Current challenges for the OpenPGP keyserver network

Is there a way forward?

Gunnar Eyal Wolf Iszaevich • Jorge Luis Ortega Arjona

LibrePlanet 2022 • 2022.03.19

Once upon a time, there was a happy and naïve network. . .

DM 5,-
HAPPY COMPUTER
2/84 FEBRUAR
B 2609 E
5 Heimcomputer der Oberklasse im Vergleichstest
Alphatronic PC, Atari 800 XL, Acorn B, Sharp MZ 700, Spectravideo SV 328
So zaubert man Comics auf den Bildschirm: Atari als Trickfilmstudio
War Games – der Kino-Hit als Spielprogramm
Listing des Monats: Zauberschloß
Abenteuerspiel mit vielen Tricks
Musikbaukasten für den Apple II
Jede Menge Listings mit Programmbeschreibung
Tolles 3D-Labyrinthspiel
Vokabeln lernen
Funktionstasten belegen
und noch viel mehr Programme sowie Softwaretests, Tips und Tricks für VC 20, TI 99/4A, Atari, Dragon 32, Colour Genie, Spectrum, PC 1500, TRS-80, Apple II, Commodore 64, ZX 81
DM 6,50
AGAZIN
mit VGA-Grafik
Fünf perfekte Kombinationen: Monitor plus Karte
Fernsehstar »Alf« als Computerspiel
Tolle Tips für heiße Spiele
B 2609 E
R-MAGAZIN
Commodore- und Schneider-Teil

freepngimg.com (Attribution)

)

Happy Computer, archive.org (CC BY-NC-ND)

But the world is full of evil. . .

positek.net / shutterstock (CC BY-SA)

Fortunately, Internet has evolved: We now have cryptography everywhere!

But... What does this cryptography really give us?

Protection against eavesdropping

Alex Briseño, Flickr (CC BY-SA-NC)

What do we get from the simple use of *public-key cryptography*? And what is still not covered?

We get

- Strong cryptography
  - Impossible to break in a reasonable time, even with current Nation-State resources
- Uses algorithms that have received public, expert scrutiny
  - ElGamal, DSA, RSA, EC
- Works over preexisting protocols
  - E-mail, local storage

We do not get

- Hiding the *fact there is communication* ocurring between two participants
  - Metadata analysis
- Verification of correct identity
  - *Equivocation* attacks
  - *Man in the Middle* (MITM)

PGP: Pretty Good Privacy

30 years flying high

Errol Cavit, Flickr (CC BY-NC-ND)

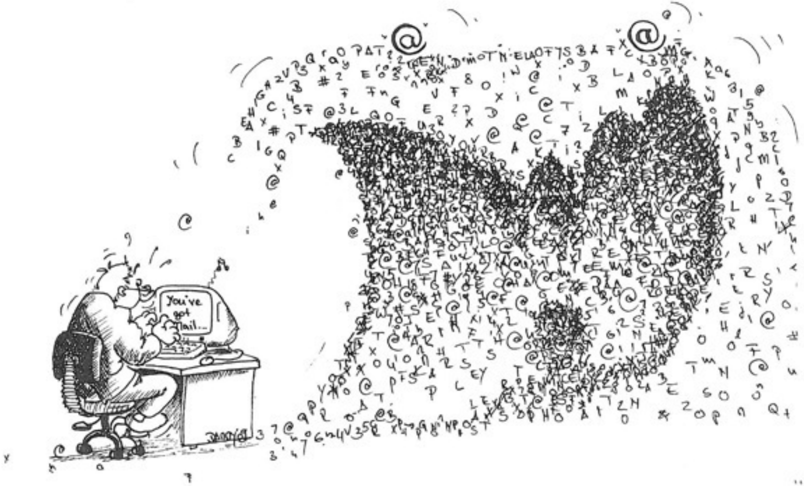
Construction blocks for *identity verification*

Andy Hay (CC-BY)

What does it mean to *verify an identity*?

Aldon Hynes (CC-BY-SA)

Internet is too big to *know* everybody I interact with!

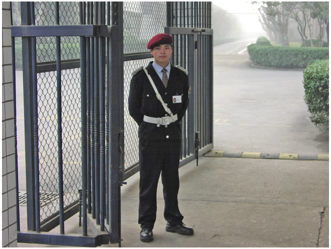


Transitive trust distribution mechanisms

. . . But we can trust *somebody*, right?

and we can trust on the *truth* of the identities they are willing to back. . .

① Centralized trust



ESTADOS UNIDOS MEXICANOS

MÉXICO

INSTITUTO NACIONAL ELECTORAL
CREDENCIAL PARA VOTAR

NOMBRE
GOMEZ
VELAZQUEZ
MARGARITA

SEXO M

DOMICILIO
VIADUCTO TLALPAN NO. 100
COL. ARENAL TEPEPAN, ALCALDIA
TLALPAN, C.P. 14610, CIUDAD DE MEXICO.

CLAVE DE ELECTOR GMVLMR80070501M100

CURP
GOVM800705MCLR01

AÑO DE REGISTRO
2008 01

FECHA DE NACIMIENTO
05/07/1980

SECCIÓN
0001

VIGENCIA
2019 - 2029

Margarita

Robbie Sproule, Wikipedia (CC BY)
Francis Sarahi Castro Ponce, Wikipedia (CC 0)

everystockphoto.com, Wikipedia (DP)

② Distributed trust

Miren Pardo, Juegos de asamblea: Conocemos a nuestros compañeros (CC BY-SA-NC)

Formalizing a little bit. . .

Centralized mechanisms

- A set of *ultimate roots of trust* are *centrally* defined
- Each *Root of trust* can *delegate* trust on several *Ceritifation Authorities* (CA)
- Communication parties (i.e. servers) provide their public key and a CA-signed *certificate*

⇓

PKI-CA model

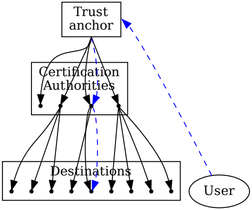
Distributed mechanisms

- Centered in *each user*
- Every user can *emit ceritifcations* for whom they personally know
  - Signing policies?
  - What does it mean to *know*?
  - Can I trust *your* criteria?
- A global *Web of Trust* global is *woven*

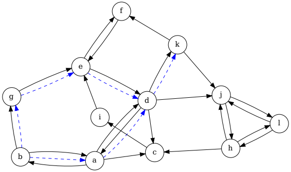
⇓

WoT model

Transitive trust distribution models



Centralized: Certification Authorities (PKI-CA)



Distributed: Web of Trust (WoT)

Focus of the work: Distributed model (WoT)

. . . But that requires *many people* to know *many people*!

Steven Fruitsmaak, WikiNews (CC BY)

So, we only need to *grow* the size of the WoT?

- Everybody verifies each other's documents (government-issued ID?)
- *Certifies* the keys of the rest of the group
- Network tust strongly increases!

Steven Fruitsmaak, WikiNews (CC BY)

So, we only need to *grow* the size of the WoT?

- Everybody verifies each other's documents (government-issued ID?)
- *Certifies* the keys of the rest of the group
- Network tust strongly increases!

- . . . In >300 people gatherings. . .

SRSLY?

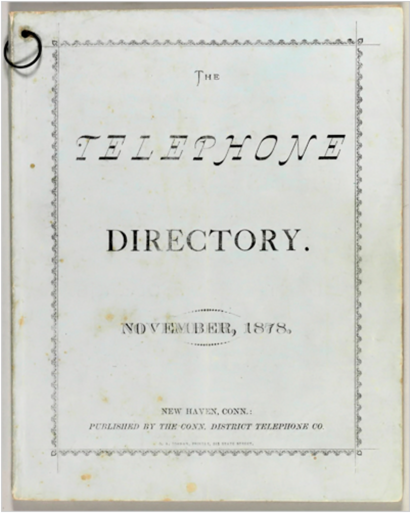
Steven Fruitsmaak, WikiNews (CC BY)

The public key distribution problem

A key distribution *infrastructure* is now needed. . .

- Under TLS (PKI-CA), key+certificates are presented upon session establishment
    - Watch out for MitM and revocations!
- Under OpenPGP (WoT), the destination key must be obtained *before sending a message*
    - Asynchronous operation

⇒ PKS keyservers



THE

TELEPHONE

DIRECTORY.

NOVEMBER, 1878.

NEW HAVEN, CONN.:

PUBLISHED BY THE CONN. DISTRICT TELEPHONE CO.

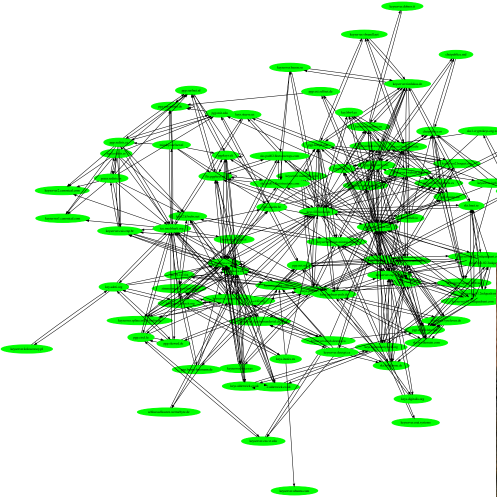
[illegible]

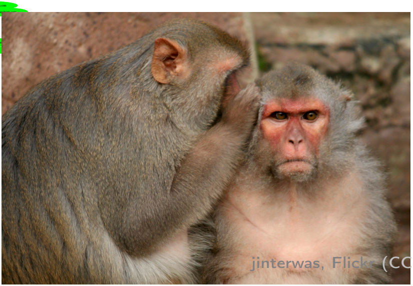
Connecticut District Telephone Company, 1878 (DP)

But. . . how do we avoid centralization?

Javier Álvarez, Flickr (CC BY-ND)

Set of keyservers running an *epidemic* or *gossip protocol* for *large sets reconciliation. . .*




jinterwas, Flickr (CC BY)

Result ①: Binary, non-modifiable, distributed, non-authenticated, eventually consistent storage

Mike Izbicki (CC BY-SA)

Result ②: Attacks on the model ☹

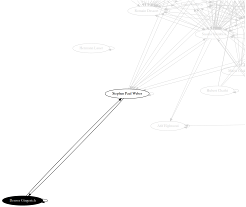
Ben Simon (CC BY)

)

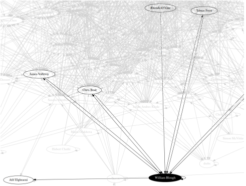
jinterwas, Flickr (CC BY)

What is *certificate poisoning*? ①

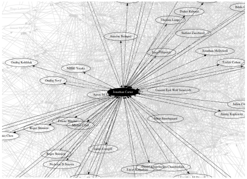
Normally, only my *direct contacts* will certify my key, allowing others to find me in the WoT



I might be little connected. . .



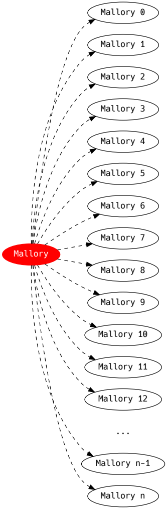
Somewhat more connected. . .



I can be *strongly* connected. . .

Normal keys will have dozens, maybe up to *hundreds* of certifications.

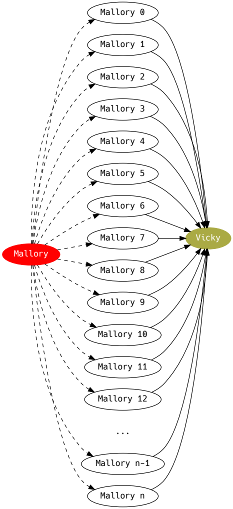
What is *certificate poisoning*? ②



An attacker, *Mallory* (M), can generate *many* throwaway identities $M_1, M_2, M_3, \ldots M_n$ ($n \approx 100\,000$)

These identities are *garbage keys*, they don't even need to be linked to *Mallory*'s real identity.

What is *certificate poisoning*? ③

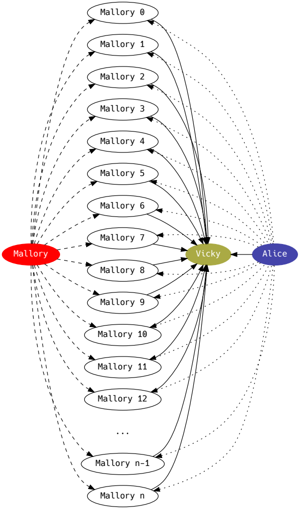


Mallory certifies victim *Vicky*'s key with all their identities — and make *Vicky*'s public key V useless.

Vicky sees herself forced to abandon her identity and generate a new pair of keys V', but. . .

- Getting her new identity connected to the WoT has a high cost (time, effort)
- Opens a time window for supplantation / ID theft

What is *certificate poisoning*? ④



When *Alice* (*A*) searches for *Vicky*'s key, upon importing it, she suffers a denial of service (and possibly an OpenPGP database corruption)

What is *certificate poisoning*? ⑤

Peter Krimbacher, Wikipedia (CC BY-SA)

Why don't we delete the spurious certificates?

Jumanji Solar, Flickr (CC BY-NC-SA)

)

Why don't we delete the spurious certificates?

Jumanji Solar, Flickr (CC BY-NC-SA)

)

José-Manuel Benito, Wikimedia (DP)

Why don't we delete the spurious certificates?

And... What about the European GDPR?

Right to be forgotten, information deletion orders...

Jumanji Solar, Flickr (CC-BY-NC-SA)

Jose Manuel Benito, Wikimedia (DP)

Why don't we delete the spurious certificates?

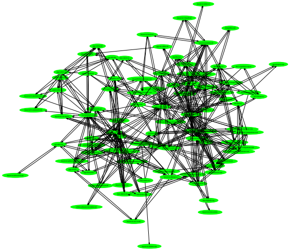
And... What about the European **GDPR**?

Right to be forgotten, information deletion orders...

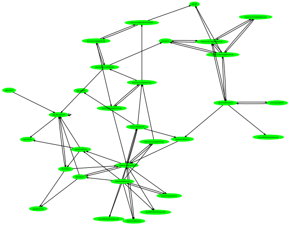
- GDPR imposes *privacy conditions* that are *impossible to comply with* for keyserver network operators
- ...All of this has caused the number of keyservers to decrease strongly... And the outlook is quite bleak ☹

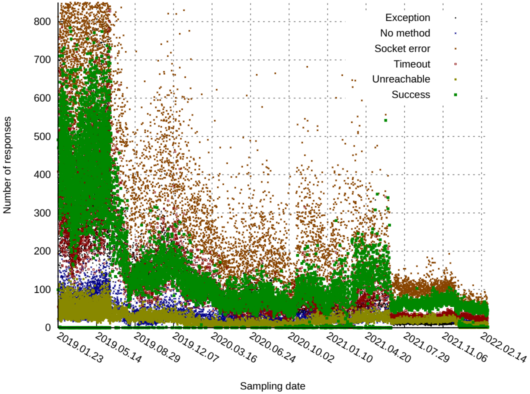
The keyserver network. . . shrinks ☹

2019.01.23



2022.03.02

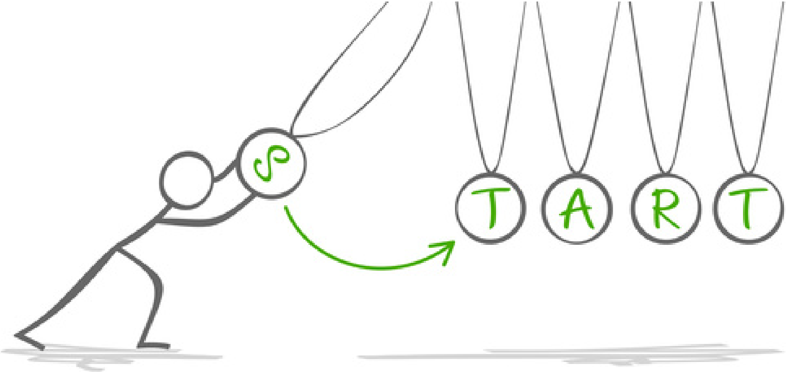




https://sks-status.gwolf.org/ ()

Project status?

As a research + implementation project. . . just warming up



Other projects addressing similar concerns

Sequoia-PGP

Hagrid Keyserver

Hockeypuck Keyserver

)

CONIKS

Key management system for key transparency

Many other ideas. . . in *academic* state

Mark Boulton (CC BY-SA)

European Southern Observatory, Flickr (CC-BY)

Central idea

Present a solution that *keeps the distributed model viable*, without requiring centralizing entities.

My main goal is to present a protocol that prevents *certificate poisoning* without compromising WoT's main positive characteristics.

First-party attested third party certification protocol → Require all OpenPGP packets modifying k to be *accepted* (signed) by k

- Certificate poisoning no longer possible
- Implementing a decades-long best-practices recommendation that has been unable to be mandated

Central idea

Present a solution that *keeps the distributed model viable*, without requiring centralizing entities.

My main goal is to present a protocol that prevents *certificate poisoning* without compromising WoT's main positive characteristics.

First-party attested third party certification protocol → Require all OpenPGP packets modifying *k* to be *accepted* (signed) by *k*

- Certificate poisoning no longer possible
- Implementing a decades-long best-practices recommendation that has been unable to be mandated
- What about information *removal*?

Expected outcome

This seemingly simple modification to the keyserver network operation pursues to:

- Allow a decentralized, public keyserver network to keep operating, mitigating the effect attacks have had on it, and allowing it to continue to exist with modern privacy expectations
- Keep the WoT decentralized transitive trust model relevant and sustainable for OpenPGP communications
  - Fundamental component for several large-scale, geographically-distributed free software development projects

Thank you very much for your attention.

Gunnar Wolf
→ gwolf@gwolf.org

Advisor:
Dr. Jorge Luis Ortega Arjona