Investigating the OpenPGP Web of Trust

Alexander Ulrich[1], Ralph Holz[2], Peter Hauck[1], Georg Carle[2]

[1] Diskrete Mathematik
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
{ulricha,hauck}@informatik.uni-tuebingen.de

[2] Network Architectures and Services
Fakultät für Informatik
Technische Universität München
{holz,carle}@net.in.tum.de

Abstract. We present results of a thorough analysis of the OpenPGP Web of Trust. We conducted our analysis on a recent data set with a focus on determining properties like usefulness and robustness. To this end, we analyzed graph topology, identified the strongly connected components and derived properties like verifiability of keys, signature chain lengths and redundant signature paths for nodes. Contrary to earlier works, our analysis revealed the Web of Trust to be only similar to a scale-free network, with different properties regarding the hub structure and its influence on overall connectivity. We also analyzed the community structure of the Web of Trust and mapped it to social relationships. Finally, we present statistics which cryptographic algorithms are in use and give recommendations.

Keywords: Web of Trust, OpenPGP, GnuPG, PGP, Community Structure

1 Introduction

Pretty Good Privacy (PGP) and the GNU Privacy Guard (GnuPG) are implementations of OpenPGP (RFC 4880 [1]). Instead of a hierarchical trust architecture with Certification Authorities as in X.509, OpenPGP employs a certification model where any entity can certify another entity. This results in a so-called Web of Trust (WoT).

In this paper, we describe the results of a thorough investigation of the Web of Trust as established by OpenPGP users. We employed graph analysis to find answers to security-related issues in the WoT. Our contributions are the following. First, we analyzed the OpenPGP WoT's graph components and identified its macro structures. We will see that this is a prerequisite for more detailed analyses as there is a single most important component. Second, we analyzed

⋆ The original source of this publication is www.springerlink.com:
http://www.springerlink.com/content/634041022x670r30/

the 'usefulness' of the WoT for its users. We investigated properties like the length of certification chains, redundant paths, the Small World effect in the WoT and mutual signatures. Third, we determined how robust the WoT is to changes like the random or targeted removal of keys, which can be the result of key expiration, revocation or even attack. Fourth, as the WoT shows properties of a social network, we used State-of-the-Art algorithms to detect community structures and map them to social relations. Finally, we analyzed which cryptographic algorithms are in use and whether this is problematic or not.

The remainder of this work is organized as follows. The following section provides background to OpenPGP. Section 3 describes our methodology. Section 4 presents our results concerning the WoT's usefulness and robustness. The results of our analysis of the community structure are presented in Section 5. Section 6 presents statistics about key properties. Section 7 puts this work into the context of previous, related publications and highlights the differences from our work.

2 Background

Essentially, the WoT is a user-centric and self-organized form of PKI. A user in OpenPGP is identified by a *user ID*, a data structure that contains a user name and e-mail address. Every user ID is associated with a public/private key pair (either DSA/ElGamal or RSA). Users 'issue certificates' to each other by signing another key (i. e. user ID and public key) with their private keys. The exact mechanism of creation of the WoT is not fully known, but it is commonly agreed that personally established contact between users plays a major role, particularly organized events like Keysigning Parties at conferences and meetings. OpenPGP keys are frequently uploaded to a network of key servers. These use the Synchronizing Keyservers (SKS) protocol for synchronization. A snapshot contains a complete history of the network: keys cannot be deleted from an SKS server and timestamps of key creation, signature creation, expiration dates and recovation dates are stored.

The advantages and disadvantages of different PKI structures and trust models have been discussed, among others, by Perlman [2] and Maurer [3]. In contrast to the hierarchical X.509, which is said to suffer from insufficient Certification Authority (CA) practices and insufficient control over intermediate CAs [4], the situation is different in OpenPGP. Firstly, certificates are not verified by following a certification chain from some Root CA (with the chain already known in advance[3]), but by finding a certification path from the own key to the key that is to be verified as belonging to some entity. Secondly, OpenPGP uses a trust metric to allow users to assess the trust in a key-entity binding. There are two notions of 'trust': 'Introducer trustworthiness' refers to how much another user is trusted to apply care when verifying an identity. This value is determined and stored locally for every *locally* known user ID. 'Public-key trustworthiness'

[3] E. g., most HTTPs servers are configured to send the full chain in the SSL/TLS handshake.

is the degree to which a user claims to be sure of a key-entity binding. This value is stored as part of a signature. Before using someone else's public key, users must determine the key-entity binding and assess whether it's likely to be correct. Different trust metrics can be applied here. GnuPG, for example, uses a default setting that focuses on introducer trustworthiness: this must either be 'full' for all keys on the certification path, or there must at least be three redundant certification paths to the key in question. Also, a certification path must not be longer than 5 keys. Trust in OpenPGP thus relies on social relations for identify verification; ideally a WoT should model real-world relationships. CAs are not forbidden in OpenPGP – they are merely a special kind of user. While very flexible, this trust model is very demanding on the user. OpenPGP's model can thus be viewed to be more focused on the local 'environment' of a user – it is infeasible for a user to determine introducer trust for everyone in the WoT. A user can only make reasonable assessments about keys to which paths are short, and lead over social contacts. This also helps with the 'Which John Smith?' problem: looking for the key of a certain 'John Smith' is much easier if it is known that John Smith should share some of one's own contacts.

The open nature of the WoT could lead one to speculate whether large-scale attacks on the WoT are possible, where a malicious entity certifies a large number of keys to trick others. However, this attack is much more difficult than it seems. Assume Alice wants to verify a fake key for the identity Bob, which has only been signed by a number of false identities signed by Mallory. Alice must establish a certification path to the 'fake Bob key' using the faked signed keys. These faked keys would *only* be used in a path search if Alice has manually and explicitly set a trust value for Mallory *and* the false identities. As setting introducer trust is a manual operation, it is unlikely that Alice would assign the needed trust to unknown and 'strange' entities. For this reason, we view it as an unlikely attack.

Also note that multiple keys per user is not uncommon and not evidence of such an attack: one might for example wish to use different keys for business and private matters, or for different levels of security. Multiple (non-revoked, non-expired) keys for one person/entity occur quite commonly in the key database without any evidence for malicious behavior.

From these considerations, we can thus derive several important properties a 'good' WoT must exhibit. It must allow to find certification paths between many keys, otherwise it is not useful. The length of paths is essential: short paths reduce the number of entities on the path that a user has to trust and thus increase a user's chances of accurate assessment of key authenticity. Giving and receiving many signatures is important, too: it increases the chances of several redundant paths between nodes, which is beneficial for GnuPG's trust metrics. It also means that removal of a key has little impact on reachability, which increases the WoT's robustness. Finally, a good WoT should model social relations and social clustering well: where 'communities' of users exist, chances of being able to accurately assess trustworthiness of users within the same community increase.

Total number of keys	2,725,504
Total number of signatures	1,145,337
Number of expired keys	417,163
Number of revoked keys	100,071
Number of valid keys with incoming or outgoing signatures	325,410
Number of valid signatures for the latter set of keys	816,785

Table 1: Our data set.

3 Methodology

In this section, we describe how we extracted the graph topology and summarize the metrics we used in the graph analysis.

3.1 Graph Extraction and Analysis

We modified the SKS software to download a snapshot of the key database as of December 2009.

Table 1 shows properties of our data set after our extraction. The data set contains about 2.7 million keys and 1.1 million signatures. Of these, about 400,000 keys were expired, another 100,000 revoked. About a further 52,000 keys were found to be in a defective binary format. *The actual WoT*, which consists only of valid keys that have actually been used for signing or have been signed, is made up of *325,410 keys* with *816,785 valid signatures between them.* Consequently, the majority of keys in the data set is not verifiable (no signature chains lead to them) and does not belong to the WoT. Note that the data set contains only keys from key servers. We cannot know the number of unpublished keys.

When representing the WoT as a graph, we represented keys as nodes and signatures as directed edges. This was a deliberate choice. An alternative would have been to map keys to individual persons. However, such a mapping is not easy to define due to changes of e-mail addresses, spelling of names and the use of pseudonyms. Ultimately, it is keys that sign other keys, and we thus chose to analyze a key-based graph.

3.2 Terms and Graph Metrics

We briefly describe terms and metrics that we use in our analysis of the WoT. For precise definitions, we refer the reader to the Appendix.

Strongly connected components (SCCs) A strongly connected component is a maximally connected sub-graph of a directed graph where there is at least one directed path between every node pair u, v. Note that the paths from u to v and v to u may incorporate different nodes.

Distances, Eccentricity, Radius and Diameter The *distance* between two nodes is the length of the shortest path between them. The *characteristic distance of the graph* is the average over all distances in the graph. *Eccentricity* is a node property that indicates the distance to the node farthest away from this node in the graph. *Graph radius* is defined as the *minimum over all eccentricities* and the *diameter* is defined as the *maximum over all eccentricities*.

Neighborhoods A *node v's neighborhood* is the set of all nodes for which the distance from v is at most a certain value.

Clustering Coefficient The *clustering coefficient* is a measure of transitivity in a graph. It indicates the probability that two neighbors of a node are themselves neighbors, i.e. have an edge between them.

Correlation of Node Degrees Pastor-Satorras et al. [5] defined a measure for the correlation of node degrees in a function knn. It determines whether nodes with similar degrees have edges between them. The *assortativity coefficient* [6] is a similar measure. It measures how many nodes with high degree are connected mainly to other nodes with high degree.

4 Results

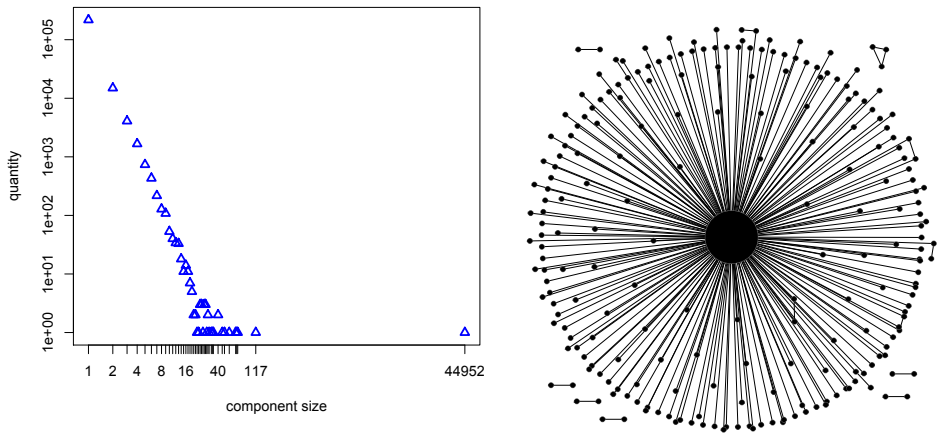
We present the results of our analysis.

4.1 Macro Structure: strongly connected components (SCCs)

Within SCCs, there is at least one signature chain between every key pair. SCCs are thus important for participants of the WoT: mutual verification of key authenticity is only possible for participants within the same SCC. An optimally meshed WoT should be one giant SCC.

We computed the SCCs of the graph, and found 240,283 SCCs in the WoT. However, more than 100,000 of these consisted of a single node and about 10,000 SCCs consist of node pairs. The largest SCC (LSCC) consists of about 45,000 nodes. The remaining SCCs mostly have a size between 10 and 100 nodes. Figure 1 (a) shows the distribution. The SCCs can be arranged in a star formation around the LSCC in the middle (Figure 1 (b)).

Many SCCs have *uni-directional* edges to the LSCC, but extremely few have edges between each other. Out of all smaller SCCs, about 18,000 nodes show a uni-directional edge into the LSCC, making it (in principle) possible for such a key to verify keys from the LSCC. In the other direction, 92,000 keys outside the LSCC are reachable from a key within the LSCC. We found three interesting hubs in the LSCC and one regional particularity. The German publisher Heise, CACert and, until recently, German DFN-Verein operate or have operated CAs to sign keys. Together, they have signed about 4,200 keys in the LSCC. The



(a) (b)

Fig. 1: (a) Size distribution of SCCs. (b) Plot of SCCs down to a size of 8.

Heise CA alone has, in total, signed 23,813 keys – yet of these only 2,578 are in the LSCC.

This SCC structure gravely impacts the usability of the WoT. First of all, the large number of smaller SCCs means that even among those users who have made the effort to upload their keys to a key server, most do not participate actively in the WoT. Otherwise, their SCCs would already have merged with the LSCC (one mutual signature is enough). This is also emphasized by the following comparison. The ratio of edges:nodes in the LSCC is 9.85; the same ratio for the total WoT is 2.51. Signature activity in the LSCC must thus be much higher than in the rest of the WoT. However, strong user activity is very desirable to achieve a better meshing in the WoT.

Second, a high percentage of participants in one of the smaller SCCs are unable to verify most keys in the WoT. The LSCC is really a structure of paramount importance: the keys in the LSCC constitute only 14% of the keys in the WoT, but only the owners of these keys can really profit in a significant way from the WoT. They can build signature chains to all keys in the LSCC plus to twice as many keys outside of the LSCC. Thus, a recommendation for new participants would be to obtain a signature from a member of the LSCC as early as possible to make their key verifiable. A good choice is also to get a (mutual) signature of one of the CAs in the LSCC. With such a signature, paths can be built to all keys in the LSCC, plus to a large number of keys outside the LSCC that are only reachable via the CA. This emphasizes that a WoT can benefit from CAs.

The remainder of our analysis focuses on the LSCC as the most relevant component for participants.

4.2 Usefulness in the LSCC

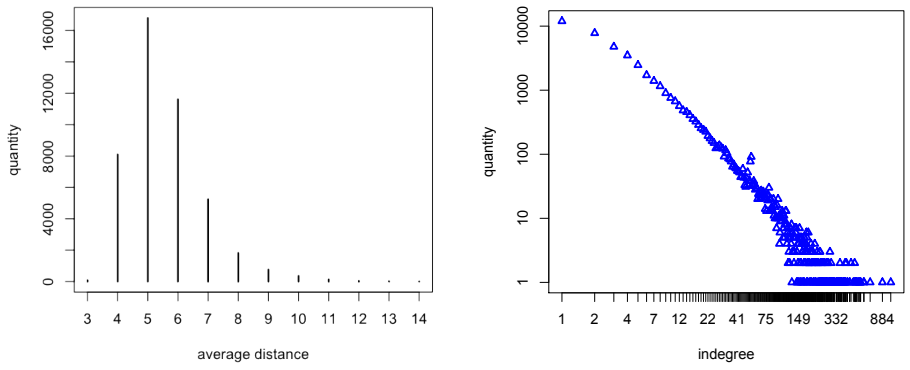
'Usefulness' is a term that is difficult to express formally. It can be defined in several dimensions. An obvious one is how many keys are verifiable from a given key, and how many paths to other keys can be found from the given key. The higher these numbers are, the more useful the WoT is from the perspective of this key. Recall that introducer trustworthiness is not stored in the signatures: the following discussion thus relates to *upper bounds*.

Distances We first analyzed distances between keys in the graph. The *average distances between nodes* in the LSCC (see Figure 2(a)) range between 4–7, which is at best just below GnuPG's limit (path length 5), but exceeds it at worst. The *eccentricity* in the LSCC is much higher: it is almost exclusively between 26–31. To determine the implications of this for usefulness, we identified how many keys are reachable from a given key within a certain distance.

We computed the set of verifiable keys as the nodes in a *h-neighborhood* for $h = 1, .., 5$ (see Definition 5 in the Appendix). Figure 3 shows the CDF of *h-neighborhoods*. For the 2-neighborhood, we see a steep incline, from which we can conclude that this neighborhood must be relatively small for all nodes. The size of the neighborhoods grows considerably for increasing h. For $h = 3$, the third quartile is about 3,300. For $h = 4$ and $h = 5$, it becomes 16,300 and 30,500, respectively.

Our findings indicate that signature chains within GnuPG's restrictions are sufficient to make a very large fraction of the keys in the LSCC verifiable. This is a good result for usefulness and shows that the LSCC is quite well meshed. However, for $h = 5$, the maximum number of reachable keys we found was 40,100. This means that, on average for all keys, there will be almost 5,000 keys (a tenth of the LSCC) to which no path at all can be found within GnuPG's restrictions.

Small World Effect and Social Links The size of 5-neighborhoods shows that paths are frequently very short. A possible explanation for this is a Small World effect, which – following [7] – can be informally understood to be the phenomenon that the average path length between any two nodes is significantly shorter than could be expected by judging from graph radius and diameter. A high clustering coefficient is often viewed as indicative. We investigated this in the LSCC. As there does not seem to be a universally accepted definition of the clustering coefficient for directed graphs, we reduced the directed graph to an undirected one (omitting the direction of edges and merging duplicates). The clustering coefficient we computed is $C = 0.46$. This indicates that, *on average*, roughly half of all neighbors of a node have edges between them. The value is of the same order as described in [7] for social networks with strong clustering. The characteristic distance in the LSCC is 6.07, while the diameter of the graph is 36 and the radius 16. Our finding is that the LSCC does indeed show a Small World effect. This indicates social clustering. Together with the short paths, this would make trust assessments easier for users. We explore the social nature of the WoT further in Section 5.



(a) (b)

Fig. 2: Distribution of (a) average distances, (b) indegree in LSCC.

Node Degrees Recall that GnuPG's trust metrics view redundant and distinct signature chains as beneficial for a key's trustworthiness. A high node indegree thus means that the corresponding key is more likely to be verifiable by other keys. A high outdegree increases the likeliness to find redundant signature chains to other keys. We computed the average indegree (and outdegree) in the LSCC as 9.29. However, as can be seen in Figure 2(b), the distribution of indegrees in the LSCC is skewed. The vast majority of nodes have a low indegree (i. e., 1 or 2). The result for the outdegrees is very similar: as can be seen in Figure 4(a), there is a positive correlation between indegree and outdegree of a node. The plot for outdegrees is indeed so similar to the one for indegrees that we omitted it here. About a third of nodes in the LSCC have an outdegree of < 3. Together, these results mean that the WoT's usefulness has an important restriction: many nodes need to rely on single certification paths with 'full' introducer trust and cannot make use of redundant paths.

Mutual Signatures (Reciprocity of Edges) If many WoT participants cross-signed each other, this would be a great improvement in overall verifiability of keys. We computed the reciprocity of edges, i. e. the fraction of uni-directional edges to which there exists a uni-directional edge in the other direction. The LSCC has a reciprocity value of 0.51. This shows that there is room for improvement: the LSCC would profit much if more mutual signatures were given, which would of course also strengthen indegree and outdegree and shorten distances.

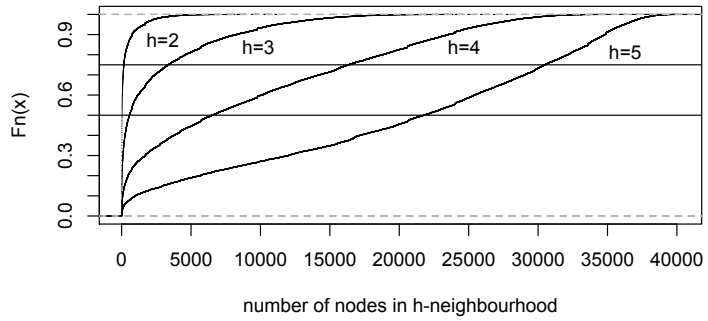


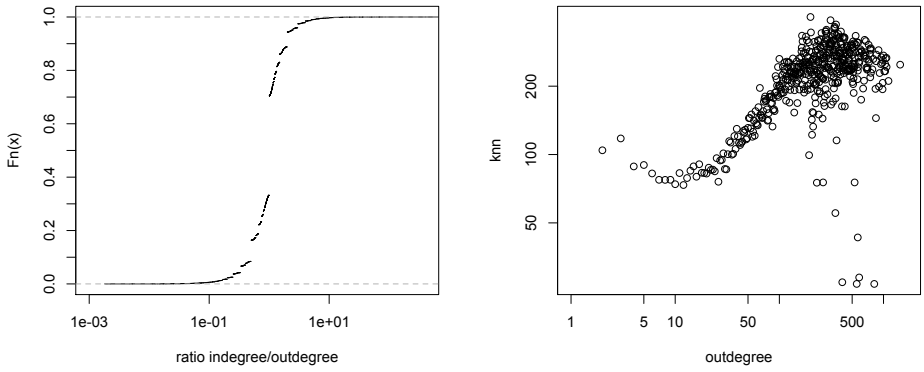
Fig. 3: CDF of reachable nodes due to h-neighbourhoods.

4.3 Robustness of the LSCC

The robustness of the LSCC is also an interesting topic: how is the LSCC connected internally, and hence how sensitive is it to removal of keys? In the context of OpenPGP, the random removal of a node can be the result of an event like key expiration or revocation, which invalidates paths leading over the key in question. These events can and do occur in practice. Targeted removal of a key, however, is very hard to accomplish as SKS never deletes keys and stays synchronized. An attacker would need an unlikely high amount of control over the SKS network to make a key disappear.

Scale-Free Property Scale-freeness in a graph means that the node degrees follow a Power Law. Connectivity-wise, scale-free graphs are said to be robust against random removal of nodes, and vulnerable against the targeted removal of hubs (which leads to partitioning). This is usually explained by the hubs being the nodes that are primarily responsible for maintaining overall connectivity [8]. We thus first investigated to which extent the WoT shows this property.

The double-log scale in Figure 2(a) could lead one to the conclusion that the distribution of node degrees follows a Power Law. However, Clauset et al. argued in [9] that this is not indicative and methods like linear regression can easily be inaccurate in determining a Power Law distribution. We followed the authors' suggestion instead and used the Maximum Likelihood method to derive Power Law coefficients and verified the quality of our fitting with a Kolmogorov-Smirnov test. [9] gives a threshold of 0.1 to safely conclude a Power Law distribution. Our values for indegrees and outdegrees were 0.012 and 0.011, respectively. As this is off by a factor of 10, our conclusion is that a Power Law distribution is not plausible. Consequently, the graph cannot be scale-free in the strict sense



(a) (b)

Fig. 4: (a) CDF of ratio indegree:outdegree in LSCC. (b) Correlation of node degrees according to Definition 8 (see Appendix): average outdegree (knn) of neighbors of nodes with degree k.

of the definition. This finding is contradictory to earlier works by Boguna et al. [10] and Capkun et al. [11].

The question is yet whether the graph is still similar to a scale-free one. Apart from high variability of node degrees, a set of high-degree nodes that act as inter-connected hubs are characteristic for scale-free graphs [8, 12]. The positive correlation between the degree of nodes and the average degree of their neighbors (Figure 4(b)) suggests that nodes with high outdegrees do indeed connect to other such nodes with high probability. To bolster our finding, we computed the assortativity coefficient (see Appendix A.4) and obtained a value of 0.113. This is similar to what has been computed for other social networks with a hub structure [7]. Our conclusion is thus that the graph is similar to a scale-free one and exhibits a hub structure, but is not scale-free in the strict sense.

Random Removal of Nodes Based on this finding, we investigated how the LSCC reacts to random removal of nodes. We removed nodes and recomputed the size of the remaining LSCC as an indication of loss in overall connectivity. For random removal, we picked the nodes from a uniform distribution. Figure 5 shows our results. The graph is very robust against the random removal of nodes: we must remove 14,000 nodes to cut the LSCC's size by half. To reduce it to a quarter, we must remove more than half the nodes (25,000).

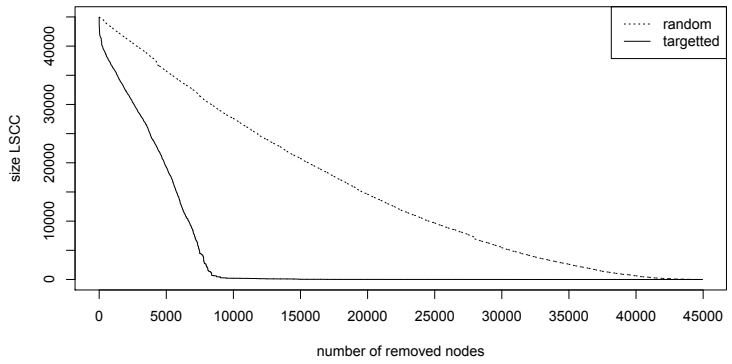


Fig. 5: Removing nodes at random and in a targeted fashion and recomputing the size of the LSCC.

The conclusion here is that events like key expiration or revocation do not greatly influence the robustness, and consequently the usability, of the WoT.

Targeted Removal of Nodes and CAs For targeted removal, we chose nodes with highest degrees first. The graph was more robust than expected. When we removed all nodes with a degree of more than 160 (240 nodes), the size of the LSCC was still 40,000. Only when we proceeded to remove all nodes with a degree of more than 18 ($\sim$ 5,000 nodes), the LSCC was half its size. Removing 2,500 more nodes, we finally cut the LSCC down to about 1/9 of its original size. This means that nodes with lower degrees (< 18) play a significant role in overall connectivity (although the decay of the LSCC is quite pronounced after they are also removed). The rather slow decay stands in contrast to the rapid decay upon removal of the best-connected nodes that is commonly observed in scale-free networks. Targeted removal of keys does not affect the WoT greatly, and is not an efficient attack. The hub structure is not the single reason for highly meshed connectivity in the WoT.

We decided to strengthen the attack by removing the keys of the three CAs. Our finding was similar: the LSCC split into one LSCC of size 42,455 and 1,058 very small SCCs. This means that the CAs, although beneficial in making keys verifiable, are not responsible for holding the LSCC together. The characteristic distance of the new LSCC remained almost unchanged (6.25); radius, diameter and eccentricity remained the same. This means that path properties did not change, either. Our conclusion here is that attempting to selectively remove keys from keyservers, even shutting down CAs, would not change the WoT's properties significantly. It is very robust in this respect.

5 Community Structure of the Web of Trust

We know from Section 4 that the WoT shows the small-world property, which hints at social clustering. Newman and Park also noted that a high degree of clustering is typical for social networks [13]. Fortunato [14] calls such subsets of nodes 'communities' if the nodes have high intra-connectivity in their subset, but the subset as such shows a much lower connectivity to nodes outside. Social clustering can make the WoT more powerful: it is more likely that members of a cluster know each other at least to some extent and can thus better assess the trustworthiness of particular keys.

Community Detection We analyzed the WoT with State-of-the-Art algorithms for community detection to determine whether a pronounced community structure exists and can be mapped to 'real-world' relationships. Also, we attempted to find whether signing events like Key Signing Parties can be identified in the graph. Unfortunately, algorithms for community detection are often defined for undirected graphs. Also, signatures store little information that helps with identifying social links and events in time. We decided to use DNS domains in user IDs and timestamps of signature creation as a basis. As an algorithm for a directed graph, we chose the one by Rosval et al. [15]. For undirected graphs, we chose the algorithms by Blondel et al. [16] and COPRA [17], based on suggestions in [18]. COPRA allows overlapping communities, but is non-deterministic. We ran it 10 times and computed differences. As a measure for the quality of a dissection, we used *Modularity* [19], which relates the amount of intra-cluster edges of a graph with communities to the expected value for a graph without communities. Note that the definition for overlapping communities is different, so the values for COPRA and BL cannot be compared directly.

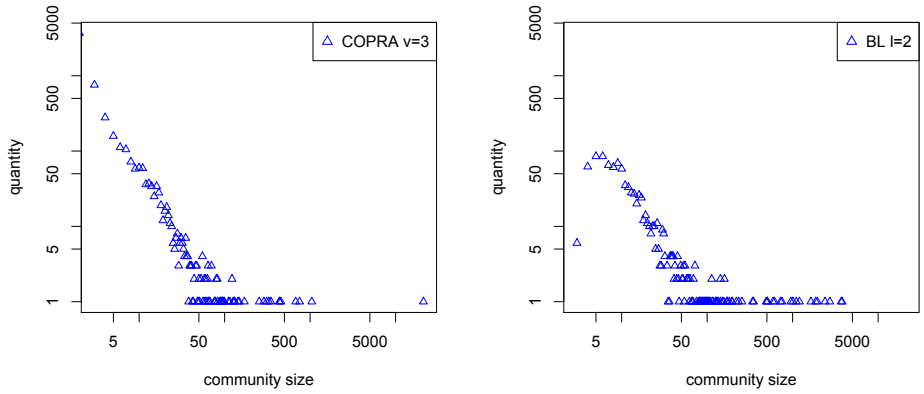
Only the algorithms by Blondel et al. and COPRA yielded useful results. The algorithm by Rosval et al. computed a dissection into 2,869 communities, almost all of them without any intra-cluster edges. We considered these results unreliable and ignored them in our subsequent evaluation.

Blondel et al. and COPRA Table 2 shows the results of dissections with Blondel et al. (BL) and COPRA for communities of size > 3. Both BL and COPRA are configurable: BL can be repeated in iterative phases and COPRA requires a (user-chosen) parameter v to reflect the degree of overlapping. For BL, phase 2 yielded the best results (plausible number of communities, high modularity). For COPRA, values of v up to 3 were found best. We know from [20] that modularity values > 0.3 indicate a significant community structure. Depending on the algorithm and chosen parameters, between 94% (COPRA) and 99% (BL) of nodes in the LSCC belonged to such a community.

BL and COPRA agree on the same orders of magnitude with respect to the number of communities and nodes therein. The high modularity values and the general shape of community distributions by size (see Figure 6) are also similar. Most communities are very small, but a significant number of large or very large

Method	Modularity	Communities found (size > 3)
BL ($l = 2$)	0.70	936
BL ($l = 5$)	0.71	186
COPRA ($v = 1$)	(0.78)	1,421
COPRA ($v = 3$)	(0.79)	1,354

Table 2: Dissection of the LSCC into communities: algorithms BL and COPRA.



(a) (b)

Fig. 6: Distribution of communities by size.

communities exist. Similarities, however, end here. COPRA indicates one extremely large community of 19,000-21,000 members. BL finds more communities of medium size (100-500) and mid-large size (500-5,000). To further investigate this, we analyzed how communities are connected. COPRA found that most small communities are clustered around the largest community and mostly only link to this community. BL found several large communities to which the smaller communities connect.

Mapping to Domain Names and Keysigning Parties We analyzed how the community dissections mapped to top-level and second-level domains (TLDs and SLDs) in the user IDs. We say a community is *dominated* by a domain if at least 80% of its nodes belong to that domain. We say a community can be *assigned* to a domain if at least 40% of its nodes belong to it.

Table 3 shows the results. For both BL and COPRA, we found that a large percentage of communities are dominated by a top-level domain: between 47% and 58%. Only if a community was not dominated, we checked if it could at least

Method	dominated by TLD	assignable to TLD	dominated by SLD	assignable to SLD	signatures within 30d
BL ($l = 2$)	499 (53%)	417 (45%)	41 (4%)	254 (27%)	115 (12%)
BL ($l = 5$)	85 (47%)	85 (47%)	15 (8%)	38 (21%)	26 (14%)
COPRA-1	824 (58%)	564 (40%)	178 (13%)	429 (30%)	572 (40%)
COPRA-3	792 (58%)	525 (38%)	187 (14%)	425 (31%)	555 (41%)

Table 3: Community structure with respect to membership in top-level domains (TLD) and second-level domains (SLD).

be assigned. A further 38%–47% could be said to be assignable to a TLD. This result did not change much when we disregarded generic TLDs (`.com` etc.): with COPRA, 38% of communities were dominated by a country's TLD and a further 23% were still assignable. Results for BL were similar. Together, assigned and dominated communities make up by far the largest part of communities found (98% for BL-2 and COPRA, $v = 1$). However, the picture changes for second-level domains. With COPRA, only about 13% of communities are dominated by an SLD and only a further 30% of communities can be assigned to an SLD.

Keysigning Parties are events where one can expect signatures to be uploaded to key servers within a short time frame. Table 3 shows the percentage of nodes in the communities where signatures were created within a month. We find poor results for BL, but much better ones for COPRA. In about 40% of communities, the signatures were created within 30 days of each other.

Conclusion with Respect to Community Detection Concerning community detection, it is difficult to reach compelling conclusions. We provide ours as a basis of discussion. Both algorithms agreed that a large number of smaller communities exist. Given the huge number of TLDs and SLDs and given that the WoT graph spans more than a decade, the results seem statistically significant enough to conclude that the community structure does indeed capture some 'social' properties of the WoT. However, grouping by TLD is a blunt measure, and the mappings to SLDs were by far not as compelling. Our tentative conclusion is that the signing process in the WoT is indeed supported, to a traceable extent, by real-world social links. The social nature of the WoT is not a myth. At least where certification paths are short, the community structure should make it easier for users to assess the trustworthiness of a key. Beyond this result, however, community detection is yet too imprecise to offer more succinct conclusions.

6 Cryptographic Algorithms

Table 4 presents results on the use of hash and public key algorithms in the WoT. Several algorithms encountered raise security concerns: MD5 can probably be said to be an unwise choice today [21]. SHA-1, although much safer, is also

Algorithm	Occurrences
SHA1	398,849
MD5	41,700
SHA256	5,031
SHA512	2,472
SHA224	532
RIPE-MD/160	122
Signatures total	446,325

(a)

Algorithm	Occurrences
DSA-1024	36,555
RSA-1024	3,903
RSA-2048	2,408
RSA-4096	1,198
RSA-768	257
RSA-512	203
RSA-3072	96
Keys total	44,952

(b)

Table 4: Occurrences of (a) hash algorithms, (b) public key algorithms.

scheduled for phase-out [22]. RSA keys of 768 bits have been factored [23] and a length of more than 1,024 bits is recommended since 2010 [24].

Especially the comparatively high number of RSA keys with a key length of $\leq$ 1,024 bits is somewhat problematic. We investigated these keys and found that a substantial number of them appears well-connected, based on their in- and outdegrees. It seems reasonable to assume that quite a few users trust these keys as introducers, thus enabling their use in certificate chains. Although not a threat yet and possibly also not for the next few years, it opens up attack opportunities if factorization of 1,024 bits keys should become feasible [23].

7 Related Work

The OpenPGP WoT has been the subject of investigation before, albeit at other stages of its development and with a focus that was less on security-relevant properties. Capkun et al. [11] analyzed several structural aspects of the WoT of 2001. They did not investigate aspects like communities but presented a model to create similar graphs. They found a small characteristic distance and a high clustering coefficient. The authors claimed to have found a Power Law distribution for node degrees. Our own findings are that a Power Law distribution is not plausible. However, the graph is similar to a scale-free one, although its hub structure is not solely responsible for robustness. Note however that the rigid methods in [9] were generally not as widely in use then, and the graph from 2001 contained 4 times fewer nodes. Boguna et al. [10] also analyzed a PGP graph from 2001. They converted the graph to an undirected one and analyzed node degrees and clustering coefficient. They also claimed a Power Law for node degrees and determined a clustering coefficient on the same order as the one we found. The authors also applied an (older) algorithm for community detection. They claimed the community distribution follows a Power Law, too. All of the above have in common that they used significantly older data sets, and the focus was less on security issues like usefulness and robustness. Furthermore, our com-

munity dissection was conducted with more recent algorithms, with the aim of mapping communities to real-world groups. The OpenPGP community has also contributed some effort in analyzing the WoT's structure. The *wotsap* project [25] creates snapshots of the signatures in the WoT. However, it only considers the LSCC and does not store other key properties. We also found the data set to be incomplete (10% of keys missing) due to a bug. Penning [26] used the wotsap data set to determine aspects like distances, node distribution and robustness based on node removal.

8 Discussion and Conclusion

We have presented several results relating to security aspects of the OpenPGP Web of Trust. We found that only keys in the Largest SCC (LSCC) can really profit from the WoT. This severely limits the reach of the WoT to a fraction of its users: only about 45,000 keys out of 2 million can use the WoT without restrictions. A large fraction of keys in the smaller SCCs can make very little use of the WoT or none at all. However, for users with keys in the LSCC, the situation is much better. We found their certification chains to be relatively short. There is also a pronounced Small World effect. We followed this up with an investigation of the community structure of the WoT. While algorithms for community detection can capture the social groups of the WoT on a very coarse level only, the graph does exhibit a very strong community structure. Another positive aspect is that about 40,000 of 45,000 keys are reachable within GnuPG's restrictions (5 hops), and several thousand even via 3 hops or less. This is positive for the WoT as it can aid users in making better trust assessments regarding other keys that are close and in the optimal case also in the same community. The CAs we found help greatly in making keys verifiable. This is a viable option for users. Random removal of keys (e. g., due to expiration or revocation) is not a problem for the robustness of the WoT. The WoT is also very robust against targeted attacks; CAs are not fundamentally relevant for robustness.

However, we found that low indegrees and outdegrees are far too common. This reduces the number of redundant paths between keys, which means that many users would need to have 'full' introducer trust in known entities. Mutually cross-signing more often would help here.

In essence, our conclusion is that the WoT is likely to be quite an effective PKI structure *within smaller node neighborhoods, and particularly for those users that frequently sign other keys and are active in the WoT*. The cryptographic algorithms that are in use can be generally considered to be still secure. However, keys that have issued MD5-based signatures should be replaced and signatures renewed. Also, a stronger move towards key lengths of more than 1,024 bits is desirable.

Acknowledgements We would like to thank both our anonymous reviewers and Radia Perlman for their valuable input.

References

1. Callas, J., Donnerhacke, L., Finney, H., Shaw, D., Thayer, R.: OpenPGP Message Format. RFC 4880 (November 2007)
2. Perlman, R.: An overview of PKI trust models. IEEE Network **13**(6) (Nov/Dec 1999) 38–43
3. Maurer, U.: Modelling a public-key infrastructure. In: Proc. 4th European Symposium on Research in Computer Security (ESORICS '96). Volume 1146 of LNCS., Springer (1996) 325–350
4. Eckersley, P., Burns, J.: An observatory for the SSLiverse. Talk at Defcon 18. `https://www.eff.org/files/DefconSSLiverse.pdf` (July 2010) [online; last retrieved in February 2011].
5. Pastor-Satorras, R., Vázquez, A., Vespignani, A.: Dynamical and correlation properties of the Internet. Phys. Rev. Lett. **87**(25) (Nov 2001) 258701
6. Newman, M.E.J.: Assortative mixing in networks. Phys. Rev. Lett. **89**(20) (Oct 2002) 208701
7. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45**(2) (2003) 167–256
8. Albert, R., Jeong, H., Barabasi, A.L.: Error and attack tolerance of complex networks. Nature **406**(6794) (July 2000) 378–382
9. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Review **51**(4) (2009) 661–703
10. Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., Arenas, A.: Models of social networks based on social distance attachment. Phys. Rev. E **70**(5) (Nov 2004) 056122
11. Capkun, S., Buttyán, L., Hubaux, J.P.: Small Worlds in security systems: an analysis of the PGP certificate graph. In: NSPW '02: Proc. 2002 Workshop on New Security Paradigms, ACM (2002) 28–35
12. Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a theory of scale-free graphs: Definition, properties, and implications. Internet Mathematics **2**(4) (March 2005) 431–523
13. Newman, M.E.J., Park, J.: Why social networks are different from other types of networks. Phys. Rev. E **68**(3) (September 2003) 036122
14. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3-5) (2010) 75–174
15. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proc. National Academy of Sciences **105**(4) (2008) 1118–1123
16. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10) (2008) P10008
17. Gregory, S.: Finding overlapping communities in networks by label propagation. New Journal of Physics **12**(10) (October 2010) 103018
18. Lancichinetti, A., Fortunato, S.: Community detection algorithms: A comparative analysis. Phys. Rev. E **80**(5) (Nov 2009) 056117
19. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2) (February 2004) 026113
20. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6) (December 2004) 066111

21. Sotirov, A., Stevens, M., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D.A., de Weger, B.: MD5 considered harmful today. Available online. http://dl.packetstormsecurity.net/papers/attack/md5-considered-harmful.pdf (2008) [online; last retrieved in May 2011].
22. NIST: Approved Algorithms. http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html (2006) [online; last retrieved in May 2011].
23. Kleinjung, T., Aoki, K., Franke, J., Lenstra, A., Thom, E., Bos, J., Gaudry, P., Kruppa, A., Montgomery, P., Osvik, D., te Riele, H., Timofeev, A., Zimmermann, P.: Factorization of a 768-bit RSA modulus. In: Advances in Cryptology – CRYPTO 2010. Volume 6223 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 333–350
24. Barker, E., Barker, W., Burr, W., Polk, W., Smid, M.: NIST special publication 800-57 part 1, recommendation for key management - part 1: General (revised). http://csrc.nist.gov/groups/ST/toolkit/key_management.html (2007)
25. Cederlöf, J.: Web of Trust statistics and pathfinder. http://www.lysator.liu.se/~jc/wotsap/ [online; last retrieved in February 2011].
26. Penning, H.P.: Analysis of the strong set in the PGP web of trust. http://pgp.cs.uu.nl/plot/ [online; last retrieved in February 2011].
27. Brinkmeier, M., Schank, T.: Network statistics. In: Network Analysis. (2004) 293–317

A Common Terms and Graph Metrics

Based on the common notions of graph theory, we define some terms, following [27] herein. In the following, let V be the set of nodes of the graph G, with $|V| = n$. u and v indicate nodes.

A.1 Distances

Distances Between Nodes The distance d between two nodes is defined as the length of the shortest path between these two nodes.

Distances in the Graph The average distance of the graph, $\bar{d}$, is the average over all distances in the graph:

$$\bar{d} = \frac{1}{n^2 - n} \sum_{u \neq v \in V} d(u, v) \tag{1}$$

Eccentricity The eccentricity of a node u, $\epsilon(u)$, is defined as the maximum distance to another node, i.e.

$$\epsilon(u) = \max\{d(u, v) | v \in V\} \tag{2}$$

Graph Radius and Diameter The diameter of a graph is defined as the maximum over all eccentricities:

$$dia(G) = \max\{e(u)|u \in G\} \tag{3}$$

The *radius* is defined as the minimum over all eccentricities:

$$rad(G) = \min\{e(u)|u \in G\} \tag{4}$$

A.2 Node Neighborhoods

We define the *h-neighborhood* of a node v as the set of all nodes from which the distance to v is at most h:

$$N_h(v) = \{u \in V|d(v,u) \leq h\} \tag{5}$$

A.3 Clustering Coefficient

The *clustering coeffcient* indicates the probability that two neighbors of a node have an edge between them.

Let $G = (V, E)$ be the undirected graph. A triangle $\triangle = \{V_\triangle, E_\triangle\}$ is a complete sub-graph of G with $|\triangle| = 3$. The number of triangles of a node v is given by $\lambda(v) = |\{\triangle : v \in V_\triangle\}|$. A *triplet* of a node v is a sub-graph of G that consists of v, 2 edges, plus 2 more nodes such that both edges contain v. The number of triplets of a node v can be given as $\tau(v) = \binom{d(v)}{2}$. The *local clustering coefficient* of v is defined as

$$c(v) = \frac{\lambda(v)}{\tau(v)} \tag{6}$$

$c(v)$ indicates how many triplets of v are triangles. The *global clustering coefficient* of G can then be defined as:

$$C(G) = \frac{1}{|V'|} \sum_{v \in V'} c(v) \tag{7}$$

with $V' = \{v \in V : d(v) \geq 2\}$ to disallow non-defined values for $\tau(v)$.

A.4 Correlation of Node Degrees

Function *knn* as defined by Pastor-Satorras et al. Following Pastor-Satorras et al. [5], we define a measure for the correlation of node degrees:

$$< knn >= \sum_{k'} k' P_c(k'|k) \tag{8}$$

gives the average node degree of neighbors of nodes with degree k. $P_c(k'|k)$ indicates the probability that an edge that starts at a node with degree k ends at a node with degree k'.

Assortativity Coefficient The *assortativity coefficient* [6] is a measure whose purpose is similar to the function defined in Definition 8. It measures the degree of *assortative mixing* in a graph: nodes with high degree that are connected mainly to other nodes with high degree. The *assortativity coefficient* takes values between -1 and 1. Positive values indicate assortative mixing, negative ones do not. According to Newman [6], assortative mixing is a property that distinguishes social networks from other real-world networks (e.g. technical or biological ones). It can thus be used to sub-differentiate between similar graphs that show a Small World effect.