PGP Key Signing: What, Why, and How.

Matt Pounsett, Nimbus Operations Inc.

OARC 29, Amsterdam

I Am Not an Expert

- Frequent user of cryptography for operational communications

- Understand the process of how the crypto works, and how to manage it safely

- Don’t ask me about the details of the math

What is PGP?

- OpenPGP is a standard for encryption
- Originally derived from the Pretty Good Privacy (PGP) software, created by Phil Zimmerman
- Currently maintained by the OpenPGP Working Group of the IETF
- Programs like PGP and Gnu Privacy Guard (GPG) implement the OpenPGP standard

What Does PGP Do?

- Provides privacy, using encryption

- Provides authentication and nonrepudiation, using cryptographic signatures

Encryption

(Plain Text + Key) → Process = Cipher Text

(Cipher Text + Key) → Process = Plain Text

Signatures

(Plain Text + Key) → Process = Signature

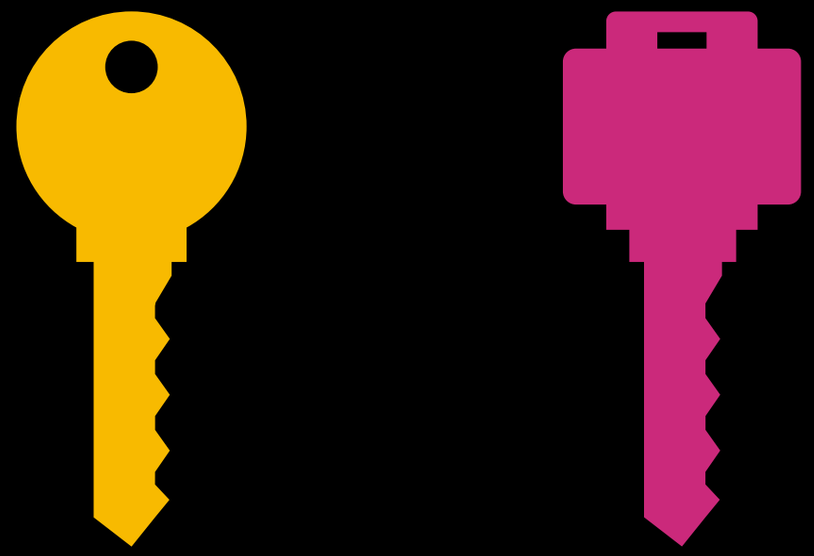
(Plain Text + Signature + Key) → Process = Result

Common Use Cases

- Private email
- Trusted email
- Software distribution

Quick Asymmetrical Cryptography Primer

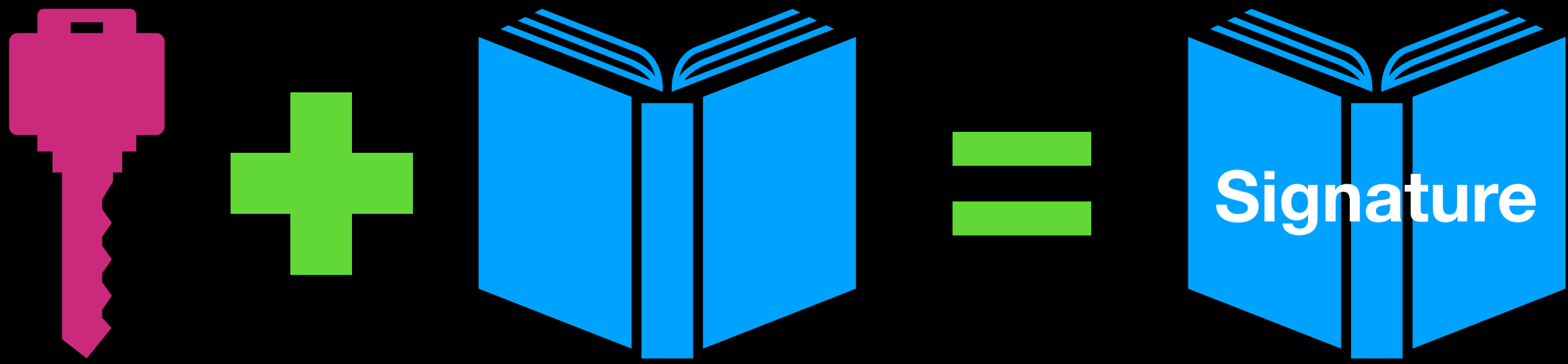
The Parts



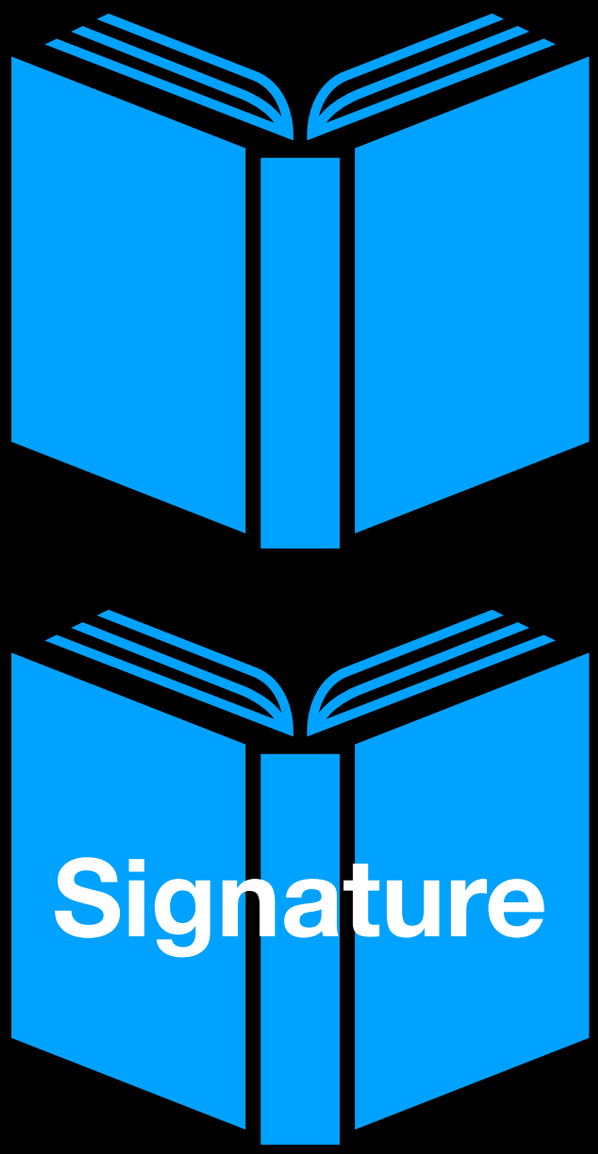
Public Key

Private Key

Create a Signature



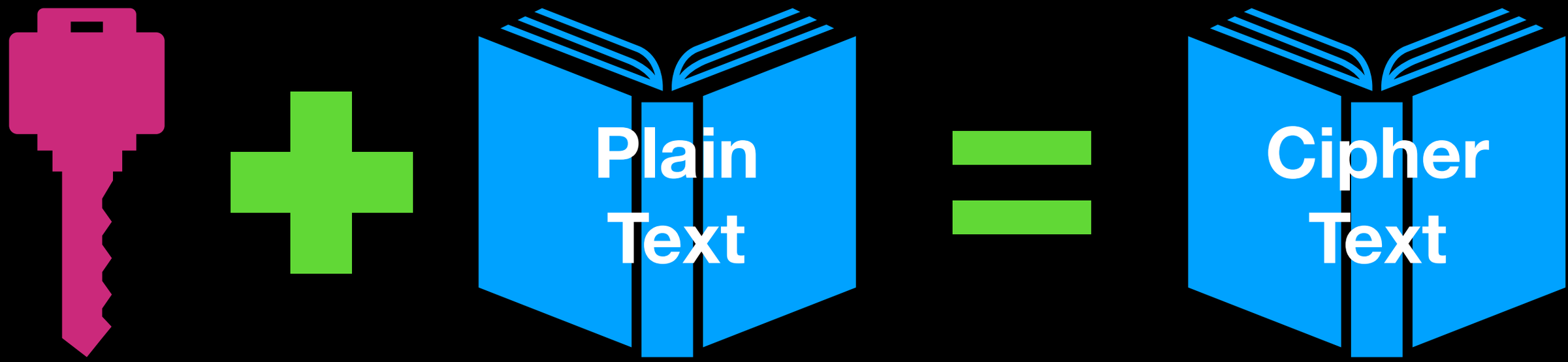
Validate a Signature



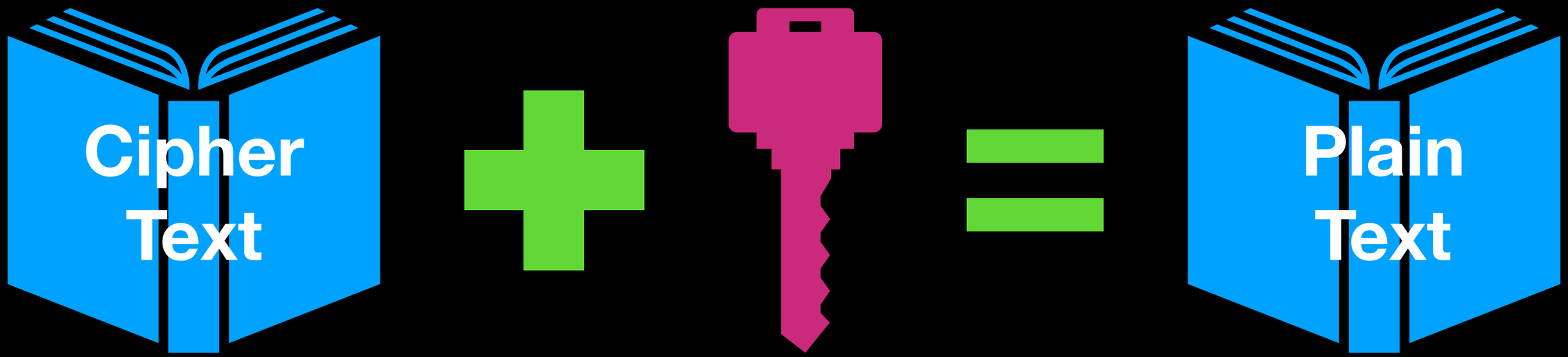
Encrypt Text

Public Key **Private Key**

Encrypt Text



Decrypt Text



Why do a Key Signing Party?

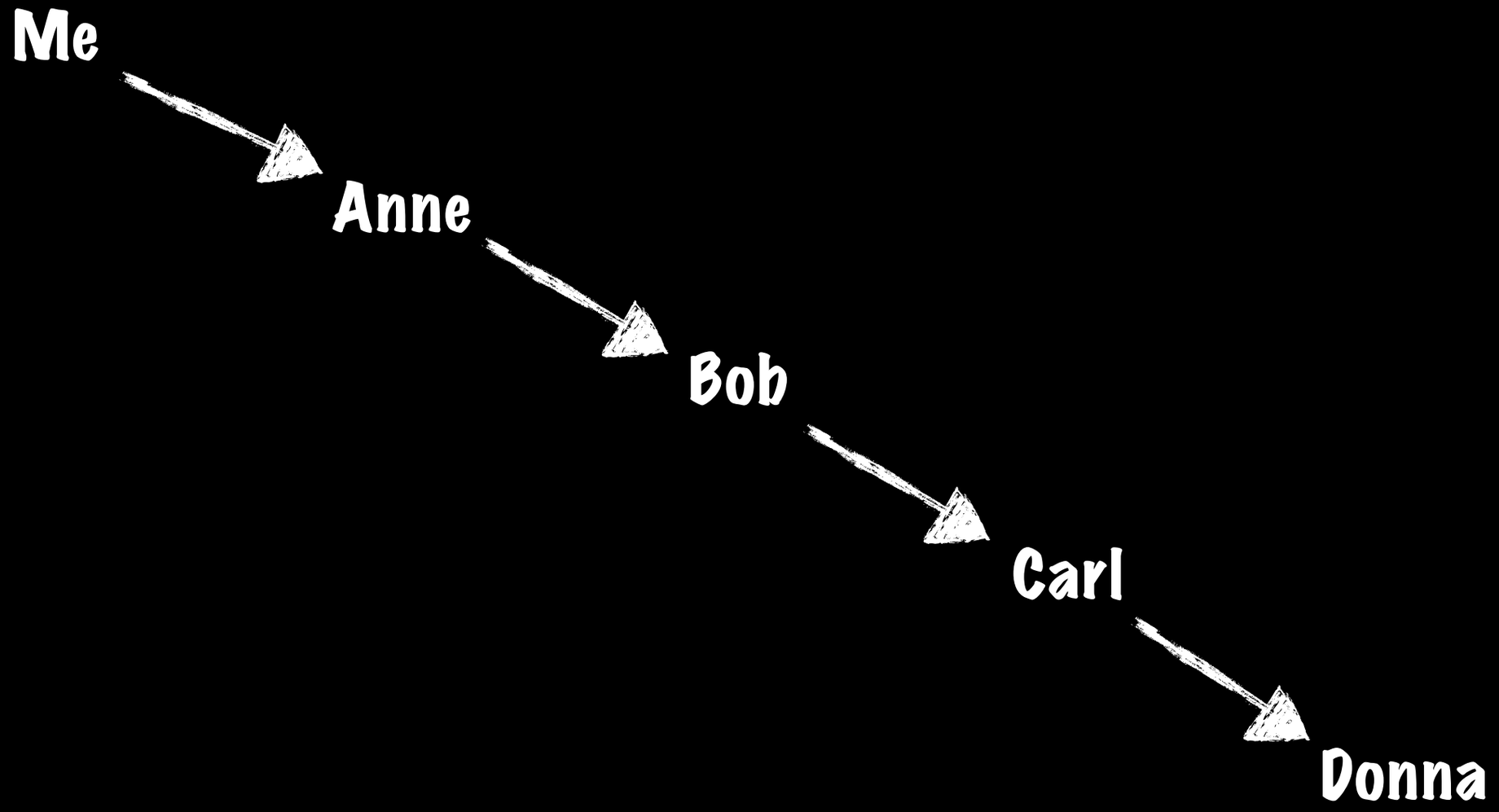
The Problem of Trust

- Need to be certain a key belongs to who you think it does
- How do you get that certainty?
- What if I want to encrypt some email for Donna, but we've never met?

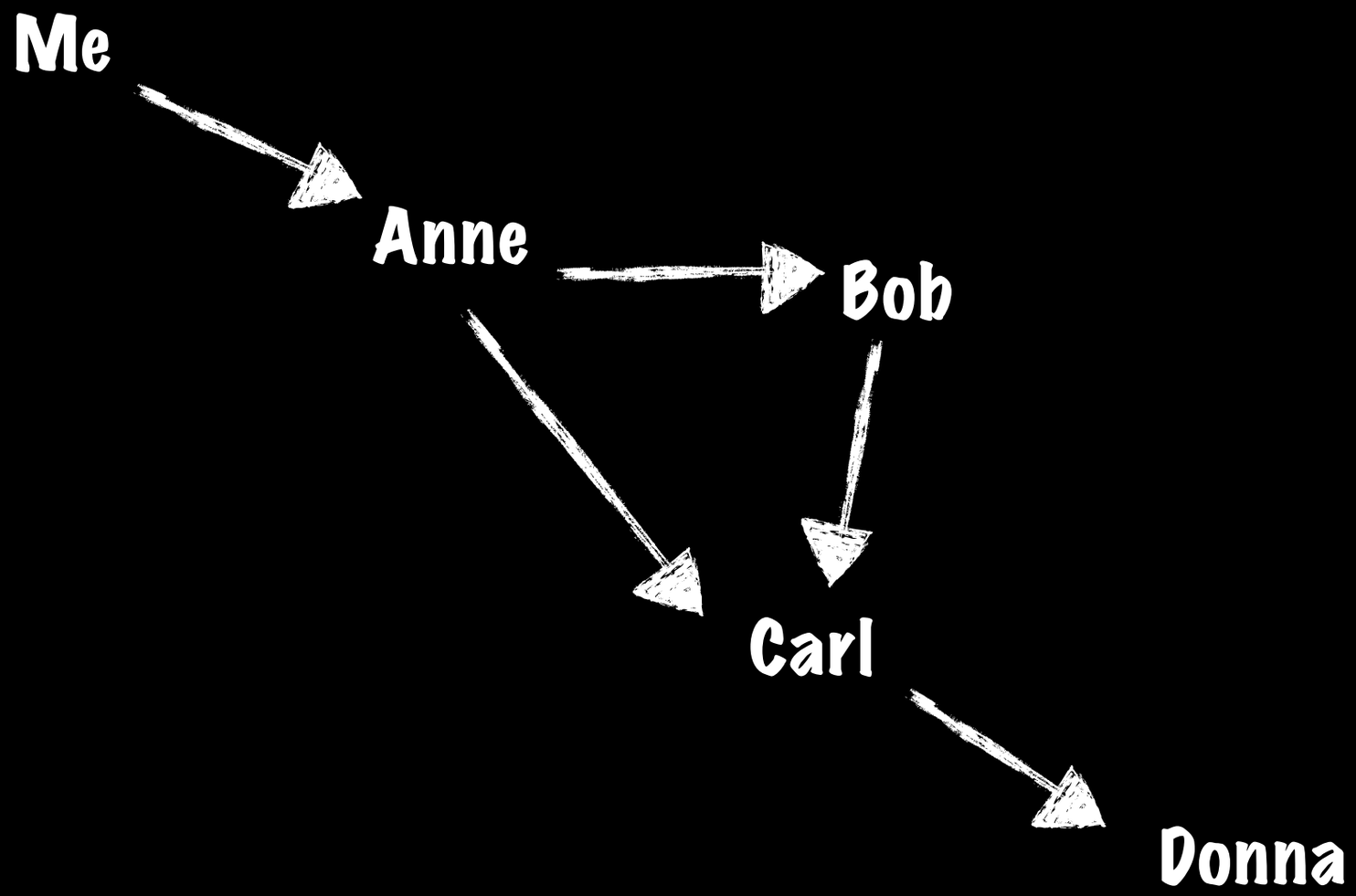
Trust Model

- Anne trusts Bob's key, and signs it
- Bob trusts Carl's key, and signs it
- Carl trusts Donna's key, and signs it
- If I trust Anne's key, then I can (probably) also trust Donna's key, by following the signatures

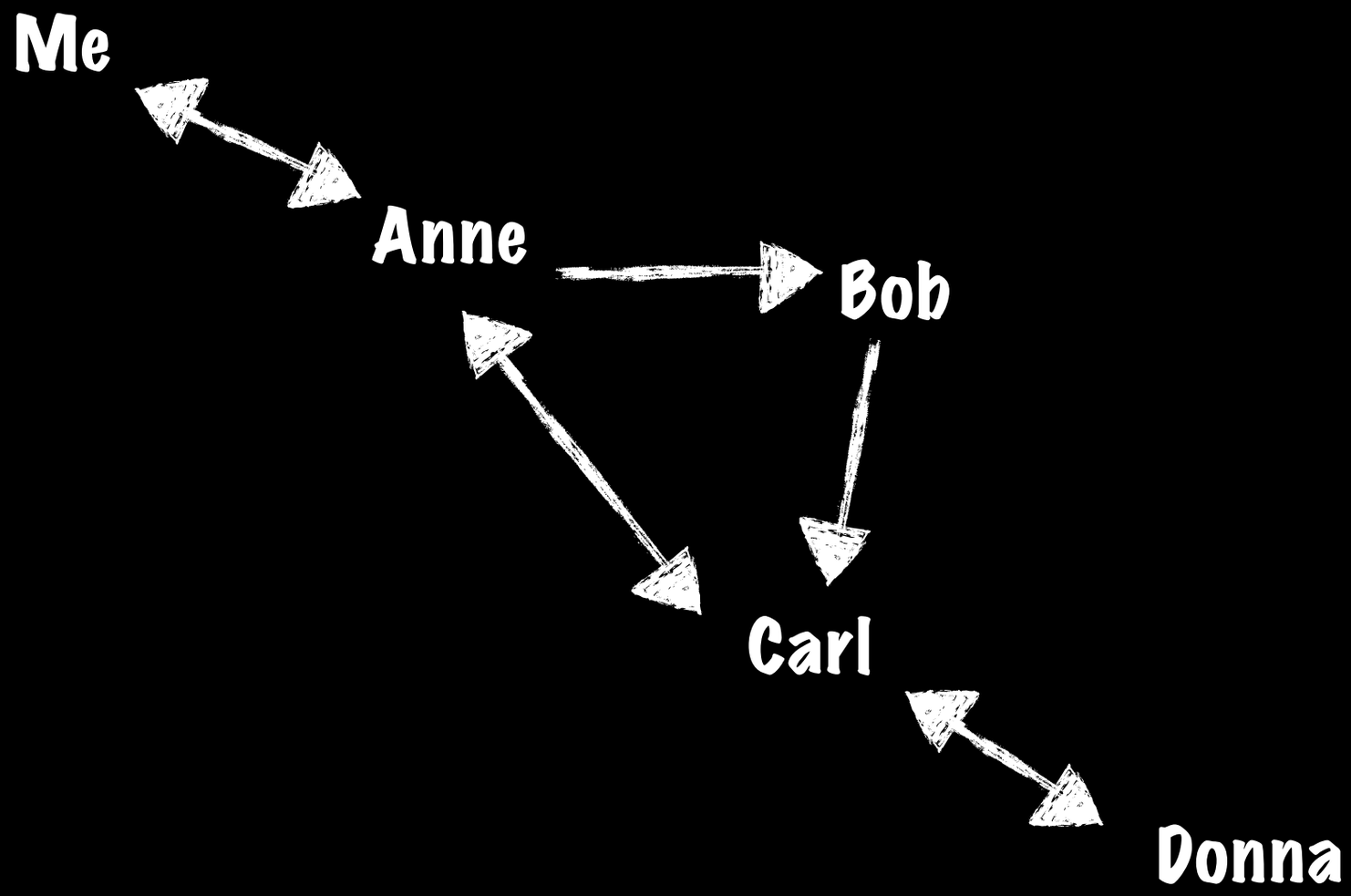
Trust Model



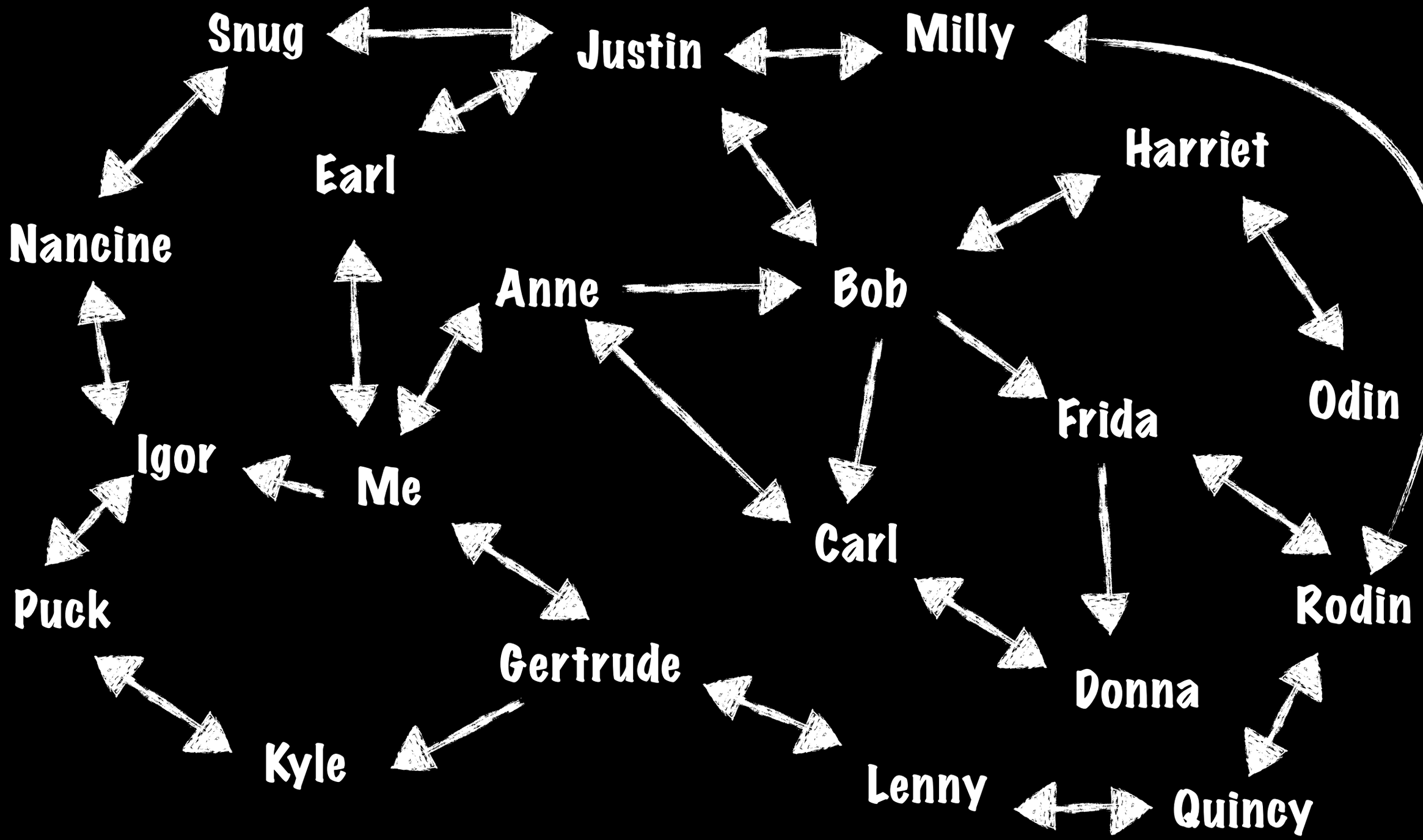
Trust Model



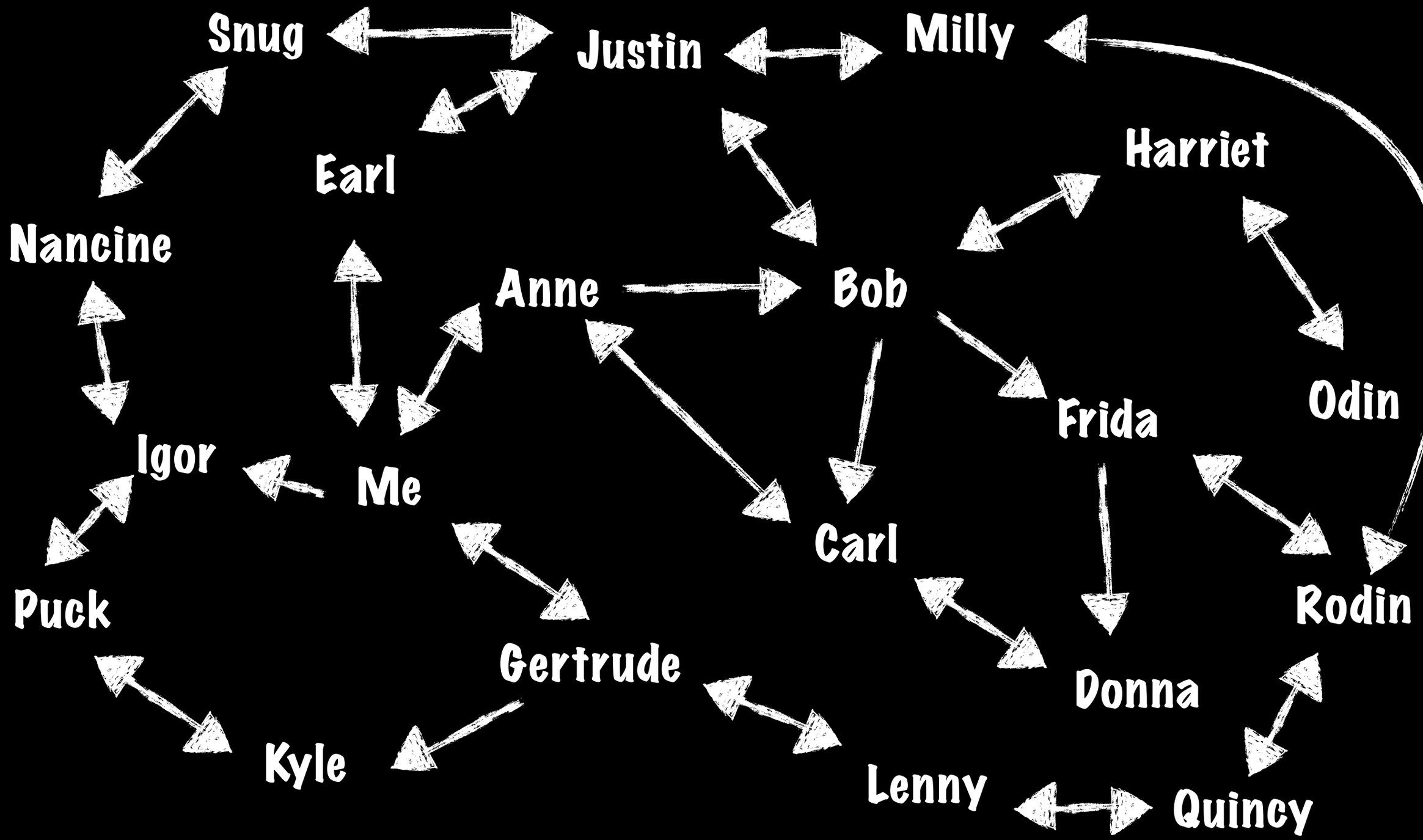
Trust Model



Trust Model


Snug
Justin
Milly
Earl
Harriet
Nancine
Anne
Bob
Odin
Frida
Igor
Me
Carl
Puck
Rodin
Gertrude
Donna
Kyle
Lenny
Quincy

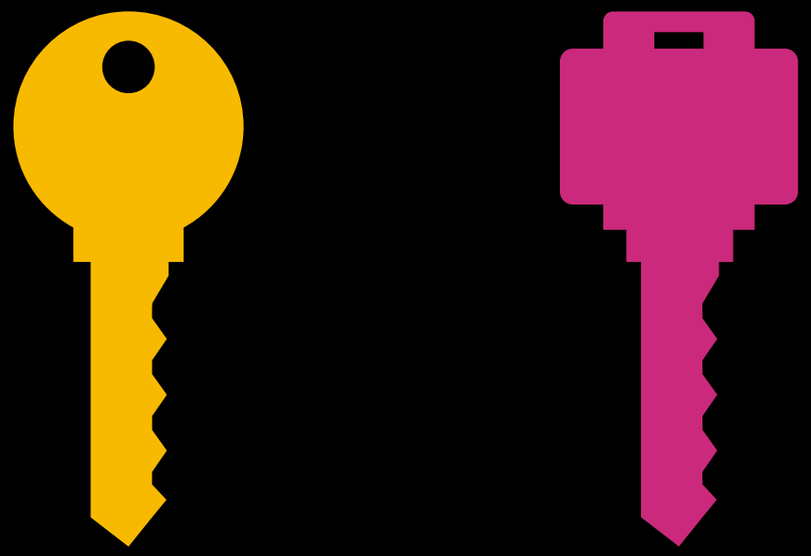
Web of Trust


Snug
Justin
Milly
Earl
Harriet
Nancine
Anne
Bob
Odin
Frida
Igor
Me
Carl
Puck
Rodin
Gertrude
Donna
Kyle
Lenny
Quincy

How Signing Parties Work

A Modified Zimmermann–Sassaman Protocol

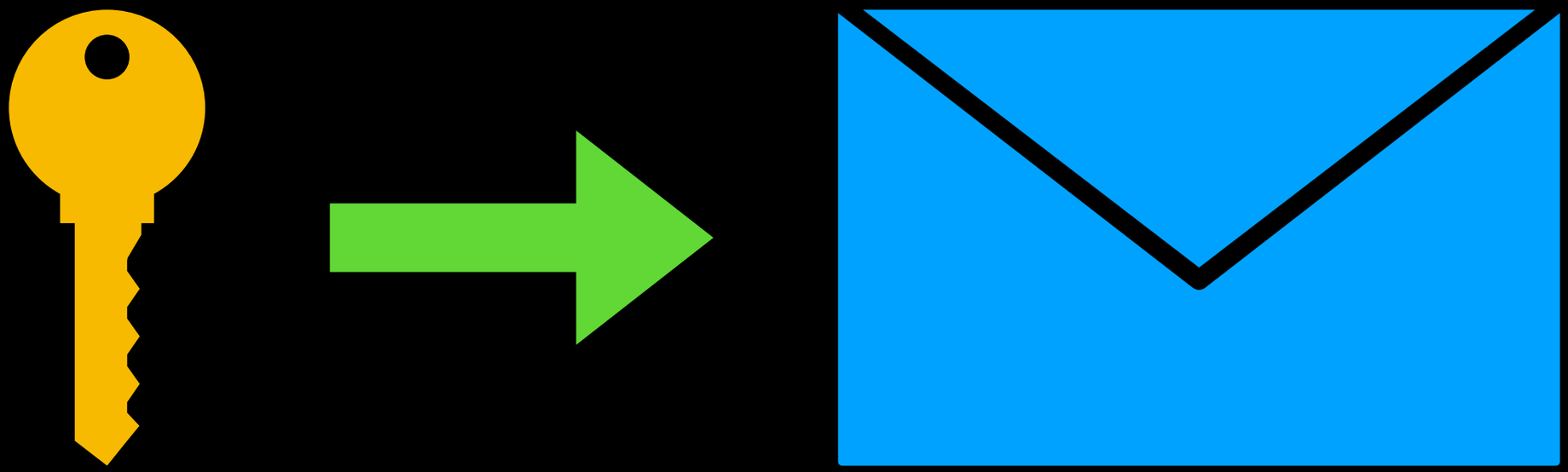
Generate Keys



Public Key

Private Key

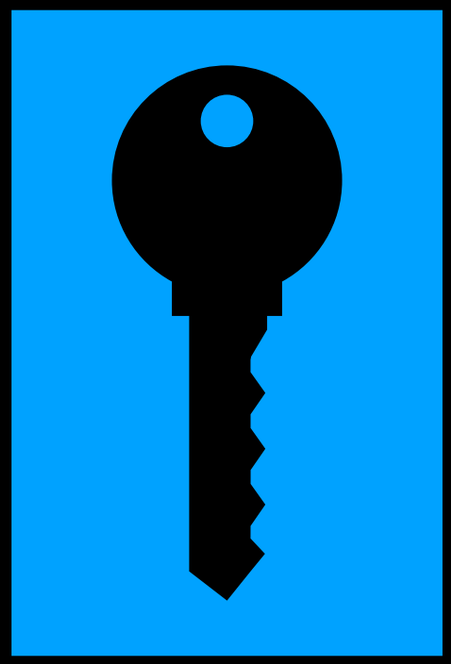
Mail Your Public Key



Send keys to:
pgpsign@dns-oarc.net

Organizer Generates a Keyring

Organizer Generates a Hash of the Keyring



e2c30cbbd12778fabd41dbef14fde93bafa9959be335d1bc7c104ada

At the Party…

- Everyone downloads the keyring and generates their own hash of it
- The organizer reads or posts their version of the hash
- Everyone compares hashes to verify that they have the same keyring the organizer does
- Everyone checks their key in the keyring and declares whether it matches their own key’s fingerprint
- Everyone checks the ID of people they don’t know, and makes note of which key belongs to that person

After the party…

- Everyone goes home
- Everyone signs the keys of those whose identities and keys they verified
- Everyone emails those signed keys back to their original owner

Caff

- Useful Key-signing tool
- Debian: ‘signing-party’
- RedHat: ‘pgp-tools’
- Last I checked, broken on MacOS and BSD

Questions?

Send keys to
pgpsign@dns-oarc.net

Don’t have a key?
https://mpounsett.github.io/pgp_key_creation/