LUPINE PUBLISHERS

Current Trends in Computer Sciences & Applications

DOI: 10.32474/CTCSA.2022.02.000134

ISSN: 2643-6744

Review Article

Status, Threats and Proposals for the WEB-of-Trust Transitive Trust Model in Open PGP

Gunnar Wolf and Jorge Luis Ortega Arjona*

National Autonomous University of Mexico, National Polytechnic Institute, Mexico

***Corresponding author:** Jorge Luis Ortega Arjona, National Autonomous University of Mexico, National Polytechnic Institute, Mexico

Received: December 16, 2021 **Published:** February 14, 2022

Abstract

With an ever increasing proportion of Internet traffic being encrypted, the role of transitive trust models (TTMs) is each day more important. From the main two TTM models, this text focuses on the decentralized Web of Trust, mostly used for e-mail exchange and document processing. Several high-profile vulnerabilities have been found in the Open PGP key server network, the main WoT implementation, leading it to an existential crisis. This article presents the main identified threat, and sketches several possible ways forward.

Introduction

Encryption is no longer a domain of few enthusiasts; although it took over 20 years from the introduction Netscape's Secure Socket Layer (SSL) protocol, which evolved and got standardized by the IETF into Transport Layer Security (TLS) [1] nowadays a large majority of the traffic traveling through the Internet is encrypted [2]. Open PGP, a protocol mostly used for encrypted e-mail exchange, encrypted document storage and document signing, has been available since 1991 as implemented in the PGP program [3], an nowadays as an IETF standard with many compatible implementations [4]. But contrary to public perception, encryption is only one of the benefits these protocols offer to the end user: Authentication, this is, ensuring the corresponding endpoint for an encrypted communication is the desired party, andnot an impostor. This work is presented as a review of existing works on the field, during the initial stages of a research project with the intention of presenting a protocolthat keeps the distributed, decentralized properties of the Web of Trust model.

Transitive trust models

In order to be sure about a given host's identity, a trust model must be followed.The most common trust models are transitive (TTM), this is, a host A directly trusts the identity of a small set of identities $\{TA_1, TA_2, TA_3\}$ Which are able to certify others. In the Public Key Infrastructure – Certification Authority model (PKI-CA, see Figure 1(a), used for TLS, end users trust a set of servers known as trust anchors. The list of trust anchors is often decided by the operating system or web browser vendor. Trust Anchors usually delegate their certifying ability to Certification Authorities, so for the example described above, consider TA2 certified $\{CA_a, CA_b, CA_c\}$.

A Web server operator, W, requests the certification services of CA, and after all relevant checks are done, gets a certificate chain including the following assertions:

$$TA_2 \rightarrow CA_c; CA_c \rightarrow W$$

When user A wants to communicate with W, A verifies TA2 is a trusted Trust Anchor (different vendors could present different lists, although in practice they consist of mostly the same systems) and that all of the certification paths are valid. Certificates will often include important information other than the identity of the parties, such as validity periods, kind of validation performed, etc. Endpoints should ensure the CA/Browser Forum recommendations are followed [5]. In the Web of Trust model (WoT, see Figure 1(b)), used for Open PGP, each user will have a different view of trust in the graph of nodes constituting the graph's reachable set (the subset of a graph bound together by inbound and outbound edges). In the example in Figure 1(b), if user Bob (b) wants to send a message to user Karen (k), a trust path must be found so that b k is possible. The following paths are available:

$$b \to a \to d \to k$$
$$b \to g \to e \to d \to k$$

Certifications under WoT can also carry validity information, and can be assigned different weights. In the example graph, if edge $a \to d$ has a weight of 0.5, but edges $g \to e$ and $e \to d$ have weights of 0.8, and assuming trust weight assessment can be represented by multiplication, the second path could result in a greater trust assigned to k than the first one. Different models can be followed for assessing trust levels [6].While it must be mentioned that there are other trust models not based on transitivity, such as Trust On First Use (TOFU) used for instant messaging networks or the fully-centralized models used for intra-corporate communications, TTMs are much better suited for communications in open, large-scale networks. Under the WoT, validating a certification path requires being able to find keys for the identities of a prospective endpoint, or for the different keys in the certification path. For Open PGP, specific key server software was developed and later standardized [7, 8]. Independent key servers synchronize with each other using the Gossip large set reconciliation protocol (Minsky and Trachtenberg 2002). Gossip ensures keys uploaded to any of the key servers participating in a network quickly reach all other servers.

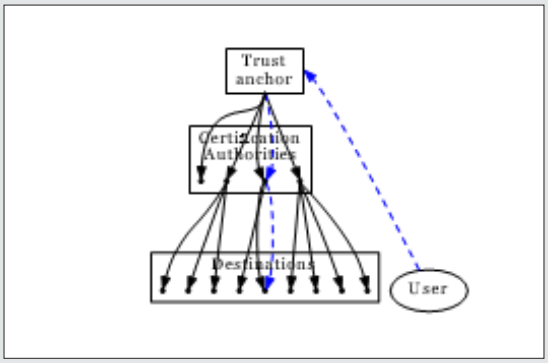


Figure 1(a): Centralized model: PKI-CA. User verifies there is a valid path of trust from a prespecified trust anchor to the destination they want to reach.

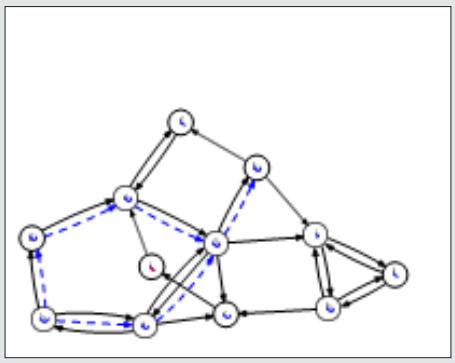


Figure 1(b): Distributed model: WoT. User b builds trust paths towards target k along the existing edges of a graph.

Figure 1: TTMs. Black lines denote all trust relationships in the network, and blue dashed lines mean trust paths followed from a user to their target.

Attacks and weaknesses

Both TTMs presented have suffered from weaknesses either in the way users understand and use them, or in how their infrastructure is supposed to work. Said weaknesses are inherent to the model, not particular to an implementation, so they are much harder to counter than implementation weaknesses. Steps can be, and have been, taken to reduce the impact of them — but there is undeniably need to reduce the magnitude of simple attacks.

On the PKI-CA model

While it is undeniable their net effect on Internet security has been clearly positive, a major source of weaknesses in the TTM is incomplete user understanding. Users have been instructed to be wary of submitting confidential information to sites presenting a broken or open padlock or a warning triangle (see Figure 2(a), and to trust interactions with web sites for which the Web browser displays a closed padlock (see Figure 2(b)). While this does aid the users identify whether a connection is encrypted or not, it does not

present information regarding the trust on the other site's identity. Figures 2(c) and 2(d) show the certificate chain is only displayed after two clicks. In an important push to increase the percentage of encrypted connections over the last decade, the depth of identity verification has decreased, and zero- cost CAs have become available. Browsers have removed the visual distinction between Domain Validation and Extended Validation certificates as they were never clearly understood [9-11].By 2018, half of all the phishing sites present valid TLS certificates [12]. CAs are required by the CA/Browser Forum to follow strict practices to protect their signing keys, and are required to be swift and open when handling any kind of security incidents [5], but over the last decade, several cases of rogue certificate issuance derived from leaked keys have been observed [13-16].Many more cases are quickly hidden and not reported, probably because of the loss of trust such an admission would imply [17]. After the Heartbleed vulnerability was disclosed in 2013, several implementations for switching TLS to TOFU-based schemes were pro- posed [18], but did not manage to get enough user traction to be adopted beyond a small group of proposers.

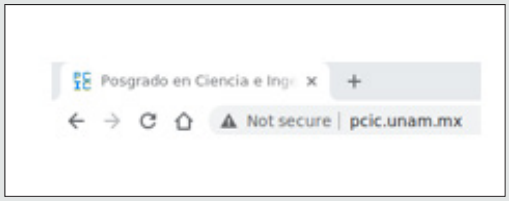
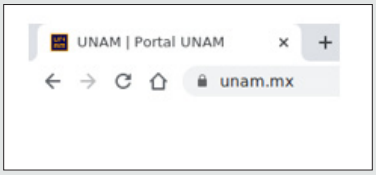
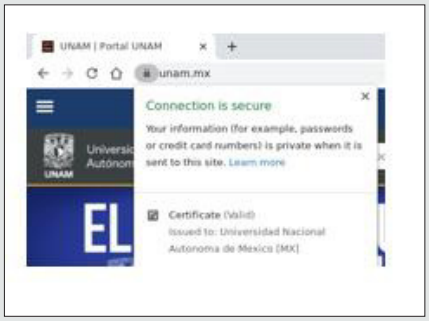


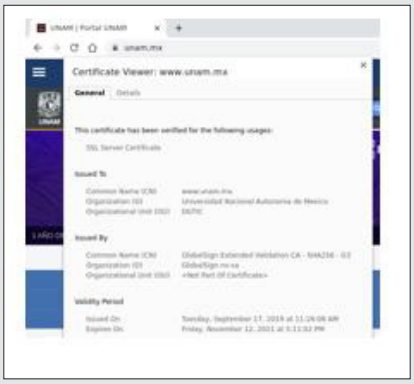
Figure 2(a): In-browser indication of an insecure.



B: In-browser indication of a secure https connection, without user intervention: only a padlock.



C: Clicking on the padlock reveals the connection is secure dialog, emphasizing the connection is encrypted (but not explaining how the target site's identity can be trusted).



D: Only a second click provides information regarding the CA that issued the certificate.

Figure 2: Getting to the trust information for the identity of a Web site requires user interaction; this simplifies browsing for the user but makes the trust model much less visible. Screenshots taken from version 90 of the Chromium Web browser.

On the WoT model

The different tools that comprise the Open PGP ecosystems are often criticized as an example of bad user interface design [19-22]. As was discussed in the previous section regarding the PKI-CA model, Open PGP is known for its encryption abilities, but many users don't pay attention at all to the authentication abilities it presents. This has led to the vast majority of keys not being at all linked to the WoT: According to Yakubov et al. 2020, at the time of their study, out of 5 217 474 keys carried by the key servers, over 84% of them existed in isolation, not having any certifications that would allow other users to establish trust in communications to them. 8.14% of the keys had a single certificate, 2.53% had only two — and only 1.71% of the keys had over 10 certificates. From the keys that are certified by other keys, only 6000 comprise the strong set (the largest bidirectionally interconnected set of keys). This means that, effectively, the WoT is useful only for roughly 1% of the keys.

The design of the SKS key server network is explicitly designed to be resistant to a threat model including national governments' interference or censorship. Data submission to the key servers is not authenticated (anybody can upload arbitrary OpenPGP data).By relying on gossip-based set synchronization protocols, the network accepts new keys and updates on existing keys, but no information can be realistically removed from it — Any server still carrying a packet not found at other servers will inject it back into the network; there is no central coordination among key servers. The protocol provides no provision for removing unwanted information. This issue has carried legal issues for key server operators, being unable to respond to deletion requests based on national privacy laws newer than the implementation [23]. The adoption of the General Data Protection Regulation (GDPR) by the European Union in 2016 makes this issue even more important. The Open PGP message format standard specifies keys should have a User ID packet that "consists of UTF-8 text that is intended to represent the name and email address of the key holder. By convention, it includes an RFC 2822 mail named, but there are no restrictions on its content" [4]. This is, there is no enforced format on the contents of User ID strings. An attack on this model is, then, the mass creation and upload of non-meaningful keys, using the WoT as some way of graffiti pad [24]. An example was presented by Lee at the Observe. Hack. Make. 2013 hacker gathering: by generating one Open PGP key per line with arbitrary ASCII characters in the field regularly used for the owner's user ID (see Figure 3(a)), he signed his own Open PGP key (Figure 3(b)), resulting in the graffiti depicted in Figure 3(c).

A: One of the throwaway identities created for the "Trolling the Web of Trust" attack

B: Beginning of the listing for the author of the attack: A regular vanity key.

C: Down the signatures list, the attack is shown as a multiline graffiti.

Figure 3: Example of the Trolling the Web of Trust attack.

The impact of this attack persé is not very high: The creation of some throwaway identities, and the use of the WoT in a way other than intended, but it resulted in highlighting a huge potential for abuse. But in 2018 –again, as a proof of concept, showing it would be impossible to properly take action against a GDPR request for information deletion [25]– a program was published that enabled the encoding of arbitrary information as key and certifications material [25], allowing the abuse of the SKS network for arbitrary file storage. While this attack has not been widely observed, its effects can be devastating on the key server network, as it potentially becomes a distributed, append-only media, with no content removal facilities. If files deemed illegal to be possessed were to be uploaded in this way, this could make many server operators to shut down their servers, with a reasoning similar to [23]. But while the above described exercise is not malicious, it does lead to a crippling attack known as certificate flooding or certificate poisoning.

Open PGP certificate poisoning

If Alice has key k_A and wants to communicate with Bob, who has key k_B and the certificate chain C_{k_B} Alice connects to the key server network and requests for bob@example.org. She verifies the results and imports the key into her local keyring. As Alice checks thoroughly to ensure that K_B is certified by their mutual friends Charly (C) and Diana (D). Open PGP keys are also self-certified, as the self-signature carries information such as the validity period. She then introduces herself to Bob. At this point, we have:

$$C_kB = k_B, cert_kB \to kB, cert_k \to kB, cert_kD \to kB$$

Mallory wants to disrupt the communication between them, so she creates thousands of throwaway keys with no meaningful information; they do not even lead back to Mallory. So Mallory controls:

$$kM1, kM2, kM3, \ldots, kM9999, kM10000$$

Mallory proceeds to certify k_A with all of her throwaway keys and uploadsthe result to the key server, so that a request for k_A will now return a substantially larger result:

$$C_kA = k_A, certk_A \to k_A, cert_kC \to kA, cert_kD \to kA, cert_kM1 \to k_A, cert_kM2 \to k_A, cert_kM3 \to k_A, cert_kM9999 \to k_A, cert_kM10000 \to k_A$$

Open PGP keys typically measure few kilobytes; very well connected keys (having many certificates) can reach the few hundreds. At this point, k_A will measure tens or hundreds of megabytes and has become poisoned unusable. When Bob attempts to get k_A his Open PGP client will be faced with orders of magnitude more information to what it is designed to handle. Observed failures from this range from the program freezing to corruption in Bob's local key store. Alice's key cannot be used anymore, and she has to migrate to a new k'_A but she will also have to meet face to face to cross-verify identities to rebuild trust and enter again the WoT. And besides that, once k'_A gained enough signatures to be useful, Mallory can repeat her attack.

While few actual such attacks have been reported [26], they are a looming threat to the Open PGP community as a whole, and are part of the reasons the key server network is perceived as dying [24].

Ways forward

There are several proposals on changes to different aspects of key server operations that in some way might lead to fixing the described situation; they can all be categorized under the broad suggestions presented by [26]. While said document is merely enunciatively of different ways the stated problem could be tackled, it does not implement or analyze in detail any of the presented alternatives. It is, however, a very valuable starting point against which other analyzed works can be measured to. We have categorized works tackling the problems here described as follows:

Key discovery mechanisms: Given the attacks involve abusing the permissive trust scheme presented by key servers, the following proposals replace it by other mechanisms:

a) Wouters 2016 is a standard for implementing DNS-based Authentication of Named Entities (DANE) for Open PGP, where keys can be queried as specific records on the DNS zone corresponding to the mail address. DANE requires, however, the mail provider to present this centralized facility, not only becoming a single point of failure, but also requiring the domain owner to take interest in providing this facility; nowadays, with the amount of mail users in large-scale mail providers such as Google's Gmail, a large amount of e-mail users would be effectively shut off this mechanism.

b) Koch 2021 proposes a Web Key Directory (WKD), presenting keys under the mail address' domain (as DANE does), but with provisions for autonomous updates by users, although it still works in a centralized way. It shares many of DANE's caveats.

c) Walfield and Koch 2016 presents an adequation of Gnu PG, the leading Open PGP implementation, to support the Trust On First Use model (TOFU), reasoning that most users don't really use the WoT model, and that while theoretical strength of WoT is better, TOFU presents a better practical protection.

Server synchronization mechanisms: Recognizing part of the issue is Gossip's impossibility to remove or stop serving given keys or certificates, the following works present different synchronization schemes:

a) Yakubov et al. 2020 suggests using an Ethereum-derived block chain for representing changes in the key server data. This solution can address propagation times of keys to the whole network and better bringing up to date servers that have fallen behind on the synchronization. While the key server operation would thus retain its decentralized way, it does introduce an administrator account that could remove toxic information from the block chain; this is often seen as antithetical to the decentralized philosophy of Open PGP.

b) Rucker 2017 presents a lighter, more efficient protocol than Gossip for synchronizing key servers, through the use of Invertible Bloom Filters (Epstein et al. 2011), allowing for more frequent polling for updates.

c) Lee 2019 presents a non-formal paper arguing the key server network state has reached a point of no return and identifies poisoned certificates as the "nail in SKS's coffin". This paper introduces the Hagrid verifying key server that among other properties, makes it impossible for keys to be poisoned by discarding all signatures from the server data — it allows for users to query for keys, but does away with all needed information for establishing a WoT.

Moving away from Open PGP: Open PGP can be characterized as belonging to a different era, and new protocols have been proposed to replace it, with different starting assumptions and threat models.

a) Borisov, Goldberg, and Brewer 2004 introduces the Off-the-Record (OTR) protocol, aimed at low-latency message-oriented interactions, such as instant messenger. It implements perfect forward secrecy, very short-lived session keys and explicit reputability. The trust model is TOFU, but with the possibility to authenticate a peer's key off-band.

Conclusions

This review illustrates the issues faced by the transitive trust model as implemented by the Open PGP standard, the most widely used encryption model, and key for the interaction of several large-scale geographically distributed software development projects. The problem presented is still an open issue to work on, and this article is limited to sketching the ground for study. Our working hypothesis is that it is possible to implement a key server network protocol that preserves the main characteristics of the Web of Trust transitive trust model, by performing relatively small modifications to the existing HKP key server interaction model.

References

1. Rescorla, Eric (2018) The Transport Layer Security (TLS) Protocol, Version 1.3. In: Internet Engineering Task Force (IETF) 8446: 1-160.
2. Finley, Klint (2017). Half the Web Is Now Encrypted. That Makes Everyone Safer.
3. Zimmermann, Philip (1999). Why I Wrote PGP.
4. Callas, Jon et al. (2007) Open PGP Message Format. In: Internet Engineering Task Force (IETF) pp. 4880.
5. CA/Browser Forum (2021) Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. Tech. rep. 1.7.4. CA/Browser Forum 99.
6. Jøsang, Audun (1999). An Algebra for Assessing Trust in Certification Chains.
7. Horowitz, Mark (1997). PGP public key server. MA thesis. MIT.
8. Shaw, David (Mar. 2003). The OpenPGP HTTP Keyserver Protocol (HKP). Internet-draft-shaw-openpgp-hkp-00: 1-9.
9. Hofmann, Johann (Aug. 2019). Intent to Ship: Move Extended Validation Information out of the URL bar.
10. O'Brien, Devon (2019) Upcoming Change to Chrome's Identity Indicators.
11. Thompson, Christopher et al. (2019) The web's identity crisis: understanding the effectiveness of website identity indicators. In: 28th USENIX Security Symposium (USENIX Security 19): 1715-1732.
12. Krebs, Brian (2018) Half of all Phishing Sites Now Have the Padlock.
13. Nightingale, Jonathan (2011) Fraudulent.
14. Prins J. R. (Sept. 2011) Interim Report DigiNotar Certificate Authority breachOperation Black Tulip p. 2- 13.
15. Espiner Tom (2012) Trustwave sold root certificate for surveillance.
16. Amann Johanna (2017) Mission Accomplished? HTTPS Security after DigiNotar. In: Proceedings of IMC'17 1(3): 325-340.
17. Berkowsky, Jake A, Thaier Hayajneh (2017) Security issues with certificate authorities. In: 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference pp. 449-455.
18. Gabor X and Tjebbe Vlieg (2013) Public Key Pinning for TLS Using a Trust on First Use Model p. 1-14.
19. Whitten, Alma and J Doug Tygar (1999) Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0: 348 pp. 169-184.
20. Sheng, Steve (2006) Why Johnny still can't encrypt: evaluating the usability of email encryption software. In: Symposium on Usable Privacy and Security. ACM: 3-4.
21. Woo, Wing Keong (2006) How to exchange email securely with Johnny who still can't encrypt. PhD thesis. University of British Columbia, USA.
22. Ruoti, Scott (2015) Why Johnny still, still can't encrypt: Evaluating the usability of a modern PGP client.
23. Pramberger, Peter (2010) Keyserver pramberger at terminating.
24. Lee, Micah F (2014) Trolling the Web of Trust.
25. Yakamo, K (2018) Are PGP key-servers breaking the Law under the GDPR.
26. Kahn Gillmor, Daniel (Aug 2019). Abuse-Resistant OpenPGP Keystores. Internet- Draft -dkg-openpgp-abuse-resistant-keystore-4: 1-58.
27. Borisov, Nikita, Ian Goldberg, Eric Brewer (2004). Off-the-record communication or why not to use PGP. In: Proceedings of the 2004 ACM workshop on Privacy in the electronic society pp. 77-84.
28. Eppstein, David (2011) What's the Difference? Efficient Set Reconciliation without Prior Context. In: Proceedings of the ACM SIGCOMM 2011 Conference. SIGCOMM '11. Toronto, Ontario, Canada: Association for Computing Machinery pp. 218-229.
29. Finley, Klint (2017). Half the Web Is Now Encrypted. That Makes Everyone Safer.
30. (1999) Network and Distributed Systems Security Symposium: The Internet Society.
31. Koch, Werner (May 2021). OpenPGP Web Key Directory. Internet draft-koch-openpgp-webkey-service-12: 1-17
32. Minsky, Yaron and Ari Trachtenberg (2002) Practical Set Reconciliation. In: 40th Annual Allerton Conference on Communication, Control and Computing pp. 248.
33. Rucker, Alexander (2017) An Efficient PGP Key server without Prior Context p. 1-6.

34. Walfield, Neal H and Werner Koch (2016) TOFU for OpenPGP. In: Eu- roSec'16: Proceedings of the 9th European Workshop on System Security p. 1-6.

35. Wouters, Paul (2016) DNS-Based Authentication of Named Entities (DANE) Bindings for Open PGP p. 1-20.

DOI: 10.32474/CTCSA.2022.02.000134