Analysis of a dynamic social network built from PGP keyrings

Robert H. Warren RHWARREN@UWATERLOO.CA
Dana Wilkinson D3WILKIN@UWATERLOO.CA
School of Computer Science, University of Waterloo, Waterloo, Canada

Mike Warnecke MWARNECKE@PSWAPPLIEDRESEARCH.COM
PSW Applied Research Inc., Waterloo, Canada

Abstract

Social networks are the focus of a large body of research. A number of popular email encryption tools make use of online directories to store public key information. These can be used to build a social network of people connected by email relationships. Since these directories contain creation and expiration time-stamps, the corresponding network can be built and analyzed dynamically. At any given point, a snapshot of the current state of the model can be observed and traditional metrics evaluated and compared with the state of the model at other times.

We show that, with this described data set, simple traditional predictive measures do vary with time. Moreover, singular events pertinent to the participants in the social network (such as conferences) can be correlated with or implied by significant changes in these measures. This provides evidence that the dynamic behaviour of social networks should not be ignored, either when analysing a real model or attempting to generate a synthetic model.

1. Introduction

One of the elements of public key cryptography systems such as Pretty Good Privacy^{TM} and GNU Privacy Guard is the need to guarantee the validity and authenticity of public keys. As a solution, key servers dispense key trust information uploaded by key owners in the form of keys signatures. The trust information is inserted into the system based on each users belief that the key that they are signing is the one belonging to the intended user. These key servers are a significant source of historical information as the public keys contain both identity and trust relationships. The common practice of limiting the lifetime of keys and signatures based on calendar time ensures that stale information can be identified. This allows us to view sets of key-rings within key servers as social networks.

Using the time-stamped data it is possible to trace the entry and departure of persons within the systems as well as the relationships connecting them. At each time stamp it is possible to compute a number of metrics and statistics on the new relationship or on the social network graph as a whole at that point in time.

We show that these metrics change over the lifetime of the network and that some of the more distinct changes are highly correlated with events relevant to the actors in the network. There was, for example, a distinct increase in the average previous-shortest distance between two newly-connected actors in a Debian mailing list immediately after a Linux conference.

This implies that static metrics are insufficient for analysing and describing the behaviour in this network, and provides general evidence that care must be taken when using only static metrics in analysing other such networks. Such metrics should be recomputed continuously and the temporal differences accounted for. Additionally, these results could be of benefit in modelling "realistic" synthetic social networks.

Further, we demonstrate that this network could, at any given timestep, consist of many disconnected components, on the order of the number of nodes in the network. This indicates that care should be taken when using algorithms or techniques which require the assumption that the network is connected, especially since in a dynamically built network components could easily be merging and splitting over time.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

The remainder of the paper is organised as follows. We start with a brief summary of graphs and social networks as well as a review of PGP^{TM}. Then, we describe in detail the two data sets that we focus on, one relatively small (extracted from the Debian developers key server) and one much larger (extracted from the U Alberta key server), as well as report some basic statistics on them. Dynamic networks were built up from those data sets so we next describe and report the various metrics and statistics measured throughout the life of these network graphs.

2. Background

2.1. Graphs and Social Networks

Graph theory is an old and well-studied field with a plethora of concepts and algorithms. For an introduction to graphs and graph algorithms see, for example, (Wilson, 1986).

Formally, a graph is a pair $G = \{V, E\}$ where V is a set of nodes and E is a set of edges which in turn are pairs of nodes (i.e. $E = \{e = (h, t) : h \in V \texttt{ and } t \in V\}$). Two edges are said to be *joined* if they share the same node between them (i.e. $e_i = (n_1, n_2), e_j = (n_2, n_3)$). A *path* between two nodes is a collection of consecutively joined edges that connect those two nodes. There are a variety of well known algorithms for determining the *shortest path* from one node to another. Finally, a *connected component* is a subset of a graph where every pair of nodes in that subset are connected by some path. If a graph is composed of only one connected component then the graph is said to be *fully connected*.

The idea of social networks is simple—to model social and sociological data using graphs. The idea probably first arose in the field of sociometry as a way of quantifying social relationships.

Interest in social networks has been around since at least the 1950's. Modelling collections of social actors as nodes in a graph and their relationships as edges provides a paradigm that has since been utilised in a variety of different areas, from studying the neural pathways of bacteria to analysing power grids(Watts & Strogatz, 1998). In 1967 Milgram (Milgram, 1967) formalised the "small world" property that seems to be present in many social networks. Given any two nodes in a small world, it is highly probable that those nodes are connected by a relatively short path. More recently, Watts' book(Watts, 1999) on the small worlds phenomenon seems to have sparked even more research in the area.

Initially, interest in social networks and small worlds was primarily focused on using the graph paradigm to model and analyse data. More recently, researchers have started looking at various methods of generating synthetic social networks on which a variety of algorithms can be tested. Typically, social networks that have the small world property are desired.

Interest in this small worlds property has translated into interest in a variety of different methods for evaluating a new relationship. Just before an edge is added to the network, the shortest path between the two nodes associated with an edge can be recorded. Presumably, if this previous shortest path is, on average, very low then the network will have the small worlds property (see for example (Kleinberg, 2000)). This leads to a useful tool in analysing social networks, as this metric is often easy to measure. Indeed there are a host of different measures that are associated with such new relationships, all of which are based in some way or another on the concept of measuring the path or paths that exist between two nodes before an edge relating them is added (see for example (Hannerman, 2001) for a summary of some of these measures).

These measures have proven useful for generation of synthetic social networks as well. Since these measures apply for any two nodes not yet joined by an edge they can be computed for all such pairs, then translated in a straightforward manner into probabilities yielding an obvious method for building a graph. By forcing these measures to be low, graphs with the small world property can be generated.

One thing to note is that some of these measures, as well as other social network algorithms, may require that the network be connected, either to guarantee a performance bound or, in some cases, to work at all.

2.2. PGP^{TM}

Pretty Good Privacy (PGP^{TM}) and variants (such as GNU Privacy Guard) are programs for encrypting and signing e-mail. They can be used to encrypt entire e-mail messages but more often are used to sign an e-mail as a way of guaranteeing that the e-mail is actually a product of the person who signed it.

PGP^{TM} uses the RSA (Rivest Shamir Adleman) public and private key crypto-system. Public key methods work by generating separate encryption (public) and decryption (private) keys in such a way that decryption of a message with the public key is nearly impossible. This allows mass distribution of the public key without concern. Anyone can encrypt messages but only someone with the private key can decrypt them.

PGP^{TM} can also be used to apply a digital signature to a message without encrypting it. This is normally used in public postings to allow others to confirm that the message actually came from a particular person. Once a digital signature is created, it is almost impossible for anyone to modify either the message or the signature without the modification being detected by PGP^{TM}.

In order to verify the signature of an e-mail, the public key is needed. Without key servers, people would have to distribute and find these keys themselves. To facilitate this process, key servers store the (public) PGP^{TM} keys and key certificates. Anyone looking for a public PGP^{TM} key can search for and retrieve it from the key servers (The key servers synchronise with each other—if someone adds a key to a key server it is distributed to all key servers).

Initially, a person must actively "sign" the key of another person (indicating that they trust that that key belongs to that person). However, once a person has signed someone's key, that key now becomes trusted by the first person. In this way it is possible to verify the validity of a particular key. A key is only trusted if it is signed.

These chains of signatures build up like a web, called the *web of trust*. This web-like structure is no accident. It is important to have as many disjoint paths as possible to reduce the chance that someone can fake a confirmation chain with a wrong signature.

Everyone who uses PGP^{TM} (or its variants) has a *keyring* of (mostly) valid public keys. Additionally, a trust value can be assigned to each public key indicating how much a person believes in the authenticity of the key. The validity of a key is can be determined by thresholding this trust value. Almost all of this data can be mined from public key servers.

3. The Data

GPG and PGP^{TM} key networks have a number of elements that make them interesting data sources for our purposes; several key analyses have been done in the past on trust relationships within key-rings with an eye at establishing the authenticity of keys and the reliability of the key signing process (e.g., (Blaze et al., 1996)). We pursue here a different approach in that we are not interested in the keys themselves as much as the relationships which they imply between the individuals within the key-ring universe.

The distinction is important in that different individuals may have multiple keys for multiple roles which have not been linked for historical or operational reasons. Hence, while historically the social distance within a group of individuals was calculated with respect to key signatures with authenticity as an objective, we only which to establish a reasonable expectation that a relationship does exist. We make an implicit assumption that the process used by people to determine key trust is directly linked to the strength of the relationship between the two people and not on a particular relationship between two specific keys.

The keys contain a free form identifier string that is set by the key owner. By a loose convention, this is usually composed of the email address ("John Doe <johndoe@somewhere.com>") of the key owner along with a brief longhand description ("Work place software distribution key").

The keys were then pre-processed to resolve individuals to their public keys, even if an owner-to-owner signature between them was missing. To do this the email labelling data was used using an m-to-n merge: keys having multiple email addresses where matched with keys labelled with those same email addresses. This ensured that we could obtain an unique identifier for each person within the database.

Signatures between keys were assumed to indicate a friendship between individuals. This assumption can be challenged in that key signatures are granted on an opportunistic basis that may not be completely based on friendship, as much as social access. This may explain with some individuals in key networks have a disproportionate 'friend' network that is not reciprocated. While a metric for the level of trust accorded to each key was available, we choose not to make use of it in this research.

Using the time-stamps we then tracked the evolution of the social network from the addition of the first node to the end of the data collection period. There are four possible changes that can occur in the network:

1. Node (person) addition (key creation)
2. Node (person) removal (key expiry/revocation)
3. Edge (relationship) addition (signature creation)
4. Edge (relationship) removal (signature expiry, signature revocation)

We thus labelled the identifying and friendship data with time-stamps. This was done to prevent stale social information from flooding our analysis network. Individuals and relationships were temporally removed from the dataset where their underlying keys and signatures where cryptographically revoked or expired.

Because in the vast majority of cases no expiration date had been set for the keys, we applied a timeout period of one year after the last sign of activity (key creation or signing) from the user.

We extracted data from the following two key servers and created social network databases from them. The key rings were::

The Debian key-ring: The developers key-ring for the Debian distribution project was used as a small data set, using data captured as of July 5th, 2004. The key-ring as used by the GPG engine is about 10MB large.

The U Alberta key server key-ring: The key-ring of the U Alberta key server was used as a large data set, using data captured as of May 27th, 2005. The key-ring as used by the GPG engine is about 4GB large.

The Debian key-ring has about 1465 unique individuals within it, on average an individual has 1.6 keys. The U Alberta key server key-ring has about 830,000 unique individuals in it, each with an average of 2.38 keys. We hypothesise that this increase over the Debian data set is a result of the longitude of the key server data set. Within the Debian key-ring, there are about 17,912 keys which sign Debian maintainer's keys but are not part of the key-ring.

Furthermore, the email address enabled us to perform linking to other data sources. In the case of the Debian server, we were able to link the debian-devel and debian-project mailing list used by debian developers and extracted the social network information from it for comparison to the GPG key network. We matching email addresses to the individuals already linked to the GPG network and added new entries for people that were not.

4. General Network Properties

Figure 1 shows the growth of the number of individuals (nodes) and relationships (edges) in the social network incrementally built up using the Debian key-ring data.

Figure 2 shows the same growth for the social network created from the U Alberta key server data. We found that the large world of the U Alberta key server key-ring behaves in a manner similar to the bow-tie structure observed with the world wide web (Broder et al., 2000). This bow-tie is composed of a core "knot" of relationships in the middle. This core is referred to by a large number of persons that are not in turn referred to (the left part of the bow-tie). The core also refers to a large number of people who do not refer back to the core (the right part of the bow-tie). Finally, there are a large number of "smaller worlds" unconnected to the rest of the key-ring (the lint).

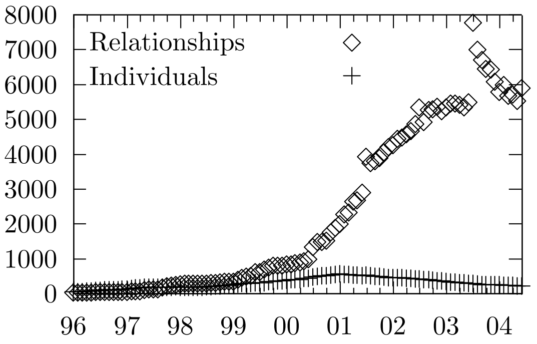

Figure 1. Population and relationships within the Debian key-ring over time.

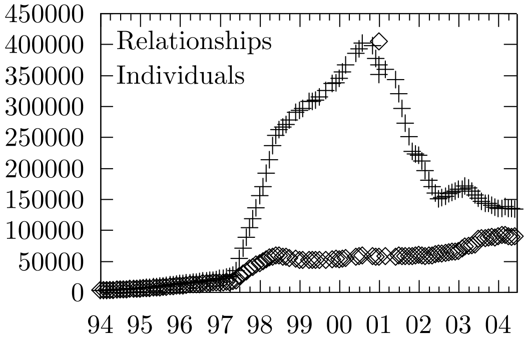

Figure 2. Population and relationships within the U Alberta key server over time.

A measure of the overall connectivity was computed for both data sets over time by picking a random individual and attempting to find a path to another random individual within the data set. By computing the number of successful paths computed for each pair, the overall connectivity for each key set was tracked.

Interestingly, as the size of the world grows, the actual social distance between people increases with the number of people within the key-ring. The connectivity of the system within the key-ring was measured by tracking the number of times a path could be found from point A to point B. overall the connectivity of the graph begins at 25% and drops to 3% when all individuals are within the world.

The Debian key-ring has a well-curated database and it's interconnectivity tends to converge to about .33 (i.e., about 1/3 of the time, a path can be found to another individual within the key server). In contrast, the overall connectivity of the U Alberta key server keeps lowering itself to about .01 with the passing of time. We hypothesise that this is a direct result of the intended purpose of both data sets. The Debian keyring is cleaned and maintained to support the Debian development process whereas the U Alberta key server is used as a database which is not trimmed. Old, obsolete or broken keys therefore accumulate and pollute the key server whereas extra and/or useless information is pruned from the Debian key-ring. A large part of the problem comes from the widespread utilisation of keys with no expiration information and which remain for an excessive amount of time.

Figure 1 shows a comparison of the number of nodes and the number of connections in the Debian network over time. Note that the number of connected components increases in the same manner as the number of nodes. This provides evidence that in certain social networks, the number of connected components continuously vary over time. In such networks, caution is required not only when making the assumption that the underlying graph is connected but even when making the assumption that there are a constant number of connected components.

As previously mentioned, we also made use of two main debian mailing lists to compare against the GPG social network. Out of the 17,305 individuals that posted to the mailing list, only 806 were part of the GPG social network. There exist many explanations for this difference, which may include the curation process that occurs with the debian keyring and one-time posts to the mailing list.

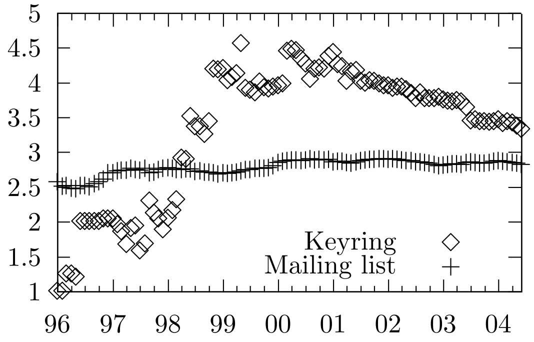

Figure 3. Comparison of the distance in both mail and keyring social networks.

Figure 3 compares the average network distances for both GPG and email data-sets. Interestingly, the email data-set rapidly converges to an average distance of slightly less than 3. We found this consistence surprising as we expected a higher amount of one-off postings and individuals within the mailing lists and thus a higher variation in the metrics. By inspecting the mailing lists we discovered that a number of the mailing lists contain a number of long running discussions between 2 or 3 individuals within the mailing lists. This explains the stability of the average social distance metric, however we are unsure of the reasons for the differences in the connectivity metrics between the GPG and mailing lists net that is plotted in Figure 4.

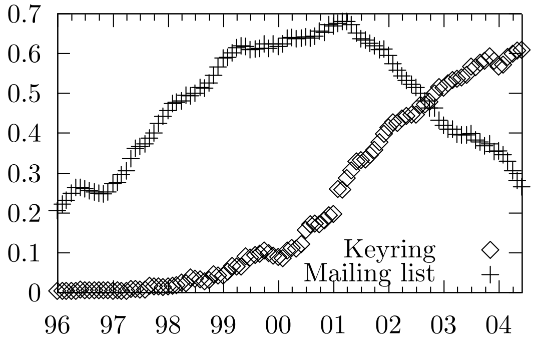

Figure 4. Comparison of the connectivity in both mail and keyring social networks.

By inspecting the graph, it becomes obvious that the mailing lists lead the GPG key network temporally. This is what we expected intuitively as posting to a mailing lists requires less preparation that creating a GPG key. Furthermore, the peak in mailing list connectivity also coincides with a number of Debian and Linux conferences already mentions. We thus propose that the GPG network is a restricted subset of the mailing list network that lags behind because of its formalised structure.

5. Relationship properties

For the Debian data set, the relationships follow a power law curve; on average each entity within the key-set would signal a relationship with about 13.8 other people on average.

For the key-ring data set the relationships still follow a power-law curve but the number of relationships has decreased to 1.93. We believe that the number of single individuals accounts for this difference.

There are 15,939 relationships between individuals declared within the Debian key-ring. Out of these, 5,515 are symmetric in nature in that the relationship is reciprocated by the signee. The rest are one-way key signatures where an individual signs another key without any acknowledging signature. One possible reason for this behaviour is the use of automated email key signing methods. A review of the relationships did not, however, yield any obvious indicators of this.

These asymmetric relationships are analyzed by Feld and Elmore (Feld & Elmore, 1982) who suggested that they are present because of logistical difficulties in interacting with other persons or because individuals may select individuals which their peers consider popular but whom they themselves do not know personally. This may have some significance for managing cryptographic and trust networks, as it indicates that trust may be asymmetrical.

Within the U Alberta key server data set there are 118,960 distinct relationships declared, of which 69,193 are asymmetric. There are more than twice as many (2.8 times) directed relationships as there are symmetric relationships. Anecdotal evidence seems to support the proposal made by Feld and Elmore (Feld & Elmore, 1982) that these specific asymmetric relationships are the result of social popularity and not actual acquaintance or social relationships. The most connected node within the key server data set is Phillip Zimmerman, the original author of the PGP^{TM} package. It is interesting that a 33% rule seems to be in effect—about 33% of all the relationships are asymmetric; this appears to be consistent with the results obtained from blogging data (MacKinnon & Warren, 2006; Kumar et al., 2004).

Earlier, we argued that a popular metric for analysing social networks is the shortest path length between nodes in the network. Figure 5 demonstrates how the average shortest path length changes over time in both data sets.

A typical use for this metric is for predicting which two nodes will be connected next in the development of the social network. When a new edge is added we are interested in the length of the previous shortest path between those two nodes (obviously after the addition of the new edge the length of the new shortest path will be one). If one can build a distribution over such lengths, it can be used to estimate the probabilities (for all possible pairs of nodes which are not already connected) that a particular edge will be added.

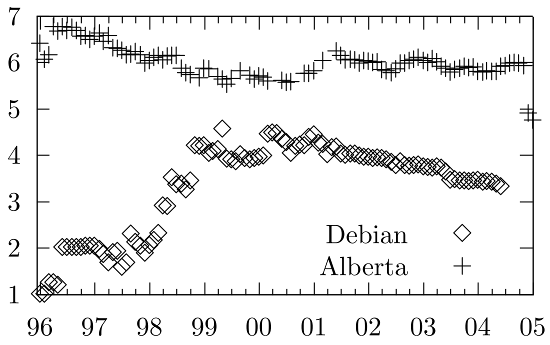

Figure 5. Social distance change over time in both data sets

Figure 6 shows a kernel density function displaying over time the average shortest path between two nodes in the Debian data set before a relationship is added connecting them (the x-axis is time and the y axis is shortest path between nodes). In other words, at a time where there is a peak, when a new relationship is added between two nodes, the average shortest path between them is longer than when there is a valley. To put it another way, the peaks correspond to times when the people (nodes) in the network reach out farther in the graph for new relationships. Note that because we used a normalized kernel density function to display this data, the scale of they y-axis has lost its units. However, this representation clearly demonstrates the relative differences in the model over time.

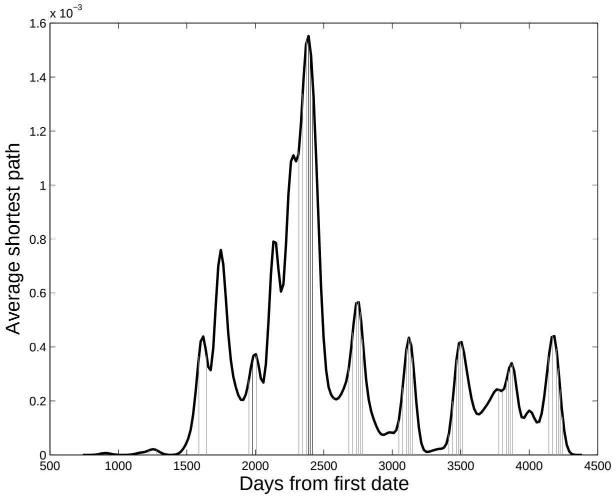

Figure 6. Average shortest paths

Note the large peak in 1999 and the periodic peaks roughly every year thereafter. We hypothesise that these peaks are explained by a number of Linux-based conferences—that contacts made while organising and attending the conference translated into key signatures. the conferences are as follows:

- Linux Expo 1995-1999, started 1995-06-26
- Linux World Expo, started 1999-08-09
- Linux Kongress 1994-2004, started 1995-05-01
- Linux Con Au 1999-2004, started 1999-07-09
- Linux Tag 1998-2004, started 1998-05-28
- Ottawa Linux Symposium 1999-2004, started 1999-07-22

The first edition of each conference is plotted in Figure 6 as a vertical black line. Subsequent editions are plotted as vertical grey lines. Clearly, the largest peak corresponds to the first edition of three of the conferences (Linux Con Au, Ottawa Linux Symposium and Linux World Expo). Also, note that there is a high degree of correlation between the remaining conference dates and the peaks in the kernel density function.

This has important implications for both analysis and synthesis of social networks. If we gathered data from an existing hypothesised social network we could easily create such a graph of shortest paths over time. If there were distinct peaks in such a graph, it is reasonable to hypothesise that they correspond to events relevant to the social actors composing the network. This provides an useful research tool which narrows down a set of time periods within which researchers can search for such events. For example, there are some peaks in Figure 6 that do not correspond to the Linux conferences listed previously. This could be indicative of some other similar event of importance to the debian community. If one has some reason to believe that such an event exists these peaks could be useful in narrowing down the timeframe where the event can be found.

Alternately, if we wish to generate a “realistic” but synthetic social network modelling people’s relationships through e-mail, we now have evidence that when determining which edges to add next as we build the graph, we should vary the probabilities of a possible edge over time to reflect the above behaviour. Perhaps by randomly generating times corresponding to important events where the probability of adding an edge between two more distant nodes should briefly spike as they do in Figure 6.

In the Debian data set the probability that a relationship joined two previously unconnected components of the graph is about 0.33. Figure 7 shows a kernel density function displaying this probability over time.

This behaviour in social networks can reasonably be interpreted in some sense as two separate groups making contact for the first time. This has important ramifications in such applications as the study of the spread of forest fires, or disease vectors.

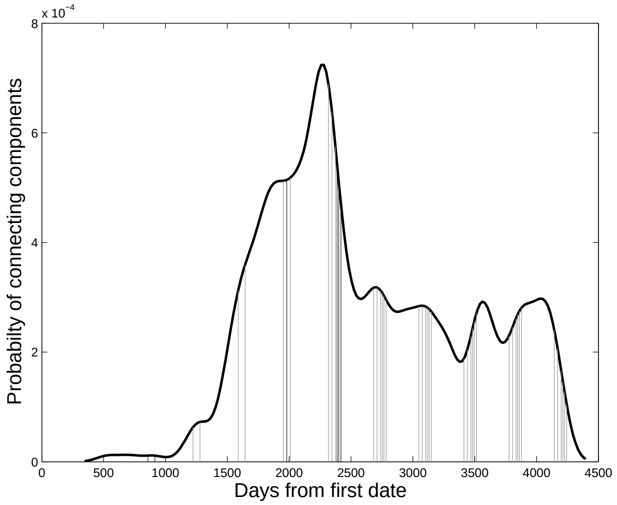

Figure 7. Average number of added edges that previously connected unconnected components.

Note that again there is some correlation between the largest mode in Figure 7 and the Linux conferences listed previously (the conferences are again plotted as vertical lines). More evidence that properties associated with social networks can vary significantly over time and thus should be tracked in a dynamic fashion.

Finally, an element of the GPG dataset that we found especially interesting is the insight into the privacy behaviour of individuals that it provides. As stated earlier, signatures between keys are required to ensure key authenticity and thus people tend to acquire signatures on an opportunistic basis for their own key.

As has been noted by Borisov et al (Borisov et al., 2004), this mechanisms has privacy implications, indeed we have used it in this paper to acquire individuals’ partial social networks. To an extent this constitutes a weakness of the system as it reveals a great deal of information to outside observers.

In the generic case, the individual makes use of his relationships to acquire signatures to solidify the authenticity of the public that he distributes. This ensures that a ’trusted’ signature path exists between the sender’s own key and the recipient’s key, as the cost of exposing the social path between both persons.

An alternative, used to prevent information from being revealed, is to only sign ones own keys as they expire or get compromised. Provided that an alternate means of securing the distribution of the public key, this effectively prevents the release of social information to the keyserver.

A final solution used is the total dis-use of the signature mechanism by the user. While in a minority, these users tend to provide limited, cryptic labelling of their key to prevent the attribution of their messages.

6. Conclusion

We demonstrated how to build a social network using publicly available data from PGP^{TM} key servers—data which is ideal for the straightforward creation of a highly dynamic network. Next, we showed that two common small world related social network parameters, number of connected components and previous shortest path before a relationship, can change significantly over the life time of the network. Finally, we provided evidence that these changes can be related to events of interest to the actors in the social network (in the case of our data sets, the events were Linux conferences). This indicates that such dynamic analysis of these parameters could be useful in analysing other social networks as well as possibly providing better algorithms for generating synthetic networks.

There are two obvious directions for future work. The first is to see whether other social networks exhibit the behaviours shown in this paper. It is unlikely that the network of e-mail relationships is unique in this respect, but testing in other domains should be performed. The second is to use knowledge about the dynamic parameters to try and generate synthetic social nets and to see if these social nets are more “realistic” than those generated by static parameters. Specifically, we hypothesise that varying these parameters in a periodic manner will lead to an increase in the robustness of a generated network to disruption.

References

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). Decentralized trust management. *IEEE Symposium on Security and Privacy* (pp. 164–173).

Borisov, N., Goldberg, I., & Brewer, E. (2004). Off-the-record communication, or, why not to use pgp. *Workshop on Privacy in the Electronic Society.*

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., & Tomkins, A. (2000). Graph structure in the web: Experiments and models. *9th World Wide Web Conference.*

Feld, S. L., & Elmore, R. (1982). Patterns of sociometric choices: Transitivity reconsidered. *Social Psychology Quarterly*, *45*, 77–85.

Hannerman, R. A. (2001). *Introduction to social network methods.* Department of Sociology, University of California.

Kleinberg, J. (2000). The Small-World Phenomenon: An Algorithmic Perspective. *Proceedings of the 32nd ACM Symposium on Theory of Computing.*

Kumar, R., Novak, J., Raghavan, P., & Tomkins, A. (2004). Structure and evolution of blogspace. *Commun. ACM*, *47*, 35–39.

MacKinnon, I., & Warren, R. H. (2006). *Age and geographic analysis of the livejournal social network* (Technical Report CS-2006-12). School of Computer Science, University of Waterloo.

Milgram, S. (1967). The small world problem. *Psycology Today*, 61–67.

Watts, D. (1999). *Small worlds: The dynamics of networks between order and randomness.* Princeton University Press.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*, 440–442.

Wilson, R. J. (1986). *Introduction to graph theory.* John Wiley & Sons, Inc.