CS 5430

Certificates

Prof. Clarkson
Spring 2017

Review

- **Previously:** token-based authentication: authenticate a human based on their possession of a small machine

- **Today:**
  - Authentication of machine itself with *certificates*
  - Authentication of human through large machine with *PGP*
  - One of two solutions to the phonebook problem!

DIGITAL CERTIFICATES

Certificates

- Digital certificate is a signature binding together:
  - **identity** of principal
  - **public key** of that principal (might be encryption or verification key)
- **Notation:** Cert(S; I) is a certificate issued by principal I for principal S
  - let b = id_S, K_S
  - Cert(S; I) = b, Sign(b; k_I)
  - Issuer I is certifying that K_S belongs to subject id_S
- Fingerprint: H(Cert(S; I))

Authentication with certificate

```
1.  S: Let m = "I'm id_S".
       Compute s = Sign(m; k_S).
2.  S -> A: m, s
3.  A: Find Cert(S; I).
       Verify I's sig on cert.
       Verify id_S.
       Retrieve K_S.
       Accept if Ver(m; s; K_S).
```

Notes:

- m might contain more data to authenticate: “I’m id_S and I say …”
- I must be trusted to issue certificate
- A must verify id_S: common error to omit

X.509 certificates

[RFC 5280]

Contents of certificate:

- serial number (unique within certs issued by this issuer)
- issuer *distinguished name*
- validity interval (start and end time)
- subject *distinguished name*
- subject public key (and the name of the algorithm)
- extensions...
- issuer's signature on the above (and the name of the algorithm)

X.500 distinguished names

- Originally designed for general purpose directory services
- As commonly used in X.509 certificates:
  - **Common name (CN):** e.g., a person's full name, a server's name or domain name
  - **Organizational unit (OU):** e.g., Finance, HR, CS
  - (might be many nested OUs...)
  - **Organization (O):** e.g., Cornell, Google
  - Other fields: Street Address, Locality, State, Country, Postal Code, etc.

Certificate examples

- https://www.google.com
- https://www.cs.cornell.edu
- https://music.cornell.edu

Finding a useful certificate

Problem:

- You receive a message signed by A
- You don't know A's public verification key
- You manage to find a certificate Cert(A; B)
- But you don't know B's public key

Solution: recurse: find a certificate for B, etc.

e.g., you might end up with a set of certificates Cert(A; B), Cert(B; C), Cert(C; D), where you already know D's public key

Finding a useful certificate

Certificate chain: sequence of certificates that certify each other

- on one end, a certificate for the principal you want to authenticate
- on the other end, a certificate for a principal you already know: the *root* or *anchor* of trust
- you must trust every issuer in the chain to issue certificates

X.509 certificate extensions

- **Informational extensions:** extra information about certificate, issuer, subject
- **Constraint extensions:** warn user of certificate about what not to do with it
- **Critical flag:** if set, software must process extension or reject certificate

Some informational extensions

- **Key usage:**
  - digital signature
  - encryption of session keys
  - encryption of data
  - verification of certificates (i.e., issuer key)
  - (others)
- **Alternative name:** anything that doesn't fit in a distinguished name, e.g., email address, URL, IP address

A constraint extension

- **"Basic constraint":** two values:
  - a Boolean: is this key permitted to be used to verify other certificates? i.e., can it be an issuer's key?
    - At best redundant w.r.t key usage extension, which itself is more precise
  - an integer: number of intermediate certificates permitted to follow this one in a chain
  - ought to be marked critical

Public-key infrastructure (PKI)

- System for managing distribution of certificates
- Two main philosophies:
  - **Decentralized:** anarchy, no leaders
  - **Centralized:** oligarchy, leadership by a few elite

PKI Example 1: PGP

- Uses a decentralized PKI philosophy (at least early versions)
- "Pretty Good Privacy" [Zimmerman 1991]
  - toolset for encryption and signing of files and emails
  - Zimmerman investigated by US Customs for arms trafficking
  - PKI is just a part of PGP functionality
  - now you might use OpenPGP as implemented by GNU Privacy Guard (GPG)

PGP keyring

Users manage a keyring:

- Alice has her own key in her keyring
- When Alice meets up with Bob at a key-signing party...
  - (there's an XKCD for that)
  - She copies his key into her keyring
  - She marks Bob as *fully* or *marginally trusted* as an introducer
  - And she copies other keys he might have collected, too
- Instead she might download keys from a keyserver, but then she has little proof to whom they belong...

PGP keyring

- Entries on the keyring effectively are certificates
- Alice's own key on her keyring:
  - Cert(A; A)
  - a self-signed certificate
- When Alice imports a key signed by Bob, she gets a certificate Cert(C; B)
  - She can choose to import as-is and copy Cert(C; B) into her keyring
  - Or she can decide she's willing to vouch for it herself and put Cert(C; A) into her keyring
- Same goes for if she gets Cert(C; D) from Bob
- If she gets certificate from keyserver, she might phone Bob and confirm the fingerprint with him

PGP keyring

- Keys on keyring are fully valid only if
  - signed by 1 fully trusted introducer or 3 marginally trusted introducers
  - and the certificate chain leading from key to user's own key has length at most 5
  - (those constants could be user configurable)
- Valid keys may be used by rest of toolchain for encryption and signing

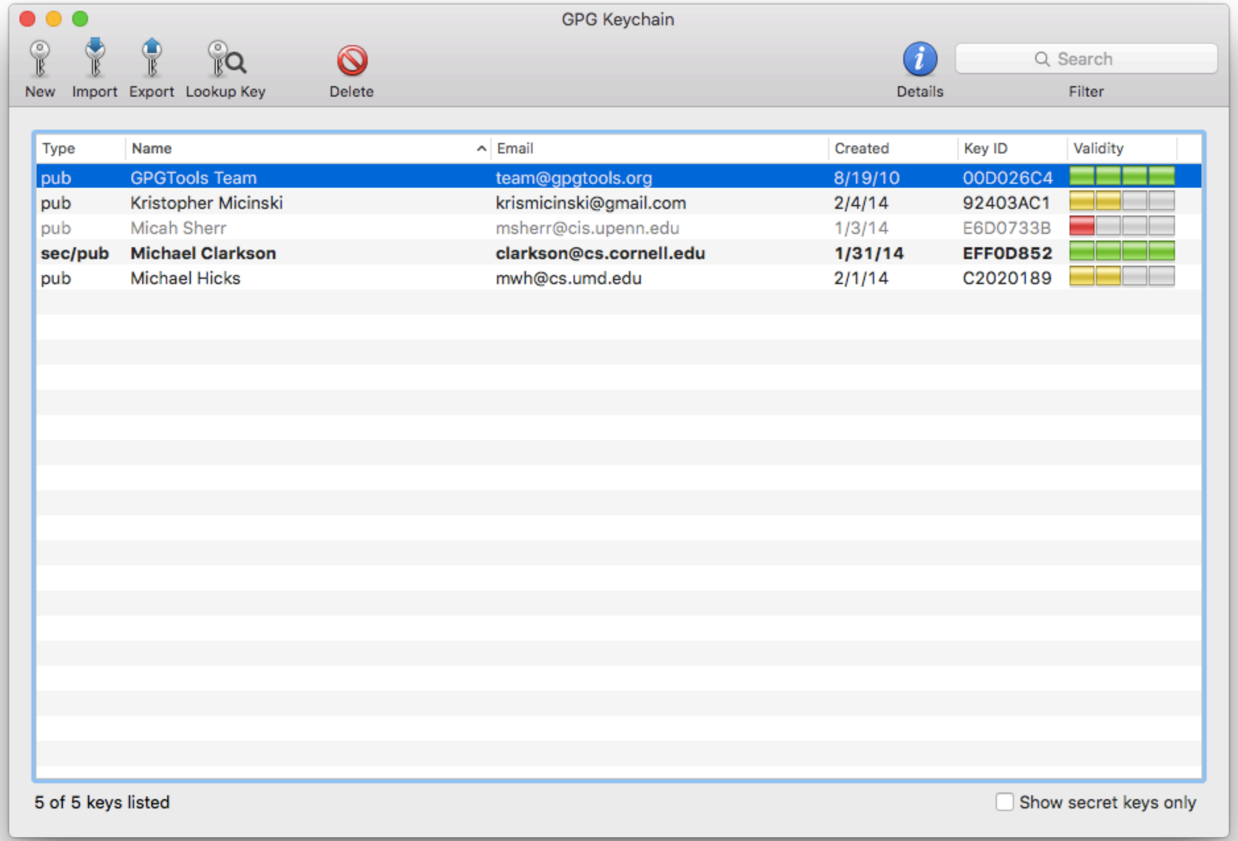
Web of trust

[Zimmerman 1992]:

As time goes on, you will accumulate keys from other people that you may want to designate as trusted introducers. Everyone else will each choose their own trusted introducers. And everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures. This will cause the emergence of a decentralized fault-tolerant web of confidence for all public keys.

Greek web of trust ca. 2012

Demo: GPG



Type	Name	Email	Created	Key ID	Validity
pub	GPGTools Team	team@gpgtools.org	8/19/10	00D026C4	🟩🟩🟩🟩
pub	Kristopher Micinski	krismicinski@gmail.com	2/4/14	92403AC1	🟨🟨⬜⬜
pub	Micah Sherr	msherr@cis.upenn.edu	1/3/14	E6D0733B	🟥⬜⬜⬜
sec/pub	**Michael Clarkson**	**clarkson@cs.cornell.edu**	**1/31/14**	**EFF0D852**	🟩🟩🟩🟩
pub	Michael Hicks	mwh@cs.umd.edu	2/1/14	C2020189	🟨🟨⬜⬜

5 of 5 keys listed

☐ Show secret keys only

Recap of PGP

PGP offers authentication of humans through machines:

- Identity is that of a human
- Private key is part of human's identity
- Private key is stored on trusted machine
- Need the machine to handle storage and computation
- So line is blurred between which we're really authenticating

Upcoming events

- [Fri] A4 due

There is a fine line between tolerant and oblivious. A lot of security software which is built around highly complex concepts like PKI works mostly because it's the latter. – Peter Gutmann