Cryptograhic Tools

ENCRYPT, DECRYPT, SIGN, VERIFY

Stéphane Roche

2019-05-02

CREATIVE COMMONS

ABOUT STEPHANE

2015

Work at Ledger - hardware wallet company

2016–2017

Work on Ethereum

- Learn and play
- Co-found non-profit organization Asseth
- Contribute to the ERC20 Consensys smart contracts
- Dether.io

2017–2019

Found Bitcoin Studio

Focus on Bitcoin education

Consultant at Chainsmiths

https://www.bitcoin-studio.com
@janakaSteph on Twitter
bitcoin-studio@protonmail.com

OUTLINE

1

OpenSSL

- Cryptographic library and SSL/TLS toolkit, founded in 1998
    - Symmetric ciphers
    - Public-key cryptography
    - Cryptographic hash functions

- One of the most used and important pieces of software in the world
    - Two-thirds of the web in 2014

- Several critical vulnerabilities
- Heartbleed
    - Introduced into the software in 2012 and publicly disclosed in April 2014
    - Allows eavesdropping and trusted website impersonation

LIBRESSL

- Heartbleed leads to LibreSSL OpenBSD fork
- Goals of modernizing the codebase, improving security, and applying development best practices
- MacOS uses LibreSSL by default

- Get openSSL version and info
  - $ openssl version -a

- Print manual
  - $ man openssl

STANDARD COMMANDS

- TLS
  - ca: create certificate authorities
  - verify: checkings for X509
  - x509: data managing for X509
- dgst: compute hash functions
- enc: encrypt/decrypt using symmetric key algorithms
- PKC
  - genpkey: generates private keys (encouraged cmd)
  - genrsa: generate a pair of public/private key for RSA algorithm
  - rsa: RSA data management
  - ec: process EC keys
  - rsautl: to encrypt/decrypt or sign/verify signature with RSA
- passwd: computes the hash of a password (crypt algo by default)
- rand: generation of pseudo-random bit strings

PRIVACY-ENHANCED MAIL (PEM)

- File format for storing and sending keys, certificates, and others
  - CERTIFICATE
  - CERTIFICATE REQUEST
  - PGP PUBLIC KEY
  - PRIVATE KEY…
- Can be .pem, .cer, .crt, .asc, .key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBFqlW5MBEACyl8c5CT3rr2CdOhmJzKCws3LesZdv5BgF8g31vICepTu9Sumq
LjJEhIeCpysIQtsJAyNl/5O3CUNBOyDVZTnguOyF2PyDIl9SH0gKt/uRkCT49Sp7
XDbDIWBXifn2ErE2ofr4sbyHaab1W2pIMqFBWrxltYGykw8hVaxBCj2PWzEd3TXY
VXx0KuFXkxnfzGRS2rXHUhbvwTHavhjyN3a69Ygt7iaATfIuVxR+l+ytuJBO66r+
hzEXyO877Y2v+5VAi1tZO9EXVCreUw3tSiEFbW4i0QtaZmSD+kIGna9LEhWnz26A
lX8iyP/hAaUr8/SVdC+UmrxtZqCxLl1enFJXOhBFub/F7E8CTW7wvN5TDNf1WsxX
Hkggf+eTiem9cP2LHxQqFrjG6wRPWLh1McRd6bKElHqEHriMBDVzSZoHtU5W4yb+
nzhwCSpQN4v1dhFUCzqa5ZtnBTYO601m3HW5BwXJ6AFQws4YaBbtBGZQ2N+LdQ/t
SR2Ll3eWP3oGHivE/m+wmpqwysi0LACQXdM0mnaXLH0a1JgqlvqcdXHvctKpaH12
qXt3E4vg[...]
=wjPp
-----END PGP PUBLIC KEY BLOCK-----
```

CIPHER SUITE SELECTION

- A common task in TLS server configuration is selecting which cipher suites are going to be supported
  - TLS needs to decide exactly which cryptographic primitives to use
- List supported ciphers
  - `$ openssl ciphers -v`

CRYPTOGRAPHIC HASH FUNCTION

- $ openssl dgst
    - Digest functions output the message digest in hex
    - Signing and verification
    - Message authentication code

- echo -n 'hello' | openssl sha256
- echo -n 'hello' | openssl dgst -sha256

- echo -n 'hello' | openssl dgst -hmac myKey

SIGN / VERIFY

- Sign the digest using the specified private key
  - $ echo -n 'hello' | openssl dgst -sign private.pem
  - $ openssl dgst -sha1 -sign private.pem -hex < test.pdf
  - $ openssl dgst -sha1 -verify public.pem -signature signature.bin < test.pdf

- Verify
  - $ openssl dgst -signature sigFile
  - $ echo -n sig | openssl dgst -verify pubKeyFile

ECDSA

- Generate an ECDSA private key
  - $ openssl ecparam -name secp256k1 -genkey -out ec-priv.pem
- Print ECDSA key pair in raw hex text
  - $ openssl ec -in ec-priv.pem -text –noout
- Extract public key from private key
  - $ openssl ec -in ec-priv.pem -pubout -out ec-pub.pem
- Compress public key
  - $ openssl ec -in ec-pub.pem -pubin -text -noout -conv_form compressed

RSA

- Generate private key using default parameters (1024 bits)
  - $ openssl genpkey -algorithm rsa
- Generate private key of 4096 bits
  - $ openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:4096
- Generate and encrypt privkey using 128-bit AES and passphrase "hello"
  - $ openssl genpkey -algorithm RSA -aes-128-cbc -pass pass:hello
- Sign/verify with RSA from a digest
  - $ openssl rsautl -sign -in digest.txt -out signature.txt -inkey private.pem
  - $ openssl rsautl -verify -in signature.txt -out digest.txt -inkey public.pem –pubin

SYMMETRIC ENCRYPTION (KEY FROM PASSWORD)

- Blowfish in CBC mode
  - $ echo -n 'hello' | openssl bf-cbc -base64 -p
  - $ echo -n 'hello' | openssl bf-cbc -base64 -p -nosalt (unsafe)

  - $ openssl enc -bf-cbc -in hello.txt -out hello_cipher.txt
  - $ openssl enc -bf-cbc -d -in hello_cipher.txt -out hello_decipher.txt

SYMMETRIC ENCRYPTION WITH EXPLICIT KEY

- Blowfish in CBC mode
  - $ openssl enc -bf-cbc –in hello.txt -out hello_cipher.txt -iv 0123456789ABCDEF -K 0123456789ABCDEF0123456789ABCDEF
  - $ openssl enc -bf-cbc –d –in hello_cipher.txt -out hello2.txt -iv 0123456789ABCDEF -K 0123456789ABCDEF0123456789ABCDEF

PRIME

- Generate a 160 bits prime number
  - $ openssl prime -bits 160 –generate

- Check if a number is prime
  - $ openssl prime 119054759245460753

PASSWD

- Generate password hashes that interoperate with traditional /etc/passwd files, newer-style /etc/shadow files, and Apache password files

- Generate password
  - $ openssl passwd MySecret

RAND

- Generate binary or base64-encoded data

- Write 128 random bytes of base64-encoded data to stdout
    - $ openssl rand -base64 128

CONCLUSION

- Lot of commands and flexibility
- Nice if you want to learn cryptography
- May be need to TLS server setup
- Not for everyday use

2

PRETTY GOOD PRIVACY

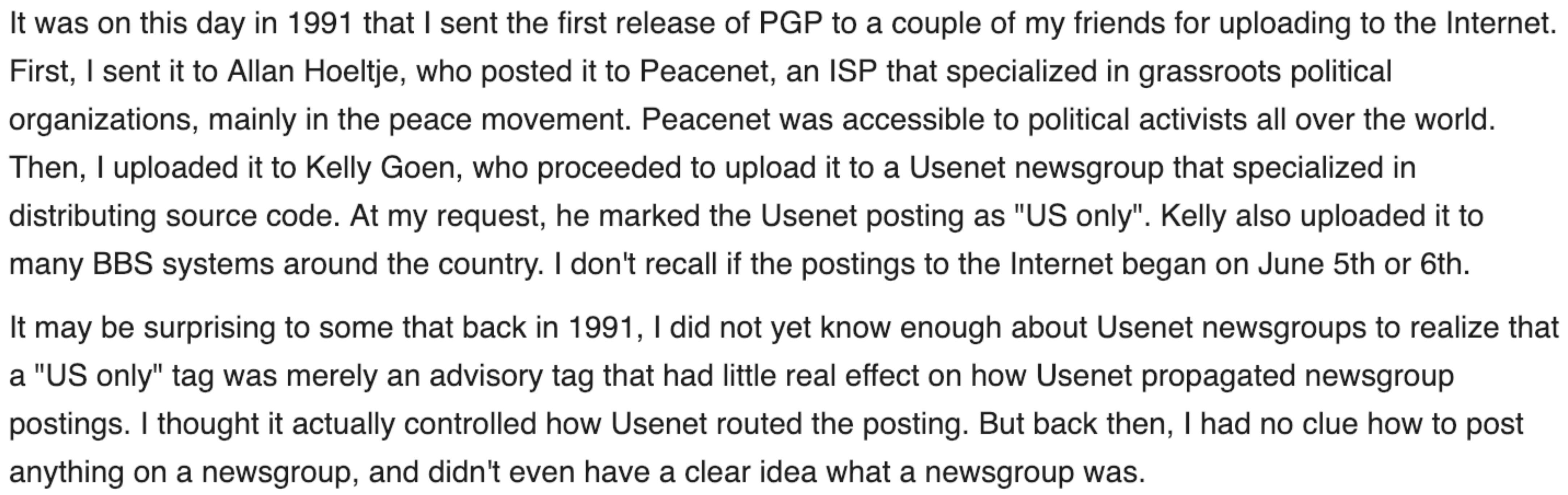
PRETTY GOOD PRIVACY (PGP)

- Encryption program that provides cryptographic privacy and authentication for data communication

- Developed by Phil Zimmermann, released in 1991

- Criminal investigation by the US Government in 1993
  - "munitions export without a license"
  - Crypto wars

PGP FIRST RELEASE



It was on this day in 1991 that I sent the first release of PGP to a couple of my friends for uploading to the Internet. First, I sent it to Allan Hoeltje, who posted it to Peacenet, an ISP that specialized in grassroots political organizations, mainly in the peace movement. Peacenet was accessible to political activists all over the world. Then, I uploaded it to Kelly Goen, who proceeded to upload it to a Usenet newsgroup that specialized in distributing source code. At my request, he marked the Usenet posting as "US only". Kelly also uploaded it to many BBS systems around the country. I don't recall if the postings to the Internet began on June 5th or 6th.

It may be surprising to some that back in 1991, I did not yet know enough about Usenet newsgroups to realize that a "US only" tag was merely an advisory tag that had little real effect on how Usenet propagated newsgroup postings. I thought it actually controlled how Usenet routed the posting. But back then, I had no clue how to post anything on a newsgroup, and didn't even have a clear idea what a newsgroup was.

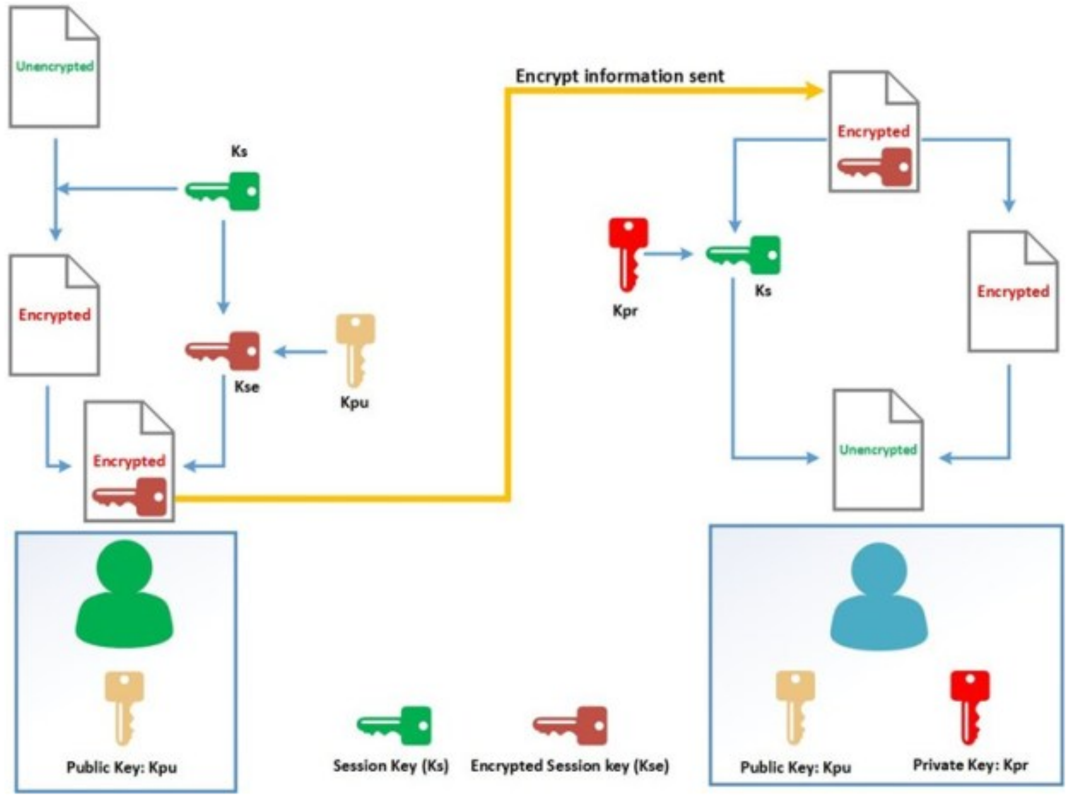
PGP Marks 10th Anniversary

GNU PRIVACY GUARD (GNUPG OR GPG)

- Free-software replacement for Symantec's PGP software suite
- Complies with OpenPGP standards (RFC 4880)
- Does not use patented or otherwise restricted software or algorithms
- GnuPG 2.x series replaces integrated cryptographic library with Libgcrypt

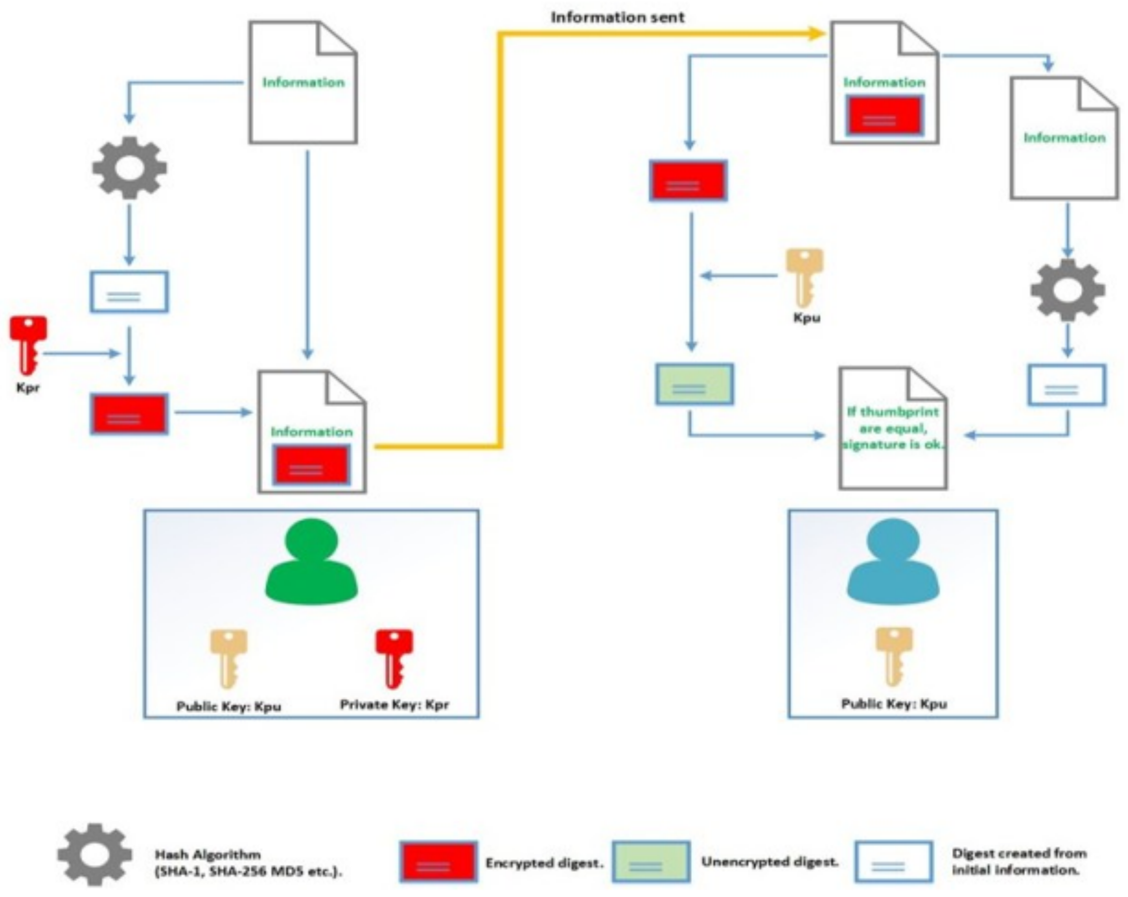
CONFIDENTIALITY

- PGP can be used to send messages confidentially
- Combines symmetric and asymmetric cryptography



DIGITAL SIGNATURES

- PGP supports message authentication and integrity checking
- Use RSA or DSA algorithms



WEB OF TRUST

- Proving ownership of a public key

- Public key infrastructure
  - Binds PK and identity using digital certificates (X.509)
  - Certification and validation authorities

- WoT is the decentralized alternative
  - Anyone can certify an identity by signing a PK
  - You can trust a PK signed by someone you trust

CERTIFICATES

- *Trust signatures* can be used to support creation of certificate authorities
    - A trust signature indicates both that the key belongs to its claimed owner and that the owner of the key is trustworthy to sign other keys at one level below their own
- Different certification level

VULNERABILITIES

- EFAIL (May 2018)
  - Leak the plaintext of encrypted emails
- SigSpoof (June 2018)
  - Allows faking signatures
  - Affect gpg software since 0.2.2 (1998)
  - Only if user has the verbose option set in their gpg.conf file

GPG SOFTWARES

- GPGTools
  - Full cryptographic software suite for MacOS
    - GPG Keychain Access
    - GPGMail
    - GPGServices
    - GPGPreferences
    - MacGPG

CHECKING SOFTWARE INTEGRITY WITH GPGTOOLS (MAC OS)

- Download the software you want

- Download the attached signature in the same folder
  - Source and/or executable signed from the developers
  - Make sure the extension is ok (i.e .asc)

- Right click on the source/exec or signature > Services > "OpenPGP: Verify signature of file"

- Even better, just double-click on the signature!

3

KEYBASE

- Key directory that maps social media identities to encryption keys (including, but not limited to PGP keys) in a publicly auditable manner

- End-to-end encrypted chat

- Cloud storage system (KBFS)
  - Public
  - Private
  - Team

- Encrypted git repository

KEYBASE CLI

- Manage keys, sign/verify, encrypt/decrypt
    - $ keybase pgp help
    - $ keybase pgp encrypt chris -m 'hello'
    - $ keybase pgp encrypt maxtaco@twitter -m 'hello'
    - $ keybase sign –m 'hello'

- Export PGP private keys from GnuPG and write them to KBFS
    - $ keybase pgp push-private --all

CONCLUSION

- Bulletproof cryptography doesn't exist
  - Best test is test of time

- Use well-established, user-friendly cryptographic softwares
  - Use a GPG software
  - Don't use openSSL directly unless it's your job
  - Bulletproof software doesn't exist

BIBLIOGRAPHY

- openSSL tutorial
  - https://www.madboa.com/geek/openssl
- GPG softwares
  - https://www.openpgp.org/software
- OpenPGP Key server
  - http://keys.gnupg.net
- NeoPG – GPG fork
  - https://neopg.io
- PGP Marks 10th Anniversary
  - http://www.philzimmermann.com/EN/news/PGP_10thAnniversary.html