

APRICOT2015 Security Workshop:  
**Pretty Good Privacy (PGP)**

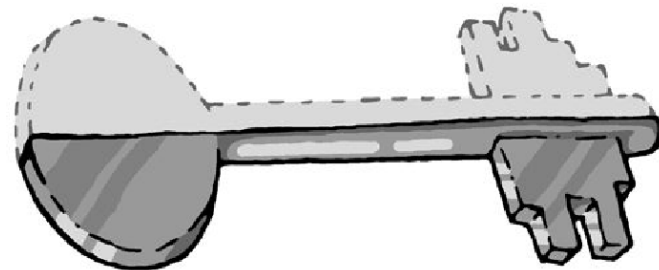
Sheryl Hermoso, APNIC  
sheryl@apnic.net

# PGP: Introduction

- Created by Phil Zimmerman in 1991 originally using symmetric encryption
- PGP 3 allowed for asymmetric encryption
- Zimmerman's team and Viacrypt (who'd licensed RSA from RSADSI) merged to form PGP Inc in 1996
- OpenPGP as a standard proposed to IETF in 1997 to avoid patent issues.
- PGP Inc now owned by Symantec
- GPG is the Free Software Foundation's implementation of the OpenPGP standard

# Asymmetric encryption refresher:

- One key mathematically related to the other.
- Public key can be generated from private key. But NOT vice versa.
- If you **encrypt** data with the **public** key, you need to **private** key to **decrypt**
- You can **sign** data with the **private key** and **verify** the signature using the **public key**



# keys

- Private key is kept SECRET.
- You **should** encrypt your private key with a **symmetric** passphrase.
- Public key is distributed.
- Anyone who needs to send you confidential data can use your public key



# Signing & Encrypting

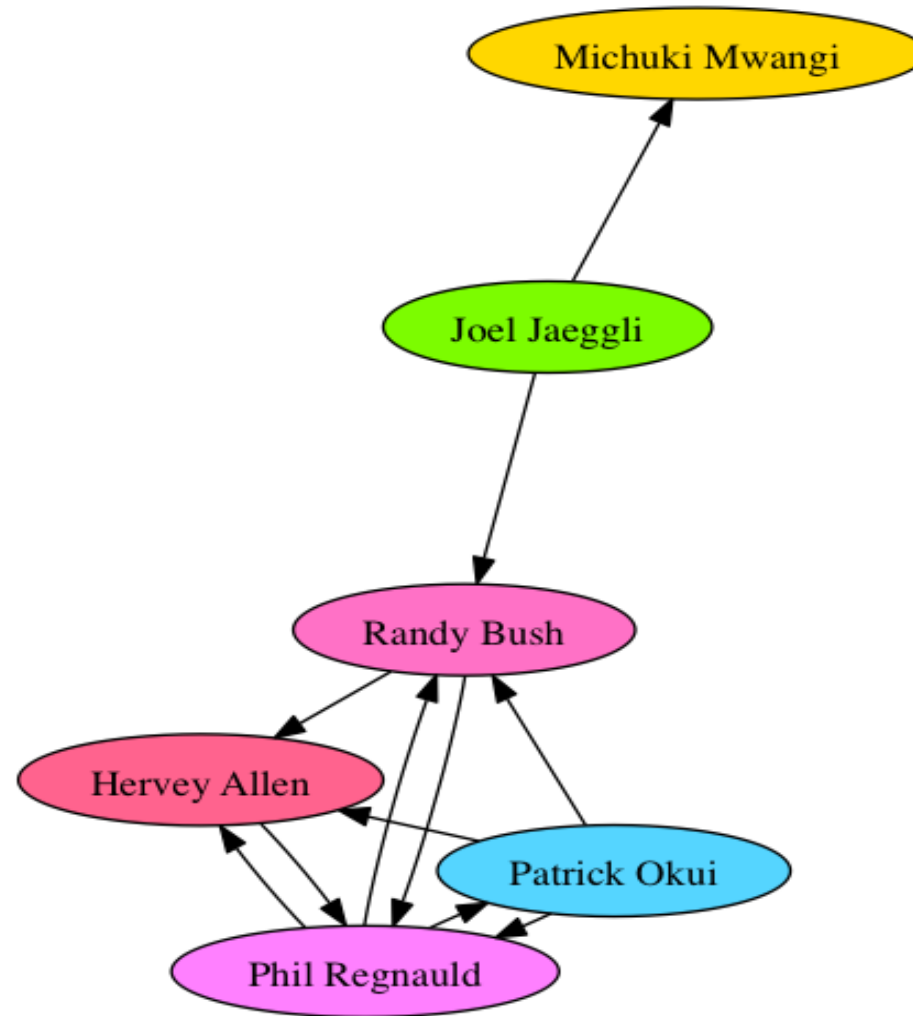
- Data is encrypted with a public key to be decrypted with the corresponding private key.
- Data can be signed with the private key to be verified by anyone who has the corresponding public key.
- Since public keys are data, they can be signed too.
- Hash functions that generate fixed length fingerprints of any input data can be used to identify keys that would otherwise be over 1024 bits long



# Trust

- Centralized / hierarchal trust – where certain globally trusted bodies sign keys for every one else.
- Decentralized webs of trust – where you pick who you trust yourself, and decide if you trust who those people trust in turn.
- Which works better for what reasons?

# Sample web of trust





# Installing GnuPG Software

- GNU Privacy Guard
- Core software either commercial from pgp or opensource from gnupg
  - GPG4Win for windows
  - GPGTools for Mac OS X
- Your package manager for Linux/UNIX
  - Source code from <https://www.gnupg.org/>
- Written by Werner Koch in 1997
  - After attending a talk by Stallman

# Key management: generation

- Using graphical tools based on what you installed above:
  - GPG Keychain Access for OS X
  - Kleopatra or GPA for windows
- Using the command line:
  - `gpg --gen-key`
- Generate a key – use your email address. The comment field can be left blank.

# Key Management

- Using graphical tools, you may see your own keypair, and all imported public keys from contacts
  - Keychain Access (Mac OS X)
  - Kleopatra (Windows)

Type	Name	Email	Created	Expires	Length	Algorithm	Short ID	ID	Fingerprint
pub	Randy Bush	randy@psg.com	9 Mar 2008	10 Jan 2015	2,048	RSA	B83A02ED	CCCC05ECB83A02ED	75F2 C48E F169 7C92
pub	Jordan Tay	jordan@apnic.net	7 Jan 2015	31 Jan 2015	2,048	RSA	E6F5A9FE	447FE895E6F5A9FE	6210 3142 A3A3 D4AF
pub	Sheryl Hermoso	shane@hermoso.me	13 May 2014	8 May 2015	2,048	DSA	E6B24E75	AF025E9AE6B24E75	94D0 38E6 CE92 CE9
pub	Yanawut Sunpornkij	syanaawut@gmail.com	13 May 2014	8 May 2015	2,048	RSA	677DA400	F1EB3500677DA400	B5C4 76DF 0287 91E3
pub	Yanawut Sunpornkij	syanaawut@gmail.com	13 May 2014	13 May 2015	2,048	RSA	97A1E8FD	60E554B097A1E8FD	4932 53D4 B317 06BE
pub	Attaporn Khaesawad	opal.attapai@gmail.com	13 May 2014	13 May 2015	2,048	RSA	D9100703	545C1FDD9100703	7BBE 1469 BF45 B2D
pub	Chai Kok Soon	chaika@ntu.edu.sg	24 Jun 2014	24 Jun 2015	2,048	RSA	BCF9B081	0F8A21D9BCF9B081	09C2 8792 CC12 86E
pub	GPGTools Team	team@pgptools.org	19 Aug 2010	18 Aug 2015	2,048	DSA	002026C4	78D78F05002026C4	85E3 9F89 0468 44C1
pub	Ruben F. Estuar Jr.	restuarjr@gmail.com	1 Oct 2014	1 Oct 2015	2,048	RSA	7A738B54	84A8C3B17A738B54	C348 D1D8 56D0 995
pub	Emergrace Puhawan	evpuhawan@uplb.edu.ph	1 Oct 2014	1 Oct 2015	2,048	RSA	EF5CB854	731E8ADDEF5CB854	FEFF D6D0 2034 2B2
pub	John D. Ultra	jdultra@up.edu.ph	2 Oct 2014	2 Oct 2015	4,096	RSA	09FCDF0	1A0C7E5909FCDF0	2D09 AED9 E1BC F9D
pub	Eugene Flores	emflores@mydestiny.net	1 Oct 2014	2 Oct 2015	2,048	RSA	4815C3C5	0C3D8B254815C3C5	CBED B030 930D B5B
pub	Nicholas Meredith	nicholasm@hostelnetworks.com.au	6 Feb 2014	6 Feb 2016	2,048	RSA	34D5997B	963C7B134D5997B	31C4 3C7E 844F 00F1
pub	Steven Bellovin	smb@cs.columbia.edu	5 Mar 2012	4 Apr 2016	2,048	RSA	821E23E4	F3B5AE1E821E23E4	F3E2 0089 8747 4E11
pub	Amanat	amanatkhalil@gmail.com	13 May 2014	13 May 2016	2,048	RSA	233557FE	4E7739B7233557FE	7804 6A57 C6E6 FCA
pub	Fakrul Alam	fakrul@dhakacom.com	11 Jun 2012	11 Jun 2016	2,048	RSA	2713A6AE	9C9C9582713A6AE	867E A9F1 F7AB 991F
sec/pub	Sheryl Hermoso	sheryli@apnic.net	18 Jun 2011	16 Jun 2016	2,048	RSA	2B058225	F57AFBF2B058225	27AF B7E0 123B 0896
pub	weixian	elvinxian@gmail.com	24 Jun 2014	24 Jun 2016	2,048	RSA	5CE763D4	07C8188F5CE763D4	D43B 321F B8C1 295F
pub	test account	rebobon2@yahoo.com	24 Jun 2014	24 Jun 2016	2,048	RSA	5845559E	99FED9BC5845559E	DAF0 D180 E711 0024
pub	Local	local8888@yahoo.com	24 Jun 2014	24 Jun 2016	2,048	RSA	84E54E60	7C3A9FB84E54E60	2D0D 5D59 9F1B E9F
pub	Abdul Rahim Ahmad	abdrhahim@uniten.edu.my	19 Aug 2014	19 Aug 2016	2,048	RSA	8920E815	DC4EECCD820E815	3B41 94D2 3F18 46BE
pub	Jerry B. Canale	jbcanale@uniten.edu.my	1 Oct 2014	5 Oct 2016	2,048	RSA	CCD6A1EA	B515FABCCD6A1EA	3F45 D5E6 A865 D1C1
pub	Rafael Lourenco Cintra	rc.alternative1@gmail.com	7 Jan 2015	6 Jan 2017	2,048	RSA	521EACFD	E4E30FF521EACFD	A2F3 3AE4 11B8 24FE
pub	Mike Lim	mikelim@vizlearn.com	24 Jun 2014	24 Jun 2018	2,048	RSA	EBFA7743	5E4432E2EBFA7743	7C77 327A 900F BAA
pub	Rahayati Zainudin	rahayati@iitl.com.my	19 Aug 2014	19 Aug 2018	2,048	RSA	00DB445E	EBAB0DD200DB445E	685A E589 F3D0 8E24
pub	Wilfredo Fajeta	wilfajeta@gmail.com	1 Oct 2014	1 Oct 2018	2,048	RSA	20395515	99845A1E20395515	3AC4 D827 A51E 42D
pub	Zen Ng	zenchuan@apnic.net	6 Jan 2015	6 Jan 2019	4,096	RSA	7C85E9E6	91A3A4E37C85E9E6	0D2D 1FA2 C6F8 212
pub	Pubudu Jayasinghe	pubudu100@yahoo.com	7 Jan 2015	7 Jan 2019	4,096	RSA	C453B583	52F28CA1C453B583	08DC 90D2 F641 BA8
pub	Peter Hill	hill.peter@gmail.com	18 Feb 2014	18 Feb 2019	4,096	RSA	BBF5F357	DBE0C92BBF5F357	28A2 C980 84DB 4A8
pub	Md. Mozammel Hoque	mohindui@gmail.com	20 Feb 2014	19 Feb 2019	2,048	RSA	82371DCA	90D876B182371DCA	0886 25A9 ED43 1B36
pub	Darwin Laurencio	dclaurencio@post.upm.edu.ph	1 Oct 2014	1 Oct 2020	2,048	RSA	76FFC43A	77C18E8A76FFC43A	3B28 4E45 A9C8 855

# Key management - Distribution

- On printed media: published book or business cards:
- Digitally in email or using sneaker-net
- Online using the openpgp key servers.
- Still does not tell you if you trust the key.

# Key management - Rollover

- Expiry dates ensure that if your private key is compromised they can only be used till they expire.
- Can be changed after creating the key.
- Before expiry, you need to create a new key, sign it with the old one, send the signed new one to everyone in your web of trust asking them to sign your new key.
- Many people create keys that don't expire. Think about the security implications of that.

# Key management - Revocation

- Used to mark a key as invalid before its expiry date
- Always generate a revocation certificate as soon as you create your key
- Do not keep your revocation certificate with your private key
  - `gpg --gen-revoke IDENTITY`

# Key management - Partying

- Key signing parties are ways to build webs of trust.
- Each participant carries identification, as well as a copy of their key fingerprint. (maybe some \$ as well 😊 )
- Each participant decides if they're going to sign another key based on their personal policy.
- Keys are easiest kept in a keyring on an openpgp keyserver in the aftermath of the party.

# Interesting gpg commands

- Get help for gpg options
  - `gpg --help` AND `man gpg`
- Print the fingerprint of a particular key
  - `gpg --fingerprint IDENTITY`
- Export a public key to an ASCII armored file.
  - `gpg -a --output my-public-key.asc --export IDENTITY`

\*IDENTITY = email or PGP key ID



# Interesting gpg commands

- Import a key from a file into your keyring
  - `gpg --import public.asc`
- Import a key from a keyserver
  - `gpg --recv-keys --keyserver hkp://keys.gnupg.net`
- Send your key to a keyserver
  - `gpg --send-keys --keyserver hkp://keys.gnupg.net`
- Sign a key
  - `gpg --sign-key IDENTITY`

# Questions

