# Digital Certificates, Certification Authorities, and Public Key Infrastructure

Sections 14.3-14.5

# Basic Problem

- What does a public-key signature verification tell you?

  Verification parameters include public key, and successful verification says "*Only someone holding the corresponding private key could have made this signature*."

- What do you *want* a signature verification to tell you?

  Probably something like "*Joe Smith signed this*."

- Problem: What assurance do you have that the public key really belongs to Joe Smith?

# What is a Digital Certificate?

- Associates an identity/properties with a public key
  - Identity can be person’s name, website, e-mail, ...
  - Properties can be valid key uses, age of individual, access rights granted, …
- Signed by someone you trust
  - Signature is trusted party vouching for ID/key pair
  - Role is similar to a notary public
- Some typical properties of certificates:
  - Good for a set time (validity period)
    - Must get a new certificate after expiration
  - Certificates may be revoked

# More on Certificates

- Common types of certificates:
  - X.509 standard (version 3)
  - PGP certificates

- Who signs certificates?  Several possibilities:
  - Independent “Certification Authority” organization
    - Disinterested third party – company or government
    - Examples:  Verisign, Deutsche Telekom, Entrust, AOL, …
  - Internal (organizational) certification authority
    - Organization controls certificates for employees or clients
  - Could be just an individual you trust
    - This is how PGP certificates are typically certified

# X.509 Certificates

- Most prevalent type of digital certificate
- Related to X.500 directory services
- An integral part of the Web
  - All major web browsers and servers support X.509
  - CA “industry” (Verisign, etc.) built around X.509
- Also part of secure e-mail specifications
  - S/MIME
- Currently “version 3” of X.509
  - Includes a flexible “extension field” capability

# X.500 Names

(Also called “Distinguished Names”)

- Hierarchical naming
- Parts of names are attribute/value pairs
- Example attributes:
  - C=country
  - ST=state
  - L=locality
  - O=organization
  - OU=organizational unit
  - CN=common name

# Important "Additional Information"

- How does a CA state how they do business?
  - A Certification Practices Statement (CPS) is a human-readable statement of practices used by CA
    - Based on this, a person/vendor may decide whether to trust or not trust the CA
    - Problem:  What if CPS becomes a dead link?  Trust the CA?
- Where to obtain the Certification Revocation List (CRL)
  - Called a CRL Distribution Point (CDP)
  - Certificates may be revoked due to
    - Private key compromised
    - Incorrectly issued certificate
    - CA compromised
    - Properties change
  - CRL contains *unexpired* revoked certificates
    - Current  (2018) size of Symantec CRL:  1,211,730 bytes (34,610 entries)
  - Newer technology: OCSP (Online Certificate Status Protocol)

# Example:  Amazon Certificate

(Extension fields removed)

```
Data:
     Version: 3 (0x2)
     Serial Number:
     79:df:6e:64:52:f0:6a:12:05:ac:c8:80:7b:0a:d5:8e
Signature Algorithm: sha256WithRSAEncryption
     Issuer: C=US, O=Symantec Corporation, OU=Symantec Trust Network, CN=Symantec
Class 3 Secure Server CA - G4
     Validity
     Not Before: Oct  6 00:00:00 2017 GMT
     Not After : Sep 21 23:59:59 2018 GMT
     Subject: C=US, ST=Washington, L=Seattle, O=Amazon.com, Inc., CN=www.amazon.com
     Subject Public Key Info:
     Public Key Algorithm: rsaEncryption
          Public-Key: (2048 bit)
          Modulus:
               00:de:59:92:15:5c:f4:ae:8e:c4:ee:8e:ff:b3:97:
               ...   [ Deleted ]       ...
          Exponent: 65537 (0x10001)
...
Signature Algorithm: sha256WithRSAEncryption
     1f:01:57:8d:2f:fe:26:bb:5d:43:59:5a:86:42:47:47:2f:5e:
```

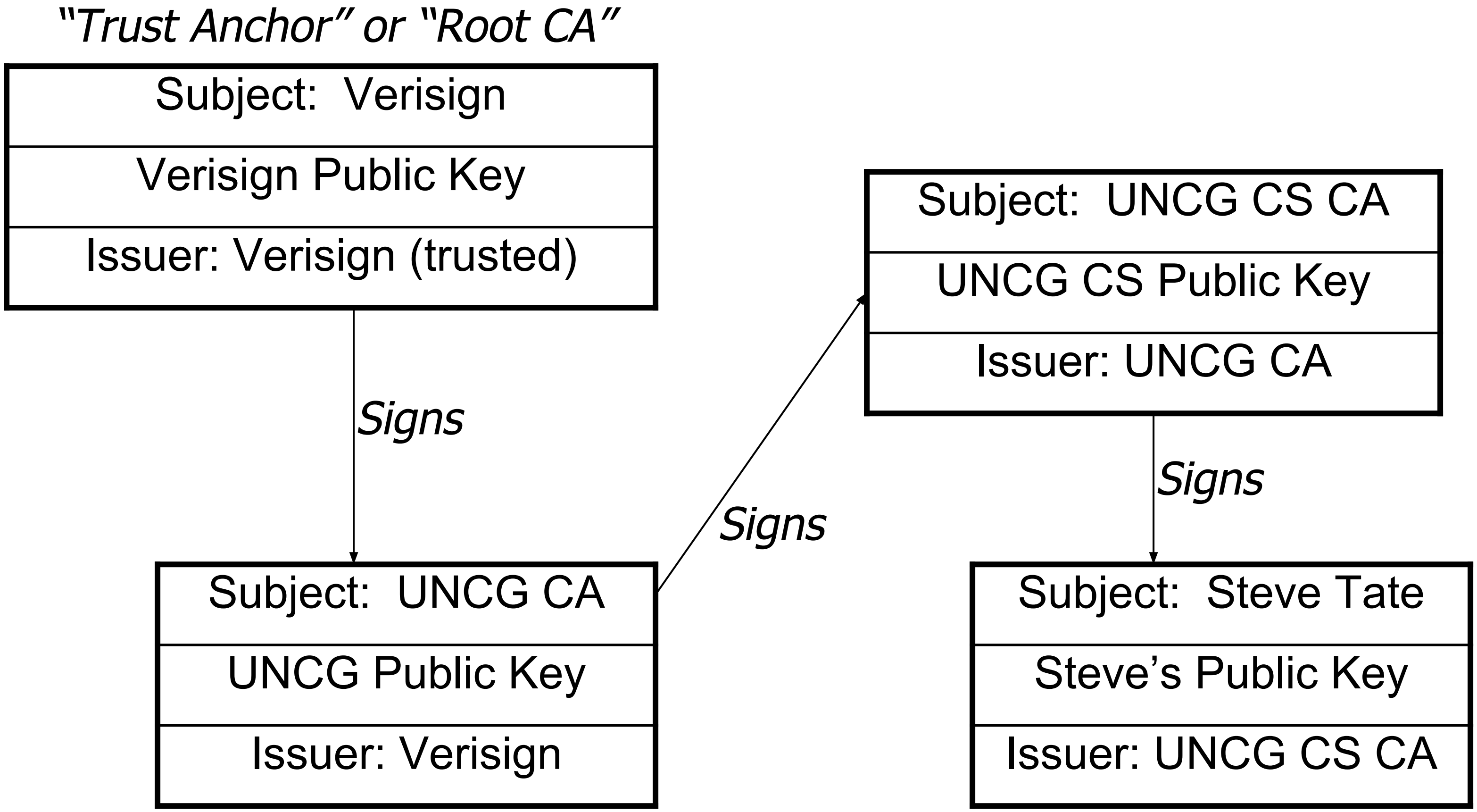
# Example:  Amazon Certificate, Part 2

## Extension fields

```
X509v3 extensions:
     X509v3 Subject Alternative Name:
     DNS:amazon.com, DNS:amzn.com, DNS:buybox.amazon.com, [ ... ]
     X509v3 Basic Constraints:
     CA:FALSE
     X509v3 Key Usage: critical
     Digital Signature, Key Encipherment
     X509v3 Extended Key Usage:
     TLS Web Server Authentication, TLS Web Client Authentication
     X509v3 Certificate Policies:
     Policy: 2.23.140.1.2.2
          CPS: https://d.symcb.com/cps
          User Notice:
          Explicit Text: https://d.symcb.com/rpa
     X509v3 Authority Key Identifier:
         keyid:5F:60:CF:61:90:55:DF:84:43:14:8A:60:2A:B2:F5:7A:F4:43:18:EF

     X509v3 CRL Distribution Points:
          Full Name:
          URI:http://ss.symcb.com/ss.crl

     Authority Information Access:
     OCSP - URI:http://ss.symcd.com
     CA Issuers - URI:http://ss.symcb.com/ss.crt
```

# Certificate Chains

(Hypothetical)



*"Trust Anchor" or "Root CA"*

Subject: Verisign

Verisign Public Key

Issuer: Verisign (trusted)

*Signs*

Subject: UNCG CA

UNCG Public Key

Issuer: Verisign

*Signs*

Subject: UNCG CS CA

UNCG CS Public Key

Issuer: UNCG CA

*Signs*

Subject: Steve Tate

Steve's Public Key

Issuer: UNCG CS CA

# Public Key Infrastructure (PKI)

- A PKI is “a collection of technologies and policies for creating and using digital certificates.” [Garfinkel and Spafford]

- Many people originally envisioned an official digital ID system
  - In reality: Very little personal ID done with certificates – mostly used for server identification
  - Could change if security tokens or smart cards become more prevalent! Maybe smartphones?

# Another Trust Model: PGP “Web of Trust”

- PGP is “Pretty Good Privacy”
  - Originally for e-mail encryption/signing
    - Now regularly used for software verification
  - Originally written by Phil Zimmerman
  - Now several free and commercial versions
  - GPG (“Gnu Privacy Guard”) is a Free-Software alternative (they use only free algorithms)

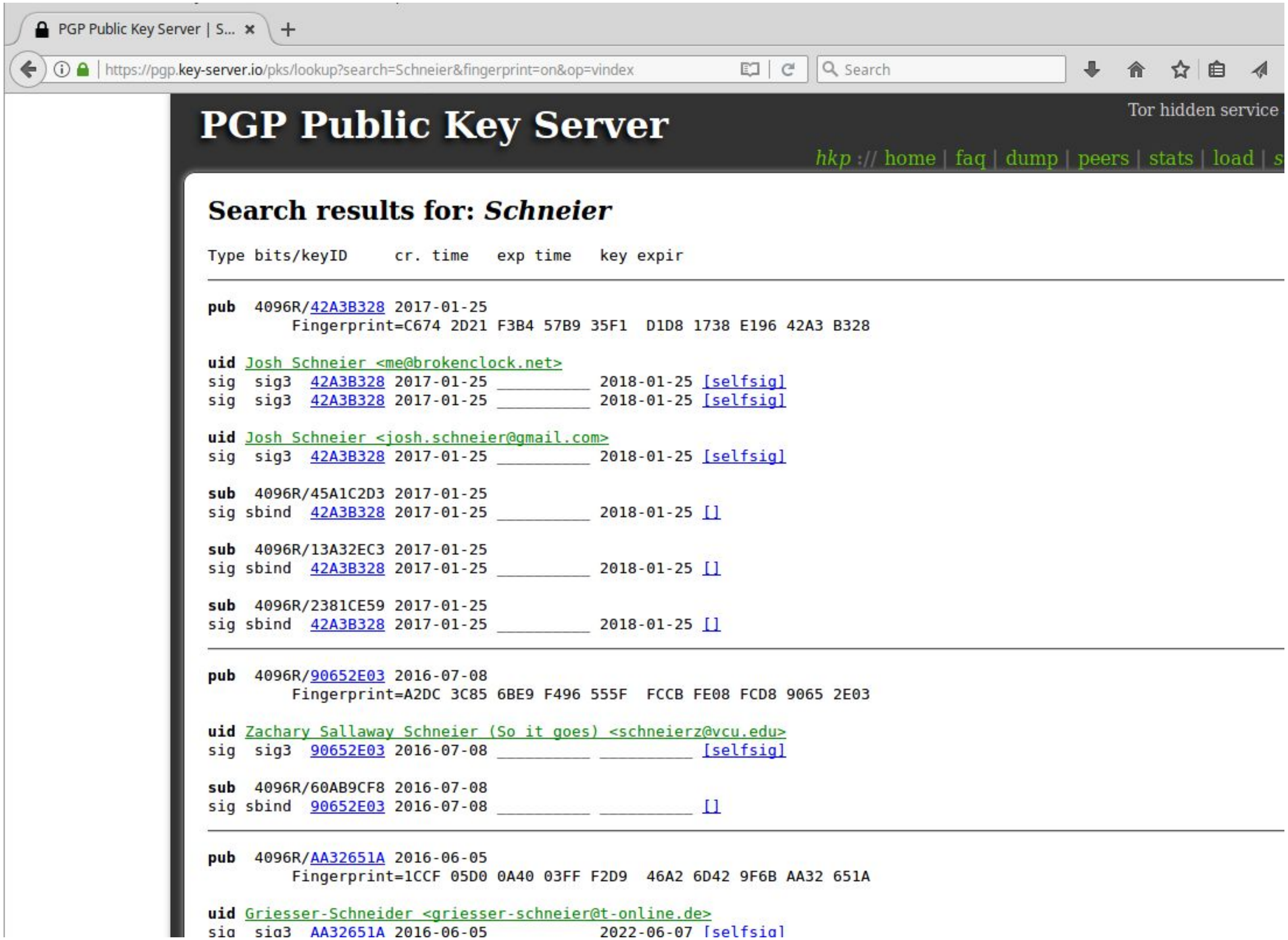
- Trust model is less hierarchical than X.509
- I can sign keys and distribute them
  - Anyone who trusts me can use me as a CA!
  - Difference between “trusted” and “valid” keys

# PGP/GPG Keyservers

- Problem:  How do you get public keys?
  - Note:  In PGP public keys are always certificates
- Solution:  Keyservers – databases of keys
  - You can submit your own keys
  - You can look up keys by name or e-mail address
  - Support integrated into many e-mail programs
- Keyservers can be accessed in many ways
  - LDAP
  - HTTP
  - E-mail

# Keyserver example – WWW interface

## Sending an encrypted email – Step 1: Look up the key


PGP Public Key Server | S...
https://pgp.key-server.io/pks/lookup?search=Schneier&fingerprint=on&op=vindex
Search
Tor hidden service
hkp :// home | faq | dump | peers | stats | load | s

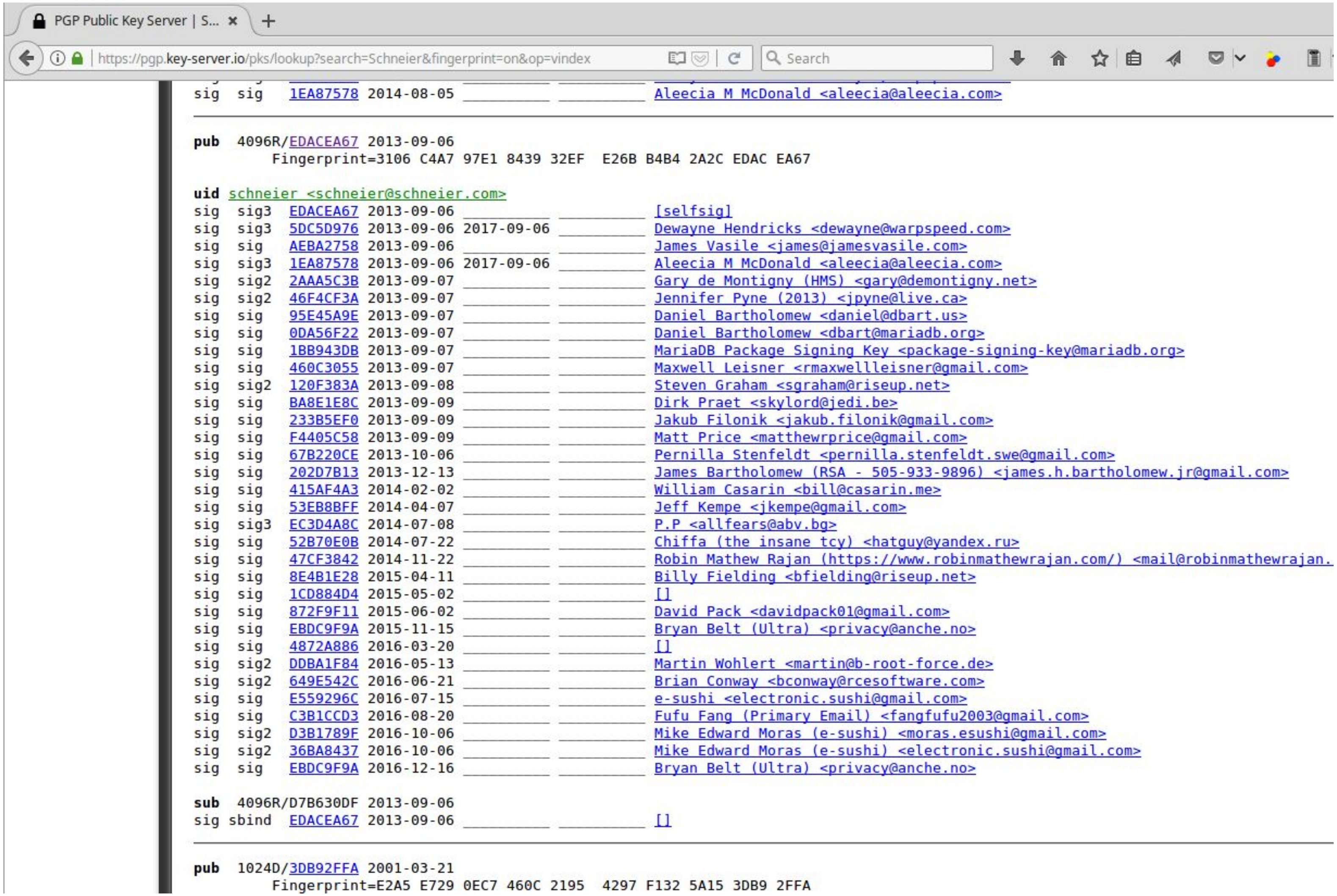
# PGP Public Key Server

## Search results for: *Schneier*

```
Type bits/keyID     cr. time   exp time   key expir

pub  4096R/42A3B328 2017-01-25
         Fingerprint=C674 2D21 F3B4 57B9 35F1  D1D8 1738 E196 42A3 B328

uid Josh Schneier <me@brokenclock.net>
sig  sig3  42A3B328 2017-01-25 __________ 2018-01-25 [selfsig]
sig  sig3  42A3B328 2017-01-25 __________ 2018-01-25 [selfsig]

uid Josh Schneier <josh.schneier@gmail.com>
sig  sig3  42A3B328 2017-01-25 __________ 2018-01-25 [selfsig]

sub  4096R/45A1C2D3 2017-01-25
sig sbind  42A3B328 2017-01-25 __________ 2018-01-25 []

sub  4096R/13A32EC3 2017-01-25
sig sbind  42A3B328 2017-01-25 __________ 2018-01-25 []

sub  4096R/2381CE59 2017-01-25
sig sbind  42A3B328 2017-01-25 __________ 2018-01-25 []

pub  4096R/90652E03 2016-07-08
         Fingerprint=A2DC 3C85 6BE9 F496 555F  FCCB FE08 FCD8 9065 2E03

uid Zachary Sallaway Schneier (So it goes) <schneierz@vcu.edu>
sig  sig3  90652E03 2016-07-08 __________ __________ [selfsig]

sub  4096R/60AB9CF8 2016-07-08
sig sbind  90652E03 2016-07-08 __________ __________ []

pub  4096R/AA32651A 2016-06-05
         Fingerprint=1CCF 05D0 0A40 03FF F2D9  46A2 6D42 9F6B AA32 651A

uid Griesser-Schneider <griesser-schneier@t-online.de>
sig  sig3  AA32651A 2016-06-05            2022-06-07 [selfsig]
```

# Keyserver example – WWW interface

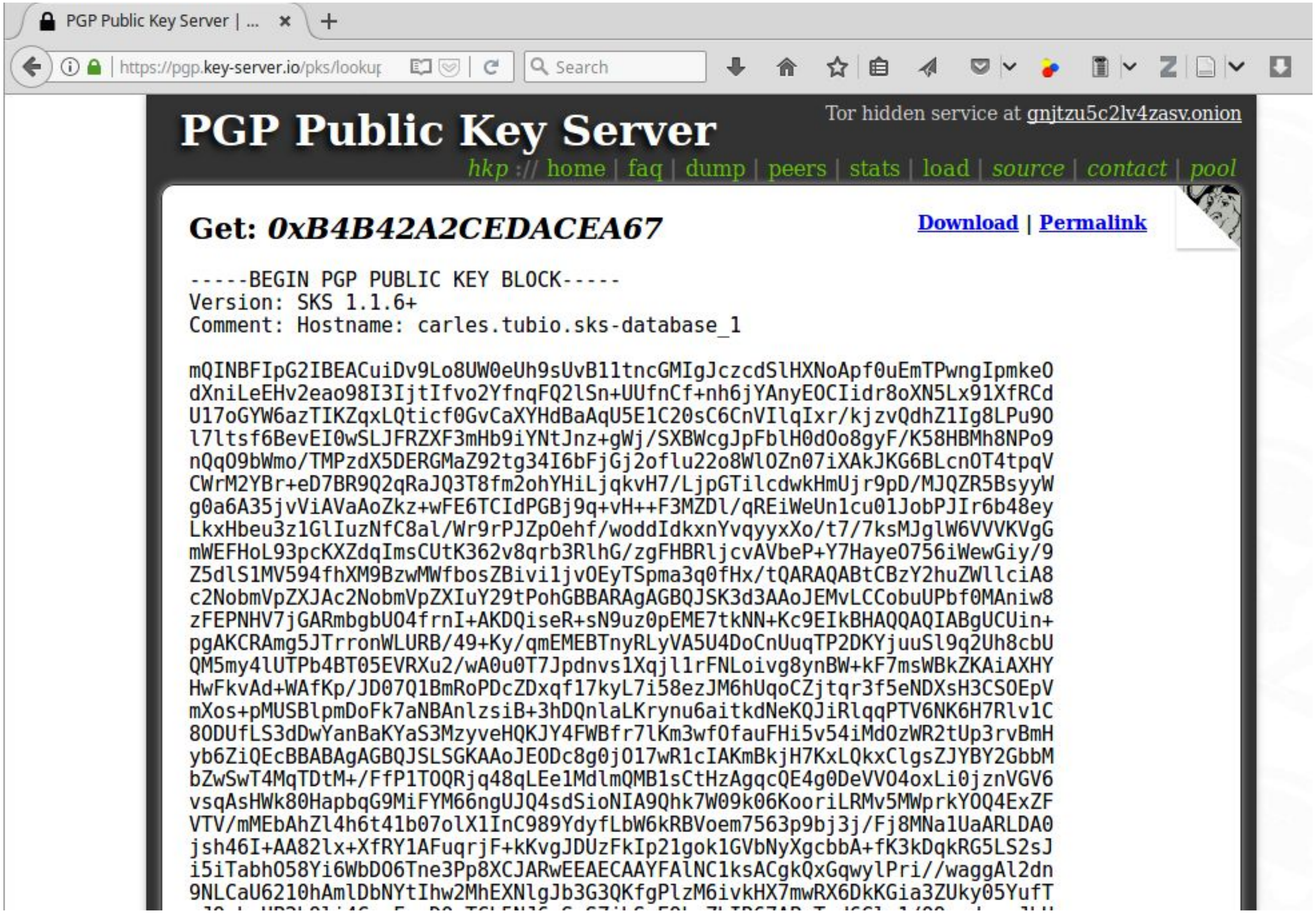
Sending an encrypted email – Step 2: Find the right one - who vouches for it?


PGP Public Key Server | S...
https://pgp.key-server.io/pks/lookup?search=Schneier&fingerprint=on&op=vindex
Search

```
sig  sig   1EA87578 2014-08-05 __________ __________ Aleecia M McDonald <aleecia@aleecia.com>

pub  4096R/EDACEA67 2013-09-06
         Fingerprint=3106 C4A7 97E1 8439 32EF  E26B B4B4 2A2C EDAC EA67

uid schneier <schneier@schneier.com>
sig  sig3  EDACEA67 2013-09-06 __________ __________ [selfsig]
sig  sig3  5DC5D976 2013-09-06 2017-09-06 __________ Dewayne Hendricks <dewayne@warpspeed.com>
sig  sig   AEBA2758 2013-09-06 __________ __________ James Vasile <james@jamesvasile.com>
sig  sig3  1EA87578 2013-09-06 2017-09-06 __________ Aleecia M McDonald <aleecia@aleecia.com>
sig  sig2  2AAA5C3B 2013-09-07 __________ __________ Gary de Montigny (HMS) <gary@demontigny.net>
sig  sig2  46F4CF3A 2013-09-07 __________ __________ Jennifer Pyne (2013) <jpyne@live.ca>
sig  sig   95E45A9E 2013-09-07 __________ __________ Daniel Bartholomew <daniel@dbart.us>
sig  sig   0DA56F22 2013-09-07 __________ __________ Daniel Bartholomew <dbart@mariadb.org>
sig  sig   1BB943DB 2013-09-07 __________ __________ MariaDB Package Signing Key <package-signing-key@mariadb.org>
sig  sig   460C3055 2013-09-07 __________ __________ Maxwell Leisner <rmaxwellleisner@gmail.com>
sig  sig2  120F383A 2013-09-08 __________ __________ Steven Graham <sgraham@riseup.net>
sig  sig   BA8E1E8C 2013-09-09 __________ __________ Dirk Praet <skylord@jedi.be>
sig  sig   233B5EF0 2013-09-09 __________ __________ Jakub Filonik <jakub.filonik@gmail.com>
sig  sig   F4405C58 2013-09-09 __________ __________ Matt Price <matthewrprice@gmail.com>
sig  sig   67B220CE 2013-10-06 __________ __________ Pernilla Stenfeldt <pernilla.stenfeldt.swe@gmail.com>
sig  sig   202D7B13 2013-12-13 __________ __________ James Bartholomew (RSA - 505-933-9896) <james.h.bartholomew.jr@gmail.com>
sig  sig   415AF4A3 2014-02-02 __________ __________ William Casarin <bill@casarin.me>
sig  sig   53EB8BFF 2014-04-07 __________ __________ Jeff Kempe <jkempe@gmail.com>
sig  sig3  EC3D4A8C 2014-07-08 __________ __________ P.P <allfears@abv.bg>
sig  sig   52B70E0B 2014-07-22 __________ __________ Chiffa (the insane tcy) <hatguy@yandex.ru>
sig  sig   47CF3842 2014-11-22 __________ __________ Robin Mathew Rajan (https://www.robinmathewrajan.com/) <mail@robinmathewrajan.
sig  sig   8E4B1E28 2015-04-11 __________ __________ Billy Fielding <bfielding@riseup.net>
sig  sig   1CD884D4 2015-05-02 __________ __________ []
sig  sig   872F9F11 2015-06-02 __________ __________ David Pack <davidpack01@gmail.com>
sig  sig   EBDC9F9A 2015-11-15 __________ __________ Bryan Belt (Ultra) <privacy@anche.no>
sig  sig   4872A886 2016-03-20 __________ __________ []
sig  sig2  DDBA1F84 2016-05-13 __________ __________ Martin Wohlert <martin@b-root-force.de>
sig  sig2  649E542C 2016-06-21 __________ __________ Brian Conway <bconway@rcesoftware.com>
sig  sig   E559296C 2016-07-15 __________ __________ e-sushi <electronic.sushi@gmail.com>
sig  sig   C3B1CCD3 2016-08-20 __________ __________ Fufu Fang (Primary Email) <fangfufu2003@gmail.com>
sig  sig2  D3B1789F 2016-10-06 __________ __________ Mike Edward Moras (e-sushi) <moras.esushi@gmail.com>
sig  sig2  36BA8437 2016-10-06 __________ __________ Mike Edward Moras (e-sushi) <electronic.sushi@gmail.com>
sig  sig   EBDC9F9A 2016-12-16 __________ __________ Bryan Belt (Ultra) <privacy@anche.no>

sub  4096R/D7B630DF 2013-09-06
sig sbind  EDACEA67 2013-09-06 __________ __________ []

pub  1024D/3DB92FFA 2001-03-21
         Fingerprint=E2A5 E729 0EC7 460C 2195  4297 F132 5A15 3DB9 2FFA
```

# Keyserver example – WWW interface

## Sending an encrypted email – Step 3: Download key (to import into PGP)


PGP Public Key Server | ...
https://pgp.key-server.io/pks/lookup
Search
hkp :// home | faq | dump | peers | stats | load | source | contact | pool

PGP Public Key Server

Tor hidden service at gnjtzu5c2lv4zasv.onion

### Get: *0xB4B42A2CEDACEA67*

Download | Permalink

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6+
Comment: Hostname: carles.tubio.sks-database_1

mQINBFIpG2IBEACuiDv9Lo8UW0eUh9sUvB11tncGMIgJczcdSlHXNoApf0uEmTPwngIpmkeO
dXniLeEHv2eao98I3IjtIfvo2YfnqFQ2lSn+UUfnCf+nh6jYAnyEOCIidr8oXN5Lx91XfRCd
U17oGYW6azTIKZqxLQticf0GvCaXYHdBaAqU5E1C20sC6CnVIlqIxr/kjzvQdhZ1Ig8LPu9O
l7ltsf6BevEI0wSLJFRZXF3mHb9iYNtJnz+gWj/SXBWcgJpFblH0dOo8gyF/K58HBMh8NPo9
nQqO9bWmo/TMPzdX5DERGMaZ92tg34I6bFjGj2oflu22o8WlOZn07iXAkJKG6BLcnOT4tpqV
CWrM2YBr+eD7BR9Q2qRaJQ3T8fm2ohYHiLjqkvH7/LjpGTilcdwkHmUjr9pD/MJQZR5BsyyW
g0a6A35jvViAVaAoZkz+wFE6TCIdPGBj9q+vH++F3MZDl/qREiWeUn1cu01JobPJIr6b48ey
LkxHbeu3z1GlIuzNfC8al/Wr9rPJZpOehf/woddIdkxnYvqyyxXo/t7/7ksMJglW6VVVKVgG
mWEFHoL93pcKXZdqImsCUtK362v8qrb3RlhG/zgFHBRljcvAVbeP+Y7HayeO756iWewGiy/9
Z5dlS1MV594fhXM9BzwMWfbosZBivi1jvOEyTSpma3q0fHx/tQARAQABtCBzY2huZWllciA8
c2NobmVpZXJAc2NobmVpZXIuY29tPohGBBARAgAGBQJSK3d3AAoJEMvLCCobuUPbf0MAniw8
zFEPNHV7jGARmbgbUO4frnI+AKDQiseR+sN9uz0pEME7tkNN+Kc9EIkBHAQQAQIABgUCUin+
pgAKCRAmg5JTrronWLURB/49+Ky/qmEMEBTnyRLyVA5U4DoCnUuqTP2DKYjuuSl9q2Uh8cbU
QM5my4lUTPb4BT05EVRXu2/wA0u0T7Jpdnvs1Xqjl1rFNLoivg8ynBW+kF7msWBkZKAiAXHY
HwFkvAd+WAfKp/JD07Q1BmRoPDcZDxqf17kyL7i58ezJM6hUqoCZjtqr3f5eNDXsH3CSOEpV
mXos+pMUSBlpmDoFk7aNBAnlzsiB+3hDQnlaLKrynu6aitkdNeKQJiRlqqPTV6NK6H7Rlv1C
8ODUfLS3dDwYanBaKYaS3MzyveHQKJY4FWBfr7lKm3wfOfauFHi5v54iMdOzWR2tUp3rvBmH
yb6ZiQEcBBABAgAGBQJSLSGKAAoJEODc8g0jO17wR1cIAKmBkjH7KxLQkxClgsZJYBY2GbbM
bZwSwT4MqTDtM+/FfP1TOQRjq48qLEe1MdlmQMB1sCtHzAgqcQE4g0DeVVO4oxLi0jznVGV6
vsqAsHWk80HapbqG9MiFYM66ngUJQ4sdSioNIA9Qhk7W09k06KooriLRMv5MWprkYOQ4ExZF
VTV/mMEbAhZl4h6t41b07olX1InC989YdyfLbW6kRBVoem7563p9bj3j/Fj8MNa1UaARLDA0
jsh46I+AA82lx+XfRY1AFuqrjF+kKvgJDUzFkIp21gok1GVbNyXgcbbA+fK3kDqkRG5LS2sJ
i5iTabhO58Yi6WbDO6Tne3Pp8XCJARwEEAECAAYFAlNC1ksACgkQxGqwylPri//waggAl2dn
9NLCaU6210hAmlDbNYtIhw2MhEXNlgJb3G3QKfgPlzM6ivkHX7mwRX6DkKGia3ZUky05YufT
```

# Some problems with certificates

- Private keys are not people
- Distinguished names are not people
- There are too many Robert Smiths
- X.509 v3 doesn’t allow selective disclosure
- Ubiquitous certificates could lead to privacy issues
- How do you loan a key?
- Signatures are “brittle”

- But overall:  Not perfect, but solves some important problems