# Pretty Good Privacy

Michuki Mwangi – AIS 2019, Kampala, Uganda
(Original slides by Joe Abley)

# Why?

- What can we use cryptography for?
- Why would we bother?
- What are the implications of not using it?

# Public Key Cryptography

- Create a public/private key pair
  - keep the private key private
  - make the public key public
- Use someone else's public key to **encrypt** data such that only they can decrypt it, using their private key
- Use your own private key to **sign** something in a way that anybody who has your public key can verify

# Trusting Public Keys

- If you want to use someone's public key (for what?) it's important to trust that the copy you have is accurate
  - How could you tell?

# Keeping Private Keys Private

- How much trouble should you go to?
- How private is private?
  - how secret is secret?

# Remember!

- You are creating keys on extremely insecure public servers
  - "afnog/afnog"
- Don't share anything that is *really* secret
- Delete your keys (public and private) when you are finished. Why?

# 1. SSH

# SSH in Practice

- SSH supports password authentication as well as key authentication. Which is better? Why?
  - SSH scanners on the Internet
- Distributing public keys
  - SSHFP records in the DNS
  - Trust on First Use (TOFU)
- Keeping up-to-date
  - Frequent enough vulnerabilities in ssh, historically, to be careful
  - OpenSSH has a great track record in responding to vulnerabilities

# Exercise

- Create a key pair on your SSH client (find out how)
  - set a passphrase to "success!"
- Transfer public key to your server
- Confirm that you can connect using ssh to your server without using a password
- Turn off password authentication on the server

# 2. PGP

# PGP in Practice

- PGP at the command line is a bit ugly
- There are plugins for mail clients to make all of this easier
  - Thunderbird
  - Mutt on the Unix/Linux command-line
  - MailMate, Apple Mail on the Mac
  - Surely something for Windows
- Web mail clients are harder. Why?

# Exercise

- Install GnuPG
- Create a key pair
- Obtain public keys from other people in the room
- Find ways to trust their public keys
- Encrypt a private message to another person, and verify that other people can't easily decrypt it