# Client Cert Deployment Models and Hardware Tokens/Smart Cards

Joint Techs, Baton Rouge LA

8:45 Monday, January 23rd, 2012

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

http://pages.uoregon.edu/joe/client-cert-models/

# The InCommon Certificate Program

- The InCommon Certificate Program offers unlimited certificates to participating institutions for a flat fee, see https://www.incommon.org/cert/

- That program includes SSL/TLS "https:" server certs, code signing certs, **and client certificates** (sometimes called "personal certs," "S/MIME certs," or "PKI certs").

- We've previously talked about SSL certificates (for example at Joint Techs in Fairbanks, in July 2011, and at the Fall 2011 Member Meeting in Raleigh); today we want to focus just on client certificate deployment.

# Client Certificate Deployment

- While the use of SSL/TLS certificates for web servers, mail servers, etc., is routine and widespread, the same cannot be said for client certificates.

- Client certificate use is still rare in higher education, and across the Internet at large, for that matter.

- **Why haven't PKI certs thrived the same way SSL certificates have?**

- **And what can we do to fix this? (assuming this is something we <u>want</u> to do – note we currently don't feel a similar urge to flog roll out of code signing certs, eh?)**

# It *Isn't* Simply That PKI Is "New"

- 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]

# But In *Some* Cases, Client Certs 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. [We'll talk about other costs, later in this talk]

- 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? Try the free client cert site mentioned a couple of slides back! (We might 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)?

- For example, 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?)
- 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

(Yes, I recognize the irony of my talk w.r.t. this point)

# AMSAC at the Previous Internet2 Member Meeting

- During a brief presentation* to the Applications, Middleware and Services Advisory Council in Raleigh NC on October 4th, 2011, I asked, with respect to client certificates,

  *"Should we be thinking about certificate use cases, or should we be thinking about the sort of credential deployment model we need?"*

  I think we've finally determined that answering *either* of these questions largely determines the answer to the other one, e.g. the questions are jointly interdependent.

* http://pages.uoregon.edu/joe/cases-or-creds/

# So Today We're Going To Focus On Deployment Models, Not On Use Cases

- So if you're sitting here hoping for an introduction to how to use personal certs to do S/MIME, you're going to be disappointed (but feel free to check out http://pages.uoregon.edu/joe/smime/ for a 1 page intro for those using Thunderbird on the Mac), or if your users are on Gmail or Zimbra, see http://www.penango.com/
- I'll also say that if you're using Apache, and your "use case" is that you want to learn how to allow just a small specific set of client cert users to access your web pages, you really should check out Section 4 of 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)
- But coming back to deployment models...

# 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 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 show what local CA to trust).

# 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 local 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)

# 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 *also* 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!*
- *Also Note: I do also recognize that there are a *lot* of subtle issues that are raised by asking those two seemingly simple questions.*
- For example, one subtle issue may be "How much rigor, or care, or vetting, *should* be associated with the potential institutional issuance of client cert credentials?"

# Two Models For Cert Deployment Rigor

- **Model A (Fairly Casual):** *Anyone* can request a client cert by completing a web form and clicking on a link received in a confirming email (implicitly, in that model, your "identity" is your email address). The cert and a provider-generated "private"/public key pair are downloaded over the Internet. The downloaded credentials are saved directly on the user's system (potentially in multiple locations, e.g., in the OS as well as in one or more browsers, or potentially even in multiple locations on multiple systems, such as a desktop and a laptop). The cryptographic credentials might even be able to be accessed/used without requiring the user to enter a password.
- **Model B (Much More Rigorous):** Client certs are deployed using hard tokens (or smart cards) ***only.*** Keys are generated internally on the token, and the private key never leaves the token (and in fact **can't** be exported from it). All access to the token is password protected. Users only get issued a token and associated cert in person, and then only after providing government-issued ID (such as a driver's license). Your credential is actually tied to your "meatspace" identity.

# We *COULD* Even Go *Beyond* Model B...

- When it comes to tying credentials to an identity, some credential issuers go even further, rather than just taking existing governmental identity documents at face value.
- For example, the US government conducts background investigations of all federal employees and contractors as part of the process of issuing PKI-based credentials to those individuals. (That's one reason why it's taken them so long to complete the roll out of their CAC/PIV cards)
- Similarly, if you seek a "Global Entry" credential, allowing you to bypass the (often painfully long lines) at Customs when entering the United States, that process involves more than just presenting existing identity documents (although reviewing existing identity documents is part of the Global Entry application process)

# But Let's NOT Go Overboard In Higher Ed

- Weak credentials don't do much for us, but the harder we make it to "do" client certs, the greater the chance that folks simply won't bother. We need to get the balance right. For example, few higher ed sites would be comfortable doing background checks on all their users, and I ***wouldn't*** recommend proposing such measure.
- Whatever the community does decide on has to be well aligned with community norms, and it must be ***affordable***.
- If client certs cost $200 to $300 per user per year, schools simply won't be able to afford it (e.g., $200/user/year amounts to $5,000,000 per year for a 25,000 user site – that's a lot of money for most sites these days).
- So what might be a "doable" target cost per user per year?

# Target Cost: $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 OTP Tokens (in a 5 pack) @ $55.60/token
- If we ***were*** to do "rigor model B," deploying personal certs on smart cards or hardware tokens, what would the hardware tokens themselves cost? One of the most widely deployed USB PKI tokens costs $36.99 (plus $20.99 for a required three year software license) from a major online vendor, or **$57.98** (quantity one retail), but academic discounts could potentially help substantially here.

# Discounts for Hard Tokens/Smart Card

- It's our hope that by the next Member Meeting, we'll be able to offer popular and widely used PKI smart cards and USB-format PKI hard tokens to our educational participants at a substantial discount.

- While no contract has yet been signed, we anticipate that obtaining substantial discounts might require minimum order quantities, perhaps on the order of 500 or 1,000 tokens.

# "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.

# "But Do We Even *Really* Need To Deploy Hard Tokens or Smart Cards?"

- After all, users CAN store a client certificate directly in their browser or operating system, and then we'd avoid all the cost/hassle of obtaining and deploying hard tokens or smart cards, right?

- Is it possible that using hard tokens or smart cards would be "overkill" for higher education client cert deployments, just like Federal-style background investigations?

- After pondering this particular point, I still think deploying client certificates on hard tokens or smart cards make a lot of sense.

# Users Don't "Map" To A Single Computer

- In the old days (when users were lucky if they had even one computer, and if they did have one, it would be what they'd use for all their work), storing one's client certificate right on that computer had some merit.
- But these days, it's routine for users to have and use multiple machines. Maybe they have a desktop at work, and a laptop they use at home or while travelling (and maybe there's an iPad or iPhone or Android mobile device in the picture, too). If you want to use PKI-enabled applications on all those devices, you need access to your cert on all those platforms, too.
- And then what about student users, with no computer of their own, working from shared lab machines? You sure don't want them to locally save their certs in THAT sort of shared computer environment

# "Why Not Just Issue New Credentials To Each Computer A User Might Work With?"

- That's certainly one model, and if all you're doing is using client certs for authentication or signing, that might work.
- On the other hand, if you're using client certificates for encryption, you're likely going to run into a problem.
- Assume you have a message that was encrypted using the key for system A, but today, because you're travelling or working from home, you're using system B, with different credentials. You won't be able to decrypt that message.
- Some might argue that this issue is a perfect example of why encryption with PKI should not be encouraged, but frankly, supporting encryption may be one of the most compelling justifications in general for deploying PKI!
- See how use cases and deployment models intertwine?

# Hard Tokens/Smart Cards Have 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 issuance processes (such as those for ID cards or door keys); ditto for client cert use processes.**

* Not currently possible for InCommon client certificates.

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

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

-- Obtaining proof of authorization, such as a letter of admission or a signed employment contract (or a completed key auth form)
-- Taking your paperwork and a drivers license or passport, and visiting the card office or key 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 university 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. Non-training is important!

-- 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.**

**Using client certs *will be* (almost) that easy *from a user perspective* if client certs are stored on smart cards or PKI hard tokens.**

# CAC/PIV Is A "Proof By Example" That Smart Cards 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.

# Granted, Smart Cards/Hard Tokens Aren't Perfect

- Some tokens/smart cards are still relatively expensive
- Limited capacity for storing credentials (72K?? *In 2012???*)
- Some tokens may not support Mac OS X/Linux/*BSD
- Some tokens or cards may require drivers to be installed
- Doing smart cards? You'll need to deploy card readers
- Some devices (particularly mobile devices) may not have a USB port (but so-called "CAC sleds" are available which use secure BlueTooth (but these aren't terribly cheap))
- Users may "lock themselves out" by repeatedly entering the wrong token PIN, requiring the token to be reset
- Users may forget physical credentials – what then? (The answer may sometimes be: "you need to go and get 'em")
- Users may lose or damage their smart card or hard token – sure hope that encryption key was escrowed, eh?

# How Easy Would Doing Hard Tokens/Smart Cards Be From the Point of View of Card Admins?

While smart cards/PKI hard tokens typically come with basic software administration tools, e.g., tools for loading a certificate or setting the user's PIN, those tools aren't really "enterprise grade." You'd go nutz trying to administer thousands of tokens using just the bundled software tools (although I'm sure at least some current users do precisely that today). Enterprise-grade token admin tools can make issuing and maintaining large numbers of tokens far easier. The downside of this approach? Cost: there's an additional cost for this capability (potentially more than the cost of the token!)

For maximum value, enterprise-grade PKI token administration tools should be tightly integrated with both the PKI token vendor, and the PKI certificate vendor.

# A Sometimes Overlooked Challenge To Deployment of PKI at Scale: Directories

- One of the tasks that the NPS trifold brochure spends time explaining is how to use the federal global address list (e.g., the Fed PKI directory). Intriguingly, directories are one of the biggest challenges to PKI deployment at national scale. Let me explain...

- Assume you want to send an encrypted email to a person you've never previously met. To do that, you need their public key.
- If you're using PGP/GNU PrivacyGuard:
  -- You can send an unencrypted email and ask that person for a copy of their key, or
  -- You can look for that person's key on a PGP public key server.
- If you're using S/MIME with client certs:
  -- you'd normally get your correspondent's public key from that user by sending them an unencrypted email asking them to send you an S/MIME signed message,
  -- or you'd get the key you need from a **directory.**

# 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 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!

# How About PGP-Style Keyservers for S/MIME?

- PGP public key servers seem to work okay for many PGP users. Should we be taking a lesson from PGP, and deploying PGP-style public key servers for S/MIME, too?
- Users who **want** to share their S/MIME creds can opt-in and submit them, and those who don't want to participate don't have to.
- There shouldn't be a problem with bogus or expired keys since all submitted certs can be verified at submission time, and automatically removed in periodic housekeeping sweeps when they expire.
- Resilience and scalability can be managed with mirroring
- At least initially, integrated automatic key retrieval would not necessarily need to be supported. [Supporting automatic key retrieval may raise subtle potential privacy risks associated with disclosure of correspondent identities (as well as receiver IP addresses) to third party key server administrators.]
- Internet2 is currently running a proof-of-concept/demo PGP-style keyserver for S/MIME credentials. Please contact me for details if you'd like to try it or you're interested in working on this project.

# One Closing Caution: 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.

# Thanks For The Chance To Talk Today!

- Are there any questions?