COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Pretty Good Privacy (PGP)

ISTA 488: Information Privacy with Applications

David Sidi (dsidi@email.arizona.edu)

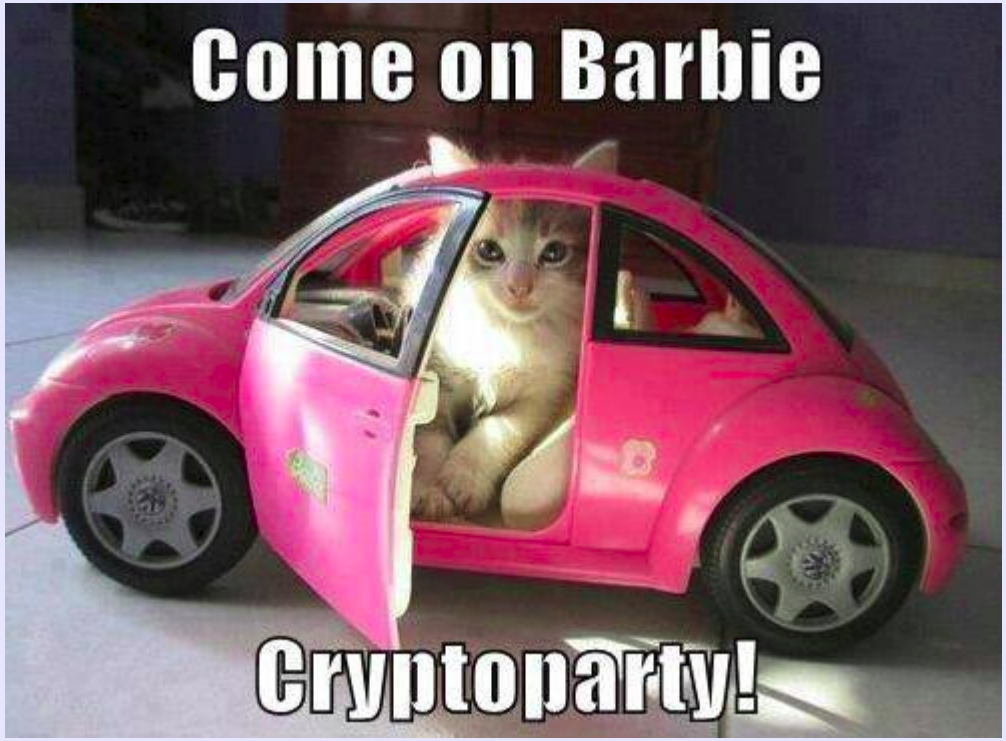
Warm-up

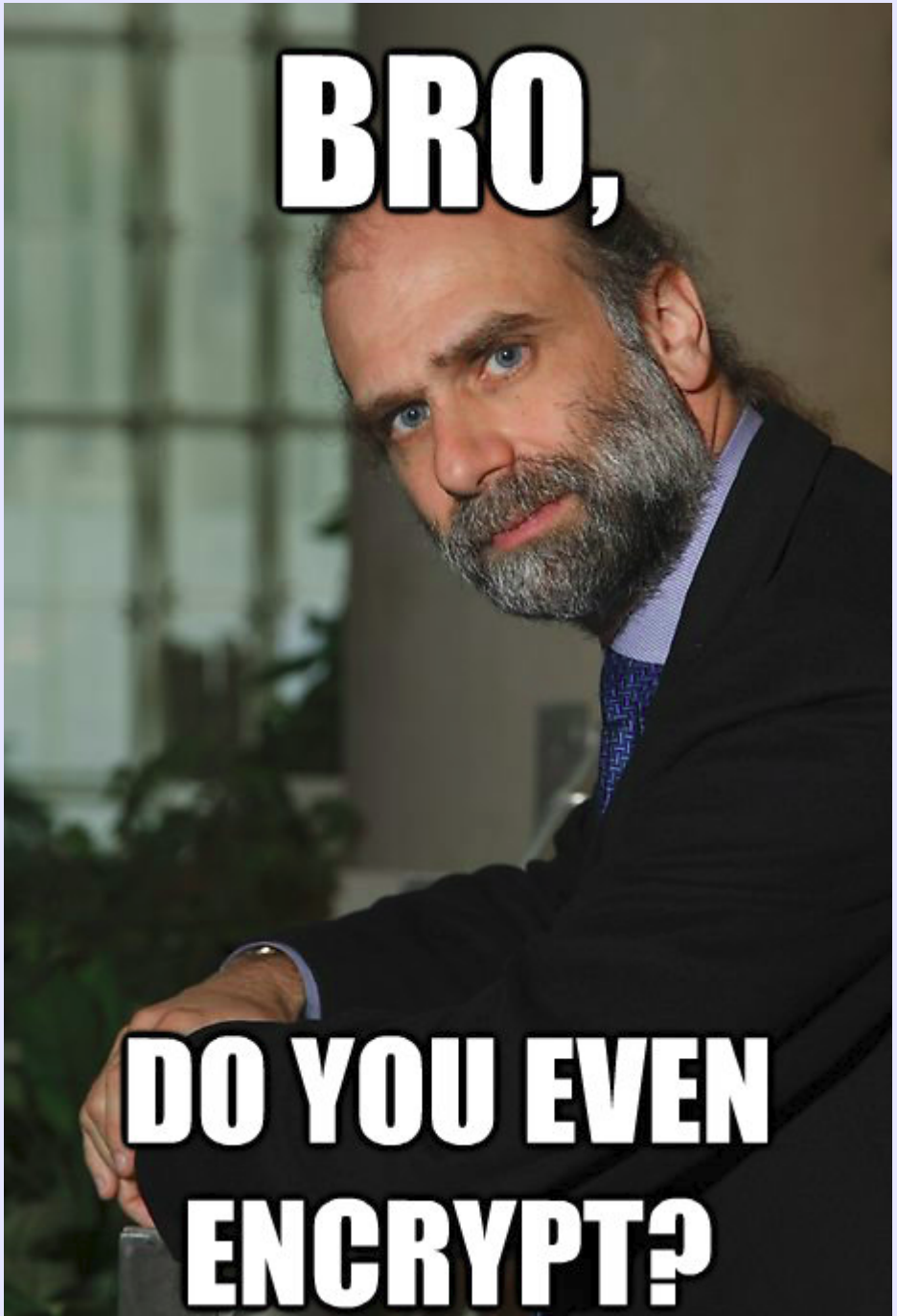
- Explain what an introducer is

Small mention of interesting things

- Cryptoparty extra-credit opportunity


COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information


Come on Barbie
Cryptoparty!


BRO,
DO YOU EVEN
ENCRYPT?

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

CRYPTO PARTY

It's going to be fun, shmucks! Become less boring!

Use Tor and I2P properly for anonymous browsing!

Set up your browser to prevent fingerprinting!

Set up a password manager you can trust!

Set up a VPN of your very own!

Set up secure messaging!

Set up secure email!

Eat pizza!

TIME: 24 Nov 2017
From 6:30 PM to some later hour

PLACE: Xerocraft
101 W 6th St, Suite 101 Tucson, AZ

All our welcome. Email david@sidiprojects.us if you want to help out, otherwise just come!

Small mention of interesting things

- Cryptoparty extra-credit opportunity
- More on assignment 2: how to disable password authentication, and use ssh keys instead, “for great good!” link (see “Client Authentication” section)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

From last time: Decoy-based encryption

Decoy-based cryptosystem

"there are secure encryption protocols that do not employ any one-way functions, but instead rely in their security on numerous 'decoys' of the actual encrypted message, and this 'decoy-based' cryptography presents an important alternative to the 'traditional', complexity-based, cryptography."

(Grigoriev and Shpilrain 2)

- To keep in mind: How will this compare Chaffing and Winnowing?

Decoy-based cryptosystem

"the general idea of decoy ...[is] combining private keys of Alice and Bob during transmission."

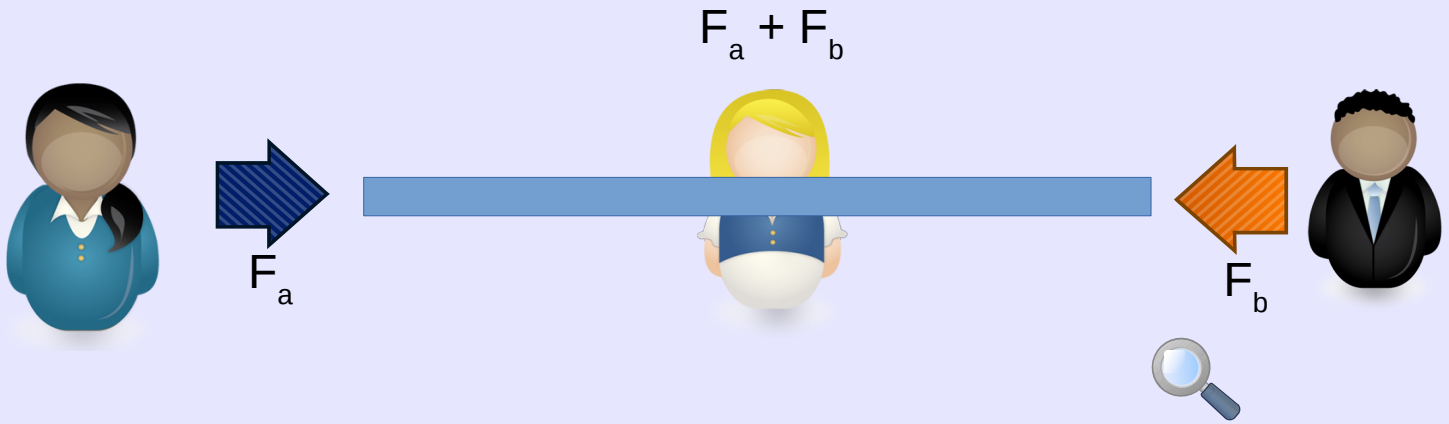
(Grigoriev and Shpilrain 2)

- To keep in mind: how does this compare to DH key agreement?

Requirements of Decoy-based Cryptography

- Requirement: a “private space,” analogous to the private computer that generates keys unobserved
- This is at the end point; it is not a reversion to requiring a secure channel
- In “public” everything is observable

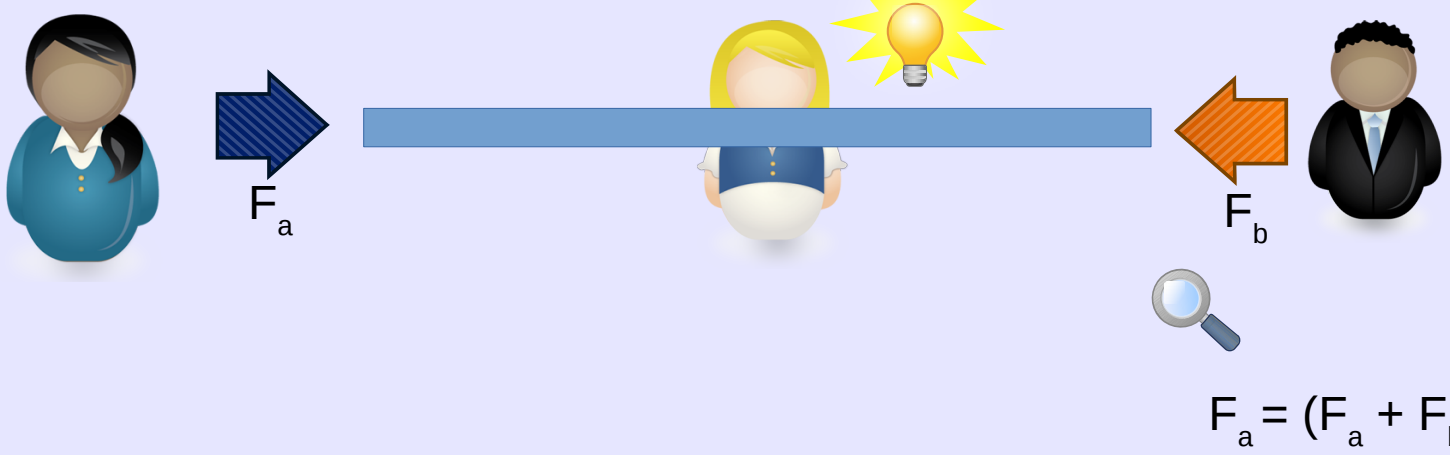
A decoy implementation that resembles “Physical DH”



$F_a = (F_a + F_b) - F_b$

F_a is the shared secret

A decoy implementation that resembles “Physical DH”



F_a is the shared secret

Secure multiparty communication (SMC)

- Up to now, we have been keeping messages private from a third party, but the communicating parties trust one another
- What if you don’t trust who you’re communicating with, but you need to get something done together?
  - What if you want to compute something with inputs from you and another party, but keep your input to the computation hidden, and not rely on any trusted third party?

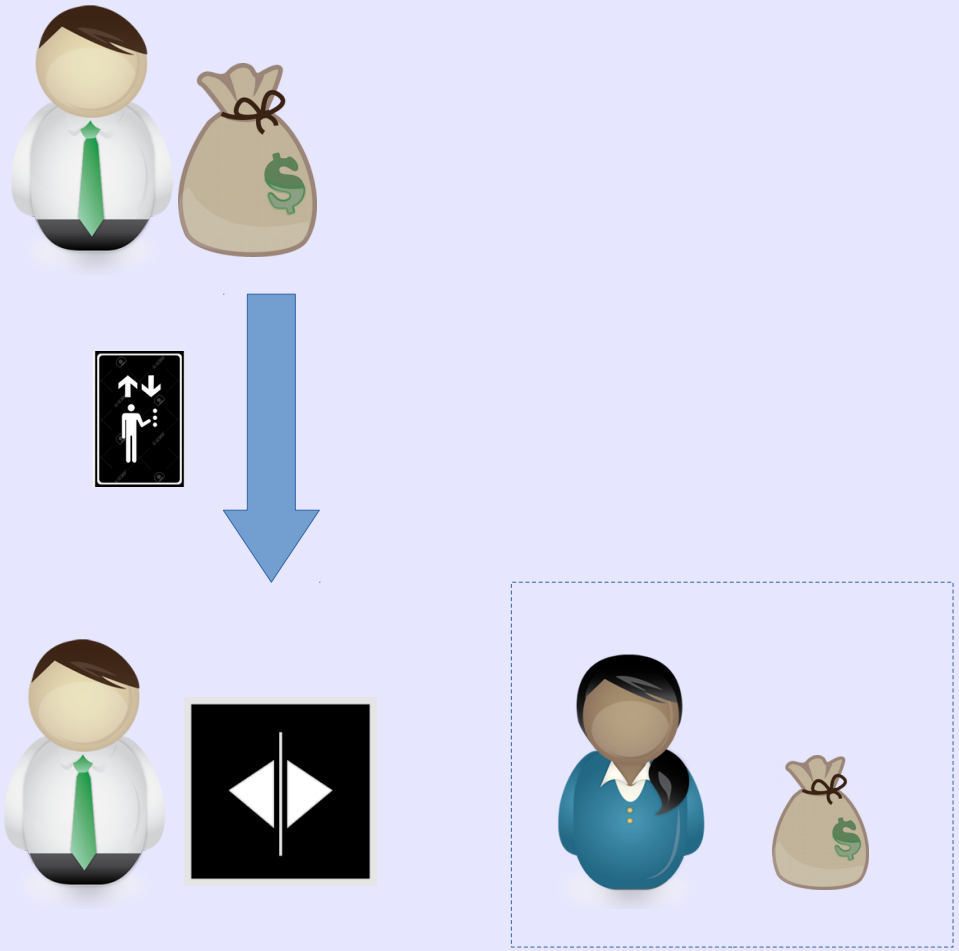
An SMC problem: Yao's millionaires

- Two millionaires have net worths N_1 and N_2
- They are interested in the difference in their net worths, but neither wants to share their particular net worth
- Correctness: we want to compute whether $N_1 > N_2$
- Privacy: we do not want party 2 to learn anything more about N_1 from following the protocol than they would by simply learning the ordering

A decoy implementation that resembles “Physical SMC”

- Requirement: a “private space,” analogous to the private computer that generates keys unobserved
- This is at the end point; it is not a reversion to requiring a secure channel
- In “public” everything is observable

Simple decoy implementation



COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Asynchronous communication privacy with Pretty Good Privacy (PGP)

PGP has several uses

- Encrypting / Decrypting files
  - Especially when you don’t own your mail server, which is common

Highlights from the Email Policy Document

- “Theft or unauthorized destruction, mutilation, defacement, alteration, falsification, removal or secretion of e-mail records may lead to class 4 or class 6 felony charges under A.R.S. § 38-421.”
- “E-mail users shall not give the impression that they are representing, giving opinions, or otherwise making statements of behalf of the University or any unit of the University unless expressly authorized to do so. Where appropriate, the following explicit disclaimer shall be included: "The opinions or statements expressed herein are my own and should not be taken as a position, opinion, or endorsement of the University of Arizona."

Highlights from the Email Policy Document

- The confidentiality of e-mail cannot be assured, and any confidentiality may be compromised by access consistent with applicable law or policy, including this Policy, by unintended redistribution, or due to current technologies inadequate to protect against unauthorized access. Users, therefore, should exercise extreme caution in using e-mail to communicate confidential or sensitive matters, and should not assume that their e-mail is private or confidential.
- Users may not access, use, or disclose personal or confidential information without appropriate authorization, and must take necessary precautions to protect confidentiality of personal or confidential information, regardless whether the information is maintained on paper or whether it is found in e-mail or other electronic records.

Highlights from the Email Policy Document

- Both the nature of e-mail and the public character of the University's business make e-mail less private than users may anticipate. For example, e-mail intended for one person sometimes may be widely distributed because of the ease with which recipients can forward it to others. A reply to an e-mail message posted on an electronic bulletin board or "listserver" intended only for the originator of the message may be distributed to all subscribers to the listserv. Furthermore, even after a user deletes an e-mail record from a computer or e-mail account it may persist in whole or in part in system logs, in the directories of the person who received the message, or on system back-up tapes which may be retained for long periods of time. All these items may be subject to disclosure under applicable law and this Policy. The University cannot routinely protect users against such eventualities.

Highlights from the Email Policy Document

“Encryption of e-mail is another emerging technology that is not in widespread use as of the date of this Policy. This technology permits the encoding of e-mail so that for all practical purposes it cannot be read by anyone who does not possess the right key. Because of Federal regulations (36 CFR 1234) and State of Arizona directives for the maintenance of e-mail public records, **encryption should not be used for storage of University e-mail.**”

“Even though an e-mail sender and recipient have deleted their e-mail, back-up copies may exist for periods of time and in locations unknown to the originator or recipient. These copies may be accessed or disclosed consistent with applicable policy or law.”

PGP has several uses

- Encrypting / Decrypting files
  - Especially when you don’t own your mail server:

PGP has several uses

- Encrypting / Decrypting files
- Signing / Verifying authenticity of files
  - Malware resistance
    - Melissa, ILOVEYOU viruses
  - Compulsion resistance
    - Suppose someone wants to force you to retroactively change a document you wrote that they don't like. With hashing In combination with clever use of bitcoin you can prove that you published the original document before a given time
    - take the SHA-256 hash, convert to a public key, and send some tiny amount of Bitcoin to the address associated with that key

Today many use PGP, but few do it correctly

- Fashionable to be down on PGP:
  - “Why Johnny can’t encrypt”
  - GPG and Me
  - What’s the matter with PGP
  - Giving up on GPG
- PGP is useful, though (see earlier): it has many uses.
- A power tool: You do need to know how to use it, but it’s not hard if you exercise some care

Hands-on: GPG

- GPG is the Gnu Privacy Guard, a FLOSS version of PGP

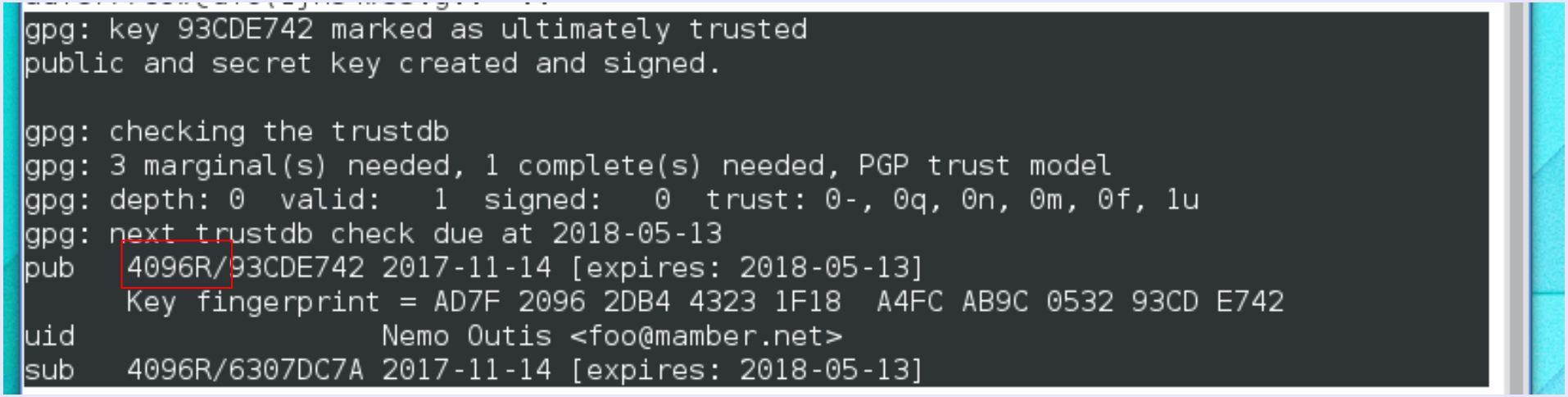
Hands-on: Generate a key

- Must be 4096, with SHA-2 rather than SHA-1

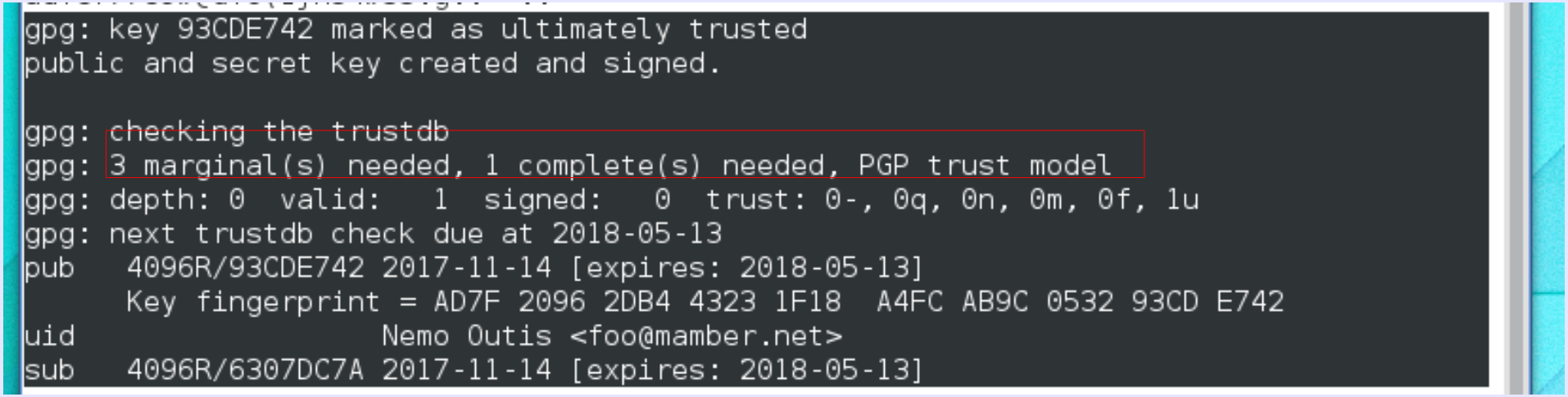
```
$ mkdir -p ~/.gnupg
$ cat >> ~/.gnupg/gpg.conf <<EOF
personal-digest-preferences SHA256
> cert-digest-algo SHA256
> default-preference-list SHA512 SHA384 SHA256 SHA224 AES256 AES192 AES CAST5 ZLIB BZIP2 ZIP Uncompressed
> EOF
```

Hands-on: Generate a key

$ gpg --gen-key



Hands-on: Generate a key

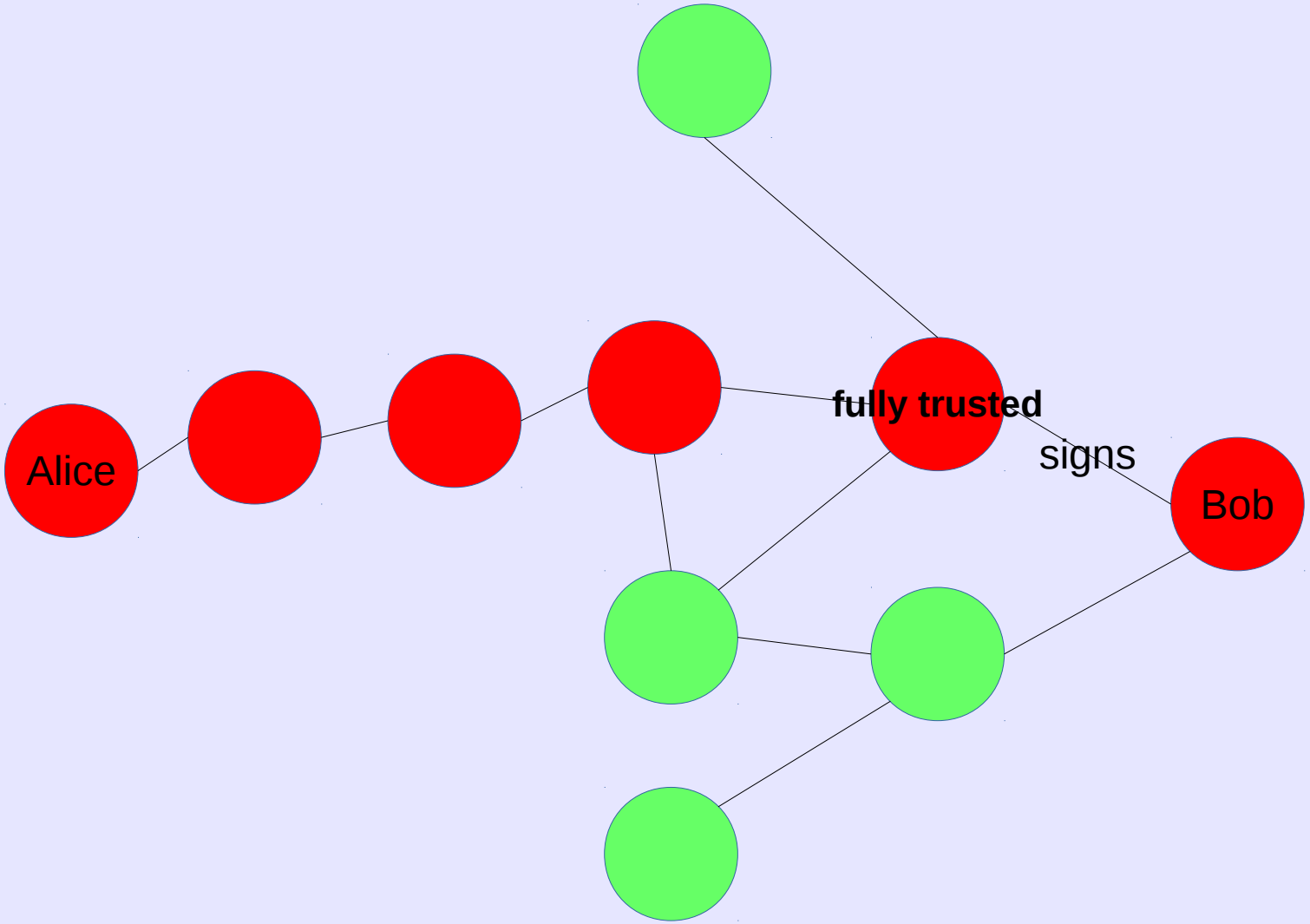


What is a fully valid key?

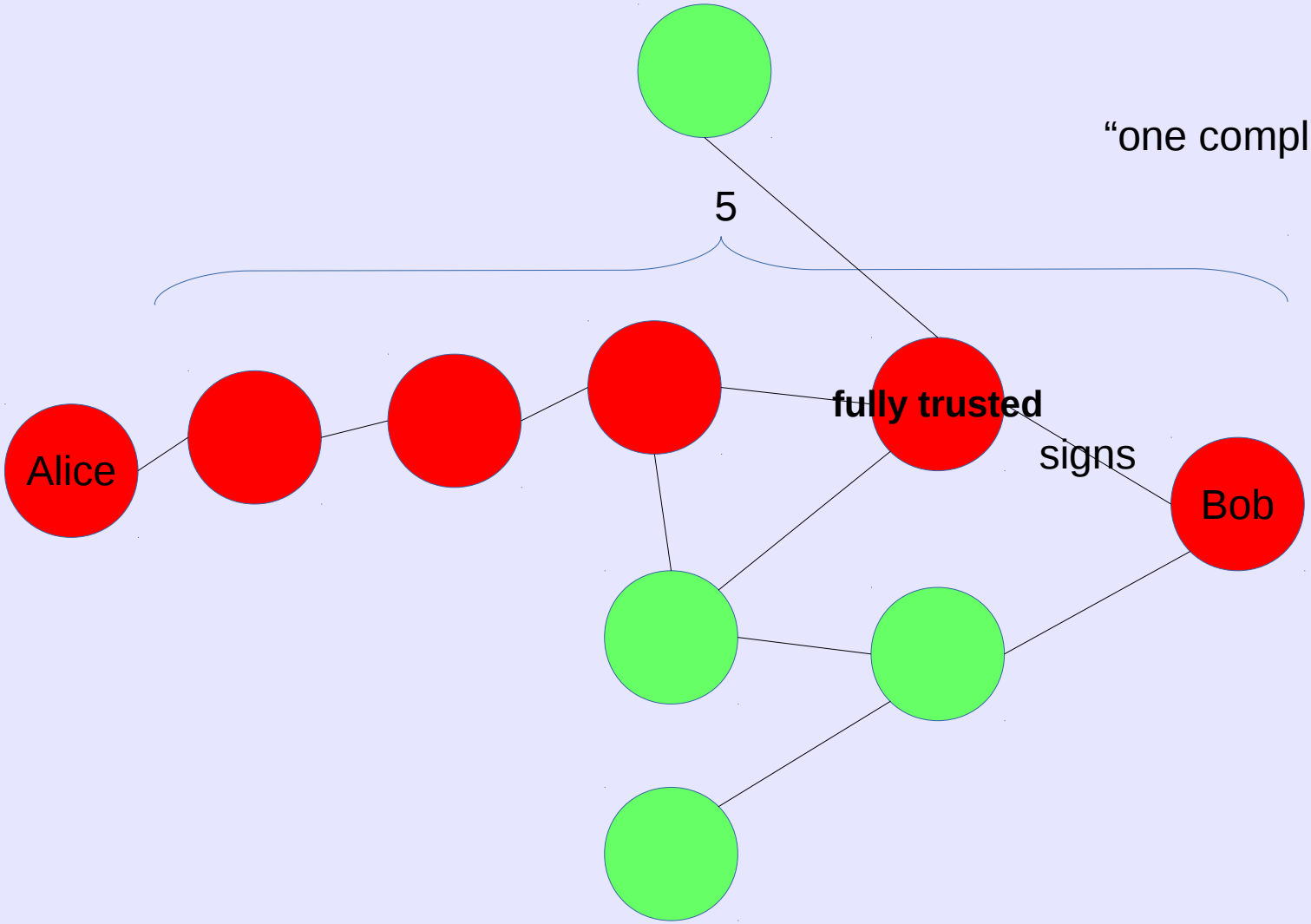
Excursus: Introducers vs. signed keys

- Essential to the PGP web of trust
- An **introducer** is trusted to some extent to verify other keys
- Extent has two dimensions: **full** or **marginal**, and **depth**
- A signed key is one that has been verified

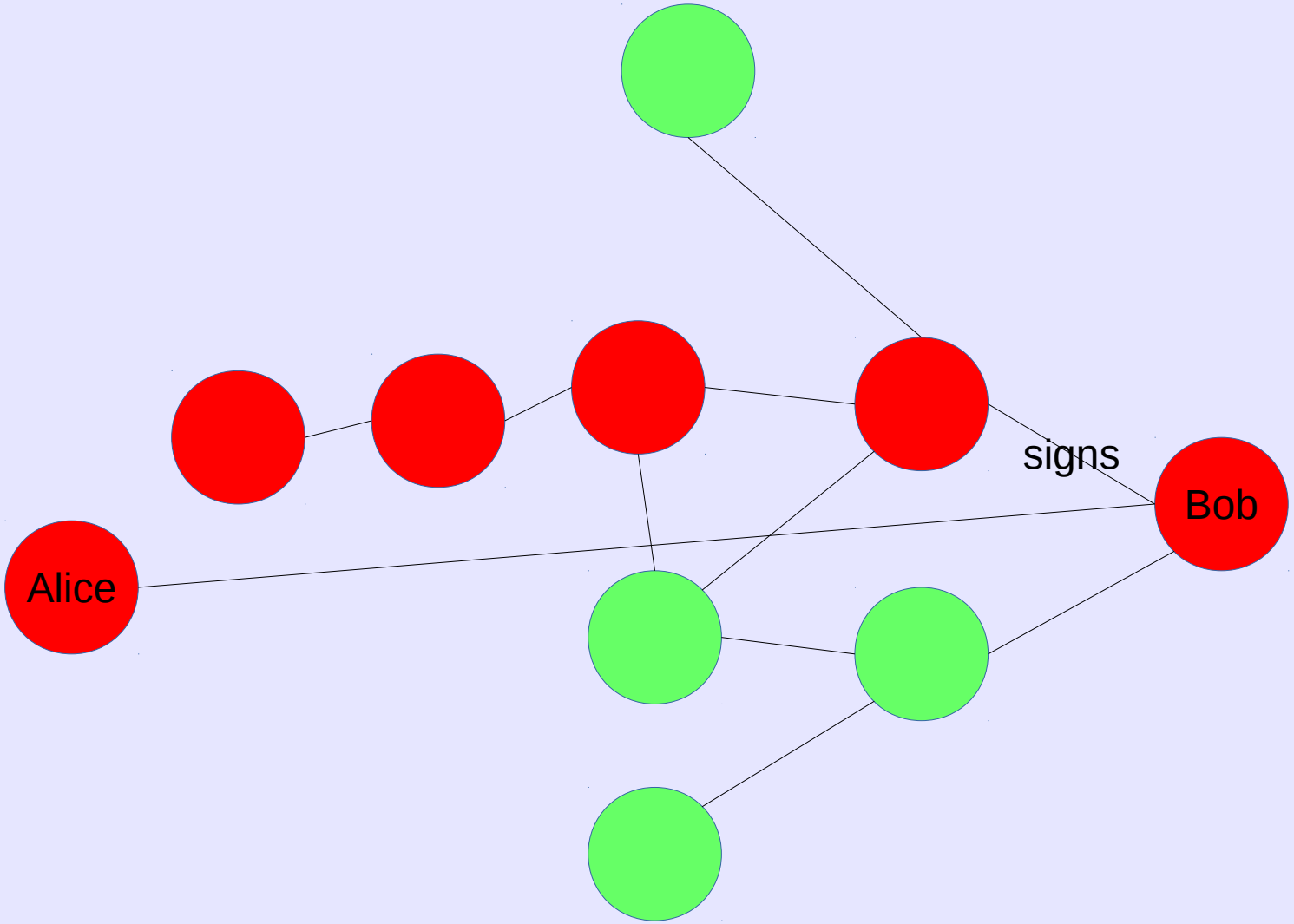
A fully valid key is signed with valid keys, and has a short signed path



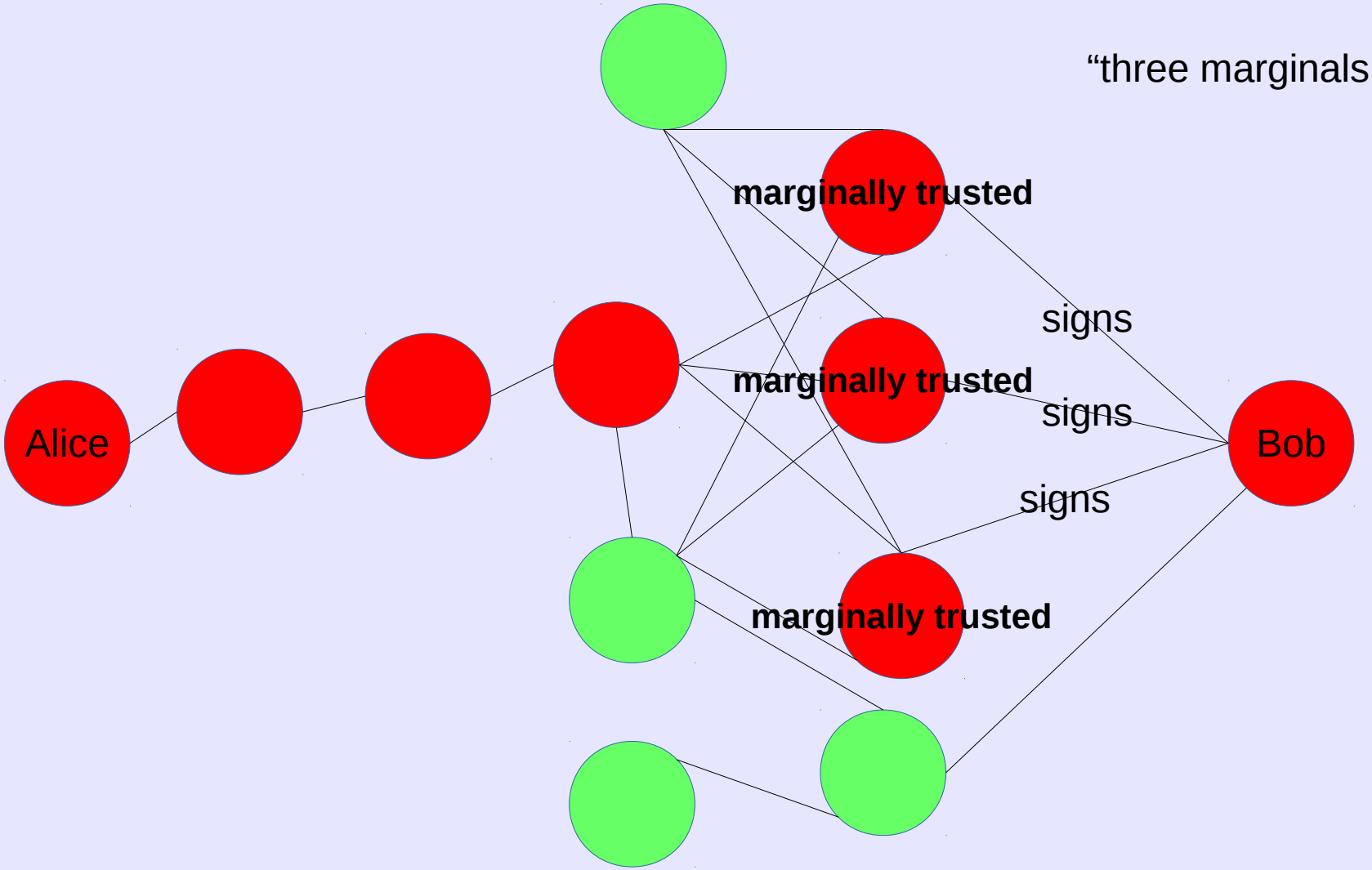
A fully valid key is signed with valid keys, and has a short signed path



A fully valid key is signed with valid keys, and has a short signed path



A fully valid key is signed with valid keys, and has a short signed path



Trust models

--trust-model pgp|classic|direct|always|auto
Set what trust model GnuPG should follow. The models are:

pgp This is the Web of Trust combined with trust signatures as used in PGP 5.x and later. This is the default trust model when creating a new trust database.

classic
This is the standard Web of Trust as introduced by PGP 2.

direct Key validity is set directly by the user and not calculated via the Web of Trust.

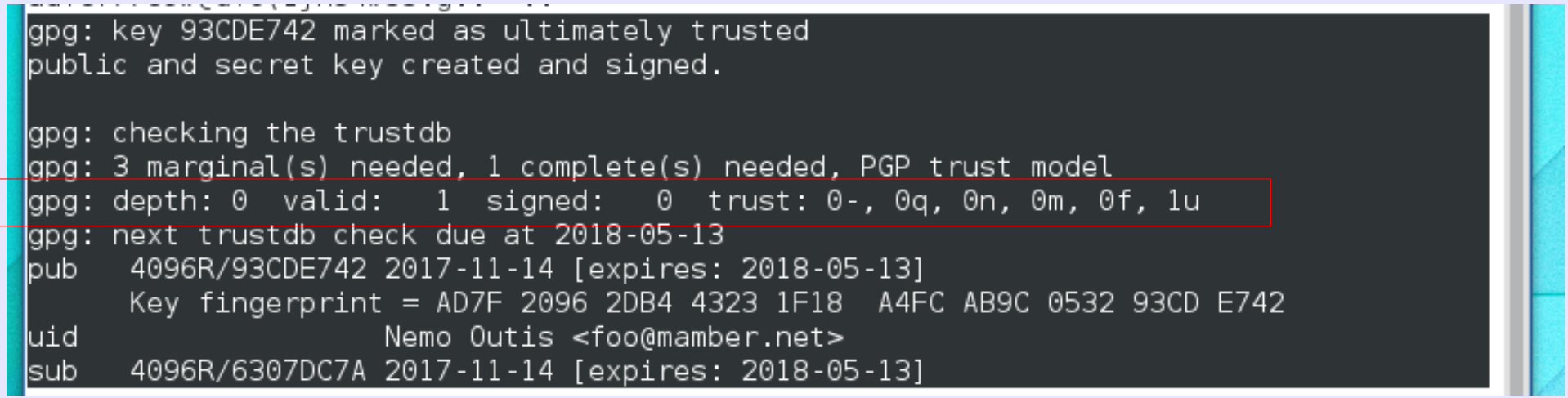
always Skip key validation and assume that used keys are always fully valid. You generally won't use this unless you are using some external validation scheme. This option also suppresses the "[uncertain]" tag printed with signature checks when there is no evidence that the user ID is bound to the key. Note that this trust model still does not allow the use of expired, revoked, or disabled keys.

auto Select the trust model depending on whatever the internal trust database says. This is the default model if such a database already exists.

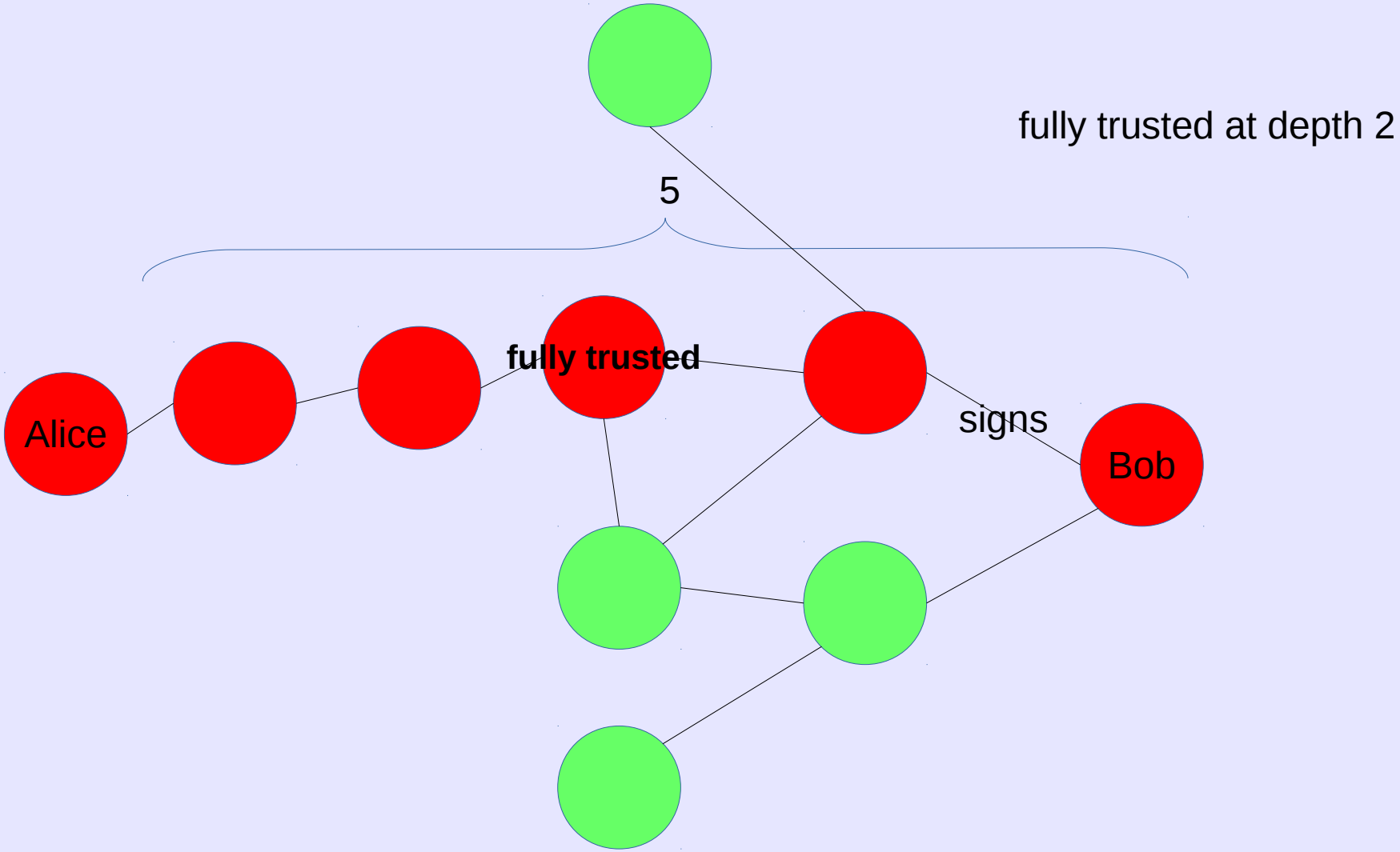
```
gpg: key 93CDE742 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2018-05-13
pub   4096R/93CDE742 2017-11-14 [expires: 2018-05-13]
      Key fingerprint = AD7F 2096 2DB4 4323 1F18  A4FC AB9C 0532 93CD E742
uid                  Nemo Outis <foo@mamber.net>
sub   4096R/6307DC7A 2017-11-14 [expires: 2018-05-13]
```

Hands-on: Generate a key

- An **introducer** is trusted to some extent to verify other keys
- Extent has two dimensions: **full** or **marginal**, and **depth**



A fully valid key is signed with valid keys, and has a short signed path



Hands on: sign my key

- Verify with government-issued identification
- `$ gpg --recv-key EEBA8245`
- `$ gpg --edit-key david@sidiprojects.us`

  (Prompt changes from ‘$’ to ‘gpg>’)
- `gpg>sign`
- or

  `gpg>tsign`

Hands on: Locally sign a key

```
$ gpg --edit-key <KEY-ID>
gpg>lsign
```

- Question: why do this instead of sign?

```
gpg> tsign
Really sign all user IDs? (y/N) y

pub  4096R/EEBA8245  created: 2014-03-14  expires: 2018-03-14  usage: SC
                     trust: unknown       validity: unknown
 Primary key fingerprint: E622 43FC 0F47 135B 28F0  02C4 8496 9123 EEBA 8245

     David Sidi <david@sidiprojects.us>
     David Sidi <davidsidi@gmail.com>
     David Sidi <dsidi@email.arizona.edu>

This key is due to expire on 2018-03-14.
Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

  1 = I trust marginally
  2 = I trust fully

Your selection? 2

Please enter the depth of this trust signature.
A depth greater than 1 allows the key you are signing to make
trust signatures on your behalf.

Your selection? 1

Please enter a domain to restrict this signature, or enter for none.

Your selection?

Are you sure that you want to sign this key with your
key "Nemo Outis <foo@mamber.net>" (93CDE742)

Really sign? (y/N) y

You need a passphrase to unlock the secret key for
user: "Nemo Outis <foo@mamber.net>"
4096-bit RSA key, ID 93CDE742, created 2017-11-14
```