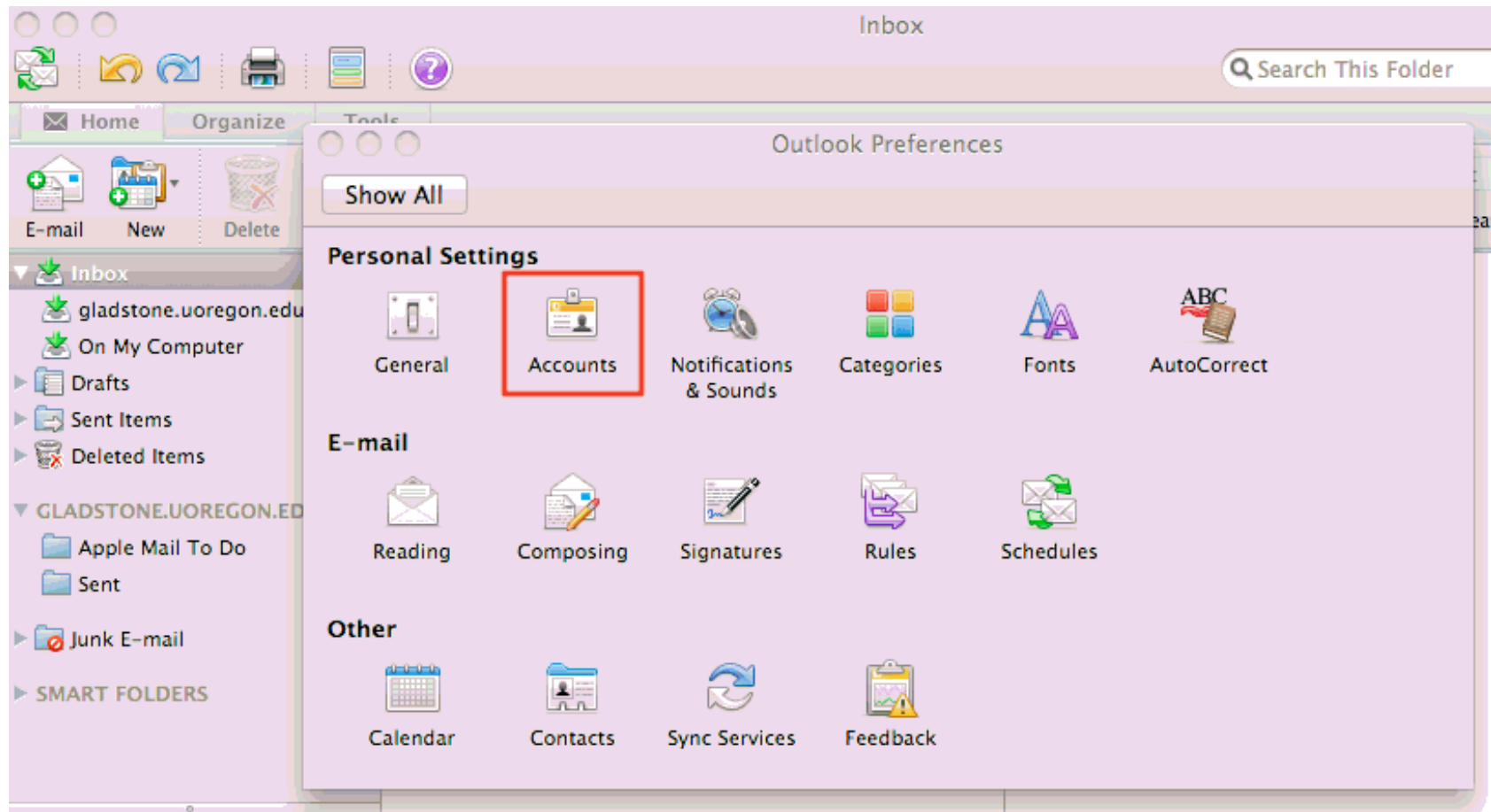
# Outlook's Opening Screen...



# Outlook --> Preferences...



# Accounts



# Advanced Button...

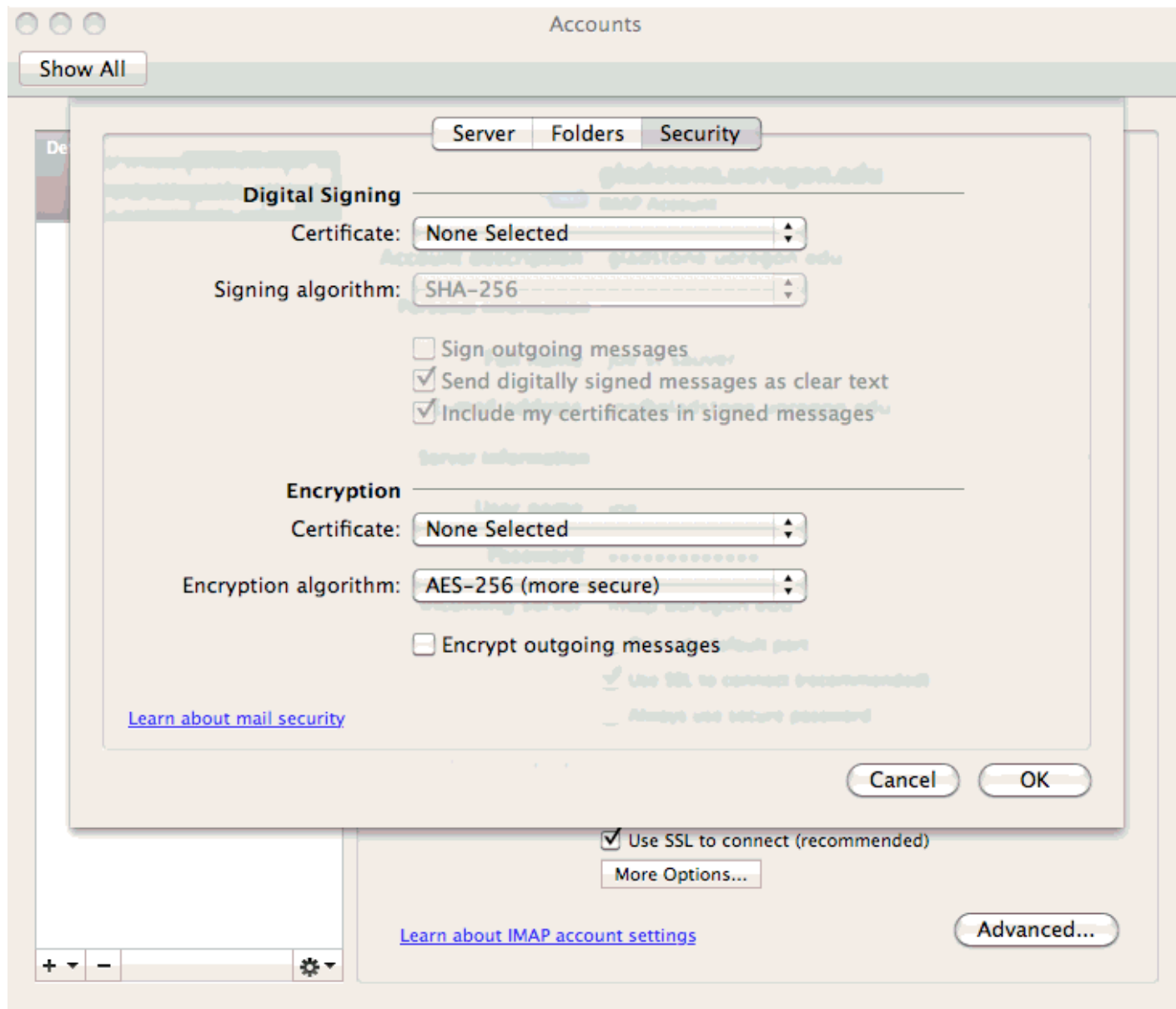
The screenshot shows a window titled "Accounts" with a "Show All" button at the top left. On the left side, there is a list of accounts under the heading "Default Account". The selected account is "gladstone.uoregon.edu" with the email address "joe@gladstone.uoregon.edu".

The main area displays the configuration for the selected account, "gladstone.uoregon.edu", which is an IMAP Account. The configuration is organized into sections:

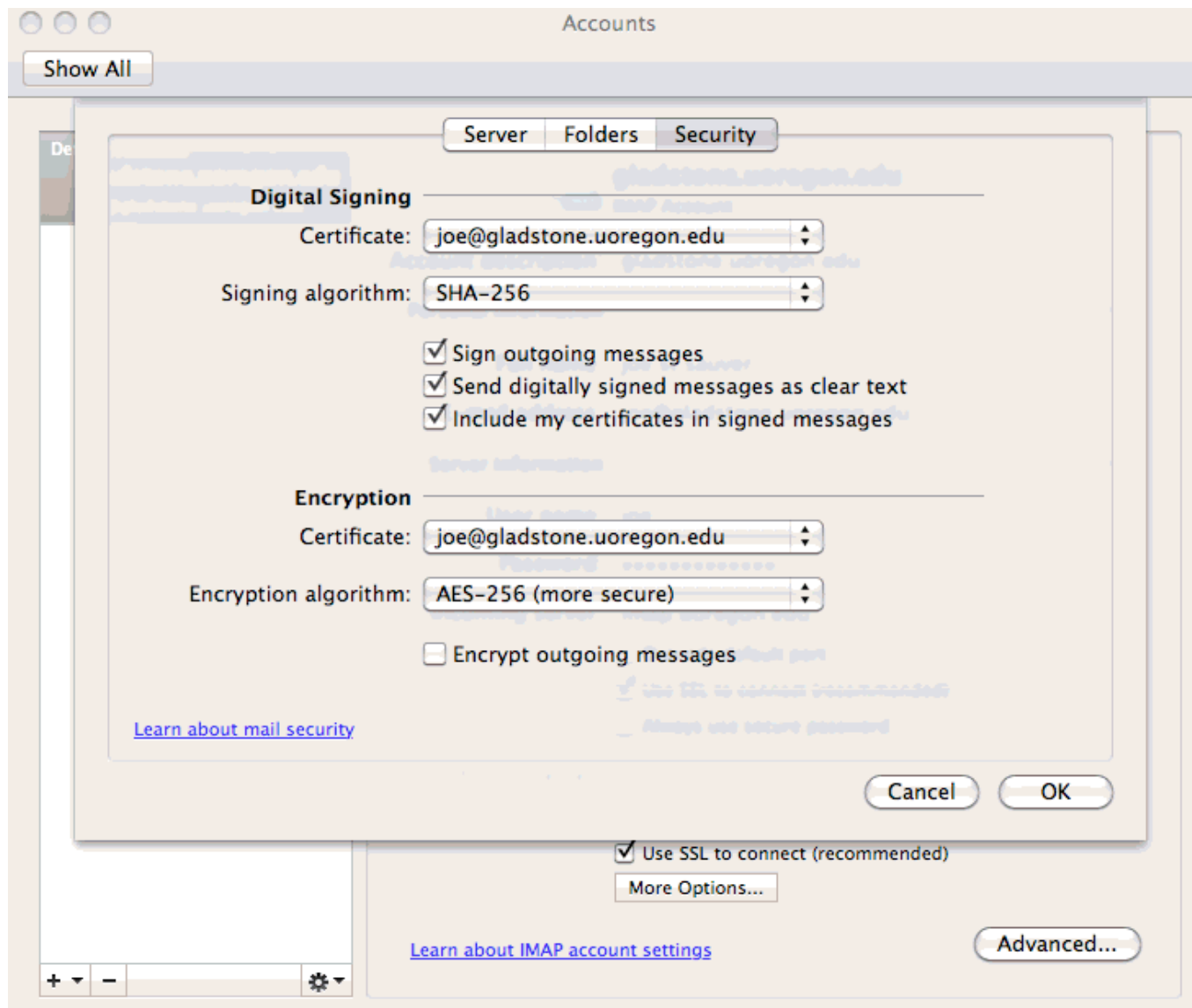
- Account description:** gladstone.uoregon.edu
- Personal information:**
  - Full name: joe st sauver
  - E-mail address: joe@gladstone.uoregon.edu
- Server information:**
  - User name: joe
  - Password: [masked]
  - Incoming server: imap.uoregon.edu : 993
    - ☐ Override default port
    - ☒ Use SSL to connect (recommended)
    - ☐ Always use secure password
  - Outgoing server: smtp.uoregon.edu : 25
    - ☐ Override default port
    - ☒ Use SSL to connect (recommended)
    - [More Options...](#)

At the bottom left, there is a link: [Learn about IMAP account settings](#). At the bottom right, there is a button labeled "Advanced..." which is highlighted with a red rectangular box.

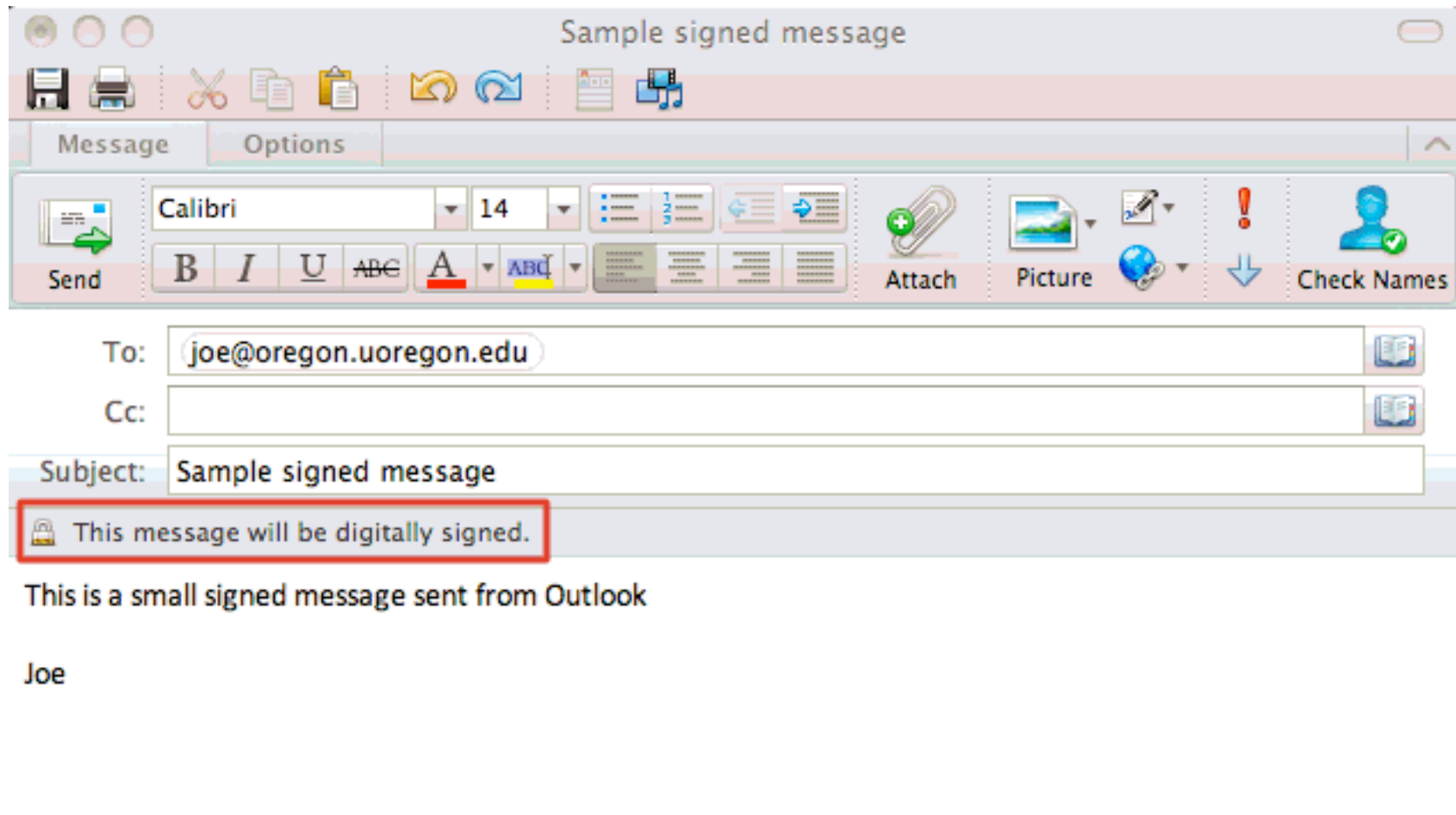
# Picking A Cert on the Account Security Tab



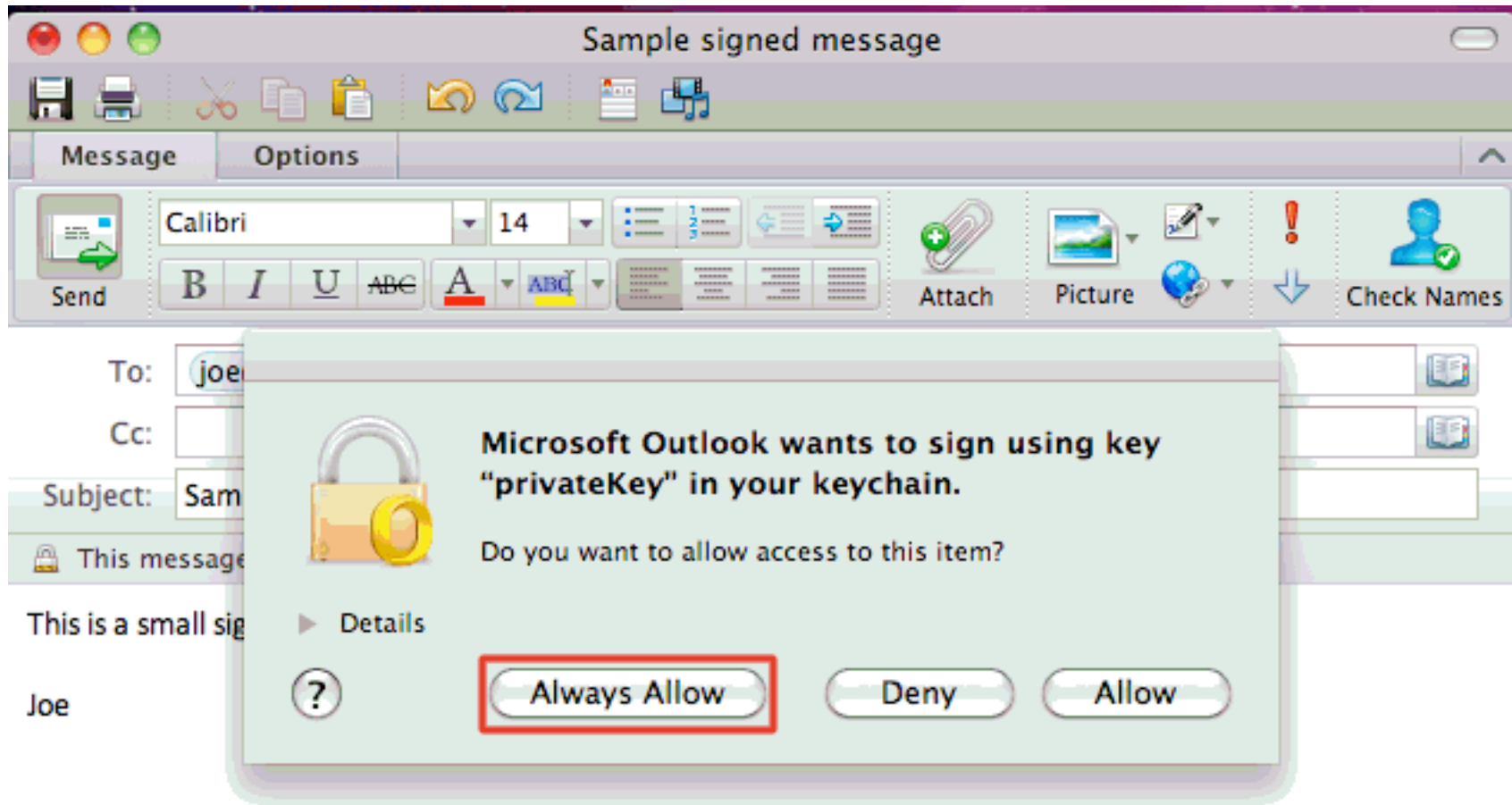




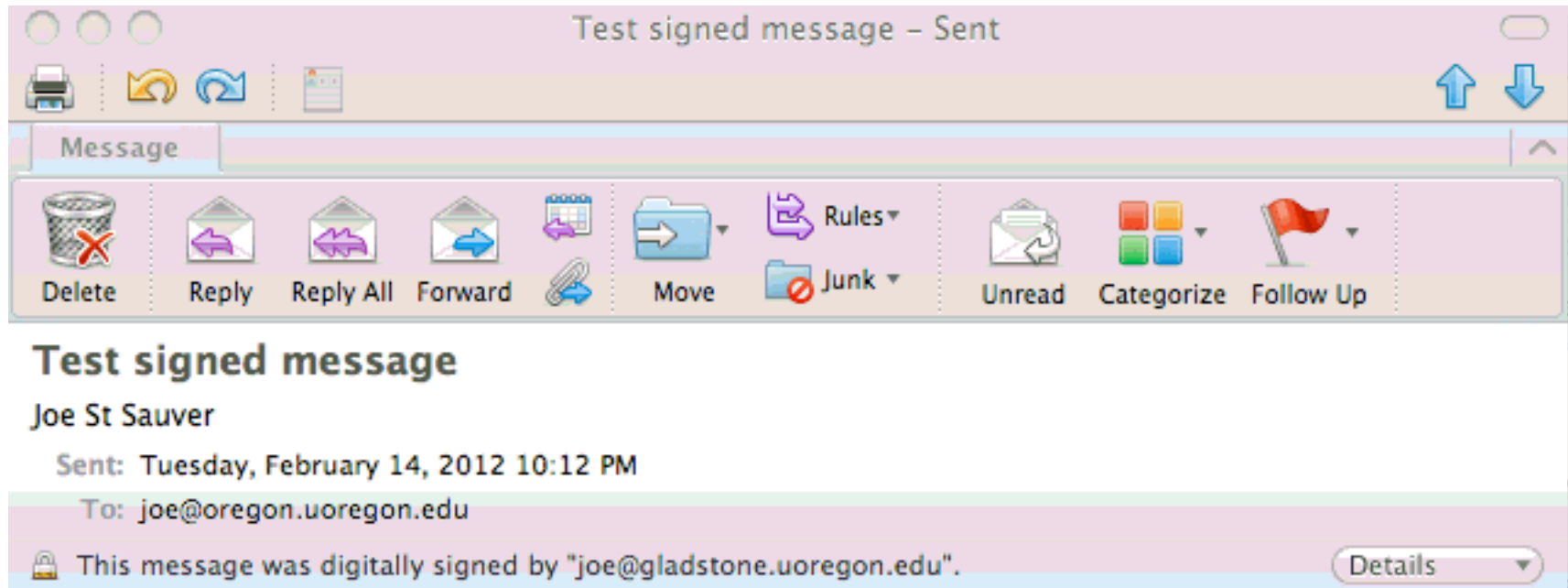
# What The Sender Sees When Sending A Signed Message in Outlook



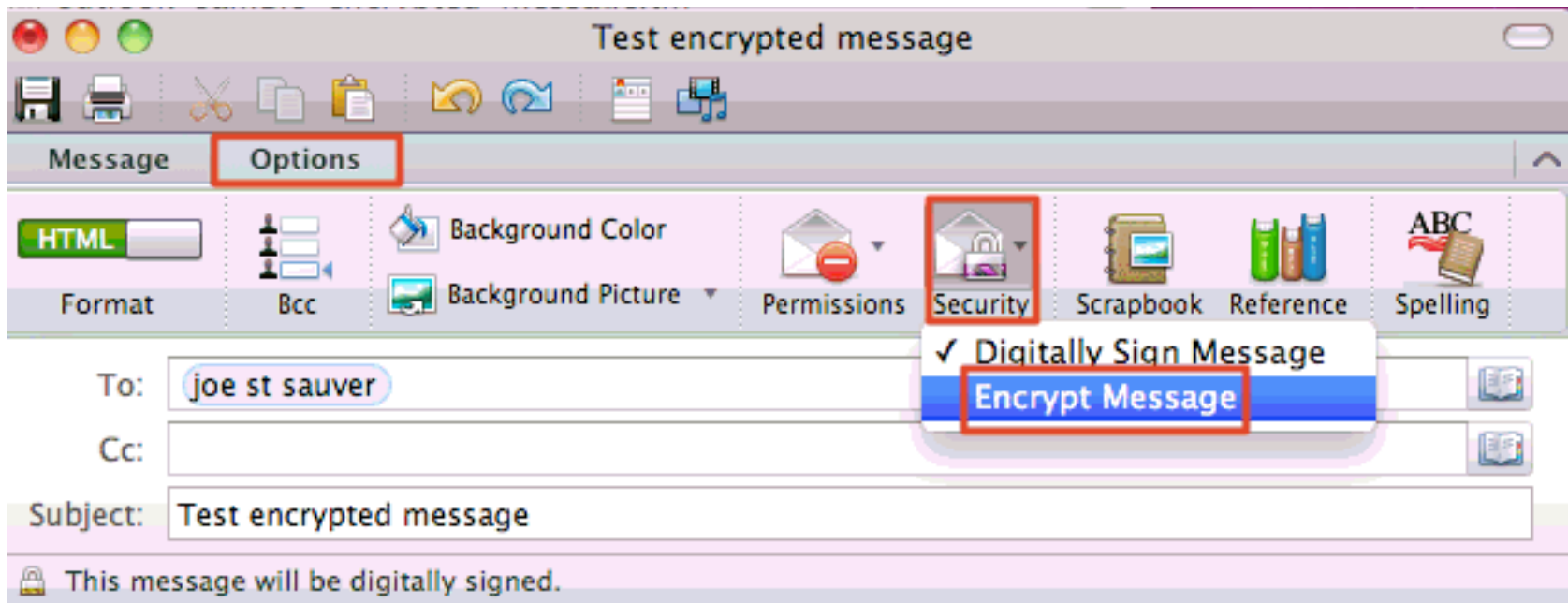
# Outlook Asks For Confirmation The First Time It Uses Your Private Key/Certificate



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



# What If We Want To Encrypt A Message?



This message is going to be signed and encrypted

## **IX. "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)

Penango

www.penango.com

penango

LearnPricingPartnersSupport

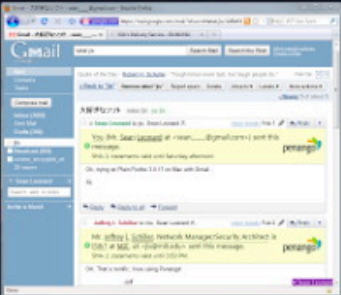
Try Penango!

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

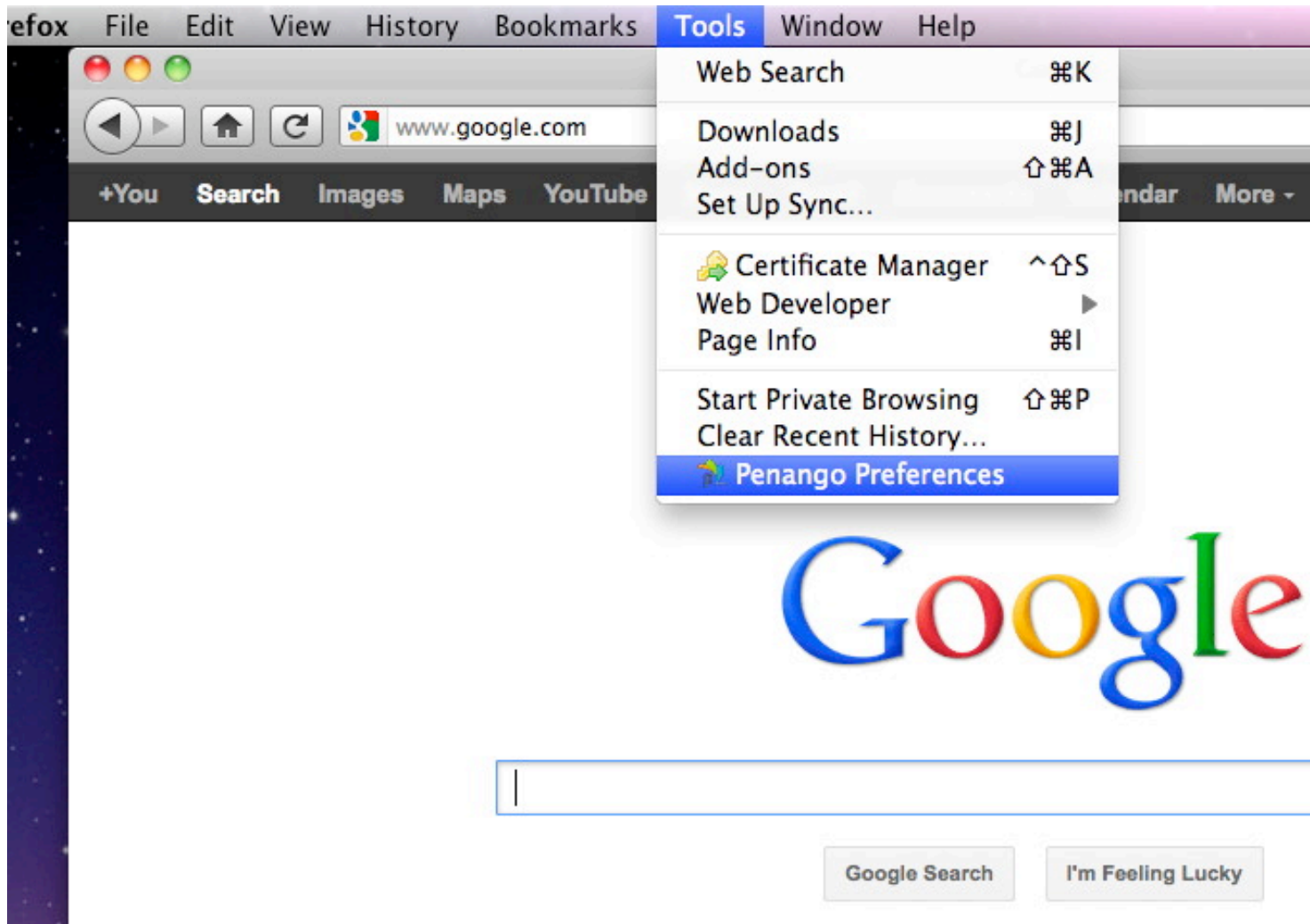


In addition to Gmail, Penango works in Zimbra Collaboration Suite and Zimbra Desktop. 66 million users can't be wrong.

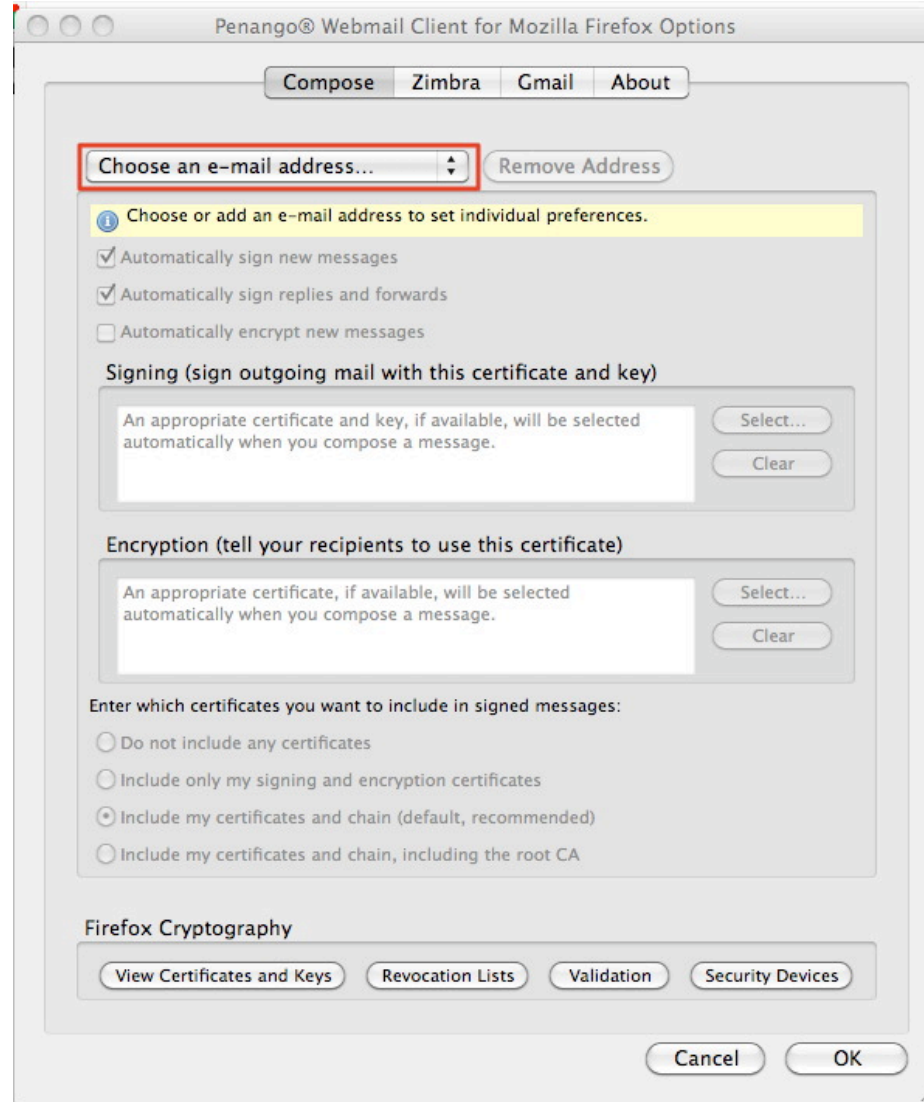
96



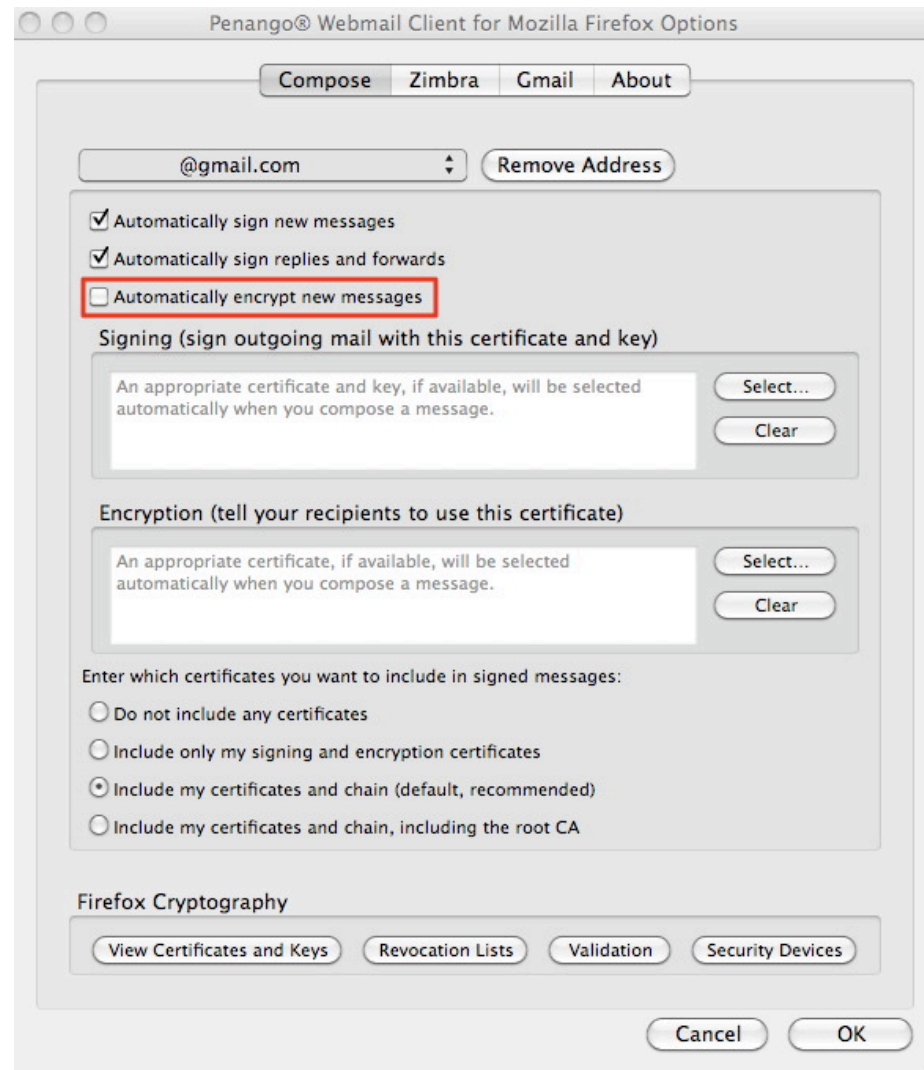
# Once You Have Penango Installed, Open Penango's Preferences in Firefox



# Plug In Your Gmail Address



# Uncheck “Automatically encrypt new messages”



# Composing a Signed Gmail Msg With Penango

The screenshot shows a Gmail 'Compose Mail' window. The browser address bar displays 'https://mail.google.com/mail/?shva=1#drafts/'. The Gmail navigation bar includes links for Maps, YouTube, News, Gmail, Documents, Calendar, and More. The compose header features a 'SEND' button, 'Save Now', 'Discard', and a status message 'Draft autosaved at 11:19 PM (0 minutes ago)'. The email fields are filled with 'To: joe@internet2.edu' and 'Subject: Test signed message (using Penango)'. A green banner with a red border contains the text: 'Recipients using Penango will see that according to Comodo, someone at <[redacted]@gmail.com> sent this message.' The Penango logo is on the right. Below the banner, there are links for 'Rich formatting' and 'Check Spelling'. The email body contains the text '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 you're just signing what was a plain-text-only message, and Penango 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.

## **X. 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.

# 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. 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.



# Safenet eToken PRO 72K

- Through the generosity of Chen Arbel at Safenet, we're able to provide each MAAWG S/MIME 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](http://commons.wikimedia.org/wiki/File:EToken_PRO_USB.jpg)

# 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

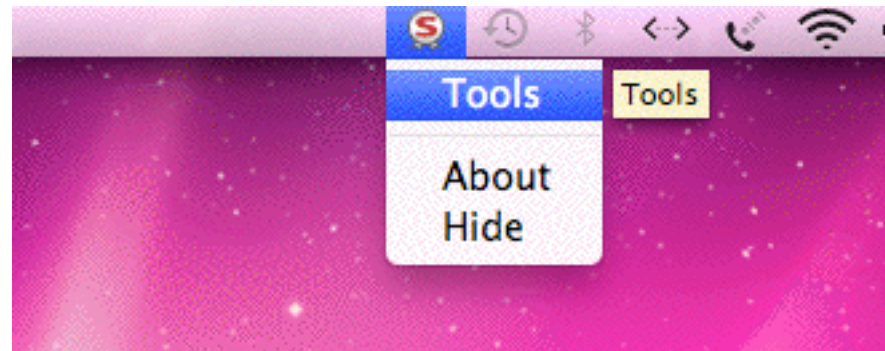
# Installing On the Mac

- 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, read the Administrator's Guide and read the ReadMe 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**

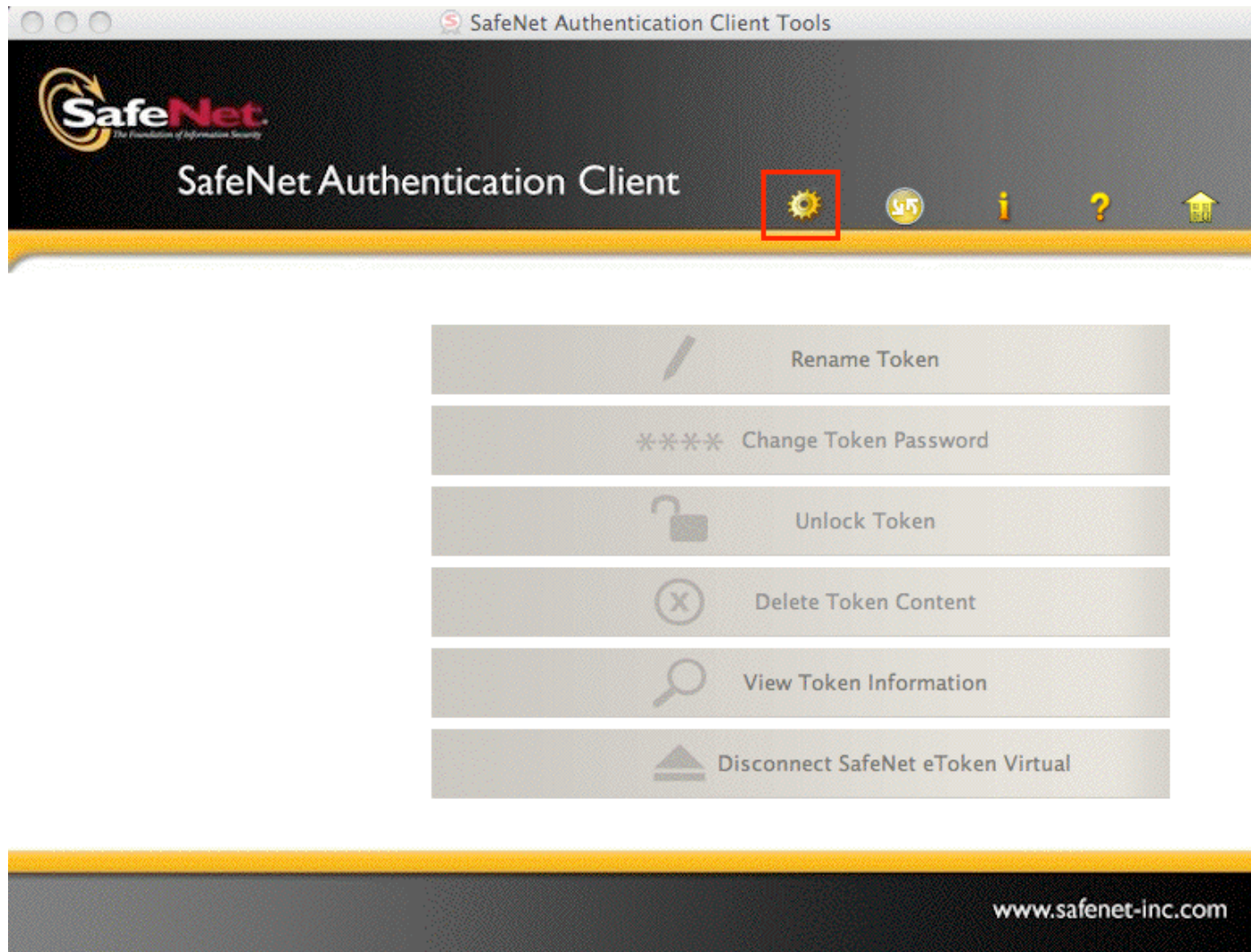
# Firefox Security Module

- 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 security security module 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.

# Now Launch the SafeNet Authentication Tools

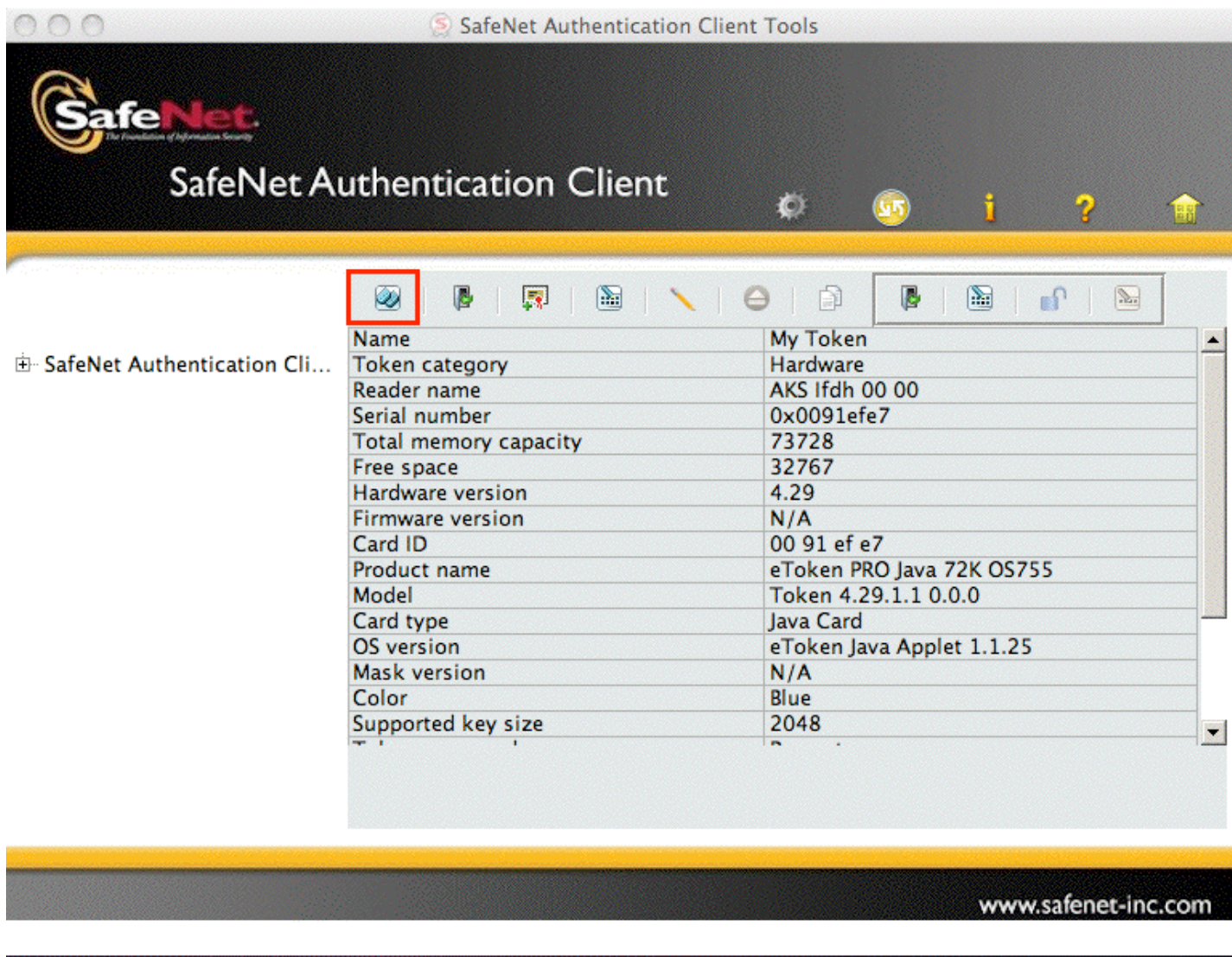


# Go To The Gear Menu ("Advanced")





# View The Token, Then Initialize It



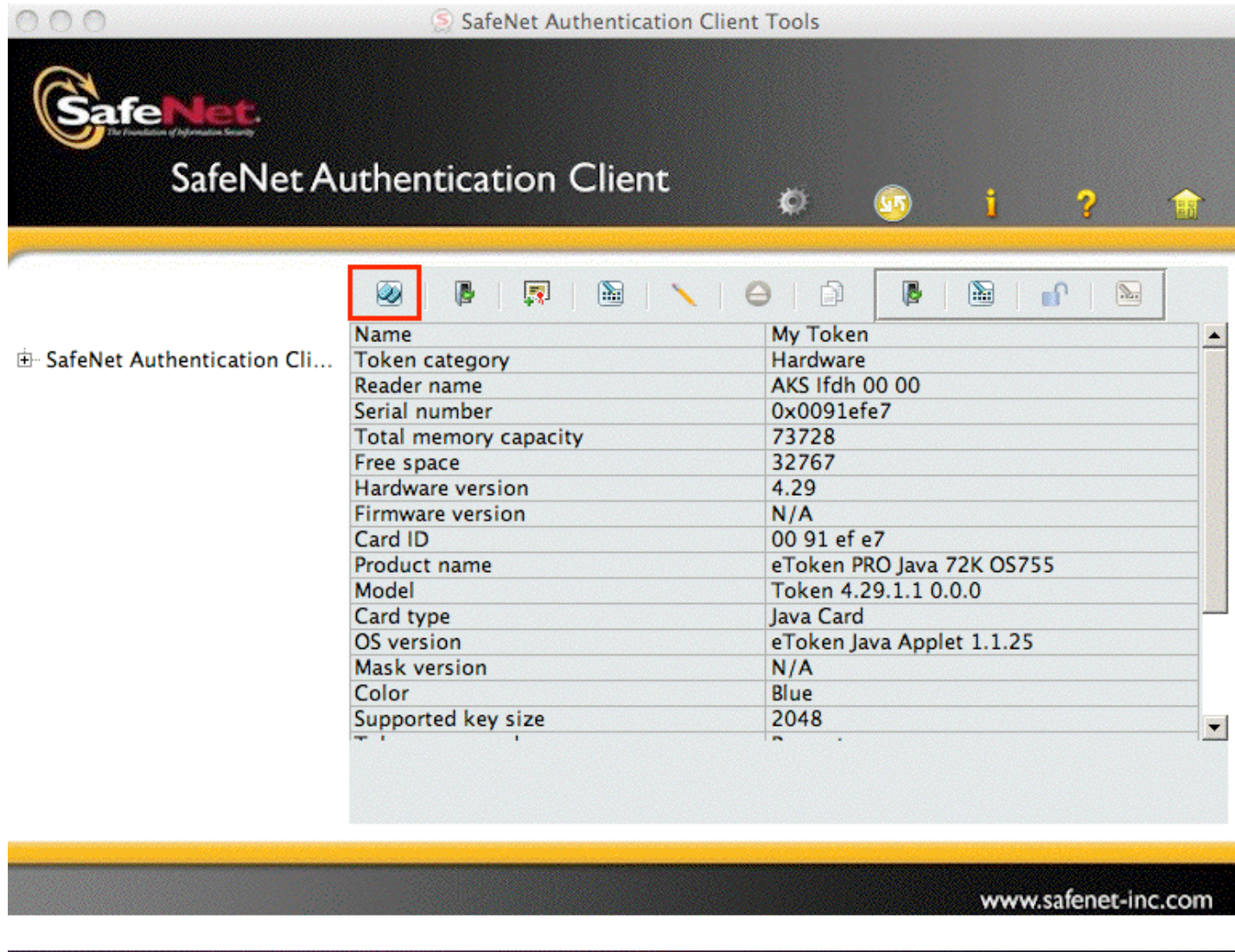
The screenshot shows the 'SafeNet Authentication Client Tools' window. The title bar reads 'SafeNet Authentication Client Tools'. The main window has a dark header with the SafeNet logo and the text 'SafeNet Authentication Client'. Below the header is a yellow bar with several icons. The main content area displays a table of token information. A red box highlights the 'My Token' icon in the top toolbar. The table lists various attributes of the token, including its name, category, reader name, serial number, memory capacity, hardware and firmware versions, card ID, product name, model, card type, OS version, mask version, color, and supported key size.

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

www.safenet-inc.com



# View The Token, Then Initialize It



The screenshot shows the 'SafeNet Authentication Client Tools' window. The title bar reads 'SafeNet Authentication Client Tools'. The main window has a dark header with the SafeNet logo and the text 'SafeNet Authentication Client'. Below the header is a yellow bar with several icons. The main content area displays a table of token information. A red box highlights the 'My Token' icon in the top toolbar. The table lists various attributes of the token, including its name, category, reader name, serial number, memory capacity, hardware and firmware versions, card ID, product name, model, card type, OS version, mask version, color, and supported key size.

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

www.safenet-inc.com



# Enter Your New Passwords and Then Go To The Advanced Screen

Initialize Token

 SafeNet Authentication Client

Token Name:

☒ Set Token Password:  Logon retries before token is locked:   
Confirm:

☒ Set Administrator Password:  Logon retries before token is locked:   
Confirm:

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 Authentication Client

☐ 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

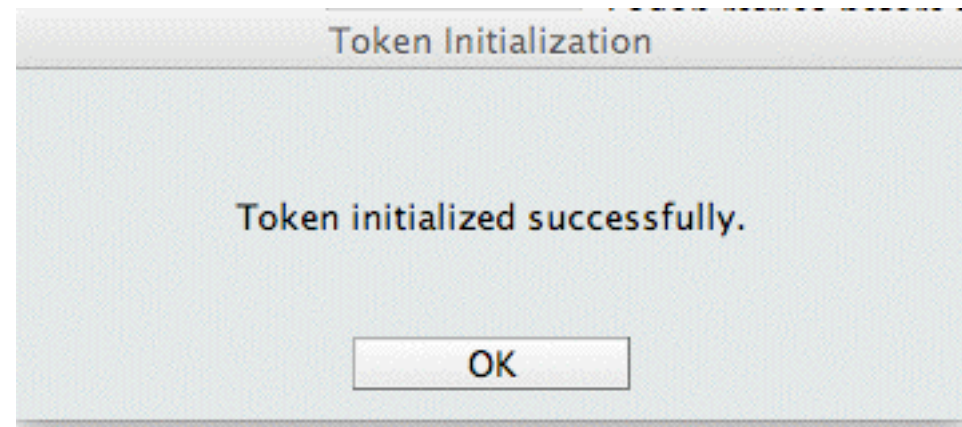
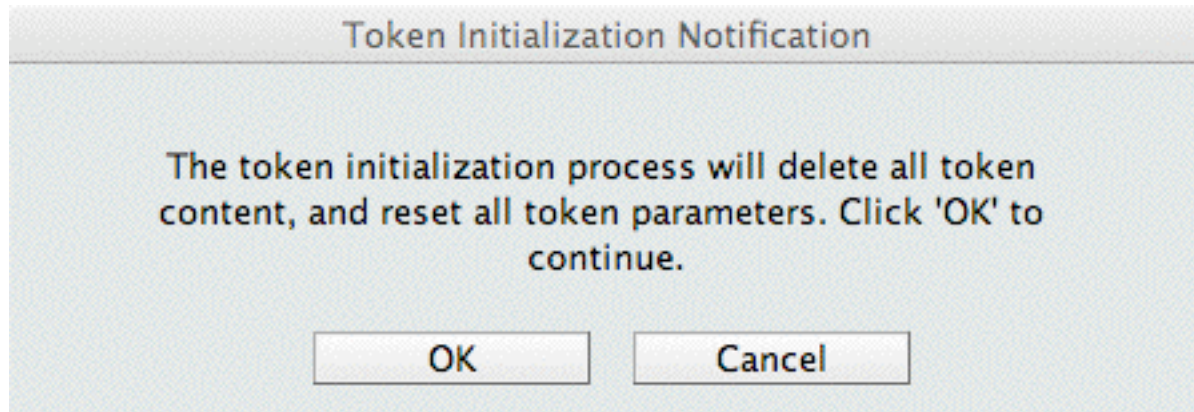
☐ Manually set the number of reserved RSA keys  
0 1024-bit keys

Change Initialization Key

OK Cancel

***DO \*NOT\* SELECT FIPS MODE!***

## Now Actually Initialize The Hard Token...





# Login To The Hard Token


The screenshot shows the 'SafeNet Authentication Client Tools' window. The title bar reads 'SafeNet Authentication Client Tools'. The main window has a dark header with the 'SafeNet' logo and the text 'SafeNet Authentication Client'. Below the header is a yellow bar with several icons. The main content area is divided into a left sidebar and a right pane. The sidebar shows a tree view with 'SafeNet Authentication Cli...', 'Tokens', and 'Client Settings'. The 'Tokens' folder is expanded, and 'Joe's Token' is selected. The right pane displays a table of token details.

Name	Joe's Token
Token category	Hardware
Reader name	AKS lfdh 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

www.safenet-inc.com

# You'll Need To Enter Your Password For It

Log on: Joe's Token

 SafeNet Authentication Client

Enter the Token Password.

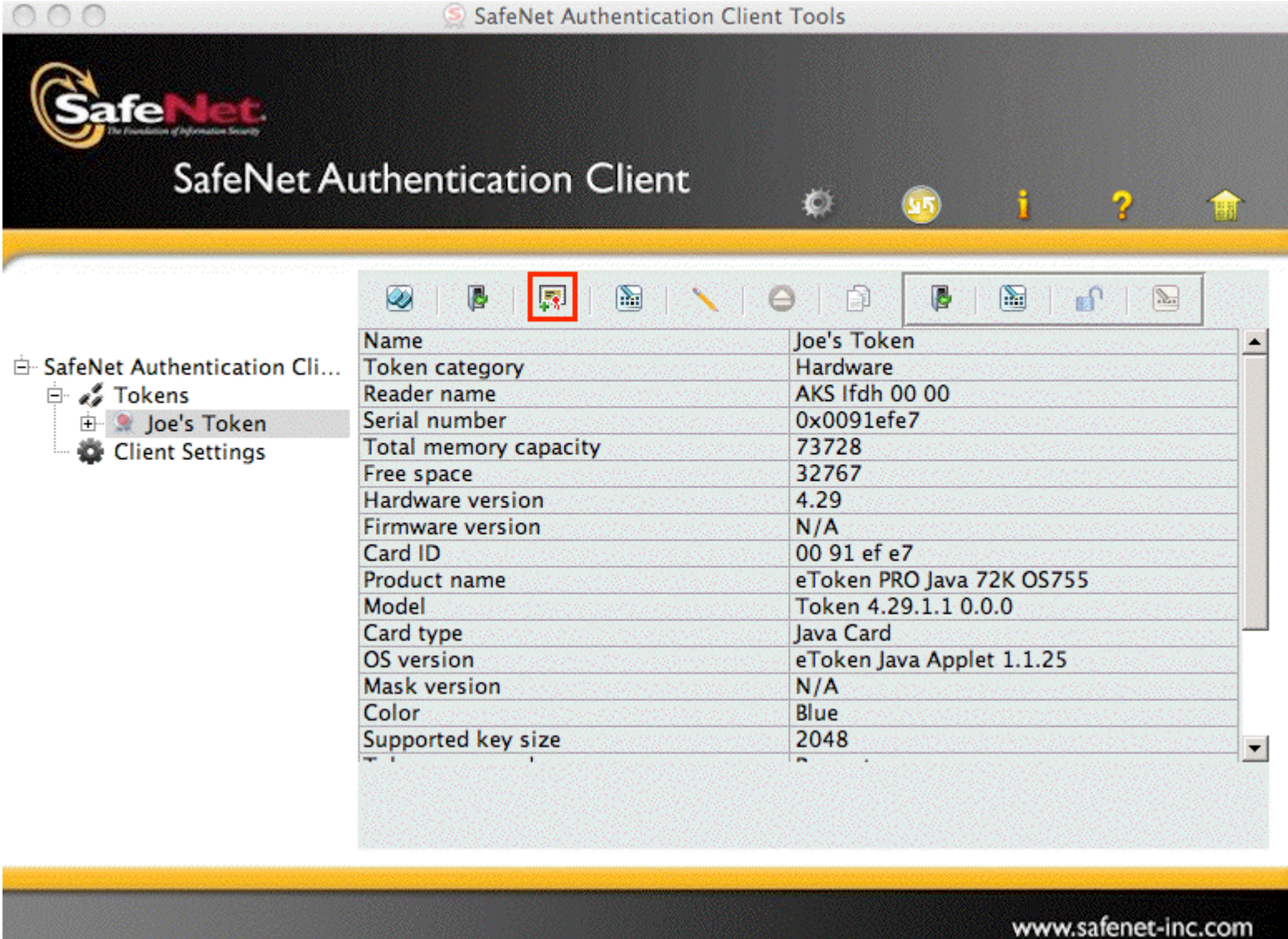
Token Name:

Password:

OK Cancel



# Go To The Import Cert Screen

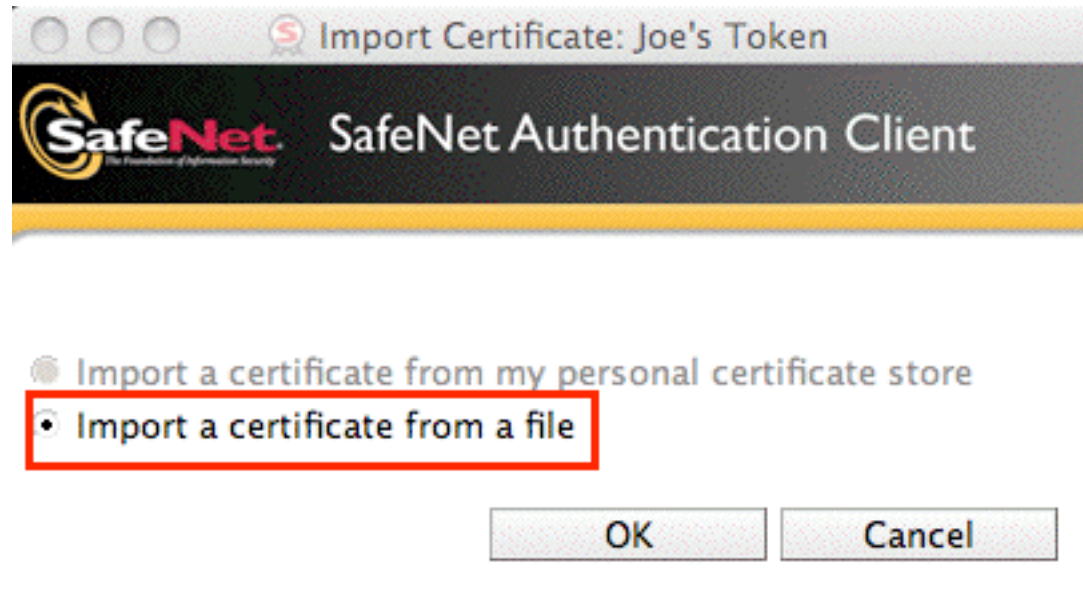


The screenshot shows the 'SafeNet Authentication Client Tools' window. The title bar reads 'SafeNet Authentication Client Tools'. The main window has a dark header with the 'SafeNet' logo and the text 'SafeNet Authentication Client'. Below the header is a yellow bar with several icons: a gear, a download icon, an information icon, a question mark, and a home icon. The main content area has a toolbar with icons for various functions. The 'Import Cert' icon, which shows a certificate and a plus sign, is highlighted with a red box. To the left of the main content area is a tree view showing the 'SafeNet Authentication Client' structure, with 'Tokens' expanded and 'Joe's Token' selected. The main content area displays a table of details for 'Joe's Token'.

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

www.safenet-inc.com

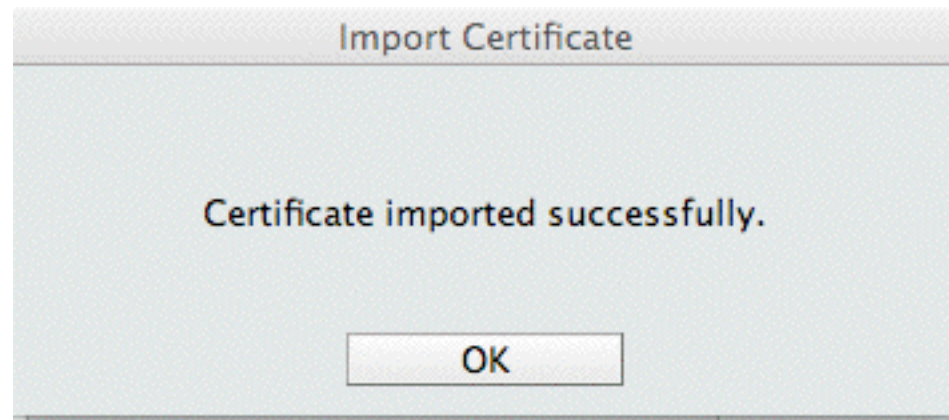
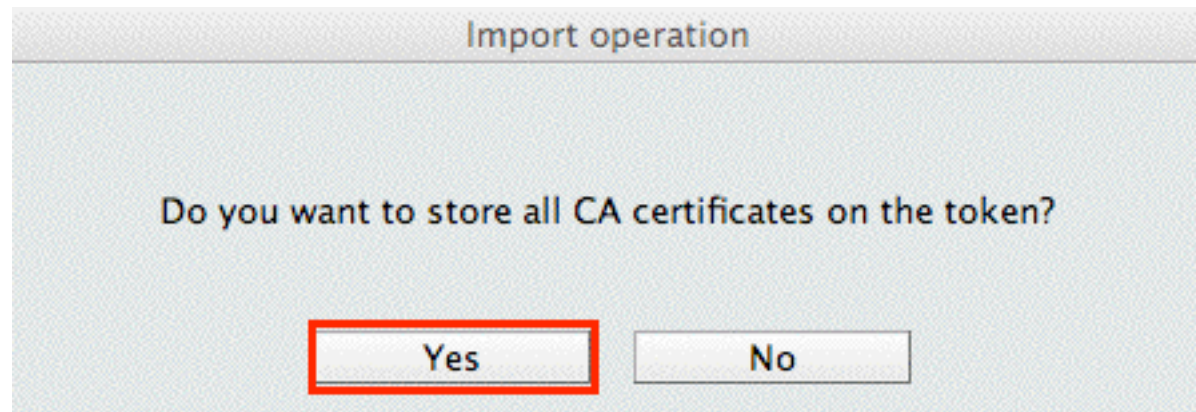
# Import Our Certificate



Pick the p12 backup file we saved earlier.

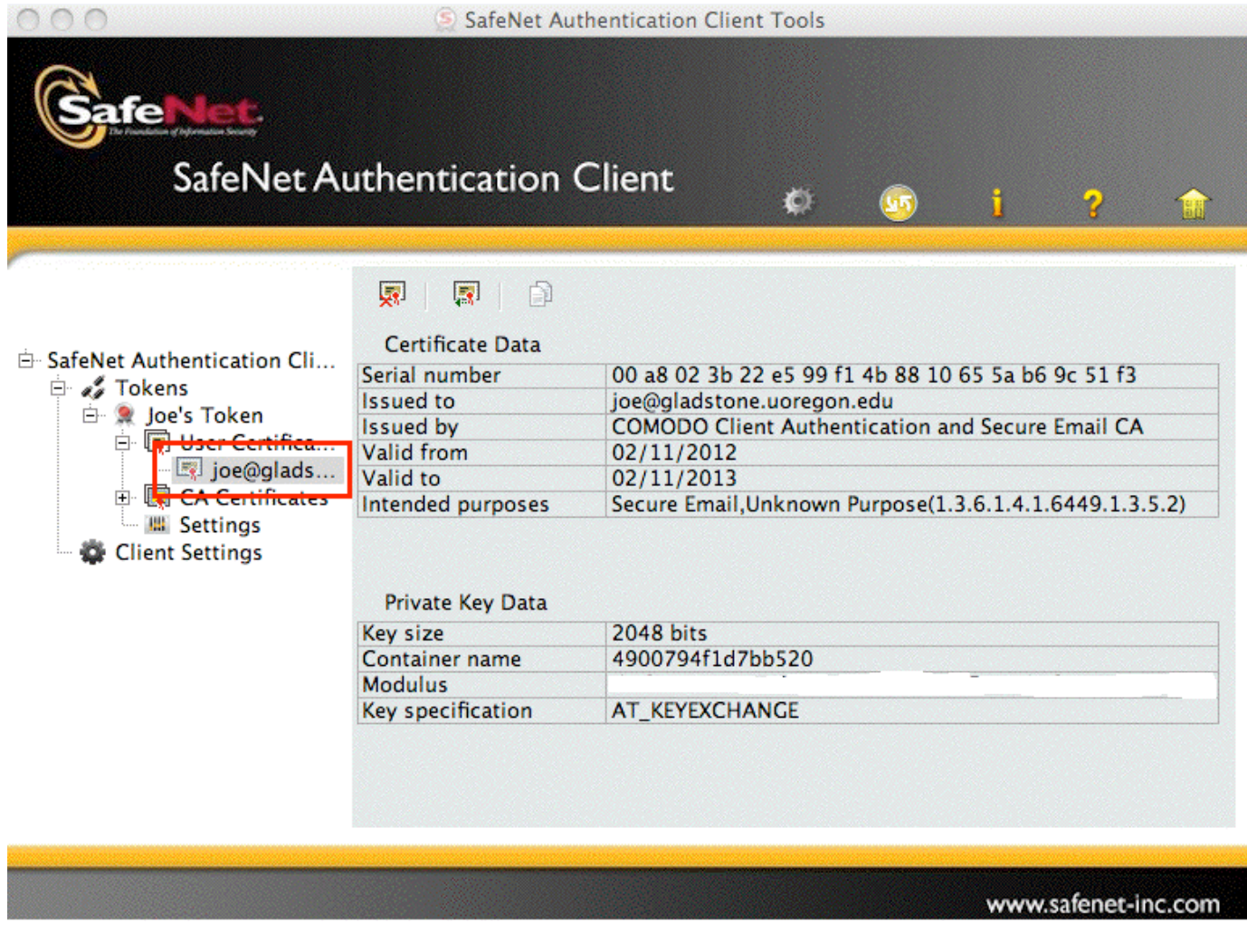
Note that you'll need to provide the password for that backup file in order to load it onto the token.

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

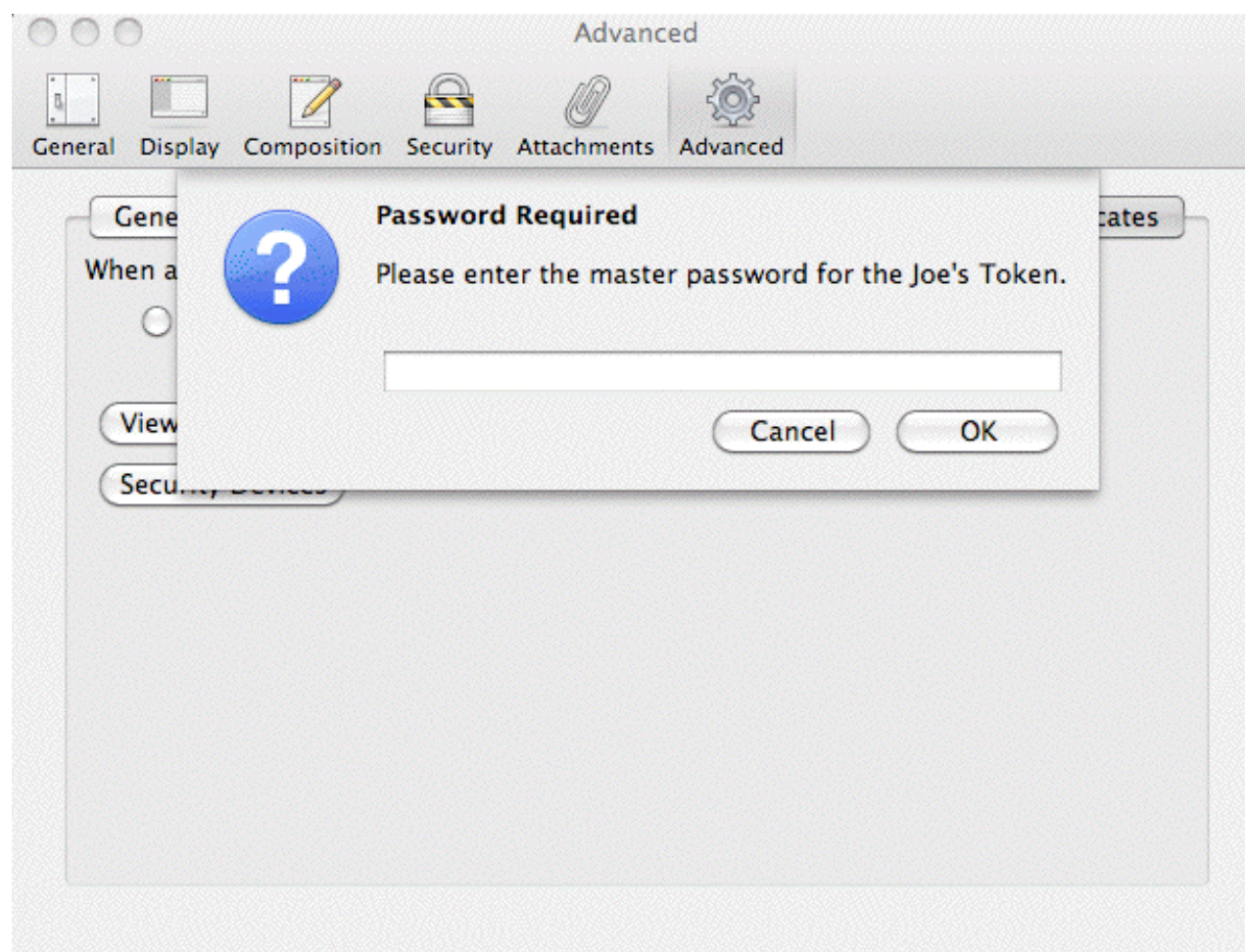




# View The Certs On The Hard Token

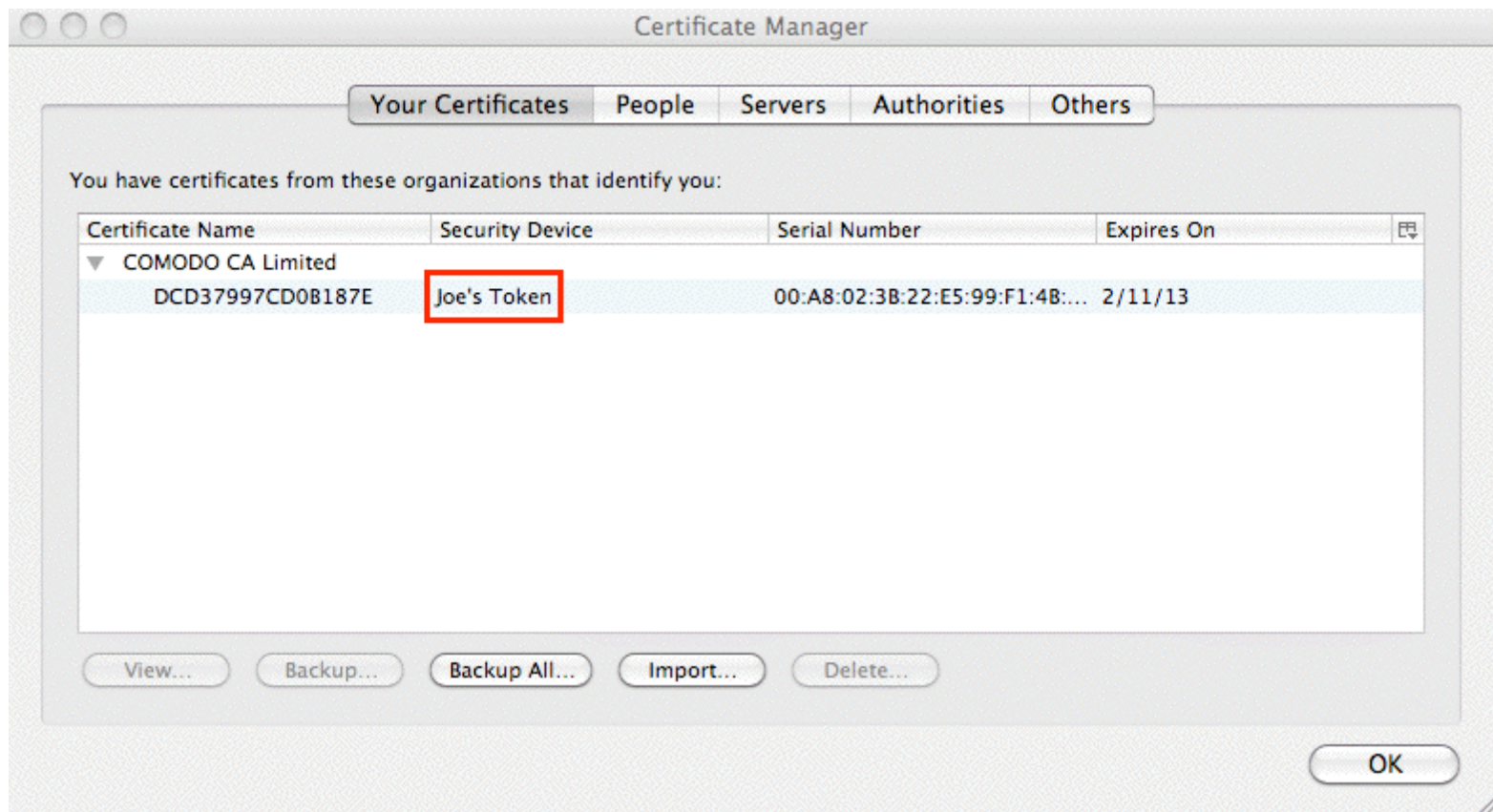


# Tell Thunderbird To Use The Hard Token; We Need To Unlock The Token, First

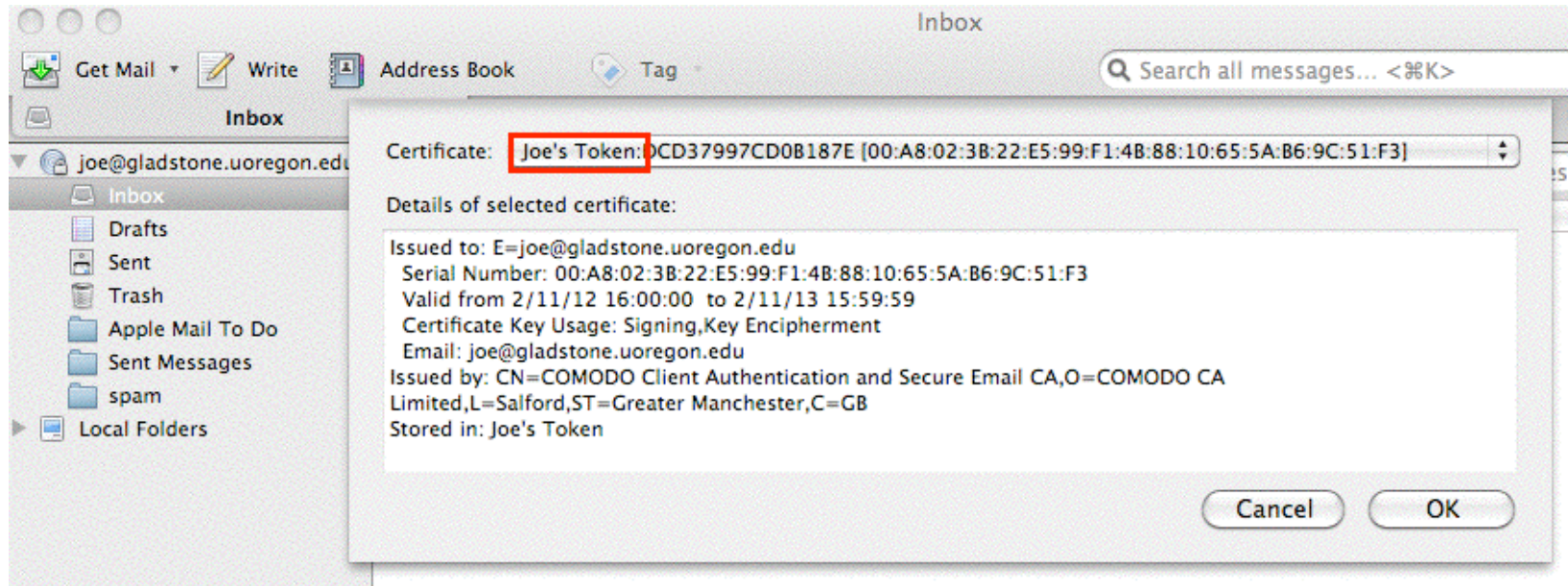




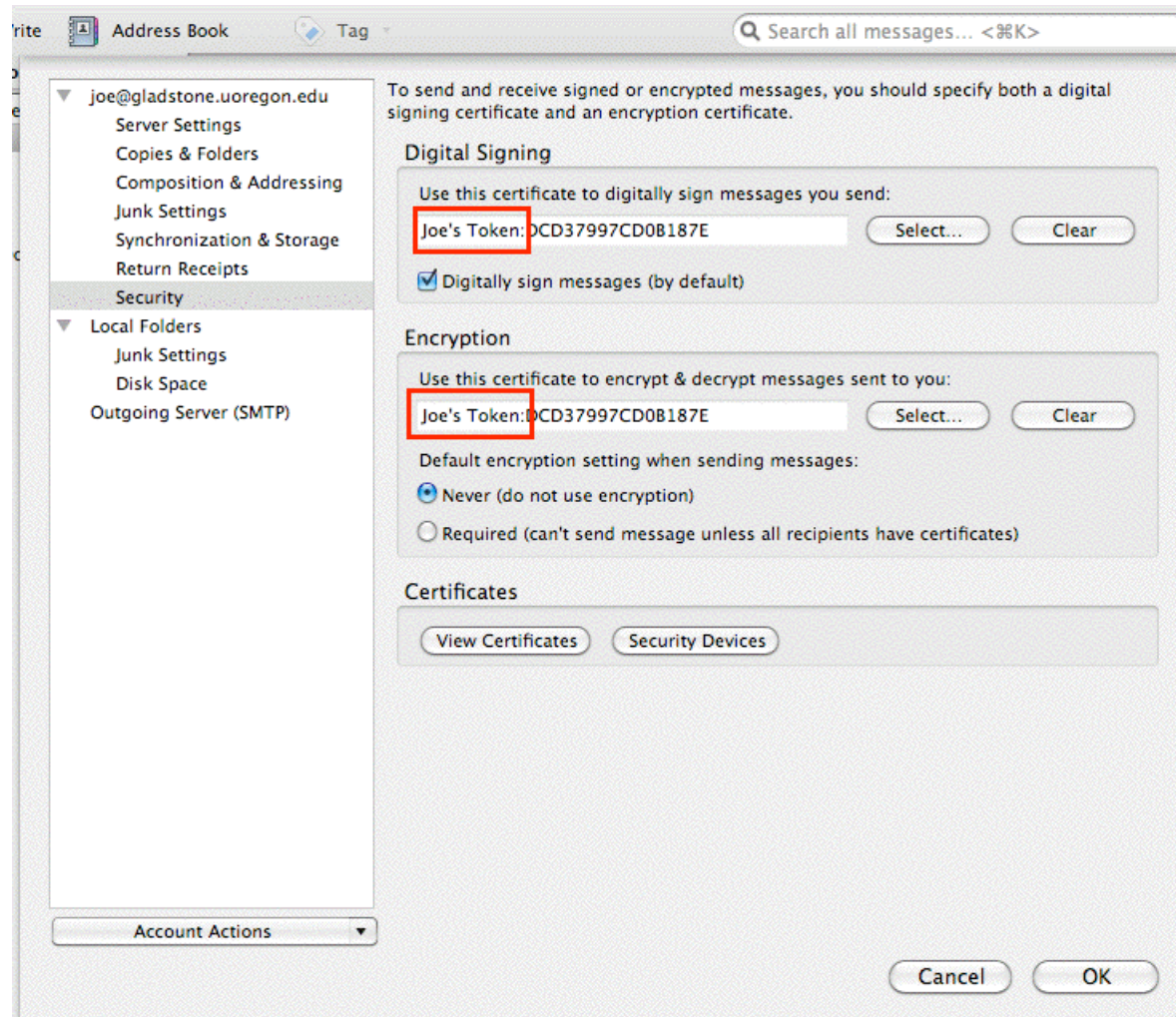
# We're Then Shown The Token and Its Cert



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



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



## **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 free client certs 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.

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

- The processes you saw earlier in this session, while they 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 will be/is working on this).
- Similarly, if you're going to use hardware tokens, instead, you need enterprise grade tools to provision and manage those devices. Those tools can be purchased, or maybe 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 USB interface). That can be very convenient – it's "all in one."
- However, smart cards tend to be cheaper than USB format tokens, 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

- Since MAAWG has a new emphasis on "mobile" :-), 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 buy so-called Bluetooth smartcard readers (sometimes also known as "CAC sleds") to allow BlackBerry or selected other mobile devices to access smart cards via secure Bluetooth, but they may cost \$200+. See [www.apriva.com/products/iss/authentication/reader](http://www.apriva.com/products/iss/authentication/reader)
- Android? iPhone? See <http://www.biometricassociates.com/products-baimobile/smart-card-reader-iphone-android.html>

# 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).
- That method of 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).

# PGP/GPG-ish S/MIME Keyservers?

- Ironically, one of the things that makes Internet scale directories difficult is... wait for it... spam. Can you imagine how much a spammer would love to be able to harvest email addresses for "everyone on the Internet" from a single central directory server?
- There is one 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 you their keys 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.

## Signing Stuff (Other Than Just Using S/MIME)

- Client certs can do lots more, including signing documents...
- 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>

# 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 deprecated native support for smart cards, but third party providers do still offer support, 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**  
[http://httpd.apache.org/docs/2.0/ssl/ssl\\_howto.html#allclients](http://httpd.apache.org/docs/2.0/ssl/ssl_howto.html#allclients)  
(see also) [www.dwheeler.com/essays/apache-cac-configuration.html](http://www.dwheeler.com/essays/apache-cac-configuration.html)
- 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](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

CAC Card Readers | PIV Card Readers | Single Door Access Solutions for Military and High Security applications

http://www.bridgepointsystems.com/

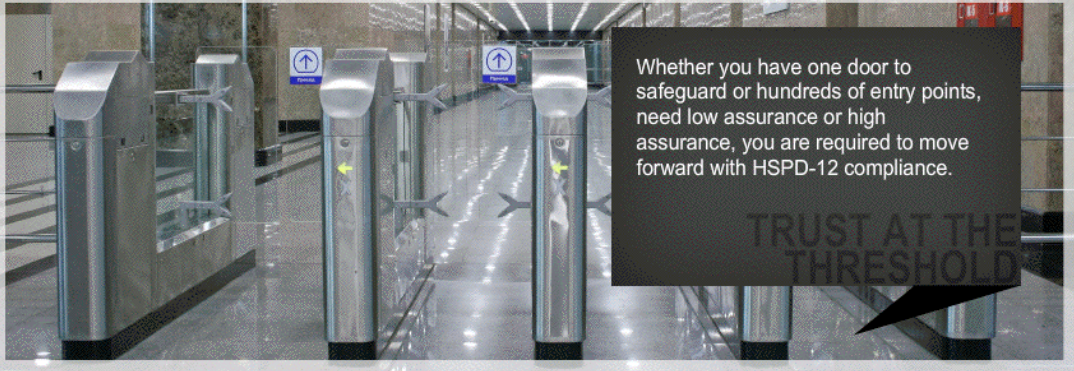
Google

**BRIDGEPOINT™**

Trust at the Threshold™

CompanyProducts and SolutionsMarkets ServedOur AdvantageFAQPartnershipNewsResource CenterContact

You are here » Home



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.

TRUST AT THE THRESHOLD

■ 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 issuers

## **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 I'm struck by is that its success is not just a technological thing, it's a sign that appropriate policies were developed by the community.
- 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.

## **Be Sure To Keep Corporate Counsel 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 issues.

# **Thanks for the Chance To Talk Today!**

- Are there any questions?