Current challenges for the OpenPGP keys server 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...
But the world is full of evil...
Fortunately, Internet has evolved: We now have cryptography everywhere!

But... What does this cryptography really give us?
Protection against eavesdropping
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* occurring between two participants
  - Metadata analysis
- Verification of correct identity
  - *Equivocation* attacks
  - *Man in the Middle* (MITM)
PGP: Pretty Good Privacy

30 years flying high
Construction blocks for identity verification
What does it mean to verify an identity?
Internet is too big to *know* everybody I interact with!

Bob Doyle (CC BY)
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

Robbie Sproule, Wikipedia (CC BY)
Francis Sarahi Castro Ponce, Wikipedia (CC 0)
everystockphoto.com, Wikipedia (DP)
2 Distributed trust
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 *Certification 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 certificiations* 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
Modelos de distribución de confianza

Centralizado: Certificación

Distribuido: Web of Trust (WoT)

Focus of the work: Distributed model (WoT)
...But that requires *many people* to know *many people*!
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 trust strongly increases!
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 trust 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

⇒ HKP keyserver network
But... how do we avoid centralization?
Set of keyservers running an epidemic or gossip protocol for large sets reconciliation...
Result ①: Binary, non-modifiable, distributed, non-authenticated, eventually consistent storage
Result 2: Attacks on the model 😞
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*?

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

These identities are *garbage keys*, they don’t even need to be linked to *Mallory’s* real identity.
What is certificate poisoning?

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?
Why don’t we delete the spurious certificates?
Why don’t we delete the spurious certificates?

And… What about the European GDPR?

Right to be forgotten, information deletion orders…
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
Project status?

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